Proceedings of Mining Minds Release Version 2.5

Research & Technical Specifications

Principal Investigator:

Professor Sungyoung Lee

April 30th, 2016

Ubiquitous Computing Laboratory

Kyung Hee University, Korea

Principal Investigator
Prof. Sungyoung Lee

Document Editor
Dr. Muhammad Bilal Amin

Contributors

DCL Team
Dr. Muhammad Bilal Amin (Team Lead)
Bilal Ali, Soungho Cho, Usman Akhtar, Taqdir Ali, Jinhyuk Gong, Mao Bui

ICL Team
Dr. Wajahat Ali Khan (Team Lead)
Taeho Hur, Jaehun Bang, Thien Huynh The, Dongwook Kang, Sangbeom Park, Muhammad Asif Razzaq

KCL Team
Maqbool Hussain (Team Lead)
Taqdir Ali, Maqbool Ali, Sangho Lee, Dohyung Kim

SCL Team
Muhammad Afzal (Team Lead)
Rahman Ali, Syed Imran Ali

SL Team
Shujaat Hussain (Team Lead)
Mahmood Ahmad, Jamil Hussain

Facilitators
Professor Byeong Ho Kang, University of Tasmania, Australia

Document History

Created By:	Dr. Muhammad Bilal Amin	**Last Updated By:**	Dr. Muhammad Bilal Amin
Date Created:	July 13, 2015	**Last Revision Date:**	Apr 15, 2016

Table of Contents

Section 1

Mining Minds
Introduction-Publication

RESEARCH

Mining Minds: a Novel Digital Health and Wellness Framework for Personalized Support

Oresti Banos[1], Muhammad Bilal Amin[1], Wajahat Ali Khan[1], Muhammad Afzal[1], Maqbool Hussain[1], Byeong Ho Kang[2] and Sungyong Lee[1*]

*Correspondence: sylee@oslab.khu.ac.kr
[1]Department of Computer Engineering, Kyung Hee University, 1732 Deokyoungdae-ro, Giheung-ug, 446-701 Yongin-si, Korea
Full list of author information is available at the end of the article

Abstract

The provision of health and wellness care is undergoing an enormous transformation. A key element of this revolution consists in prioritizing prevention and proactivity based on the analysis of people's conducts and the empowerment of individuals in their self-management. Digital technologies are unquestionably destined to be the main engine of this change, with an increasing number of domain-specific applications and devices commercialized every year; however, there is an apparent lack of frameworks capable of orchestrating, and intelligently leveraging, all the data, information and knowledge generated through these systems. This work presents Mining Minds, a novel framework that builds on the core ideas of the digital health and wellness paradigms to enable the provision of personalized support. Mining Minds embraces some of the most prominent digital technologies, ranging from Big Data and Cloud Computing to Wearables and Internet of Things, as well as modern concepts and methods, such as context-awareness, knowledge bases or analytics, to holistically and continuously investigate on people's lifestyles and provide a variety of smart coaching and support services. This paper aims at comprehensively describing the efficient and rational combination and interoperation of these technologies and methods through Mining Minds, while meeting the essential requirements posed by a framework for personalized health and wellness support. Moreover, this work presents a realization of the key architectural components of Mining Minds, as well as various exemplary user applications and expert tools to illustrate some of the potential services supported by the proposed framework.

Keywords: human behavior; digital health; dHealth framework; quantified self; wearable sensors; big data; cloud computing; context-awareness; knowledge bases; user experience

1 Background

Healthcare systems are facing unprecedented financial limitations at a time of rising demand for their services [1]. The magnitude of these constrains makes utterly necessary to change current care models in a bold manner, from late disease management to preventive personalized health, involving a major shift in when, where and how care and support is delivered to each particular patient and service user [2]. In fact, it is generally recognized that most prevalent diseases are partly caused or aggravated by lifestyle choices that people make in their everyday life. Unwholesome diets, tobacco use and sedentary conducts, among other unhealthy habits, potentially contribute to develop severe illnesses [3, 4] and also limit the effectiveness of medical treatments [5]. Thus, enabling people to make healthier choices, to be more resilient, and to deal more effectively with illness and disability when it arises turns to be a fundamental part of this necessary new health perspective.

Information and communication technology is called upon to be a cornerstone of the new health era, playing a crucial role in empowering people to take charge of their own health and wellness, by providing them timely and ubiquitously with personalized information, support and control [6]. In fact, an extraordinary interest has been lately shown by the industry in the development of specific applications and systems for health and wellness management, particularly boomed by the growth of wearable and mobile technology [7]. The immediate targets of these solutions are healthy lifestyle services, especially oriented to the fitness domain, which primarily allow to track primitive user routines and provide simple motivational instructions. For example, mainstream commercial systems such as Withings Activite [8], Garmin Vivofit [9], Fitbit Surge [10] or Misfit Shine [11], which consist of sensorized bracelets and gadgets normally accompanied by mobile apps, provide some basic healthy recommendations based on the measured taken steps or slept hours. More prominent health and wellness systems have been shown at the research level, for example, to alert on physical conditions [12] or detect chronic illnesses [13], yet most of them are prototypes or work-in-progress. Some of these systems also provide educational modules and personal coaching for promoting healthier lifestyles and managing health conditions [14]. Despite their interest, main limitations of these solutions refer to misperformance, limited scope and lack of interoperability with other similar systems and applications.

To overcome the shortcomings of application-specific solutions and leverage the potential of health information systems in a wide sense, general frameworks capable of managing these resources are required. A few attempts are found in this respect in the literature, for example, in [15] a middleware framework integrating multiple interfaces and multiparameter monitoring of physiological measurement is presented. In [16], distributed signal processing algorithms for the analysis and classification of sensor data are provided as part of a framework for rapid prototyping of body sensor networks. A mobile platform to collect users' psychological, physiological and activity information for mental health research is presented in [17]. The authors of [18] propose a healthcare platform particularly devised for interfacing and processing data from body-worn physiological sensors and home appliances, with a proven utility in daily medication management. A novel framework that provides advanced functionalities for resource and communication abstraction, wearable health data acquisition and knowledge extraction is introduced in [19]. Most visible initiatives are especially being underpinned in the mobile health domain. That is the case of [20], an open mobile health project to help developers produce digital health data as useful and actionable as possible. Google Fit [21] by Google, SAMI [22] by Samsung or HealthKit [23] by Apple are examples of new commercial platforms also devised to integrate and share users health data among diverse health and wellness applications.

Despite important contributions have been made through these platforms, there is still much room for improvement. For example, most mobile health frameworks are bound to the computational capabilities of the smartphone, require continuous maintenance and updates of end-user applications and normally trap data into their devices. Moreover, multiple systems and applications can generate similar health data and outcomes leading to unnecessary redundancy and overcomputation. These systems mostly operate on-demand, thus determinants of health and wellness states can be also lost if not registered in a continuous manner. Platforms devised to share and integrate health and wellness data underutilize cloud resources while simply using them for storage. In the light of these limitations we present Mining Minds [24], an innovative distributed framework that builds on some of the most prominent digital technologies to enable the provision of personalized healthcare and wellness support. This framework is particularly devised to seamlessly investigate on people's behavior and lifestyles in an holistic manner through mining human's daily living data generated through heterogeneous resources. Mining Minds aims to innovatively exploit the potential of cloud computing not only for storage but also for high performance computation supporting the discovery of personal and public health and wellness patterns, of primal necessity to facilitate proactive and preventive support.

2 Requirements of a Digital Health and Wellness Framework

Diverse types of data are normally required to neatly describe a person's health and wellness state, ranging from physical -sensory- and logical -personal profile and interests-, to social -human relations- and clinical -medical- data. Many technologies are increasingly available for the collection of these data, such as wearable devices, ambient sensors, social networks or advanced clinical systems. Thus, an important requirement of a digital health and wellness framework is to provide a certain level of abstraction from heterogeneous resources to make their utilization transparent to the user. Health and wellness data go beyond standardized structured formats such as "traditional" electronic health records, particularly including other multimedia and unstructured data. Therefore, another primal requirement is to be capable of dealing with this dimension of heterogeneous data, as well as the underlying implications of the management of structured, semi-structured and unstructured data. Not only data variety constitutes a key factor, but also data volume. Massive amounts of data are generated over time on and around the subject with the advent of new sensing and multimedia technologies. Accumulating and digesting these amounts of data are not trivial tasks, and need to involve sophisticated processing and storage mechanisms to enable the persistence and availability of the data. Similarly, the rapid pace of data generation makes necessary to also take into account data velocity as a reference factor. This proves to be especially challenging when referred to data that represents real-time regular monitoring, such as continuous electrocardiogram measurements or body motion data. Another important concept that applies to health and wellness data is veracity. Different data types may represent similar concepts or contradict each other, or even be of little interest. Therefore, digital health and wellness frameworks should count on governance mechanisms to determine the consistency of the data, ensuring it is certain, meaningful, clean and precise.

Extracting the determinants of health and wellness is a very challenging task that requires more than simply collecting and persisting personal data. Accordingly, digital health and wellness frameworks must include automatic intelligent mechanisms to process person-centric data and extract interpretable information and insights for ensuring a personalized health and wellness support. Moreover, insights should not only be gained from individual users but from the collectivity. Thus, another important requirement consists in the application of advanced techniques to process information in “de-identified” form to enable population management and deeper insights into cause and effect. These insights can be particularly leveraged by health and wellness care systems to extend, adapt and evolve the knowledge provided by human domain experts.

Health and wellness information and knowledge are principally devoted to support advanced care services. Mechanisms such as alerts, recommendations or guidelines are particularly used as services to catalyze both information and knowledge to be delivered in a human-understandable fashion to users and stakeholders in general. However, most digital health and wellness systems only support general services that do not differentiate among people particular needs or interests. Therefore, an important requirement is to provide services that operate on a person-centric manner. To do so, expert systems are required, for example, to precisely map user needs to the best possible recommendations, personalize the recommendations explanation or customize the mechanisms for the communication of these recommendations.

Users of health and wellness systems may be of a very diverse nature and play different roles. For example, busy patients may require to get a quick glimpse of their health conditions, fitness enthusiasts wish to observe a detailed description of their vitals and clinical experts be interested in an “in-depth” description of both health and wellness outcomes of multiple people. Accordingly, user interfaces need to be customized to the needs of each particular subject. Similarly, the user experience is of worth consideration. Users perceptions of system aspects such as utility, ease of use and efficiency should be taken into account to provide the most personalized experience. In fact, the user experience is dynamic as it is constantly modified over time due to the person changing circumstances. Thus, user responses and behavior need to be continuously tracked to support a sufficient level of personalization that helps guarantee adoption and engagement.

Finally, as it may be obvious, but unfortunately not often considered, all the aforementioned requirements need to be neatly accommodated to user security and privacy principles. The necessity of privacy and security is crucial for systems that build over sensitive information, and further augmented when data and services are shared by multiple entities in a distributed way. Data ownership, malicious data usage, as well as regulatory and legal policies are important hindrances in the widespread use and acceptance of health and wellness care systems. Therefore, it is of utmost importance to neatly adequate privacy, security, protection and risk management measures to all the processes concerned in a digital health and wellness framework.

3 Mining Minds Architecture

In the light of the aforementioned requirements we present here “Mining Minds”, a novel framework aimed at comprehensively mining human’s daily life data generated from heterogeneous resources for producing personalized health and wellness support. Mining Minds philosophy revolves around the concepts of data, information, knowledge and service curation, which refer to the discovery, processing, adaptation and evolution of both contents and mechanisms for the provision of high quality support services. Motivated by these concepts, a multilayer architecture is particularly devised for Mining Minds (Figure 1). In a nutshell, the Data Curation Layer (DCL) is in charge of processing and persisting the data obtained from the Multimodal Data Sources (MDS), which abstractly defines the possible sources of user health and wellness data. This includes, but is not limited to, data from social networks, questionnaires, wearable biomedical devices or ambient intelligence systems. The data processed by DCL is primarily used by the Information Curation Layer (ICL) to infer low-level and high-level person-centric information. This information mainly describes the user context and behavior, and, to some extent, their physical, mental and social state. The information extracted by ICL is leveraged by the Knowledge Curation Layer (KCL) to nurture and evolve the health and wellness knowledge primarily created by human experts. Data, information and knowledge are used by the Service Curation Layer (SCL) to create intelligent health and wellness support services, mostly in the form of smart coaching and support recommendations. All the contents and processes are accommodated in terms of security and privacy by the Supporting Layer (SL), which also provides analysis of user experience, feedback and trends to guarantee the highest personalization.

3.1 Data Curation Layer

DCL is responsible for acquiring, curating and persisting the data obtained from MDS so it can be processed for higher level understanding. To that end DCL relies on two main modules, Sensory Data Processing and Curation, and Big Data Storage. Within the former, Data Acquisition supports the acquisition and synchronization of raw sensory data obtained from diverse sources, both in real-time and offline manner, as generic data streams. Due to the heterogeneous nature of the data, it is acquired asynchronously in real-time and temporarily cached in data buffers. These data buffers are initialized depending upon the number of data sources, i.e., each data source has a data buffer in the Data Acquisition component. All the data buffers are synchronized and communicated to ICL for the determination of the associated low and high-level contexts. In parallel, this synchronized data is stored in Big Data Storage for non-volatile persistence.

Upon receiving the context information determined by ICL, the context instances are curated by the Representation and Mapping component as a time-based log registering the detected human behaviors. This time-based log is termed as user Life-Log or simply Life-Log and persisted in the Intermediate Database for shareability with other layers and applications. The stream of life-log instances is analyzed by a monitoring component called Life-Log Monitor (LLM). The responsibility of the LLM is to perform time-based monitoring of the different attributes and variables hosted in the Life-Log, and support trigger-based mechanisms to notify SCL for the occurrence of an abnormal or special event related to a given user. These abnormal events normally represent risky or unhealthy behaviors and are here defined as “situation events” or “situations” in general, which are described through diverse constraints -e.g., age, gender, medical conditions- and monitorable variables -e.g., intensity of a particular activity and its duration-. Situation events can be generated both statically at design-time and dynamically at run-time upon request from KCL.

The life-log data persisted in the Intermediate Database is regularly synchronized with the Big Data Storage. Big Data Storage also provides read access to raw sensory and life-log data. In case of historic data required by SL for analytics or KCL for data-driven rule generation, Big Data Storage provides queries for data streaming and intermediate data generation. These queries can be customized on request and return the data based on the attributes selected by KCL and SL. Security and privacy components from SL are further involved in these processes to request authentication and data stream encryption before its persistence or sharing.

3.2 Information Curation Layer

ICL represents the Mining Minds core for the inference and modeling of the user context [25]. ICL is composed by two main modules, namely, Low Level Context Awareness (LLCA) and High Level Context Awareness (HLCA). LLCA is in charge of converting the wide-spectrum of data obtained from the user interaction with the real and cyber-world, into abstract concepts or categories, such as physical activities, emotional states, locations and social patterns. These categories are intelligently combined and processed at HLCA in order to identify more meaningful semantic representations of the user context.

LLCA is composed by four key components, respectively, Activity Recognizer, Emotion Recognizer, Location Detector and SNS Analyzer. The identification of the user physical actions is performed through the Activity Recognizer. This component may build on several sensing modalities as they happen to be available to the user, such as wearable inertial sensors, video and audio. The output of this component corresponds to elementary activity categories such as “sitting” or “walking”. The Emotion Recognizer is defined to infer user emotional states, such as “surprise” or “sadness”, by using video and audio data as well as more sophisticated sources exploring human physiological variations and responses. The user situation is determined by the Location Detector, which essentially builds on the data collected through indoor and outdoor positioning sensors, such as video and GPS, to specify the exact location of the user. The SNS Analyzer is in charge of processing the information generated by the user during their interactions in regular social networks, including posts, mentions, traces and even global social trends, in the form of both text and multimedia data. From here, personal and general interests, conducts and sentiments may be determined. All these components require compatible multimodal sensory data to operate. The provisioning of the necessary data is performed through the Input Adapter, which receives and routes the data curated by DCL to each LLCA component depending on its nature. Once new low-level context categories are identified after the analysis of this data, the Output Adapter serves them to DCL for persistence and to HLCA for further processing.

HLCA makes use of two components, namely, High-Level Context Builder and High-Level Context Reasoner, to represent, verify, classify and categorize the user high-level context. The context representation and verification is performed through ontologies, adopted in the past as a unified conceptual backbone for modeling context, while its classification and categorization is done through ontological inference and reasoning. Whenever new information is received from LLCA, a new ontological instance is created by the High-Level Context Builder and categorized into one of the considered high-level contexts by the High-Level Context Reasoner. Thus for example, based on the actual time (e.g., midday), location (e.g., restaurant) and inferred activities (e.g., sitting), this component can determine the precise user context (e.g., lunch).

3.3 Knowledge Curation Layer

KCL is devised to enable the creation and evolution of both health and wellness knowledge. The knowledge is created either by the domain expert or knowledge engineer, by using expert-driven or data-driven approaches. The Expert-Driven module provides a set of rule authoring components to allow specialists to describe in a logical form causes or premises and effects or conclusions, e.g., "if gender is male and age lower than 65 then activity level should be moderate". The authoring process is further supported through evidence materials and domain vocabularies to confirm the viability of the rules and facilitate their elaboration. The Data-Driven module leverages the contents of the life-log for the automatic generation of rules. To that end, a data broker interface is defined to glean the contents of interest from the data persisted in DCL based on the features or factors established by the expert, e.g., "gender, emotional state and activity level". The process is automated by selecting and learning diverse mining models to discover and represent the underlying relationship among the considered health and wellness factors.

In both expert-driven and data-driven cases the generated rules are verified in terms of consistency and validated to avoid potential violations or redundancy with existing rules prior to be stored into the Knowledge Bases. KCL rules are not only persisted in traditional knowledge bases but also indexed according to salient conditions of these rules, also called "causes" or "situations". These situations refer to particular attributes of the rules than can be monitored by the platform and used for triggering the execution of specific rules. Accordingly, during the rule creation process the expert can select these condition attributes for their particular monitoring at DCL as explained in Section 3.1. The categorization of the knowledge bases through these indexes is particularly considered to enhance the performance of the reasoning processes hosted in SCL. In fact, once a situation is detected only its associated rules are shared with SCL upon request of this layer.

The evolution of the knowledge is procured through two main mechanisms. On the one hand, the expert creation process can be considered as a sort of maintenance per se. In that view, rules may be dynamically updated or replaced based on new health and wellness findings from experts. On the other hand, rules can be added, replaced or modified through the data-driven approach while using new life-log contents collected from different users.

3.4 Service Curation Layer

SCL provides the means to transform the data, information and knowledge curated by DCL, ICL and KCL into actual health and wellness support services. The services are managed by the Service Orchestrator, in charge of attending the potential requests, invoking the necessary services and coordinating the processes involved in the curation of the services. The requests may be of various types, i.e., scheduled on time (e.g, "every day at 8 am"), triggered by direct user queries ("suggest me an exercise plan for today's workout") or based on events (e.g., "user arrives at home"). The last type of request particularly relates to the concept of situation, already described in previous sections. The idea is that the LLM component from DCL triggers SCL once a situation event is identified in order to generate a new recommendation for the user.

The services needed to satisfy a given request are invoked from an extensible catalog containing reference and auxiliary services. A major reference service is devised for this architecture for the generation of personalized health and wellness recommendations. This service consists of two parts. First, generalized recommendations are developed by the Recommendation Builder component through reasoning on the user profile and life-log data provided by DCL and the knowledge facilitated by KCL for the specific domain of the service. In the case of handling a request derived from a situation detection the indexed rules hosted by KCL are particularly employed. Second, the recommendations undergo a personalization process through the Recommendation Interpreter component in order to deliver the one that best fits the user interests and demands. In here, all the potential recommendations are filtered based on the user preferences, conditions and possessions, as well as their actual context. Thus, for example, when the objective of the recommendation is to encourage the user to exercise, cycling would be avoided if the user does not own a bike, or a visit to the regular gym omitted in case the person is on a business trip. Prior to be communicated to the user, the recommendation is refined to be easily interpreted, for example, including multimedia contents to increase the interpretability and also incorporating motivational and engagement strategies to foster the user interest and attention.

3.5 Supporting Layer

The role of SL is to enrich the overall Mining Minds functionalities through advanced analytics, interactive and personalized UI/UX, implicit and explicit feedback analysis, and adequate privacy and security mechanisms.

The Analytics module is in charge of mining in a multi-dimensional and retrospective manner the data sets collected and curated from multiple users to reveal population health and wellness associations, patterns and trends. These trends may refer to current facts as well as expected or future tendencies. The exploration of present trends is performed through the Descriptive Analytics, which employs statistical techniques to relate explanatory variables of the persisted data. Thus for example, based on the analysis of the inferred people lifestyles, it can be found that there is a growing use of hot beverages among adolescents, which further relates to a dramatic increase of stress patterns. The discovery of potential future facts is carried out by the Predictive Analytics, which develops on the outcomes of the Descriptive Analytics to make forecasts by using regression and machine learning models. Descriptive and predictive analytics contents are organized by the Visualization Enabler, which adjusts the style of the information to be communicated to the users based on their expertise and role.

Evaluating the services supported by Mining Minds requires feedback from the users, which is here powered by the Feedback Analysis component. The sources of feedback may be of a diverse nature, ranging from explicit feedback provided by the user, for example, through questionnaires, to implicit feedback obtained from the user behavioral responses. Analyzing implicit and explicit feedback from the users is motivated by the aspects of functionality, content, and presentation. Functionality-based feedback refers to the findings obtained while comparing, for example, the system recommendations and the behavioral reaction of the user to those recommendations. Content-based feedback measures the user satisfaction with respect to the specific information provided as part of the delivered services. Finally, presentation-based feedback measures the human-computer interaction with respect to the user interface (UI), which is of particular utility to understand the user experience (UX). All these types of feedback are devised to help assessing the level of interest and adherence of users to the services provided through Mining Minds as well as to evolve and maintain the internal contents and processes handled by the platform.

Considering user preferences, habits or mood, the UI/UX module enables the end-user applications interface to be adapted accordingly. This adaptation is needed to adjust the human-computer interaction experience with respect to font size, theme, or audio levels, among other characteristics. Two main components are involved in this process. First, the UI Interaction Tracker collects the data from the interaction between the person and the application to analyze the user's ability to understand and use the system, e.g., the readability of the contents or the perceptibility of the controls. Then, the UX component measures the satisfaction level based on the analysis of the collected data. The immediate result is a dynamic adaptation of the UI based on the measurements extracted from the evaluation of the UX.

Given the sensitivity of the collected user data, privacy and security need to be assured and exhibited, not only for storage, but also during the processing and delivery of services. To that end, state-of-the-art cryptographic primitives along with indigenous protocols are considered. For secure storage, the AES standard is particularly used, whereas for oblivious processing, homomorphic encryption and private matching is used. Considering the intensive data flow between end-user applications and Mining Minds, data randomization techniques are used to ensure a high entropy for minimal leakage of information. An authorized model ensures the legitimate disclosure of personal data and services with users. Slow processing of information is a common byproduct of the encryption; thus, to assist partial swiftness to Mining Minds, sensitive and non-sensitive information is decoupled where required. Anonymization procedures are also considered to enable the use of the collected and mined users data by third party agents, e.g., for research purposes.

4 Mining Minds Implementation

An initial implementation of the proposed framework particularly oriented to promote healthy lifestyles and physical activity management is described here. Mining Minds is a distributed platform where the cloud environment plays a key role for supporting both persistence and limitless computational power. The Mining Minds implementation has been deployed over a hybrid cloud including Microsoft Azure public cloud environment [26] and a Xen private cloud [27] the for big data storage, which runs over Hadoop File System with MapReduce [28]. For better scalability and performance each layer is deployed over a separate virtual instance on Microsoft Azure. DCL, ICL, KCL and SCL are hosted on standard Microsoft Azure instances with Windows Server 2012 R2 as guest operating system [29], while SL functionalities partake of the others. The cloud-based deployment of layers allows the encapsulation of their responsibilities as well as the re-usability of their features through an inter-layer communication. This communication is implemented by establishing service contracts among the layers, which communicate by means of RESTful web services [30] and high performance sockets [31]. Communication between MDS and DCL is real-time and asynchnorous in nature. The most important service contracts are supported by DCL RESTful web services, which serve a data model with the structure of the Intermediate Database, here hosted by Microsoft SQL Server [32]. This data model is shared among the layers as an object model of service contract. The required data and information is populated by DCL and provided as responses to the upper layers. A high performance socket-based implementation is particularly used for DCL-ICL communication for the transference of sensory data and context determination in real-time, and communication between DCL and the big data storage on private cloud.

To support active lifestyle services in this version ICL only implements the Activity Recognizer. This component consists of various steps that mainly combine signal processing and machine learning techniques to define a specific human activity recognition model, here capable of distinguishing among various commonplace activities [33]. The main input of this model is body motion data, namely, acceleration, which can be broadly obtained from smartphones and wearable inertial sensors. Acceleration is preferentially used here since it is the most prevalent sensor modality in standard activity recognition approaches [34]. A non-overlapping sliding window of three seconds is used for the data segmentation [35], and time and frequency features extracted for their discrimination potential [36]. The implemented model combines Support Vector Machines and Gaussian Mixture Models for the classification process, which have been demonstrated of particular utility in this domain [37, 38]. The developed Activity Recognizer further supports two operation modes depending on the available data registered from the user. Specifically, a hierarchical approach is developed so that the model can determine the user activity based only on the inertial data collected through the smartphone or a combination of smartphone and smartwatch data if the latter is available.

Health and wellness knowledge is defined by medical experts and hosted in the Knowledge Bases of KCL. To that end, a simple rule authoring tool [39] is considered for the rule creation. Evidences and domain vocabularies are particularized to the definition of physical management and activity promotion plans [40]. SCL processes the contents generated by DCL, ICL and KCL for the generation of personalized activity recommendations. After a request is processed by the service orchestrator, generalized recommendations are produced by applying rule-based reasoning [41] on the existing knowledge and user data. User health and wellness data is transformed into a proper input query by using auxiliary services hosted in the service catalog. Similarly, auxiliary services are implemented for user goal discovery, e.g., ideal weight and calories to be burned per day [42]. During the reasoning, the interpreter analyzes each rule in the knowledge bases and fires the appropriate rules using a forward chaining procedure [43]. Recommendations are personalized by using content-based filtration techniques [44] employing user personal activity level and preferred physical activities.

Security and privacy components of SL are distributed among the different layers. Encryption techniques are employed to withstand any compromise on data storage facility or its unauthorized acquisition, as well as to make health-related data processing and evaluation HIPAA compliant. Concretely, AES [45], private matching [46] and anonymization [47] have been chosen to support the encryption. Moreover, since the systems are deployed on public clouds, processing over direct encryption without losing accuracy is required. The indigenously proposed system of oblivious term matching [48] is considered to that end.

5 Health and Wellness Promotion Services

Various exemplary applications and tools have been developed to showcase some of the potential health and wellness services supported by Mining Minds (Figure 2). Personalized weight management is procured through an application that promotes activity routines customized to the user characteristics and preferences in order to attain a healthy weight. The app further provides the person with valuable information regarding their physical behavior, energy expenditure and weight loss patterns. Behavior change and healthy lifestyle promotion is intended through a personal coaching application which delivers action recommendations and educational facts upon detection of unhealthy physical conducts. Conversely to other digital health and wellness systems and platforms, Mining Minds is not only devised to support regular users or patients but also specialists. Medical experts are facilitated with a comprehensive tool to inspect users behavior, engagement and satisfaction in a continuous and retrospective manner. Apart from diverse statistics reporting personal goals, achievements and physical activity patterns, the tool allows the specialist to check the specific information and recommendations delivered by the platform to each particular user. Finally, an intuitive rule authoring tool has also been developed to enable the creation and management of the health and wellness knowledge exploited by Mining Minds. The main features and utilities of these applications and tools are described next.

5.1 Personalized Weight Management App

A poor estimation of calories and activities as well as an unrealistic definition of milestones represent two of the most common reasons for failure in most weight loss programs. Accordingly, the main objective of this service is to empower people in the control of their weight through a continuous track of exercise and energy consumption and a personalized physical routine promotion to achieve the expenditure goals. Users are initially requested to sign up into the application by entering their personal information such as demographics -age, gender, weight and height-, preferences in terms of activities and exercise level -sedentary, moderate or intense-. All this information is securely stored and processed by the Mining Minds platform to calculate the user physical state, ideal weight, as well as the calories to be burned every day, all displayed for simple access on the app main dashboard (Figure 3.a). The amount of calories burned by the user on the present day is also displayed in this view. This value is estimated by the platform by analyzing the user activity patterns. To determine these patterns, Mining Minds elaborates on the acceleration data measured by the user smartphone, which is timely streamed through WiFi or 4G to the platform. To promote the user activity to achieve the daily calorie goal, exercise recommendations are given in an easy-to-understand manner. The recommendations contain precise indications on the duration of the activity and its execution style as well as motivational statements for the sake of encouragement. The recommended activities, their duration and intensity are personalized to each individual based on their profile. The evolution of the user actual weight with respect to the planned one is presented in a different frame (Figure 3.b). Here the user can easily self-report their current weight upon timely request of the platform. Other supportive features of the application provide the user with statistical analysis of burned calories and activity patterns (Figure 3.c) and a calendar view of the user comportment (Figure 3.d), specifically devised to support users in their self-monitoring and control.

5.2 Physical Lifestyle Coaching App

Behavior change and healthy lifestyle promotion constitute central objectives in public health interventions. The service defined here explores sophisticated coaching mechanisms to raise people's health awareness while inducing wholesome activity habits, changing unhealthy routines, and educating on healthier physical lifestyles. To that end the developed application continuously captures the user's body motion data registered through the inertial sensors of the smartwatch and smartphone. The data is then streamed to Mining Minds which processes it to infer the user behavior and determine potential risk or unhealthy situations. After an unhealthy behavior is detected (e.g., "one hour of continuous sitting") the platform automatically generates a personalized physical recommendation or healthy educational fact (e.g., "stretch your legs, arms and back"). Recommendations and facts are conveniently delivered according to the user context and availability, and displayed on the application main screen in a timeline view (Figure 4). Both recommendations and facts are also accompanied by multimedia contents -video, images and audio- to instruct the user on how to follow them as well as to attract and increase their interest and understanding. Moreover, users can value the delivered recommendations and facts according to their experience -"likes"/"dislikes"- and also provide comments on them -e.g., "I cannot carry out the recommended stretching exercises" or "My back hurts when I bend my waist"-. This information constitutes a key source of feedback for experts and Mining Minds itself to realize the comprehensiveness, applicability and impact of the services delivered by the platform.

5.3 Behavior Inspection Tool

Intelligent monitoring and smart coaching mechanisms are not planned to replace the role of specialists but rather complement it. In fact, the idea is that not only patients but also medical experts can benefit from the data, information, knowledge and services handled by Mining Minds. The expert inspection tool developed here is particularly devised to facilitate and expedite the task of health and wellness counseling specialists. The tool (Figure 5) presents in an intuitive yet comprehensive fashion some of the most prominent user-centric information managed by the platform. On the left side of this expert view the specialist can check the recommendations and facts delivered by Mining Minds to the user, the reason behind these suggestions as well as the feedback provided on them. On the right side, diverse sort of analytics describing the physical achievements of the user, their behavioral patterns and their rating of recommendations and facts are shown. Energy expenditure achievements and physical activity patterns are displayed in a daily, weekly and monthly basis, thus providing the expert with a detailed view of the user past and present status. The user feedback analytics is of particular interest to help experts identify what kind of recommendations and facts are more positively valued and which ones may not be accepted. The tool is also incorporated with a feature that allows the specialist to directly communicate with the user through the apps by sending comments in the form of notifications. By using this tool experts can in principle deal with more users while reducing the time required for the assessment of their progresses and evolution.

5.4 Rule Authoring Tool

Health and wellness experts are not only consumers of the services supported by Mining Minds but also content producers. The creation and management of Mining Minds health and wellness knowledge is handled by the specialists through an advanced rule authoring tool (Figure 6). This rule authoring tool is an adapted version of a prior one first introduced in [39]. The rule authoring tool provides domain experts with an easy to use dashboard to manage the existing rules, thus making possible their addition, update or deletion. An intuitive environment is provided for the creation of new rules and associated meta-information. The rule authoring tool incorporates a sophisticated physical activity wellness model which incorporates multiple domain concepts and vocabularies that facilitates the rule creation task. The tool is also equipped with intellisense technology to expedite the rule creation process and reduce the chance of errors. After the rule is created, the expert can simply save it, thus making it available for its use in Mining Minds.

6 Evaluation and Discussion

A preliminary evaluation of the implemented version of the platform and services is performed here. An important asset of the platform refers to the curation and persistence of sensory data by DCL. Most health applications delete sensory data after processing it; however, persisting this information is of worth for generating datasets that can be used to evolve the knowledge models or learn new ones. To benchmark DCL capabilities, the accuracy and performance of the platform in the collection, processing and storage of the sensory data is measured here. To that end, continuous data service calls over the period of 24 hours are generated and evaluated. The accuracy is measured by the rate of missing data packets, here summarized in Table 1. The results show a very low error, 0.06% in average, which means that practically all the sensory data sent to the platform is safely processed. The performance, depicted in Figure 7, measures the capacity of the system to store the data packets into the Intermediate Database. The stress test shows a high consistency with the increasing usage of the system, which is capable of writing 2.2 requests or packets per second in average, each one composed by 7800 records of sensory data.

ICL capability of inferring user activities presents important advantages with respect to other wellness systems, which frequently rely on simple step counting for activity tracking. For example, it permits to derive more precisely the user energy expenditure based on the cost of each performed activity, specially for those that do not entail any ambulation. To evaluate the potential of the implemented activity recognition model ten volunteers aging from 23 to 37 years were requested to perform the supported activities: "walk", "jog", "rest", "ride the bus" and "take the subway". The performance of the model is evaluated by comparing both actual and detected activities. The results, shown in Table 2, prove notable recognition capabilities, yielding a 94% accuracy in overall, although misclassification of some activities can be experienced.

A initial user-centric analysis is also performed in terms of adherence to the provided recommendations. Ten volunteers aging 26 and 38 years were asked to use the developed applications during a couple of weeks to measure the response time to recommendations. This time accounts for the period elapsed since the user receives a recommendation and follows it. The average number of recommendations per day were 9, ranging from 5 to 14. The subjects response time varied from 1 minute to 1 hour, with average values shown in Table 3. These results may give some clues about the interest shown in the use of these services, although further analysis, including more subjects and longer time spans, is required to obtain solid conclusions and determine their foundation.

Finally, the effectiveness and usability of the developed expert tools is also assessed. To that end, different aspects of the tools were evaluated by 6 medical experts -two nutritionists, two fitness instructors and two nurses- from an independent health and wellness counseling company from South Korea. The experts were instructed on how to use the tools and then provided with a set of questionnaires to evaluate their look and feel, interface layout complexity, time required to access a given resource or create a new rule, as well as the understandability and correctness of the concepts and contents facilitated by these tools. The results of the evaluation prove a satisfaction level greater than 75% in average. The aspects that were more highly rated correspond to the simple and easy accessibility to the diverse health and wellness related concepts as well as the organization of the information. For the behavior inspection tool the experts particularly valued the benefit of having a user-centric description of the behavioral patterns plus the possibility of identifying the acceptability of the delivered recommendations through the feedback report. For the rule authoring tool the specialists especially considered the benefits provided by the health and wellness models although they were unsatisfied with the amount of time required to write a given rule.

All these apps and tools have been designed as end-user interfaces to the contents and services curated by Mining Minds, thus presenting important advantages for the customers, such as an effective reduction of the resources consumption - mainly in terms of storage, computation and battery -, no need of regular updates of the client application, shareability of contents among diverse systems and applications, as well as a more dynamic and interactive experience. Mining Minds builds on the assumption that, in the short-term, most mobile devices and systems of the Internet of Things will be fully and seamlessly connected. However,... Although most trends predict this, meanwhile temporary local storage and offline data transmission might be required to overcome current internet disconnections.

Applications such as the one presented here operate over WiFi and 4G interfaces. While the use of WiFi presents no economic burden, some users could be concerned about using their data plans when huge amounts of data need to be transferred. For example, this application transmits around 500kB/min to communicate the sensory data to the platform, which translates into approximately 30GB/month when used nonstop. With the advent of 5G communications, flat-rate data plans are expected to be a must, and accordingly, help reduce the possible burden for the end-user. In either case, the use of *compressed sensing* techniques [49] is particularly envisioned to make the data transmission more efficient. These mechanisms and other sophisticated strategies are also worth considering to reduce battery consumption, for example, by interrupting the transmission of sensory data during periods of user inactivity.

7 Conclusions

This work has presented Mining Minds, a novel digital framework for personalized healthcare and wellness support. The framework has been neatly designed taking into account crucial requirements of the digital health and wellness paradigm. This work has also described a unique architecture defined to provide the necessary functionality to enable curation and mining of data, information, knowledge and services for personalized health and wellness support. An initial realization of the key architectural components, as well as an exemplary application that showcases some of the benefits provided by Mining Minds, have also been presented. The work is ongoing to complete the implementation of the devised architecture with new additional components as well as to evaluate its services on a large scale testbed.

Consent

Written informed consent was obtained from the participants for publication of this case report and any accompanying images. A copy of the written consent is available for review by the Editor of this journal.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

This work was supported by the Industrial Core Technology Development Program (10049079, Development of mining core technology exploiting personal big data) funded by the Ministry of Trade, Industry and Energy (MOTIE, Korea).

Author details

[1]Department of Computer Engineering, Kyung Hee University, 1732 Deokyoungdae-ro, Giheung-ug, 446-701 Yongin-si, Korea. [2]School of Computing and Information Systems, University of Tasmania, Churchill Avenue Hobart, 7005 Tasmania, Australia.

References

1. Hafner, T., Shiffman, J.: The emergence of global attention to health systems strengthening. Health Policy and Planning **28**(1), 41–50 (2013)
2. Hood, L., Flores, M.: A personal view on systems medicine and the emergence of proactive P4 medicine: predictive, preventive, personalized and participatory. New Biotechnology **29**(6), 613–624 (2012)
3. Matheson, E.M., King, D.E., Everett, C.J.: Healthy lifestyle habits and mortality in overweight and obese individuals. The Journal of the American Board of Family Medicine **25**(1), 9–15 (2012)
4. Gillen, M.M., Markey, C.N., Markey, P.M.: An examination of dieting behaviors among adults: Links with depression. Eating Behaviors **13**(2), 88–93 (2012)
5. Demark-Wahnefried, W., Jones, L.W.: Promoting a healthy lifestyle among cancer survivors. Hematology/Oncology Clinics of North America **22**(2), 319–342 (2008)
6. Swan, M.: Health 2050: the realization of personalized medicine through crowdsourcing, the quantified self, and the participatory biocitizen. Journal of Personalized Medicine **2**(3), 93–118 (2012)
7. Powell, A.C., Landman, A.B., Bates, D.W.: In search of a few good apps. The Journal of the American Medical Association **311**(18), 1851–1852 (2014)
8. Withings Activité Available online: www.withings.com/eu/en/products/activite (accessed on 29 Aug 2015)

9. Garmin Vivofit Available online: http://sites.garmin.com/en-US/vivo/vivofit/ (accessed on 29 Aug 2015)
10. Fitbit Surge Available online: https://www.fitbit.com/surge (accessed on 29 Aug 2015)
11. Misfit Shine Available online: http://www.misfitwearables.com/products/shine (accessed on 29 Aug 2015)
12. Banos, O., Villalonga, C., Damas, M., Gloesekoetter, P., Pomares, H., Rojas, I.: Physiodroid: Combining wearable health sensors and mobile devices for a ubiquitous, continuous, and personal monitoring. The Scientific World Journal **2014**(490824), 1–11 (2014)
13. Oresko, J.J., Jin, Z., Cheng, J., Huang, S., Sun, Y., Duschl, H., Cheng, A.C.: A wearable smartphone-based platform for real-time cardiovascular disease detection via electrocardiogram processing. IEEE Transactions on Information Technology in Biomedicine **14**(3), 734–740 (2010)
14. Hermens, H., op den Akker, H., Tabak, M., Wijsman, J., Vollenbroek, M.: Personalized coaching systems to support healthy behavior in people with chronic conditions. Journal of Electromyography and Kinesiology **24**(6), 815–826 (2014)
15. Chen, P.-H., Chen, H.-M.: Framework design-integrating an android open platform with multi-interface biomedical modules for physiological measurement. Journal of Convergence Information Technology **7**(12), 310–319 (2012)
16. Fortino, G., Giannantonio, R., Gravina, R., Kuryloski, P., Jafari, R.: Enabling effective programming and flexible management of efficient body sensor network applications. IEEE Transactions on Human-Machine Systems **43**(1), 115–133 (2013)
17. Gaggioli, A., Pioggia, G., Tartarisco, G., Baldus, G., Corda, D., Cipresso, P., Riva, G.: A mobile data collection platform for mental health research. Personal Ubiquitous Comput. **17**(2), 241–251 (2013)
18. Yang, G., Xie, L., Mantysalo, M., Zhou, X., Pang, Z., Xu, L.D., Kao-Walter, S., Chen, Q., Zheng, L.-R.: A health-iot platform based on the integration of intelligent packaging, unobtrusive bio-sensor, and intelligent medicine box. IEEE Transactions on Industrial Informatics **10**(4), 2180–2191 (2014)
19. Banos, O., Villalonga, C., Garcia, R., Saez, A., Damas, M., Holgado, J.A., Lee, S., Pomares, H., Rojas, I.: Design, implementation and validation of a novel open framework for agile development of mobile health applications. Biomedical Engineering Online **14**(S2:S6), 1–20 (2015)
20. Estrin, D., Sim, I.: Open mhealth architecture: An engine for health care innovation. Science **330**(6005), 759–760 (2010)
21. Google Fit Available online: https://developers.google.com/fit/overview (accessed on 29 Aug 2015)
22. SAMI, S. Available online: https://developer.samsungsami.io/sami/sami-documentation/ (accessed on 29 Aug 2015)
23. Apple HealthKit Available online: https://developer.apple.com/healthkit/ (accessed on 29 Aug 2015)
24. Banos, O., Bilal-Amin, M., Ali-Khan, W., Afzel, M., Ahmad, M., Ali, M., Ali, T., Ali, R., Bilal, M., Han, M., Hussain, J., Hussain, M., Hussain, S., Hur, T.H., Bang, J.H., Huynh-The, T., Idris, M., Kang, D.W., Park, S.B., Siddiqui, M., Vui, L.-B., Fahim, M., Khattak, A.-M., Kang, B.-H., Lee, S.: An innovative platform for person-centric health and wellness support. In: Proceedings of the International Work-Conference on Bioinformatics and Biomedical Engineering (2015)
25. Banos, O., Bang, J.H., Hur, T.H., Siddiqui, M., Huynh-The, T., Vui, L.-B., Ali-Khan, W., Ali, T., Villalonga, C., Lee, S.: Mining human behavior for health promotion. In: Proceedings of the 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2015) (2015)
26. Microsoft Azure Available online: http://azure.microsoft.com (accessed on 29 August 2015)
27. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., Warfield, A.: Xen and the art of virtualization. ACM SIGOPS Operating Systems Review **37**(5), 164–177 (2003)
28. Padhy, R.P.: Big data processing with hadoop-mapreduce in cloud systems. International Journal of Cloud Computing and Services Science **2**(1), 16–27 (2012)
29. Microsoft Windows Server Available online: http://www.microsoft.com/en-us/server-cloud/products/windows-server-2012-r2/ (accessed on 29 August 2015)
30. Richardson, L., Ruby, S.: RESTful Web Services. O'Reilly Media, Inc., ??? (2008)
31. Kim, K., Kim, C., Jung, S.-I., Shin, H.-S., Kim, J.-S.: Inter-domain socket communications supporting high performance and full binary compatibility on xen. In: Proceedings of the Fourth ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, pp. 11–20 (2008)
32. Microsoft SQL Server Available online: http://www.microsoft.com/en-us/server-cloud/products/sql-server/ (accessed on 29 August 2015)
33. Banos, O., Damas, M., Pomares, H., Rojas, F., Delgado-Marquez, B., Valenzuela, O.: Human activity recognition based on a sensor weighting hierarchical classifier. Soft Computing **17**, 333–343 (2013)
34. Lara, O., Labrador, M.: A survey on human activity recognition using wearable sensors. IEEE Communications Surveys Tutorials **PP**(99), 1–18 (2012)
35. Banos, O., Galvez, J.-M., Damas, M., Pomares, H., Rojas, I.: Window size impact in human activity recognition. Sensors **14**(4), 6474–6499 (2014)
36. Preece, S.J., Goulermas, J.Y., Kenney, L.P.J., Howard, D.: A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data. IEEE Transactions on Biomedical Engineering **56**(3), 871–879 (2009)
37. Han, M., Vinh, L.T., Lee, Y.-K., Lee, S.: Comprehensive context recognizer based on multimodal sensors in a smartphone. Sensors **12**(9), 12588–12605 (2012)
38. Banos, O., Damas, M., Pomares, H., Prieto, A., Rojas, I.: Daily living activity recognition based on statistical feature quality group selection. Expert Systems with Applications **39**(9), 8013–8021 (2012)
39. Ali, T., Hussain, M., Ali Khan, W., Afzal, M., Lee, S.: Authoring tool: acquiring sharable knowledge for Smart CDSS. In: International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 1278–1281 (2013)
40. Ainsworth, B.E., Haskell, W.L., Herrmann, S.D., Meckes, N., Bassett, D.R., Tudor-Locke, C., Greer, J.L., Vezina, J., Whitt-Glover, M.C., Leon, A.S.: 2011 compendium of physical activities: a second update of codes

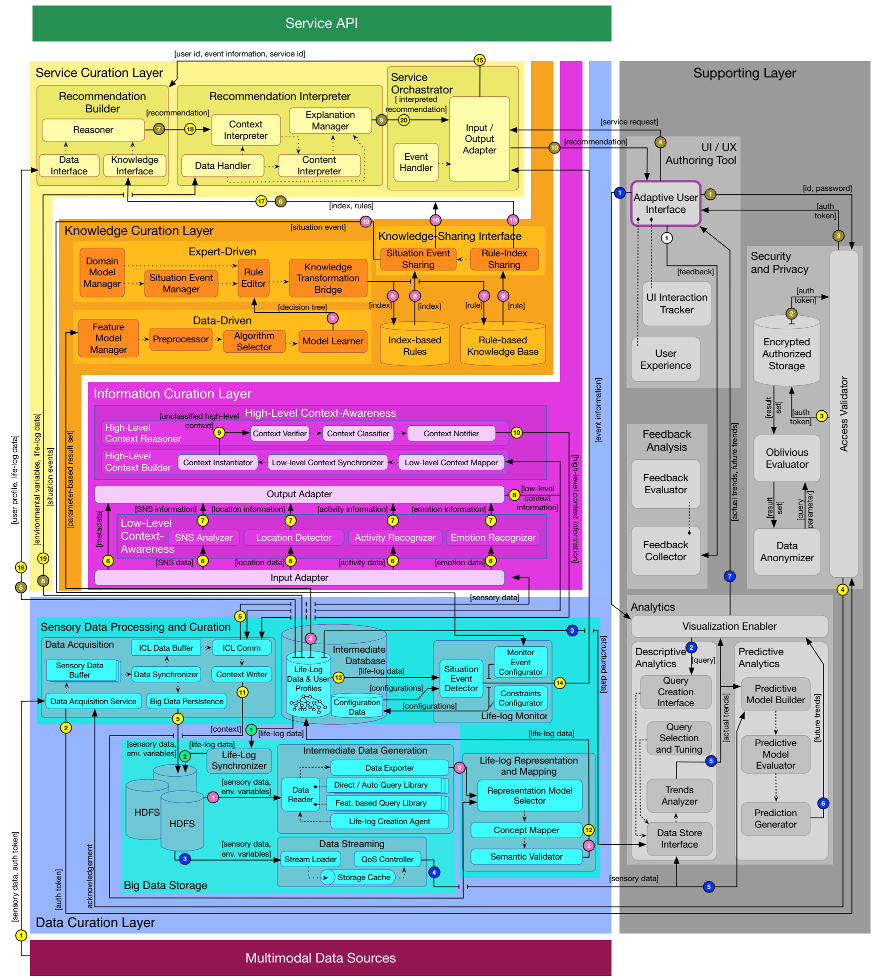


Figure 1: Mining Minds Framework Architecture and Operational diagram.

and met values. Medicine and Science in Sports and Exercise **43**(8), 1575–1581 (2011)
41. Forgy, C.L.: Rete: A fast algorithm for the many pattern/many object pattern match problem. Artificial intelligence **19**(1), 17–37 (1982)
42. Schofield, W.: Predicting basal metabolic rate, new standards and review of previous work. Human nutrition and Clinical nutrition **39**, 5–41 (1984)
43. Baget, J.-F.: Improving the forward chaining algorithm for conceptual graphs rules. In: International Conference on Principles of Knowledge Representation and Reasoning, pp. 407–414 (2004)
44. Gong, S.: Learning user interest model for content-based filtering in personalized recommendation system. International Journal of Digital Content Technology and its Applications **6**(11), 155–162 (2012)
45. Daemen, J., Rijmen, V.: The Design of Rijndael: AES-the Advanced Encryption Standard. Springer, ??? (2002)
46. Zhong, S., Chen, T.: An efficient identity-based protocol for private matching. International Journal of Communication Systems **24**(4), 543–552 (2011)
47. Ghinita, G., Karras, P., Kalnis, P., Mamoulis, N.: A framework for efficient data anonymization under privacy and accuracy constraints. ACM Transactions on Database Systems **34**(2), 9 (2009)
48. Pervez, Z., Awan, A.A., Khattak, A.M., Lee, S., Huh, E.-N.: Privacy-aware searching with oblivious term matching for cloud storage. The Journal of Supercomputing **63**(2), 538–560 (2013)
49. Li, S., Xu, L.D., Wang, X.: A continuous biomedical signal acquisition system based on compressed sensing in body sensor networks. IEEE Transactions on Industrial Informatics **9**(3), 1764–1771 (2013)

Figures

Tables

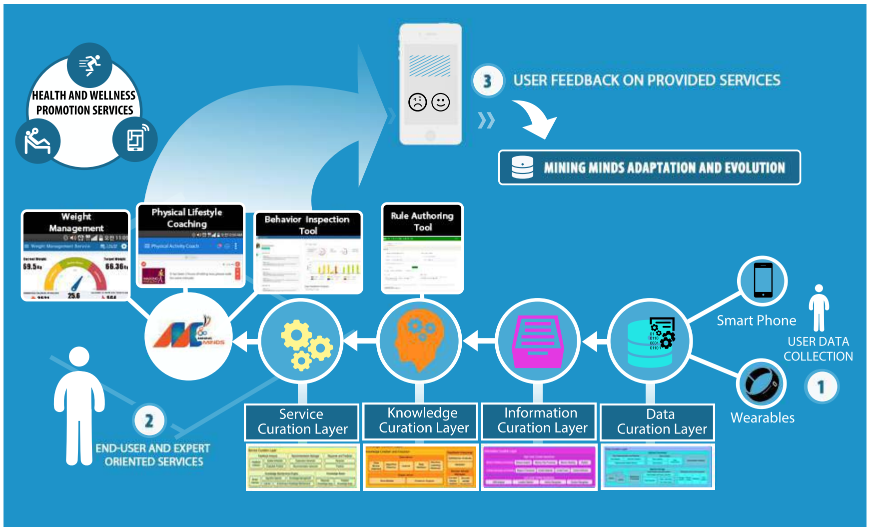


Figure 2: Mining Minds health and wellness service scenario.

Table 1: Accuracy of the data curation process

No. of service calls	No. of Missed data packets	Error (%)
30,000	6	0.02
60,000	22	0.04
90,000	39	0.04
120,000	55	0.05
150,000	96	0.06
180,000	308	0.17
Average		0.06

Table 2: Activity recognition performance. Each metric correspond to sensitivity (SE), specificity (SP), positive predictive value (PPV), negative predictive value (NPV) and F-score.

Activity	**SE**	**SP**	**PPV**	**NPV**	**F-score**
Eating	0.89	1.00	0.88	1.00	0.88
Running	0.97	1.00	0.99	1.00	0.98
Sitting	0.95	0.98	0.94	0.98	0.95
Standing	0.91	0.99	0.95	0.98	0.93
Walking	0.99	0.99	0.98	1.00	0.99
Jogging	0.98	1.00	0.98	1.00	0.98
Stretching	0.97	0.99	0.92	1.00	0.94
Sweeping	0.94	1.00	0.94	1.00	0.94
Lying down	0.90	1.00	0.93	1.00	0.92

Table 3: Average user response time (in minutes) to recommendations

User	1	2	3	4	5	6	7	8	9	10
Avg response time	24.47	34.44	3.42	5.38	40.44	7.21	28.29	13.99	8.56	36.84

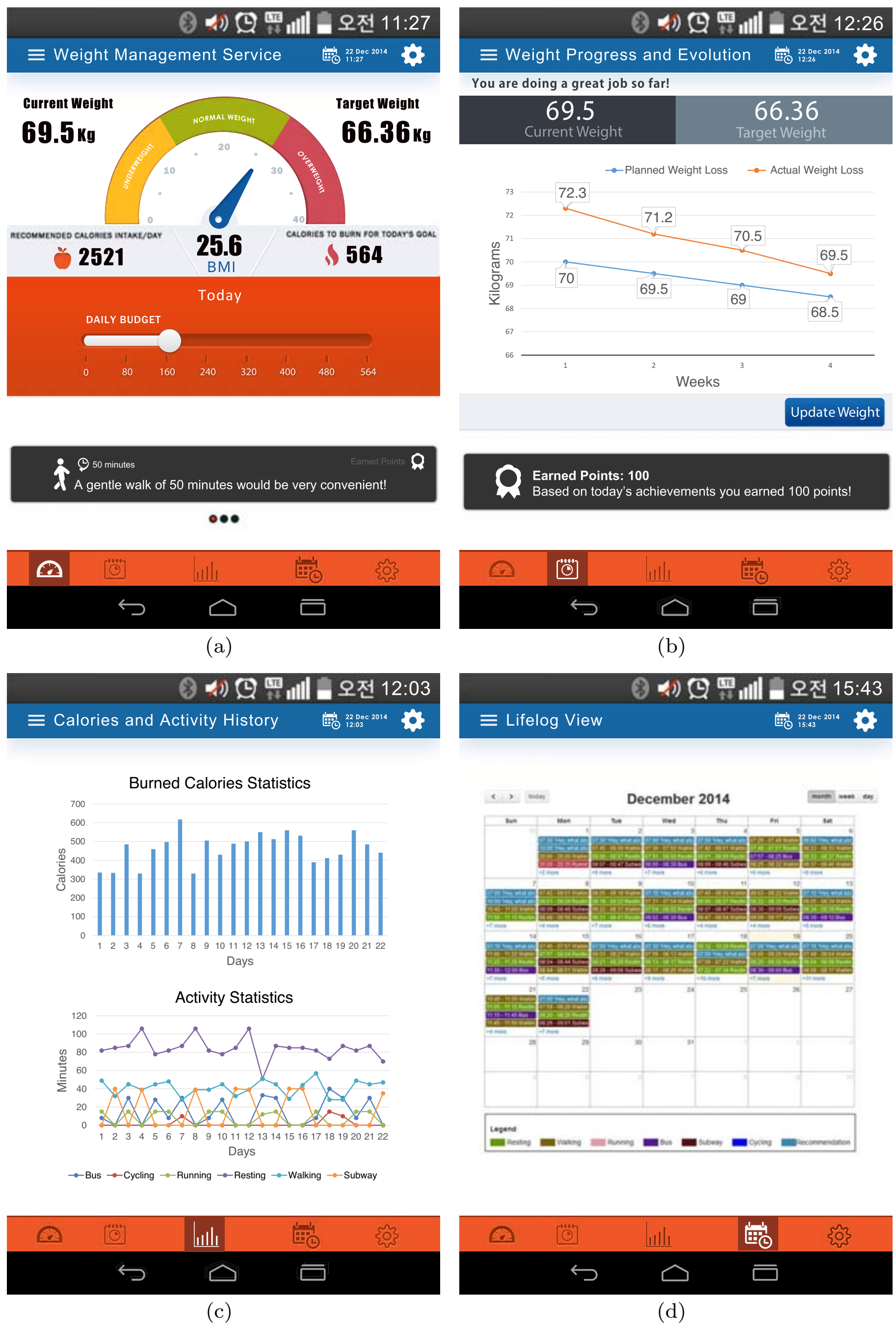


Figure 3: Personalized weight management app.

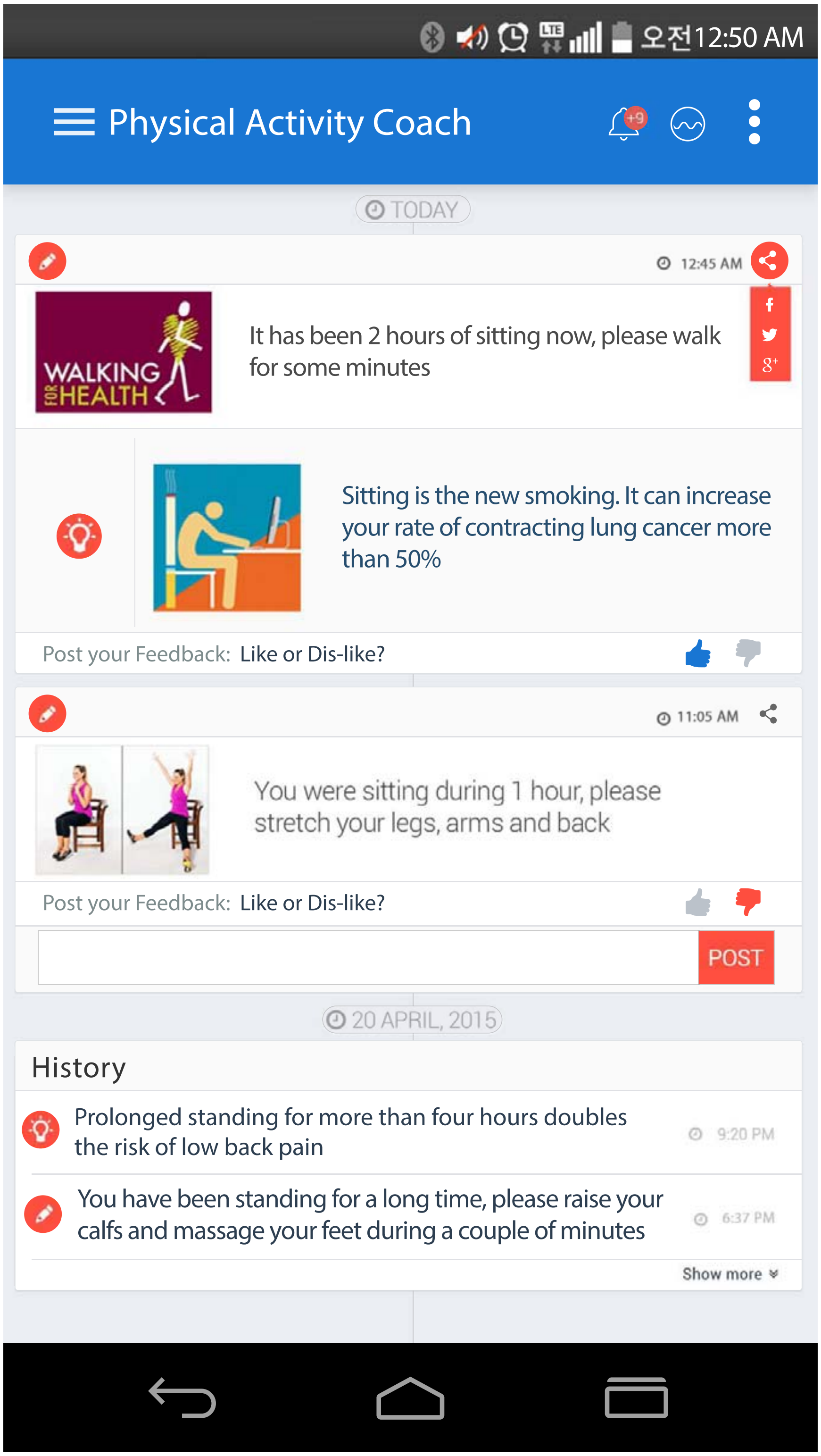


Figure 4: Physical lifestyle coaching app.

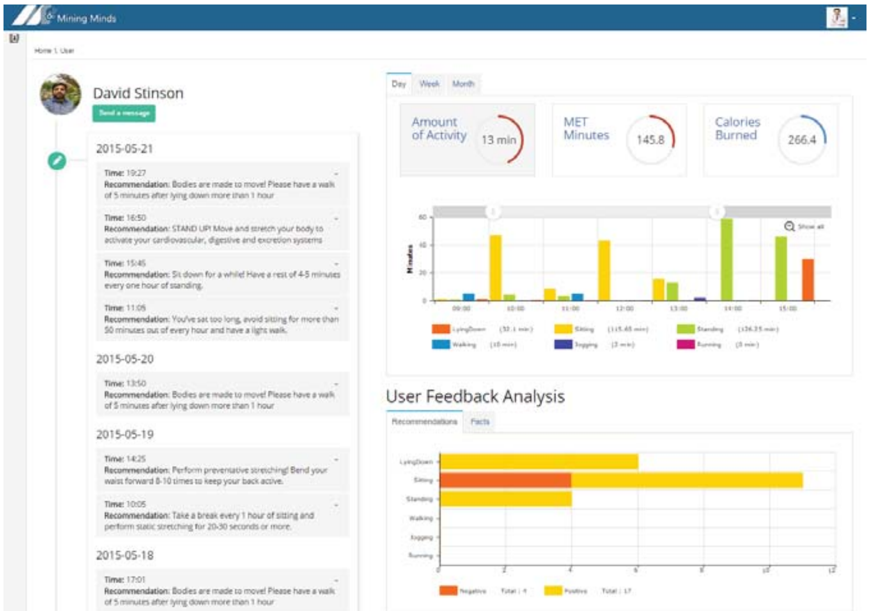


Figure 5: Behavior inspection tool.

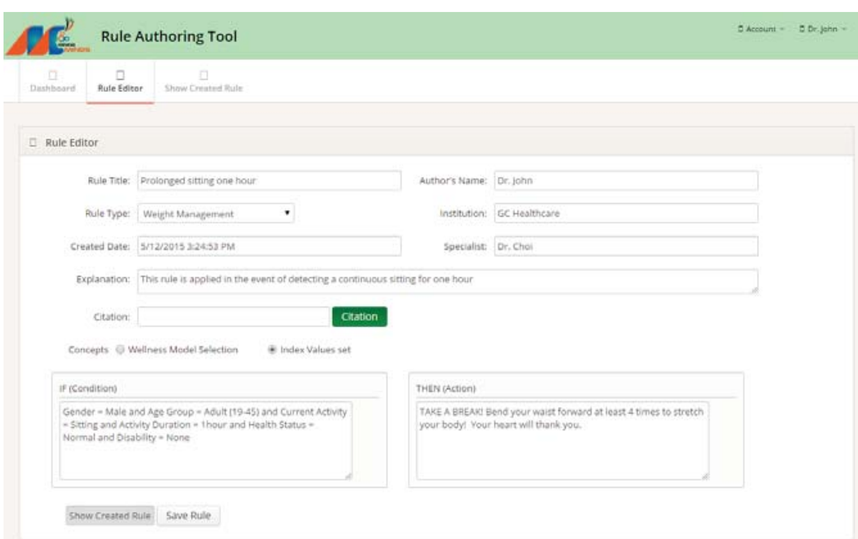


Figure 6: Rule authoring tool.

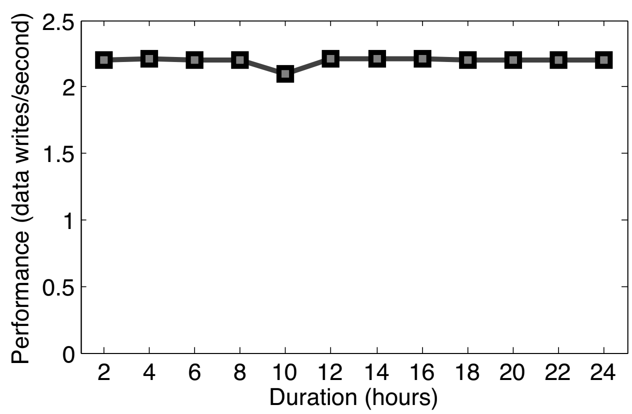


Figure 7: Performance of the data persistence process.

Section 2

Mining Minds Version 2.5 Overview and Demonstration

퍼스널 빅데이터를 활용한

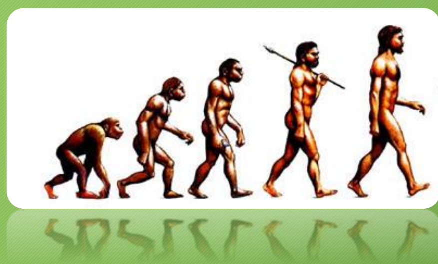
마이닝 마인즈 핵심기술 개발

경희대학교

이 승 룡

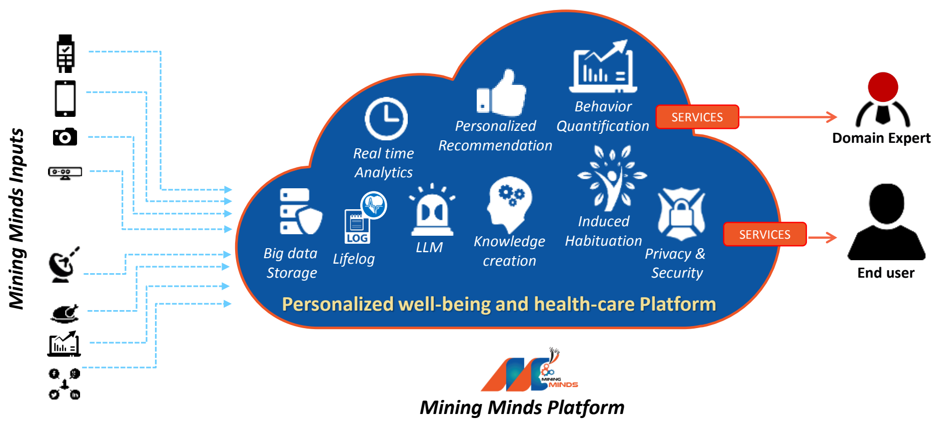
Agenda

- Evolution of Mining Minds
- Mining Minds V2.5 Service Scenario
- Mining Minds V2.5 Architecture
- Platform Operations, Uniqueness and Contribution
- Mining Minds V2.5 Microdemos
- Future Plan
- Conclusions



Evolution of Mining Minds

Mining Minds Platform Overview



History of Mining Minds

MM V1.0
Dec 2014

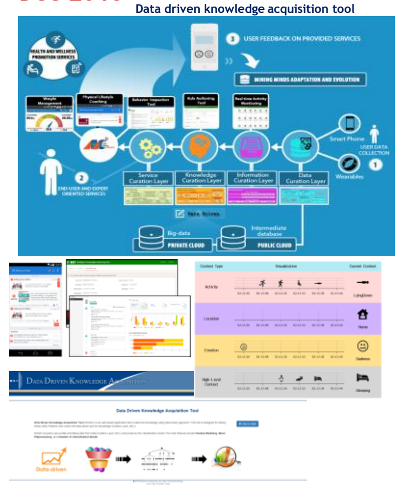
Weight Management App
Management Tool

MM V1.5
May 2015

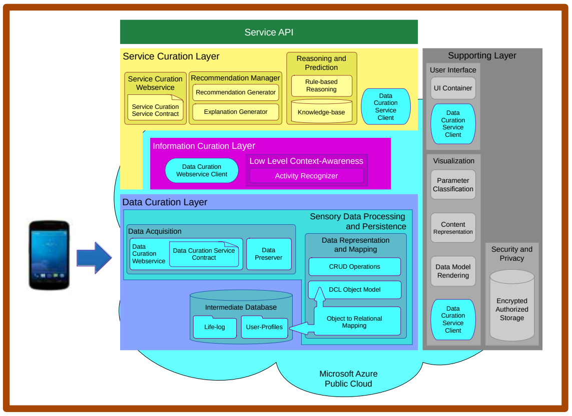
Personalized Lifestyle Coaching App
Rule Authoring Tool
Behavior Inspection Tool

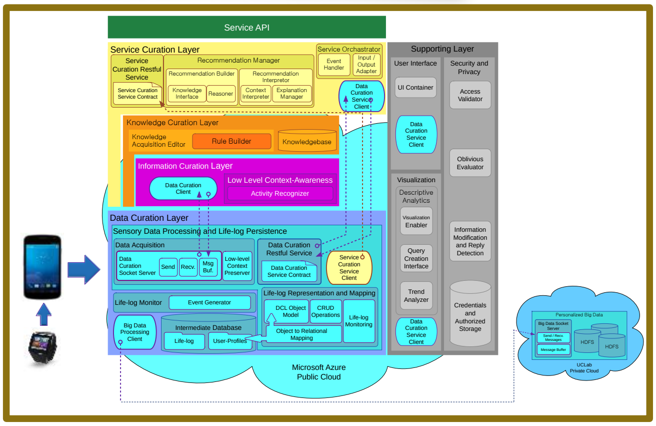
MM V2.0
Dec 2015

Personalized Lifestyle Coaching App
Rule Authoring Tool
Behavior Inspection Tool
Real time activities simulation tool
Data driven knowledge acquisition tool



History of Mining Minds





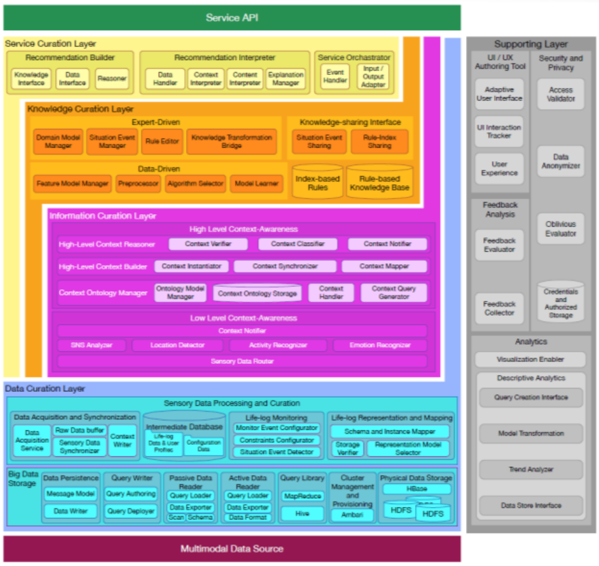
MM V1.0
Dec 2014

Primitive Data Curation (Representation and Mapping)
Smartphone-based Activity Recognition (5 activities)
Prebuilt Recommendations Generation (static)
Static Admin Tool (raw data and activities)

MM V1.5
May 2015

Big Data (only persistence) & Lifelog (representation and mapping)
Smartphone + Smartwatch-based Activity Recognition (8 activities)
Rule-authoring tool (no executable knowledge)
Situation-triggered Personalized Recommendations (local knowledge)
Expert Inspection Tool (single/multi-user stats)

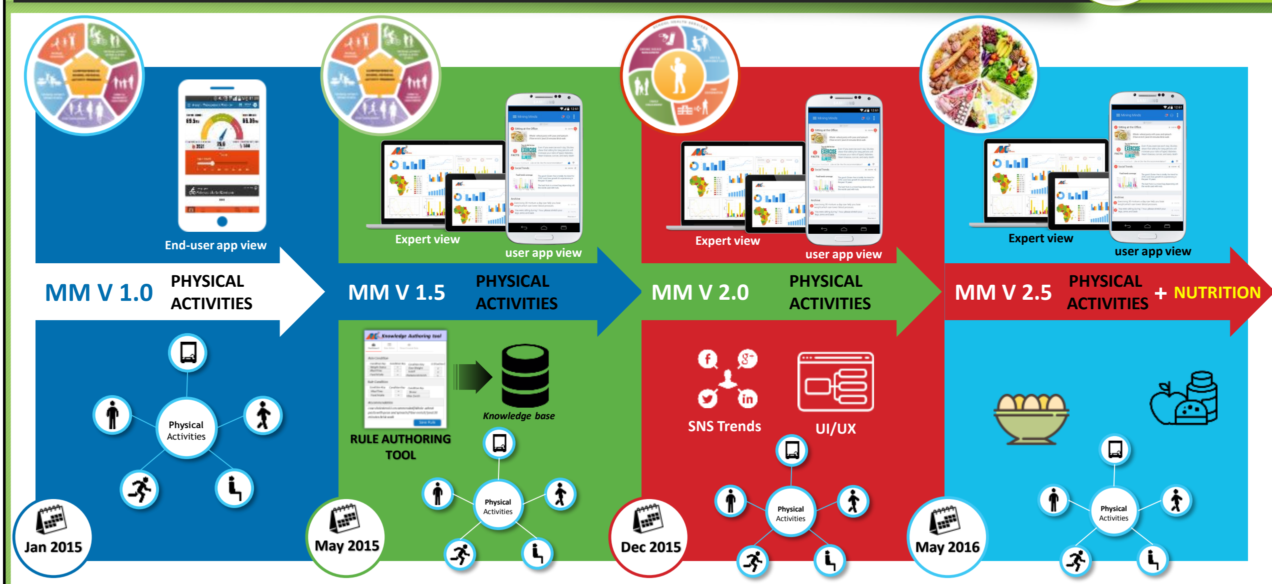
History of Mining Minds



MM V2.0
Dec 2015

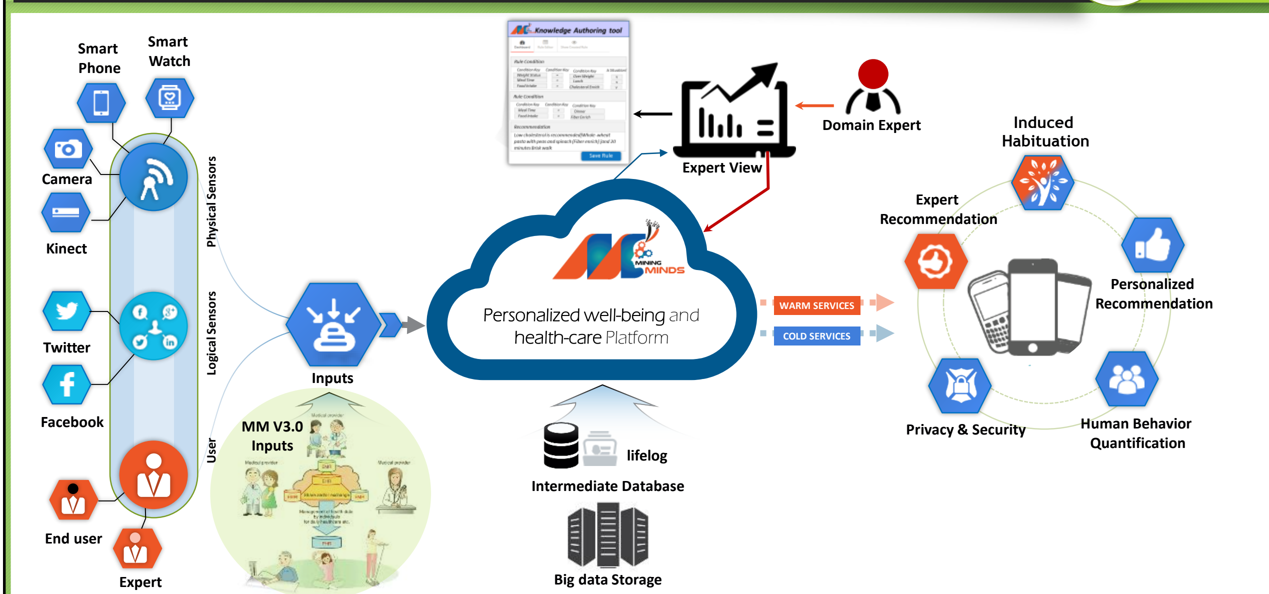
Big Data (only persistence) & Lifelog (representation and mapping)
Smartphone + Smartwatch-based Activity Recognition (8 activities)
Rule-authoring tool (no executable knowledge)
Situation-triggered Personalized Recommendations (local knowledge)
Expert Inspection Tool (single/multi-user stats)
Real time activities monitoring (low level and corresponding high level)
Data driven knowledge acquisition (utilizing big data approach)

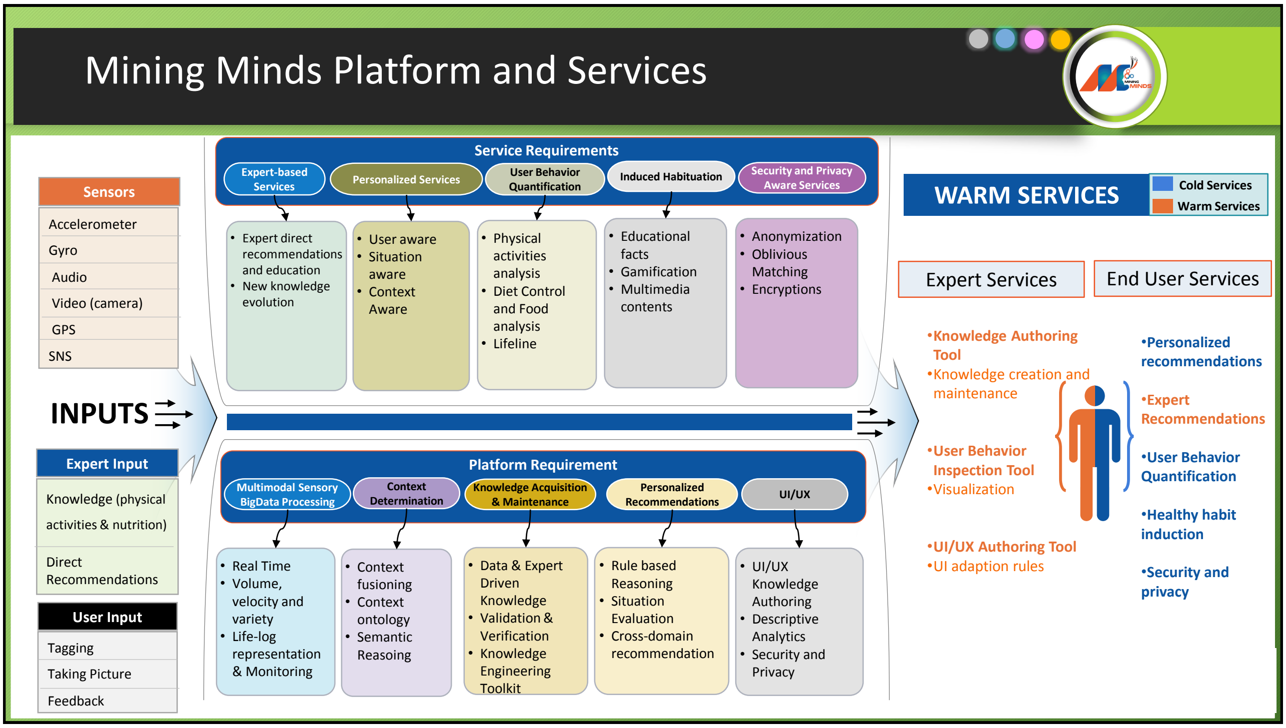
Mining Minds Evolution

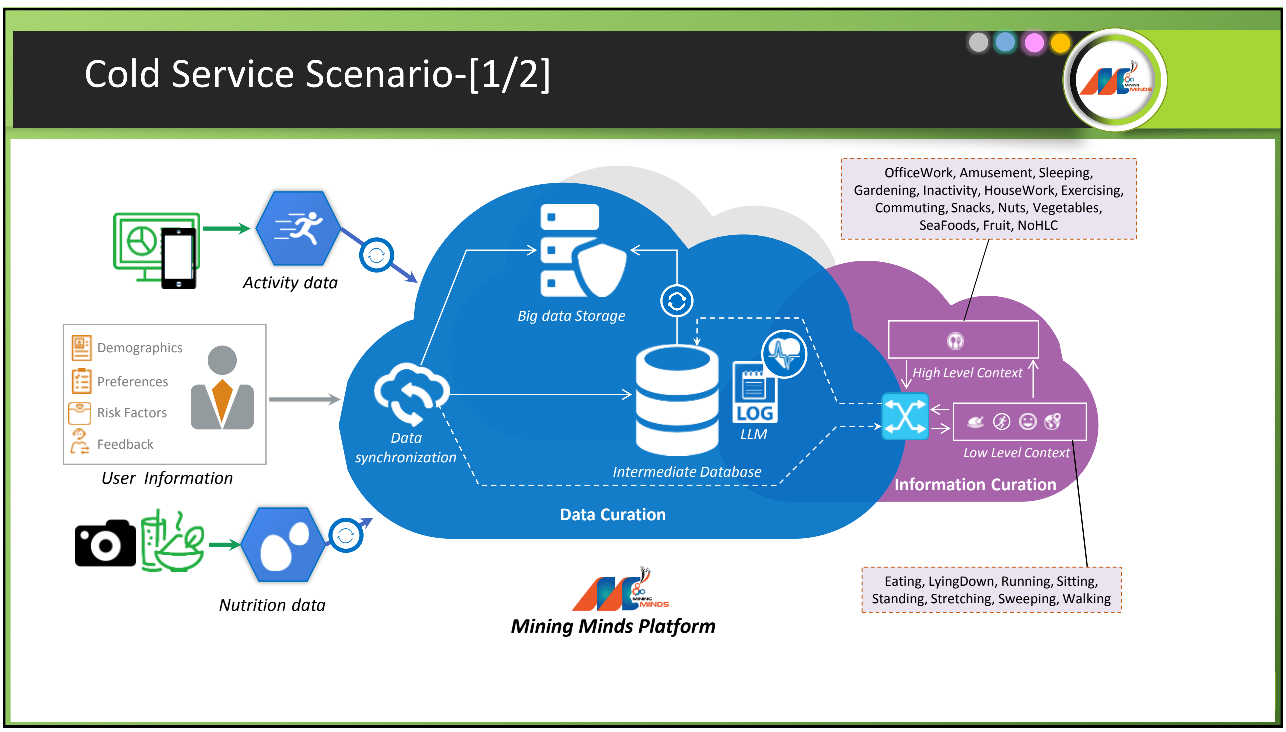


Mining Minds V2.5 Service Scenario

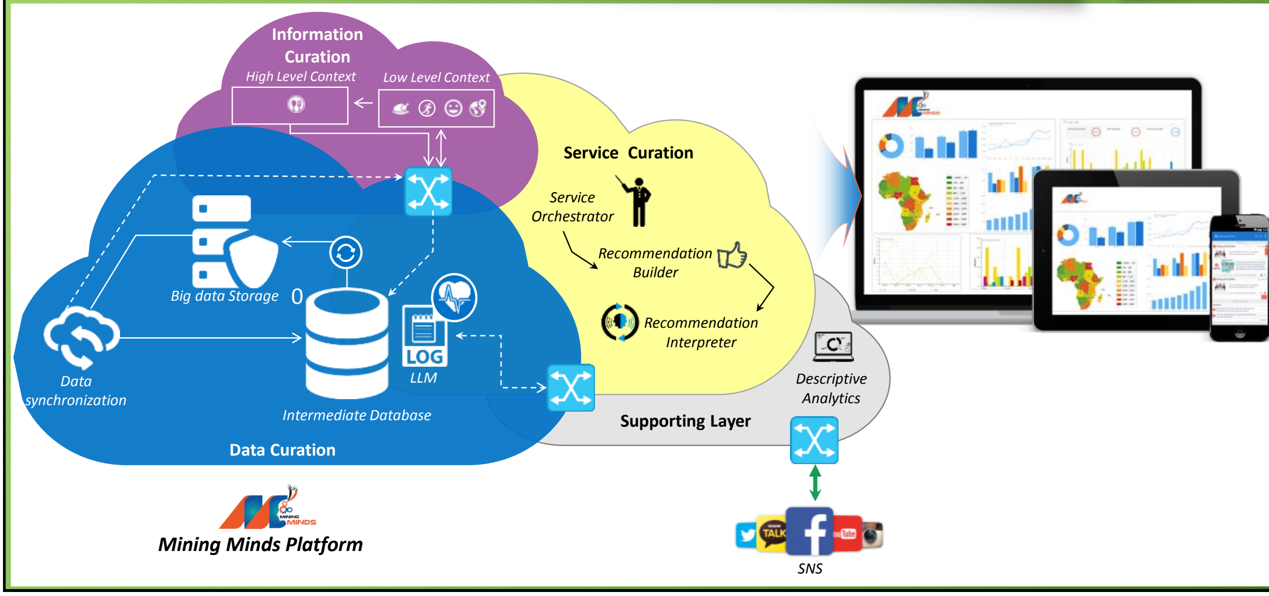
Overall Service Scenario for MM2.5



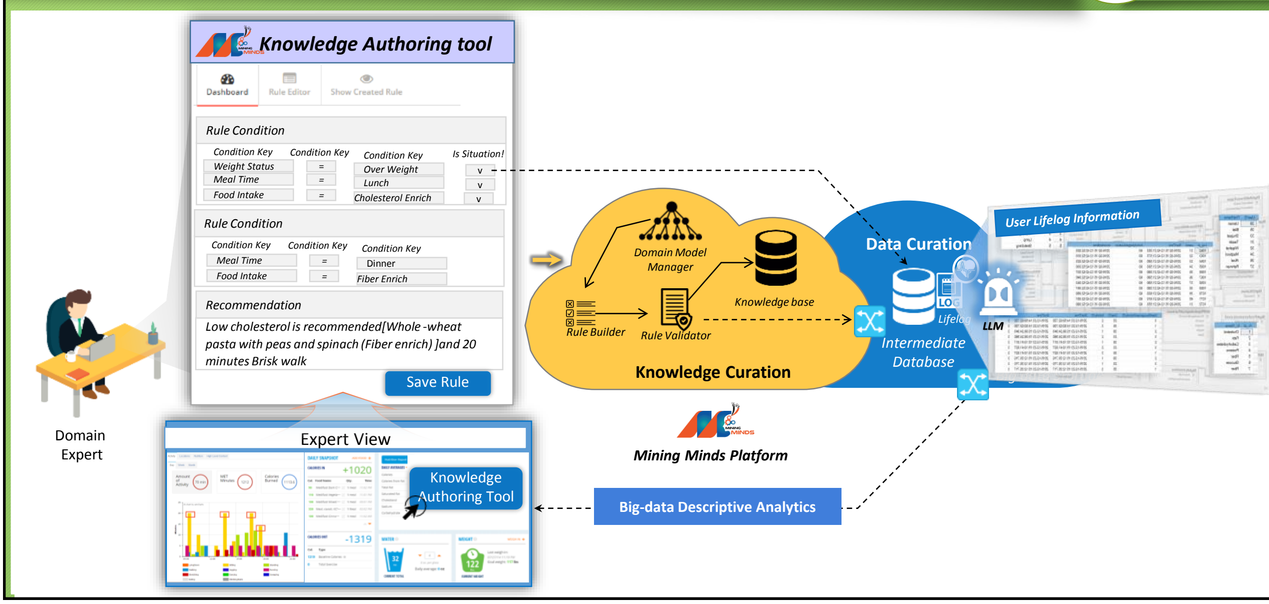

Mining Minds Platform and Services
Sensors
Accelerometer
Gyro
Audio
Video (camera)
GPS
SNS
INPUTS
Expert Input
Knowledge (physical activities & nutrition)
Direct Recommendations
User Input
Tagging
Taking Picture
Feedback
Service Requirements
Expert-based Services
Personalized Services
User Behavior Quantification
Induced Habituation
Security and Privacy Aware Services
Expert direct recommendations and education
New knowledge evolution
User aware
Situation aware
Context Aware
Physical activities analysis
Diet Control and Food analysis
Lifeline
Educational facts
Gamification
Multimedia contents
Anonymization
Oblivious Matching
Encryptions
Platform Requirement
Multimodal Sensory BigData Processing
Context Determination
Knowledge Acquisition & Maintenance
Personalized Recommendations
UI/UX
Real Time
Volume, velocity and variety
Life-log representation & Monitoring
Context fusioning
Context ontology
Semantic Reasoning
Data & Expert Driven Knowledge
Validation & Verification
Knowledge Engineering Toolkit
Rule based Reasoning
Situation Evaluation
Cross-domain recommendation
UI/UX Knowledge Authoring
Descriptive Analytics
Security and Privacy
WARM SERVICES
Cold Services
Warm Services
Expert Services
End User Services
•Knowledge Authoring Tool
•Knowledge creation and maintenance
•User Behavior Inspection Tool
•Visualization
•UI/UX Authoring Tool
•UI adaption rules
•Personalized recommendations
•Expert Recommendations
•User Behavior Quantification
•Healthy habit induction
•Security and privacy


Cold Service Scenario-[1/2]
Activity data
Demographics
Preferences
Risk Factors
Feedback
User Information
Nutrition data
Big data Storage
Data synchronization
Intermediate Database
LOG
LLM
Data Curation
OfficeWork, Amusement, Sleeping, Gardening, Inactivity, HouseWork, Exercising, Commuting, Snacks, Nuts, Vegetables, SeaFoods, Fruit, NoHLC
High Level Context
Low Level Context
Information Curation
Eating, LyingDown, Running, Sitting, Standing, Stretching, Sweeping, Walking
Mining Minds Platform

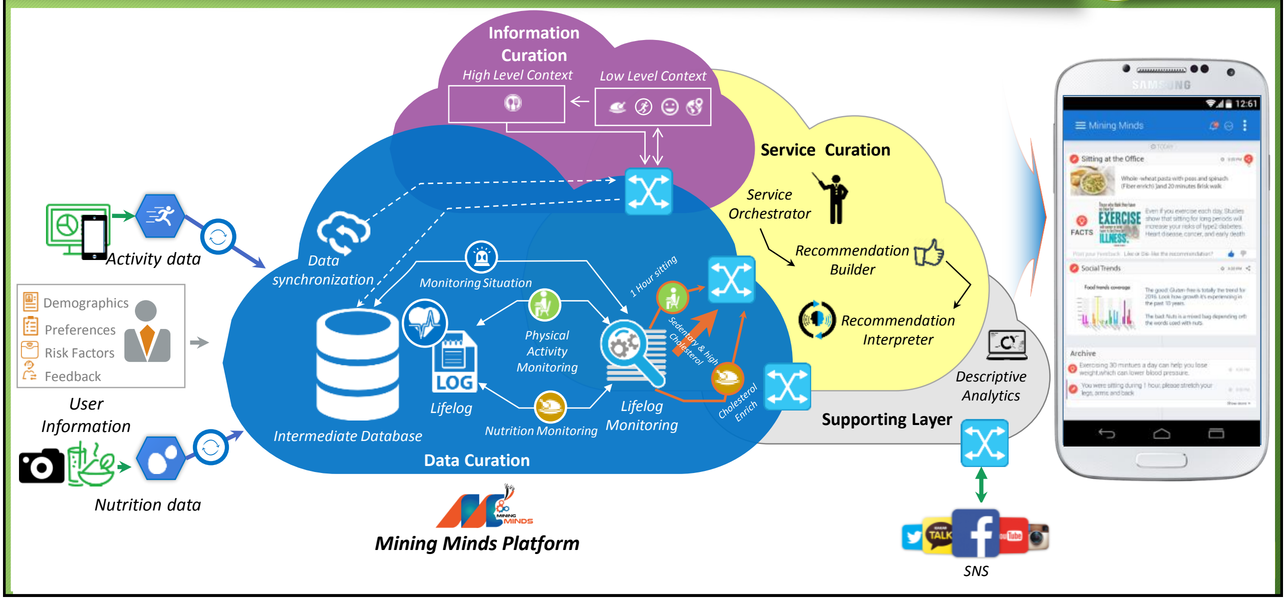
Cold Service Scenario-[2/2]



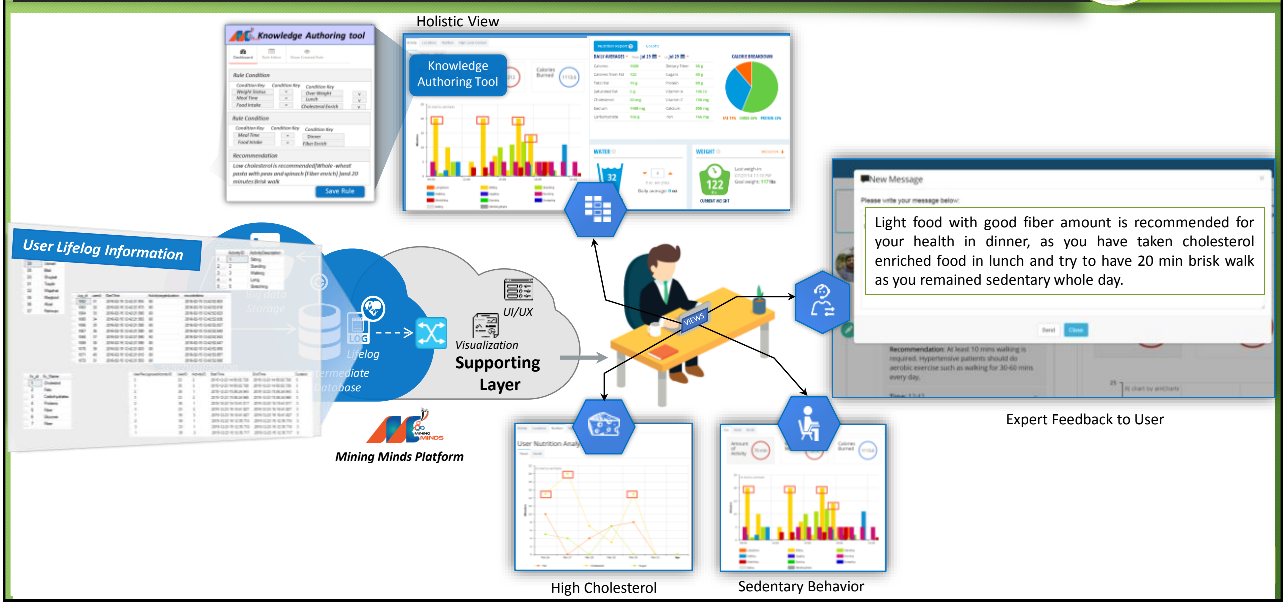
Warm Service Scenario-[1/4]



Warm Service Scenario-[2/4]

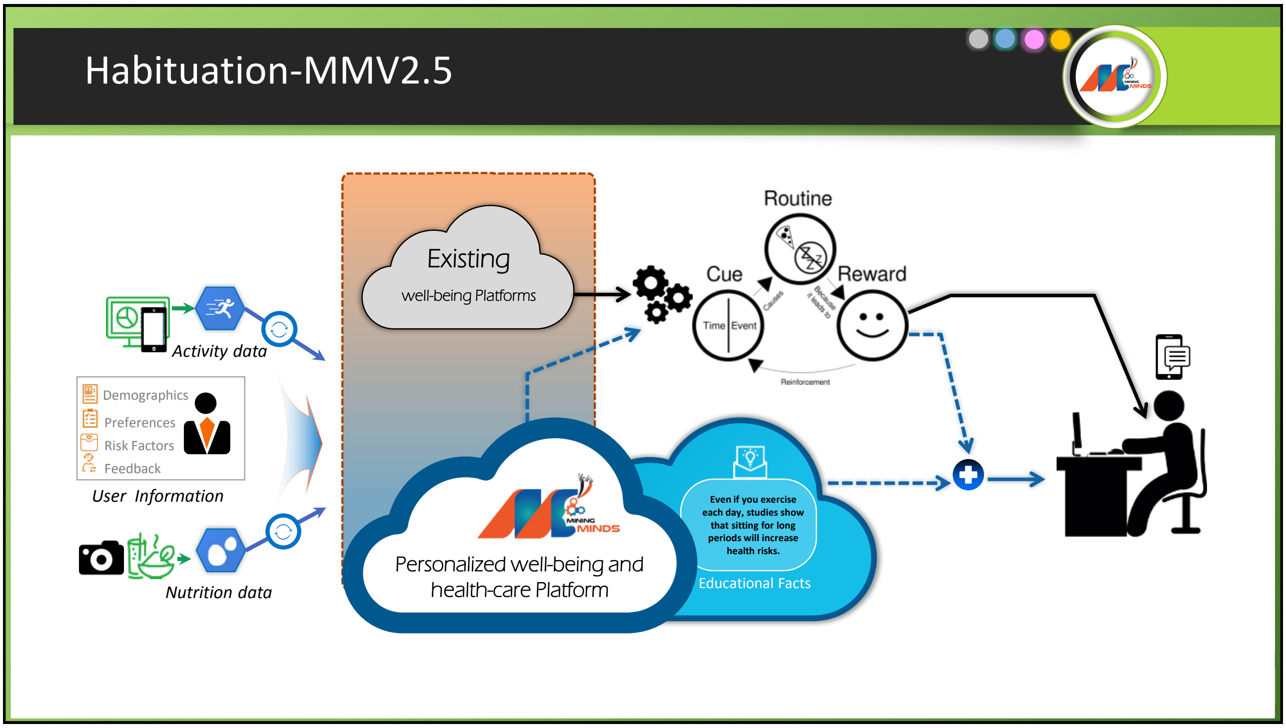


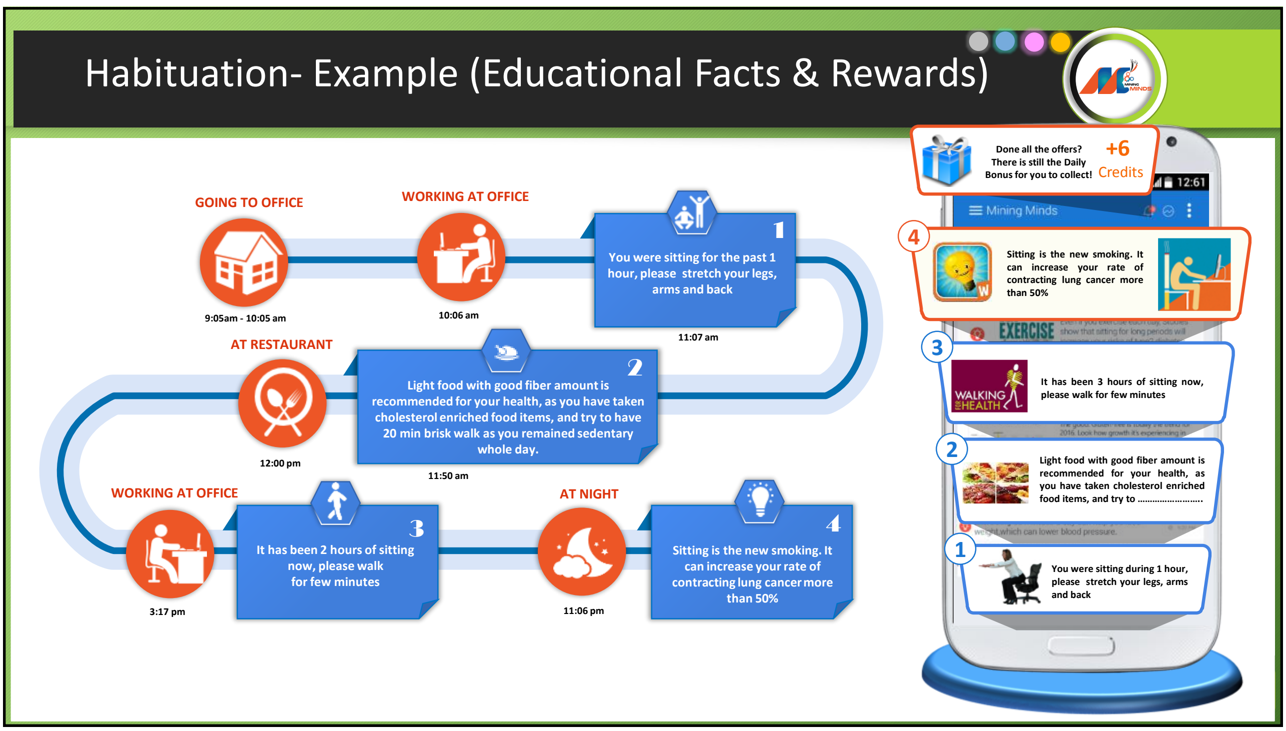
Warm Service Scenario-[3/4]



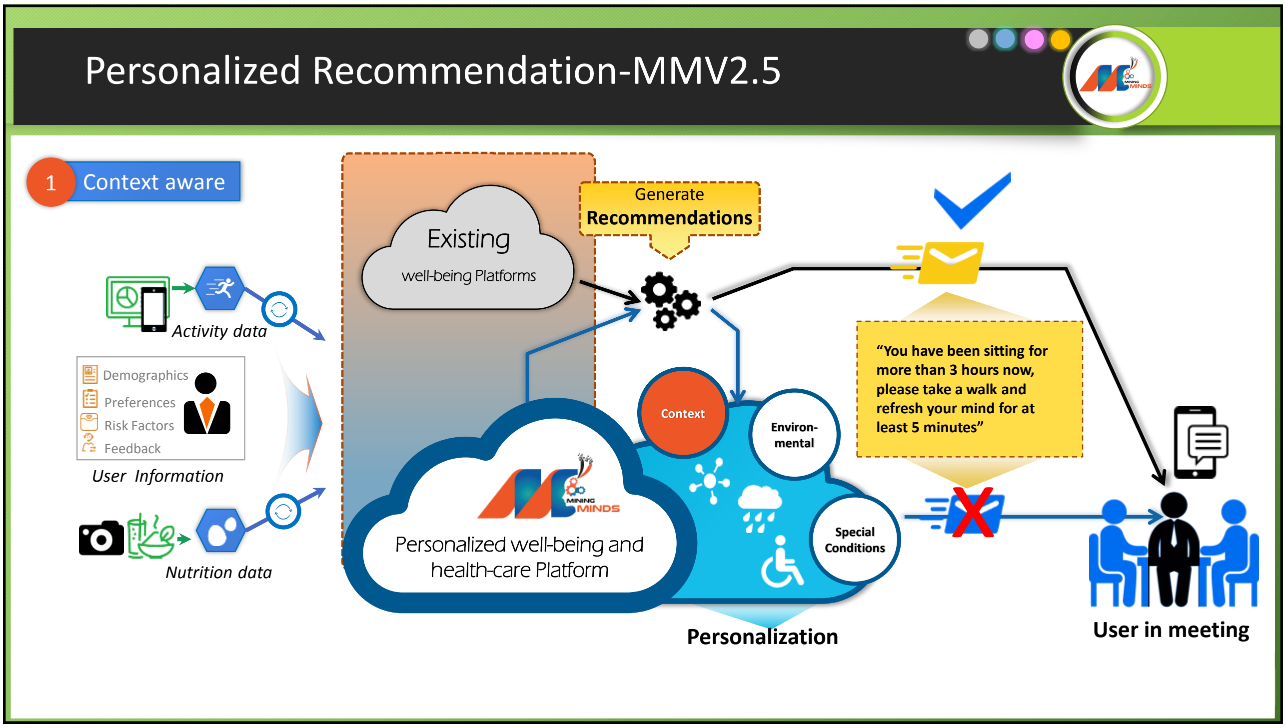
Warm Service Scenario-[4/4]

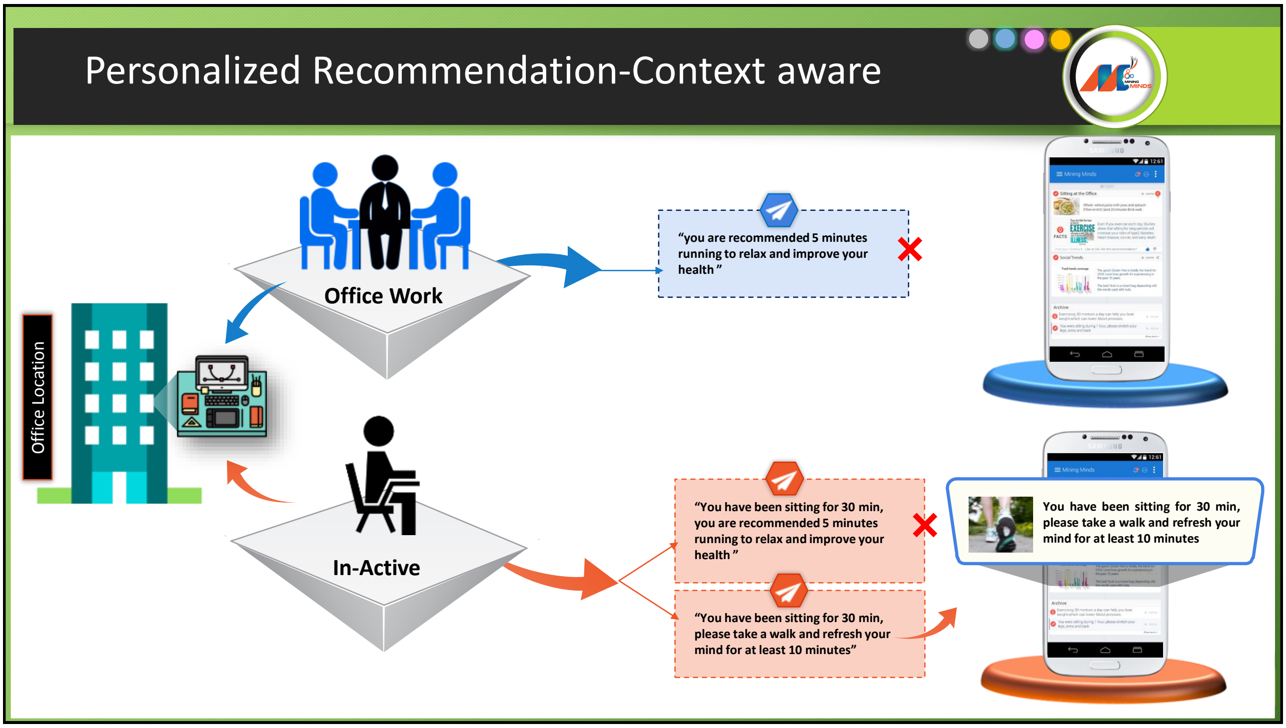
Induced Habituation

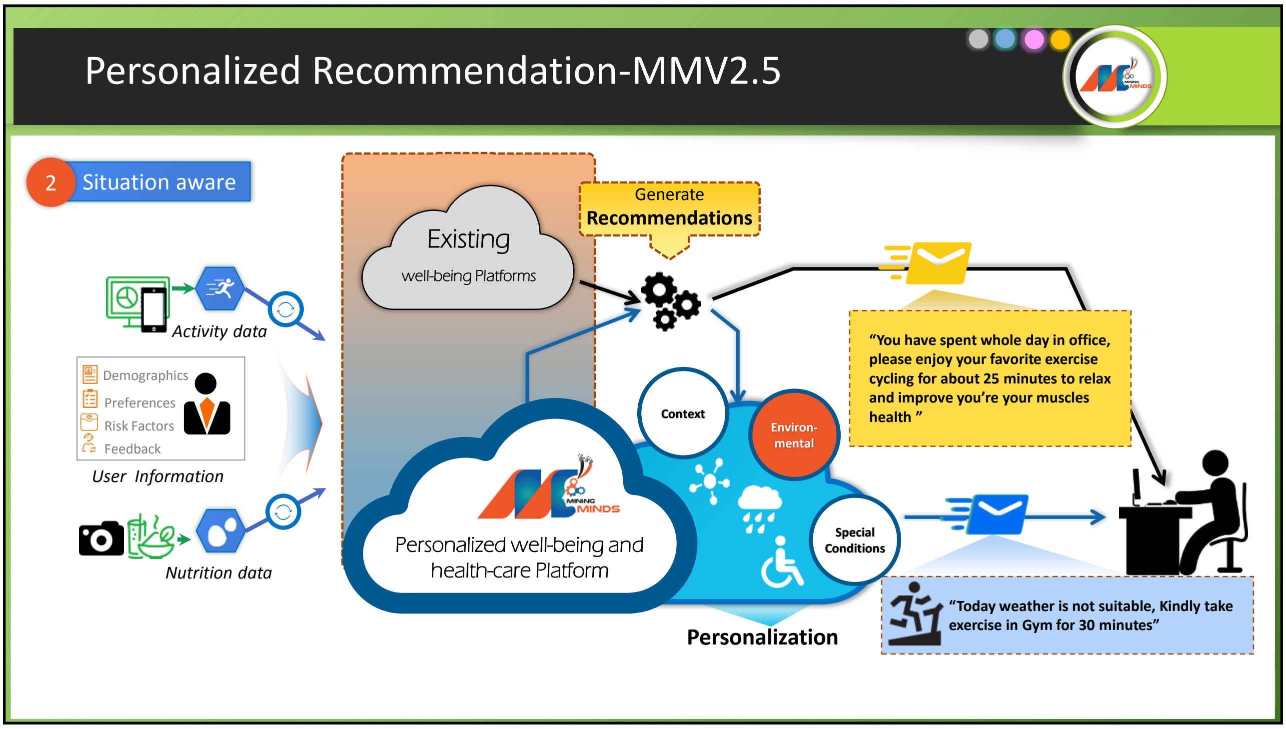

Habituation-MMV2.5
Existing
well-being Platforms
Routine
Cue
Reward
Time
Event
Reinforcement
Activity data
Demographics
Preferences
Risk Factors
Feedback
User Information
Nutrition data
Personalized well-being and
health-care Platform
Even if you exercise each day, studies show that sitting for long periods will increase health risks.
Educational Facts

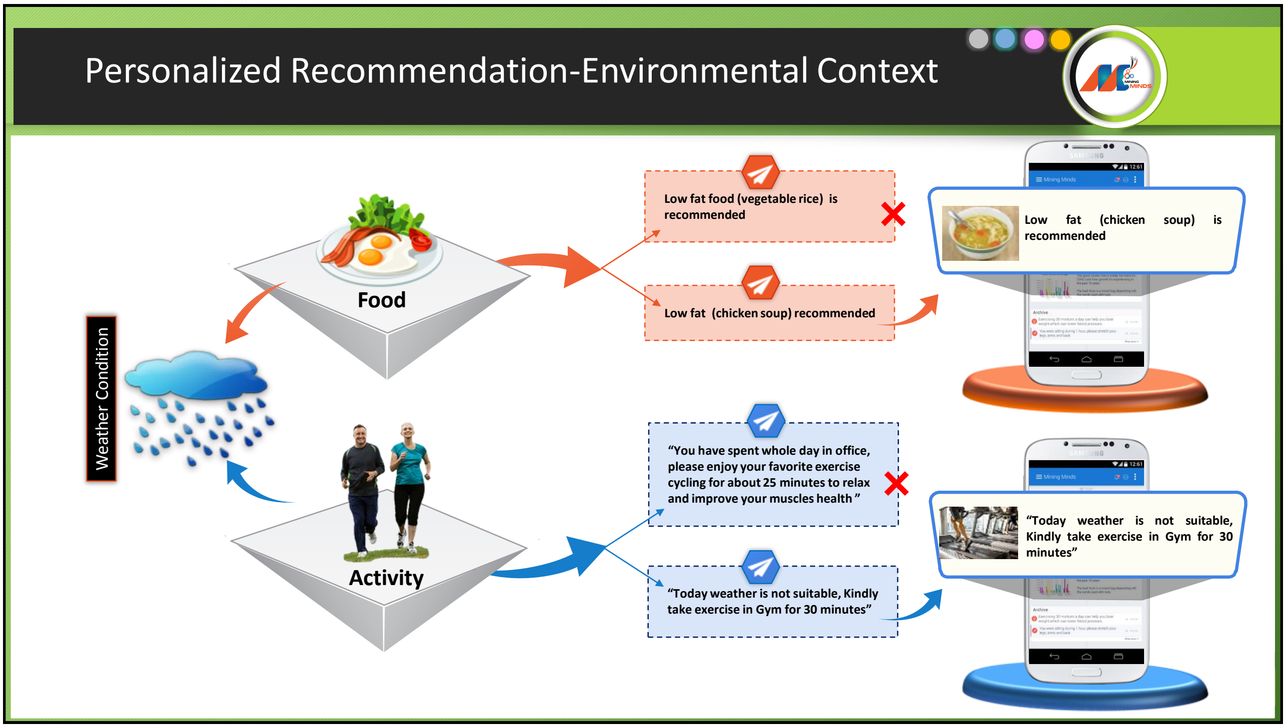

Habituation- Example (Educational Facts & Rewards)
GOING TO OFFICE
9:05am - 10:05 am
WORKING AT OFFICE
10:06 am
1
You were sitting for the past 1 hour, please stretch your legs, arms and back
11:07 am
AT RESTAURANT
12:00 pm
2
Light food with good fiber amount is recommended for your health, as you have taken cholesterol enriched food items, and try to have 20 min brisk walk as you remained sedentary whole day.
11:50 am
WORKING AT OFFICE
3:17 pm
3
It has been 2 hours of sitting now, please walk for few minutes
AT NIGHT
11:06 pm
4
Sitting is the new smoking. It can increase your rate of contracting lung cancer more than 50%
Done all the offers? There is still the Daily Bonus for you to collect!
+6
Credits
12:61
Mining Minds
4
Sitting is the new smoking. It can increase your rate of contracting lung cancer more than 50%
EXERCISE
3
WALKING HEALTH
It has been 3 hours of sitting now, please walk for few minutes
2
Light food with good fiber amount is recommended for your health, as you have taken cholesterol enriched food items, and try to ....
1
You were sitting during 1 hour, please stretch your legs, arms and back

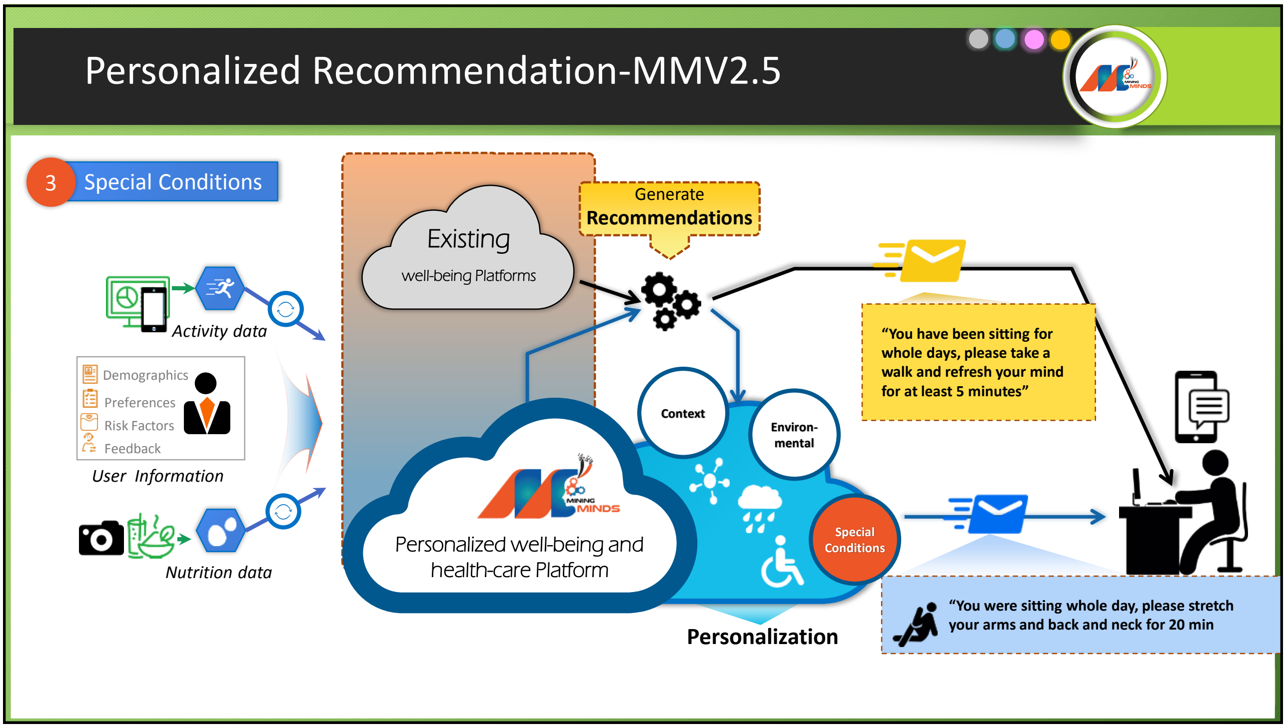

Personalized Recommendations
Induced Habituation
Expert Recommendation
Personalized Recommendation
Privacy & Security
Human Behavior Quantification

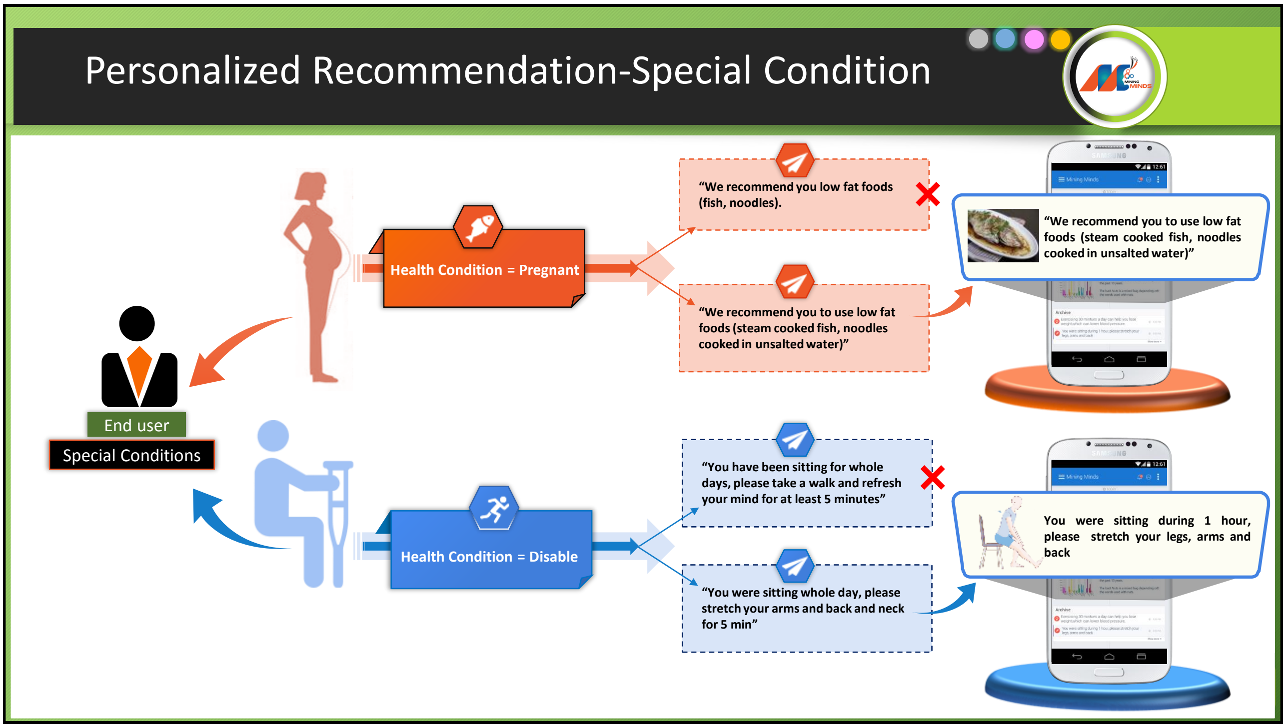

Personalized Recommendation-MMV2.5
1 Context aware
Activity data
Demographics
Preferences
Risk Factors
Feedback
User Information
Nutrition data
Existing
well-being Platforms
Generate
Recommendations
Personalized well-being and health-care Platform
Context
Environ-mental
Special Conditions
"You have been sitting for more than 3 hours now, please take a walk and refresh your mind for at least 5 minutes"
Personalization
User in meeting


Personalized Recommendation-Context aware
Office Location
Office Work
In-Active
"you are recommended 5 minutes running to relax and improve your health"
"You have been sitting for 30 min, you are recommended 5 minutes running to relax and improve your health"
"You have been sitting for 30 min, please take a walk and refresh your mind for at least 10 minutes"
You have been sitting for 30 min, please take a walk and refresh your mind for at least 10 minutes

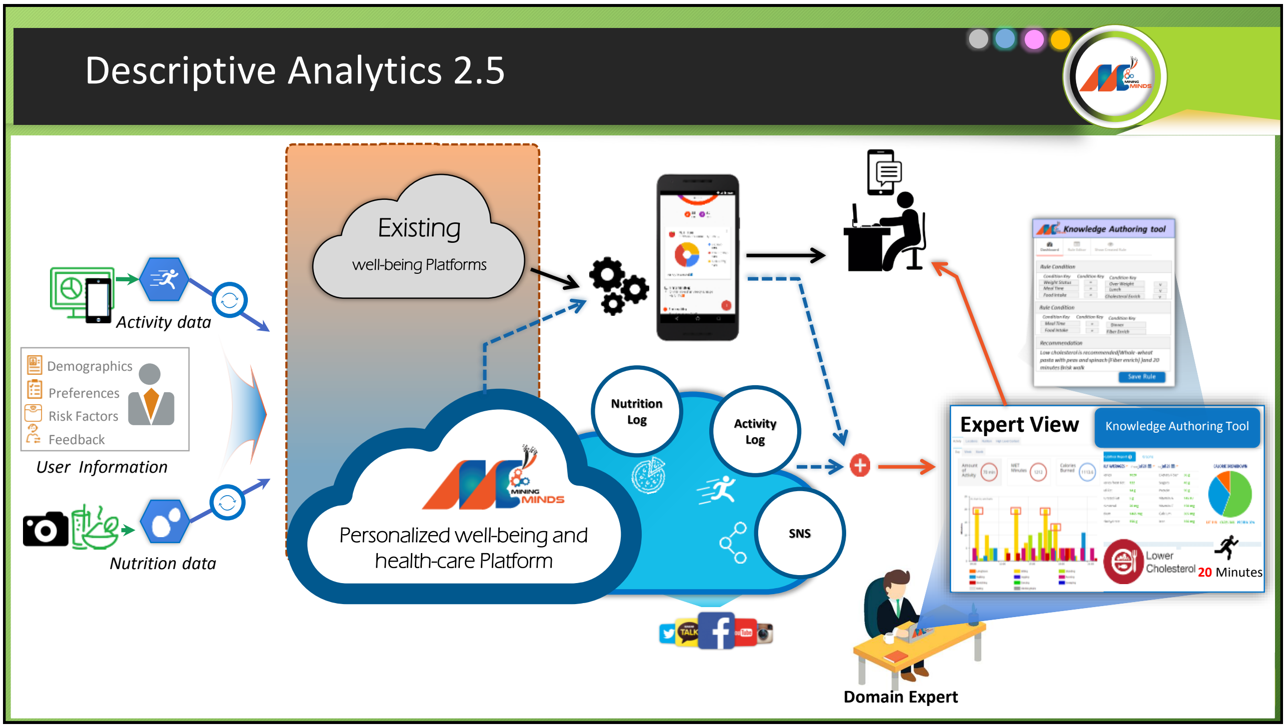

Personalized Recommendation-MMV2.5
2 Situation aware
Activity data
Demographics
Preferences
Risk Factors
Feedback
User Information
Nutrition data
Existing
well-being Platforms
Generate
Recommendations
Personalized well-being and health-care Platform
Context
Environ-mental
Special Conditions
Personalization
"You have spent whole day in office, please enjoy your favorite exercise cycling for about 25 minutes to relax and improve you're your muscles health"
"Today weather is not suitable, Kindly take exercise in Gym for 30 minutes"


Personalized Recommendation-Environmental Context
Food
Weather Condition
Activity
Low fat food (vegetable rice) is recommended
Low fat (chicken soup) recommended
Low fat (chicken soup) is recommended
"You have spent whole day in office, please enjoy your favorite exercise cycling for about 25 minutes to relax and improve your muscles health"
"Today weather is not suitable, Kindly take exercise in Gym for 30 minutes"
"Today weather is not suitable, Kindly take exercise in Gym for 30 minutes"

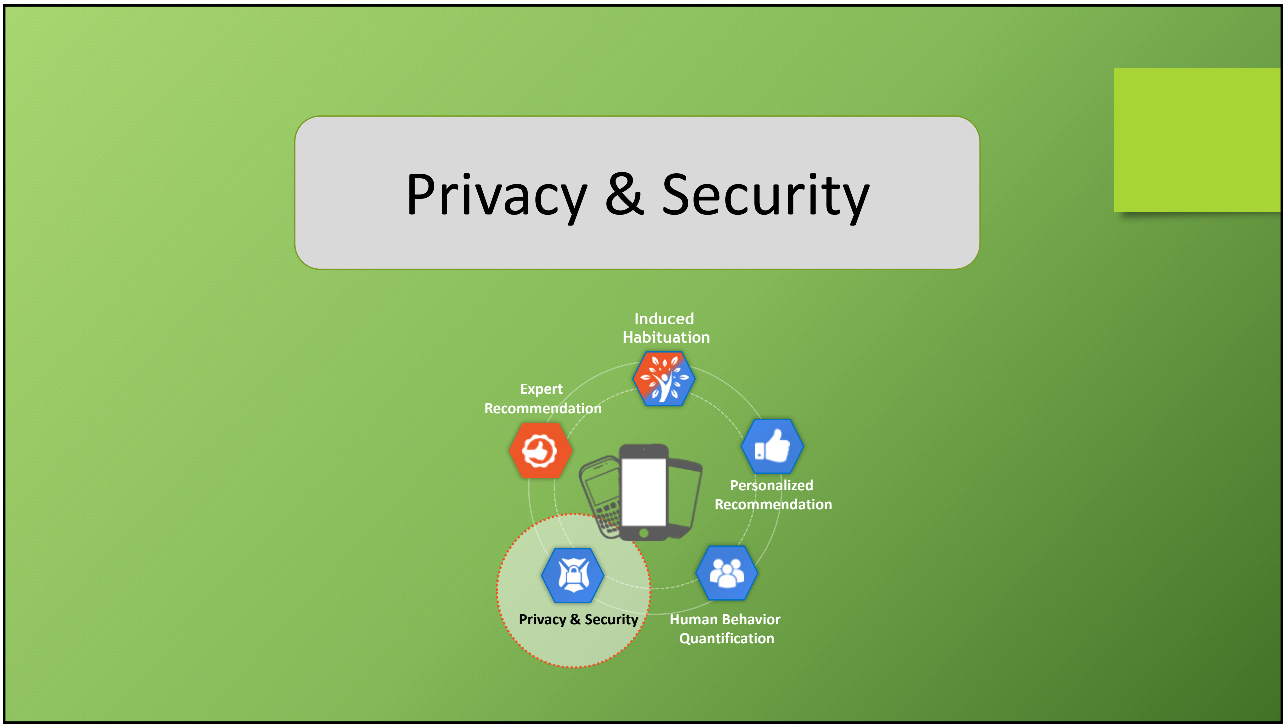

Personalized Recommendation-MMV2.5
3 Special Conditions
Activity data
Demographics
Preferences
Risk Factors
Feedback
User Information
Nutrition data
Existing
well-being Platforms
Generate
Recommendations
Personalized well-being and health-care Platform
Context
Environ-mental
Special Conditions
Personalization
"You have been sitting for whole days, please take a walk and refresh your mind for at least 5 minutes"
"You were sitting whole day, please stretch your arms and back and neck for 20 min


Personalized Recommendation-Special Condition
End user
Special Conditions
Health Condition = Pregnant
"We recommend you low fat foods (fish, noodles).
"We recommend you to use low fat foods (steam cooked fish, noodles cooked in unsalted water)"
"We recommend you to use low fat foods (steam cooked fish, noodles cooked in unsalted water)"
Health Condition = Disable
"You have been sitting for whole days, please take a walk and refresh your mind for at least 5 minutes"
"You were sitting whole day, please stretch your arms and back and neck for 5 min"
You were sitting during 1 hour, please stretch your legs, arms and back

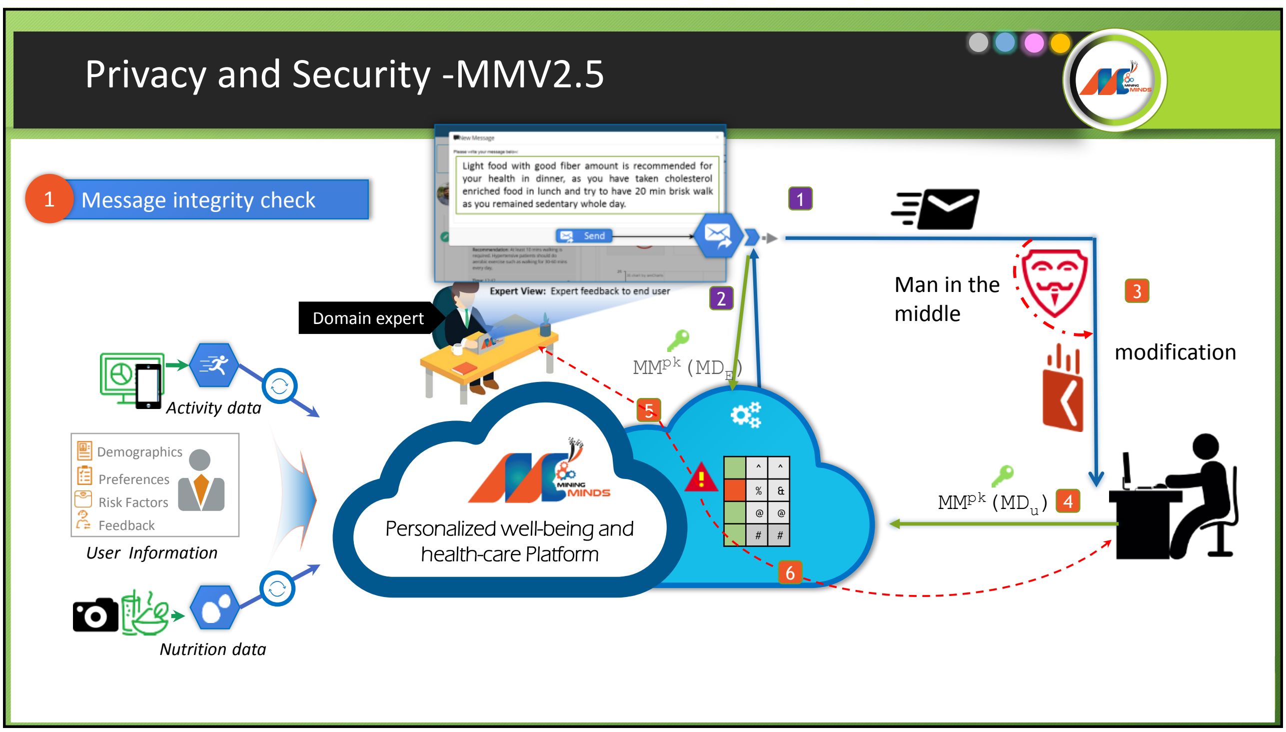

Human Behavior Quantification
Induced Habituation
Expert Recommendation
Personalized Recommendation
Privacy & Security
Human Behavior Quantification


Descriptive Analytics 2.5
Existing
well-being Platforms
Activity data
Demographics
Preferences
Risk Factors
Feedback
User Information
Nutrition data
Nutrition Log
Activity Log
SNS
Personalized well-being and health-care Platform
Expert View
Knowledge Authoring Tool
Knowledge Authoring tool
Lower Cholesterol
20 Minutes
Domain Expert

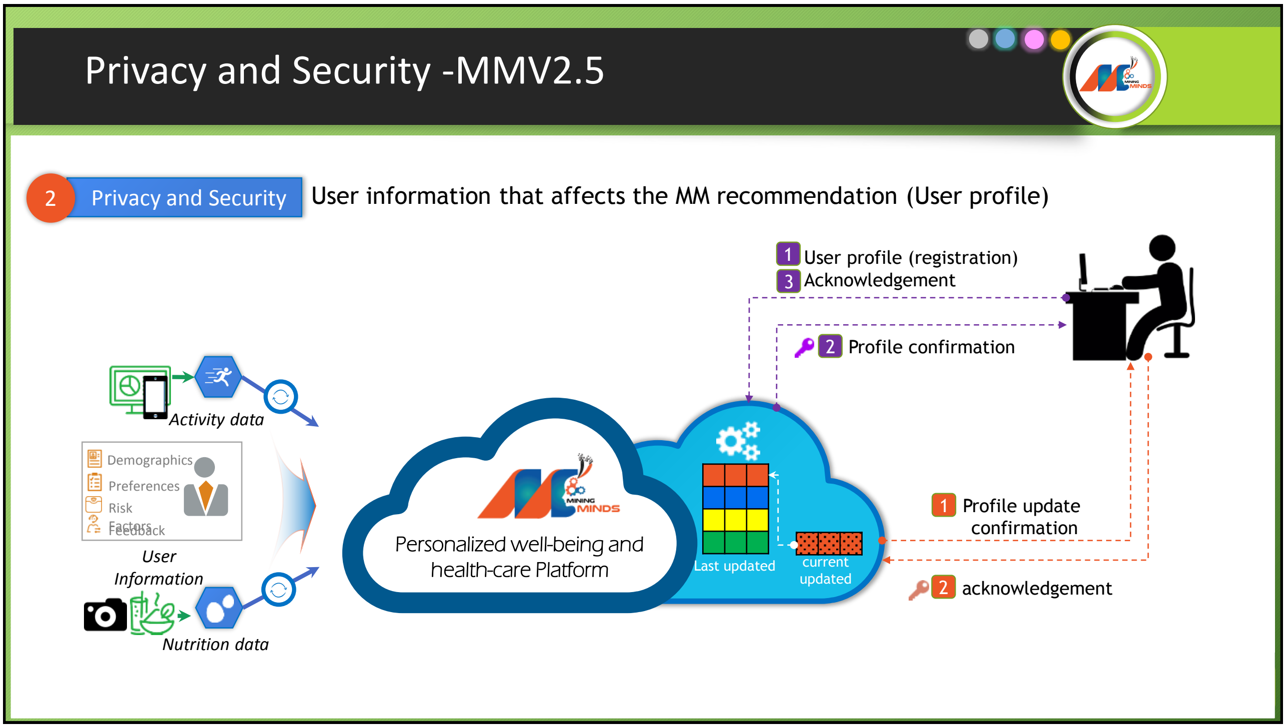

Descriptive Analytics – Activities, Nutrition, & SNS
SNS
Nutrition Log
Activity Log
Dashboard
Dashboard / Admin
Dashboards
Dashboard
Pages
Activities
Proteins/Fats /Carbs Consumed overall based on food
SNS: Meat /Fish Trends

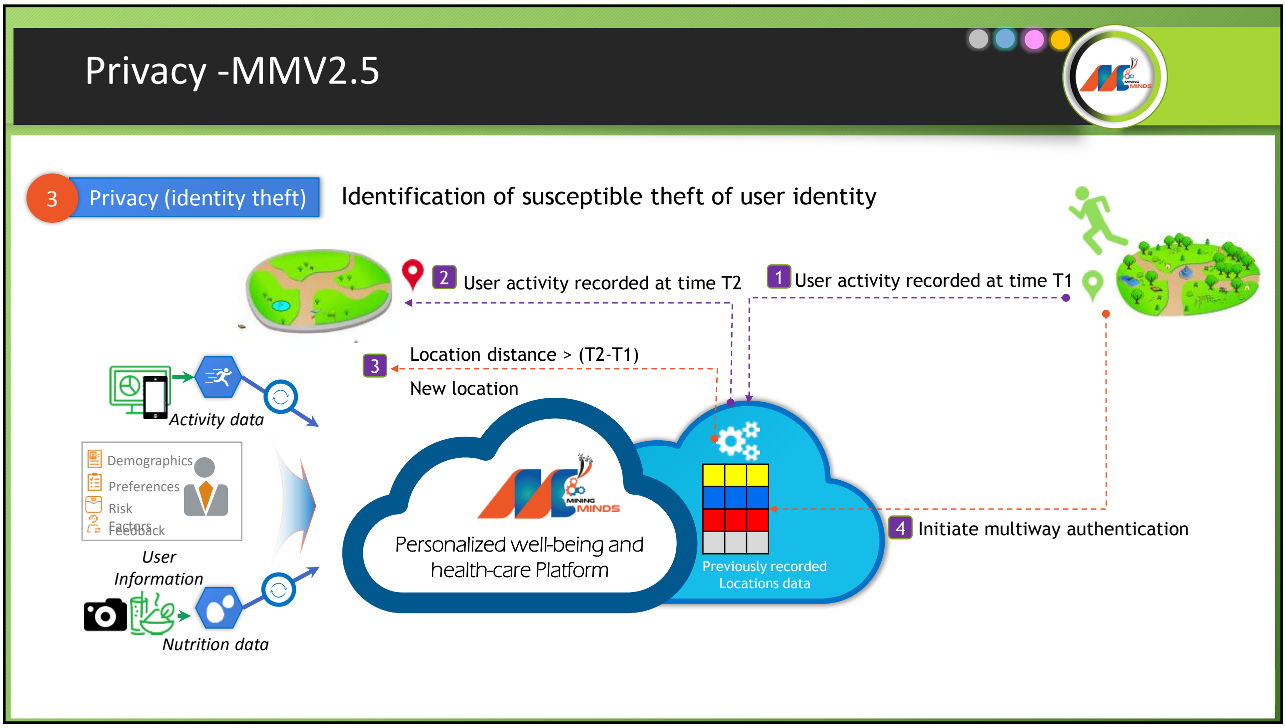


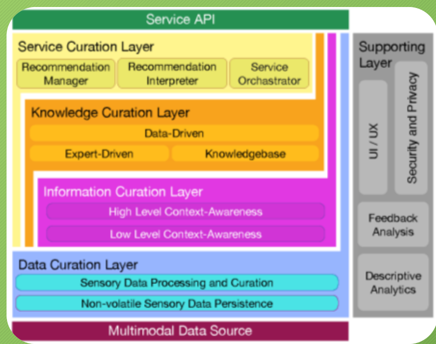
Privacy & Security
Induced Habituation
Expert Recommendation
Personalized Recommendation
Privacy & Security
Human Behavior Quantification



Privacy and Security -MMV2.5
1 Message integrity check
Light food with good fiber amount is recommended for your health in dinner, as you have taken cholesterol enriched food in lunch and try to have 20 min brisk walk as you remained sedentary whole day.
Send
Expert View: Expert feedback to end user
Domain expert
1
2
3
4
5
6
Man in the middle
modification
MMpk (MDE)
MMpk (MDu)
Activity data
Demographics
Preferences
Risk Factors
Feedback
User Information
Nutrition data
Personalized well-being and health-care Platform

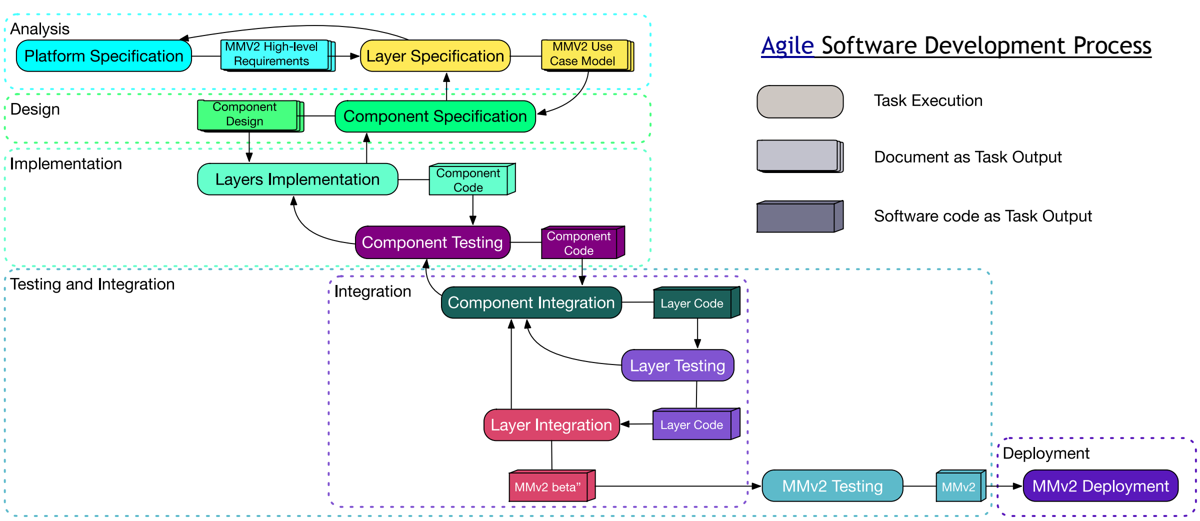

Privacy and Security -MMV2.5
2
Privacy and Security
User information that affects the MM recommendation (User profile)
1 User profile (registration)
3 Acknowledgement
2 Profile confirmation
Activity data
Demographics
Preferences
Risk
Factors
Feedback
User
Information
Nutrition data
Personalized well-being and
health-care Platform
Last updated
current
updated
1 Profile update
confirmation
2 acknowledgement


Privacy -MMV2.5
3
Privacy (identity theft)
Identification of susceptible theft of user identity
2 User activity recorded at time T2
1 User activity recorded at time T1
3
Location distance > (T2-T1)
New location
Activity data
Demographics
Preferences
Risk
Factors
Feedback
User
Information
Nutrition data
Personalized well-being and
health-care Platform
Previously recorded
Locations data
4 Initiate multiway authentication



Mining Minds V2.5 Architecture

Software Engineering Principles



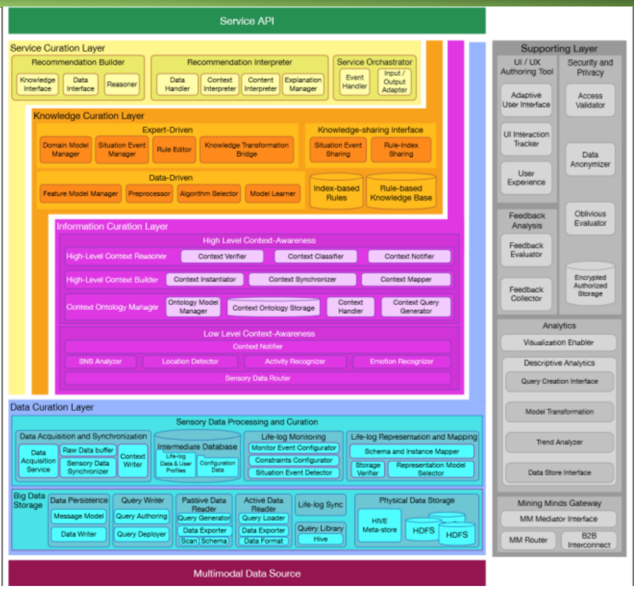
MMv2.5: Architecture

Delivers timely and accurate personalized cross-domain recommendation based on domain knowledge and users preferences/context

Creates and maintains health and wellness knowledge using expert-driven and data-driven approaches

Converts the data obtained from the user interaction with the real and cyberworld, into abstract concepts or categories, such as physical activities, emotional states, locations and social patterns, which are intelligently combined to determine and track context and behavior

Provides real-time data acquisition from multimodal data sources and its persistence using big data technologies. Context data are mapped for life-logging and personalized predictions from life-log

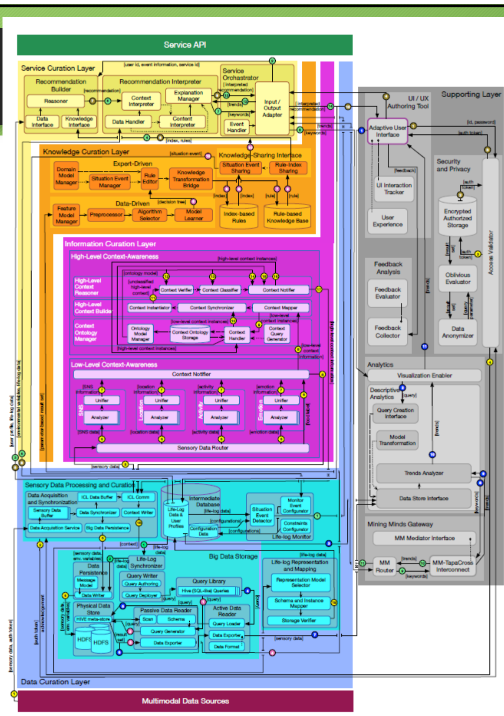


Facilitates information to the users in the most intuitive manner, in a secure environment reflecting their personal needs and preferences

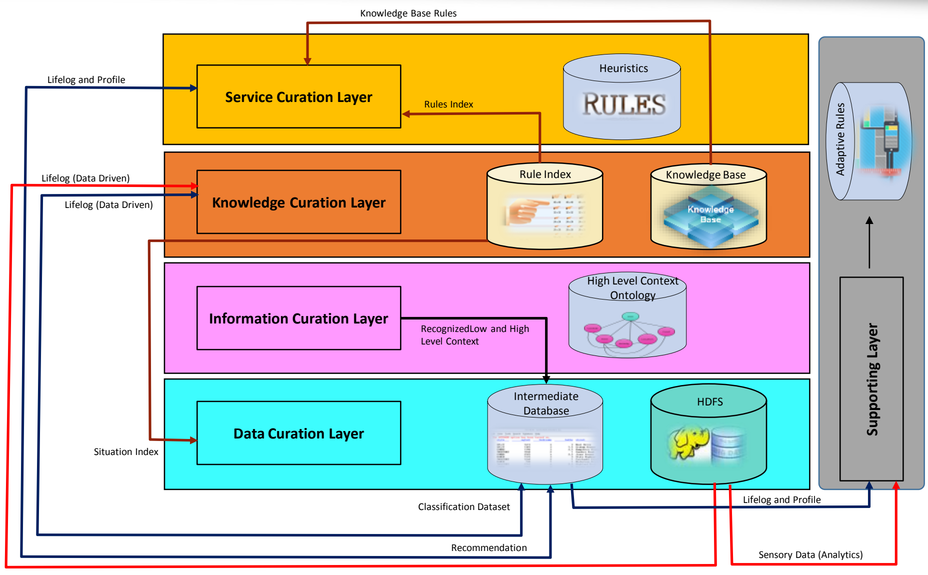
MMv2.5: Operation Flow

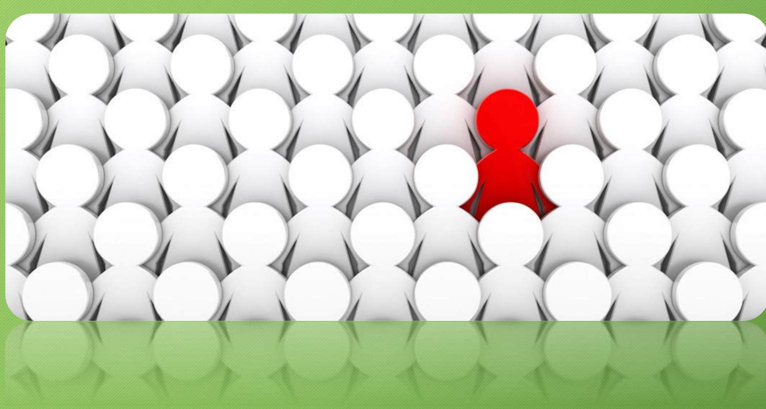
Legend

- Real-time Heterogenous Data Acquisition
- Data-driven & Expert-driven Rules Creation
- Push-model Recommendation Generation
- Recommendation Request (Pull-model)
- Visualization and Analytics Request
- Feedback Request



MMv2.5: Distributed Database Model

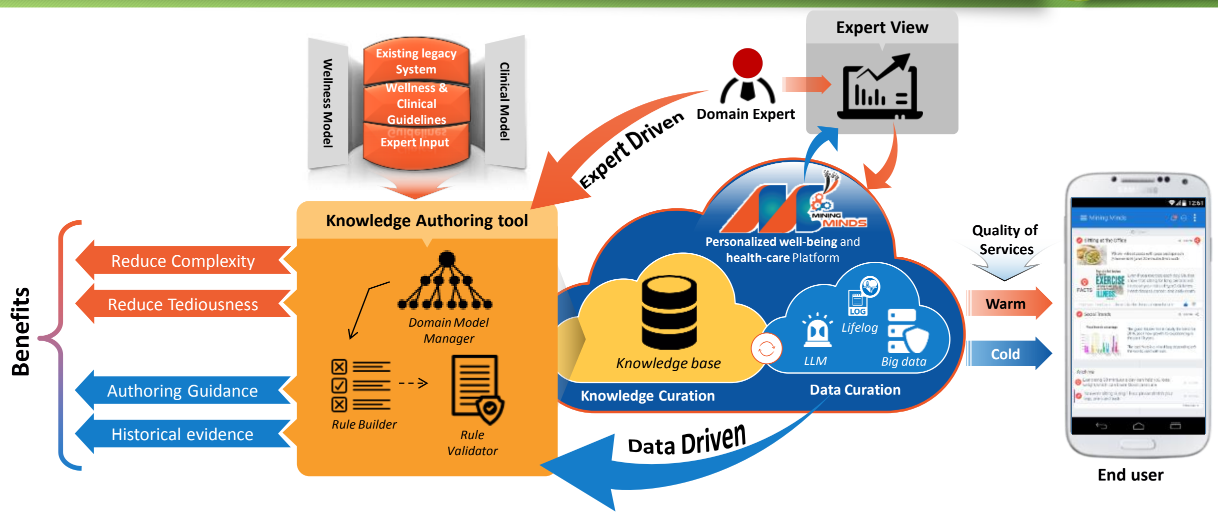


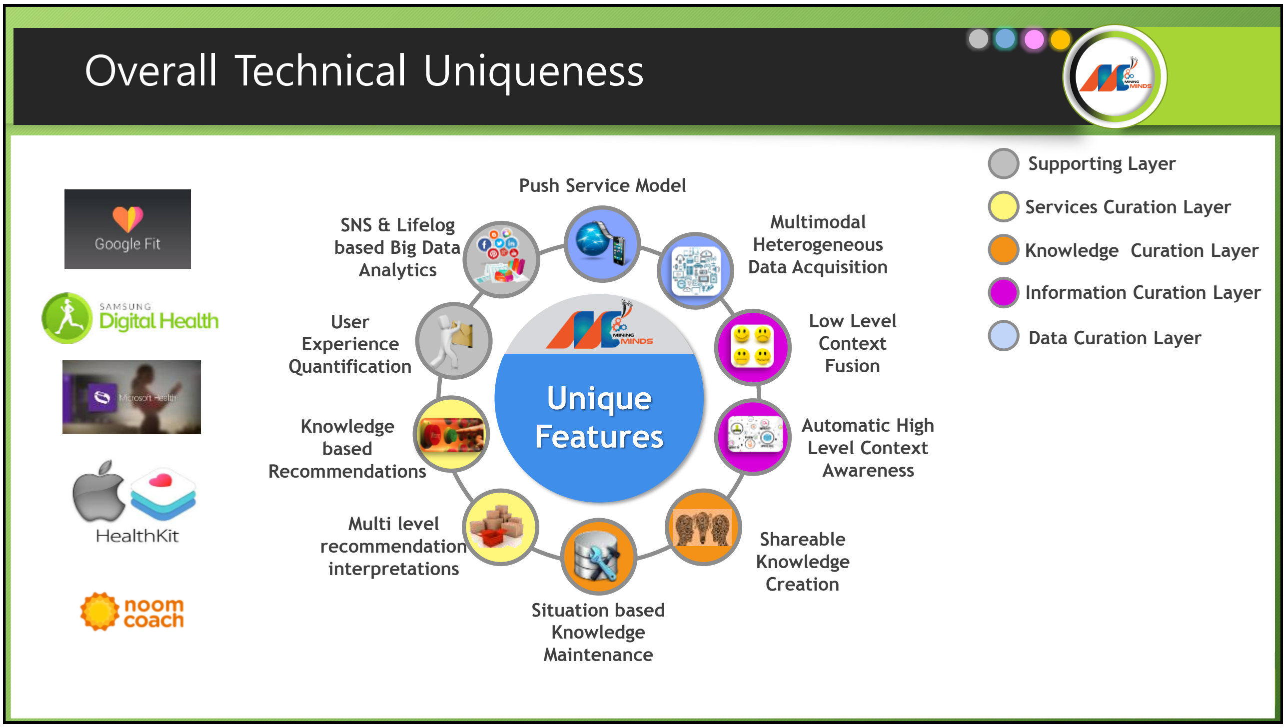


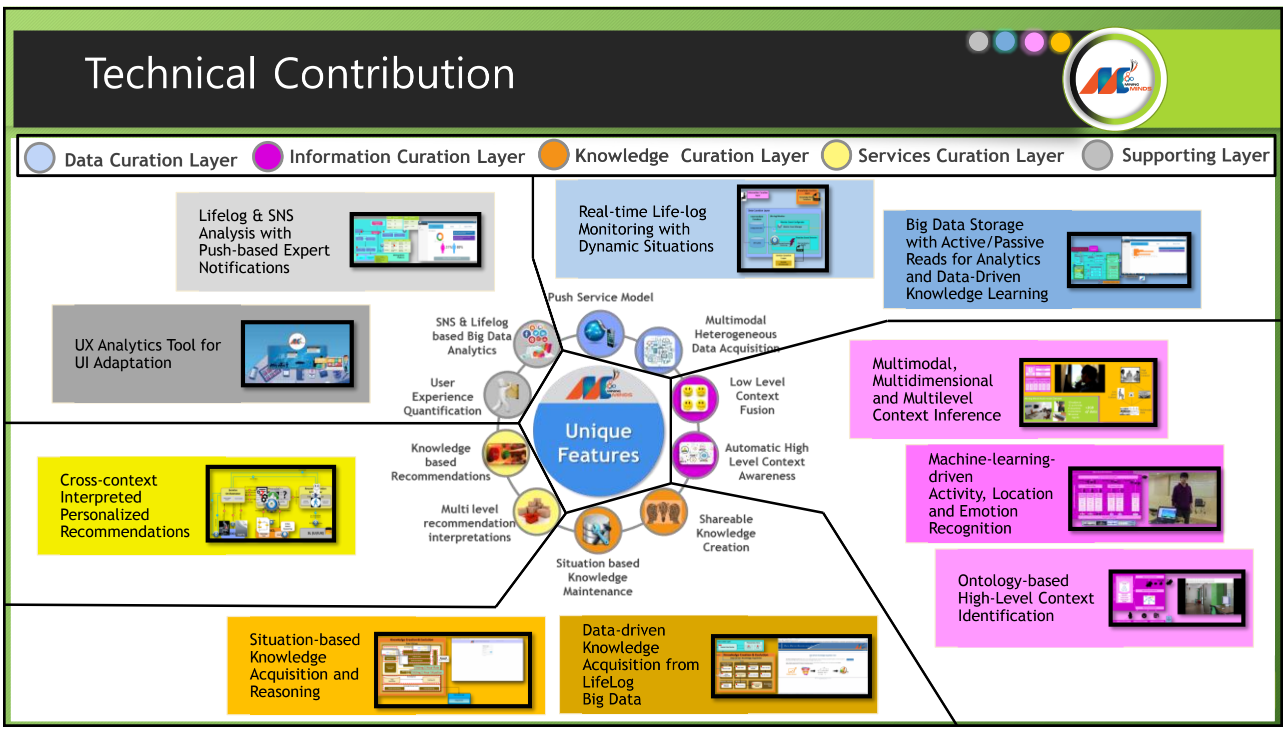
Platform Operations, Uniqueness and Contributions

Platform Operation

MM Service Uniqueness

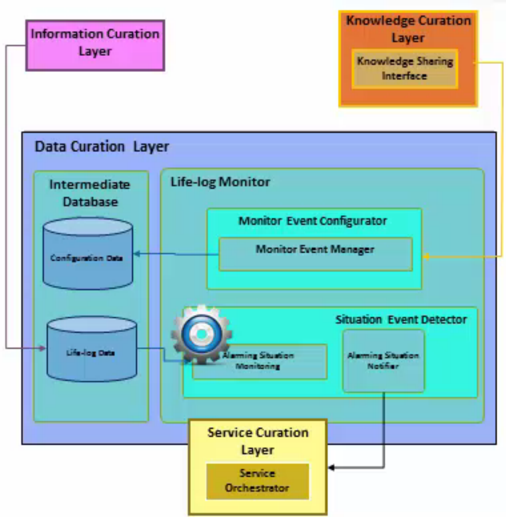



Overall Technical Uniqueness
Google Fit
SAMSUNG Digital Health
HealthKit
noom coach
Push Service Model
SNS & Lifelog based Big Data Analytics
Multimodal Heterogeneous Data Acquisition
User Experience Quantification
Low Level Context Fusion
Unique Features
Knowledge based Recommendations
Automatic High Level Context Awareness
Multi level recommendation interpretations
Shareable Knowledge Creation
Situation based Knowledge Maintenance
Supporting Layer
Services Curation Layer
Knowledge Curation Layer
Information Curation Layer
Data Curation Layer


Technical Contribution
Data Curation Layer
Information Curation Layer
Knowledge Curation Layer
Services Curation Layer
Supporting Layer
Lifelog & SNS Analysis with Push-based Expert Notifications
Real-time Life-log Monitoring with Dynamic Situations
Big Data Storage with Active/Passive Reads for Analytics and Data-Driven Knowledge Learning
UX Analytics Tool for UI Adaptation
Push Service Model
SNS & Lifelog based Big Data Analytics
Multimodal Heterogeneous Data Acquisition
User Experience Quantification
Low Level Context Fusion
Unique Features
Knowledge based Recommendations
Automatic High Level Context Awareness
Multi level recommendation interpretations
Shareable Knowledge Creation
Situation based Knowledge Maintenance
Multimodal, Multidimensional and Multilevel Context Inference
Machine-learning-driven Activity, Location and Emotion Recognition
Cross-context Interpreted Personalized Recommendations
Ontology-based High-Level Context Identification
Situation-based Knowledge Acquisition and Reasoning
Data-driven Knowledge Acquisition from LifeLog Big Data

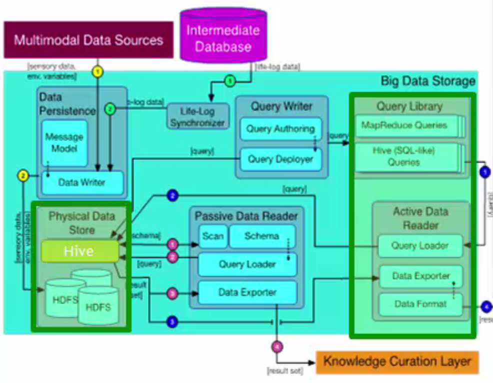
Mining Minds Versions Microdemos

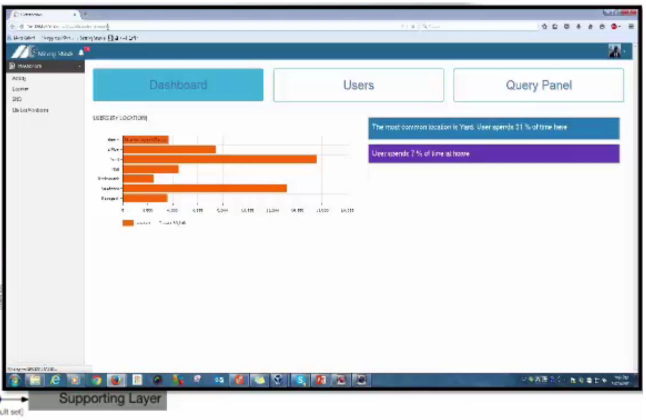
DCL Real-time Life-log Monitoring with Dynamic Situations



DCL

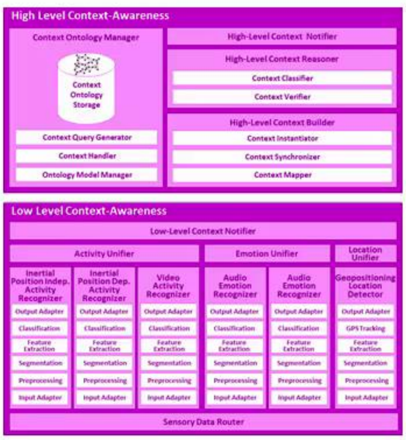
Big Data Storage with Active and Passive Reads for Analytics and Data-Driven Knowledge Model Learning

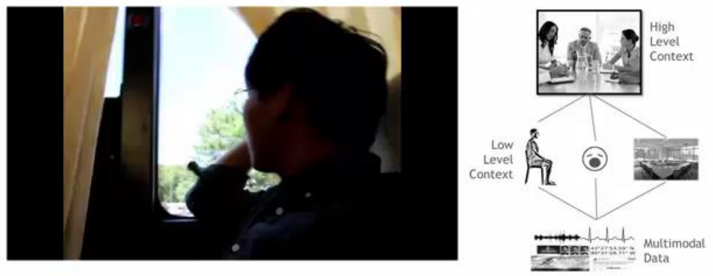




ICL

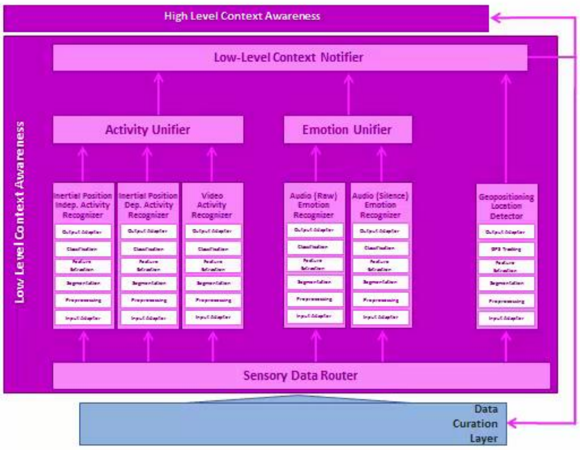
Multimodal, Multidimensional and Multilevel Context Inference





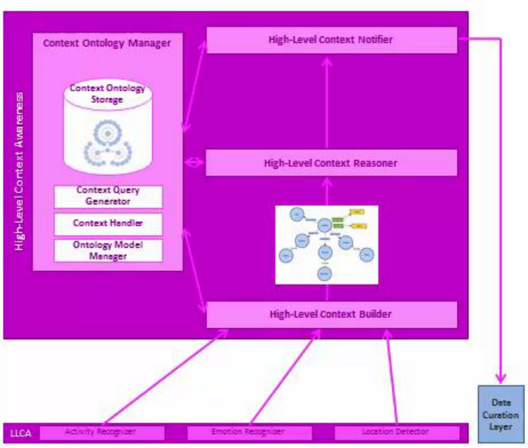
ICL

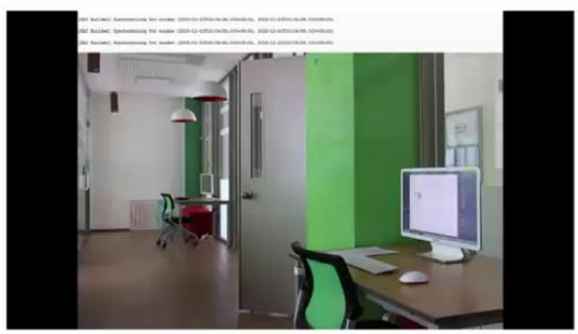
Machine-learning-driven Activity, Location and Emotion Recognition



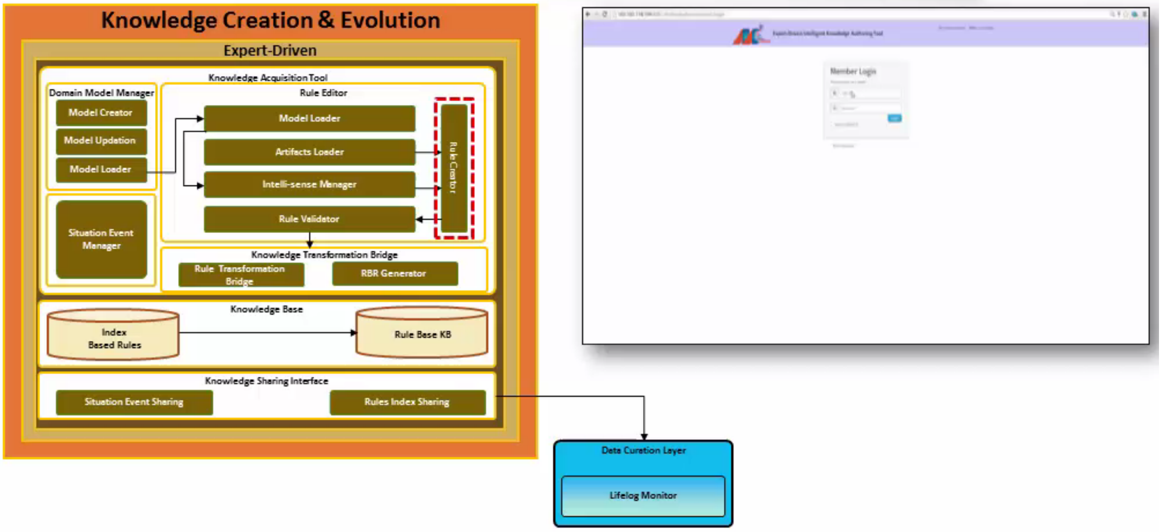
ICL

Ontology-based High-Level Context Identification

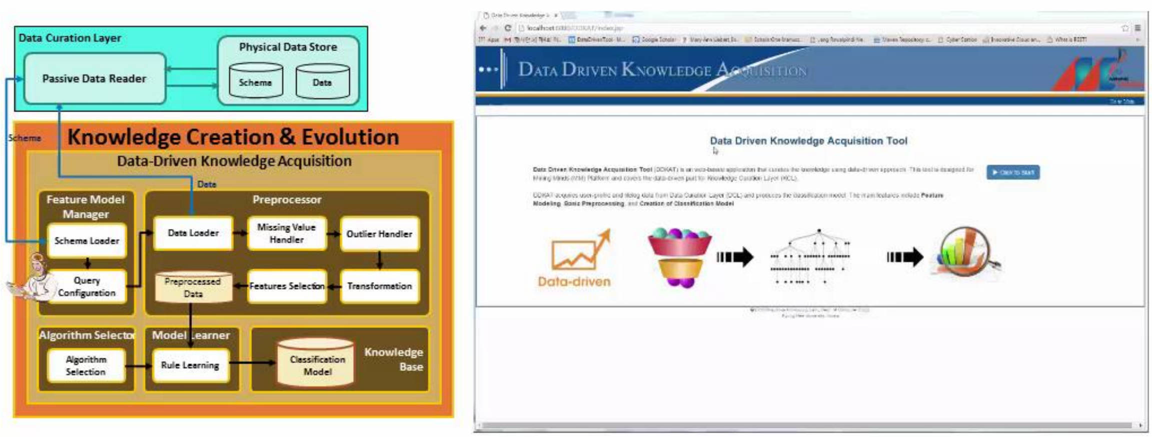




KCL Situation-based Knowledge Acquisition and Reasoning

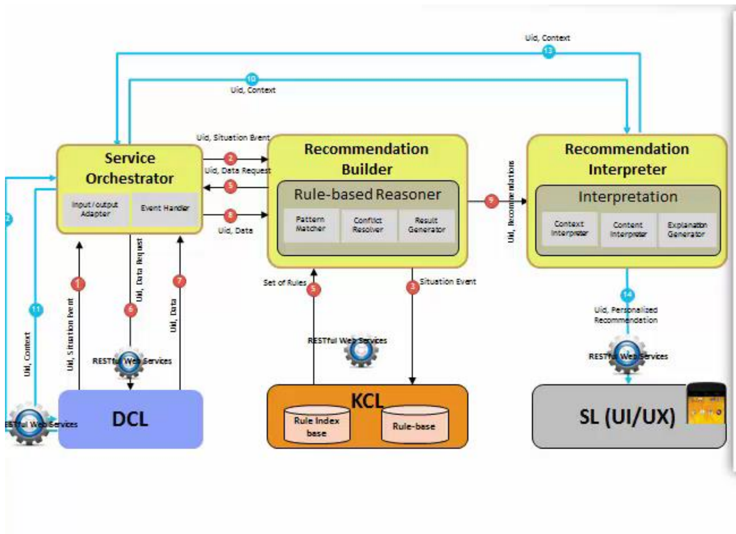


KCL Data-driven Knowledge Acquisition from LifeLog Big Data



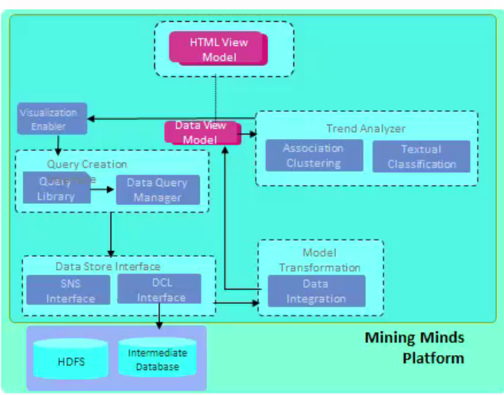
SCL

Cross-context Interpreted Personalized Recommendations

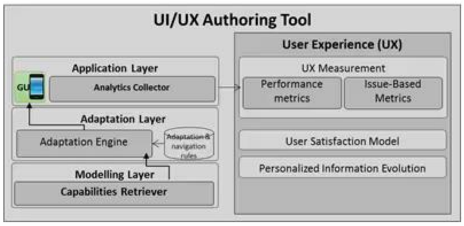


SL

Lifelog & SNS Analysis with Push-based Expert Notifications



SL UX Analytics Tool for UI Adaptation



Comparison with Existing Systems

Comparison with Existing Systems

Platforms

- Google Fit
- Samsung S Health
- Microsoft Health
- Apple Health Kit
- Open mHealth

Applications

- NoomCoach
- Fitbit
- Argus
- Runtastic
- RunKeeper
- ZombieRun

Services

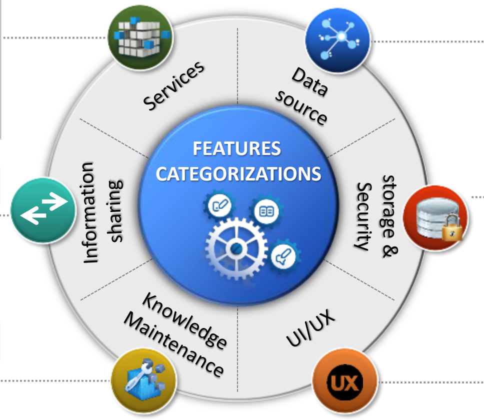
1. Activity Recognition
2. Expert Services
3. Wellness services
4. Personalized recommendations
5. Clinical services
6. SDK/API

Information Sharing

1. With other apps
2. Social media sharing
3. Other users (authorized circle)

Knowledge Maintenace

1. Open knowledge
2. Knowledge acquisition
3. Knowledge evolution



Data Soruce

1. Sensory Data
2. User profile
3. IOT
4. Other apps
5. Clinical data
6. Social media

Storage & Security

1. User device storage
2. Cloud storage
3. Bigdata storage
4. Encrypted data storage
5. Anonymized access

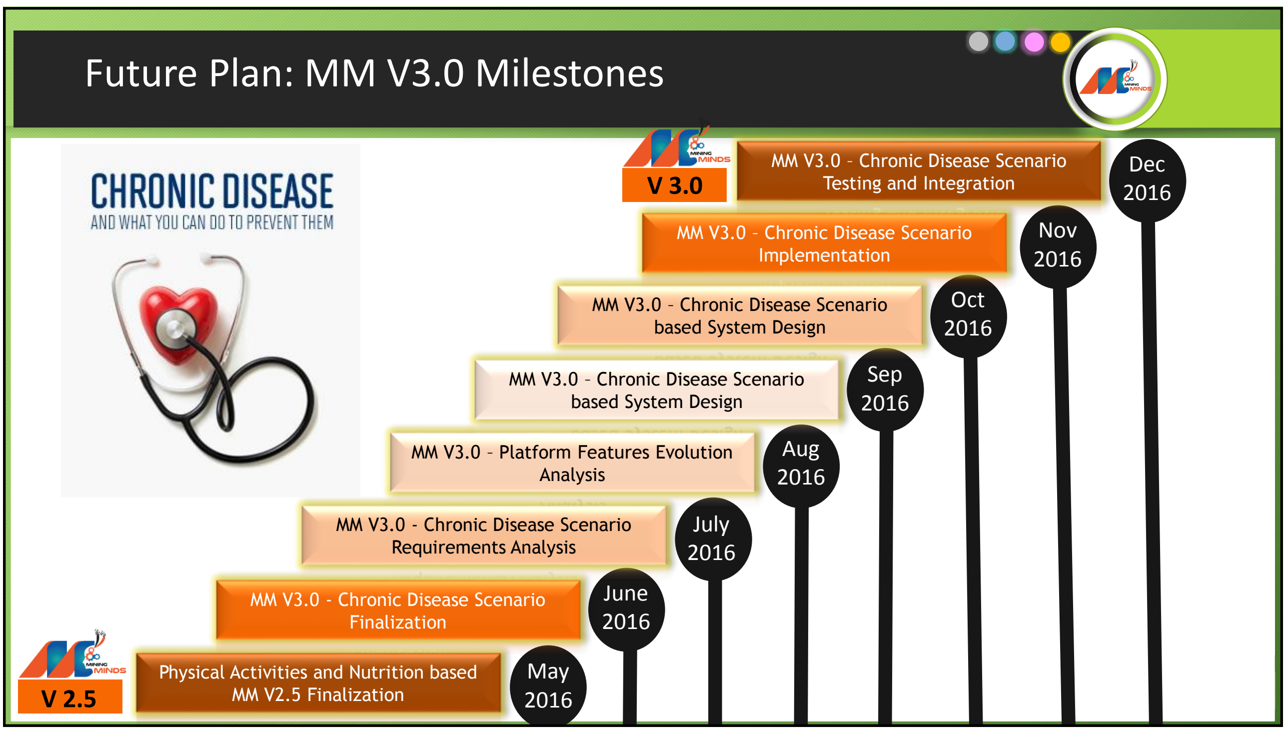
UI/UX

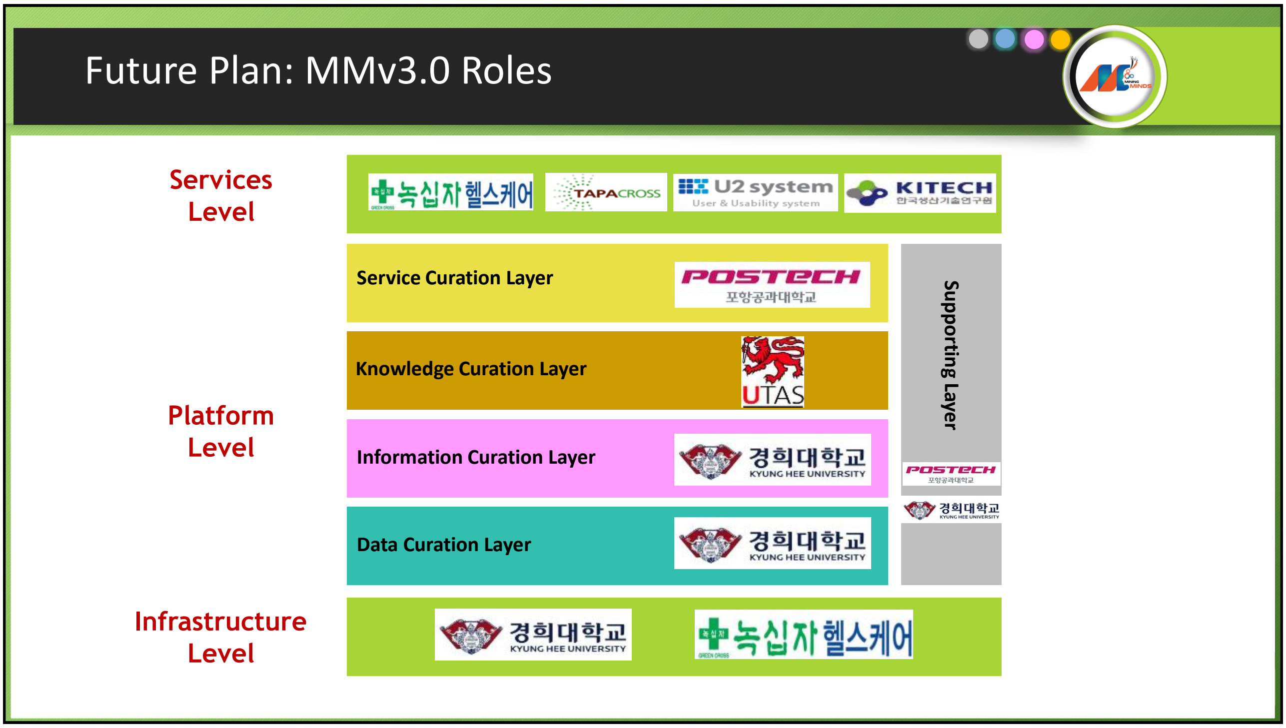
1. User Experience
2. User Modelling
3. Adaptation of UI

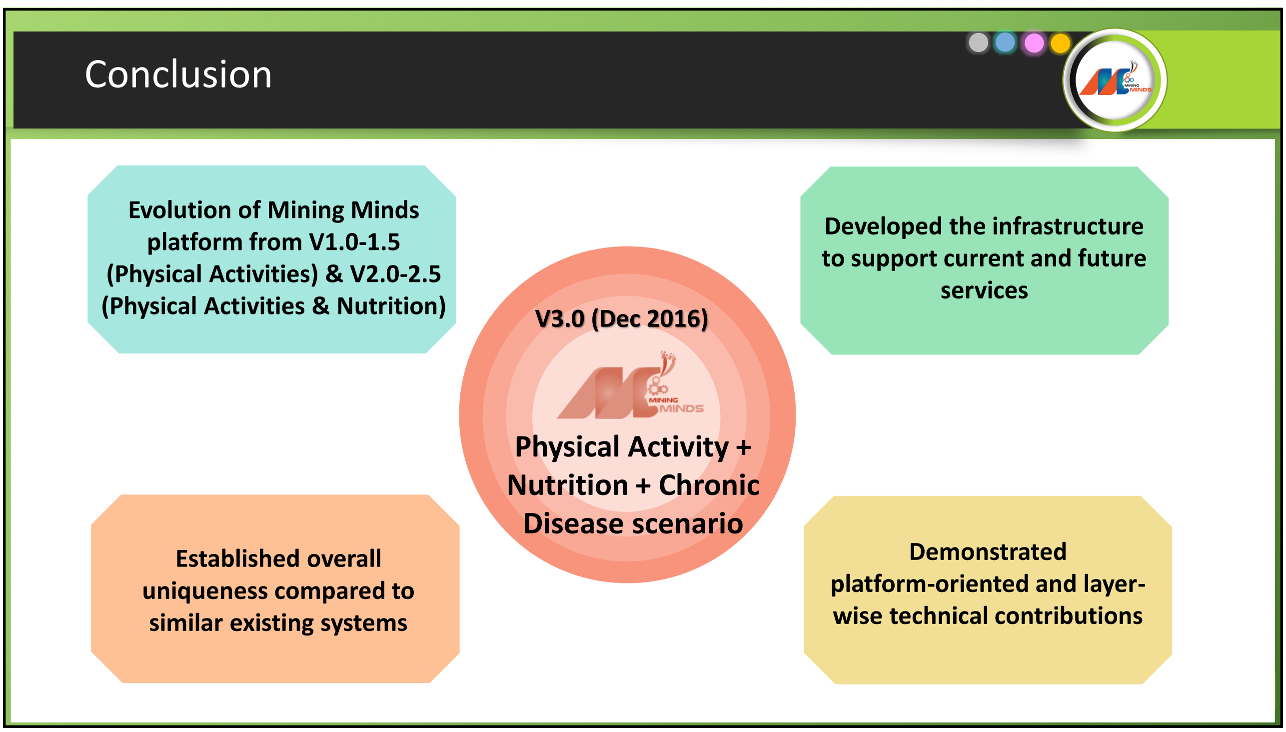
	Data Source						Storage & Security					UI/UX		
	Sensory Data	User profile	IOT	Other apps	Clinical data	Social media	User device	Cloud storage	Big data storage	Encryp storagey	Anon access	User Experience	User Modelling	Adaptation of UI
Google Fit	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	✗	✗	✓	✗
Samsung S health	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	✗	✗	✓	✗
Microsoft health	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✗
Apple healthkit	✓	✓	✓	✓	✗	✗	✓	✓	✗	✓	✗	✗	✓	✗
Open mhealth	✓	✓	✓	✓	✗	✗	✓	✓	✗	✗	✗	✗	✓	✗
Fitbit	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	✗	✓	✓	✗
NoomCoach	✓	✓	✓	✓	✗	✗	✓	✗	✗	✗	✗	✗	✓	✗
Argus	✓	✓	✓	✗	✗	✗	✓	✓	✗	✓	✗	✗	✓	✗
Runtastic	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	✗	✗	✓	✗
Runkeeper	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	✗	✗	✓	✗
Zombie Run	✓	✓	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗	✓	✗
Mining Minds	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

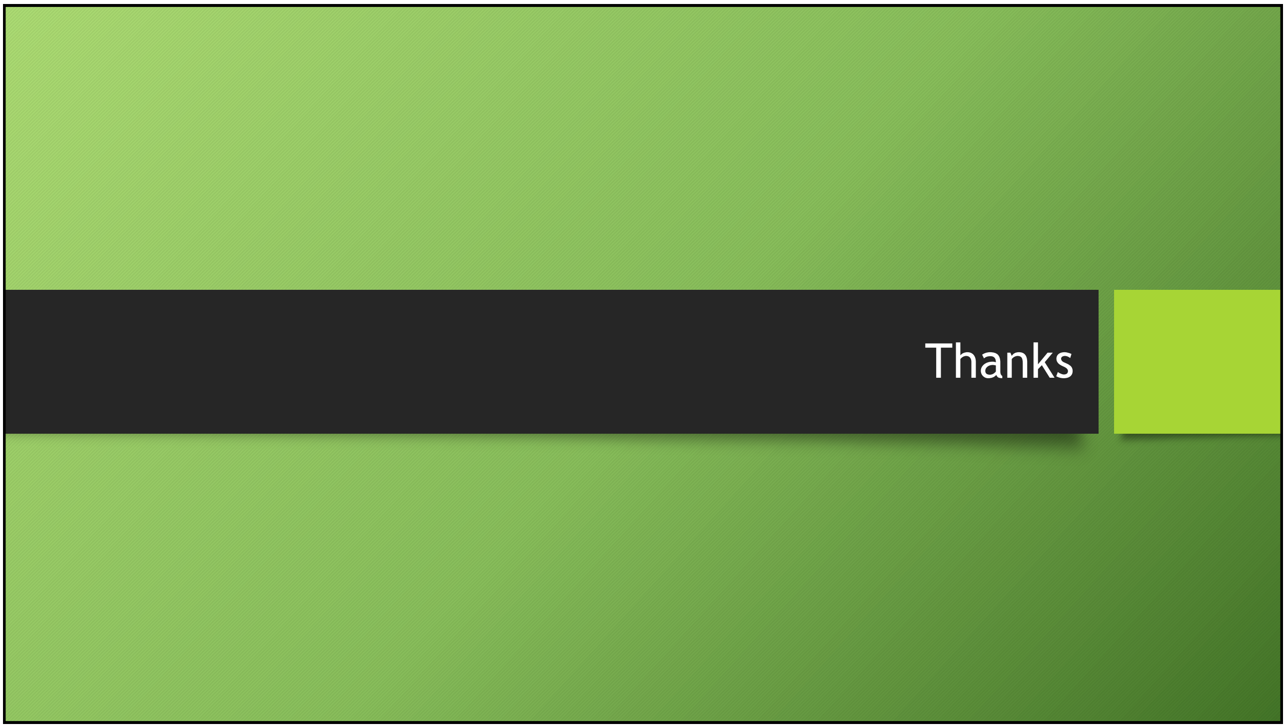
	Services						Information Sharing			Knowledge Maintenance		
	Activity Recog	Expert Svc	Wellness svc	Personal recom	Clinical svc	SDK/API	With other apps	Social media	Other users	Open Know	Know acq	Know evolution
Google Fit	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✗
Samsung S health	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
Microsoft health	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Apple healthkit	✓	✓	✓	✓	✗	✓	✓	✓	✓	✗	✓	✗
Open mhealth	✓	✓	✓	✓	✗	✓	✗	✓	✓	✗	✓	✗
Fitbit	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
NoomCoach	✓	✓	✓	✓	✗	✗	✓	✓	✗	✗	✓	✗
Argus	✓	✓	✓	✗	✗	✗	✗	✓	✗	✗	✓	✗
Runtastic	✓	✓	✓	✓	✗	✓	✓	✓	✗	✓	✓	✗
Runkeeper	✓	✓	✓	✗	✗	✓	✓	✓	✗	✓	✓	✗
Zombie Run	✓	✗	✓	✗	✗	✗	✗	✓	✗	✗	✓	✗
Mining Minds	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓


FUTURE
Future Plan


Future Plan: MM V3.0 Milestones
CHRONIC DISEASE
AND WHAT YOU CAN DO TO PREVENT THEM
V 3.0
MM V3.0 - Chronic Disease Scenario Testing and Integration
Dec 2016
MM V3.0 - Chronic Disease Scenario Implementation
Nov 2016
MM V3.0 - Chronic Disease Scenario based System Design
Oct 2016
MM V3.0 - Chronic Disease Scenario based System Design
Sep 2016
MM V3.0 - Platform Features Evolution Analysis
Aug 2016
MM V3.0 - Chronic Disease Scenario Requirements Analysis
July 2016
MM V3.0 - Chronic Disease Scenario Finalization
June 2016
V 2.5
Physical Activities and Nutrition based MM V2.5 Finalization
May 2016


Future Plan: MMv3.0 Roles
Services Level
녹십자헬스케어
TAPACROSS
U2 system
User & Usability system
KITECH
Platform Level
Service Curation Layer
POSTECH
포항공과대학교
Knowledge Curation Layer
UTAS
Information Curation Layer
경희대학교
KYUNG HEE UNIVERSITY
Data Curation Layer
경희대학교
KYUNG HEE UNIVERSITY
Supporting Layer
POSTECH
경희대학교
Infrastructure Level
경희대학교
KYUNG HEE UNIVERSITY
녹십자헬스케어


Conclusion
Evolution of Mining Minds platform from V1.0-1.5 (Physical Activities) & V2.0-2.5 (Physical Activities & Nutrition)
Developed the infrastructure to support current and future services
V3.0 (Dec 2016)
MINING MINDS
Physical Activity + Nutrition + Chronic Disease scenario
Established overall uniqueness compared to similar existing systems
Demonstrated platform-oriented and layer-wise technical contributions


Thanks

Section 3

Mining Minds Features Evolution

Mining Minds Version 2.5

Feature Evolution

경희대학교 KYUNG HEE UNIVERSITY KITECH 한국생산기술연구원 POSTECH 포항공과대학교 UTAS 녹십자헬스케어 TAPACROSS U2 system User & Usability system

DATA CURATION LAYER

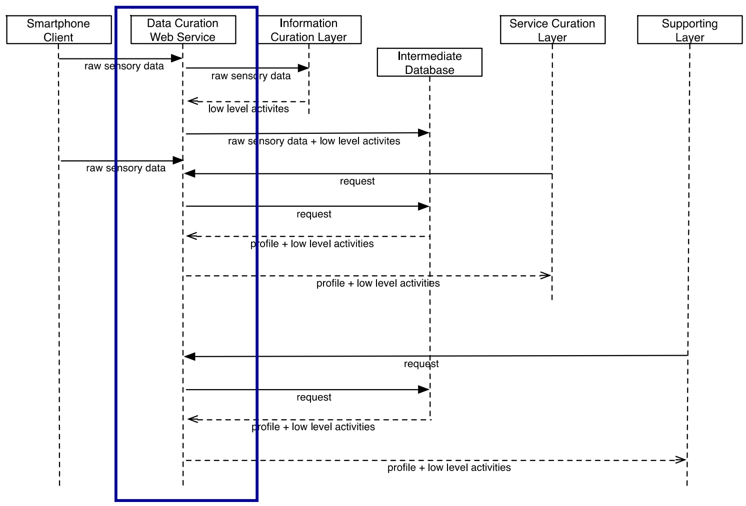
경희대학교 KYUNG HEE UNIVERSITY KITECH 한국생산기술연구원 POSTECH 포항공과대학교 UTAS 녹십자헬스케어 TAPACROSS U2 system User & Usability system

Challenges and solutions (MMv2.5-DCL)

1. Communication

Benefit | Limitation

V1. SOAP-based Web services → Faster Implementation → Performance bottleneck



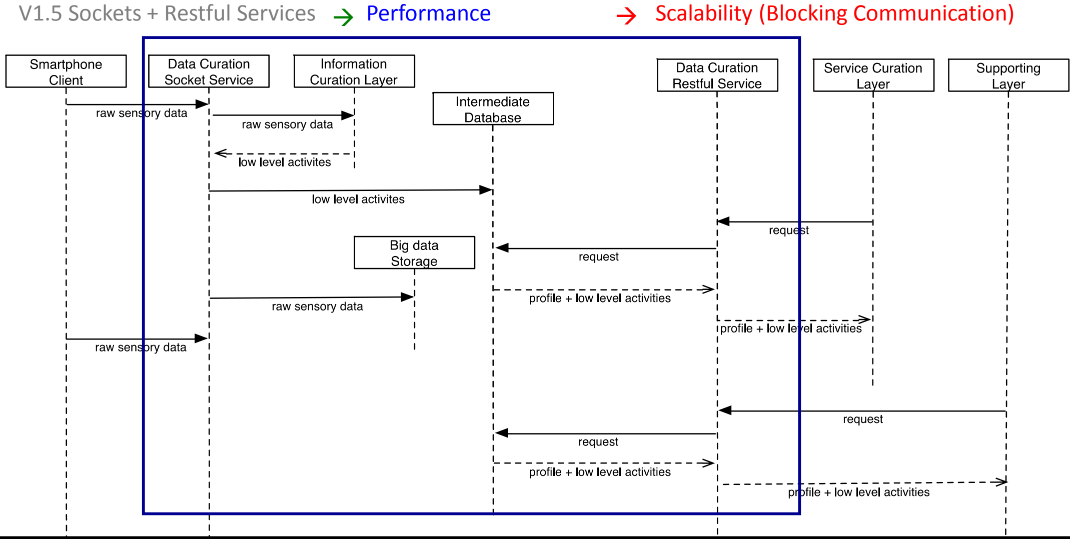
Challenges and solutions (MMv2.5-DCL)

1. Communication

Benefit | Limitation

V1. SOAP-based Web services → Faster Implementation → Performance bottleneck

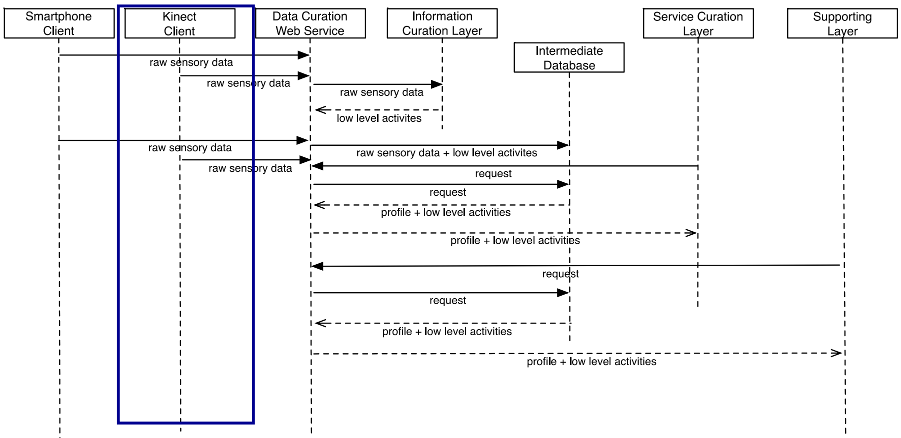
V1.5 Sockets + Restful Services → Performance → Scalability (Blocking Communication)



Challenges and solutions (MMv2.5-DCL)

1. Communication

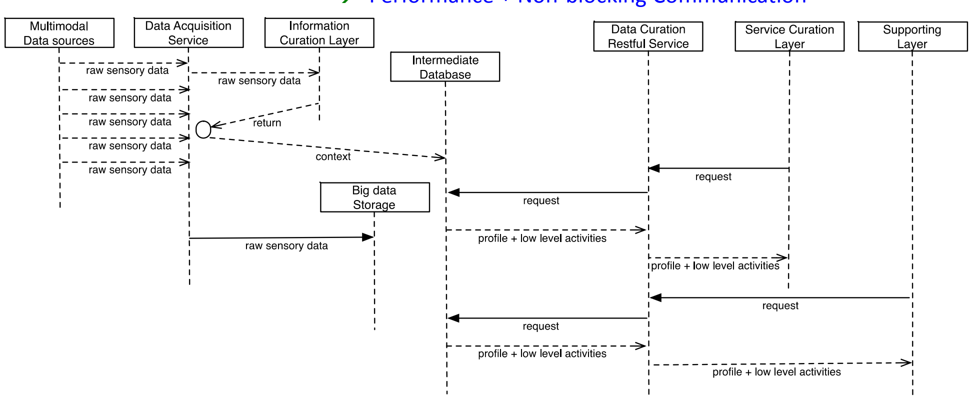
		Benefit		Limitation
V1. SOAP-based Web services	→	Faster Implementation	→	Performance bottleneck
V1.5 Sockets + Restful Services	→	Performance	→	Scalability (Blocking Communication)
V2 Sockets + Restful Services	→	Performance	→	Scalability (Blocking Communication)



Challenges and solutions (MMv2.5-DCL)

1. Communication

		Benefit		Limitation
V1. SOAP-based Web services	→	Faster Implementation	→	Performance bottleneck
V1.5 Sockets + Restful Services	→	Performance	→	Scalability (Blocking Communication)
V2 Sockets + Restful Services	→	Performance	→	Scalability (Blocking Communication)
V2.5 Node.js + Restful Services	→	Performance + Non-blocking Communication		



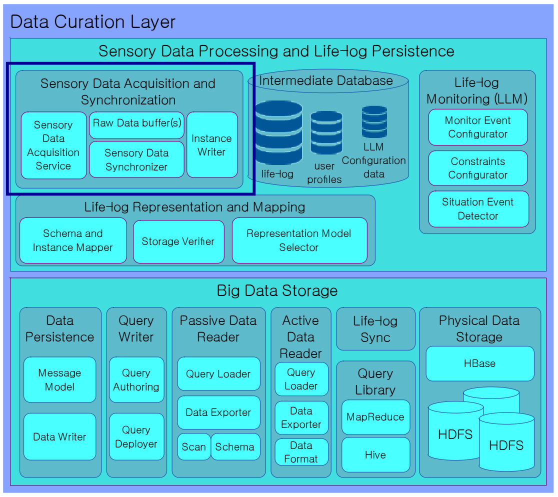
Challenges and solutions (MMv2.5-DCL)

1. Communication

V2.5 Node.js + Restful Services

→

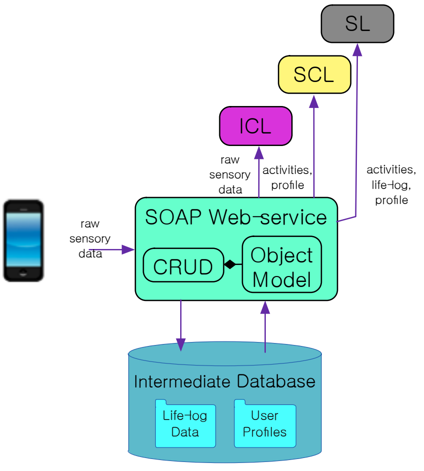
Performance + Non-blocking Communication



Challenges and solutions (MMv2.5-DCL)

2. Big Data + Intermediate DB

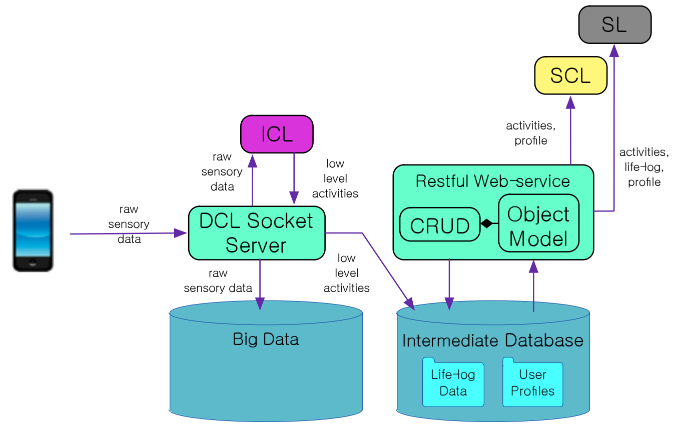
		Benefit		Limitation
V1. Only Intermediate DB	→	Faster Implementation	→	Intermediate DB Stress (Performance Bottleneck)



Challenges and solutions (MMv2.5-DCL)

2. Big Data + Intermediate DB

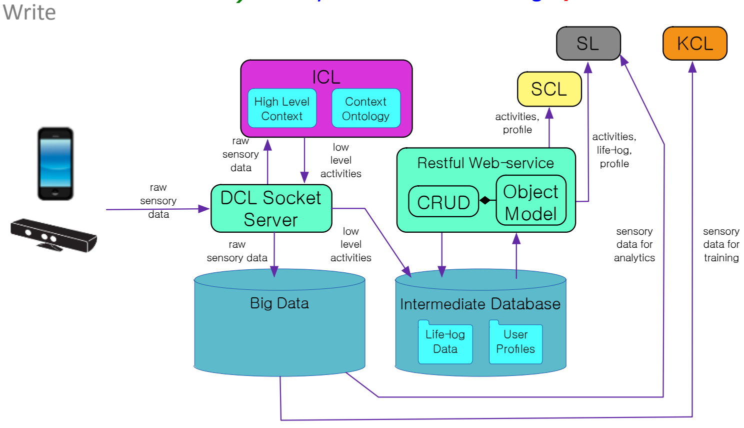
		Benefit		Limitation
V1. Only Intermediate DB	→	Faster Implementation	→	Intermediate DB Stress (Performance Bottleneck)
V1.5 Big Data For Sensory Data Write	→	Performance	→	No Analytics on Sensory Data + No Data for Training



Challenges and solutions (MMv2.5-DCL)

2. Big Data + Intermediate DB

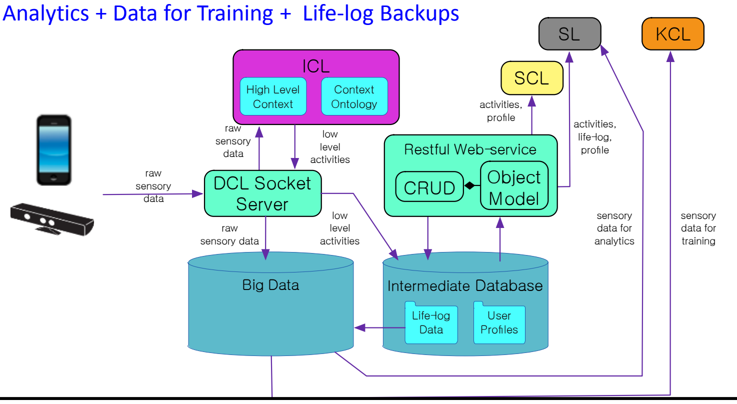
		Benefit		Limitation
V1. Only Intermediate DB	→	Faster Implementation	→	Intermediate DB Stress (Performance Bottleneck)
V1.5 Big Data For Sensory Data Write	→	Performance	→	No Analytics on Sensory Data + No Data for Training
V2 Big Data For Sensory Data Read + Write	→	Analytics + Data for Training	→	No Periodic Backups for Life-log data



Challenges and solutions (MMv2.5-DCL)

2. Big Data + Intermediate DB

		Benefit		Limitation
V1. Only Intermediate DB	→	Faster Implementation	→	Intermediate DB Stress (Performance Bottleneck)
V1.5 Big Data For Sensory Data Write	→	Performance	→	No Analytics on Sensory Data + No Data for Training
V2 Big Data For Sensory Data Read + Write	→	Analytics + Data for Training	→	No Periodic Backups for Life-log data
V2.5 Big Data For Sensory Data Read + Write + life-log	→	Analytics + Data for Training + Life-log Backups		



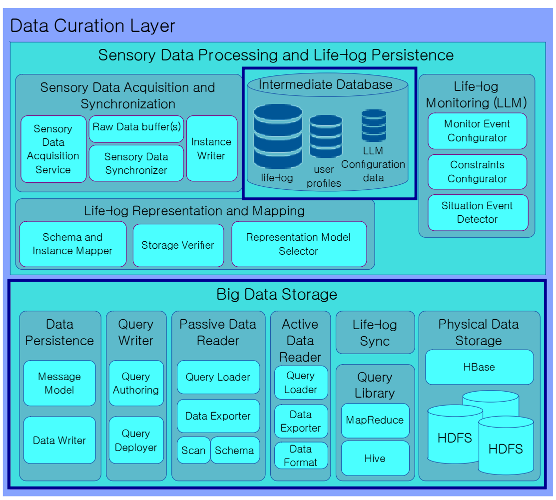
Challenges and solutions (MMv2.5-DCL)

2. Big Data + Intermediate DB

V2.5 Big Data For Sensory Data Read + Write + life-log

→

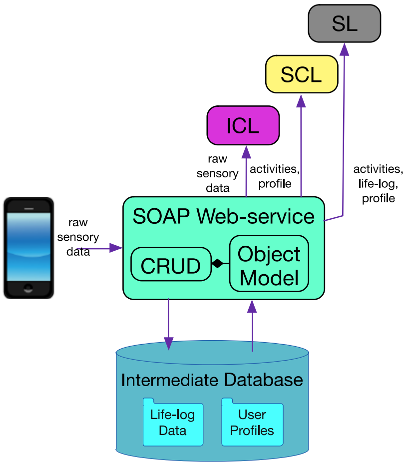
Analytics + Data for Training + Life-log Backups



Challenges and solutions (MMv2.5-DCL)

3. Life-log Representation and Mapping

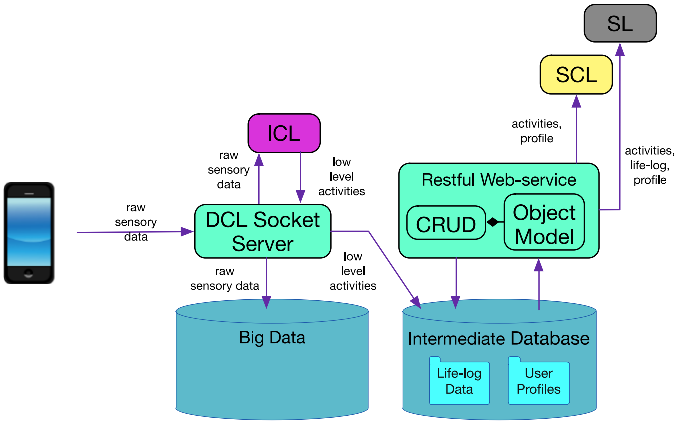
		Benefit		Limitation
V1. Object Model	→	Performance	→	No Inference (No High Level Context in ICL)



Challenges and solutions (MMv2.5-DCL)

3. Life-log Representation and Mapping

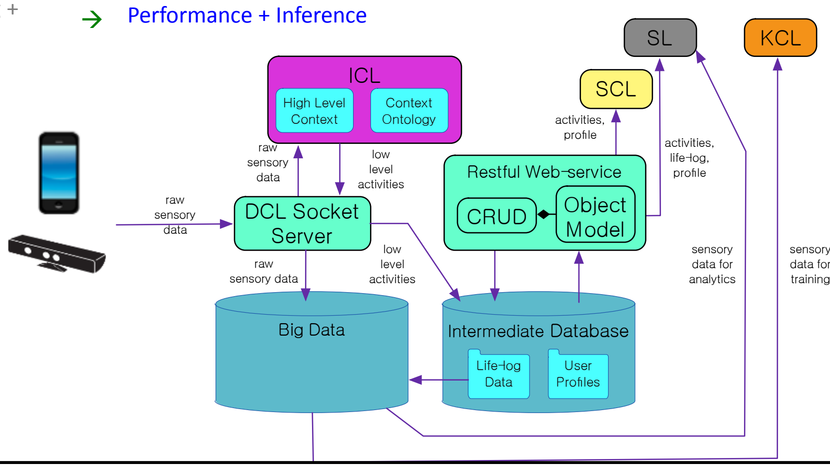
		Benefit		Limitation
V1. Object Model	→	Performance	→	No Inference (No High Level Context in ICL)
V1.5 Object Model	→	Performance	→	No Inference (No High Level Context in ICL)



Challenges and solutions (MMv2.5-DCL)

3. Life-log Representation and Mapping

		Benefit		Limitation
V1. Object Model	→	Performance	→	No Inference (No High Level Context in ICL)
V1.5 Object Model	→	Performance	→	No Inference (No High Level Context in ICL)
V2.x Object Oriented Life-log + Ontology for High Level Context at ICL	→	Performance + Inference		



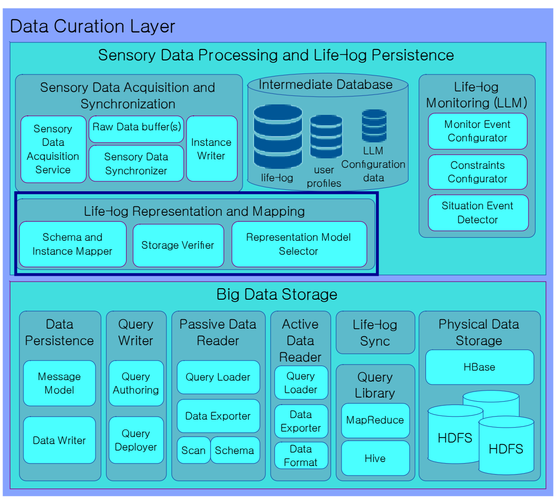
Challenges and solutions (MMv2.5-DCL)

3. Life-log Representation and Mapping

V2.x Object Oriented Life-log + Ontology for High Level Context at ICL

→

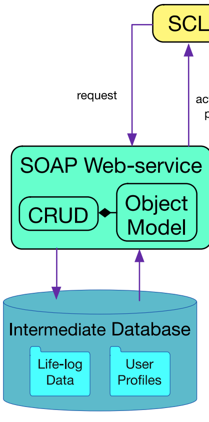
Performance + Inference



Challenges and solutions (MMv2.5-DCL)

4. Life-log Monitoring and Prediction

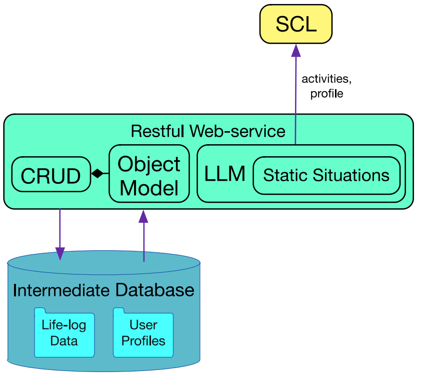
		Benefit		Limitation
V1. NA	→	NA	→	No trigger/situation based recommendations



Challenges and solutions (MMv2.5-DCL)

4. Life-log Monitoring and Prediction

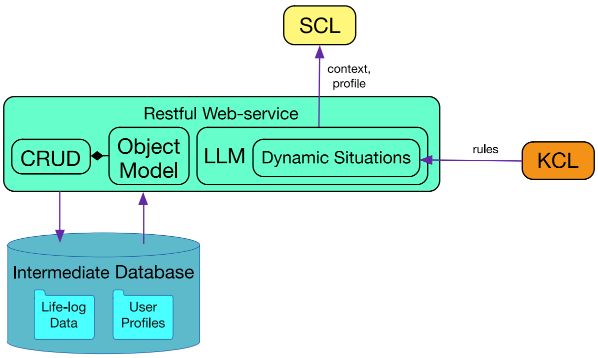
		Benefit		Limitation
V1. NA	→	NA	→	No trigger/situation based recommendations
V1.5 Life-log Monitoring (LLM)	→	Trigger-based recommendation generation	→	Design-time static situations



Challenges and solutions (MMv2.5-DCL)

4. Life-log Monitoring and Prediction

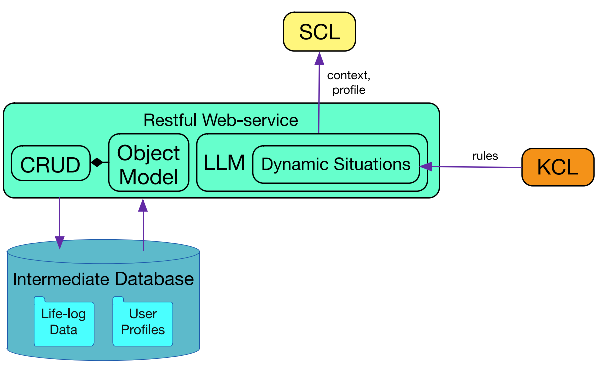
		Benefit		Limitation
V1. NA	→	NA	→	No trigger/situation based recommendations
V1.5 Life-log Monitoring (LLM)	→	Trigger-based recommendation generation	→	Design-time static situations
V2 Life-log Monitoring (LLM)	→	Dynamic situations with runtime-invocation	→	Only Physical Activities monitoring



Challenges and solutions (MMv2.5-DCL)

4. Life-log Monitoring and Prediction

		Benefit		Limitation
V1. NA	→	NA	→	No trigger/situation based recommendations
V1.5 Life-log Monitoring (LLM)	→	Trigger-based recommendation generation	→	Design-time static situations
V2 Life-log Monitoring (LLM)	→	Dynamic situations with runtime-invocation	→	Only Physical Activities monitoring
V2.5 Life-log Monitoring (LLM)	→	Physical Activities and Nutrition Based Monitoring		



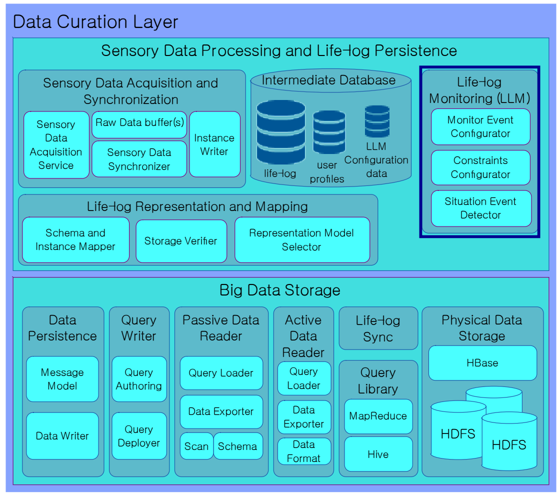
Challenges and solutions (MMv2.5-DCL)

4. Life-log Monitoring and Prediction

V2.5 Life-log Monitoring (LLM)

→

Physical Activities and Nutrition Based Monitoring

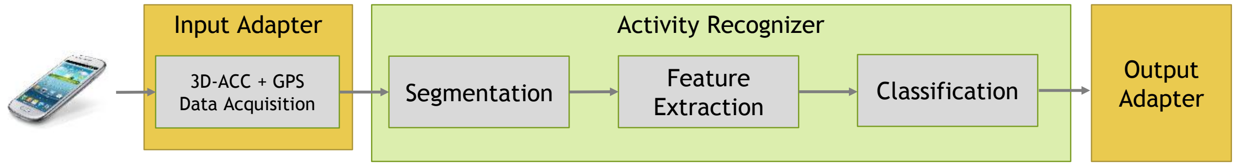


Information Curation Layer (ICL)

Challenges and solutions (MMv2.5-ICL)

1. Human activity recognition (AR)

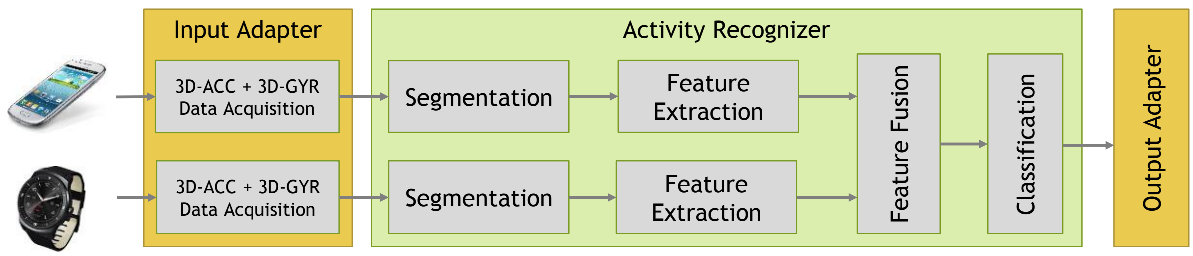
	Benefit	Limitation
V1. Smartphone-based AR	→ High accuracy	→ Only 5 activities



Challenges and solutions (MMv2.5-ICL)

1. Human activity recognition (AR)

	Benefit	Limitation
V1. Smartphone-based AR	→ High accuracy	→ Only 5 activities
V1.5 Smartphone and Wearable-based AR	→ More activities (up to 15)	→ Poor recognition performance for some activities (e.g., “eating”)

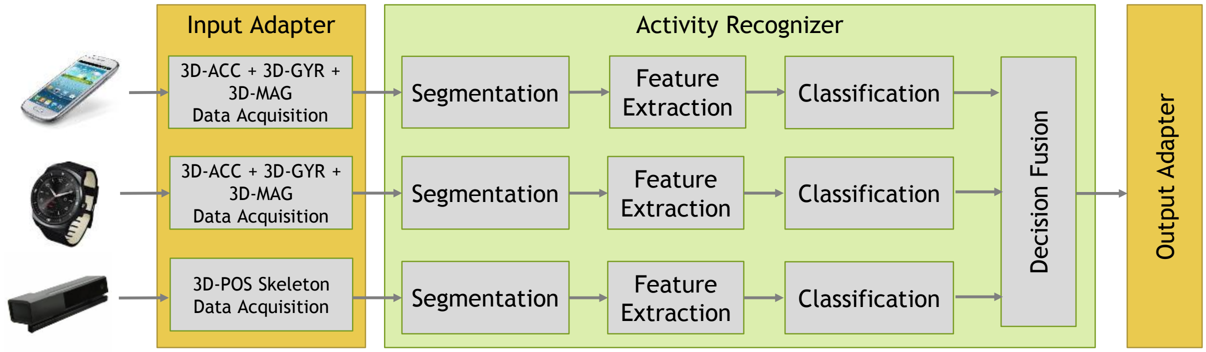


Challenges and solutions (MMv2.5-ICL)

1. Human activity recognition (AR)

	Benefit	Limitation
V1. Smartphone-based AR	→ High accuracy	→ Only 5 activities
V1.5 Smartphone and Wearable-based AR	→ More activities (up to 15)	→ Poor recognition performance for some activities (e.g., "eating")
V2 Smartphone, Wearable and Video-based AR (multisensor fusion)	→ High accuracy with the support of different sensor	

Confidence level: Medium

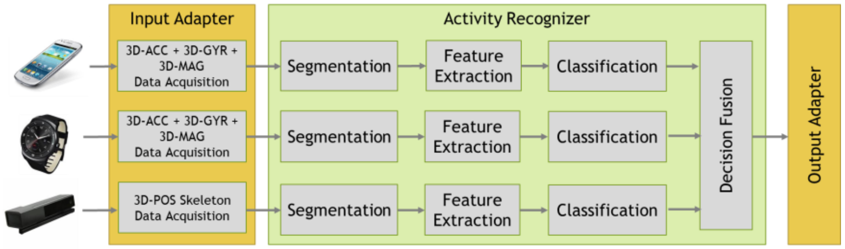


Challenges and solutions (MMv2.5-ICL)

1. Human activity recognition (AR)

	Benefit	Limitation
V1. Smartphone-based AR	→ High accuracy	→ Only 5 activities
V1.5 Smartphone and Wearable-based AR	→ More activities (up to 15)	→ Poor recognition performance for some activities (e.g., "eating")
V2 Smartphone, Wearable and Video-based AR (multisensor fusion)	→ High accuracy with the support of different sensor	→ Poor recognition performance for some activities (e.g., "eating")
V2.5 Smartphone, Wearable and Video-based AR (multisensor fusion)	→ Reduce activities (8) for high accuracy & detailed eating motions	

Confidence level: Medium

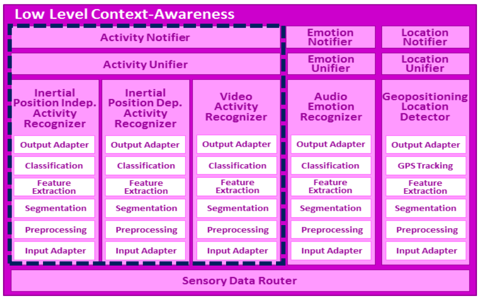


Challenges and solutions (MMv2.5-ICL)

1. Human activity recognition (AR)

	Benefit	Limitation
V1. Smartphone-based AR	→ High accuracy	→ Only 5 activities
V1.5 Smartphone and Wearable-based AR	→ More activities (up to 15)	→ Poor recognition performance for some activities (e.g., “eating”)
V2 Smartphone, Wearable and Video-based AR (multisensor fusion)	→ High accuracy with the support of different sensor	→ Poor recognition performance for some activities (e.g., “eating”)
V2.5 Smartphone, Wearable and Video-based AR (multisensor fusion)	→ Reduce activities (8) for high accuracy & detailed eating motions	

Confidence level: Medium

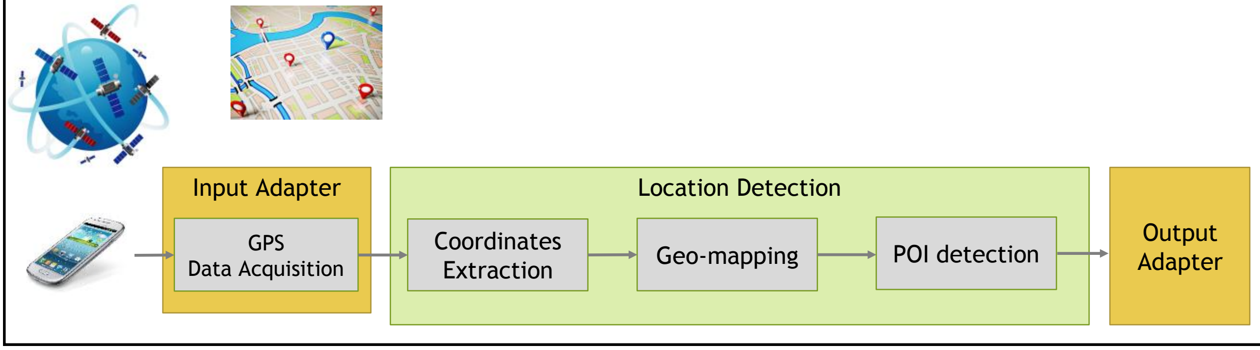


Challenges and solutions (MMv2.5-ICL)

2. Location detection (LD)

	Benefit	Limitation
V1. (Not implemented)	→ ( )	→ ( )
V1.5 (Not implemented)	→ ( )	→ ( )
V2 Outdoor location detection	→ Identification of the user geoposition for behavior tracking	

Confidence level: High

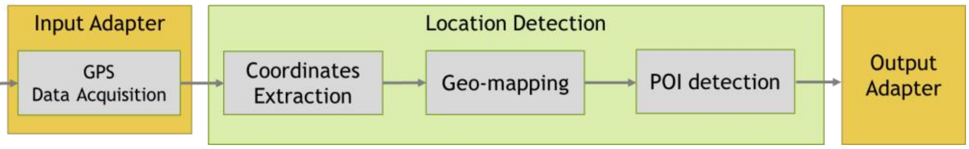


Challenges and solutions (MMv2.5-ICL)

2. Location detection (LD)

	Benefit	Limitation
V1. (Not implemented)	→ ( )	→ ( )
V1.5 (Not implemented)	→ ( )	→ ( )
V2 Outdoor location detection	→ Identification of the user geoposition for behavior tracking	→ Location can only be set at the stage of user registration
V2.5 Outdoor location detection	→ New location can be added anytime	

Confidence level: High

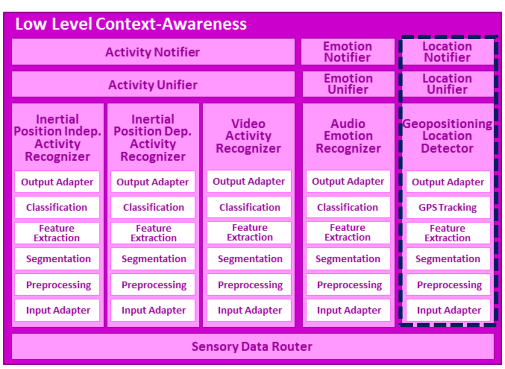


Challenges and solutions (MMv2.5-ICL)

2. Location detection (LD)

	Benefit	Limitation
V1. (Not implemented)	→ ( )	→ ( )
V1.5 (Not implemented)	→ ( )	→ ( )
V2 Outdoor location detection	→ Identification of the user geoposition for behavior tracking	

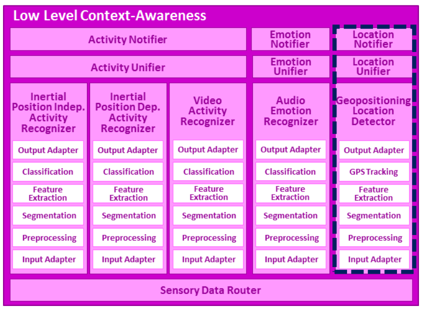
Confidence level: High



Challenges and solutions (MMv2.5-ICL)

2. Location detection (LD)

	Benefit	Limitation
V1. (Not implemented)	→ ( )	→ ( )
V1.5 (Not implemented)	→ ( )	→ ( )
V2 Outdoor location detection	→ Identification of the user geoposition for behavior tracking	→ Location can only be set at the stage of user registration
V2.5 Outdoor location detection	→ New location can be added anytime	



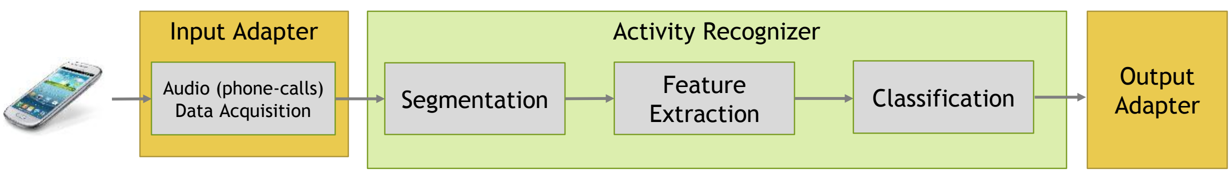
Confidence level: High

Challenges and solutions (MMv2.5-ICL)

3. Emotion Recognition (ER)

	Benefit	Limitation
V1. (Not implemented)	→ ( )	→ ( )
V1.5 (Not implemented)	→ ( )	→ ( )
V2 Audio-based ER	→ Identification of the user emotional states for a holistic description of the people behavior	

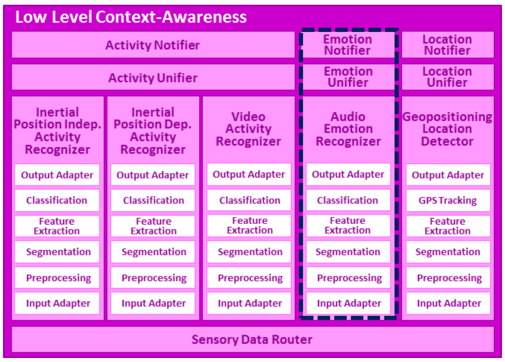
Confidence level: Low-Medium



Challenges and solutions (MMv2.5-ICL)

3. Emotion Recognition (ER)

	Benefit	Limitation
V1. (Not implemented)	→ ( )	→ ( )
V1.5 (Not implemented)	→ ( )	→ ( )
V2 Audio-based ER	→ Identification of the user emotional states for a holistic description of the people behavior	

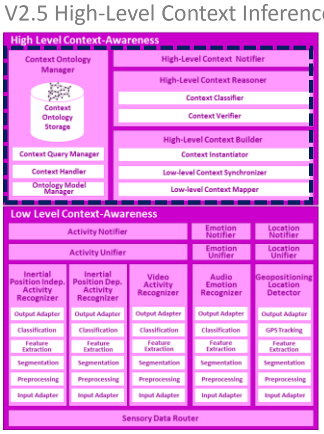


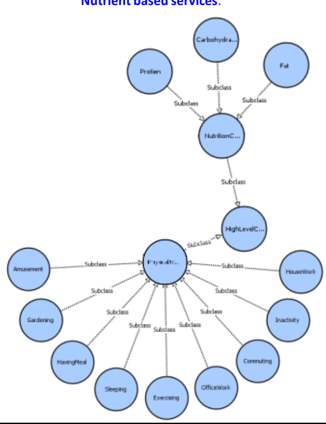
Confidence level: Low-Medium

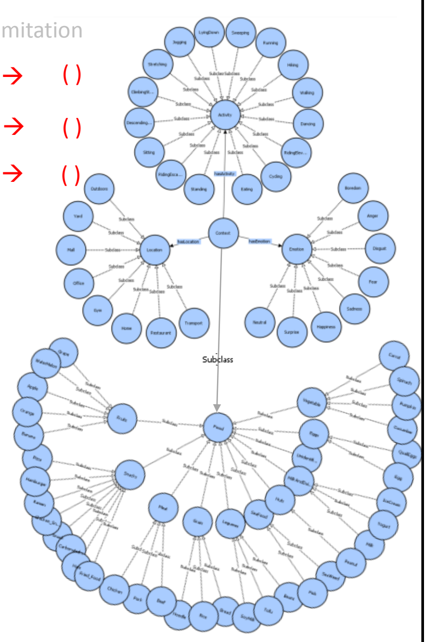
Challenges and solutions (MMv2.5-ICL)

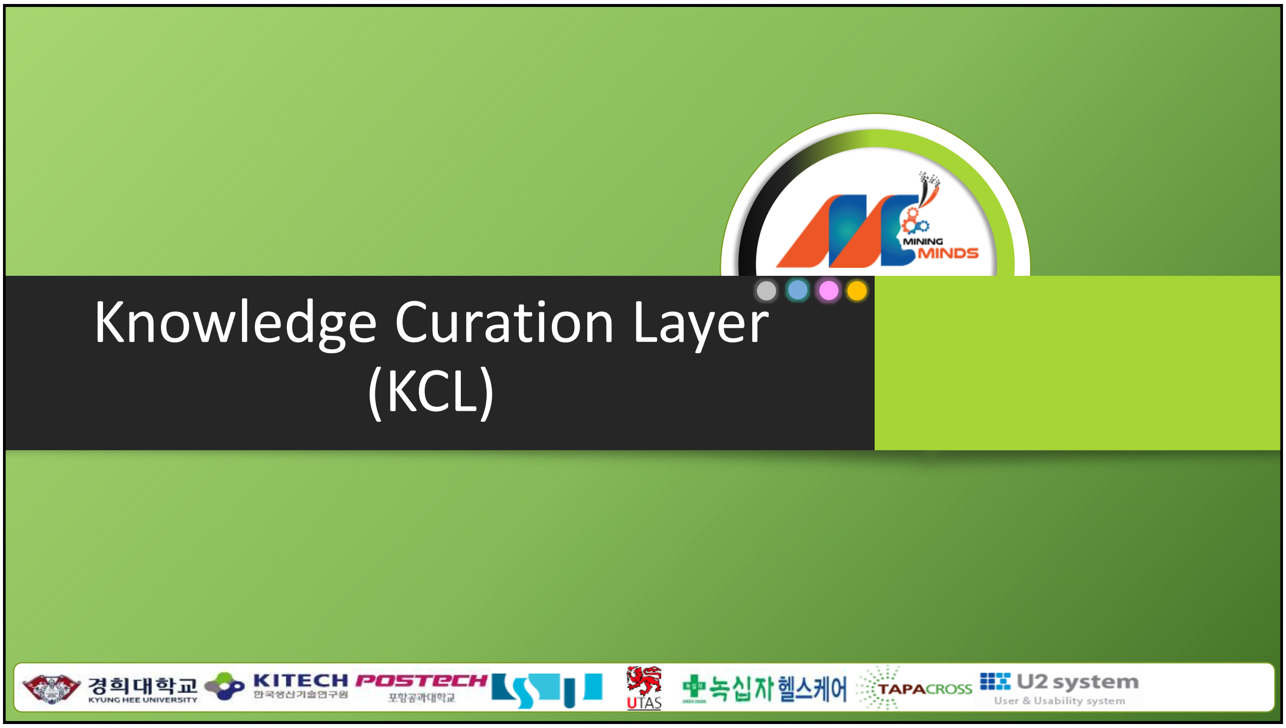
4. High-Level Context Awareness (HLCA)

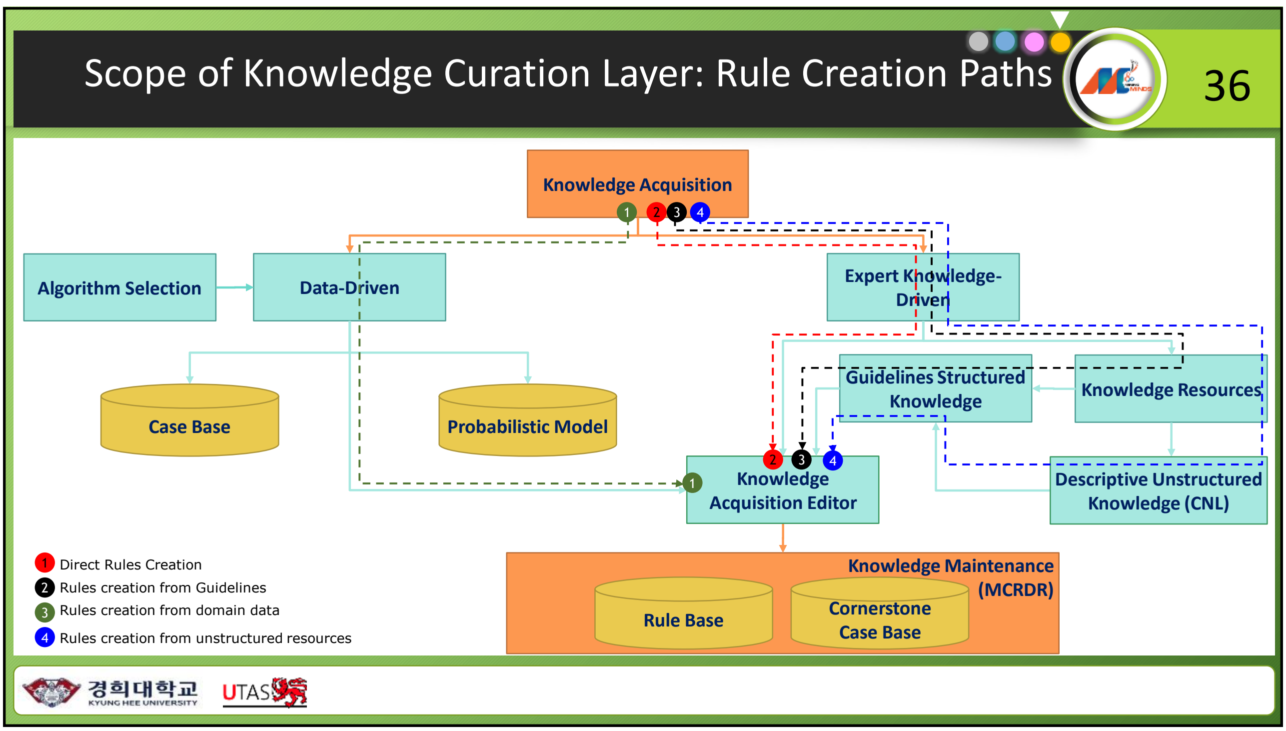
	Benefit	Limitation
V1. (Not considered)	→ ( )	→ ( )
V1.5 (Not considered)	→ ( )	→ ( )
V2 High-Level Context Inference	→ Enrichment of the user behavior description. **Physical Activity based services**.	→ ( )
V2.5 High-Level Context Inference	→ Enrichment of the user behavior description. **Nutrient based services**.	

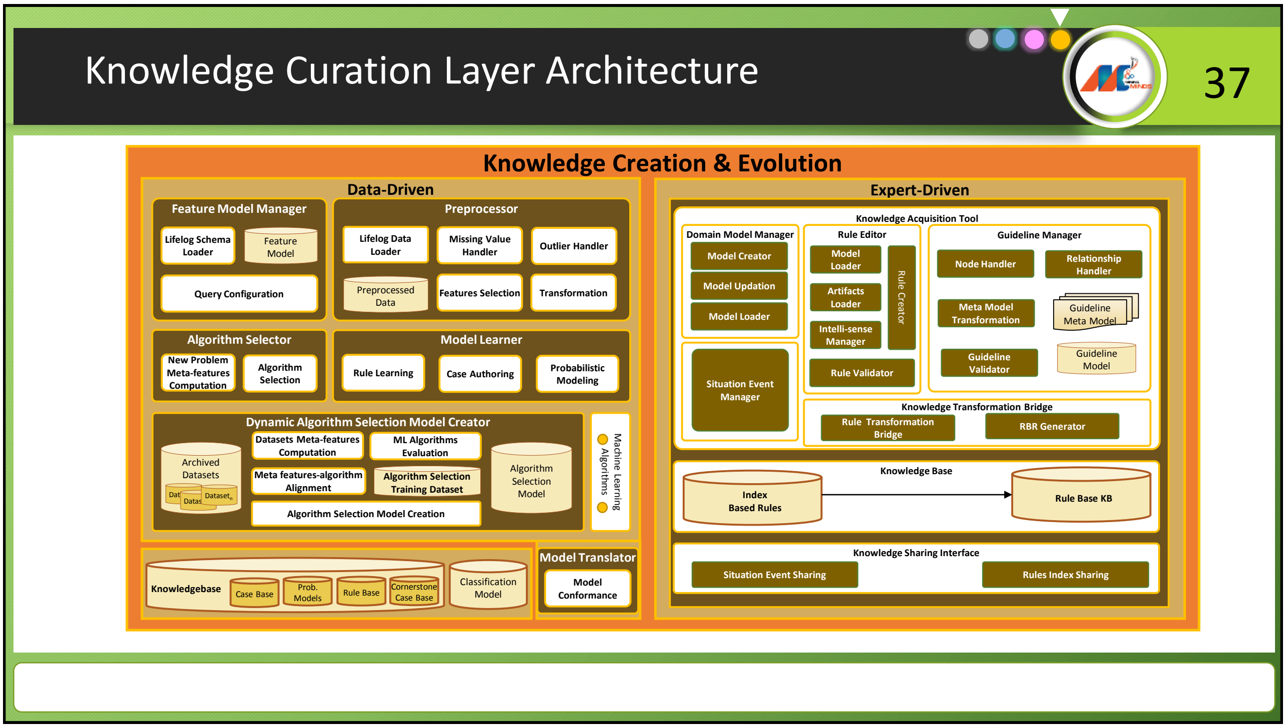







MINING MINDS
Knowledge Curation Layer
(KCL)
경희대학교
KYUNG HEE UNIVERSITY
KITECH
POSTECH
UTAS
녹십자헬스케어
TAPACROSS
U2 system
User & Usability system

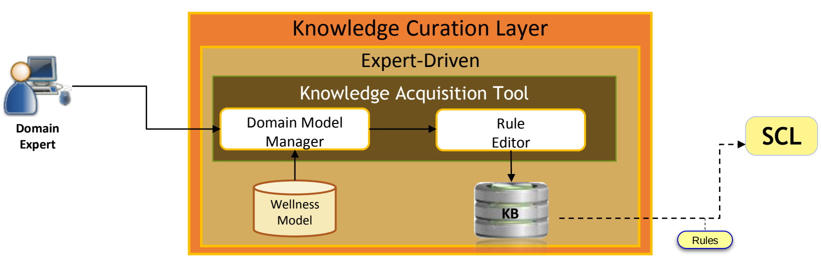

Scope of Knowledge Curation Layer: Rule Creation Paths
36
Knowledge Acquisition
Algorithm Selection
Data-Driven
Expert Knowledge-Driven
Case Base
Probabilistic Model
Guidelines Structured Knowledge
Knowledge Resources
Knowledge Acquisition Editor
Descriptive Unstructured Knowledge (CNL)
Knowledge Maintenance (MCRDR)
Rule Base
Cornerstone Case Base
1 Direct Rules Creation
2 Rules creation from Guidelines
3 Rules creation from domain data
4 Rules creation from unstructured resources
경희대학교
KYUNG HEE UNIVERSITY
UTAS



Challenges and solutions (MMv2.5-KCL)

1. Expert Driven Knowledge Acquisition

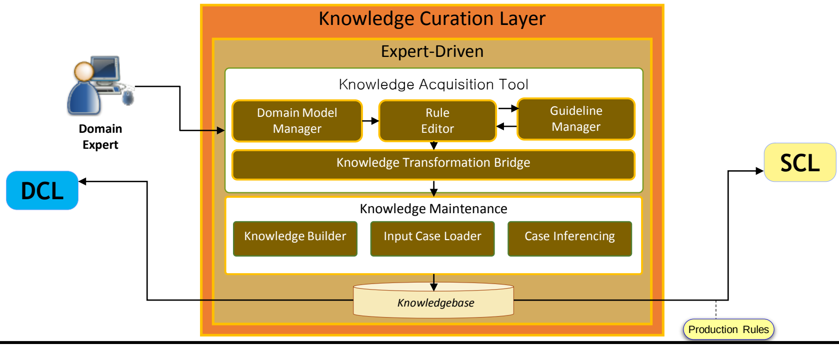
MM Version	Benefits	Limitations
V1. (Not considered)	→ ( )	→ ( )
V1.5: Expert base knowledge (Path-I)	→ Rule base KA	→ • No Validation • No SCL direct Integration



Challenges and solutions (MMv2.5-KCL)

1. Expert Driven Knowledge Acquisition

MM Version	Benefits	Limitations	Challenges
V1. (Not considered)	→ ( )	→ ( )	→ ( )
V1.5: Expert base knowledge (Path-I)	→ Rule base KA	→ • No Validation • No SCL direct Integration	→ ( )
V2.0: Rule Editor(Path-I)	→ • Facilitate expert for knowledge creation	→ ( )	→ • Creation without validation

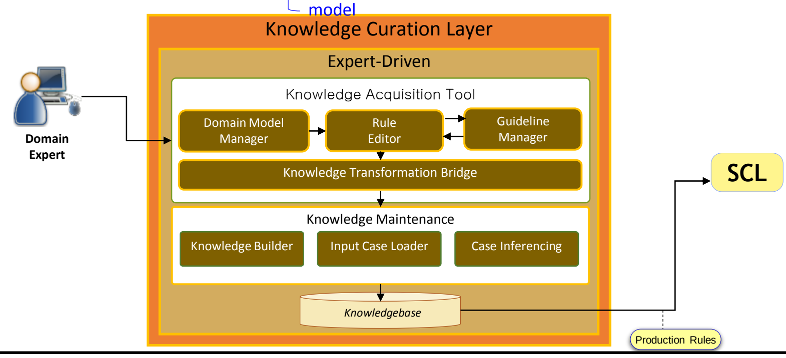


Confidence level: Low-Medium

Challenges and solutions (MMv2.5-KCL)

1. Expert Driven Knowledge Acquisition

MM Version	Benefits	Limitations	Challenges
V1. (Not considered)	→ ( )	→ ( )	→ ( )
V1.5: Expert base knowledge (Path-I)	→ Rule base KA	→ • No Validation • No SCL direct Integration	→ ( )
V2.0: Rule Editor(Path-I)	→ • Facilitate expert for knowledge creation	→ ( )	→ • Creation without validation
V2.5: Rule Editor with validation (Path-I) Domain Model Manager	→ • Facilitate expert for knowledge creation and validation • Facilitate expert to create domain model	→ ( )	→ • Domain model Creation without validation

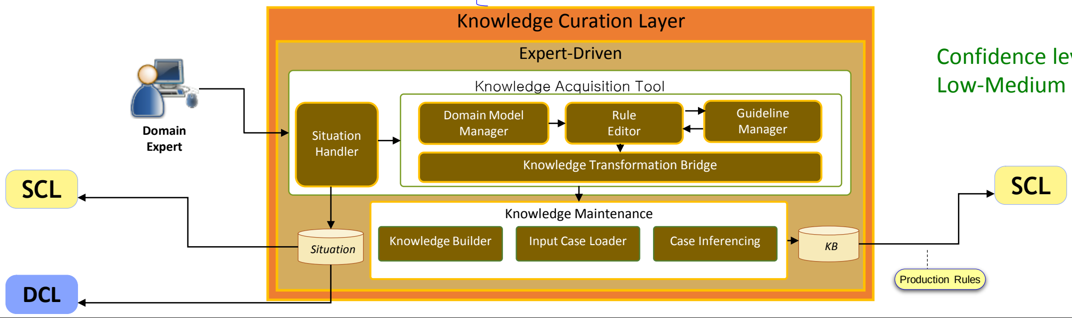


Confidence level: Low-Medium

Challenges and solutions (MMv2.5-KCL)

1. Expert Driven Knowledge Acquisition

MM Version	Benefits	Limitations	Challenges
V1. (Not considered)	→ ( )	→ ( )	→ ( )
V1.5: Expert base knowledge (Path-I)	→ Rule base KA	→ • No Validation • No SCL direct Integration	→ ( )
V2.0: Situation Handling	→ • Situation Definition • Situation based rules creation	→ ( )	→ • Common format definition
V2.5: Situation sharing with DCL	→ • DCL will monitor situation in lifelog	→ ( )	→ • Common format definition

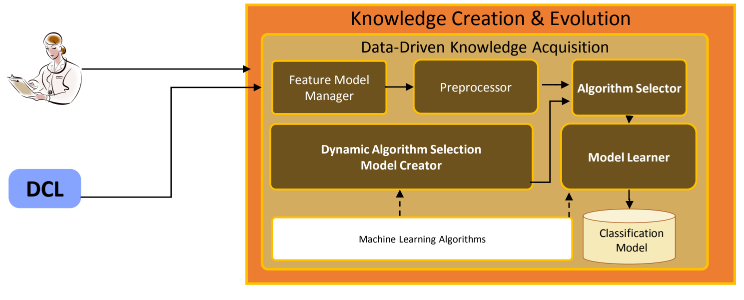


Confidence level: Low-Medium

Challenges and solutions (MMv2.5-KCL)

2. Data Driven Knowledge Acquisition

MM Version	Benefits	Limitations	Challenges
V1: (Not considered)	→ ( )	→ ( )	→ ( )
V1.5: (Not considered)	→ ( )	→ ( )	→ ( )
V2.0: Creating classification model using life log data (Path-III)	→ • DTs from lifelog data	→ ( )	→ • Overall accuracy depends on available data • Big data handling

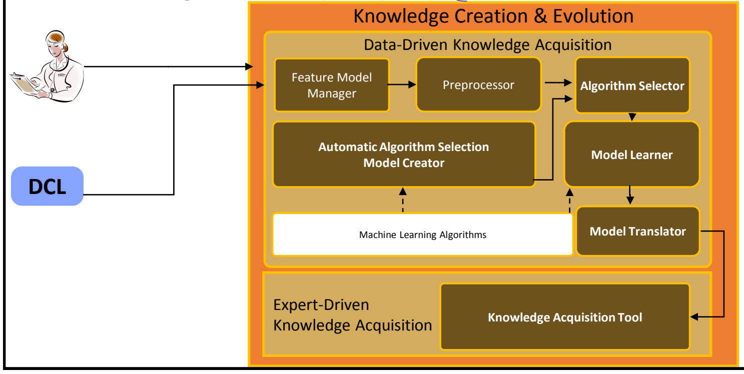


Confidence level: Low-Medium

Challenges and solutions (MMv2.5-KCL)

2. Data Driven Knowledge Acquisition

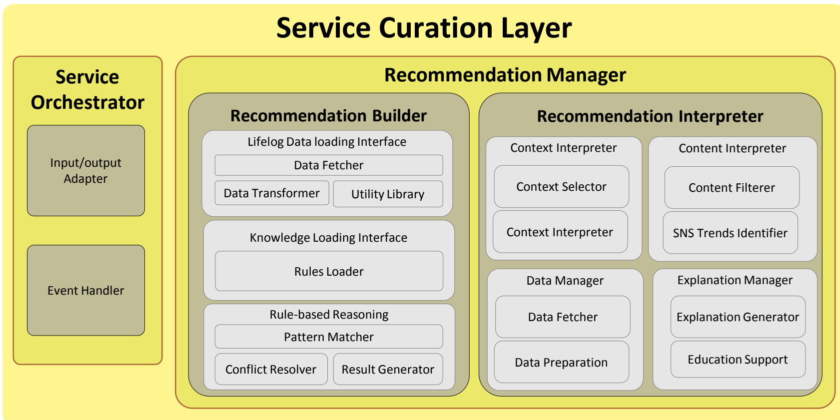
MM Version	Benefits	Limitations	Challenges
V1: (Not considered)	→ ( )	→ ( )	→ ( )
V1.5: (Not considered)	→ ( )	→ ( )	→ ( )
V2.0: Creating classification model using life log data (Path-III)	→ • DTs from lifelog data	→ ( )	→ • Overall accuracy depends on available data • Big data handling
V2.5: Creating classification model using automatic algorithm selection (Path-III)	→ • Accurate Algorithm Selection • DTs conformance with domain expert	→ ( )	→ • Algorithm selection accuracy • Rule extraction from model



Confidence level:
Low-Medium

Service Curation Layer (SCL)

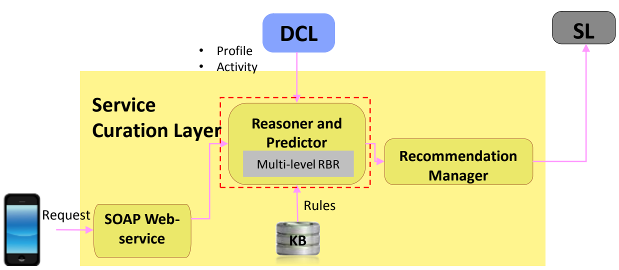
MMv2.5-SCL Architecture



Challenges and solutions (MMv2.5-SCL)

1. Reasoning

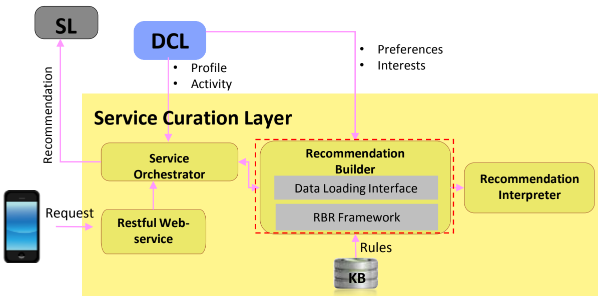
V1. Rule-based Reasoning (Reasoner and Predictor) → Benefit: 1. Granular personalization → Limitation:
1. Less flexible
2. Service and Data Management
3. No Automatic Rules Retrieval
4. Handcrafted Rules (not scalable)



Challenges and solutions (MMv2.5-SCL)

1. Reasoning

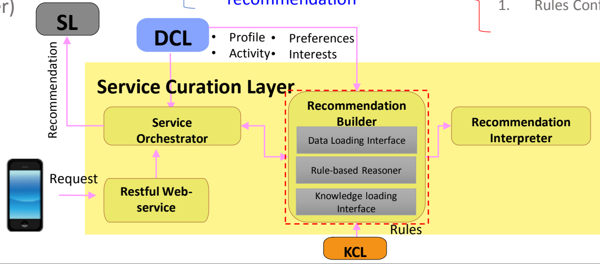
		Benefit		Limitation
V1. Rule-based Reasoning (Reasoner and Predictor)	→	1. Granular personalization	→	1. Less flexible 2. Service and Data Management 3. Handcrafted Rules (not scalable) 4. No Automatic Rules Retrieval
V1.5. Rule-based Reasoning (Recommendation Builder)	→	1. Flexibility 2. Improved data & service communication	→	1. Partial Automatic Rules (authoring tool) 2. No Unified Interface with KCL 3. Low precision



Challenges and solutions (MMv2.5-SCL)

1. Reasoning

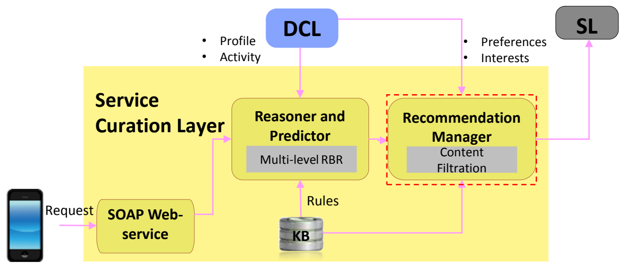
		Benefit		Limitation
V1. Rule-based Reasoning (Reasoner and Predictor)	→	1. Granular personalization	→	1. Less flexible 2. Service and Data Management 3. Handcrafted Rules (not scalable) 4. No Automatic Rules Retrieval
V1.5. Rule-based Reasoning (Recommendation Builder)	→	1. Flexibility 2. Improved data & service communication	→	1. Partial Automatic Rules (authoring tool) 2. No Unified Interface with KCL
V2. RBR (Recommendation Builder)	→	1. Preciseness 2. Scalability 3. Unified knowledge interface	→	Challenges: 1. Interfacing with KCL 2. Design & development of extensible reasoning framework
V2.5 RBR forward chaining & conflict resolution (Recommendation Builder)	→	1. Accuracy 2. Non-conflicting recommendation	→	Challenges: 1. Rules Conflicts resolution



Challenges and solutions (MMv2.5-SCL)

2. Interpretation and Explanation

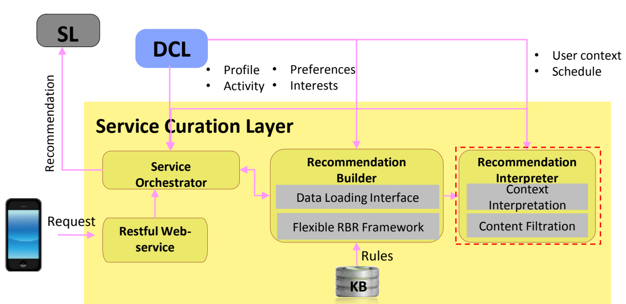
	Benefit	Limitation
V1. Content Filtration (Recommendation Manager) →	• Preference-based Personalization →	1. Lack of understandability 2. No context interpretation 3. No explanation



Challenges and solutions (MMv2.5-SCL)

2. Interpretation and Explanation

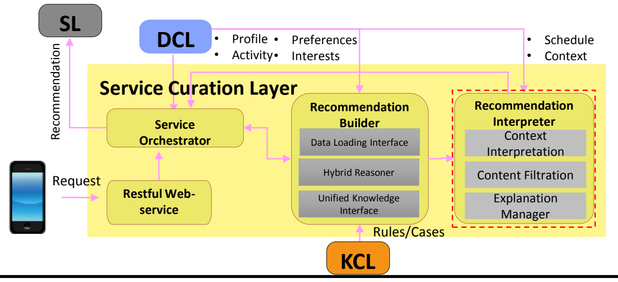
	Benefit	Limitation
V1. Content Filtration (Recommendation Manager) →	1. Preference-based Personalization →	1. Lack of understandability 2. No context interpretation 3. No explanation
V1.5. Context Interpretation (Recommendation Interpreter) →	1. Understandable 2. Contextually (time-based) interpreted reco →	1. Limited and static explanation



Challenges and solutions (MMv2.5-SCL)

2. Interpretation and Explanation

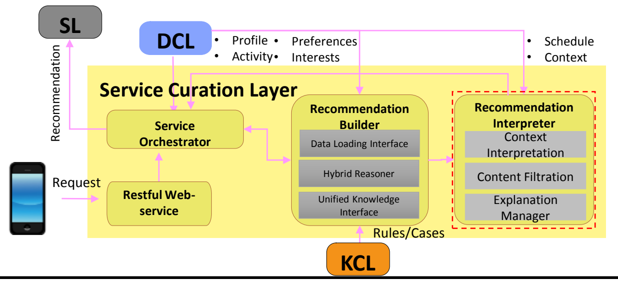
		Benefit		Limitation
V1.0 Content Filtration (Recommendation Manager)	→	1. Preference-based Personalization	→	1. Lack of understandability 2. No context interpretation 3. No explanation
V1.5. Context Interpretation (Recommendation Interpreter)	→	1. Understandable 2. Contextually (time-based) interpreted rec	→	1. Limited explanation 2. Static explanation
V2.0 Explanation	→	1. Situation-aware Explanation 2. Education Support	→	Challenges: 1. Cross-Context Interpretations 2. Dynamic Explanations



Challenges and solutions (MMv2.5-SCL)

2. Interpretation and Explanation

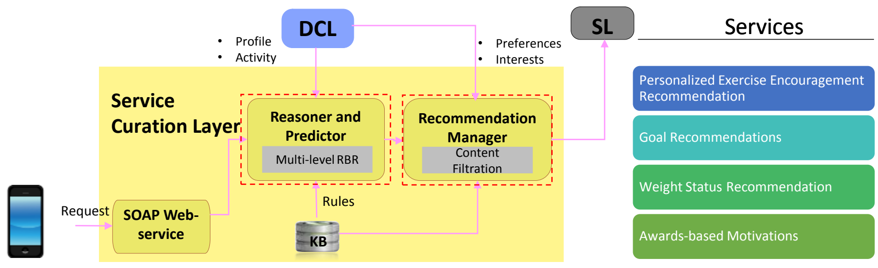
		Benefit		Limitation
V1.0 Content Filtration (Recommendation Manager)	→	1. Preference-based Personalization	→	1. Lack of understandability 2. No context interpretation 3. No explanation
V1.5. Context Interpretation (Recommendation Interpreter)	→	1. Understandable 2. Contextually (time-based) interpreted rec	→	1. Limited explanation 2. Static explanation
V2.0 Explanation V2.5 Social Trends	→	1. Situation-aware Explanation 2. Education Support 3. Social Trends	→	Challenges: 1. Cross-Context Interpretations 2. Dynamic Explanations



Challenges and solutions (MMv2.5-SCL)

3. Services Scope and Types

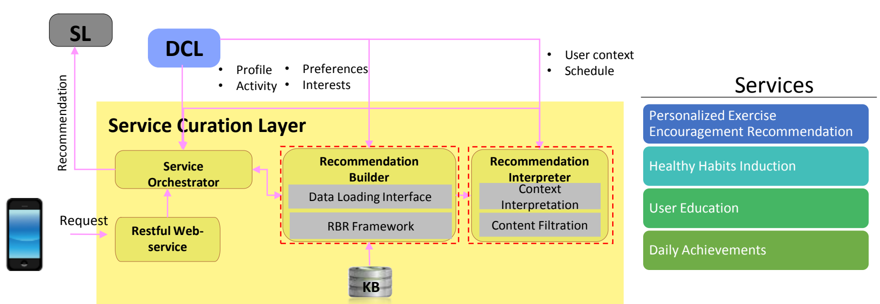
- V1. Physical Activity Recommendations (Calories Based) → Benefit: 1. User encouragement for phy. activity rec. using goal and award-based services → Limitation: 1. Less practical in terms of user understandability



Challenges and solutions (MMv2.5-SCL)

3. Services Scope and Types

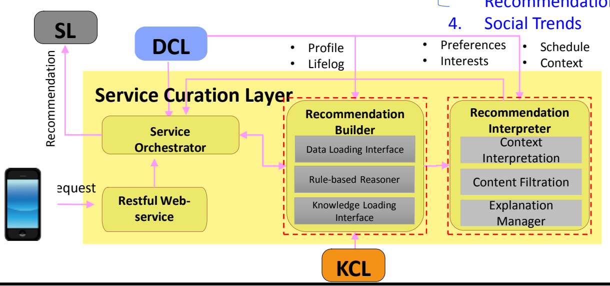
- V1. Physical Activity Recommendations (Calories Based) → Benefit: 1. User encouragement for phy. activity rec. using goal and award-based services → Limitation: 1. Less practical in terms of user understandability
- V1.5 Physical Activity Recommendations and Educational Facts → Benefit: 1. Physical actv. rec 2. Avoiding abnormal sedentary behaviors → Limitation: 1. Static situation handling 2. Limited context-awareness



Challenges and solutions (MMv2.5-SCL)

3. Services Scope and Types

Version	Benefit	Limitation
V1. Physical Activity Recommendations (Calories Based)	1. User encouragement for phy. activity rec. using goal and award-based services	1. Less practical in terms of user understandability
V1.5 Physical Activity Recommendations and Educational Facts	1. Avoiding abnormal sedentary behaviors 2. User education	1. Static situation handling 2. Limited context-awareness
V2.0 Physical Activity Recommendations and Educational Facts V2.5 Nutrition Recommendations	1. Avoiding abnormal sedentary behaviors 2. User education 3. Food-integrated phy. act. Recommendations 4. Social Trends	Challenges: 1. Increased Service dependencies management 2. Integration of food and physical activity services



Services

- Personalized Exercise Encouragement Recommendation
- Healthy Habits Induction
- User Education
- Nutrition Services with Social Trends

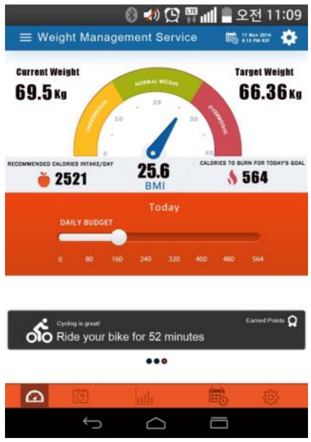
Supporting Layer (SL)

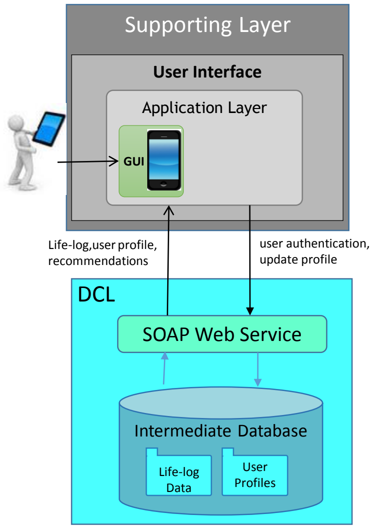
Challenges and solutions (MMv2.5-SL)

1. User Interface and User Experience

	Benefit		Limitation
V1. User Interfaces	→ Services User Interface	→	Static UI

- User interfaces for Calories based Weight Management Scenario
- Admin view for monitoring the activities of the users and system



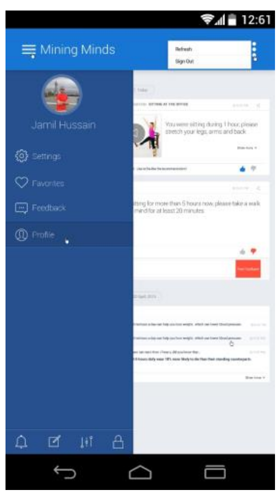


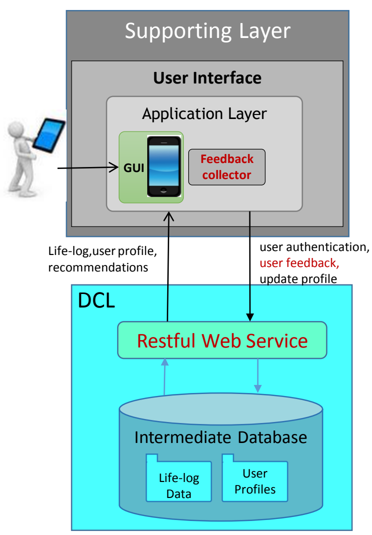
Challenges and solutions (MMv2.5-SL)

1. User Interface and User Experience

	Benefit		Limitation
V1. User Interfaces	→ Services User Interface	→	Static UI
V1.5 User Interfaces + Feedback	→ Services User Interface	→	Static UI

- User interfaces for educational and facts based Weight Management Scenario
- Expert view for monitoring user behavior and feedback provision



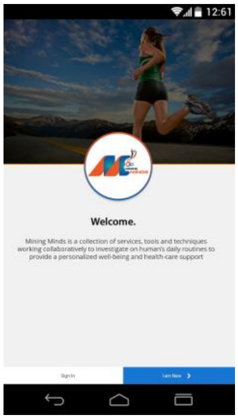


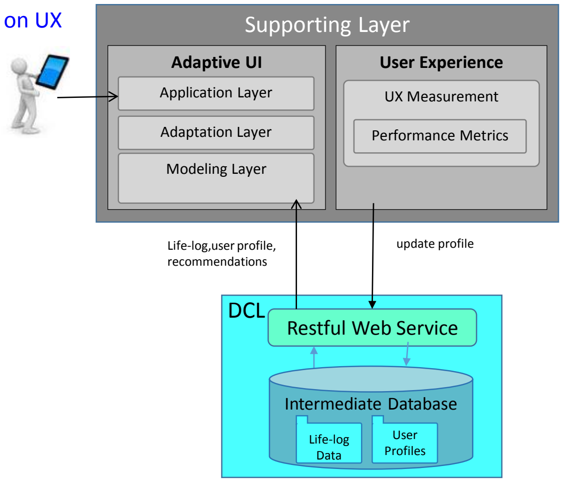
Challenges and solutions (MMv2.5-SL)

1. User Interface and User Experience

	Benefit	Limitation
V1. User Interfaces	→ Services User Interface	→ Static UI
V1.5 User Interfaces + **Feedback**	→ Services User Interface	→ Static UI
V2 Adaptive UI + User Experience	→ Adaptive UI based on UX	

- User interfaces for weight management scenario
- Adaptive User Interface based on user profile, device and



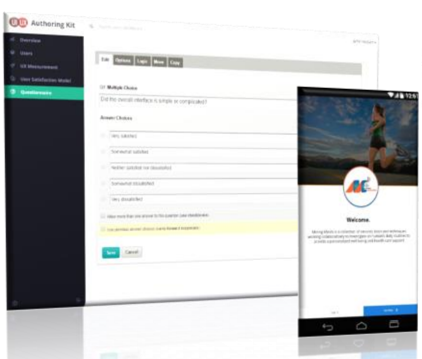


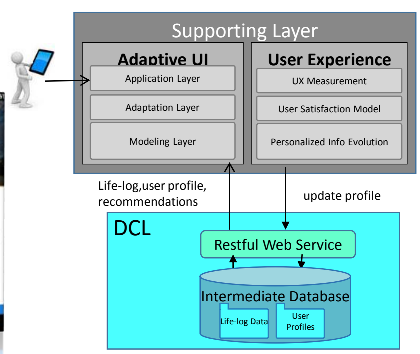
Challenges and solutions (MMv2.5-SL)

1. User Interface and User Experience

	Benefit	Limitation
V1. User Interfaces	→ Services User Interface	→ Static UI
V1.5 User Interfaces + **Feedback**	→ Services User Interface	→ Static UI
V2 Adaptive UI + User behavior analytics	→ Adaptive UI based on UX	→ User Experience Quantification
V2.5 Adaptive UI + User Experience	→ User Experience Quantifications	

- User experience quantification based on user behavior data along with self-reported data
- User feedback collection and its data analysis
- Evaluation of user experience

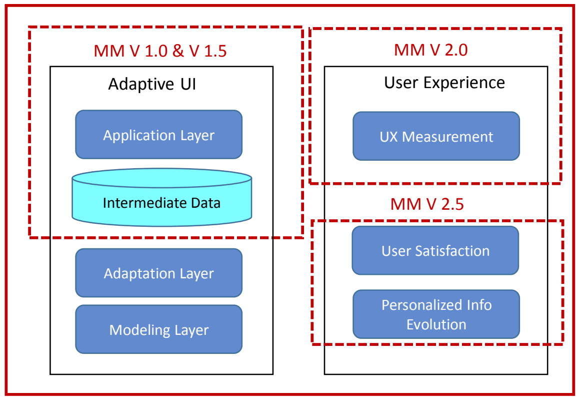


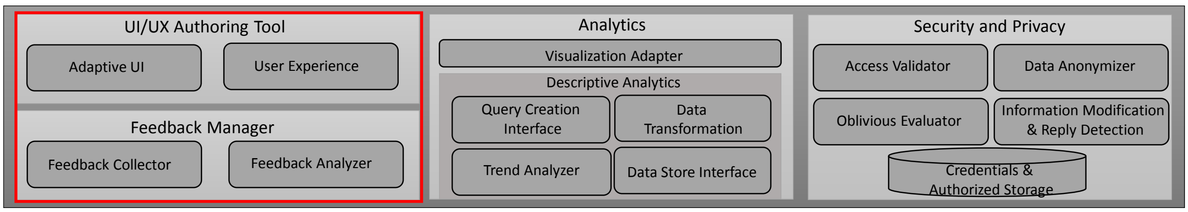


Challenges and solutions (MMv2.5-SL)

1. User Interface and User Experience

		Benefit		Limitation
V1. User Interfaces	→	Services User Interface	→	Static
V1.5 User Interfaces + **Feedback**	→	Services User Interface	→	Static
V2 Adaptive UI + User Experience	→	AdaptiveUI based on UX	→	User Experience Quantification
V2.5 Adaptive UI + User Experience	→	User Experience Quantifications		



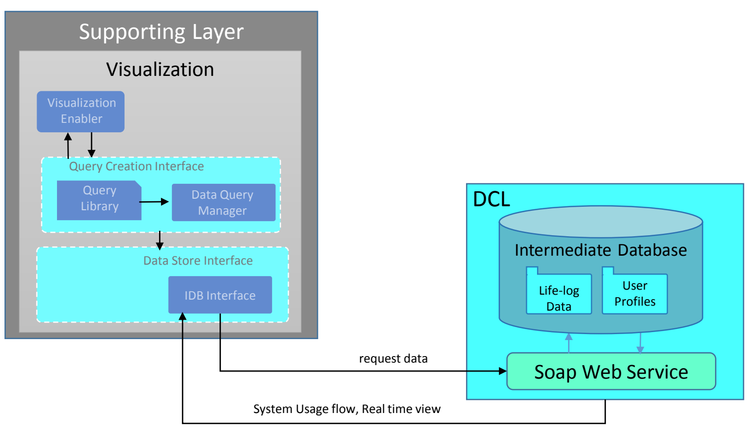


Challenges and solutions (MMv2.5-SL)

2. Analytics

	Benefit	Limitation
V1. Admin View	System usage visualization and data correlation between layers	Not intuitive and performance inefficient due to high volume in Intermediate DB

- Admin view for monitoring work flow and trigger points of the sensory data
- System Usage only from intermediate data base

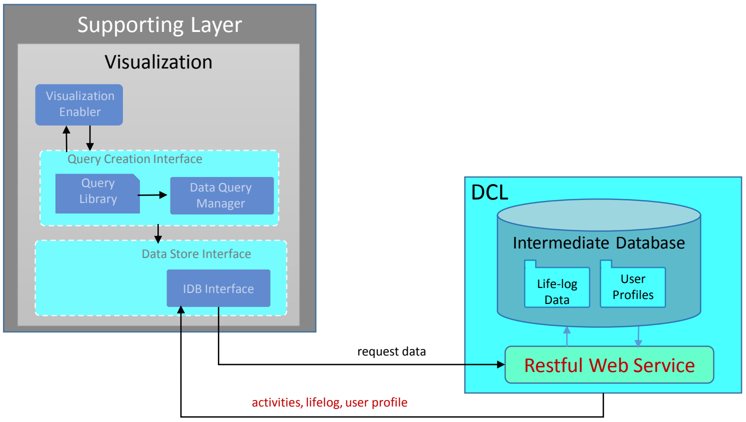


Challenges and solutions (MMv2.5-SL)

2. Analytics

	Benefit	Limitation
V1. Admin View	→ System usage visualization and data correlation between layers	→ Not intuitive and performance inefficient due to high volume in Intermediate DB
V1.5 Expert View	→ Users feedback, activities and system usage visualization	→ Limited features due to intermediate data schema and size

- Expert view for visualizing user activities and feedback for recommendations
- System Usage only from intermediate data base

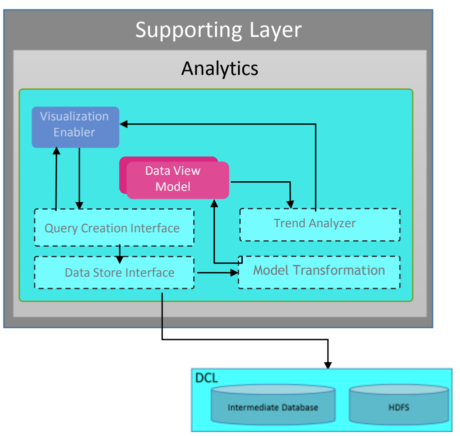


Challenges and solutions (MMv2.5-SL)

2. Analytics

	Benefit	Limitation
V1. Admin View	→ System usage visualization and data correlation between layers	→ Not intuitive and performance inefficient due to high volume in Intermediate DB
V1.5 Expert View	→ Users feedback, activities and system usage visualization	→ Limited features due to intermediate data schema and size
V2 Big data analytics	→ Users feedback, activities and facts analytics. Insights into big data from sensory data from system usage perspective.	

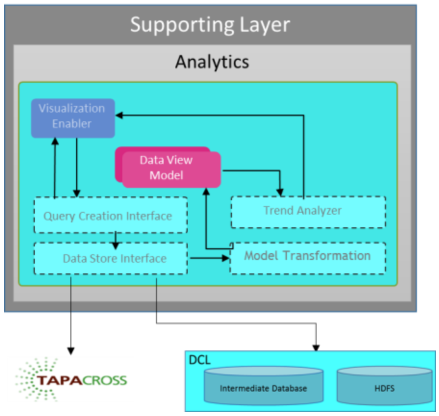
- Analytics based on statistics, clustering and association
- Insights from the big data repository and communication between both repositories
- Query library for the big data repository



Challenges and solutions (MMv2.5-SL)

2. Analytics

	Benefit	Limitation
V1. Admin View	System usage visualization and data correlation between layers	Not intuitive and performance inefficient due to high volume in Intermediate DB
V1.5 Expert View	Users feedback, activities and system usage visualization	Limited features due to intermediate data schema and size
V2 Big data analytics	Users feedback, activities and facts analytics. Insights into big data from sensory data from system usage perspective.	No SNS for experts and for users and analytics limited to activity data
V2.5 Big data/SNS analytics	SNS trends for food, Users nutrition trends, activities and facts analytics. SNS trends for users and experts.	

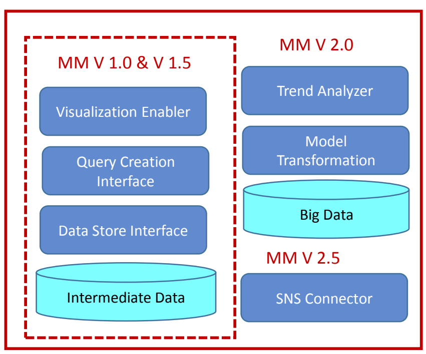


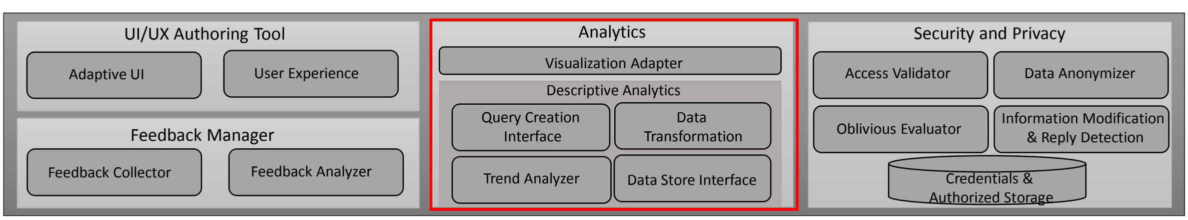
- Nutrition trends and analytics for expert panel and user through service curation layer
- SNS gateway for SNS trends
- Analytics based on statistics, clustering and association
- Insights from the big data repository and communication between both repositories
- Query library for the big data repository

Challenges and solutions (MMv2.5-SL)

2. Analytics

	Benefit	Limitation
V1. Admin View	System usage visualization and data correlation between layers	Not intuitive and performance inefficient due to high volume in Intermediate DB
V1.5 Expert View	Users feedback, activities and system usage visualization	Limited features due to intermediate data schema and size
V2 Big data analytics	Users feedback, activities and facts analytics. Insights into big data from sensory data from system usage perspective.	No SNS for experts and for users and analytics limited to activity data
V2.5 Big data analytics	SNS trends for food, Users nutrition trends, activities and facts analytics. SNS trends for users and experts.	

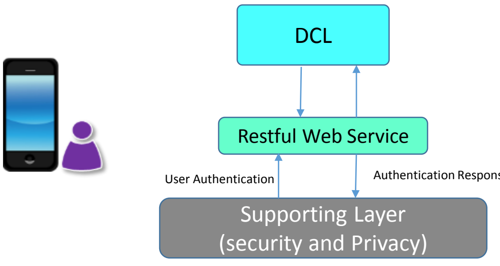




Challenges and solutions (MMv2.5-SL)

3. Security

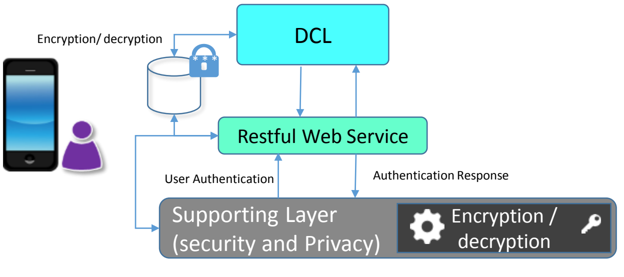
	Benefit	Limitation
V1.0 Expert View Authentication	→ User Authentication	→ Lacking encrypted data storage



Challenges and solutions (MMv2.5-SL)

3. Security

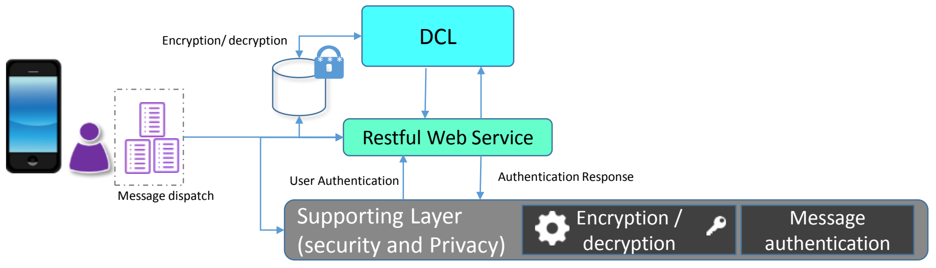
	Benefit	Limitation
V1.0 Expert View Authentication	User Authentication	Lacking encrypted data storage
V1.5 Secure data storage	→ Acquisition of data security through encryption	→ No secure communication between databases and mobile application



Challenges and solutions (MMv2.5-SL)

3. Security

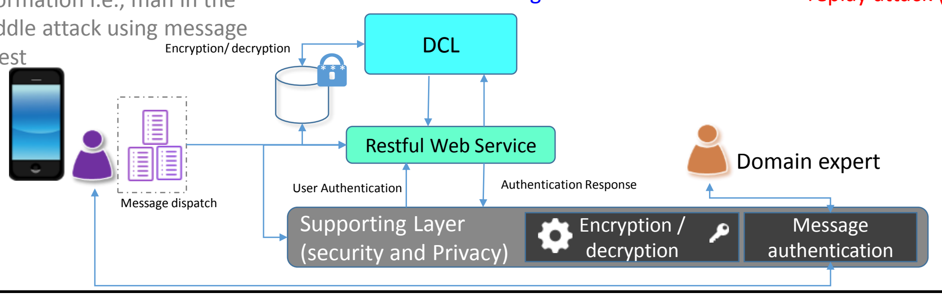
	Benefit	Limitation
V1.0 Expert View Authentication	User Authentication	Lacking encrypted data storage
V1.5 Secure data storage	Acquisition of data security through encryption	No secure communication between databases and mobile application
V2.0 Communication between user application and data repositories	→ Replay Detection implementation for secure communication.	→ Missing security for the modified replay attack (man in the middle)



Challenges and solutions (MMv2.5-SL)

3. Security

	Benefit	Limitation
V1.0 Expert View Authentication	User Authentication	Lacking encrypted data storage
V1.5 Secure data storage	Acquisition of data security through encryption	No secure communication between databases and mobile application
V2.0 Communication between user application and data repositories	Replay Detection implementation for secure communication.	Missing security for the modified replay attack (man in the middle)
V2.5 replay attack with modified information i.e., man in the middle attack using message digest	→ Detection of modified messages Resilience against the inference attack	→ Missing security for the modified replay attack (man in the middle)



Section 4

Mining Minds Version 2.5
Design and Development Methodology

Section 4-A

Requirements Specifications

Introduction

MMv2.5 is the 4rth significant integration for Mining Minds Platform, constituting upon the duration of 6 months, i.e., (Dec 2015 – May 2016). MMv2.5 utilizes the analysis, design and implementation efforts of previous versions, i.e., MMv1.0, MMv1.5, MMv2.0, and MMv2.5, for its benefit and builds on the top of already identified layered abstractions and primary components. The overall architecture of MMv2.5 is similar to previous iterations; however, components have been added to incorporate newer requirements.

At Data level, Data Curation Layer of MMv2.5 (DCL 2.5), is incorporating not only activity data from smartphone and smartwatch but also acquiring video stream from a camera, making the data input more heterogeneous per user. Furthermore, to support nutrition based data, DCL 2.5 also incorporates the acquisition of images of meals with tags. This data acquisition is purely in real-time with asynchronous, non-blocking communication from the data source to the DCL 2.5. Moreover, acquired data is monitored in real-time for user-based anomaly detection; thus contributing to push-based services of MMv2.5. DCL 2.5 is also providing the read access to sensory data persisted in big data storage for data-driven knowledge acquisition, descriptive analytics, and visualization.

At Information level, Information Curation Layer of MMV2 (ICL 2), is incorporating new low-level context awareness mechanisms for the identification of the user location, emotion apart from their physical activities. Moreover, this version also includes high-level context awareness for a reliable and comprehensive determination of the user's context. The communication between DCL 2 and ICL 2 has also been updated in MMV2 as a buffer is introduced between DCL 2 and ICL 2 for incoming sensory data, such that ICL 2 is not stressed by strong influx of data from DCL 2. Furthermore, ICL 2 only communicates with DCL 2 in response when the context of the user changes, reducing the communication overhead.

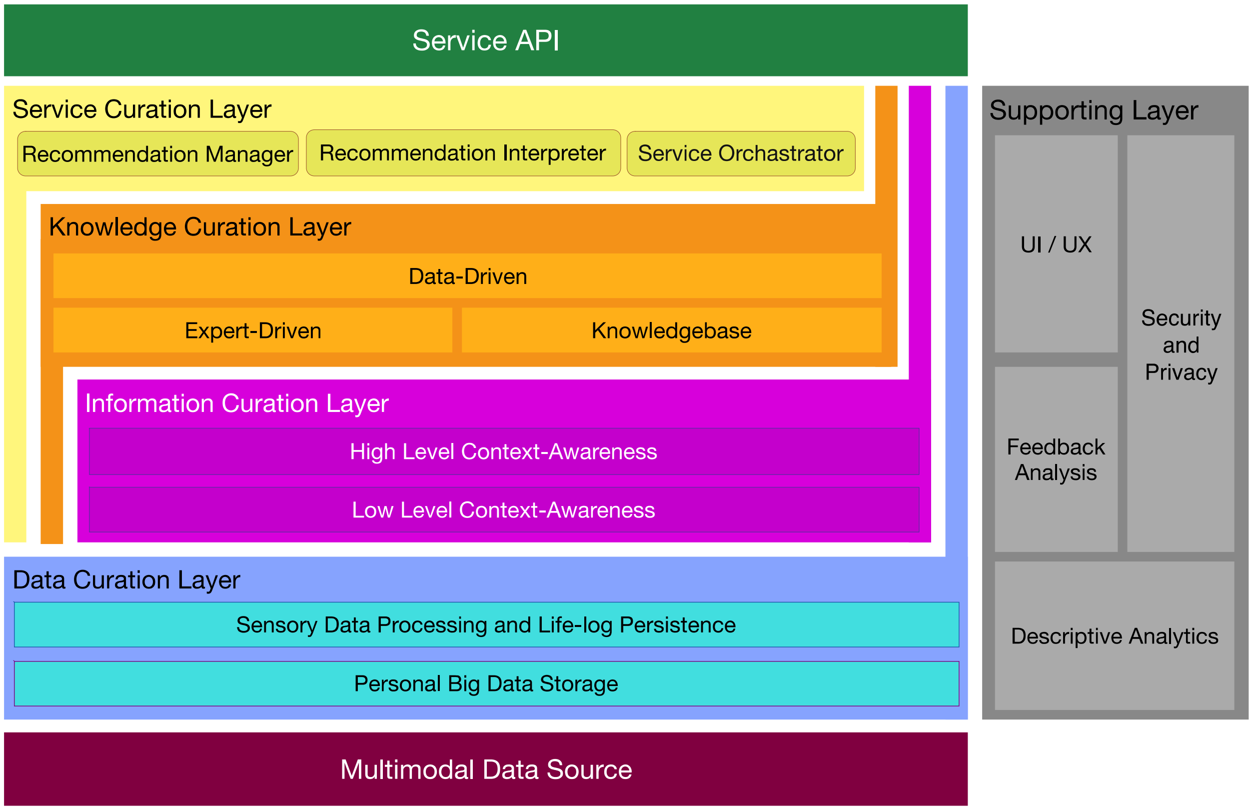
Knowledge curation, introduced in MMv1.5 is evolved in MMv2.5 as Knowledge Curation Layer (KCL 2.5). For data driven knowledge acquisition, this layer now incorporates features selection, data preprocessing, and creation of classification model directly from the data persisted as life-log in the big data storage. After model creation, it extracts the rules from model and shares them with domain expert to conform each extracted rule. For expert driven knowledge acquisition, KCL 2.5 provides production rules for service curation components and derive situations to be hosted at DCL 2.5 and serviced at high-level service curation. Additionally, this version implements the knowledge base validation for duplication and conflict of new rules with existing rules. Furthermore, KCL 2.5 facilitates the experts to create and maintain the wellness model using a user-friendly interface.

At service level, Service Curation Layer (SCL 2.5) is dealing with service request handling, recommendation generation, recommendation interpretations, and service response delivering. It receives service requests from service requester directly or through a trigger based on the events identified in life-log of a user, SCL 2.5 builds the recommendations based on user profile, life-log data, and production rules. Based on the context, user characteristics, and environment variables, the recommendation are interpreted, explained, extended with nutrition related social trends and are delivered to the Supporting Layer i.e. SL 2.5 in order to serve the service requester.

Supporting layer (SL 2.5) deals with providing services to every other layer of MMv2.5. It is responsible to provide personalized recommendations, trend analytics through adaptive user interface with added services of privacy and security. The user experience is quantified to give the user better experience and to evolve the mobile application. SL 2.5 provides security services for securing the communication among different layers in MMv2.5. SL 2.5 handles the recommendations from SCL 2.5 and stores the feedback of the user in DCL 2.5. Also, for analysis of the data stored as big data, SL 2.5 provides services of analytics by communicating with DCL 2.5. The timeline of all the individual users can be seen with nutrition and physical activity trends in the expert view. Different analytics and user behavior pattern is highlighted. The trends and analytics are shown for grouped users with similar traits and also according to the expert query.

This document provides requirement specifications for MMv2.5 with high-level use cases, sequence, and collaboration diagrams for the implemented platform.

Abstract Architecture



Functional Requirements (FR)

FR ID#	Description
MM-FR-01	The platform shall read the raw sensory data of the user from his/her data source
MM-FR-02	The platform shall provide permanent persistence to the user generated raw sensory data
MM-FR-03	The platform shall provide raw sensory data for context determination of the user
MM-FR-04	The platform shall maintain user profile data
MM-FR-05	The platform shall maintain user timeline as a life-log of daily behaviors
MM-FR-06	The platform shall provide read, write, delete, and update access to the subscribers of life-log data
MM-FR-07	The platform shall provide read access to the subscribers of raw sensory data
MM-FR-08	The platform shall monitor the life-log of a user for notify-able situations
MM-FR-09	The platform shall persist user feedback regarding generated recommendations and identified context
MM-FR-10	The platform shall provide each low-level context recognizer with the appropriate raw sensory data for recognition
MM-FR-11	The platform shall identify the user's low-level context
MM-FR-12	The platform shall identify the user's high-level context
MM-FR-13	The platform shall provide low-level context information for the generation of the life-log
MM-FR-14	The platform shall provide high-level context information for the generation of the life-log
MM-FR-15	The data-driven knowledge acquisition shall know schema detail of life-log and user profile data in order to load the data and extract feature model
MM-FR-16	The data-driven knowledge acquisition shall load all related life log and user profile data according to feature model. The loaded life log and user profile data will be used for classification model

	creation
MM-FR-17	The expert-driven knowledge acquisition shall share the production rules for enabling recommendation services
MM-FR-18	The expert-driven environment shall create “Situations” and share its configuration and its associated rules for monitoring and handling it
MM-FR-19	The platform shall receive the service request from user application, third party application, or mining mind platform generated events
MM-FR-20	The platform shall retrieve data from intermediate database (user profile, life-log, and environmental variables)
MM-FR-21	The platform shall retrieve production knowledge from knowledge base
MM-FR-23	The platform shall deliver the results to the service requester and to corresponding layer of mining mind for persistence
MM-FR-24	The platform shall read and display the recommendations generated according to user capabilities, context of use, and device characteristics
MM-FR-25	The platform shall retrieve the user profile information, context of use and device information for adaptation of the user interface
MM-FR-26	The platform shall collect the user data such as user feedback and user observational data for the enhancement of user interface
MM-FR-27	The platform shall utilize the user profile data, life-log and raw sensory data for analytics
MM-FR-28	The platform shall persist nutrition data for food intake monitoring of the user
MM-FR-29	The platform shall utilize the SNS trends according to personalized user profile and behavior.

Non-functional Requirements (NFR)

FR ID#	Description
MM-NFR-01	The platform shall read the raw sensory data of the user from his/her personal device in real-time with delay no later than 3 seconds
MM-NFR-02	The platform shall provide raw sensory data for low level activities determination in real-time with delay no later than 3 seconds
MM-NFR-03	The platform shall only read the raw sensory data from verified personal device
MM-NFR-04	The platform shall maintain the consistency, integrity, and reliability of raw sensory data in non-volatile storage
MM-NFR-05	Overall low-level context recognition accuracy of the platform shall be be greater than or equal to 80%
MM-NFR-06	Overall high-level context inference accuracy of the platform shall be greater than or equal to 73%
MM-NFR-07	The platform shall persist only verified and validated rules into knowledge base.
MM-NFR-08	The platform shall ensure consistency of distributed copies of knowledge base.
MM-NFR-09	The user application and the platform shall have high speed internet available.
MM-NFR-10	The platform shall provide user interface that is easy to use and intuitive
MM-NFR-11	The platform response time from big data shall be within 30 seconds

Specification Terms and Definition

Term	Definition
DCL	Data Curation Layer
ICL	Information Curation Layer
KCL	Knowledge Curation Layer
SCL	Service Curation Layer
SL	Supporting Layer
Life-log	Information associated to the user's life-events over time
Life-log schema	Life-log Schema represents the structure and associated semantics of user profile and life log data.
User profile	Information describing the user characteristics (i.e., age, gender, etc.)
Data source	User devices sending the required data, i.e., smartphone, video camera
Raw sensory data	Numerical values describing a physical phenomenon such as human body motion (e.g., acceleration) and photos depicting user meal in-take
Sensory metadata	Information that describes, at least, the source of data (e.g., video), the user to which the raw sensory data belongs (e.g., user ID) and the time in which the raw sensory data was registered (e.g., timestamp)
Sensory data	Raw sensory data plus sensory metadata
SNS data	Data from social networks (i.e., twitter, Facebook)
Environmental variables	Information representing non-human factors (i.e., weather, time, season etc.)
Low-level context	Information describing the user activities (e.g., sitting), user locations (e.g., restaurant) and user emotions (e.g., happy)
High-level context	Information describing the situation of the user (e.g., lunch)
Context	General concept to refer either to low-level context

	and/or high-level context
Situation	An abnormal status of a subject caused by unhealthy behaviors
Production rule	Production rule is ultimate and shareable rule which is used in reasoning to produce recommendation
Domain expert	Domain expert is an actor who will interact with system to create knowledge base
Rule verification and validation	Verification ensures that rule created is consistent with requirements and validation ensures that the rule created is correctly working on real data
Unresolved case	A new case for which the existing knowledge is insufficient to solve
Recommendation	An actionable statement provided to the subject for healthy habit induction
Fact	An informative statement provided to the subject for education
Interpreted recommendation	The recommendation processed on the basis of user context, user characteristics, and environmental variables.
Context of use	Environmental variables and low level context (location)
Device characteristics	Screen size, resolution, memory, and battery
UI adaptation	The changes in user interface
Observational data	User interaction data with the user interface
User experience	User perception, satisfaction about the user interfaces
Query library	A set of predefined queries
Nutrition Data	Nutrition-based variables (e.g. Calories, Proteins, Carbohydrates, Fats)

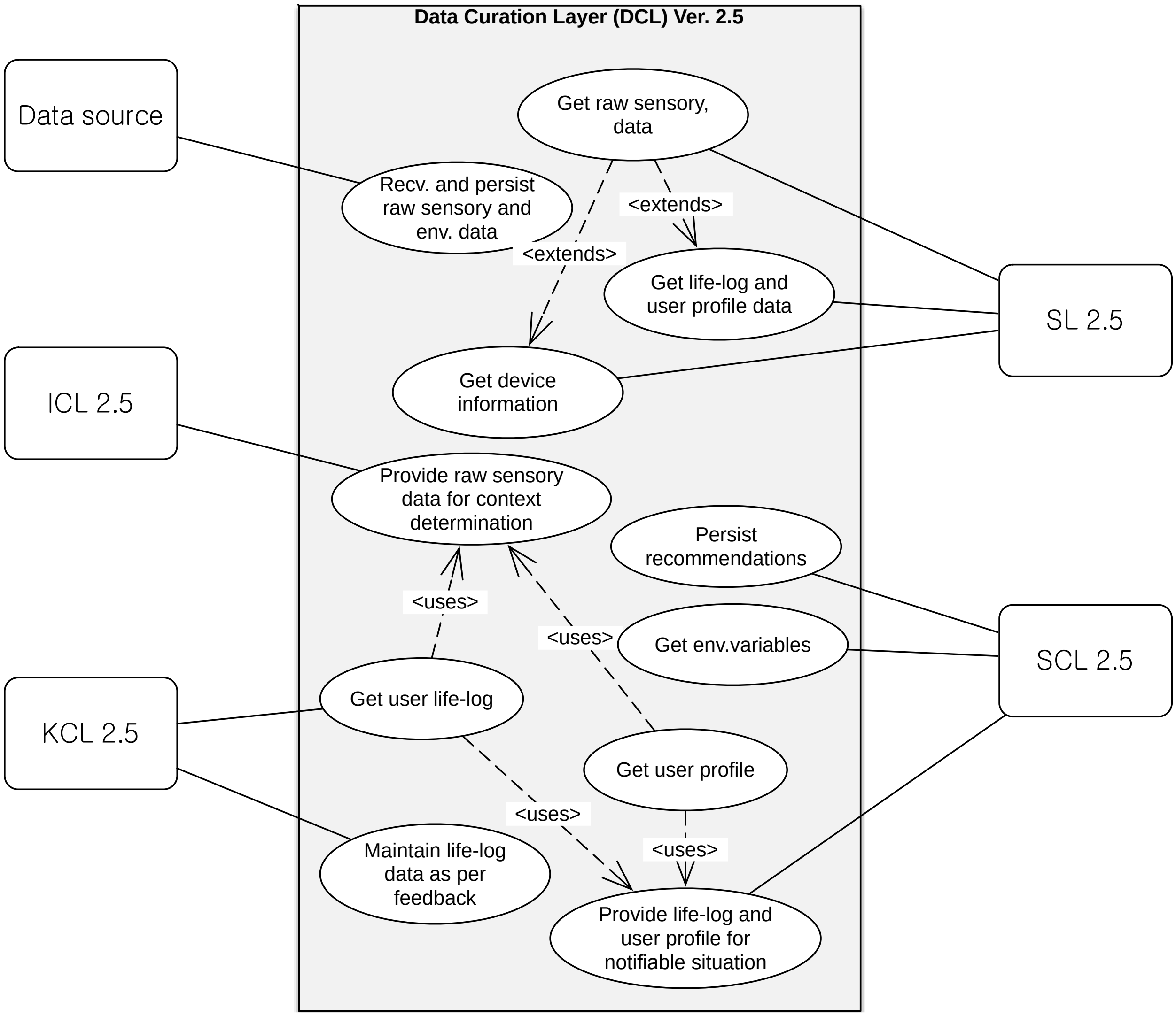
High Level Use cases

Data Curation Layer Ver. 2.5 (DCL 2.5)

Use case List

Use case ID#	Name
DCL2.5-UC-01	Receive and persist raw sensory data and environmental variables from the data source
DCL2.5-UC-02	Provide raw sensory data for context determination
DCL2.5-UC-03	Get user profile and life-log data (offline process)
DCL2.5-UC-04	Get raw sensory, life-log, and user profile data (online process)
DCL2.5-UC-05	Get device information, life-log and user profile data
DCL2.5-UC-06	Provide life-log and user profile data (trigger) for notify-able situation
DCL2.5-UC-07	Get life-log and user profile data
DCL2.5-UC-08	Get environmental variables data of the user
DCL2.5-UC-09	Persist the recommendation in life-log

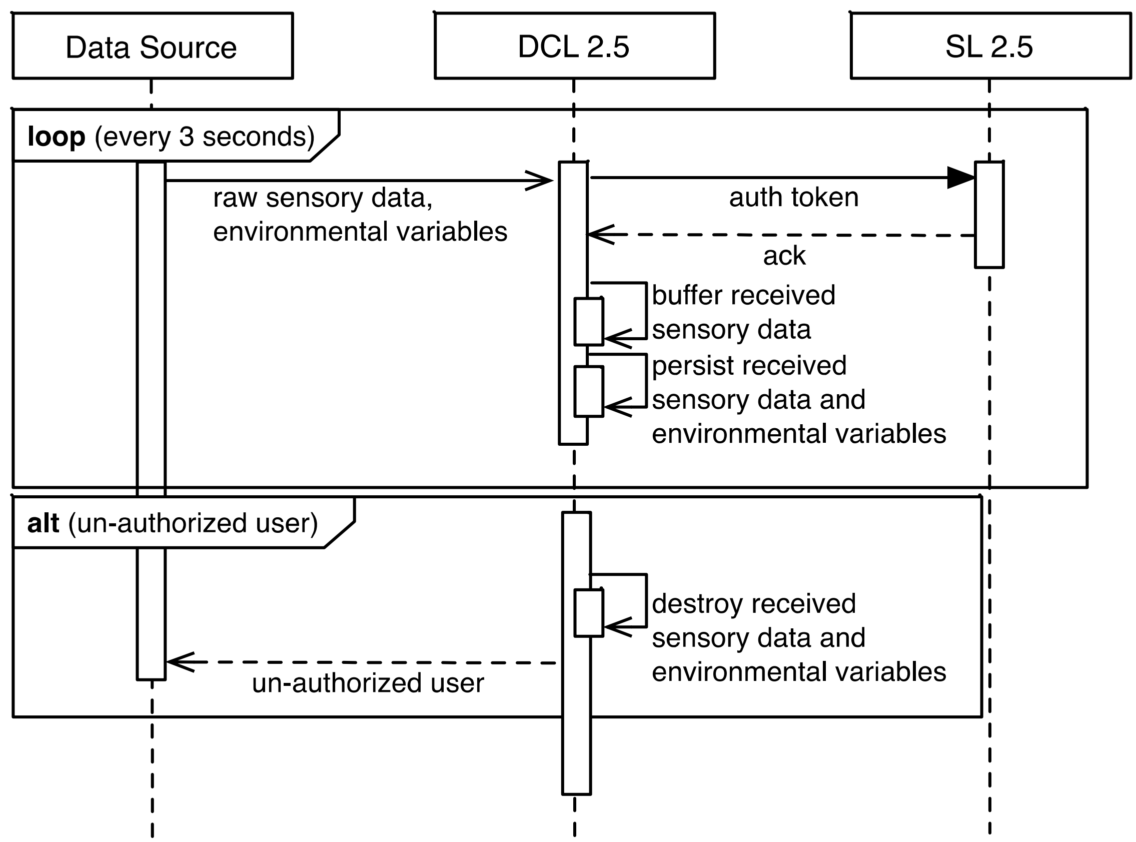
Use case Diagram



Detailed Use case

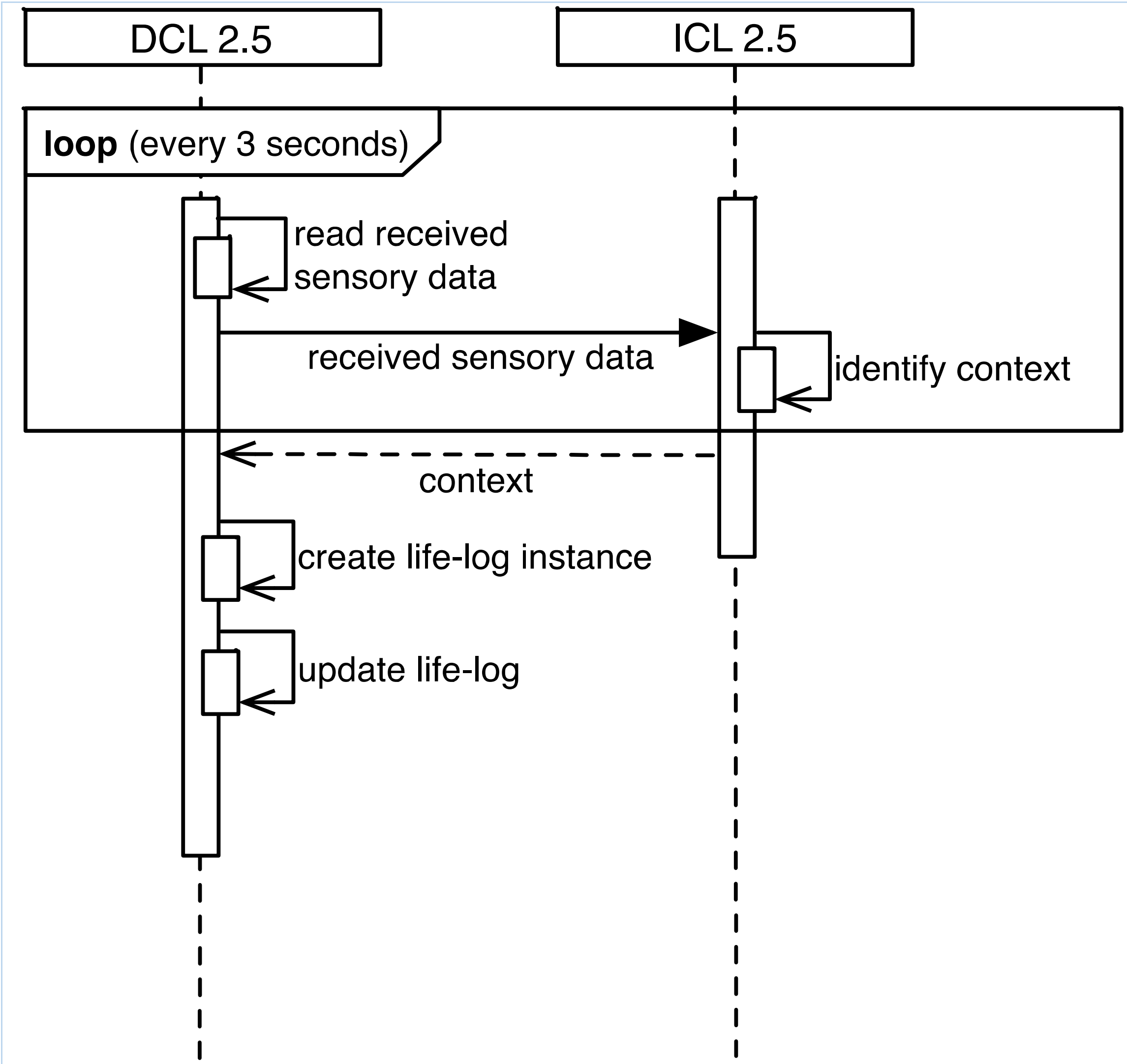
Use Case ID:	*DCL2.5-UC-01*		
Use Case Name:	Receive and persist raw sensory data and environmental variables from the data source		
FR ID:	MM-FR-01, MM-FR-02, MM-FR-04		
Created By:	Bilal Amin	**Last Updated By:**	Bilal Amin
Date Created:	06 July 2015	**Last Revision Date:**	10 Apr 2016
Actors:	Data Source, SL 2.5		
Description:	DCL 2.5 receives raw sensory data with environmental variables and persists this data in non-volatile data storage.		
Trigger:	User activity of at least 3 seconds		
Pre-conditions:	User is a registered client of MM platform		
Post-conditions:	1. Raw sensory data with environmental variables is received by the DCL 2.5 2. Raw sensory data with environmental variables is persisted in a non-volatile storage by the DCL 2.5		
Normal Flow:	1. DCL 2.5 receives the accumulated raw sensory data with environmental variables (e.g., temperature, weather, etc.) sent from the data source 2. DCL 2.5 authenticates the sensory data source 3. DCL 2.5 buffers the received sensory data for low level context determination 4. Received sensory data with environmental variables is persisted in a non-volatile storage		
Alternative Flows:	NA		
Exceptions:	2a. In step 2 of the normal flow, if the user is detected to be un-authorized 1. DCL 2.5 destroys the received sensory data and environmental variables 2. Un-authorized user message is sent to the raw sensory data source		

Includes:	NA
Frequency of Use:	Very frequent: every 3 seconds
NFR ID:	MM-NFR-01, MM-NFR-03, MM-NFR-04
Assumptions:	Communication between the data source and DCL 2.5 is secure
Notes and Issues:	NA
Sequence Diagram:	



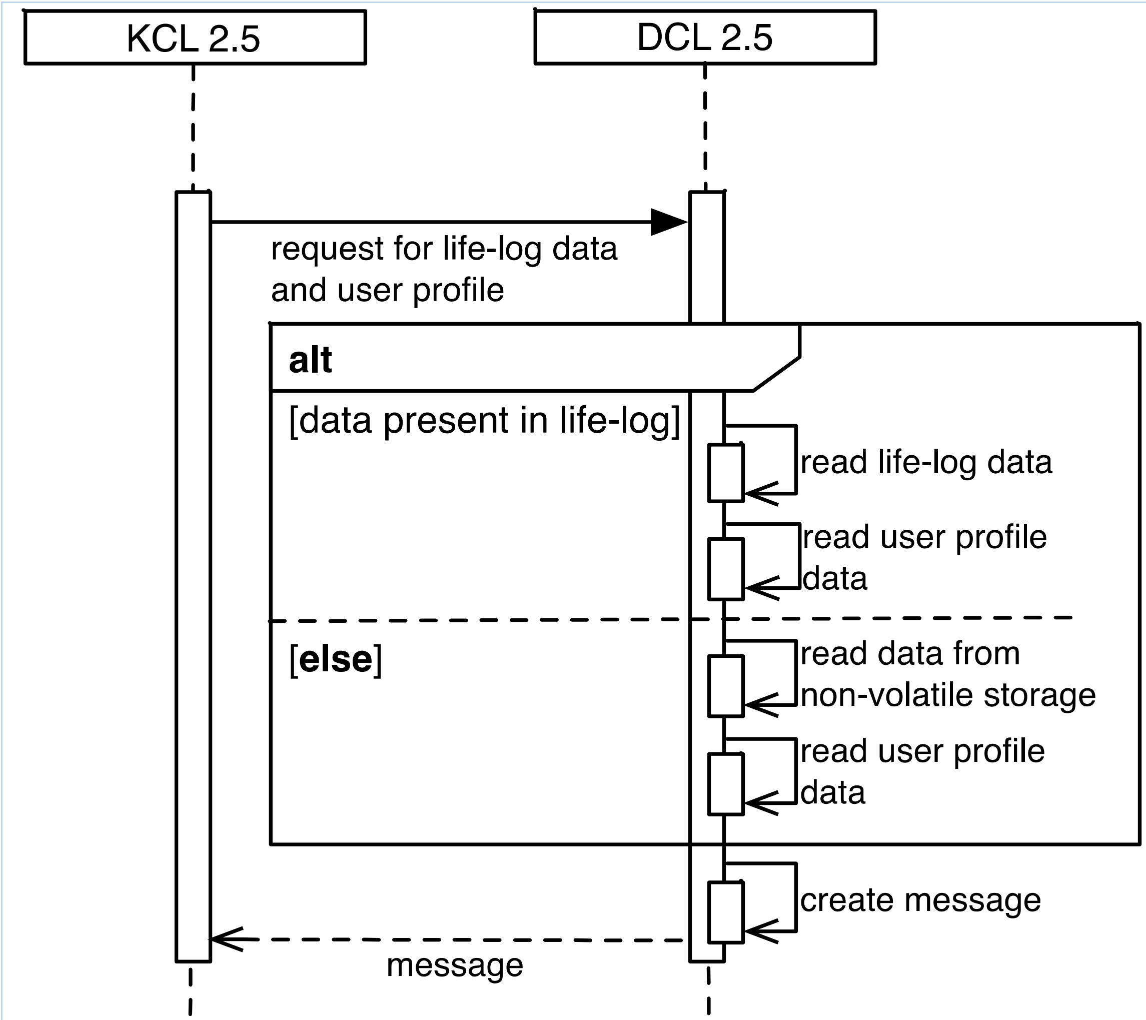
Use Case ID:	*DCL2.5-UC-02*		
Use Case Name:	Provide raw sensory data for context determination		
FR ID:	MM-FR-03, MM-FR-04		
Created By:	Bilal Amin	**Last Updated By:**	Bilal Amin
Date Created:	06 July 2015	**Last Revision Date:**	10 Apr 2016
Actors:	DCL 2.5, ICL 2.5		

Description:	Received sensory data from data source is communicated with ICL 2.5 for context determination. Determined context is added as life-log instances.
Trigger:	User activity of at least 3 seconds
Pre-conditions:	Raw sensory data has been buffered by DCL 2.5
Post-conditions:	1. Context have been received by DCL 2.5 2. Life-log has been updated
Normal Flow:	1. DCL 2.5 reads the buffered raw sensory data and sends it to ICL 2.5 2. ICL 2.5 returns the determined context 3. Life-log instance of the user for received context is created 4. Life-log is updated
Alternative Flows:	NA
Exceptions:	NA
Includes:	NA
Frequency of Use:	Very frequent: every 3 seconds
NFR ID:	MM-NFR-02, MM-NFR-04
Assumptions:	Raw sensory data buffer can be updated and read in parallel
Notes and Issues:	N/A
Sequence Diagram:	



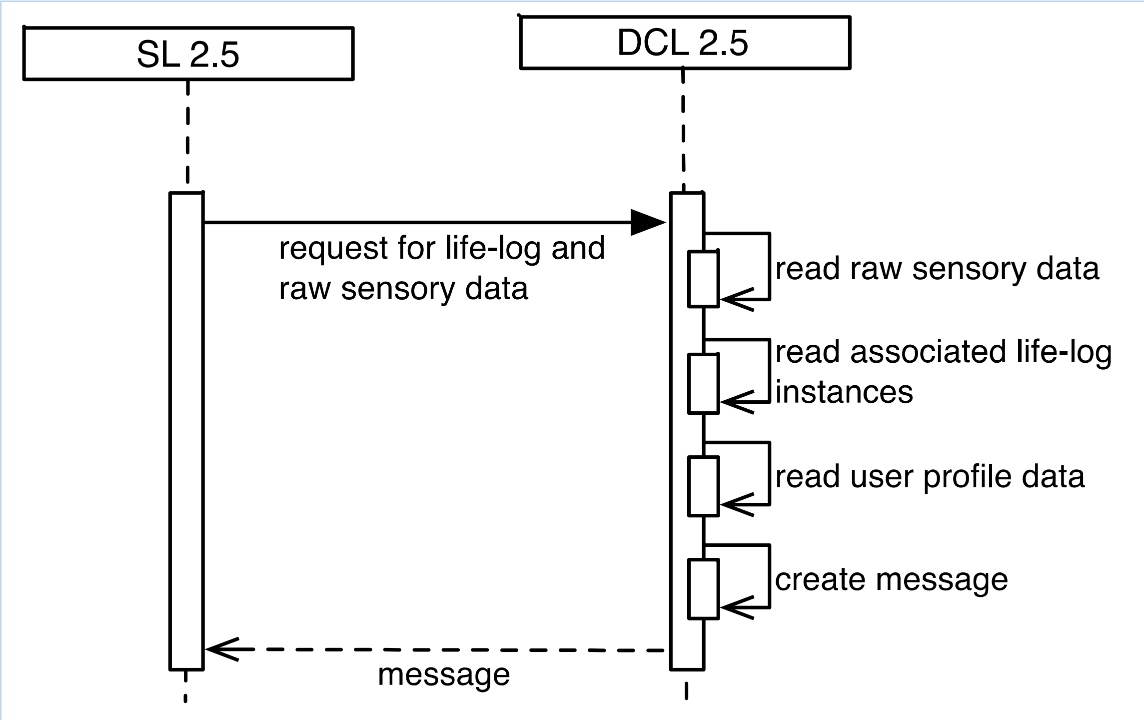
Use Case ID:	*DCL2.5-UC-03*		
Use Case Name:	Get user profile and life-log data (offline process)		
FR ID:	MM-FR-04, MM-FR-05		
Created By:	Bilal Amin	**Last Updated By:**	Bilal Amin
Date Created:	06 July 2015	**Last Revision Date:**	10 Apr 2016
Actors:	KCL 2.5		
Description:	DCL 2.5 gets user profile and life-log data for the learning of classification models in KCL 2.5		
Trigger:	Data request from KCL 2.5 (offline)		
Pre-conditions:	1. Life-log data is available		

	2. Raw sensory data has been persisted in a non-volatile storage
Post-conditions:	Required user profile and life-log data is provided to the KCL 2.5
Normal Flow:	1. DCL 2.5 receives request for user profile and life-log data from KCL 2. DCL 2.5 reads life-log data from the storage depending upon the attributes provided by KCL 2.5 3. DCL 2.5 creates data message 4. Message is sent to KCL 2.5 as a response
Alternative Flows:	2a. If KCL 2 requires data from non-volatile storage 1. DCL 2 queries non-volatile storage for life-log data depending upon the attributes provided by KCL 2 2. DCL 2 creates data message 3. Message is sent to KCL 2 as a response
Exceptions:	NA
Includes:	NA
Frequency of Use:	Less Frequent: offline process may be executed just once
NFR ID:	MM-NFR-04
Assumptions:	Service contract between DCL 2.5 and KCL 2.5 is defined
Notes and Issues:	NA
Sequence Diagram:	



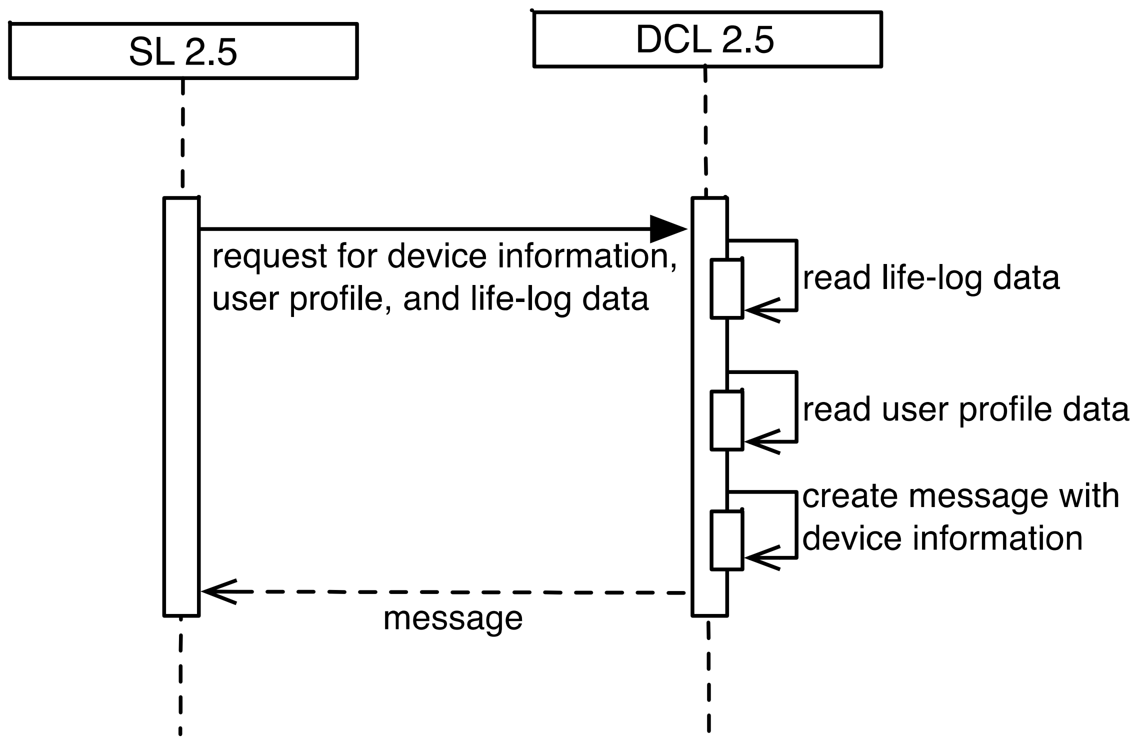
Use Case ID:	*DCL2.5-UC-04*		
Use Case Name:	Get raw sensory, life-log, and user profile data (online process)		
FR ID:	MM-FR-03, MM-FR-04, MM-FR-05		
Created By:	Bilal Amin	**Last Updated By:**	Bilal Amin
Date Created:	06 July 2015	**Last Revision Date:**	10 April 2016
Actors:	SL 2.5		
Description:	DCL 2.5 provides raw sensory, life-log, and user profile data for descriptive analytics		
Trigger:	Request for descriptive analytics is received		
Pre-conditions:	1. Context data is persisted as Life-log instances 2. Raw sensory data is persisted in non-volatile		

	storage
Post-conditions:	Required life-log instance and user sensory data is sent to SL 2.5
Normal Flow:	1. DCL 2.5 receives request for Life-log instances, user profile, and persisted raw sensory data from SL 2.5 2. DCL 2.5 reads data from non-volatile storage depending upon the attributes provided by SL 2.5 3. DCL 2.5 reads associated user life-log instances 4. DCL 2.5 reads user profile data 5. DCL 2.5 creates a message containing raw sensory data, Life-log instances, and user profile 6. Message is sent to SL 2.5 as a response
Alternative Flows:	NA
Exceptions:	NA
Includes:	NA
Frequency of Use:	Frequent: request by SL 2.5
NFR ID:	MM-NFR-04
Assumptions:	Service contract between DCL 2.5 and SL 2.5 is defined
Notes and Issues:	NA
Sequence Diagram:	



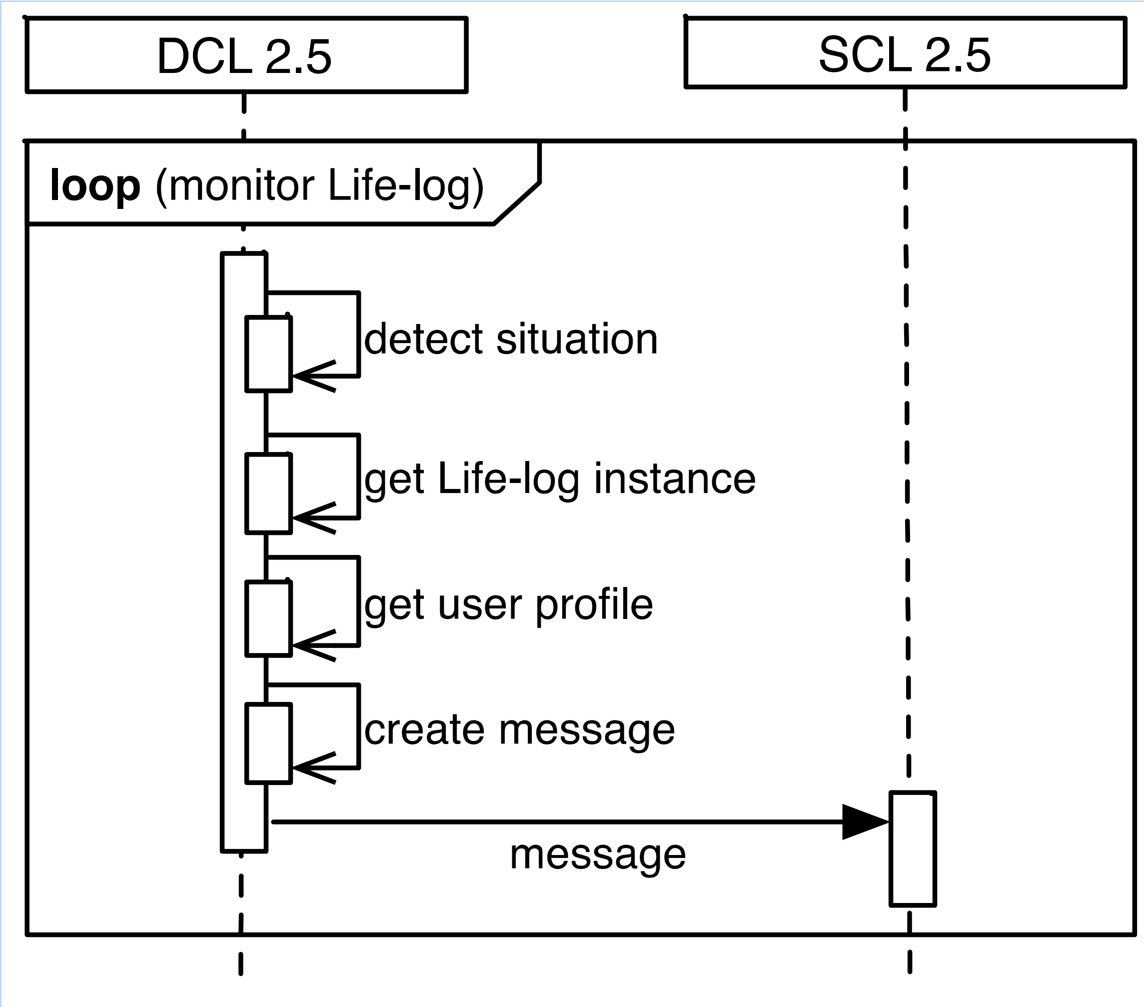
Use Case ID:	*DCL2.5-UC-05*		
Use Case Name:	Get device information, life-log and user profile data		
FR ID:	MM-FR-04, MM-FR-05		
Created By:	Bilal Amin	**Last Updated By:**	Bilal Amin
Date Created:	06 July 2015	**Last Revision Date:**	10 April 2016
Actors:	SL 2.5		
Description:	DCL 2.5 provides device information, user profile and life-log data for adaption of UI		
Trigger:	Request by SL 2.5		
Pre-conditions:	Device information, user profile and life-log data is available		
Post-conditions:	Device information, user profile and life-log data is sent to SL 2.5		
Normal Flow:	1. DCL 2.5 receives request for user device information, user profile, and life-log data 2. DCL 2.5 reads life-log data 3. DCL 2.5 reads user-profile 4. DCL 2.5 creates message by accumulating life-		

	log data with user profile and device information 5. Message is sent to SCL 2.5
Alternative Flows:	NA
Exceptions:	NA
Includes:	NA
Frequency of Use:	Frequent: request by SL 2.5
NFR ID:	MM-NFR-04
Assumptions:	Service contract between DCL 2.5 and SL 2.5 is defined
Notes and Issues:	NA
Sequence Diagram:	



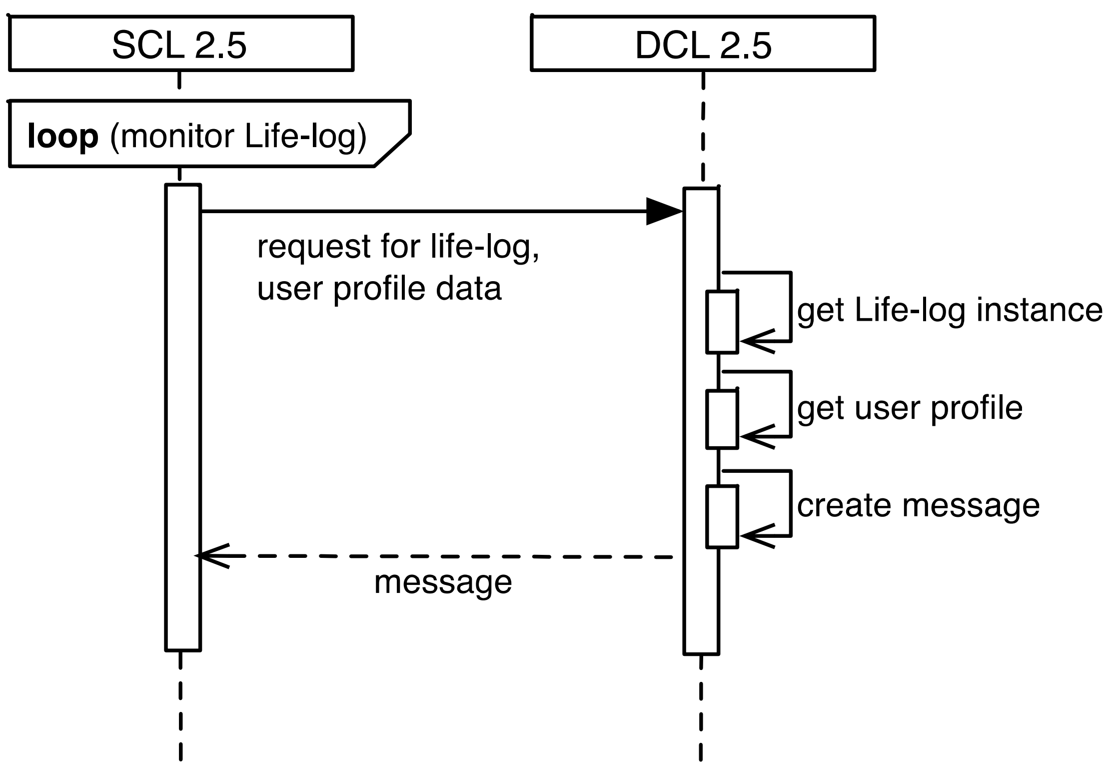
Use Case ID:	*DCL2.5-UC-06*
Use Case Name:	Provide life-log and user profile data (trigger) for notify-able situation
FR ID:	MM-FR-04, MM-FR-05, MM-FR-08
Created By:	Bilal Amin | **Last Updated By:** | Bilal Amin

Date Created:	06 July 2015	**Last Revision Date:**	10 April 2016
Actors:	DCL 2.5, SCL 2.5		
Description:	DCL 2.5 detects a situation over life-log data and triggers SCL 2.5 for recommendation generation		
Trigger:	Situation detected over Life-log data		
Pre-conditions:	Context data is persisted as Life-log instances		
Post-conditions:	Required life-log instance and user profile is sent to SCL 2.5		
Normal Flow:	1. DCL 2.5 performs continuous monitoring of life-log data 2. DCL 2.5 detects situation over life-log data 3. DCL 2.5 creates message by accumulating related life-log data and user profile 4. Message is sent to SCL 2.5 as a trigger		
Alternative Flows:	NA		
Exceptions:	NA		
Includes:	NA		
Frequency of Use:	Frequent: implicit invocation of recommendation		
NFR ID:	MM-NFR-04		
Assumptions:	Service contract between DCL 2.5 and SCL 2.5 is defined		
Notes and Issues:	NA		
Sequence Diagram:			

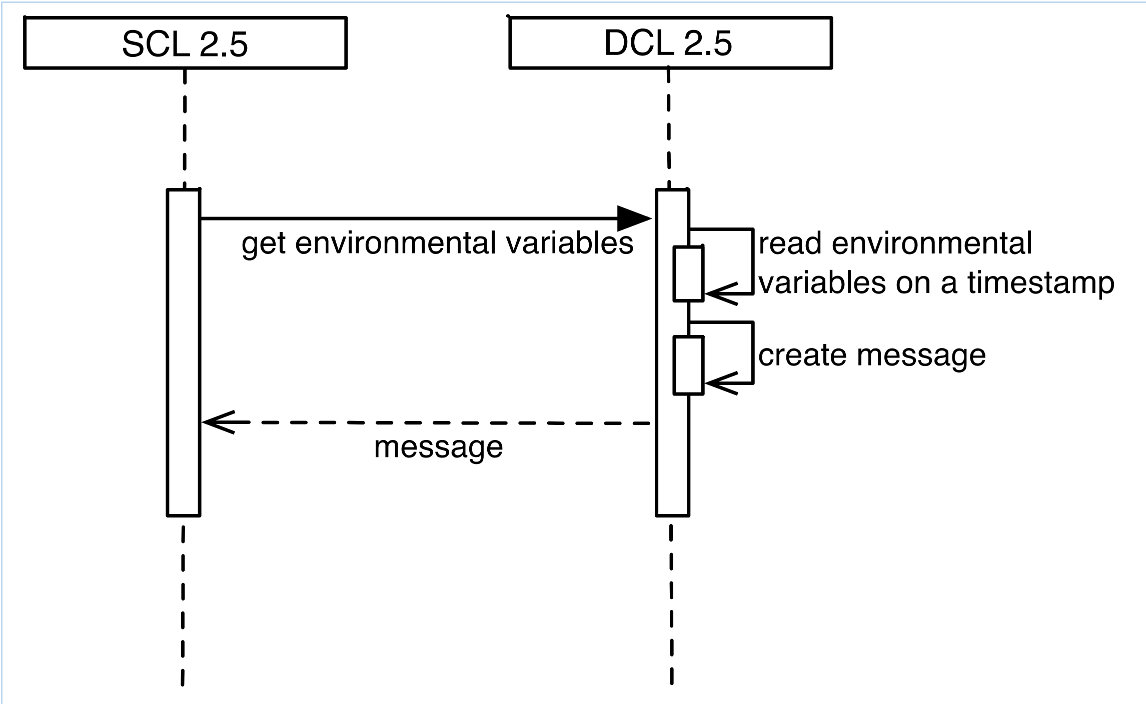


Use Case ID:	*DCL2.5-UC-07*		
Use Case Name:	Get life-log and user profile data		
FR ID:	MM-FR-04, MM-FR-05		
Created By:	Bilal Amin	**Last Updated By:**	Bilal Amin
Date Created:	06 July 2015	**Last Revision Date:**	10 April 2016
Actors:	SCL 2.5		
Description:	DCL 2.5 provides life-log and user profile data to SCL 2.5 for recommendation generation		
Trigger:	Request for recommendation generation by SCL 2.5		
Pre-conditions:	Context data is persisted as Life-log instances		
Post-conditions:	Required life-log instance and user profile is sent to		

	SCL 2.5
Normal Flow:	1. DCL 2.5 receives request for life-log and user profile data from SCL 2.5 2. DCL 2.5 creates message by accumulating related life-log data and user profile 3. Message is sent to SCL 2.5 as a response
Alternative Flows:	NA
Exceptions:	NA
Includes:	NA
Frequency of Use:	Frequent: implicit invocation of recommendation
NFR ID:	MM-NFR-04
Assumptions:	Service contract between DCL 2.5 and SCL 2.5 is defined
Notes and Issues:	NA
Sequence Diagram:	

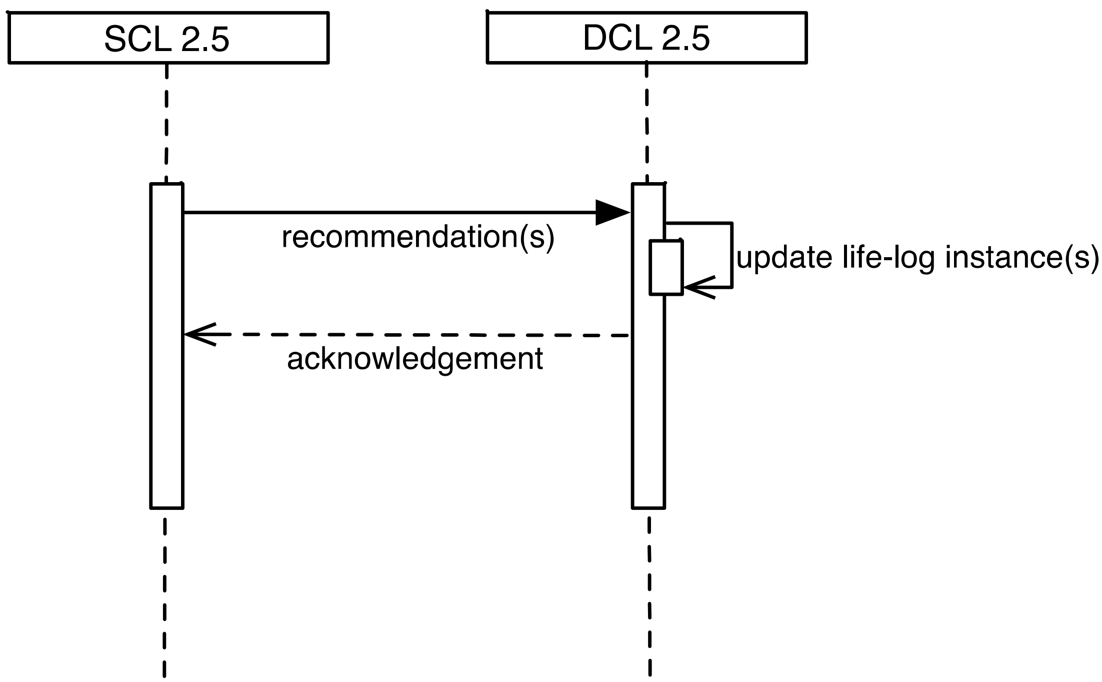


Use Case ID:	*DCL2.5-UC-08*		
Use Case Name:	Get environmental variables data of the user		
FR ID:	MM-FR-04		
Created By:	Bilal Amin	**Last Updated By:**	Bilal Amin
Date Created:	06 July 2015	**Last Revision Date:**	10 April 2016
Actors:	SCL 2.5		
Description:	DCL 2.5 retrieves the environmental variables persisted for a user according to timestamp		
Trigger:	Recommendation needs to be generated		
Pre-conditions:	Recommendations are available for generation		
Post-conditions:	Environmental variables are sent to SCL 2.5		
Normal Flow:	1. DCL 2.5 receives request from SCL 2.5 for environmental variables 2. DCL 2.5 retrieves environmental variables based on timestamp from user profile 3. DCL 2.5 sends the environmental variables as a message to SCL 2.5		
Alternative Flows:	NA		
Exceptions:	NA		
Includes:	NA		
Frequency of Use:	Frequent: as a response to a recommendation or identified context		
NFR ID:	MM-NFR-04		
Assumptions:	Service contract between DCL 2.5 and SCL 2.5 is defined		
Notes and Issues:	NA		
Sequence Diagram:			



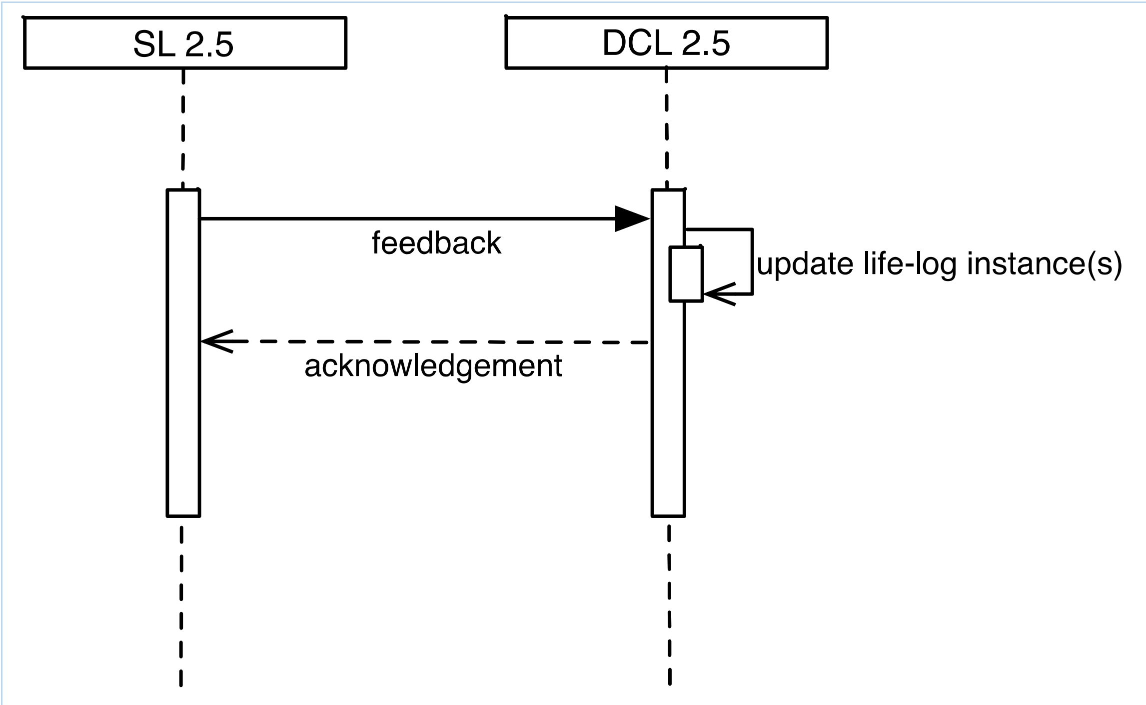
Use Case ID:	*DCL2.5-UC-09*		
Use Case Name:	Persist the recommendation in life-log		
FR ID:	MM-FR-05		
Created By:	Bilal Amin	**Last Updated By:**	Bilal Amin
Date Created:	06 July 2015	**Last Revision Date:**	10 April 2016
Actors:	SCL 2.5		
Description:	DCL 2.5 updates the life-log instance with the recommendations provided to the user		
Trigger:	Generated recommendations need to be persisted		
Pre-conditions:	Recommendation has been generated		
Post-conditions:	Life-log instances are updated with recommendations		
Normal Flow:	1. DCL 2.5 receives generated recommendations from the SCL 2.5 2. DCL 2.5 updates the associated life-log instance(s) 3. DCL 2.5 acknowledges the update to SLC 2.5		
Alternative Flows:	NA		

Exceptions:	NA
Includes:	NA
Frequency of Use:	Frequent: as a response to a recommendation generation
NFR ID:	MM-NFR-04
Assumptions:	Service contract between DCL 2.5 and SCL 2.5 is defined
Notes and Issues:	NA
Sequence Diagram:	



Use Case ID:	*DCL2.5-UC-10*		
Use Case Name:	Maintain (update/delete) life-log entries as per user feedback		
FR ID:	MM-FR-09		
Created By:	Bilal Amin	**Last Updated By:**	Bilal Amin
Date Created:	06 July 2015	**Last Revision Date:**	10 April 2016
Actors:	SL 2.5		

Description:	DCL 2.5 updates the life-log instance from the feedback provided by user
Trigger:	User provide feedback on a generated recommendation or identified context
Pre-conditions:	1. Activities data is persisted as Life-log instance(s) 2. Recommendations are available
Post-conditions:	Life-log instances are updated
Normal Flow:	1. DCL 2.5 receives feedback from the SL 2.5 2. DCL 2.5 updates the associated life-log instance(s) 3. DCL 2.5 acknowledges the update to SL 2.5
Alternative Flows:	NA
Exceptions:	NA
Includes:	NA
Frequency of Use:	Less Frequent: as a response to a recommendation or identified context
NFR ID:	MM-NFR-04
Assumptions:	Service contract between DCL 2.5 and SL 2.5 is defined
Notes and Issues:	NA
Sequence Diagram:	

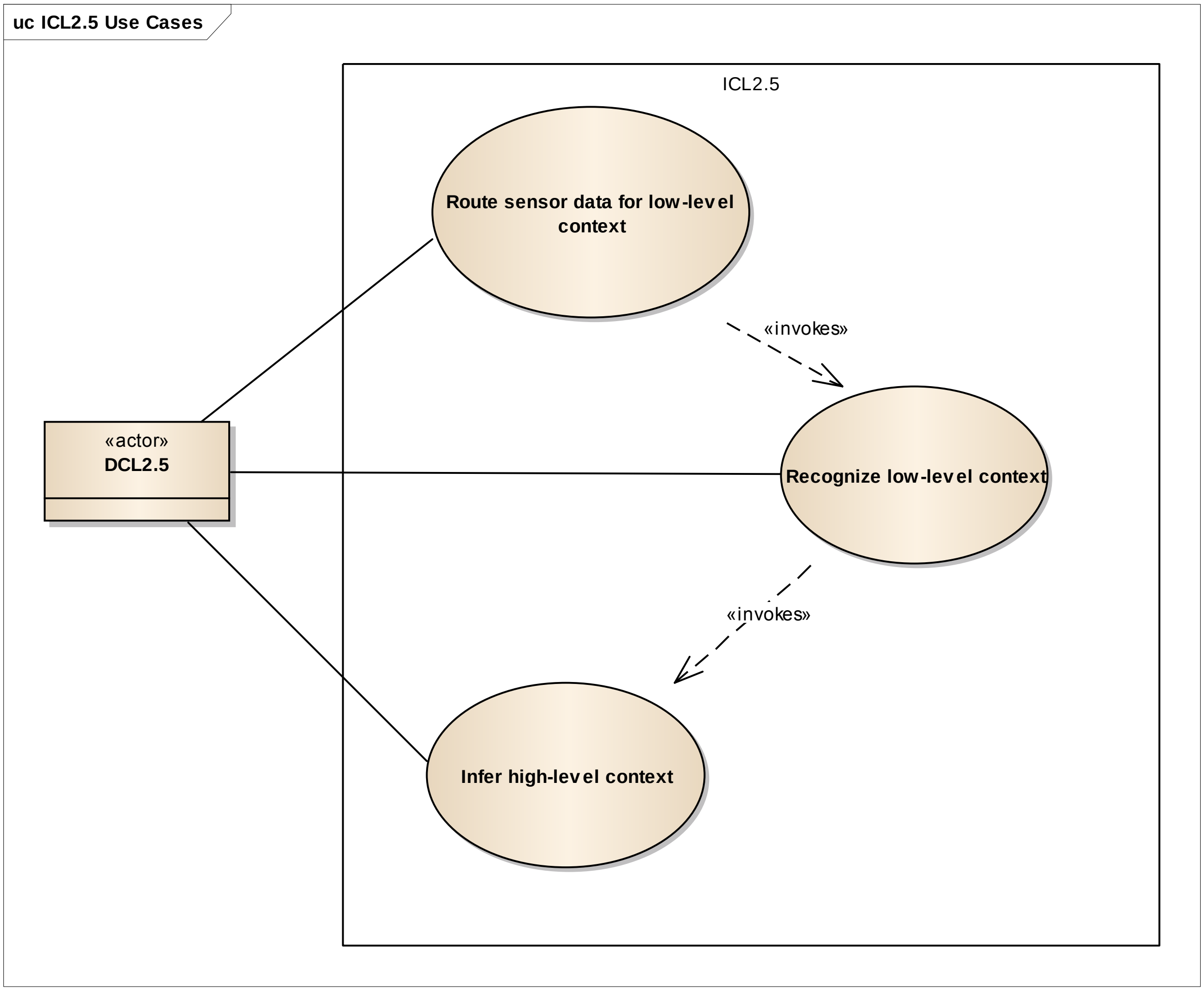

SL 2.5
DCL 2.5
feedback
update life-log instance(s)
acknowledgement

Information Curation Layer Ver. 2.5 (ICL 2.5)

Use case List

Use case ID#	Name
ICL2.5-UC-01	Route raw sensory data for the low-level context identification
ICL2.5-UC-02	Recognize low-level context from raw sensory data
ICL2.5-UC-03	Infer high-level context from low-level context

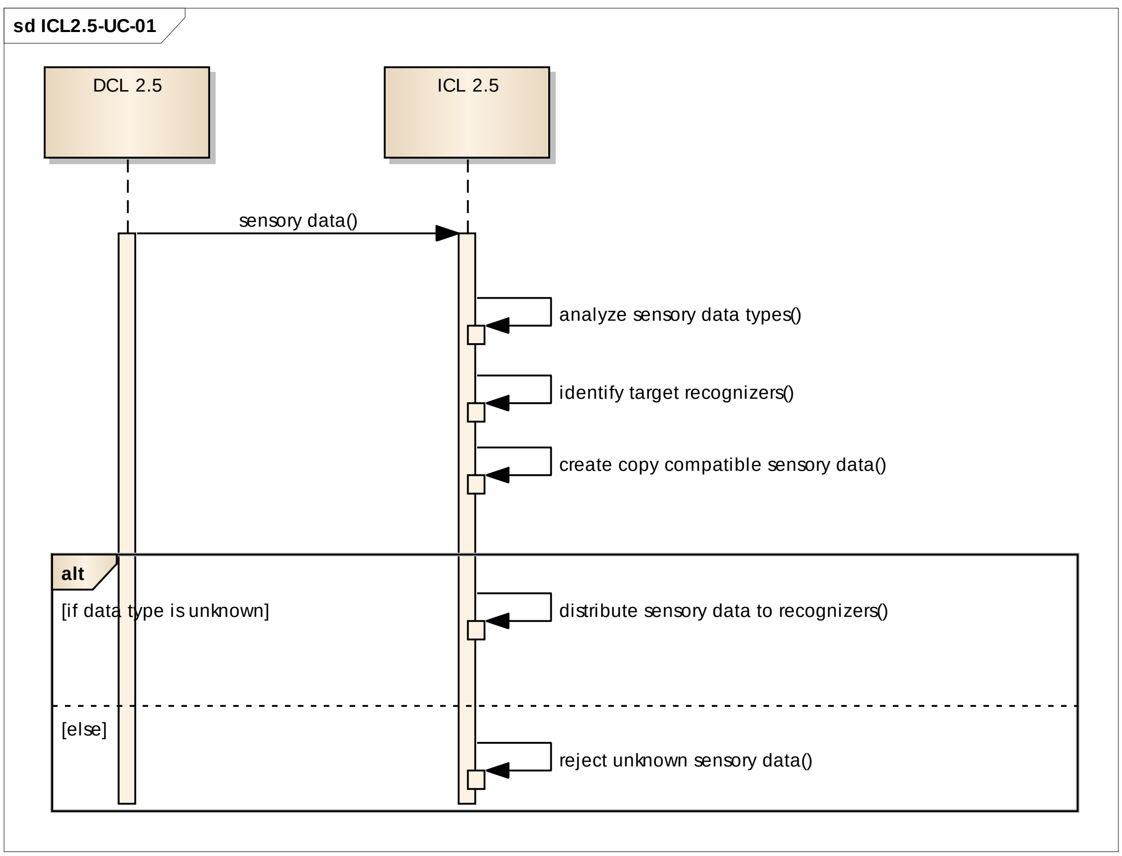
Use case Diagram



Detailed Use case

Use Case ID:	*ICL2.5-UC-01*		
Use Case Name:	Route raw sensory data for the low-level context identification		
FR ID:	MM-FR-10		
Created By:	Oresti Banos	**Last Updated By:**	Wajahat Ali Khan
Date Created:	06 July 2015	**Last Revision Date:**	14 April 2016
Actors:	DCL 2.5		
Description:	Sensory data is received from DCL 2.5 and it is distributed to the corresponding low-level context recognizer based on the data type(s).		
Trigger:	Receive sensory data send by DCL 2.5 to ICL 2.5		
Pre-conditions:	DCL 2.5 sends sensory data, i.e., raw sensory data including metadata (e.g., data type, time stamp and user ID)		
Post-conditions:	The adequate raw sensory data is sent to each low-level context recognizer in order to perform the recognition process		
Normal Flow:	1. Receive sensory data 2. Analyze the data type(s) 3. Identify the low-level context recognizer(s) that require this sensory data 4. Create copies with the data required for each low-level context recognizer 5. Distribute the data to each corresponding low-level context recognizer(s)		
Alternative Flows:	NA		
Exceptions:	3a. If data type is unknown or not of the type processible by the actual low-level context recognizer(s) 1. Reject the unknown sensory data		
Includes:	NA		
Frequency of Use:	Very frequent: determined by the rate of sensory data		

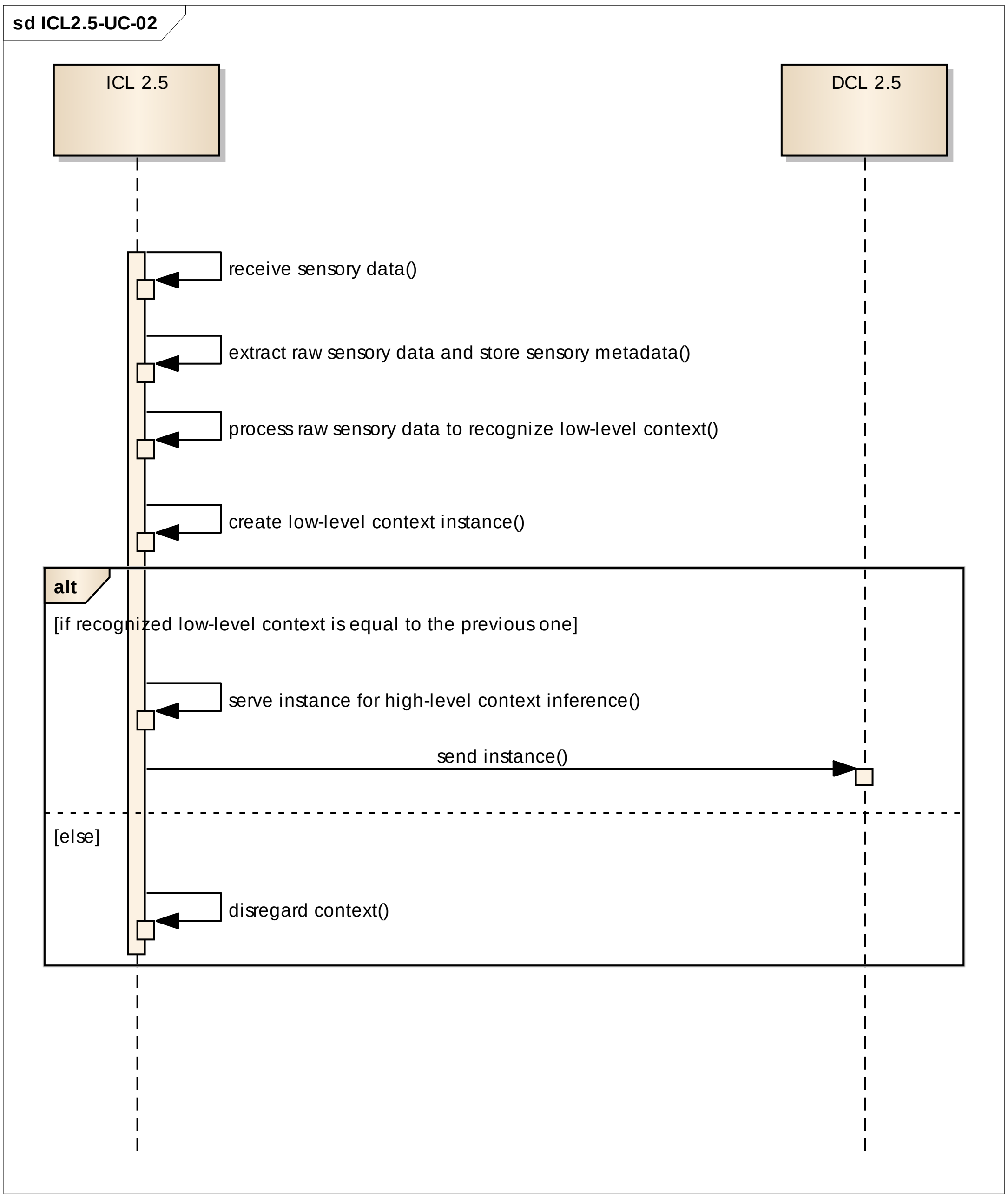
	reception from DCL 2.5
NFR ID:	NA
Assumptions:	1. There is an established communication between DCL 2.5 and ICL 2.5 2. The communication channel between the DCL 2.5 and the ICL 2.5 is secure 3. Incoming sensory data is already preprocessed (i.e., without missing samples and with synchronized streams)
Notes and Issues:	NA
Sequence Diagram:	



Use Case ID:	*ICL2.5-UC-02*
Use Case Name:	Recognize low-level context from raw sensory data
FR ID:	MM-FR-11, MM-FR-13

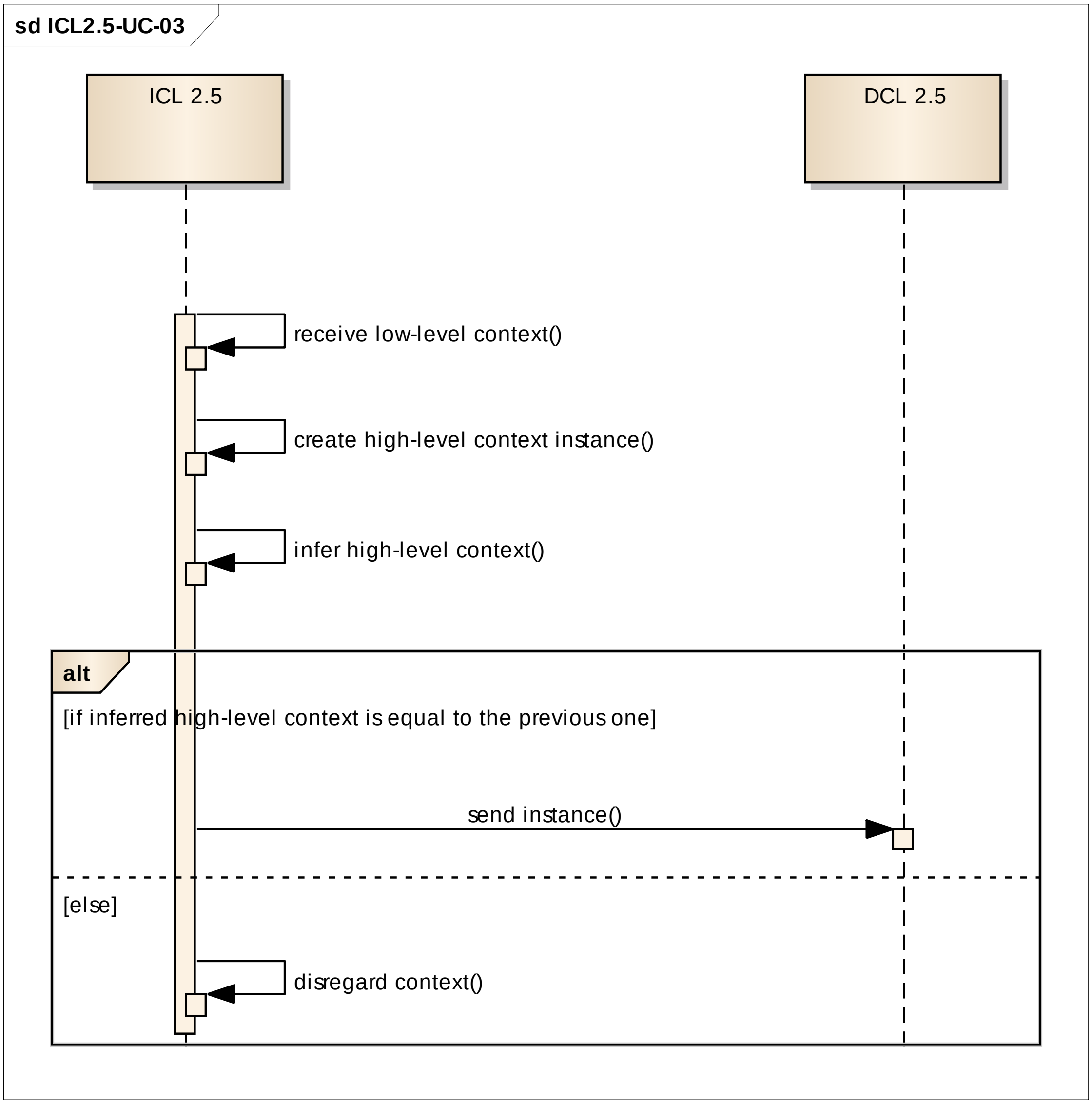
Created By:	Oresti Banos	**Last Updated By:**	Wajahat Ali Khan
Date Created:	06 July 2015	**Last Revision Date:**	14 April 2016
Actors:	ICL 2.5, DCL 2.5		
Description:	Low-level contexts are recognized for a given user based on the received sensory data. The identified contexts are served for the high-level context inference and also communicated to DCL 2.5 for the generation of the life-log.		
Trigger:	Receive compatible sensory data		
Pre-conditions:	Compatible sensory data is sent to the low-level context recognizer		
Post-conditions:	1. The recognized low-level context(s) are served for the identification of the high-level context(s) 2. The recognized low-level context(s) including metadata (e.g., time stamp and user ID) are sent to DCL 2.5		
Normal Flow:	1. Sensory data is received by a given low-level context recognizer 2. The raw sensory data is extracted and the sensory metadata (i.e., time stamp and user ID) temporarily stored 3. The raw sensory data is processed and the corresponding low-level context(s) are recognized 4. A new low-level context instance is created, including the cashed sensory metadata (i.e., time stamp and user ID) 5. The instance is served for high-level context inference 6. The instance is sent to DCL 2.5 for the generation of the life-log		
Alternative Flows:	4a. If the identified low-level context is equal to the previous one 1. Disregard low-level context		
Exceptions:	NA		

Includes:	NA
Frequency of Use:	Frequent: at every reception of sensory data
NFR ID:	MM-NFR-05
Assumptions:	Only compatible sensory data is received by each corresponding low-level context recognizer
Notes and Issues:	N/A
Sequence Diagram:	



Use Case ID:	*ICL2.5-UC-03*		
Use Case Name:	Infer high-level context from low-level context		
FR ID:	MM-FR-12, MM-FR-14		
Created By:	Oresti Banos	**Last Updated By:**	Wajahat Ali Khan
Date Created:	06 July 2015	**Last Revision Date:**	14 April 2016
Actors:	ICL 2.5, DCL 2.5		
Description:	High-level contexts are recognized for a given user based on the identified low-level contexts. High-level context(s) are communicated to DCL 2.5 for the generation of the life-log.		
Trigger:	Receive low-level context		
Pre-conditions:	3. A new low-level context instance is served to the high-level context inferrer		
Post-conditions:	3. The inferred high-level context(s) including metadata (e.g., time stamp and user ID) are sent to DCL 2.5		
Normal Flow:	1. A low-level context instance is received 2. An unknown high-level context instance is created 3. The type of high-level context instance is inferred 4. The inferred high-level context is communicated to DCL 2.5 for the generation of the life-log		
Alternative Flows:	4a. If the high-level context is equal to the previous one 1. Disregard high-level context		
Exceptions:	NA		
Includes:	NA		
Frequency of Use:	Frequent: whenever a new low-level context is recognized		
NFR ID:	MM-NFR-06		

Assumptions:	Low-level contexts are interpretable
Notes and Issues:	NA
Sequence Diagram:	

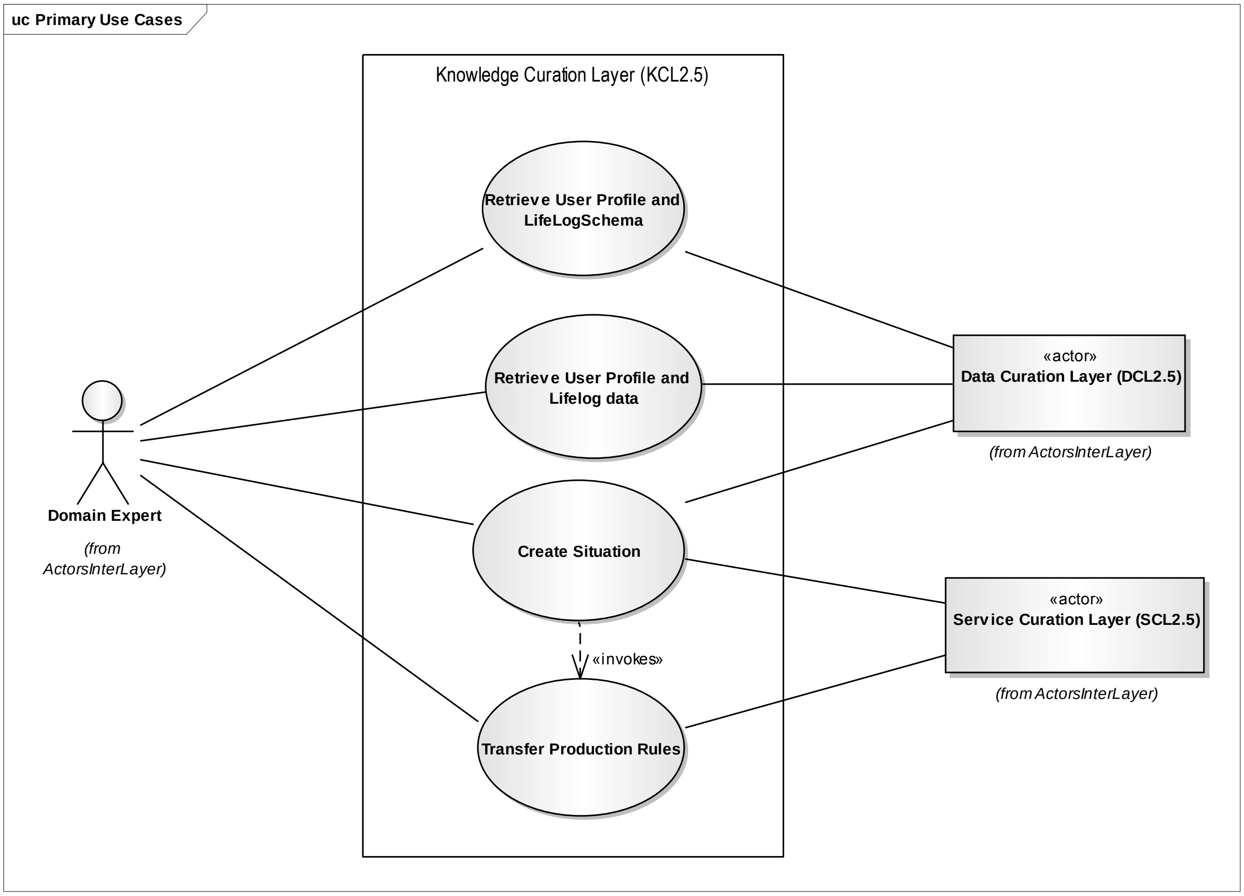


Knowledge Curation Layer ver. 2.5 (KCL 2.5)

Use case List

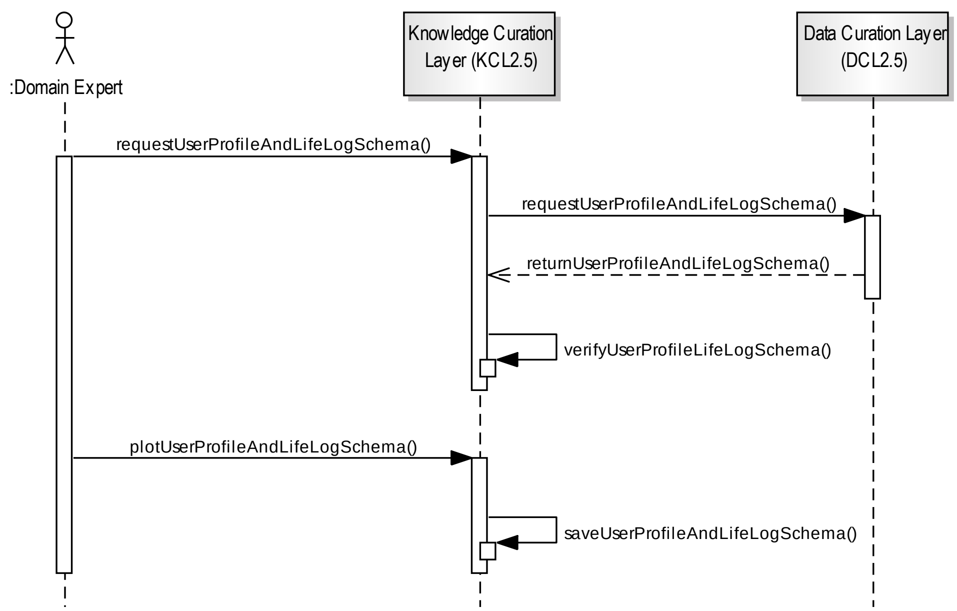
Use case ID#	Name
KCL2.5-UC-01	Retrieve lifelog and user profile schema from DCL 2.5 to create meta feature model and load lifelog and user profile data.
KCL2.5-UC-02	Retrieve lifelog and user profile data in accordance to feature model selected by domain expert for generating classification model.
KCL2.5-UC-03	Share the newly/updated knowledge with SCL 2.5 for final reasoning.
KCL2.5-UC-04	Create "Situation" and shared its configuration with SCL 2.5 and DCL 2.5. Furthermore, share all associated rules with SCL 2.5 for handling the created situation.

Use case Diagram

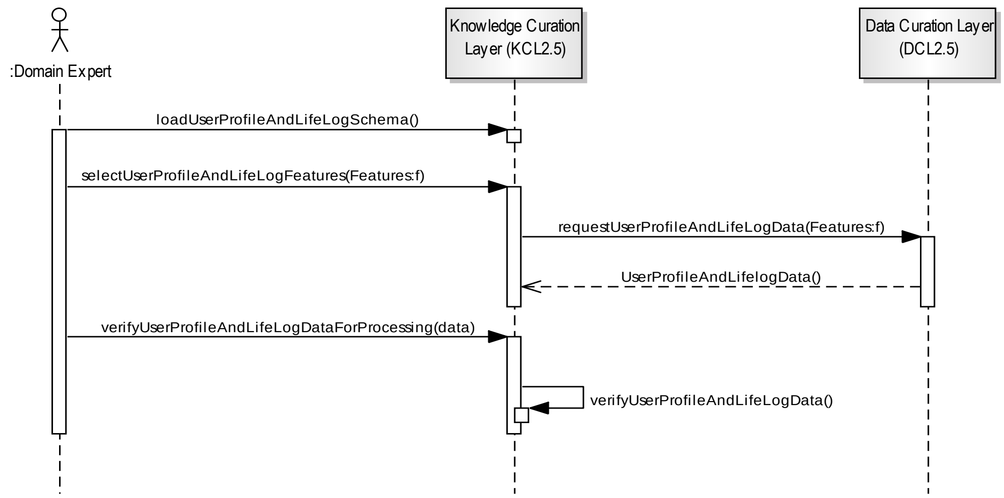


Detailed Use cases

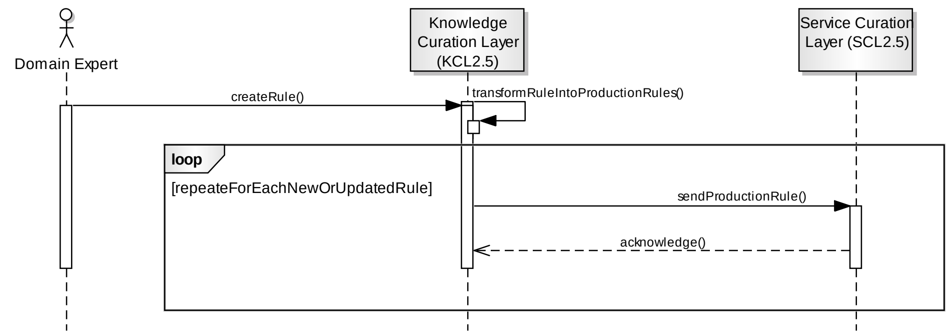
Use Case ID:	**KCL2.5-UC-01**		
Use Case Name:	Retrieve User Profile and LifeLog Schema		
Created By:	Maqbool Ali	**Last Updated By:**	Maqbool Hussain
Date Created:	03-07-2015	**Last Revision Date:**	15-04-2016
Actors:	Domain Expert ,DCL 2.5		
Description:	In domain engineering, features modeling enables domain experts to capture variability in a domain. Lifelog and user profile schema retrieval help domain expert to select important features/attributes for generating high quality of knowledge and reusability.		
Trigger:	Prior to classification model creation needed for required domain		
Preconditions:	• KCL 2.5 has access through service interface to retrieve lifelog and user profile schema from DCL 2.5 • KCL 2.5 and DCL 2.5 has agreement on common schema representation format • DCL 2.5 has capability to share lifelog and user profile schema in secure environment.		
Postconditions:	KCL 2.5 will receive lifelog and user profile schema conform to its representation scheme.		
Normal Flow:	1. KCL 2.5 connects to DCL 2.5 via unified service interface and sends request for lifelog and user profile schema. 2. DCL 2.5 retrieves lifelog and user profile schema, transform into common representation format and sends to KCL 2.5. 3. KCL 2.5 receives lifelog and user profile schema and verifies its conformance and sends received acknowledgement to DCL 2.5. 4. Domain expert plots the schema and saves it after verification, for further process.		
Alternative Flows:	N/A		
Exceptions:	1a. KCL 2.5 unable to connect to DCL 2.5 a. KCL 2.5 connection is failed during retrieving lifelog and user profile schema b. KCL 2.5 hold and will retry after sometime to connect to DCL 2.5 and retrieve the lifelog and user profile schema 3a. KCL 2.5 unable to verify lifelog and user profile schema conformance a. KCL 2.5 fail to conform the schema representation from		

	DCL 2.5 b. KCL 2.5 will send message to DCL 2.5 about incompatible schema format
Includes:	N/A
Frequency of Use:	When new service is required and mining mind have sufficient data for classification model creation
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	If DCL 2.5 is unable to send lifelog and user profile schema in required format, then alternate strategy has to be considered.
Sequence Diagram:	

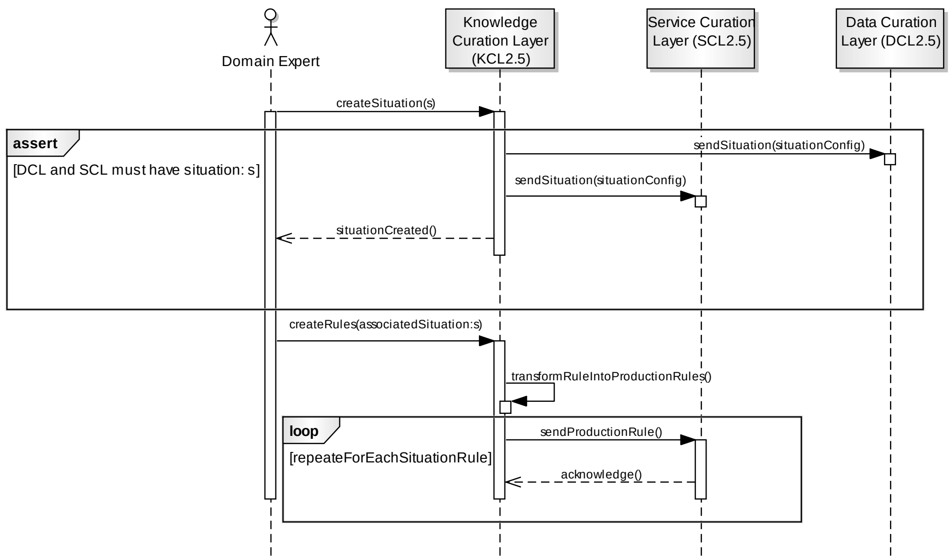
Use Case ID:	**KCL2.5-UC-02**		
Use Case Name:	Retrieve User Profile and LifeLog data		
Created By:	Maqbool Ali	**Last Updated By:**	Maqbool Hussain
Date Created:	03-07-2015	**Last Revision Date:**	15-04-2016
Actors:	Domain Expert, DCL 2.5		
Description:	Lifelog and user profile data has hidden knowledge and it is important to load the life-log information for exploring the hidden knowledge. Furthermore, lifelog and user profile data are used for model learning to explore the hidden knowledge.		
Trigger:	Prior to classification model creation needed for required domain		
Preconditions:	• KCL 2.5 has access through service interface to retrieve lifelog and user profile data from DCL 2.5 • DCL 2.5 has capability to share lifelog and user profile data in secure environment. • KCL 2.5 has already loaded the previously imported lifelog and user profile schema		
Postconditions:	KCL 2.5 will receive lifelog and user profile data based on selected schema		
Normal Flow:	1. Domain expert selects the required features of lifelog and user profile schema from whole DCL 2.5 provided schema 2. KCL 2.5 connects to DCL 2.5 via unified service interface and sends request for lifelog and user profile data based on selected features of schema. 3. DCL 2.5 retrieves lifelog and user profile data and sends to KCL 2.5. 4. KCL 2.5 receives lifelog and user profile data for further processing and sends received acknowledgement to DCL 2.5.		
Alternative Flows:			
Exceptions:	2a. KCL 2.5 unable to connect to DCL 2.5 a. KCL 2.5 connection is failed during retrieving lifelog and user profile data b. KCL 2.5 hold and will retry after sometime to connect to DCL 2.5 and retrieve the lifelog and user profile data 3a. KCL 2.5 receives irrelevant data a. KCL 2.5 detects the irrelevant data sent by DCL 2.5. b. KCL 2.5 request again DCL 2.5 to make sure that data received is according to feature selected.		

Includes:	N/A
Frequency of Use:	When new service is required and mining mind have sufficient data for classification model creation
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	If DCL 2.5 is unable to send data based on dynamic feature selection from schema, then alternate strategy has to be considered.
Sequence Diagram:	

Use Case ID:	**KCL2.5-UC-03**		
Use Case Name:	Transfer Production Rules		
Created By:	Maqbool Hussain	**Last Updated By:**	Maqbool Hussain
Date Created:	03-07-2015	**Last Revision Date:**	15-04-2016
Actors:	Domain Expert ,SCL 2.5		
Description:	Transfer Production Rules use case transfer final production rules to SCL 2.5. SCL 2.5 integrate these rules to support recommendations for different services.		
Trigger:	• New rule creation • Update existing rule		
Preconditions:	SCL 2.5 and KCL 2.5 should agree on common rule representation for final reasoning process		
Postconditions:	SCL 2.5 will have latest knowledge from KCL 2.5		
Normal Flow:	1. Domain expert using KCL 2.5 authoring tool to create rule(s) a. Using guidelines or b. Using classification model obtained from data-driven approach or c. Using directly rule editor (such as situation based rule) 2. The created rule(s) is/are transformed by KCL 2.5 into production rule(s) representation scheme 3. The transform production rule(s) is/are stored by KCL 2.5 for maintaining knowledge base and establish connection with SCL 2.5 for sharing the production rule(s). 4. The production rule(s) is/are transferred to SCL 2.5. 5. SCL 2.5 receive the production rule(s) and acknowledge with updated knowledge at SCL 2.5.		
Alternative Flows:	1a. Domain expert modify existing rule(s) using KCL 2.5 authoring tool a. Updated rule(s) is/are transformed by KCL 2.5 into production rule(s) representation scheme b. Step 2-5 of normal flow is used by indicating updated rule(s) in step 4 to SCL 2.5.		
Exceptions:	3a. KCL 2.5 unable to connect to SCL 2.5 a. KCL 2.5 connection is failed during sharing production rule(s) b. KCL 2.5 hold and will retry after sometime to connect to SCL 2.5 and transfer the production rule(s)		

Includes:	N/A
Frequency of Use:	Once per change in knowledge base
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	Common knowledge representation scheme is challenging to implement.
Sequence Diagram:	

<table>
<tr><td>Use Case ID:</td><td colspan="3">KCL2.5-UC-04</td></tr>
<tr><td>Use Case Name:</td><td colspan="3">Create Situation Event</td></tr>
<tr><td>Created By:</td><td>Maqbool Hussain</td><td>Last Updated By:</td><td>Maqbool Hussain</td></tr>
<tr><td>Date Created:</td><td>03-07-2015</td><td>Last Revision Date:</td><td>15-04-2016</td></tr>
<tr><td>Actors:</td><td colspan="3">Domain Expert, DCL 2.5 and SCL 2.5</td></tr>
<tr><td>Description:</td><td colspan="3">Situation is important features of mining mind which includes set of associated recommendation rules. Moreover, situation configurations are needed to be shared with DCL 2.5 to take care and notify SCL 2.5 to handle it if situation is observed. SCL 2.5 handle the situation and use KCL 2.5 production rule(s) to provide appropriate recommendations.</td></tr>
<tr><td>Trigger:</td><td colspan="3">Situation is identified by domain expert</td></tr>
<tr><td>Preconditions:</td><td colspan="3">KCL 2.5, DCL 2.5 and SCL 2.5 should agree on common representation of sharing Situation configuration</td></tr>
<tr><td>Postconditions:</td><td colspan="3">• KCL 2.5 has configured situation, shared with DCL 2.5 and
• all associated rules are created and shared with SCL 2.5</td></tr>
<tr><td>Normal Flow:</td><td colspan="3">1. Domain expert uses authoring tool of KCL 2.5 and create situation.
2. KCL 2.5 connect to DCL 2.5 and send the newly created situation in common configuration format.
3. DCL 2.5 receives the situation configuration and respond with acknowledgement message.
4. Domain expert creates all associate rules and apply following steps;
  • shares situation configuration with SCL 2.5
  • performs step 2-5 of Transfer Production Rules by transferring the production rule(s) to SCL 2.5.</td></tr>
<tr><td>Alternative Flows:</td><td colspan="3">N/A</td></tr>
<tr><td>Exceptions:</td><td colspan="3">2a. KCL 2.5 unable to connect to SCL 2.5
  a. KCL 2.5 connection is failed during sharing production rule(s)
  b. KCL 2.5 hold and will retry after sometime to connect to SCL 2.5 and transfer the production rule(s)</td></tr>
<tr><td>Includes:</td><td colspan="3">Transfer Production Rules</td></tr>
<tr><td>Frequency of</td><td colspan="3">Invoked per situation creation</td></tr>
</table>

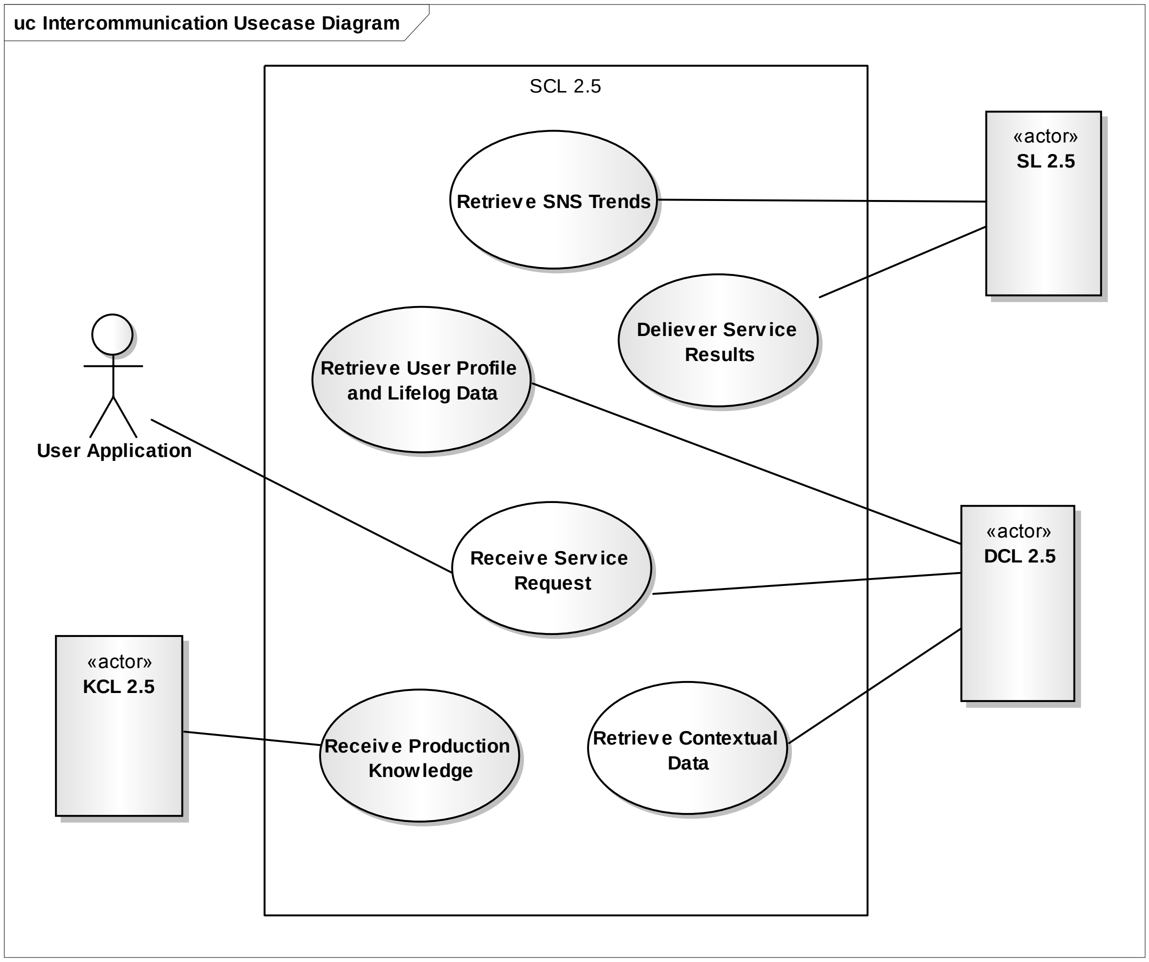
Use:	
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	Common configuration format is challenging task.
Sequence Diagram:	

Service Curation Layer ver. 2.5 (SCL 2.5)

Use case List

Use case ID#	Name
SCL2.5-UC-01	Receive service request from user application or mining mind generated events
SCL2.5-UC-02	Retrieve user profile and life-log data from intermediate database for reasoning
SCL2.5-UC-03	Retrieve Production Knowledge from the knowledge base of knowledge curation layer
SCL2.5-UC-04	Retrieve contextual data from intermediate database for interpretations
SCL2.5-UC-05	Retrieve SNS trends supporting layer through information gateway
SCL2.5-UC-06	Deliver service results to service requester and to intermediate database for persistence

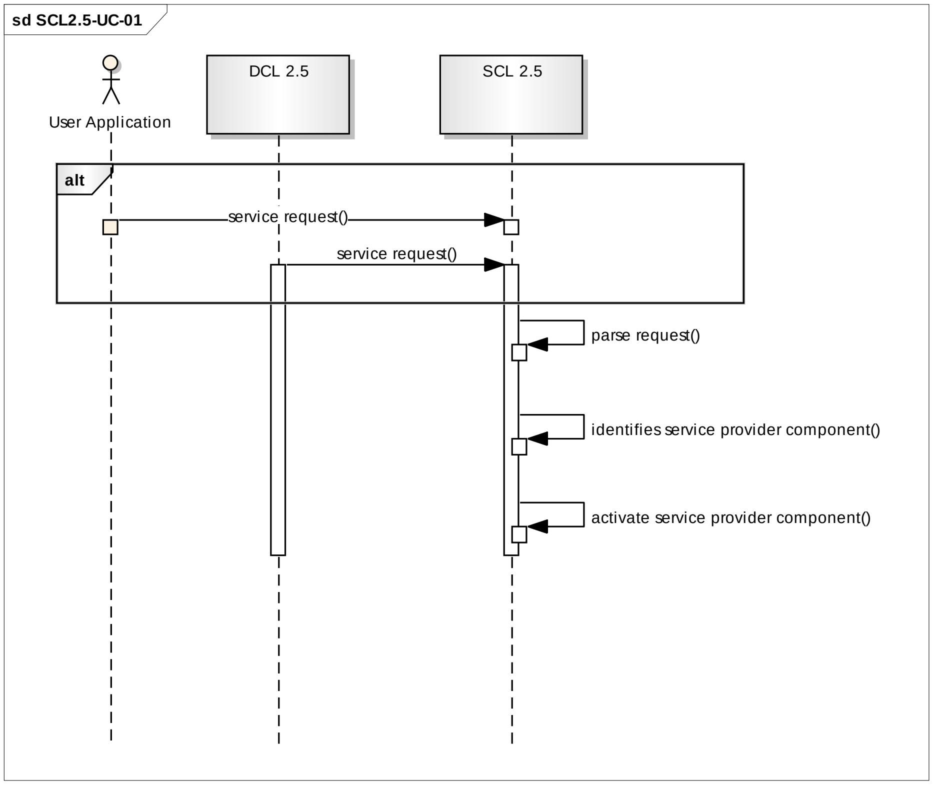
Use case Diagram



Detailed Use cases

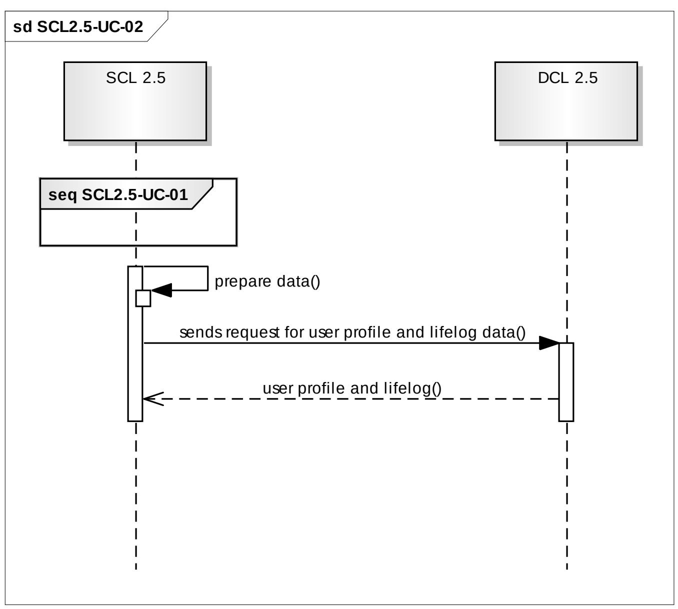
Use Case ID:	*SCL2.5-UC-01*		
Use Case Name:	Receive service request from user application or mining mind generated events		
FR ID:	MM-FR-19		
Created By:	Muhammad Afzal	**Last Updated By:**	Muhammad Afzal
Date Created:	12 April 2016	**Last Revision Date:**	15 April 2016
Actors:	User Application, DCL 2.5		
Description:	SCL 2.5 allows to receive request from the user application, or DCL 2.5 for a service. SCL 2.5 parses the request and invokes the required service hosted by Mining Mind to respond.		
Trigger:	At the time of a request from the user application, or from mining mind generated events		
Pre-conditions:	User is registered with Mining Mind		
Post-conditions:	The request is received and handled by SCL 2.5		
Normal Flow:	1. SCL 2.5 receives the service request from user application 2. SCL 2.5 parses the request 3. SCL 2.5 identifies the service requirements 4. SCL 2.5 passes the message to invoked the service in SCL 2.5		
Alternative Flows:	1.a SCL 2.5 receives the request as an interrupt from DCL 2.5 whenever a situation occurs 4.a SCL 2.5 passes the message to invoke the service in SL 2.5		
Exceptions:	NA		
Includes:	NA		
Frequency of Use:	Very frequent: at every service request		
NFR ID:	MM-NFR-09		
Assumptions:	SCL 2.5 and SL 2.5/DCL 2.5 contract is agreed.		

Notes and Issues:	NA
Sequence Diagram:	



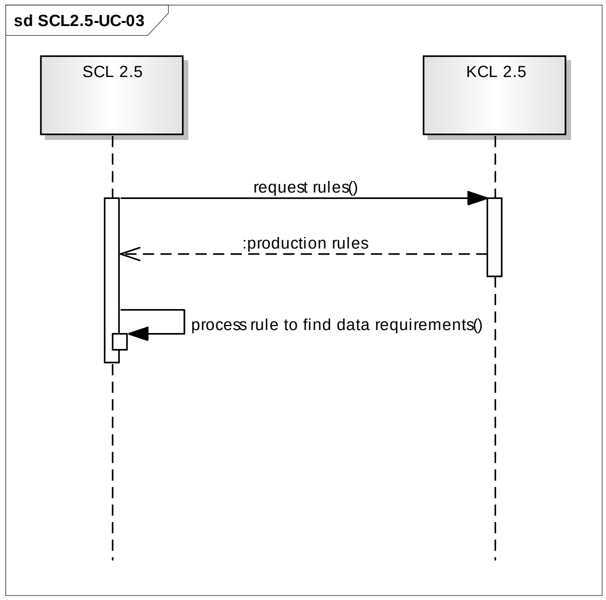
Use Case ID:	*SCL2.5-UC-02*		
Use Case Name:	Retrieve user profile/life-log data from user life-log		
FR ID:	MM-FR-20		
Created By:	Muhammad Afzal	**Last Updated By:**	Muhammad Afzal
Date Created:	12 April 2016	**Last Revision Date:**	15 April 2016
Actors:	SCL 2.5, DCL 2.5		
Description:	Retrieving user profile and life-log data is required for reasoning over knowledge. Since the data resides in DCL 2.5, so SCL 2.5 as a primary actor originates the request to DCL 2.5 for data in order to fulfill the reasoning process.		
Trigger:	At service request time		
Pre-conditions:	User profile and life-log data is available in user life-log		

Post-conditions:	User profile and life-log data is successfully retrieved and readily available for SCL 2.5 to process.
Normal Flow:	1. SCL 2.5 prepares the data request 2. SCL 2.5 sends the request to DCL 2.5 for user profile and life-log data as per the service contract 3. SCL 2.5 receives the response and make it part of SCL 2.5 internal process
Alternative Flows:	NA
Exceptions:	NA
Includes:	SCL2.5-UC-01
Frequency of Use:	Very frequent: at every service request
NFR ID:	MM-NFR-09
Assumptions:	SCL 2.5 and DCL 2.5 contract is agreed.
Notes and Issues:	
Sequence Diagram:	



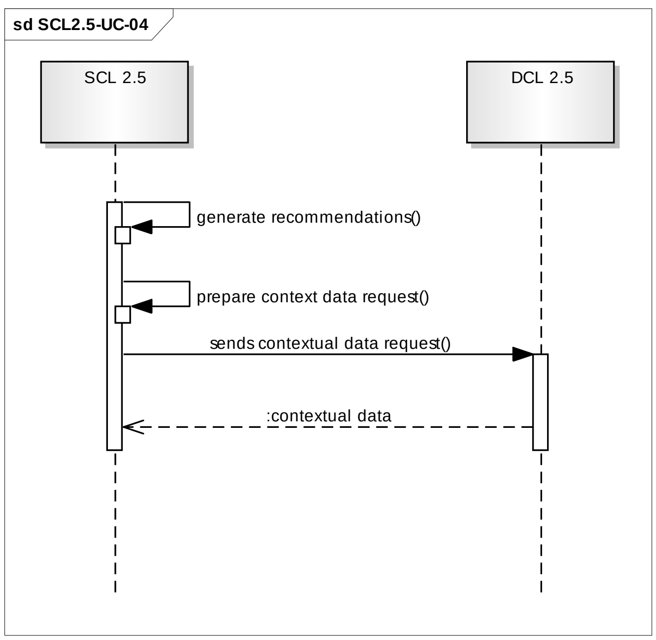
Use Case ID:	*SCL2.5-UC-03*
Use Case Name:	Receive Production Knowledge from knowledge curation layer

FR ID:	MM-FR-21		
Created By:	Muhammad Afzal	**Last Updated By:**	Muhammad Afzal
Date Created:	12 April 2016	**Last Revision Date:**	15 April 2016
Actors:	KCL 2.5, SCL 2.5		
Description:	The knowledge is originated by KCL 2.5 and is transfer to SCL 2.5 to keep a local copy of the production knowledge.		
Trigger:	At knowledge creation/update time		
Pre-conditions:	SCL 2.5 and KCL 2.5 has a common representation agreement		
Post-conditions:	The SCL 2.5 copy of knowledge is updated and is synchronized with KCL 2.5		
Normal Flow:	1. SCL 2.5 receives interrupt from KCL 2.5 2. SCL 2.5 verifies the knowledge 3. SCL 2.5 make a local of the received knowledge		
Alternative Flows:	NA		
Exceptions:	NA		
Includes:	NA		
Frequency of Use:	Less frequent: at knowledge creation/update time		
NFR ID:	MM-NFR-09		
Assumptions:	NA		
Notes and Issues:	NA		
Sequence Diagram:			



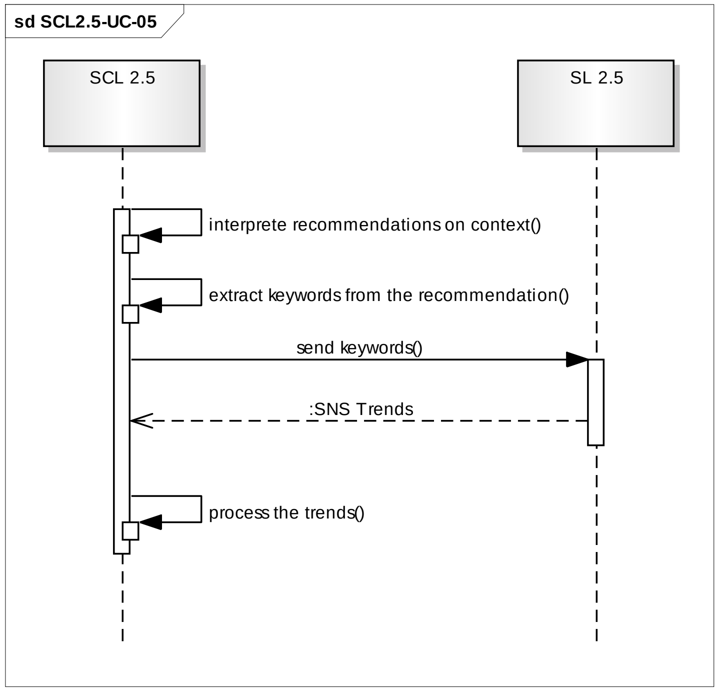
Use Case ID:	SCL2.5-UC-04		
Use Case Name:	Retrieve contextual data from intermediate database		
FR ID:	MM-FR-20		
Created By:	Muhammad Afzal	**Last Updated By:**	Muhammad Afzal
Date Created:	12 April 2016	**Last Revision Date:**	15 April 2016
Actors:	SCL 2.5, DCL 2.5		
Description:	Retrieving contextual data is required by Recommendation Interpreter component to perform its interpretations. Since the data resides in DCL 2.5, so SCL 2.5 as a primary actor originates the request to DCL 2.5 for data in order to fulfill the interpretation process.		
Trigger:	After the generation of recommendations		
Pre-conditions:	Recommendation has been generated.		
Post-conditions:	Contextual information is successfully retrieved and is readily available for SCL 2.5 to process.		
Normal Flow:	1. SCL 2.5 generates the recommendations 2. SCL 2.5 prepare the information request 3. SCL 2.5 sends the request to DCL 2.5 for		

	contextual data 4. SCL 2.5 receives the response and make it part of SCL internal process
Alternative Flows:	NA
Exceptions:	NA
Includes:	NA
Frequency of Use:	Very frequent: every recommendation service
NFR ID:	MM-NFR-09
Assumptions:	SCL and DCL service contract is already agreed.
Notes and Issues:	NA
Sequence Diagram:	



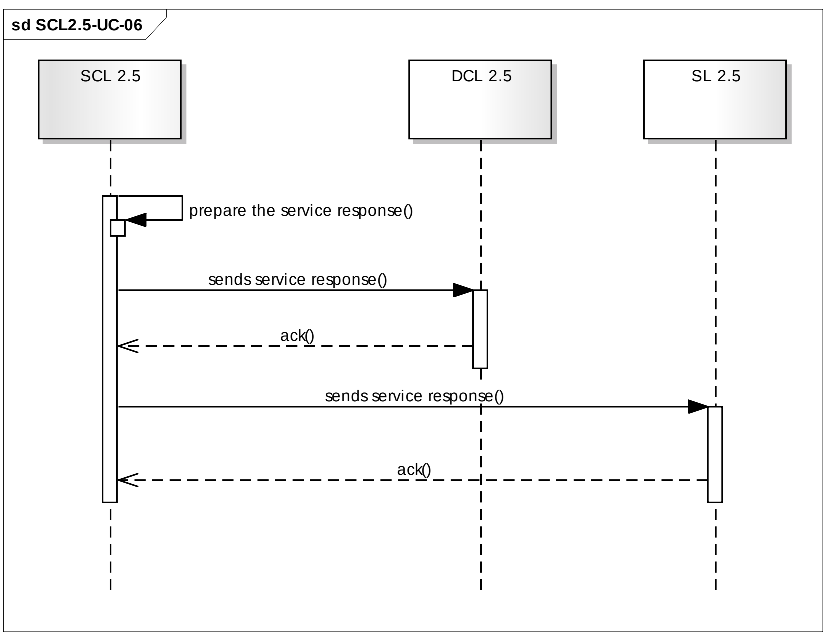
Use Case ID:	*SCL2.5-UC-05*		
Use Case Name:	Retrieve SNS Trends from Supporting Layer		
FR ID:	MM-FR-28		
Created By:	Muhammad Afzal	**Last Updated By:**	Muhammad Afzal
Date Created:	12 April 2016	**Last Revision Date:**	15 April 2016
Actors:	SCL 2.5, SL 2.5		

Description:	Retrieving SNS trends is required by Recommendation Interpreter component to identify the top trends in order to extend the recommendation with SNS trends. Since SL 2.5 is connected to outer world SNS service, so SCL 2.5 as a primary actor originates the request to SL 2.5 for social trends data in order to fulfill the SNS trends requirements.
Trigger:	After contextual interpretation of recommendations
Pre-conditions:	Recommendation has been interpreted.
Post-conditions:	SNS trends data is successfully retrieved and is readily available for SCL 2.5 to process.
Normal Flow:	1. SCL 2.5 interprets the recommendations 2. SCL 2.5 prepare the SNS trends request 3. SCL 2.5 sends the request to SL 2.5 for SNS trends data 4. SCL 2.5 receives the response and make it part of SCL 2.5 internal process.
Alternative Flows:	NA
Exceptions:	NA
Includes:	NA
Frequency of Use:	Very frequent: every recommendation service
NFR ID:	
Assumptions:	SCL 2.5 and SL 2.5 service contract is already agreed.
Notes and Issues:	NA
Sequence Diagram:	



Use Case ID:	SCL2.5-UC-06		
Use Case Name:	Deliver service results to service requester and to intermediate database for persistence		
FR ID:	MM-FR-23		
Created By:	Muhammad Afzal	**Last Updated By:**	Muhammad Afzal
Date Created:	12 April 2016	**Last Revision Date:**	15 April 2016
Actors:	SCL 2.5, SL2, DCL 2.5		
Description:	It is required to send response to the service requester. SCL prepares the recommendation and interprets according to the context and delivers to the requester.		
Trigger:	At the time of completion of interpretations		
Pre-conditions:	Service results are completed		
Post-conditions:	Service results are successfully delivered in SL 2.5 and DCL 2.5		

Normal Flow:	1. SCL 2.5 prepares the recommendation output 2. SCL 2.5 sends the output to DCL for storage 3. SCL 2.5 receives the acknowledgement 4. SCL 2.5 sends the output to SL 5. SCL 2.5 receives the acknowledgement
Alternative Flows:	NA
Exceptions:	NA
Includes:	NA
Frequency of Use:	Very frequent: at every service request
NFR ID:	MM-NFR-09
Assumptions:	SCL 2.5 has the agreed contract with SL 2.5 and DCL 2.5
Notes and Issues:	NA
Sequence Diagram:	

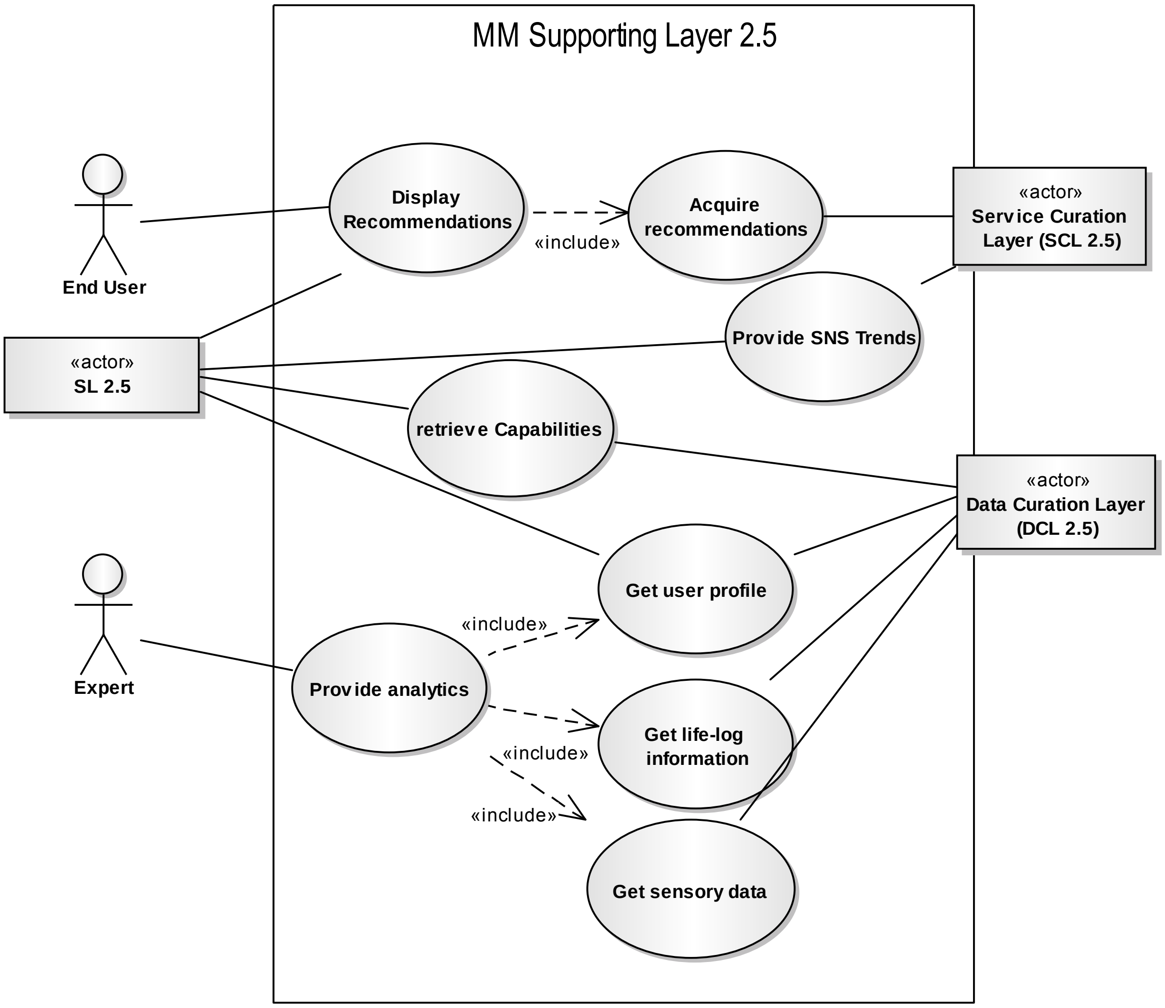


Supporting Layer ver. 2.5 (SL 2.5)

Use case List

Use case ID#	Description
SL2.5-UC-01	Acquire Recommendations for displaying to end user
SL2.5-UC-02	Retrieve Capabilities for user interface adaption
SL2.5-UC-03	Collect user profile data for user satisfaction
SL2.5-UC-04	Get user profile, life-log and sensory data for analytics
SL2.5-UC-05	Get SNS trends from Tapacross to SCL
SL2.5-UC-06	Receive user credentials and verifies
SL2.5-UC-07	Receive user queries over authorized data
SL2.5-UC-08	Write data
SL2.5-UC-09	Inference Detector (The system shall maintain query logs to analyze the pattern of user accessing the system)
SL2.5-UC-10	Integrity Check for tempered recommendations

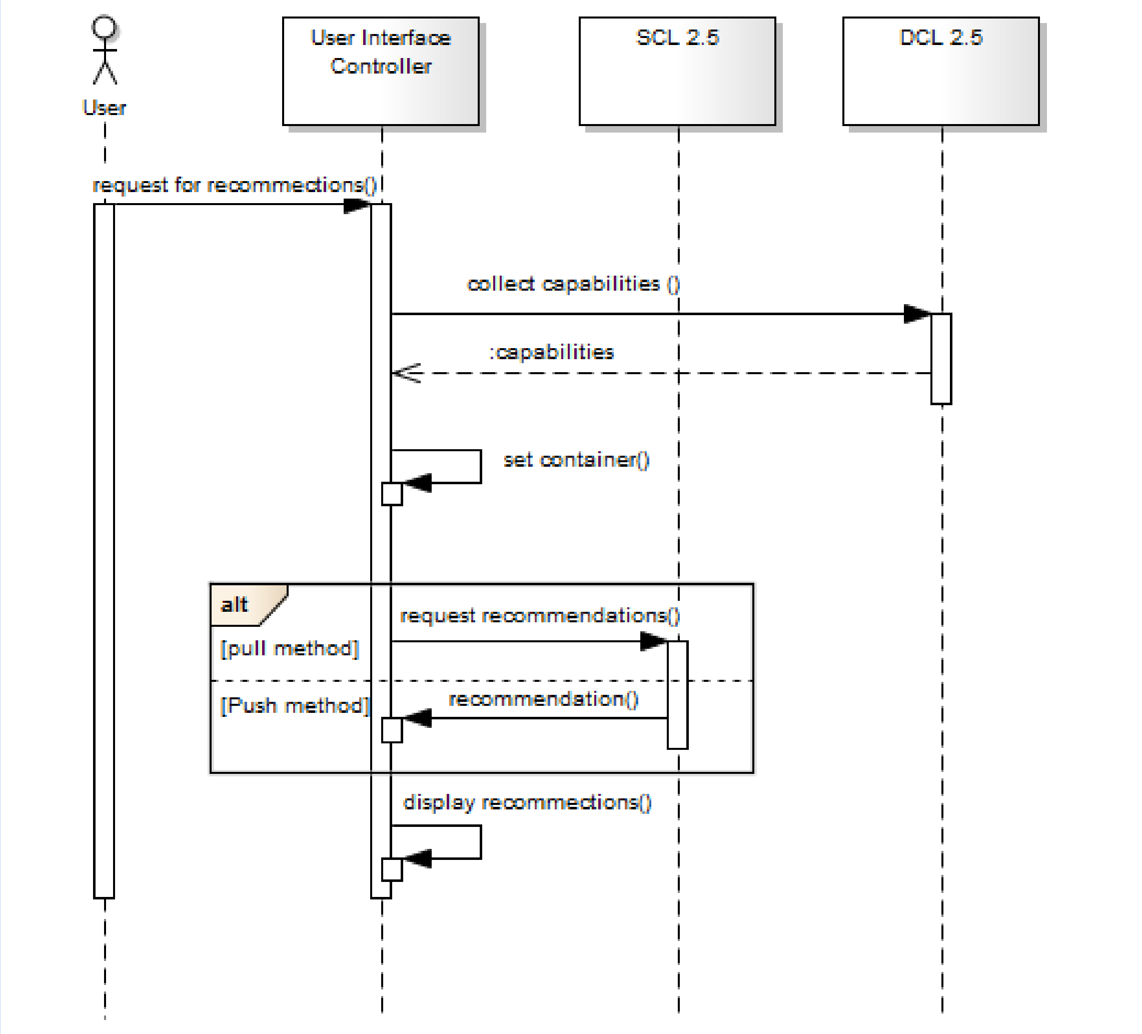
Use case Diagram



Detailed Use cases

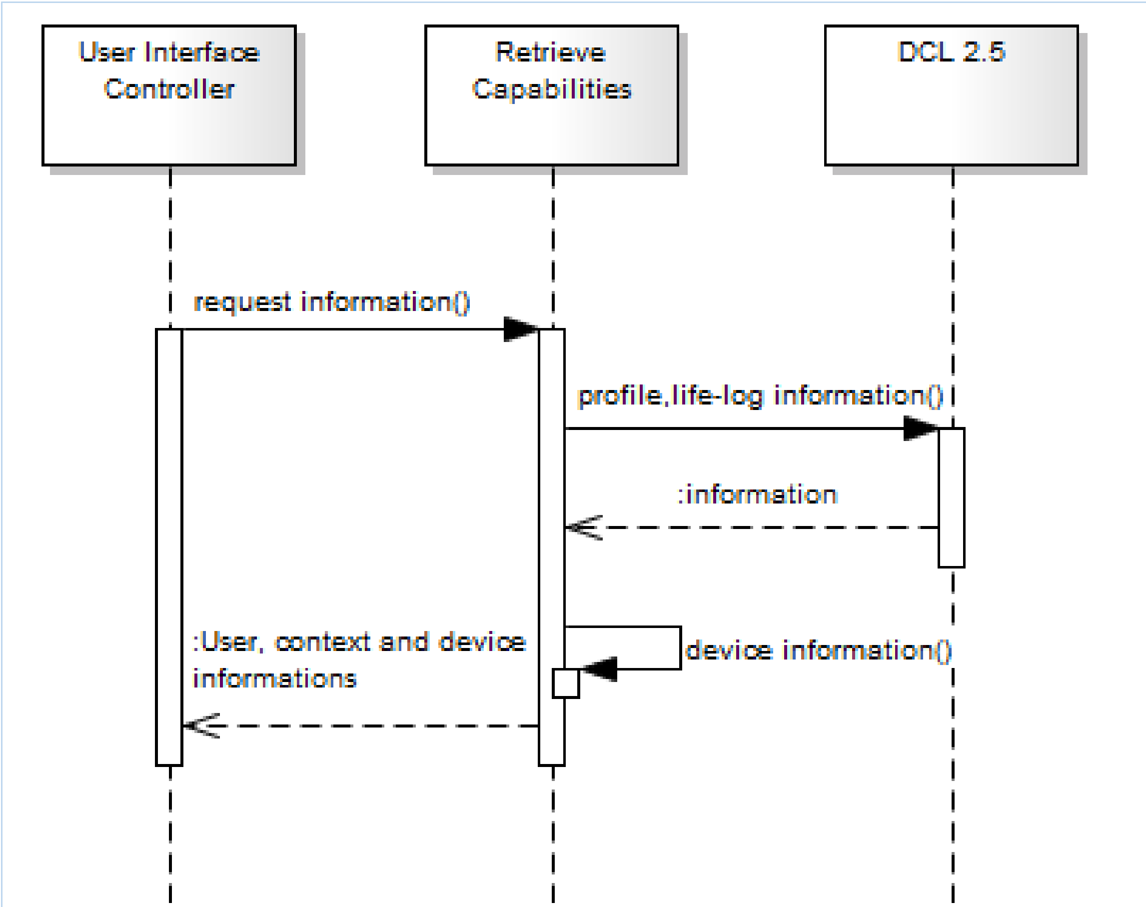
Use Case ID:	*SL2.5-UC-01*		
Use Case Name:	Acquire Recommendations for displaying to end user		
FR ID:	MM-FR-24		
Created By:	Jamil Hussain	**Last Updated By:**	Jamil Hussain
Date Created:	06 July 2015	**Last Revision Date:**	15 April 2016
Actors:	End-user, SCL 2.5, DCL 2.5		
Description:	This use case collects the recommendations generated by SCL 2.5 and displays it on the user interface for the end users. The provided recommendations are displayed according to user capabilities, context of use, and device characteristics. This information is obtained from the DCL 2.5.		
Trigger:	SCL 2.5 push the recommendations to the App or end-user send request for recommendations		
Pre-conditions:	End-user subscribes to particular services		
Post-conditions:	All recommendations are successfully displayed according to user capabilities, context, and device characteristics.		
Normal Flow:	1. SCL 2.5 generate the recommendations and provide it to user interface 2. The SCL 2.5 recommendations are acquired by the SL 2.5 3. SL 2.5 investigates the user capabilities, context of use, and device characteristics by obtaining from DCL 2.5 4. The recommendation are displayed in graphical user interface based on collected capabilities of user, context and device information.		
Alternative Flows:	2a. In step 2. The SCL 2.5 recommendations are acquired by the SL 2.5 1. user request for recommendations (pull method)		

	2b. In step 2. The SCL 2.5 recommendations are acquired by the SL 1. SL 2.5 push recommendations to App based on situations
Exceptions:	2a. In step 2 of the normal flow, if the user is detected to be un-authorized 1. DCL 2.5 destroys the received sensory data and environmental variables 2. Un-authorized user message is sent to the raw sensory data source
Includes:	NA
Frequency of Use:	Less Frequent: whenever the recommendations are generated by SCL 2.5
NFR ID:	MM-NFR-10
Assumptions:	The user profile data and context information should exist in the DCL 2.5
Notes and Issues:	NA
Sequence Diagram:	



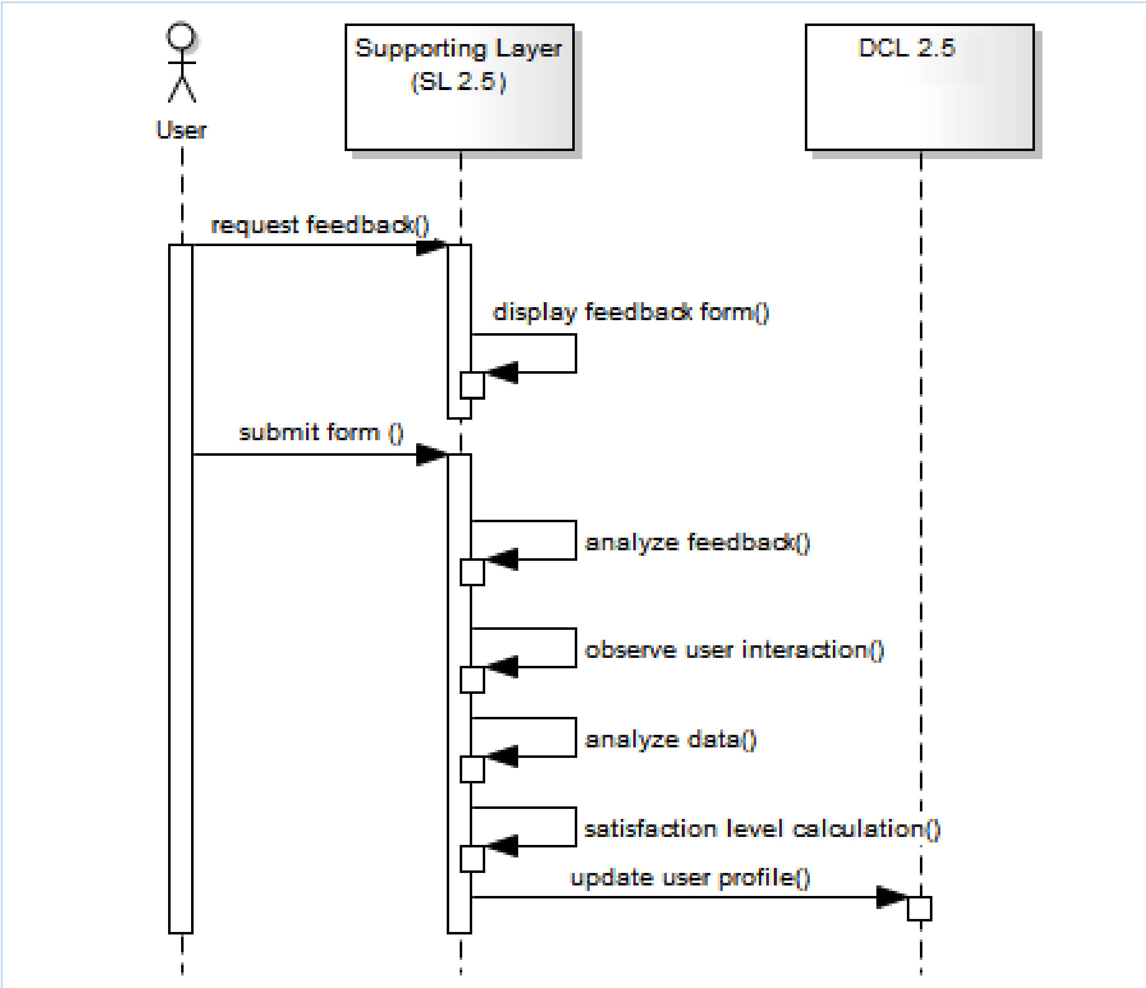
Use Case ID:	***SL2.5-UC-02***		
Use Case Name:	Retrieve Capabilities for user interface adaptation		
FR ID:	MM-FR-25		
Created By:	Jamil Hussain	**Last Updated By:**	Jamil Hussain
Date Created:	06 July 2015	**Last Revision Date:**	15 April 2016
Actors:	SL 2.5, DCL 2.5		
Description:	This use case focuses on the retrieval of the capabilities for user interface adaptation. The capabilities include user profile information, context information and device information. It is utilized for adaptation based on changes or observational data.		

Trigger:	SL
Pre-conditions:	The DCL 2.5 provide the access to required information
Post-conditions:	All required capabilities are successfully collected.
Normal Flow:	1. SL 2.5 generates request for user, device, and context information collection from DCL 2.5 1. This information is utilized for the adaptation of the user interfaces 2. The adaptation is based on changes in user profile, context information or collected observational data
Alternative Flows:	NA
Exceptions:	If there is not capabilities information then the default user interfaces will be displayed.
Includes:	NA
Frequency of Use:	Very Frequent. always when the application is running
NFR ID:	NA
Assumptions:	The capabilities information should be available with the DCL 2.5.
Notes and Issues:	N/A
Sequence Diagram:	



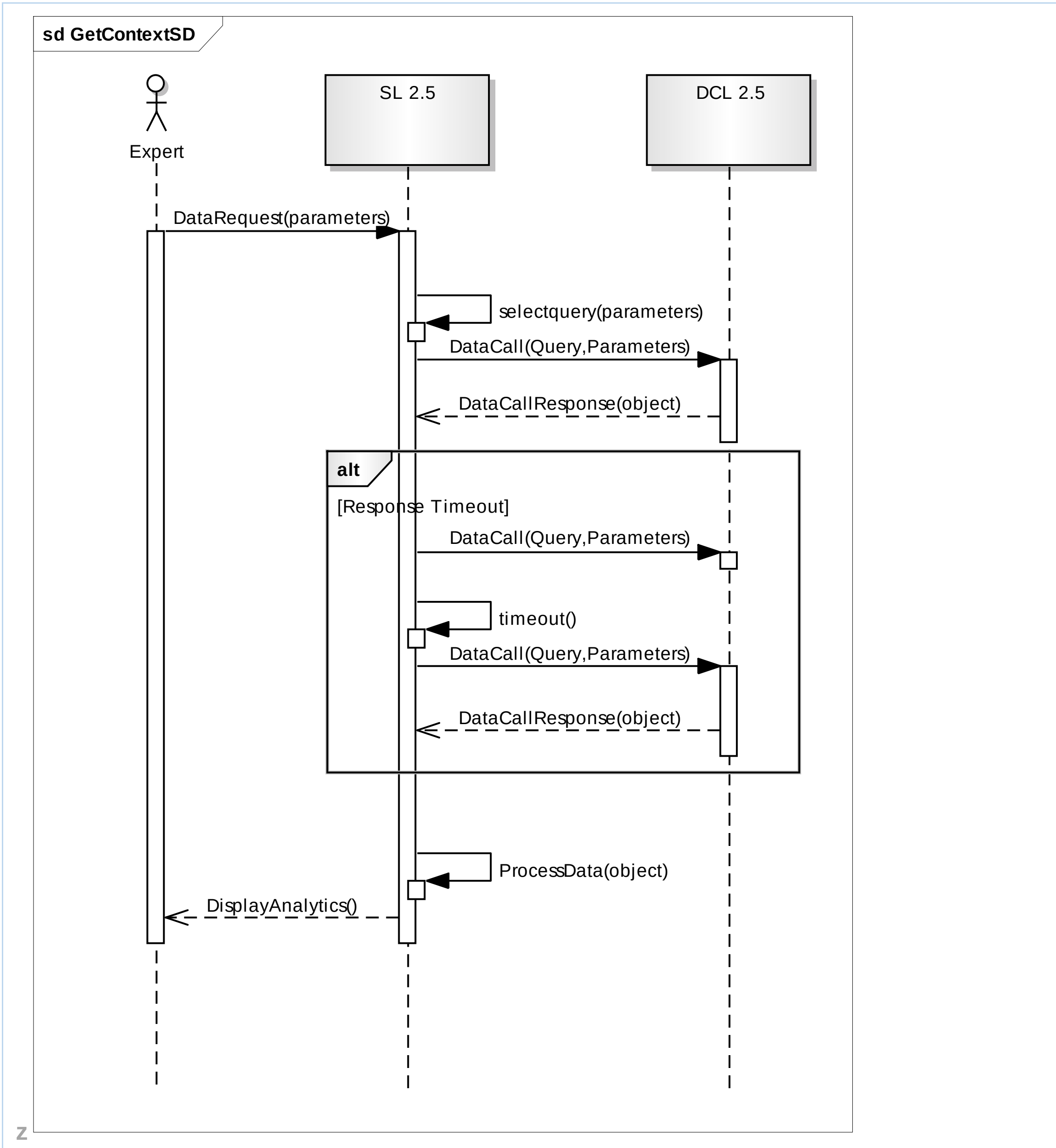
Use Case ID:	*SL2.5-UC-03*		
Use Case Name:	Collect user profile data for user satisfaction		
FR ID:	MM-FR-26		
Created By:	Jamil Hussain	**Last Updated By:**	Jamil Hussain
Date Created:	06 July 2015	**Last Revision Date:**	15 April 2016
Actors:	SL 2.5, DCL 2.5		
Description:	This use case focuses on the collection of user data such as user feedback and user observational data for enhancement of user interface. The user satisfaction level is calculated based on collected information for adaptability of user interface. The whole process finds the user experience based on the user satisfaction.		
Trigger:	End-user, SL 2.5		

Pre-conditions:	User should be login and observational tracker are installed
Post-conditions:	User profile update successfully.
Normal Flow:	1. The user utilizes user interfaces and enters feedback as response 2. The SL sends the feedback collected for user interaction analysis to DCL for persistence 3. SL analyzes the feedback stored in DCL 4. SL collects the observational data based on user interaction 5. The collected observational data is analyzed for user satisfaction calculation 6. SL sends the user profile information to DCL for persistence
Alternative Flows:	NA
Exceptions:	If the feedback fails then acknowledgment message shall be displayed to user.
Includes:	NA
Frequency of Use:	1. Very Frequent, When the user interacts with the system 2. Less Frequent, When the user wants to provide feedback on particular services
NFR ID:	NA
Assumptions:	NA
Notes and Issues:	NA
Sequence Diagram:	



Use Case ID:	*SL2.5-UC-04*		
Use Case Name:	Get user profile, life-log and sensory data for analytics		
FR ID:	MM-FR-27		
Created By:	Shujaat Hussain	**Last Updated By:**	Shujaat Hussain
Date Created:	06 July 2015	**Last Revision Date:**	16 April 2016
Actors:	Domain Expert, DCL 2.5		
Description:	This use case focuses on getting the user profile, life-log and raw sensory data from DCL for processing to show analytics.		
Trigger:	The request from the expert panel for analytics		
Pre-conditions:	A predefined query library for retrieving the big data		

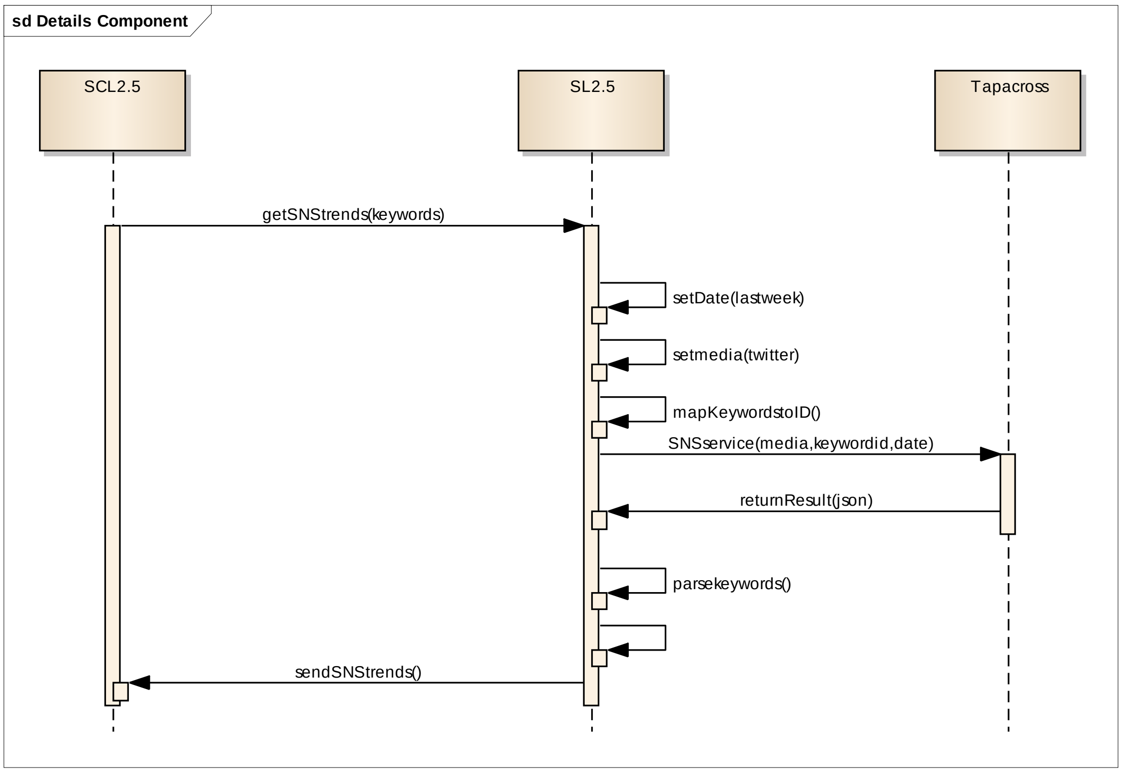
Post-conditions:	1. The data reaches the expert panel in a specific format for processing. 2. Facts and analytics are displayed based on the data
Normal Flow:	1. The expert requests the analytics for a specific context. 2. Query is sent through the restful web service to the data curation layer. 3. The social and personal data is obtained from the Data Curation Layer 4. After performing analytics the transformed data is provided to the experts
Alternative Flows:	2a. In step 2 of the normal flow, if there is delay in response time 1. The timeout message is occurred on the web interface 2. The request is again sent and normal flow resumes from step 1.
Exceptions:	NA
Includes:	NA
Frequency of Use:	Less Frequent, This use case can be used by the domain expert about 5-10 times based on the volume of data
NFR ID:	MM-NFR-11
Assumptions:	Queries are pre-built
Notes and Issues:	1. What is the maximum response time acceptable? 2. How many queries are there in the query library?
Sequence Diagram:	


sd GetContextSD
Expert
SL 2.5
DCL 2.5
DataRequest(parameters)
selectquery(parameters)
DataCall(Query,Parameters)
DataCallResponse(object)
alt
[Response Timeout]
DataCall(Query,Parameters)
timeout()
DataCall(Query,Parameters)
DataCallResponse(object)
ProcessData(object)
DisplayAnalytics()

Use Case ID:	*SL2.5-UC-05*		
Use Case Name:	Get SNS trends from Tapacross to SCL 2.5		
FR ID:	MM-FR-28		
Created By:	Shujaat Hussain	**Last Updated By:**	Shujaat Hussain
Date Created:	14 April 2015	**Last Revision Date:**	14 April 2015
Actors:	SL2.5, SCL 2.5		
Description:	This use case focuses on getting the SNS trends from SCL 2.5 to SL 2.5		
Trigger:	The request from the SCL 2.5 will be given to SL 2.5		
Pre-conditions:	A keyword is sent from SCL 2.5 to SL 2.5		
Post-conditions:	1. SCL 2.5 gets the SNS trends of the keyword categories with the trend value.		
Normal Flow:	1. SCL 2.5 sends keyword to the SL 2.5 communication gateway. 2. The gateway then calls the SNS connector to get the SNS trends from tapascross 3. Tapacross sends the trends according to the data and keywords 4. The keywords and the trends are sent back to SCL 2.5 from SL 2.5		
Alternative Flows:	2a. In step 2 of the normal flow, if there is delay in response time 1. The timeout message is occurred on the web interface 2. The request is again sent and normal flow resumes from step 1.		
Exceptions:	NA		
Includes:	NA		
Frequency of Use:	Less Frequent, This use case can be used by the domain expert about 5-10 times based on the volume of data		

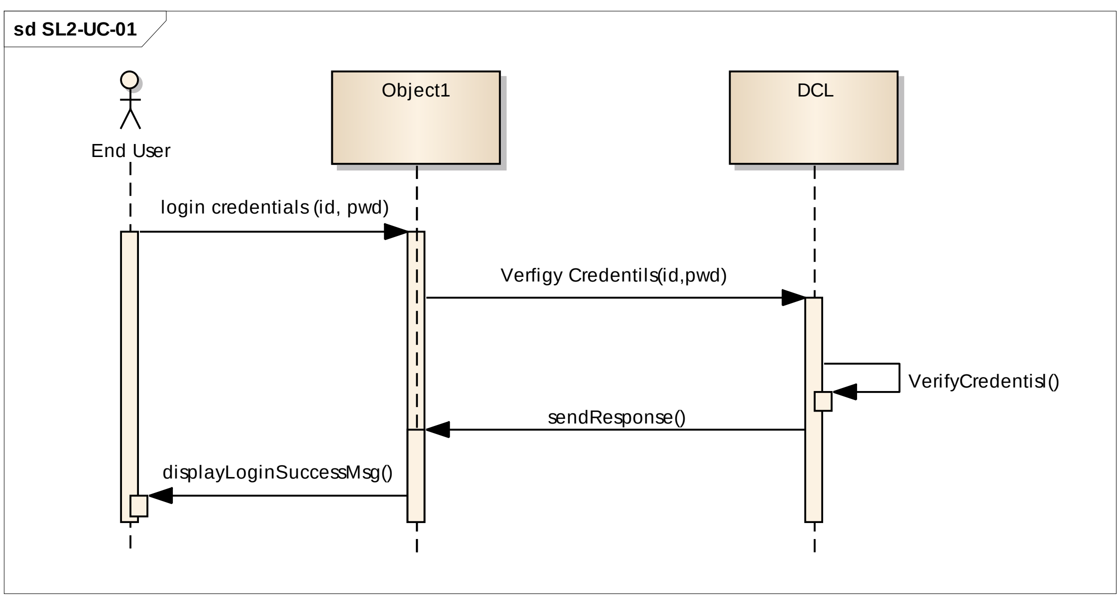
NFR ID:	
Assumptions:	Keywords are pre-defined
Notes and Issues:	1. What is the maximum response time acceptable?

Sequence Diagram:

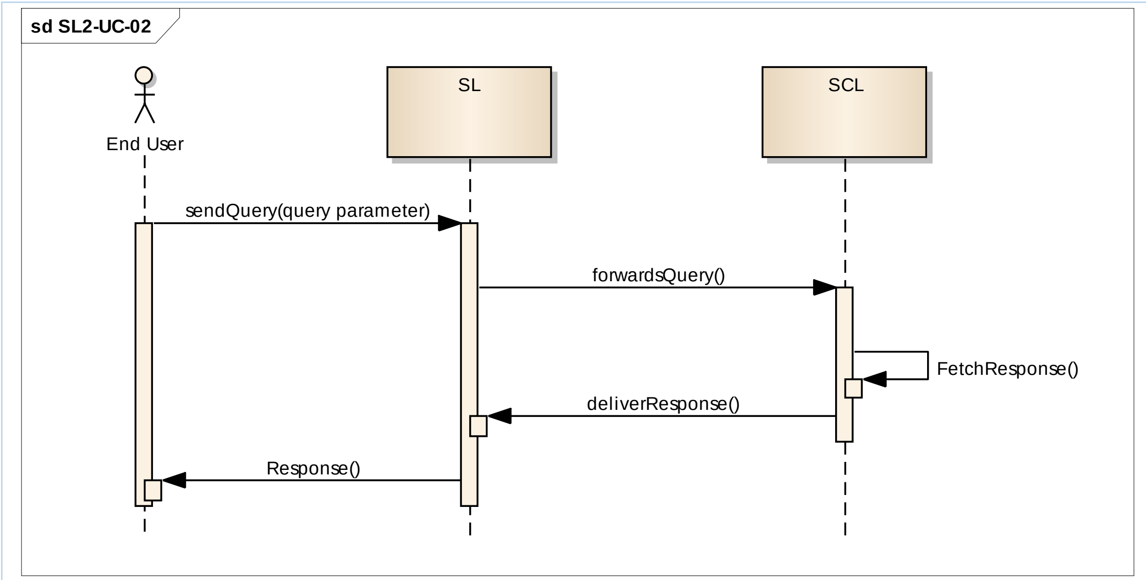


Use Case ID:	***SL2.5-UC-06***		
Use Case Name:	Receive user credentials and verifies		
FR ID:	MM-FR-01		
Created By:	Mahmood Ahmad	**Last Updated By:**	Mahmood Ahmad
Date Created:	20 Mar 2016	**Last Revision Date:**	23 Mar 2016
Actors:	User Application, DCL 2.5		
Description:	SL 2.5 allows a user to login into the MM with authorized credentials which are verified by engaging DCL 2.5		
Trigger:	User initiating a request to login into the system		

Pre-conditions:	User is registered with Mining Mind
Post-conditions:	The request is received and handled by SL 2.5
Normal Flow:	1. SL 2.5 receives the login request from user application 2. SL 2.5 verifies the credentials by passing it to the DCL 3. Dashboard appears with successful login
Alternative Flows:	NA
Exceptions:	Invalid login credentials
Includes:	Credentials verification
Frequency of Use:	User dependent
NFR ID:	MM-NFR-01, MM-NFR-02
Assumptions:	NA
Notes and Issues:	NA
Sequence Diagram:	

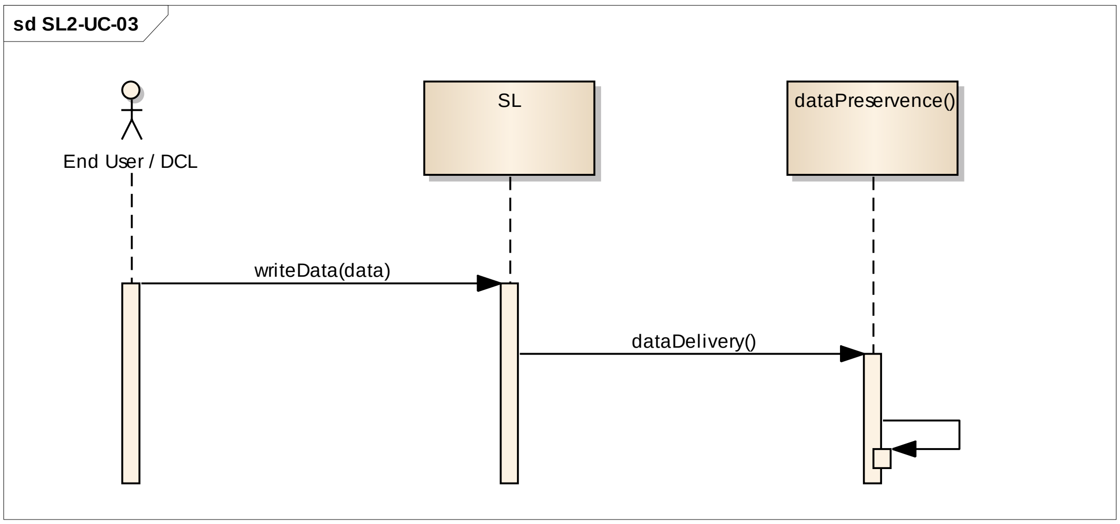


Use Case ID:	*SL2.5-UC-07*		
Use Case Name:	Receive user queries over authorized data		
FR ID:	MM-FR-02		
Created By:	Mahmood Ahmad	**Last Updated By:**	Mahmood Ahmad
Date Created:	20 Mar 2016	**Last Revision Date:**	23 Mar 2016
Actors:	User Application, SCL 2.5		
Description:	SL 2.5 allows a user to forward queries to MM after authorization by engaging SCL 2.5		
Trigger:	User initiating a query request to the system		
Pre-conditions:	Active Session (User is logged in)		
Post-conditions:	User request ends up in appropriate response		
Normal Flow:	1. SL 2.5 receives the user query 2. SL 2.5 forwards the query to SCL 2.5 3. Query response is fetched through SCL 2.5		
Alternative Flows:	NA		
Exceptions:	Query for unauthorized resource		
Includes:	NA		
Frequency of Use:	Infrequent		
NFR ID:	MM-NFR-01, MM-NFR-02		
Assumptions:	NA		
Notes and Issues:	NA		
Sequence Diagram:			



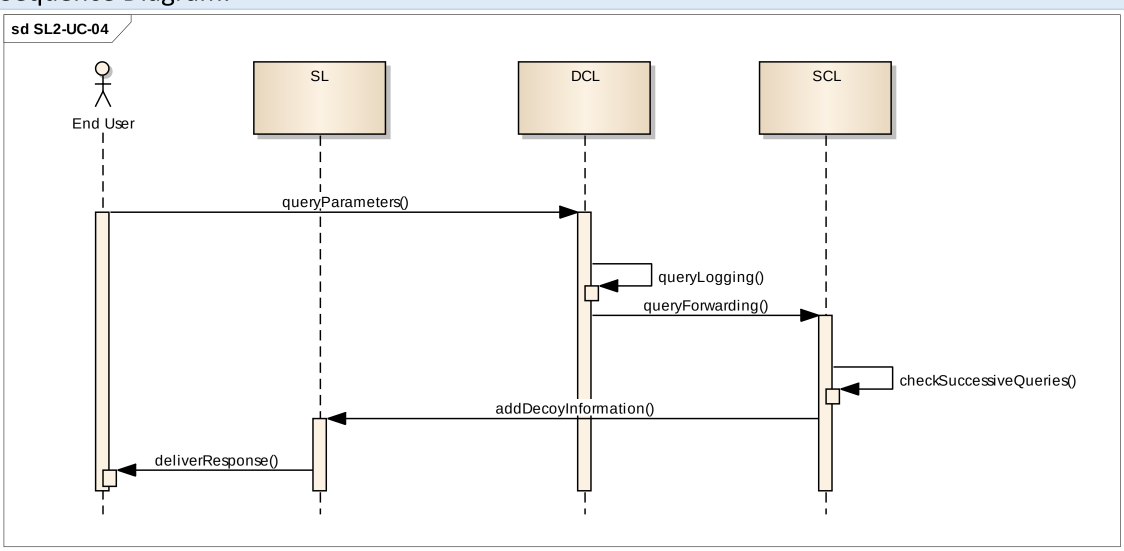
Use Case ID:	***SL2.5-UC-08***		
Use Case Name:	Write data		
FR ID:	MM-FR-03		
Created By:	Mahmood Ahmad	**Last Updated By:**	Mahmood Ahmad
Date Created:	20 Mar 2016	**Last Revision Date:**	23 Mar 2016
Actors:	User Application, DCL 2.5		
Description:	SL 2.5 allows a user and DCL 2.5 to write data into MM		
Trigger:	User data, data generating devices		
Pre-conditions:	Active Session (User is logged in)		
Post-conditions:	NA		
Normal Flow:	1. SL 2.5 receives data from user or DCL (sensory data) 2. SL 2.5 forwards the data to DCL 2.5 3. DCL perform the data preservence		
Alternative Flows:	NA		
Exceptions:	NA		
Includes:	NA		

Frequency of Use:	Frequent
NFR ID:	MM-NFR-02
Assumptions:	NA
Notes and Issues:	NA
Sequence Diagram:	



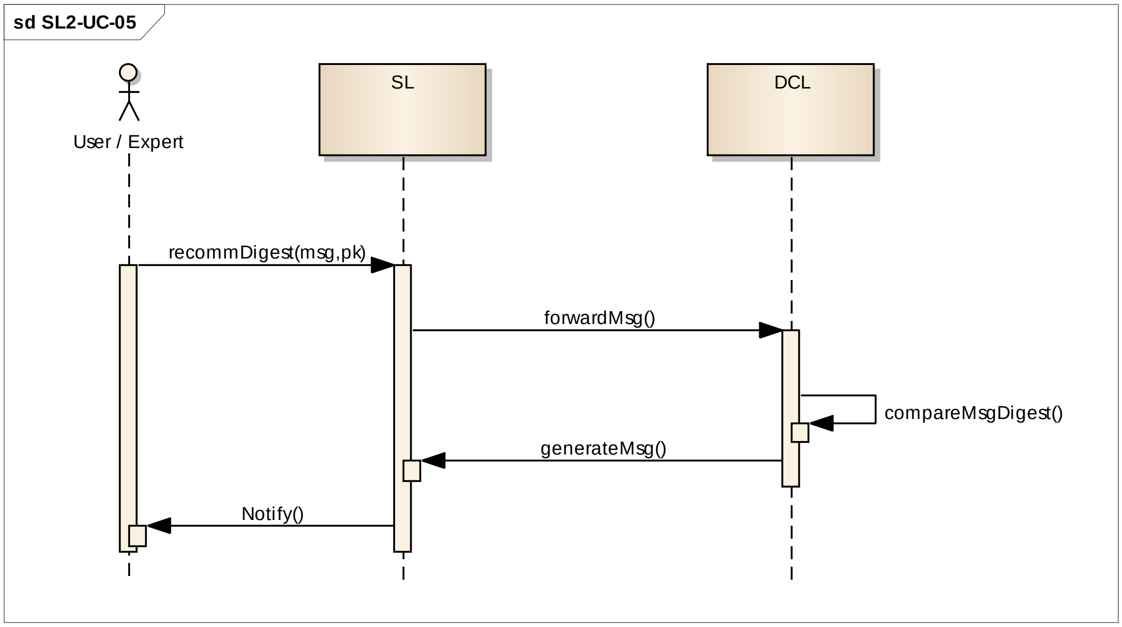
Use Case ID:	***SL2.5-UC-09***		
Use Case Name:	Inference Detector (The system shall maintain query logs to analyze the pattern of user accessing the system)		
FR ID:	MM-FR-04		
Created By:	Mahmood Ahmad	**Last Updated By:**	Mahmood Ahmad
Date Created:	20 Mar 2016	**Last Revision Date:**	23 Mar 2016
Actors:	User Application, DCL 2.5, SCL 2.5		
Description:	SL 2 preserves the individual identity against inference attacks		
Trigger:	Successive queries by a user		
Pre-conditions:	Active Session (User is logged in)		
Post-conditions:	Avoid inference attack with decoy information		

Normal Flow:	1. SL 2.5 receives user successive queries 2. Queries are logged at DCL 2.5 3. Analysis of queries and response helps SL 2.5 to activate the inclusion of decoy information into the response as a mitigation strategy against inference attack
Alternative Flows:	NA
Exceptions:	NA
Includes:	NA
Frequency of Use:	rare
NFR ID:	MM-NFR-04
Assumptions:	NA
Notes and Issues:	NA
Sequence Diagram:	

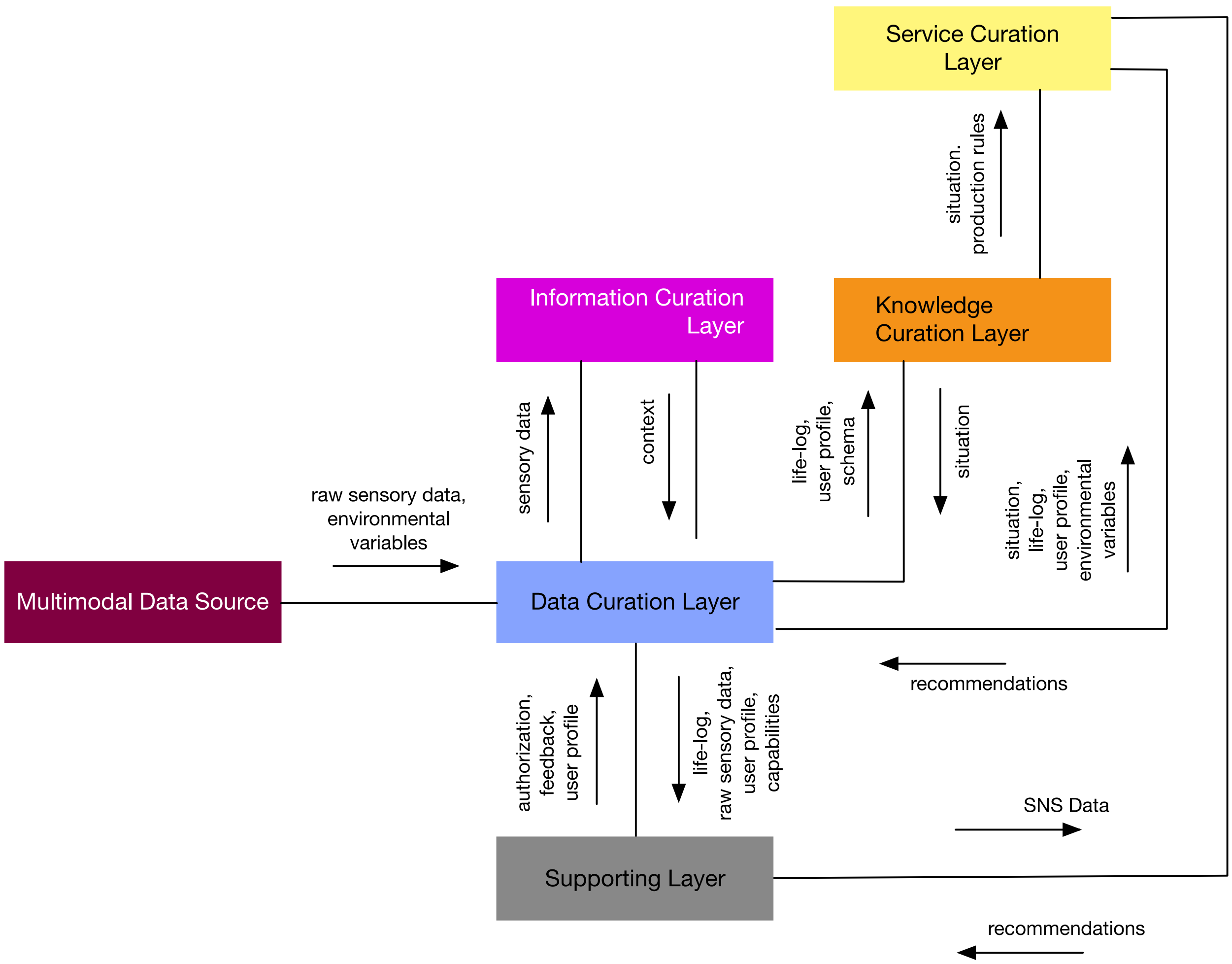


Use Case ID:	SL2.5-UC-10		
Use Case Name:	Integrity Check for tempered recommendations		
FR ID:	MM-FR-05		
Created By:	Mahmood Ahmad	**Last Updated By:**	Mahmood Ahmad
Date Created:	20 Mar 2016	**Last Revision Date:**	23 Mar 2016
Actors:	User Application, DCL 2		
Description:	The recommendations generated by an expert and the recommendations that are received by an end user are compared for integrity compromise		
Trigger:	Integrity violation on recommendations (generated and received)		
Pre-conditions:	Active Session (User is logged in) and DCL has received 02 message digest from user and by an expert		
Post-conditions:	Trigger password reset		
Normal Flow:	1. SL 2 forwards the recommendations generated by an expert and received by an end user to DCL 2 respectively under the public key of MM 2. DCL 2 stores the message digest received from step 1 3. In case of digest mismatch, an appropriate message (generate new recommendations to expert, and discard recommendation to the end user) is forwarded		
Alternative Flows:	NA		
Exceptions:	NA		
Includes:	NA		
Frequency of Use:	rare		
NFR ID:	MM-NFR-05		
Assumptions:	NA		
Notes and Issues:	NA		

Sequence Diagram:



Collaboration Diagram



Section 4-B

Design Document

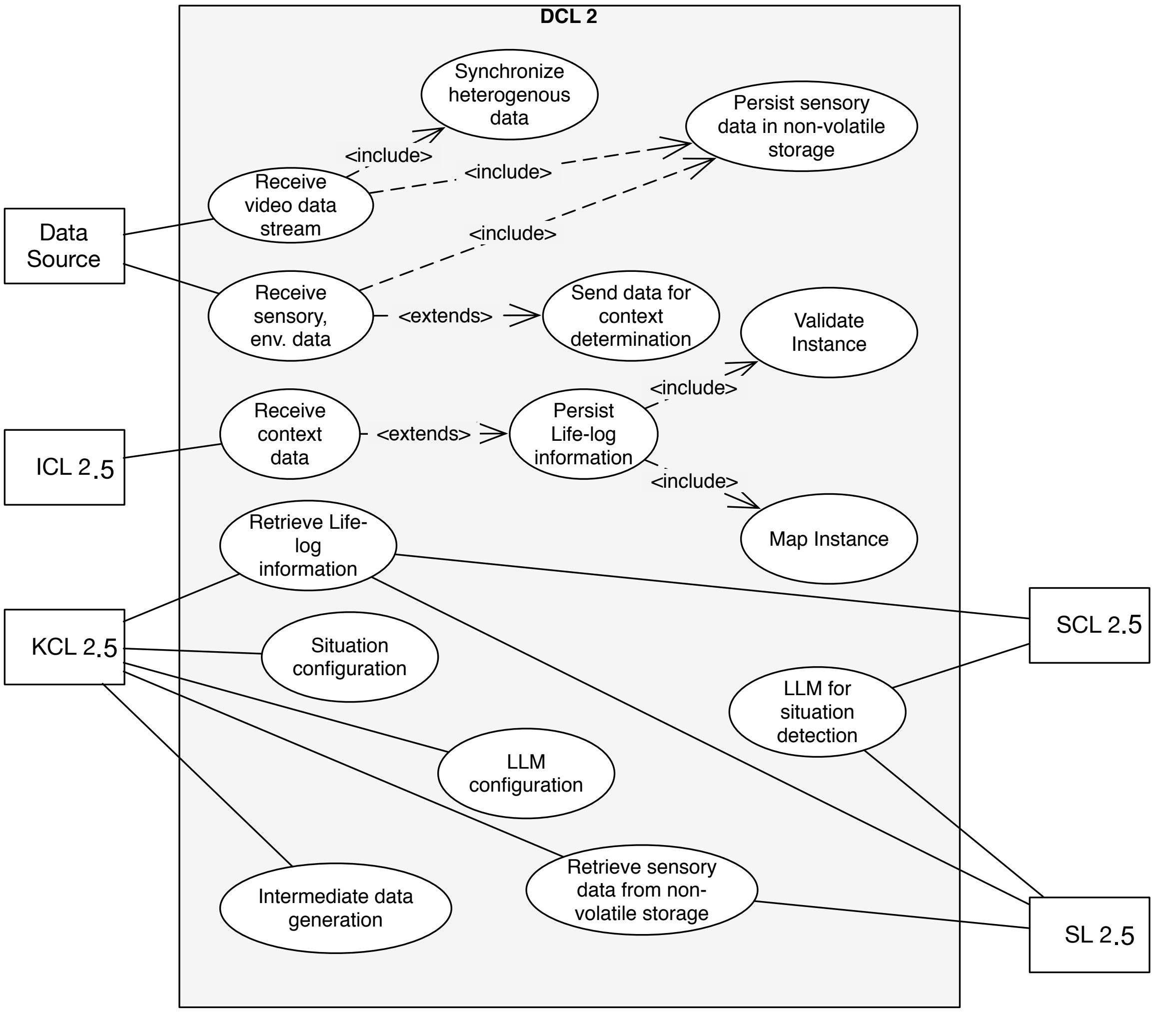
Data Curation Layer (DCL Ver. 2.5)

System Level Use cases

List of Use cases

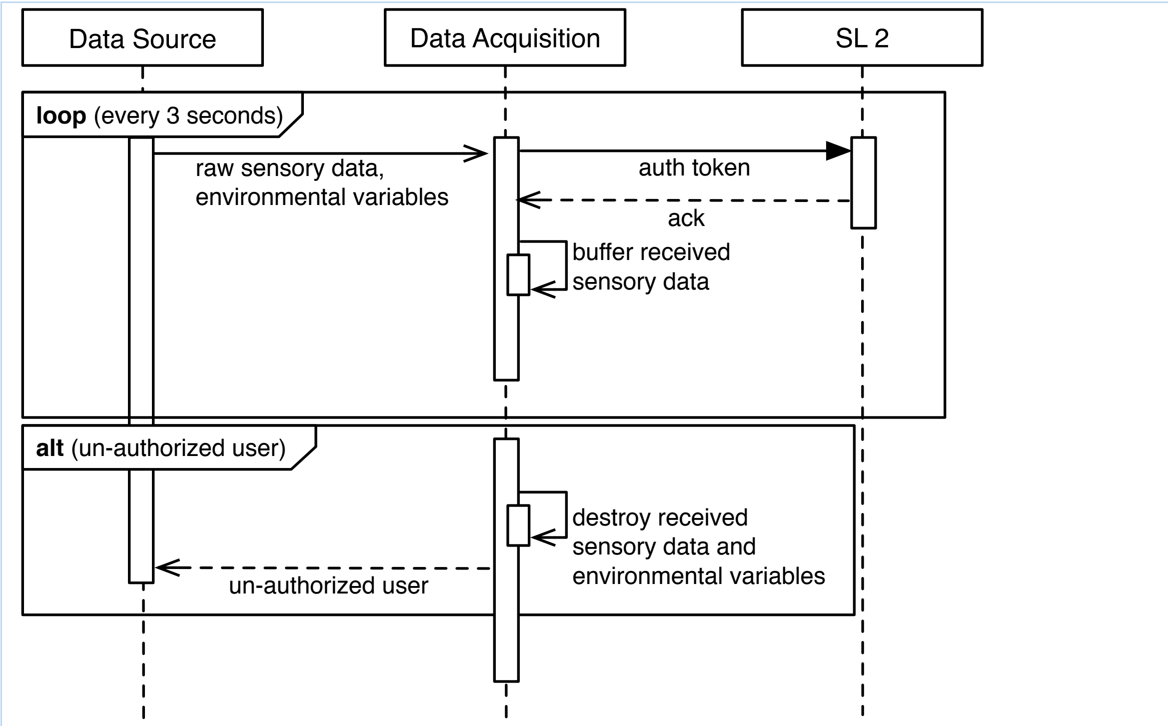
Use case ID#	Name
DCL2.5-SUC-01	Receive sensory and environmental data from data source
DCL2.5-SUC-02	Receive video data stream from data source
DCL2.5-SUC-03	Synchronize heterogeneous user data
DCL2.5-SUC-04	Send data for context determination
DCL2.5-SUC-05	Receive context data
DCL2.5-SUC-06	Retrieve Life-log Information
DCL2.5-SUC-07	Persist Life-log Information
DCL2.5-SUC-08	Map Instances
DCL2.5-SUC-09	Validate Instances
DCL2.5-SUC-10	Situation Configuration
DCL2.5-SUC-11	LLM configuration for target variables
DCL2.5-SUC-12	LLM for situation detection
DCL2.5-SUC-13	Retrieve sensory data from non-volatile storage (offline)
DCL2.5-SUC-14	Retrieve sensory data from non-volatile storage (online)
DCL2.5-SUC-15	Persist sensory data in non-volatile storage

Use case Diagram



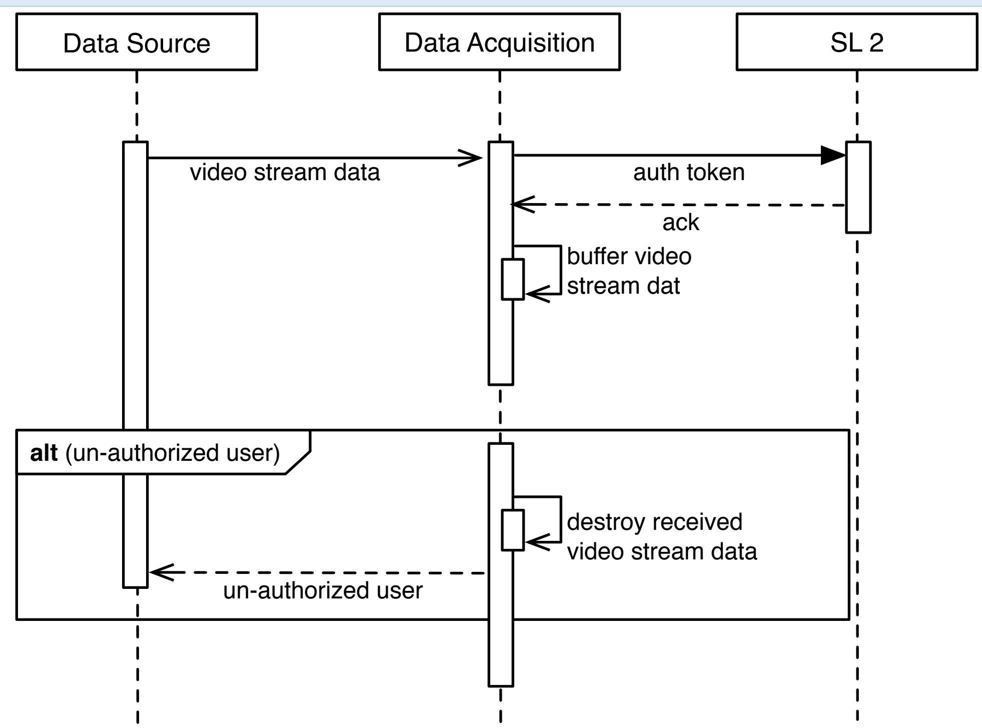
Use case Description

Use Case ID:	*DCL2.5-SUC-01*		
Use Case Name:	Receive sensory and environmental data from data source		
Created By:	Bilal Amin	**Last Updated By:**	Bilal Amin
Date Created:	15 July 2015	**Last Revision Date:**	20 Apr 2016
Actors:	Data source		
Description:	User sensory data and environmental is received and buffered from data source in real time		
Trigger:	User activity of at least 3 seconds		
Pre-conditions:	User is a registered client of MM platform		
Post-conditions:	Sensory and environmental data is persisted in the buffer		
Normal Flow:	1. Sensory and environmental data is received by a data acquisition component 2. Data source is authenticated and contents of the data are verified 3. Data is temporary buffered for context determination		
Alternative Flows:	N/A		
Exceptions:	2a. In step 2 of the normal flow, if the user is detected to be un-authorized or contents are un-verifiable 1. Data acquisition component destroys the data		
Includes:	N/A		
Frequency of Use:	Very frequent: every 3 second		
Assumptions:	Communication contract is defined between data source and data acquisition component		
Notes and Issues:	NA		
Sequence Diagram:			



Use Case ID:	*DCL2.5-SUC-02*		
Use Case Name:	Receive video data stream from data source		
Created By:	Bilal Amin	**Last Updated By:**	Bilal Amin
Date Created:	15 July 2015	**Last Revision Date:**	20 Apr 2016
Actors:	Data source		
Description:	User video data stream is received and buffered from data source in real time		
Trigger:	Video camera is streaming user feed		
Pre-conditions:	User is a registered client of MM platform		
Post-conditions:	User video data stream is persisted in the video stream buffer		
Normal Flow:	1. Video data stream is received by a data acquisition component 2. Data source is authenticated and contents of the data stream are verified 3. Video data stream is temporary buffered for context determination		

Alternative Flows:	N/A
Exceptions:	2a. In step 2 of the normal flow, if the user is detected to be un-authorized or contents are un-verifiable 2. Data acquisition component destroys the data
Includes:	N/A
Frequency of Use:	Less frequent: If video streaming based data source is available
Assumptions:	Video data streaming communication contract is defined between data source and data acquisition component
Notes and Issues:	NA
Sequence Diagram:	

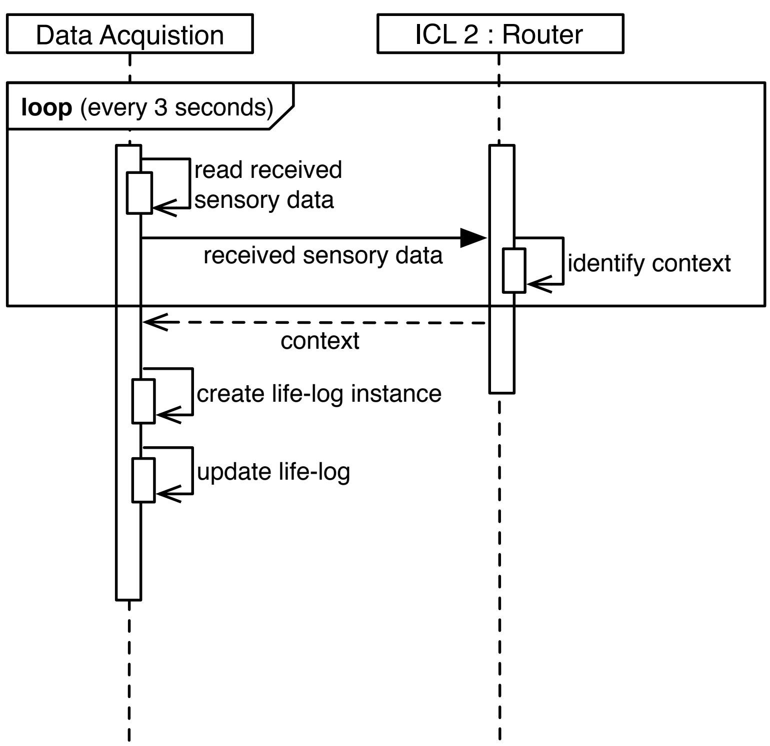


Use Case ID:	*DCL2.5-SUC-03*		
Use Case Name:	Synchronize heterogeneous user data		
Created By:	Bilal Amin	**Last Updated By:**	Bilal Amin
Date Created:	15 July 2015	**Last Revision Date:**	20 Apr 2016
Actors:	DCL 2.5		

Description:	User video data stream is synchronized with the corresponding sensory data
Trigger:	Video camera is streaming user feed
Pre-conditions:	Video data stream is persisted in the video stream buffer
Post-conditions:	Video data stream is synchronized with its corresponding sensory data
Normal Flow:	1. Time stamp and user id of video data stream is read from the video stream buffer 2. Sensory data is searched and retrieved from buffer based on the time stamp and user id 3. Retrieved sensory data is concatenated with the video stream data and stored back in the sensory data buffer for context determination
Alternative Flows:	2a. In step 2 of the normal flow, if the sensory data is not found 1. Data acquisition component deletes the video data stream from the video data stream buffer
Exceptions:	NA
Includes:	NA
Frequency of Use:	Less frequent: If video streaming based data source is available
Assumptions:	4. Video data streaming communication contract is defined between data source and data acquisition component
Notes and Issues:	NA
Sequence Diagram:	

Use Case ID:	*DCL2.5-SUC-04*		
Use Case Name:	Send data for context determination		
Created By:	Bilal Amin	**Last Updated By:**	Bilal Amin
Date Created:	15 July 2015	**Last Revision Date:**	20 Apr 2016
Actors:	Context sender		

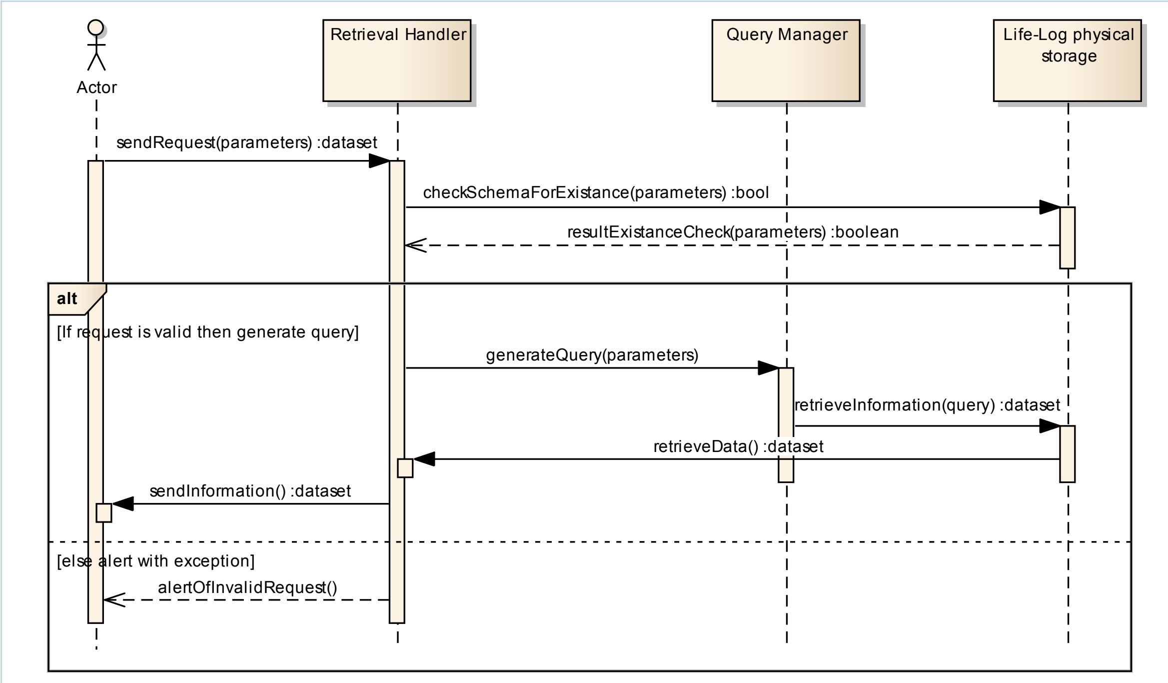
Description:	Sensory data buffer is sent to ICL 2.5 for context determination
Trigger:	Sensory data is available for context determination
Pre-conditions:	Sensory data is persisted in the buffer
Post-conditions:	Sensory data is sent for context determination
Normal Flow:	1. Context sender reads sensory data from the sensory data buffer 2. Context sender creates communication object by serialization 3. Communication object is sent to the ICL 2.5 server
Alternative Flows:	NA
Exceptions:	NA
Includes:	NA
Frequency of Use:	Frequent: whenever context need to be determined
NFR ID:	Leave it blank
Assumptions:	NA
Notes and Issues:	NA
Sequence Diagram:	



Use Case ID:	*DCL2.5-SUC-05*		
Use Case Name:	Receive context data		
Created By:	Bilal Amin	**Last Updated By:**	Bilal Amin
Date Created:	15 July 2015	**Last Revision Date:**	20 Apr 2016
Actors:	ICL 2.5 server		
Description:	After the determination by ICL 2.5, context is received by context receiver component and forwarded for non-volatile storage		
Trigger:	New context or change in previous context is determined by ICL 2		
Pre-conditions:	Context data is available		
Post-conditions:	Context data is sent for non-volatile storage		
Normal Flow:	1. Context receiver receives context object 2. Context receiver de-serializes context object 3. Context object is sent for non-volatile persistence (async) 4. Context object is sent for life-log mapping		
Alternative Flows:	NA		
Exceptions:	NA		
Includes:	NA		
Frequency of Use:	Frequent: whenever context is determined		
Assumptions:	NA		
Notes and Issues:	NA		
Sequence Diagram:			

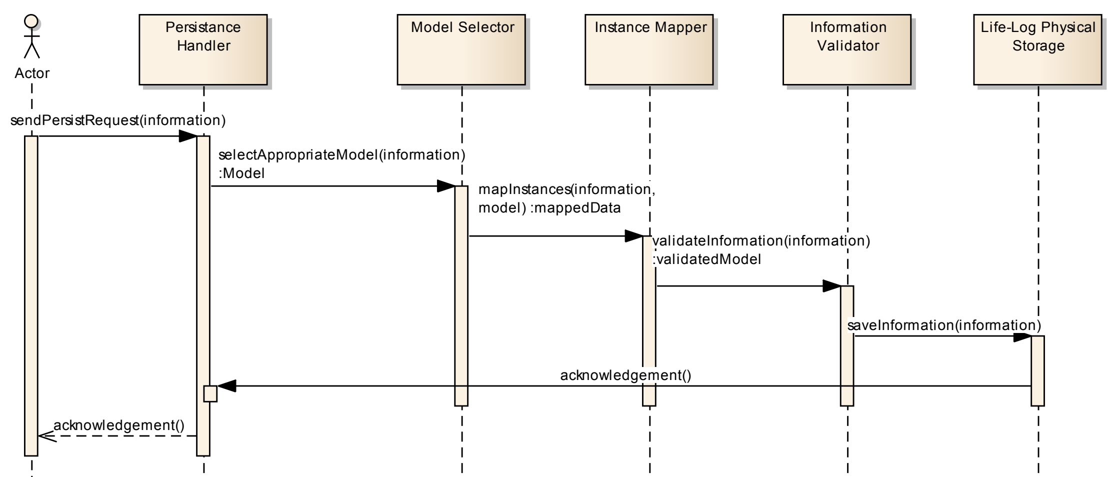
Use Case ID:	*DCL2.5-SUC-06*		
Use Case Name:	Retrieve Life-log Information		
Created By:	Taqdir Ali	**Last Updated By:**	Taqdir Ali
Date Created:	15 July 2015	**Last Revision Date:**	20 Apr 2016
Actors:	SCL 2.5, KCL 2.5, SL 2.5		

Description:	Each actor needs information from life log for further processing. All actors shall request their related and desired Life-log information from physical storage.
Trigger:	On request of a particular actor to access required information
Pre-conditions:	The actor shall be authorized with full access on the Life-Log data.
Post-conditions:	Provide the required data to particular layer
Normal Flow:	1. Actor sends request for desired Life-log information. 2. The desired request shall be checked for information existence. If request is valid a. Prepare the query for desired information based on request b. Load the requested information from physical storage c. Send back the loaded information to the actor
Alternative Flows:	2b. The desired data is not exist in the schema, invalid request 1. Acknowledge the actor with exception of invalid request.
Exceptions:	NA
Includes:	NA
Frequency of Use:	Whenever Life-log information is required.
Assumptions:	NA
Notes and Issues:	NA
Sequence Diagram:	



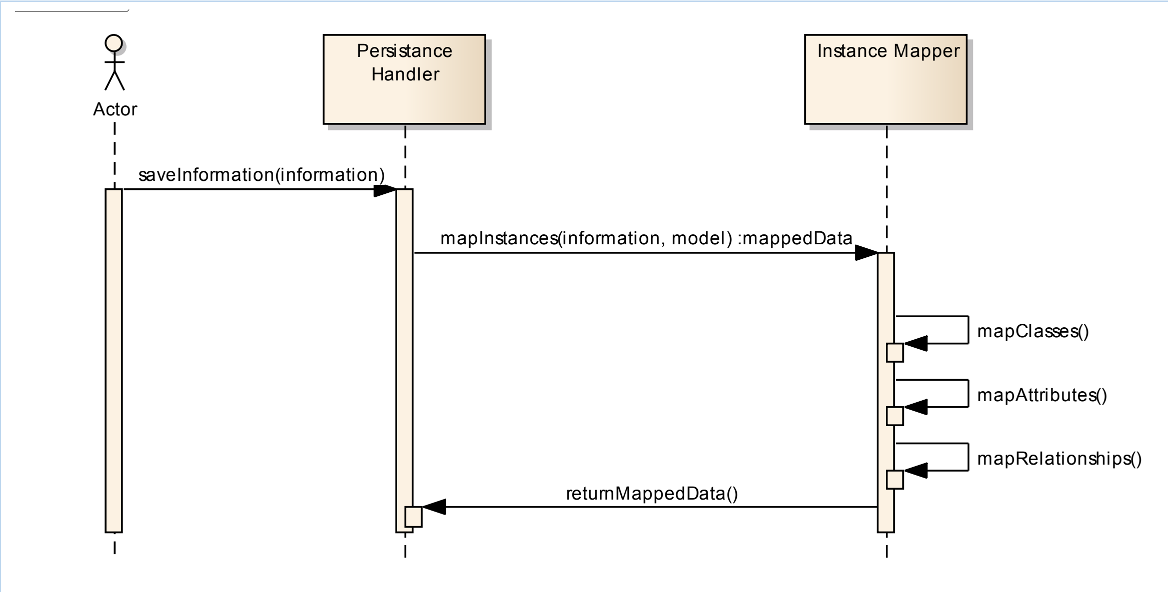
Use Case ID:	*DCL2.5-SUC-07*		
Use Case Name:	Persist Life-log Information		
Created By:	Taqdir Ali	**Last Updated By:**	Taqdir Ali
Date Created:	15 July 2015	**Last Revision Date:**	20 Apr 2016
Actors:	ICL 2.5		
Description:	Each actor performed some specific operations on incoming data from external resources or on already existing information in Life-log repository. In both cases the information shall be updated and stored in Life-log repository.		
Trigger:	On request of a particular actor to persist required information		
Pre-conditions:	The actor shall be authorized with full access on the Life-Log data.		
Post-conditions:	Successfully stored the created Life-log information		
Normal Flow:	1. Actor sends request to persist new generated Life-log information. 2. Passes the new created information to check the appropriate hierarchical structure.		

	3. The appropriate selected hierarchical structure with input information passes to find the information instances. 4. Check the consistency among the records and their relationship. 5. Store the validated and structured information into physical storage
Alternative Flows:	NA
Exceptions:	NA
Includes:	Map Instances, Validate Instances, save information to physical storage.
Frequency of Use:	Whenever new information is generated.
Assumptions:	NA
Notes and Issues:	Capability to process multiple actors requests for storage.
Sequence Diagram:	



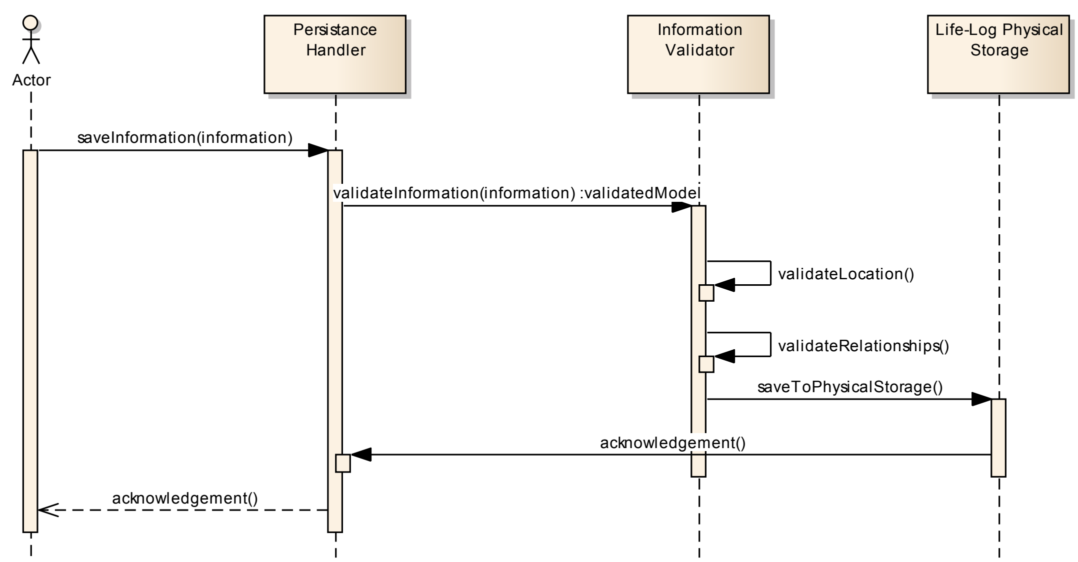
Use Case ID:	*DCL2.5-SUC-08*		
Use Case Name:	Map Instances		
Created By:	Taqdir Ali	**Last Updated By:**	Taqdir Ali
Date Created:	15 July 2015	**Last Revision Date:**	20 Apr 2016
Actors:	DCL2.5, ICL2.5, SCL2.5, SL2.5, and Life-Log physical storage		

Description:	The information produced by each actor shall be mapped against the hierarchical structure of storage.
Trigger:	On request of a particular actor to persist required information
Pre-conditions:	The actor shall be authorized with full access on the Life-Log data.
Post-conditions:	Successfully mapped the instances with correct Life-log information schema
Normal Flow:	1. Actor sends request to persist new generated information. 2. System searches each information records against hierarchical structure. 3. System finds appropriate classes of the instances 4. System extracts attributes in the instances. 5. Find the relationship among the information records. 6. Pass the annotated information for validation.
Alternative Flows:	
Exceptions:	NA
Includes:	NA
Frequency of Use:	Whenever new information is persist.
Assumptions:	NA
Notes and Issues:	Capability to process multiple actors requests for storage.
Sequence Diagram:	



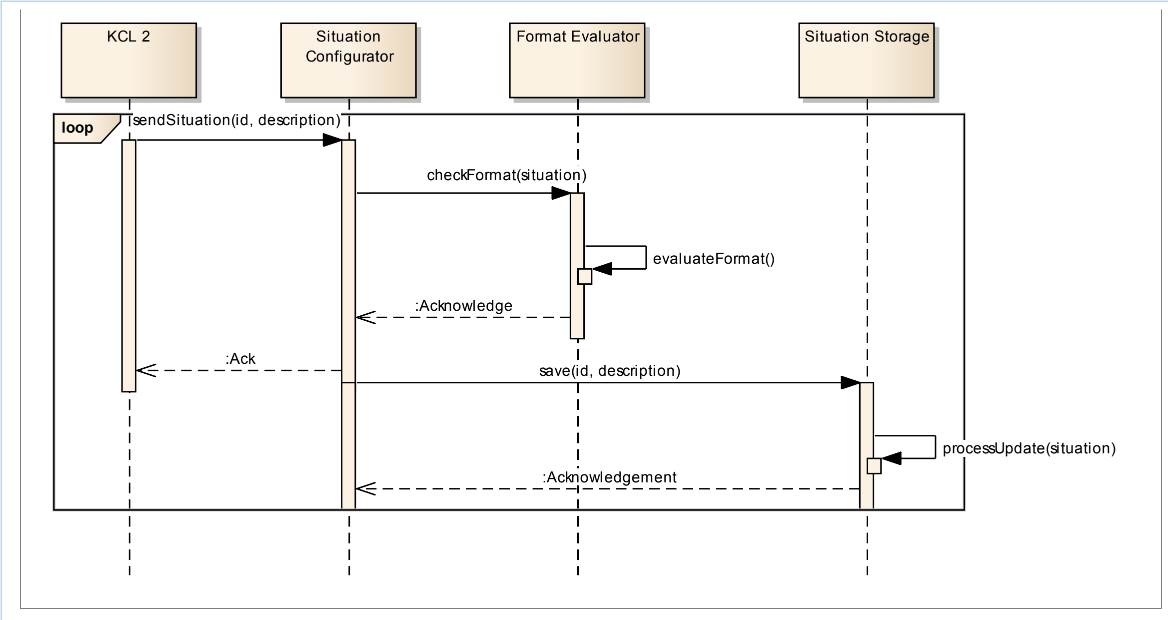
Use Case ID:	*DCL2.5-SUC-09*		
Use Case Name:	Validate Instances		
Created By:	Taqdir Ali	**Last Updated By:**	Taqdir Ali
Date Created:	15 July 2015	**Last Revision Date:**	20 Apr 2016
Actors:	DCL2.5, ICL2.5, SCL2.5, SL2.5, and Life-Log physical storage		
Description:	The mapped information in previous use case shall be checked for consistency among the existing information.		
Trigger:	On request of a particular actor to persist required information		
Pre-conditions:	The actor shall be authorized with full access on the Life-Log data.		
Post-conditions:	Successfully validate the instances with correct Life-log information schema		
Normal Flow:	1. Actor send request to persist new generated information. 2. DCL passes the mapped information for validation of information and their relationships. 3. The system checks the information according to the specific location in the hierarchy. 4. The system checks and builds the relationship		

	among concepts. 5. The verified information shall be passed for persistence.
Alternative Flows:	NA
Exceptions:	NA
Includes:	NA
Frequency of Use:	Whenever new information is persist.
Assumptions:	NA
Notes and Issues:	Capability to process multiple actors requests for storage.
Sequence Diagram:	



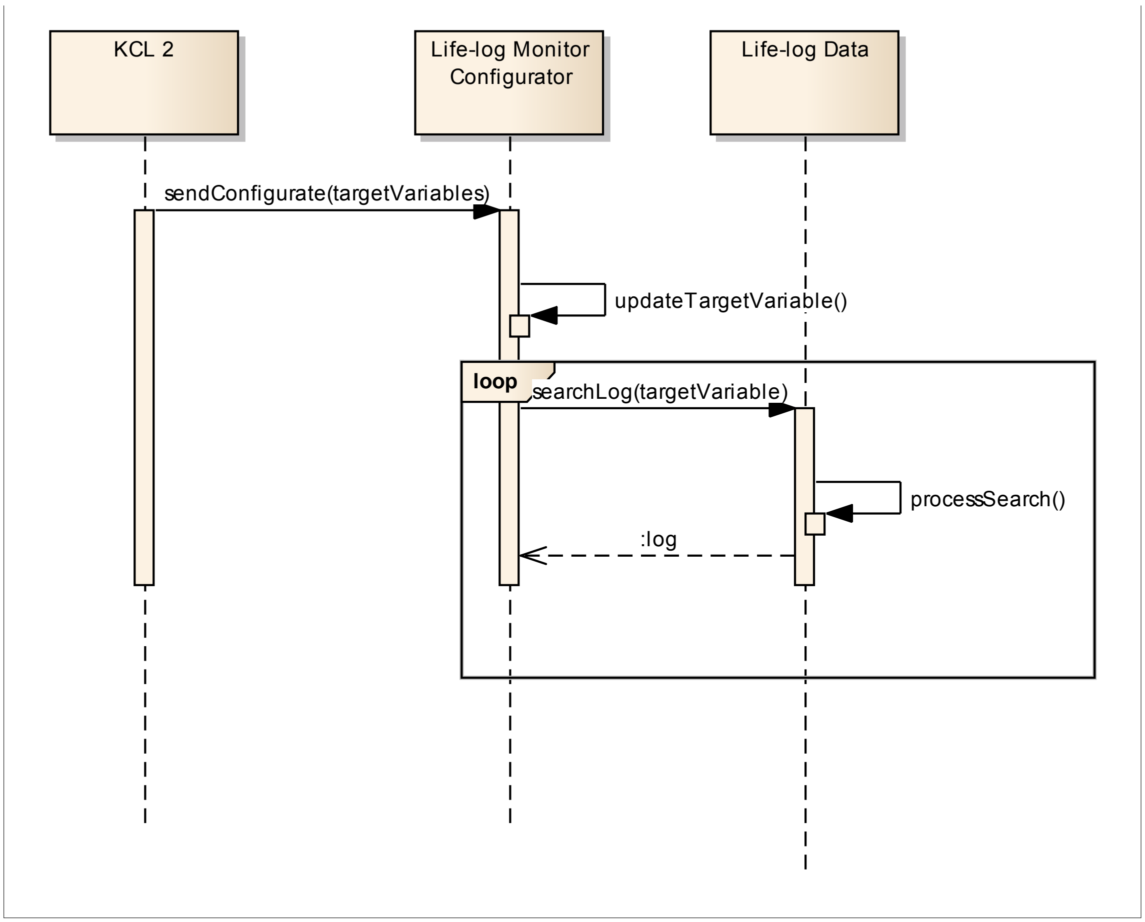
Use Case ID:	*DCL2.5-SUC-10*		
Use Case Name:	Situation Configuration		
Created By:	Bilal Ali	**Last Updated By:**	Bilal Ali
Date Created:	15 July 2015	**Last Revision Date:**	20 Apr 2016
Actors:	KCL 2.5		
Description:	Situation is determined by experts and is communicated to DCL 2.5 for monitoring the Life-log.		

Trigger:	Creation of new rule to capture a situation.
Pre-conditions:	KCL 2.5 and DCL 2.5 should agree on common representation of sharing information of Situation configuration
Post-conditions:	1. Situation is stored against a specific category. 2. Situation is available for monitoring the Life-log.
Normal Flow:	1. KCL 2.5 connects to DCL 2.5 and send the newly created situation in common configuration format. 2. DCL 2.5 evaluates the format of received situation configuration. 3. DCL 2.5 responds with acknowledgement message. 4. Situation will be parsed into components. 5. Parsed components are updated in persistent storage as per categories.
Alternative Flows:	NA
Exceptions:	Format of situation is not according the agreement.
Includes:	NA
Frequency of Use:	Invoked per situation creation by the expert.
Assumptions:	Well defined schema is available to store situation persistently
Notes and Issues:	Standardize situation format is a challenging task
Sequence Diagram:	



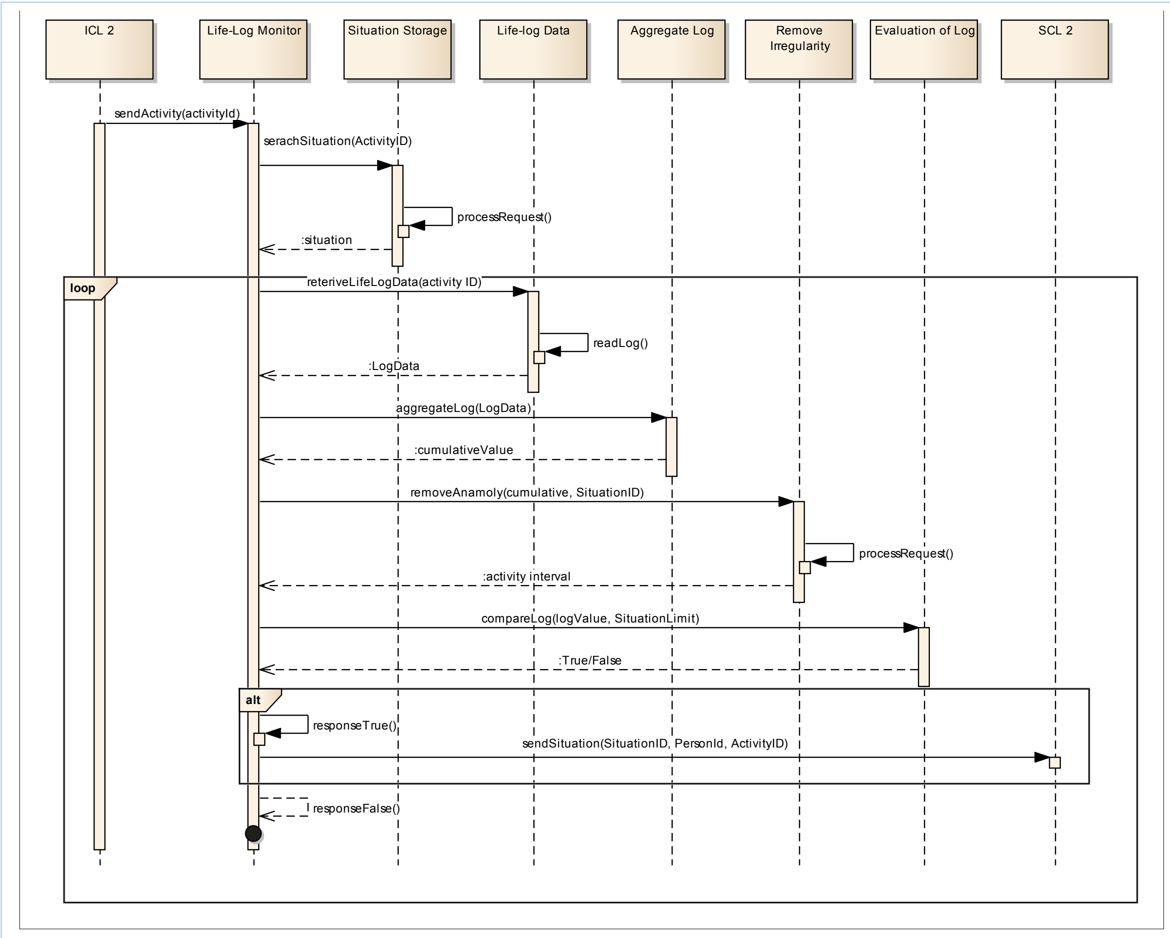
Use Case ID:	*DCL2.5-SUC-11*		
Use Case Name:	LLM Configuration for target Variables		
Created By:	Bilal Ali	**Last Updated By:**	Bilal Ali
Date Created:	15 July 2015	**Last Revision Date:**	20 Apr 2016
Actors:	KCL2.5, Experts		
Description:	Configure the Life-log monitor for the screening of the target variable from Life-log data.		
Trigger:	On start of user's monitored activity		
Pre-conditions:	1. Expert defines target variable in common configuration format. 2. Access to Life-log.		
Post-conditions:	Targeted log is retrieved from Life-log data as per target variable requirements.		
Normal Flow:	1. KCL 2.5 will share the target variables in common configured format created by expert. 2. Life-log monitor is configured on the basis of the shared target variable. 3. Life-log monitor retrieve log data from Life-log against the target variables.		

Alternative Flows:	NA
Exceptions:	NA
Includes:	NA
Frequency of Use:	On update of common configuration format.
Assumptions:	
Notes and Issues:	Real time accommodation of update in common configuration format is challenging task.
Sequence Diagram:	



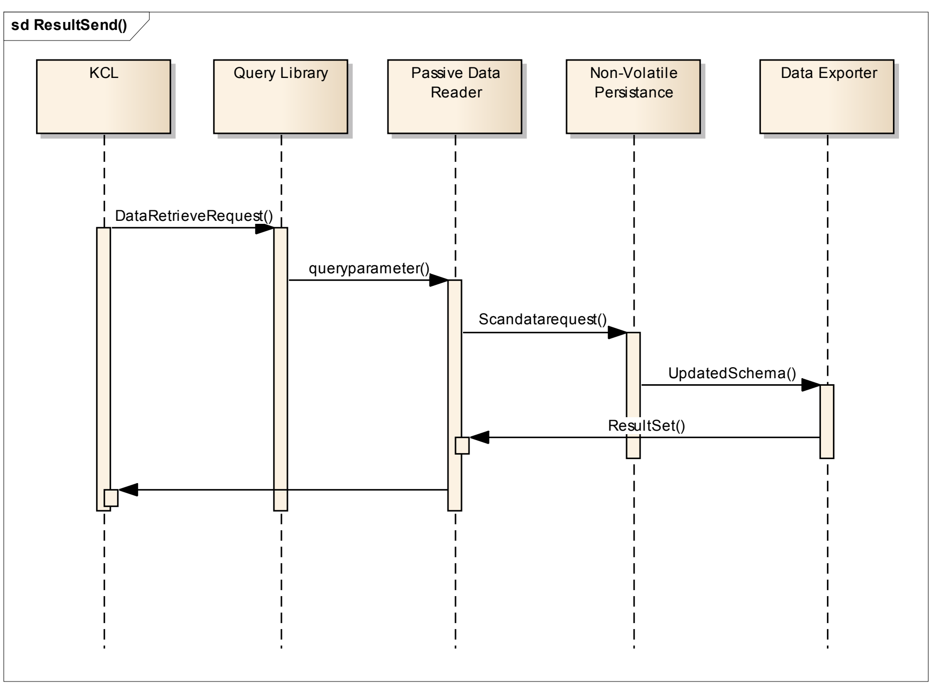
Use Case ID:	*DCL2.5-SUC-12*		
Use Case Name:	LLM for Situation Detection		
Created By:	Bilal Ali	**Last Updated By:**	Bilal Ali
Date Created:	15 July 2015	**Last Revision Date:**	20 Apr 2016

Actors:	Life-log Data, SCL 2.5, ICL 2.5
Description:	Identification of the existence of a condition in user activities to highlight the alarming situation as per experts' understanding.
Trigger:	On start of user's monitored activity
Pre-conditions:	1. Activity is identified. 2. Situation is configured. 3. Access to Life-log.
Post-conditions:	Alarming situation is detected and triggered the SCL 2.5 with situation and user.
Normal Flow:	1. ICL 2.5 recognizes activity and sends to Life-log. 2. Life-log monitor identify the target activity. 3. Retrieve associated situation with the activity. 4. Continuous access that activity log. 5. Aggregate the interval/duration of activity. 6. Remove the irregularity in activity as per situation. 7. Evaluate the duration of activity against the situation. a. If situation condition meets then send message to SCL 2.5 to inform about the occurrence of a situation along with user information. b. If situation condition does not occur, don't send message to SCL 2.5
Alternative Flows:	NA
Exceptions:	NA
Includes:	NA
Frequency of Use:	For every activity with configured situation.
Assumptions:	
Notes and Issues:	Management of irregularity in activity is a challenging task.
Sequence Diagram:	



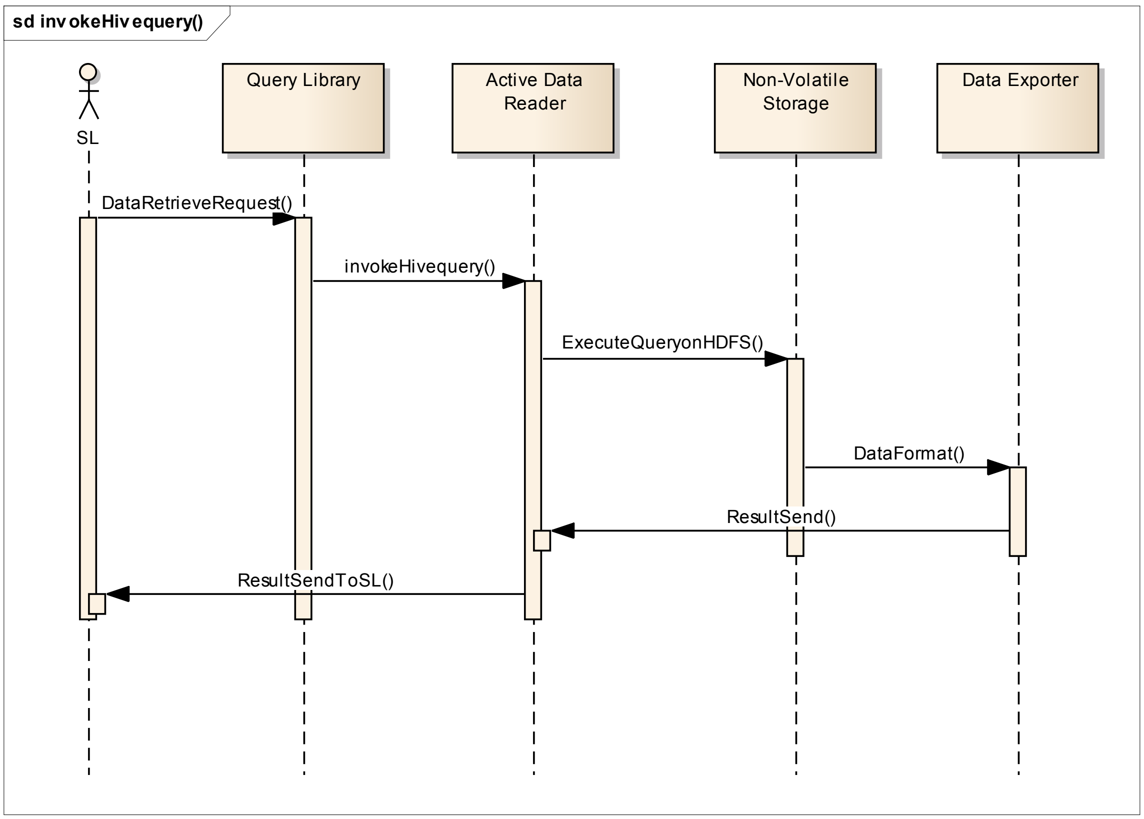
Use Case ID:	*DCL2.5-SUC-13*		
Use Case Name:	Retrieve sensory data from non-volatile storage for intermediate data generation (offline)		
Created By:	Cho / Usman	**Last Updated By:**	Bilal Amin
Date Created:	20 Apr 2016	**Last Revision Date:**	20 Apr 2016
Actors:	KCL 2.5		
Description:	Request for Schema of the persisted data is received by the Passive Data Reader. Scanned and most updated schema from Non-volatile storage is returned to KCL.		
Trigger:	KCL 2.5 requests the data		
Pre-conditions:	1. Relational IDB schema has already been described and shared 2. The data exists in HDFS		

Post-conditions:	1. The data has been transformed and exported to IDB 2. The KCL 2.5 is informed
Normal Flow:	1. DCL 2.5 receives requests from KCL and creates a Hive query 2. Hive query is executed using on HDFS to retrieve the data 3. Required data is returned as a result set to Data exporter 4. Result-set is converted into data message and returned to KCL
Alternative Flows:	NA
Exceptions:	NA
Includes:	NA
Frequency of Use:	Less frequent, offline process, may be executed once
Assumptions:	Hive query results are easy to transform to relational format
Notes and Issues:	NA
Sequence Diagram:	



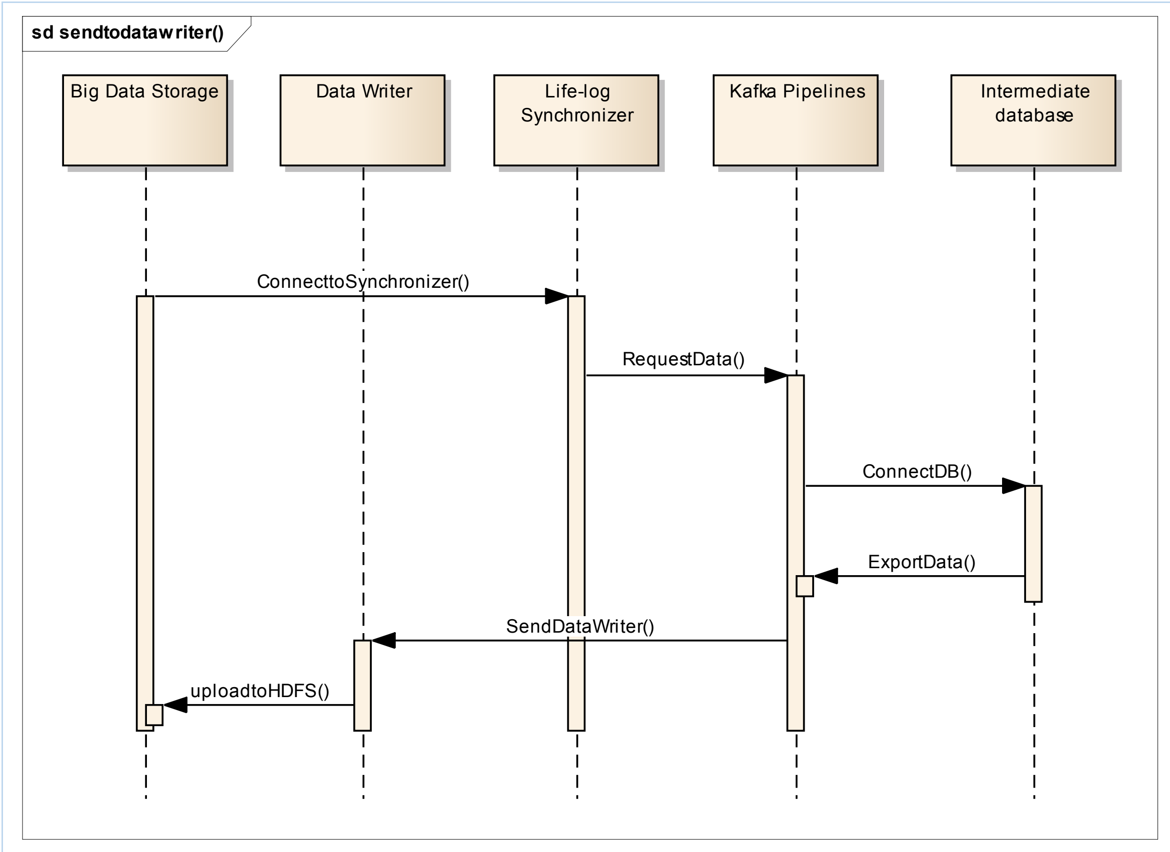
Use Case ID:	*DCL2.5-SUC-14*		
Use Case Name:	Retrieve life-log data from active data reader for visualization and analytics (online)		
Created By:	Usman Akhtar	**Last Updated By:**	Bilal Amin
Date Created:	21 April 2016	**Last Revision Date:**	20 Apr 2016
Actors:	SL 2.5		
Description:	Raw life-log data in HDFS is retrieved and provided to analytics component from active data reader component for visualization and analytics		
Trigger:	Request for data from descriptive analytics is received		
Pre-conditions:	Life-log data exists and persisted in HDFS (non-volatile storage) and access through Hive metastore.		
Post-conditions:	1. Required data is retrieved from HDFS 2. Request is sent back to data exporter to appropriate format 3. Required data is communicated to descriptive analytics in SL 2.5 directly (online)		
Normal Flow:	1. Descriptive analytics in SL 2.5 requests data 2. Hive query is selected from Query Library and implicitly transformed into a MapReduce Job. 3. MapReduce job is executed on HDFS and results are retrieved. 4. Retrieved results are send to data exporter to adjust appropriate format 5. Results are forwarded directly to descriptive analytics		
Alternative Flows:	1. Descriptive analytics requests not from the active data reader 2. Active Data Reader receives but unable to run over the Hive 3. Duplicate request results are directly forwarded to descriptive analytics 4. First time requests follow the normal flow.		

Exceptions:	NA
Includes:	NA
Frequency of Use:	Frequent requests from descriptive analytics in SL
Assumptions:	Request format contract already defined between SL(descriptive analytics) and DCL (Active Data Reader)
Notes and Issues:	NA
Sequence Diagram:	



Use Case ID:	*DCL2.5-SUC-15*		
Use Case Name:	Life-log sync between intermediate database and Big Data		
Created By:	Usman Akhtar	**Last Updated By:**	Bilal Amin
Date Created:	21 April 2016	**Last Revision Date:**	21 April 2016
Actors:	DC		
Description:	Receive and persist life-log data from intermediate		

	database in to HDFS through Kafka Pipelines
Trigger:	Whenever the changes occurs in intermediate the trigger send to Big data storage for updates
Pre-conditions:	1. Data storage structure, directory structure in HDFS defined 2. File formats and data formats in HDFS known 3. Life-log data is received from intermediate database 4. Hive metastores and Big Data server is already running
Post-conditions:	1. Life-log data is persisted in HDFS non-volatile storage 2. Data is available for processing and access by SL and KCL
Normal Flow:	1. Big Data server listening for data requests from intermediate database through Kafka pipelines 2. Kafka push the data from intermediate and send to the data writer 3. Write to the HDFS for non-volatile persistence
Alternative Flows:	NA
Exceptions:	NA
Includes:	NA
Frequency of Use:	When data updates
Assumptions:	Data format and specifications already defined between intermediate and Big Data
Notes and Issues:	NA
Sequence Diagram:	



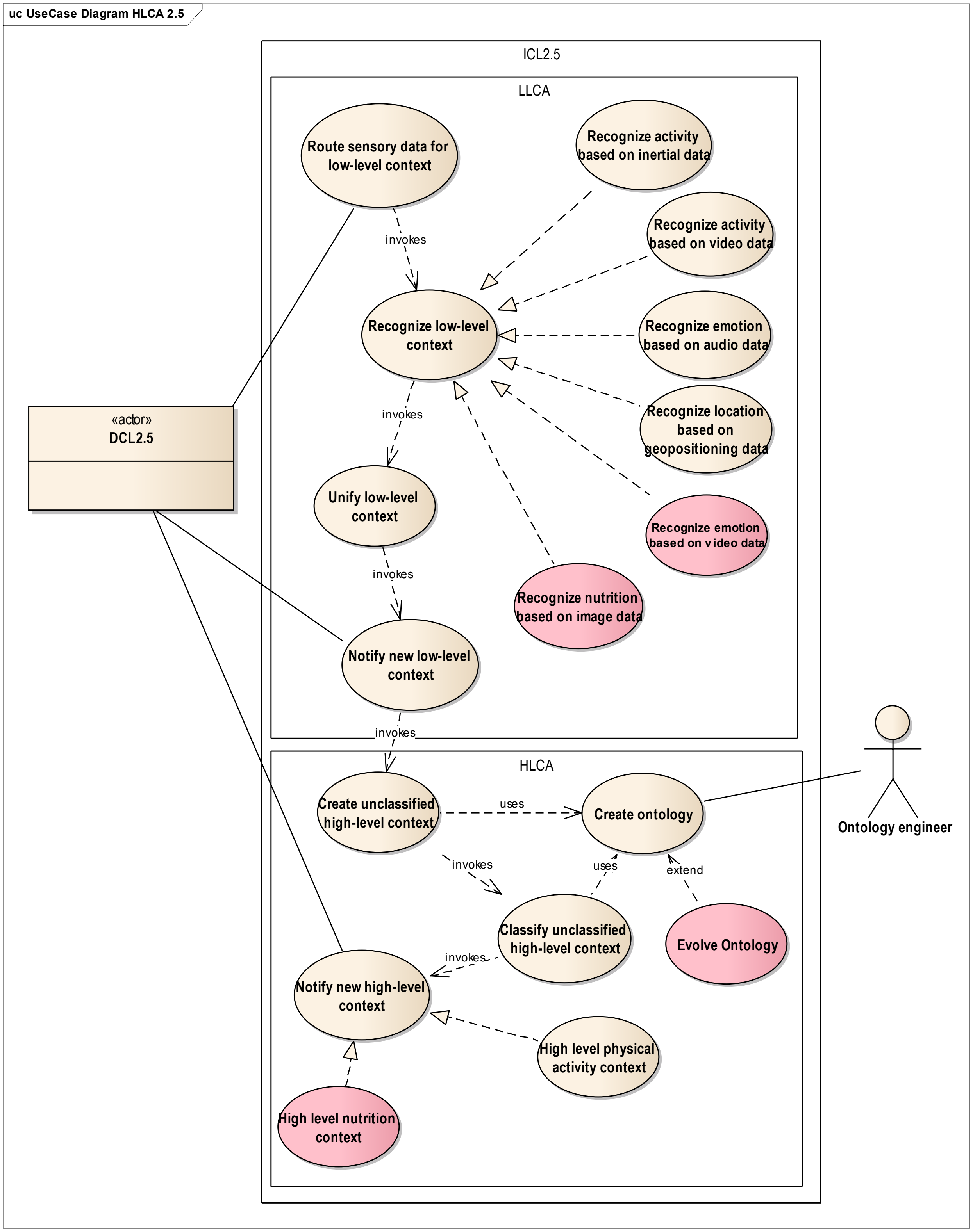
sd sendtodatawriter()
Big Data Storage
Data Writer
Life-log Synchronizer
Kafka Pipelines
Intermediate database
ConnecttoSynchronizer()
RequestData()
ConnectDB()
ExportData()
SendDataWriter()
uploadtoHDFS()

Information Curation Layer (ICL Ver. 2.5)

List of Use cases

Use case ID#	Name
ICL2.5-SUC-01	Derive optimal low-level context recognizer
ICL2.5-SUC-02	Create low-level context recognizers
ICL2.5-SUC-03	Route sensory data for the low-level context identification
ICL2.5-SUC-04	Recognize user low-level context
ICL2.5-SUC-05	Recognize user activity based on inertial raw sensory data
ICL2.5-SUC-06	Recognize user activity based on video raw sensory data
ICL2.5-SUC-07	Recognize user location based on geopositioning raw sensory data
ICL2.5-SUC-08	Recognize user emotion based on audio raw sensory data
ICL2.5-SUC-09	Unify low-level contexts
ICL2.5-SUC-10	Notify new low-level context
ICL2.5-SUC-11	Create unclassified high-level context instance
ICL2.5-SUC-12	Classify high-level context instance
ICL2.5-SUC-13	Notify new high-level context
ICL2.5-SUC-14	Load context ontology model
ICL2.5-SUC-15	Store context instance
ICL2.5-SUC-16	Retrieve context instance
ICL2.5-SUC-17	Recognize emotion based on video data
ICL2.5-SUC-18	Recognize nutrition based on image data
ICL2.5-SUC-19	Evolve Ontology
ICL2.5-SUC-20	High Level Physical Activity Context
ICL2.5-SUC-21	High Level Nutrition Context

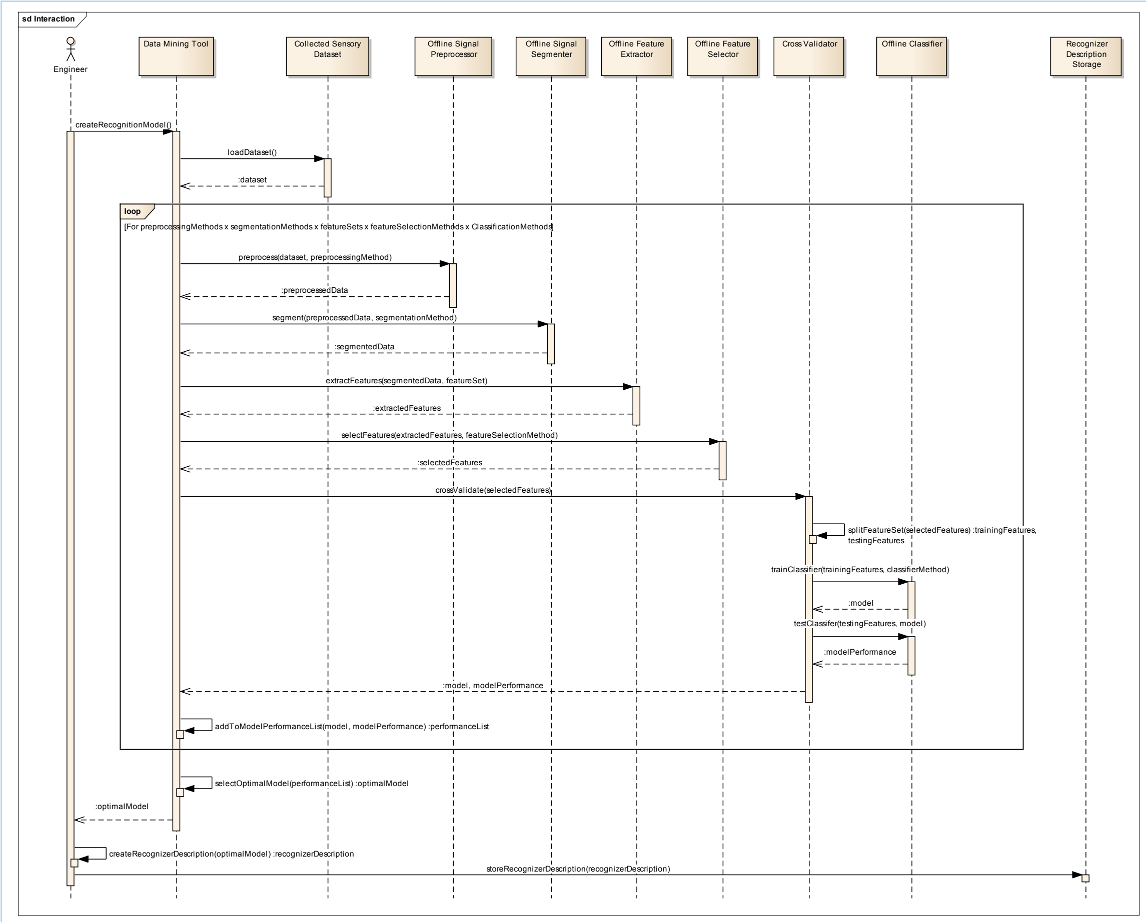
Use case Diagram



Use case Description

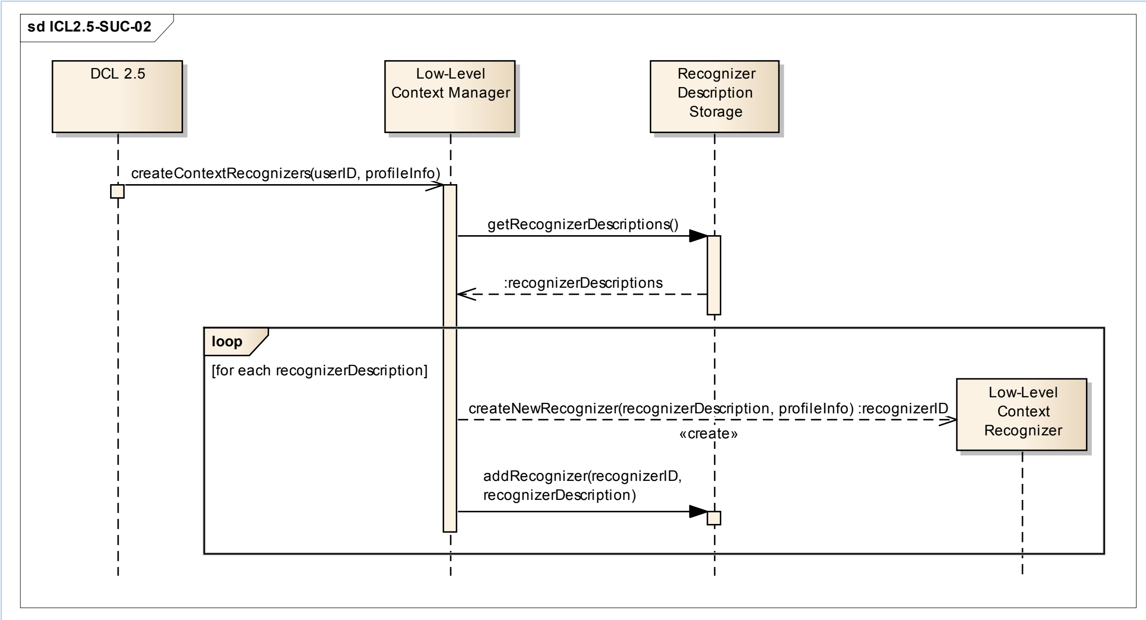
Use Case ID:	*ICL2.5-SUC-01*		
Use Case Name:	Derive optimal low-level context recognizer		
FR ID:	MM-FR-11		
Created By:	Oresti Banos	**Last Updated By:**	Oresti Banos
Date Created:	22 July 2015	**Last Revision Date:**	24 July 2015
Actors:	Engineer		
Description:	Create an optimal recognition model through the evaluation of multiple recognition model candidates (offline process).		
Trigger:	Engineer initiates the process for creating an optimal recognition model		
Pre-conditions:	1. A human expert or engineer sets up the experimental setup for the evaluation process		
Post-conditions:	2. The optimal recognition model among considered is delivered to the expert 3. A recognizer descriptor containing the characteristics of the optimal model is stored		
Normal Flow:	1. Load dataset 2. For each combination of preprocessing methods, segmentation methods, feature sets, feature selection methods, and classification methods, the dataset is preprocessed (e.g., filtered) 3. The preprocessed dataset is segmented (e.g., partitioned into windows) 4. Features (e.g., mean, variance) are extracted from each segment of the dataset 5. The best features are selected 6. Cross validation is applied to the selected features a. The feature set is split into training and testing b. The classifier is trained using the training features c. The classifier is tested in order to		

	determine the model performance 7. The model performance is stored 8. Once the model performance has been calculated for all the possible combinations, the optimal model is selected 9. A recognizer descriptor is generated according to the characteristics of the model (e.g., median filtering, 3 sec window size, etc.) 10. The generated recognizer descriptor is stored
Alternative Flows:	NA
Exceptions:	NA
Includes:	NA
Frequency of Use:	Infrequent
NFR ID:	MM-NFR-05
Assumptions:	NA
Notes and Issues:	Matlab and Weka tools will be used for this task. A multimodal dataset must be collected for the training and evaluation of the candidates models.
Sequence Diagram:	



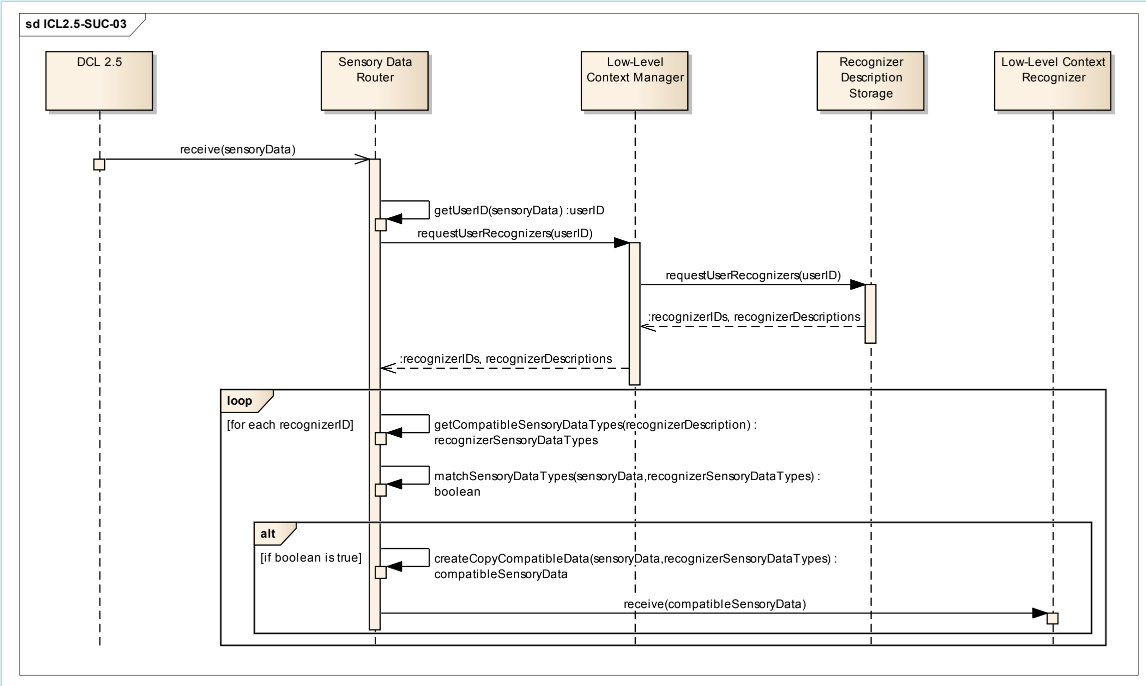
Use Case ID:	*ICL2.5-SUC-02*		
Use Case Name:	Create low-level context recognizers		
FR ID:	MM-FR-11		
Created By:	Oresti Banos	**Last Updated By:**	Oresti Banos
Date Created:	22 July 2015	**Last Revision Date:**	24 July 2015
Actors:	DCL 2.5		
Description:	Sensory data is received from DCL 2.5 and it is distributed to the corresponding low-level context recognizer based on the data type(s).		
Trigger:	Receive userID and part of the user profile information send by DCL 2.5 to ICL2.5		
Pre-conditions:	DCL 2.5 sends the userID and part of the user profile information to ICL 2.5 whenever a new user is registered in the platform		

Post-conditions:	Low-level context recognizers are generated for the new user for all context types
Normal Flow:	1. Receive UserID and (part of the) user profile information 2. Load the recognizer descriptions containing the low-level context model types (e.g., emotion recognizer) and characteristics (e.g., median filtering, 3 sec window size, etc.) 3. Create a new recognizer for each recognizer description 4. Create a recognizer identifier for the generated recognizer 5. Save the recognizer identifier in a persistent storage
Alternative Flows:	NA
Exceptions:	NA
Includes:	NA
Frequency of Use:	Less frequent
NFR ID:	NA
Assumptions:	• DCL 2.5 will send the required user profile information together with the userID only the first time a user is registered • No user profile updates are considered in this version
Notes and Issues:	NA
Sequence Diagram:	



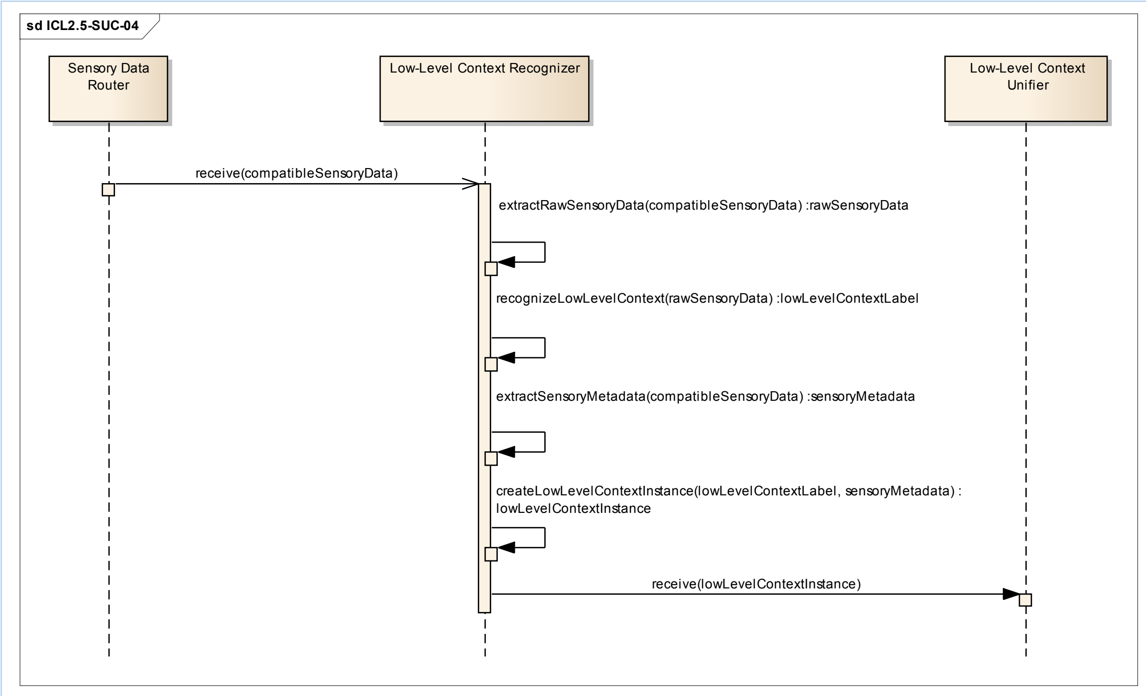
Use Case ID:	*ICL2.5-SUC-03*		
Use Case Name:	Route sensory data for the low-level context identification		
FR ID:	MM-FR-10		
Created By:	Oresti Banos	**Last Updated By:**	Oresti Banos
Date Created:	14 July 2015	**Last Revision Date:**	24 July 2015
Actors:	DCL 2.5, Low-Level Context Recognizer		
Description:	Sensory data is received from DCL 2.5 and it is distributed to the corresponding low-level context recognizer based on the data type(s).		
Trigger:	Receive sensory data send by DCL 2.5 to ICL2.5		
Pre-conditions:	DCL 2.5 sends sensory data, i.e., raw sensory data plus sensory metadata (e.g., data type, time stamp, device ID, device type, and user ID)		
Post-conditions:	The adequate raw sensory data is sent to each low-level context recognizer in order to perform the recognition process		

Normal Flow:	1. Receive sensory data 2. Get the user identifier to which the sensory data belongs 3. Load the low-level context recognizers identifiers for the given user 4. For each low-level context recognizer identifier, get the sensory data type(s) it requires 5. Match the received sensory data with the sensory data type(s) required by the low-level context recognizer 6. Create a copy with the compatible data required by the low-level context recognizer 7. Distribute the data to the corresponding low-level context recognizer
Alternative Flows:	NA
Exceptions:	a. If no compatible data types are identified for the given low-level context recognizer 1. Go to step 3
Includes:	NA
Frequency of Use:	Very frequent: determined by the rate of sensory data reception from DCL 2.5
NFR ID:	NA
Assumptions:	• There is an established communication between DCL 2.5 and the Sensory Data Router • The communication channel between the DCL 2.5 and the Sensory Data Router is secure • Incoming sensory data is already preprocessed (i.e., without missing samples and with synchronized streams)
Notes and Issues:	NA
Sequence Diagram:	



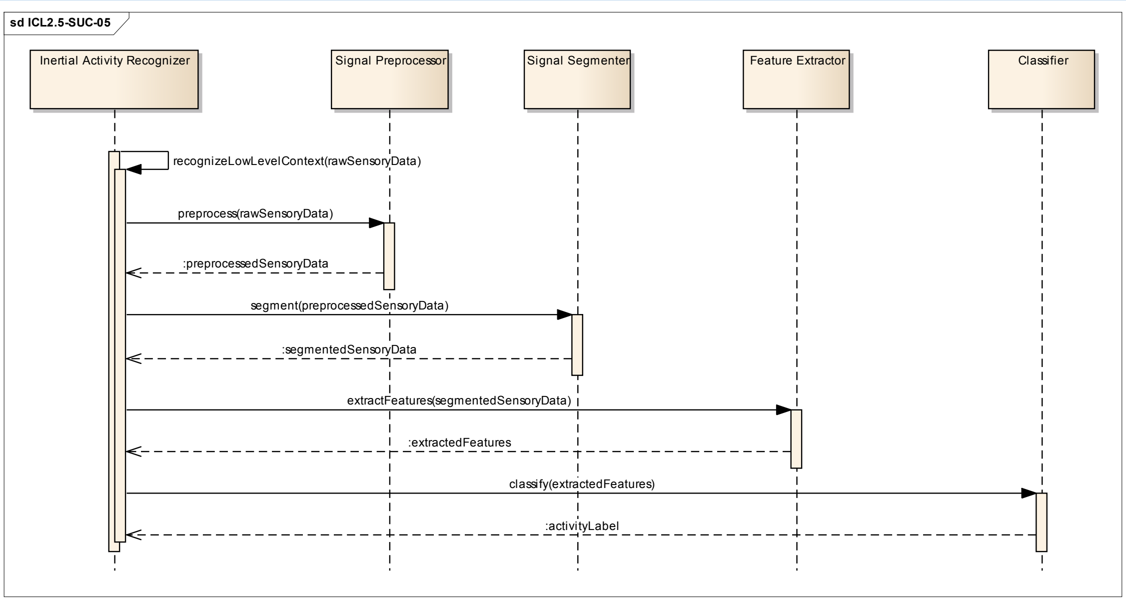
Use Case ID:	*ICL2.5-SUC-04*		
Use Case Name:	Recognize user low-level context		
FR ID:	MM-FR-11		
Created By:	Oresti Banos	**Last Updated By:**	Oresti Banos
Date Created:	14 July 2015	**Last Revision Date:**	20 July 2015
Actors:	Sensory Data Router, Low-Level Context Unifier		
Description:	The low-level context associated to a given user is identified based on the received compatible sensory data. The low level context recognition may be of diverse nature depending upon the data types, thus this use case defines an abstract representation of the process to be followed.		
Trigger:	Receive compatible sensory data		
Pre-conditions:	Compatible sensory data is sent to a given low-level context recognizer		
Post-conditions:	The recognized low-level context instance is provided to the Low-Level Context Unifier		

Normal Flow:	1. Compatible sensory data is received by a given low-level context recognizer 2. The raw sensory data is extracted from the sensory data 3. The low-level context label is recognized 4. The sensory metadata is extracted from the sensory data 5. A low-level context instance is generated by combining the low-level context label and the sensory metadata 6. The generated low-level context instance is provided to the Low-Level Context Unifier
Alternative Flows:	NA
Exceptions:	NA
Includes:	NA
Frequency of Use:	Frequent: at every reception of sensory data
NFR ID:	MM-NFR-05
Assumptions:	Only compatible sensory data is received by each corresponding low-level context recognizer
Notes and Issues:	NA
Sequence Diagram:	



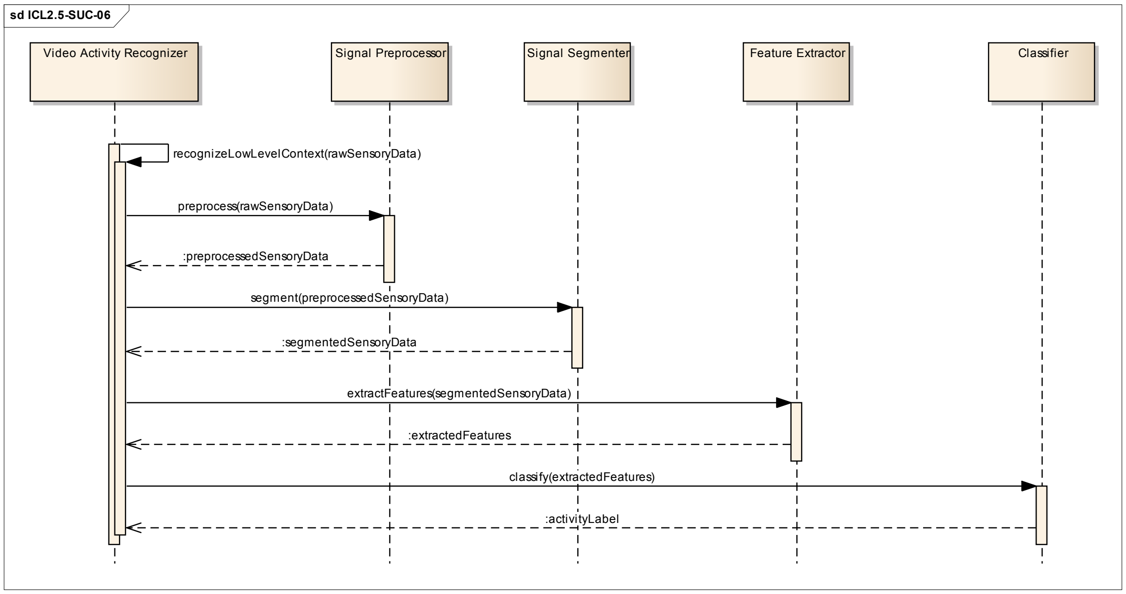
Use Case ID:	*ICL2.5-SUC-05*		
Use Case Name:	Recognize user activity based on inertial raw sensory data		
FR ID:	MM-FR-11		
Created By:	Oresti Banos	**Last Updated By:**	Oresti Banos
Date Created:	14 July 2015	**Last Revision Date:**	20 July 2015
Actors:	ICL2.5-SUC-04		
Description:	Identification of the user physical activity (e.g., “sitting”) based on the processing of the body-motion raw sensory data collected from an inertial sensor. The body-motion raw sensory data consists of triaxial acceleration, triaxial rate of turn and triaxial magnetic field data.		
Trigger:	Request for the recognition of the user activity based on a given inertial raw sensory data		
Pre-conditions:	Raw sensory data is extracted from compatible sensory data (inertial sensory data)		
Post-conditions:	A label corresponding to the recognized activity is		

	generated
Normal Flow:	1. Inertial raw sensory data is received for analysis 2. The raw sensory data is preprocessed (e.g., filtered) 3. The preprocessed raw sensory data is segmented (e.g., partitioned into windows) 4. Features (e.g., mean, variance) are extracted from each segment of raw sensory data 5. The extracted features are classified 6. A label identifying the corresponding user activity is generated
Alternative Flows:	NA
Exceptions:	NA
Includes:	NA
Frequency of Use:	Frequent: at every reception of inertial raw sensory data
NFR ID:	MM-NFR-05
Assumptions:	The raw sensory data is of the nature required by the inertial activity recognizer
Notes and Issues:	NA
Sequence Diagram:	



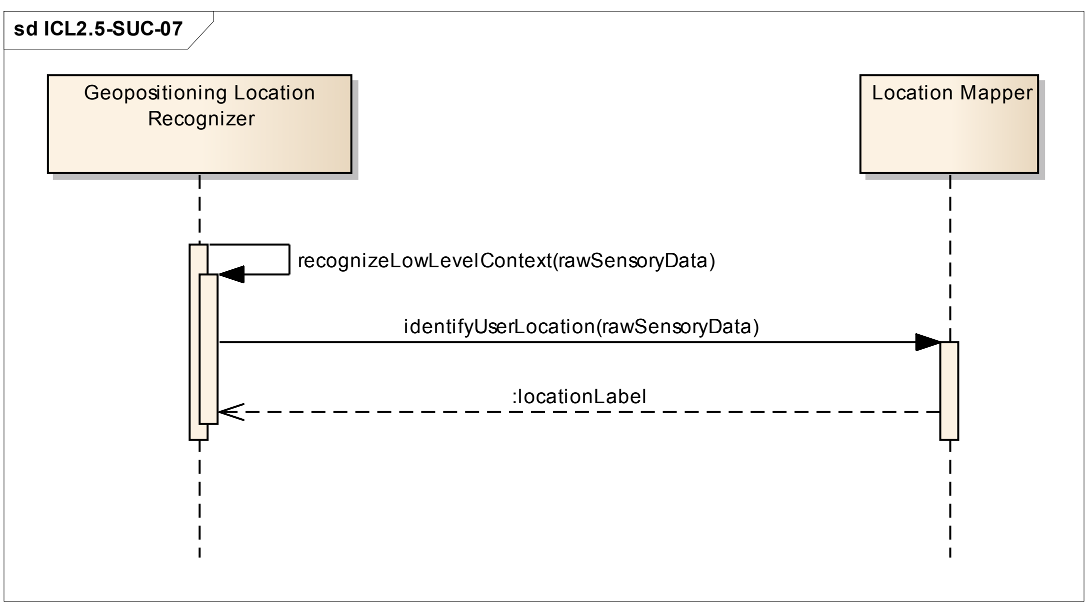
Use Case ID:	*ICL2.5-SUC-06*		
Use Case Name:	Recognize user activity based on video raw sensory data		
FR ID:	MM-FR-11		
Created By:	Oresti Banos	**Last Updated By:**	Oresti Banos
Date Created:	17 July 2015	**Last Revision Date:**	20 July 2015
Actors:	ICL2.5-SUC-04		
Description:	Identification of the user physical activity (e.g., “standing”) based on the processing of the body-motion raw sensory data collected through a video camera. The body-motion raw sensory data consists of RGB and depth video.		
Trigger:	Request for the recognition of the user activity based on a given video raw sensory data		
Pre-conditions:	Raw sensory data is extracted from compatible sensory data (video sensory data)		
Post-conditions:	A label corresponding to the recognized activity is generated		
Normal Flow:	1. Video raw sensory data is received for analysis 2. The raw sensory data is preprocessed (e.g., filtered) 3. The preprocessed raw sensory data is segmented (e.g., partitioned into windows) 4. Features (e.g., SIFT, HOG) are extracted from each segment of raw sensory data 5. The extracted features are classified 6. A label identifying the corresponding user activity is generated		
Alternative Flows:	NA		
Exceptions:	NA		
Includes:	NA		
Frequency of Use:	Frequent: at every reception of video raw sensory data		
NFR ID:	MM-NFR-05		

Assumptions:	The raw sensory data is of the nature required by the video activity recognizer
Notes and Issues:	NA
Sequence Diagram:	



Use Case ID:	*ICL2.5-SUC-07*		
Use Case Name:	Recognize user location based on geopositioning raw sensory data		
FR ID:	MM-FR-11		
Created By:	Oresti Banos	**Last Updated By:**	Oresti Banos
Date Created:	17 July 2015	**Last Revision Date:**	20 July 2015
Actors:	ICL2.5-SUC-04		
Description:	Identification of the user location (e.g., "restaurant") based on the processing of the geopositioning raw sensory data collected from a portable GPS sensor. The body-motion raw sensory data consists of latitude, longitude and speed data.		
Trigger:	Request for the recognition of the user location based on a given geopositioning raw sensory data		
Pre-conditions:	Raw sensory data is extracted from compatible sensory data (geopositioning sensory data)		

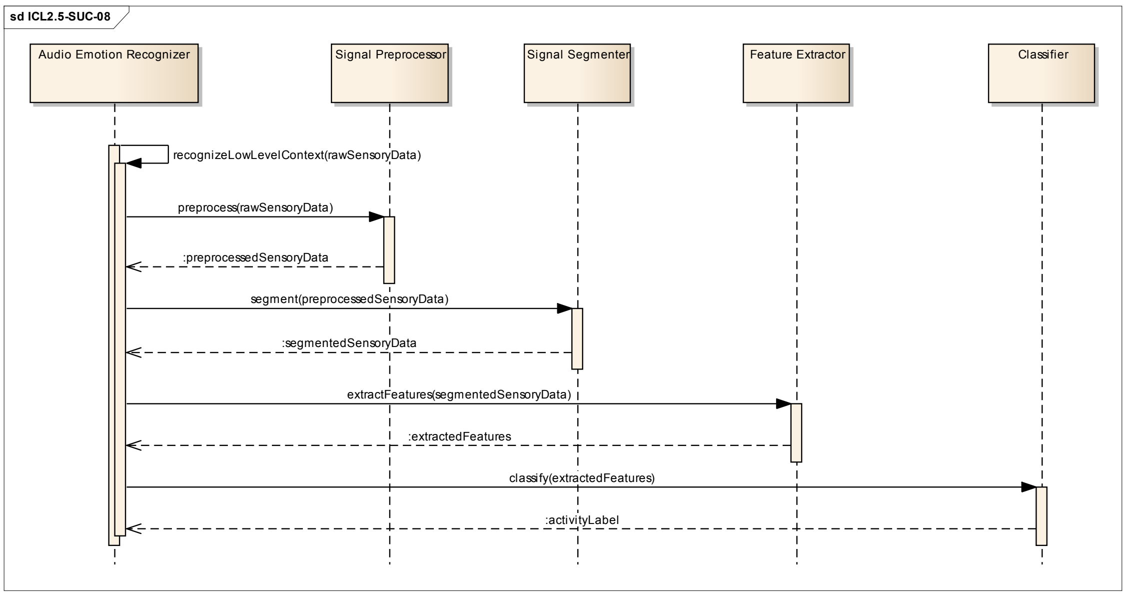
Post-conditions:	A label corresponding to the recognized location is generated
Normal Flow:	1. Geopositioning raw sensory data is received for analysis 2. The geopositioning raw sensory data is compared with the predefined map coordinates 3. A label identifying the corresponding user location is generated
Alternative Flows:	NA
Exceptions:	NA
Includes:	NA
Frequency of Use:	Frequent: at every reception of geopositioning raw sensory data
NFR ID:	MM-NFR-05
Assumptions:	The raw sensory data is of the nature required by the geopositioning location recognizer
Notes and Issues:	NA
Sequence Diagram:	



Use Case ID:	*ICL2.5-SUC-08*
Use Case Name:	Recognize user emotion based on audio raw sensory

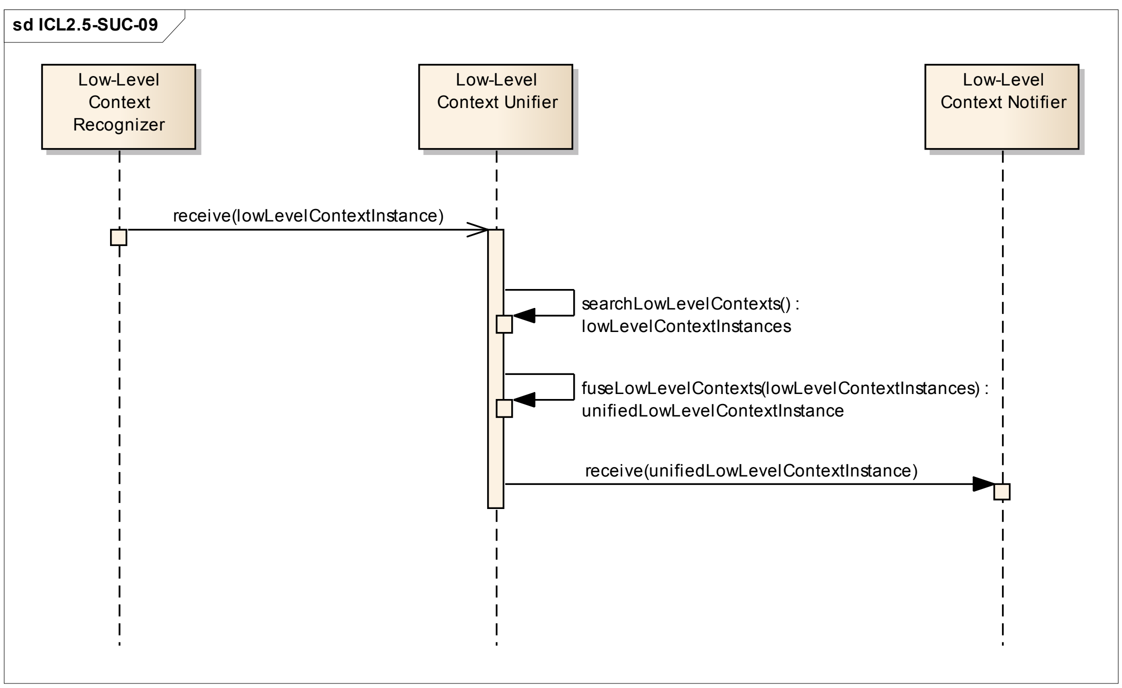
	data		
FR ID:	MM-FR-11		
Created By:	Oresti Banos	**Last Updated By:**	Oresti Banos
Date Created:	17 July 2015	**Last Revision Date:**	20 July 2015
Actors:	ICL2.5-SUC-04		
Description:	Identification of the user emotional state (e.g., “happy”) based on the processing of the audio raw sensory data collected from a microphone sensor. The audio raw sensory data consists of the user voice data.		
Trigger:	Request for the recognition of the user emotion based on a given audio raw sensory data		
Pre-conditions:	Raw sensory data is extracted from compatible sensory data (audio sensory data)		
Post-conditions:	A label corresponding to the recognized emotion is generated		
Normal Flow:	1. Audio raw sensory data is received for analysis 2. The raw sensory data is preprocessed (e.g., filtered) 3. The preprocessed raw sensory data is segmented (e.g., partitioned into windows) 4. Features (e.g., LPC, MFCC) are extracted from each segment of raw sensory data 5. The extracted features are classified 6. A label identifying the corresponding user emotion is generated		
Alternative Flows:	NA		
Exceptions:	NA		
Includes:	NA		
Frequency of Use:	Frequent: at every reception of inertial raw sensory data		
NFR ID:	MM-NFR-05		
Assumptions:	The raw sensory data is of the nature required by the audio emotion recognizer		
Notes and Issues:	NA		

Sequence Diagram:



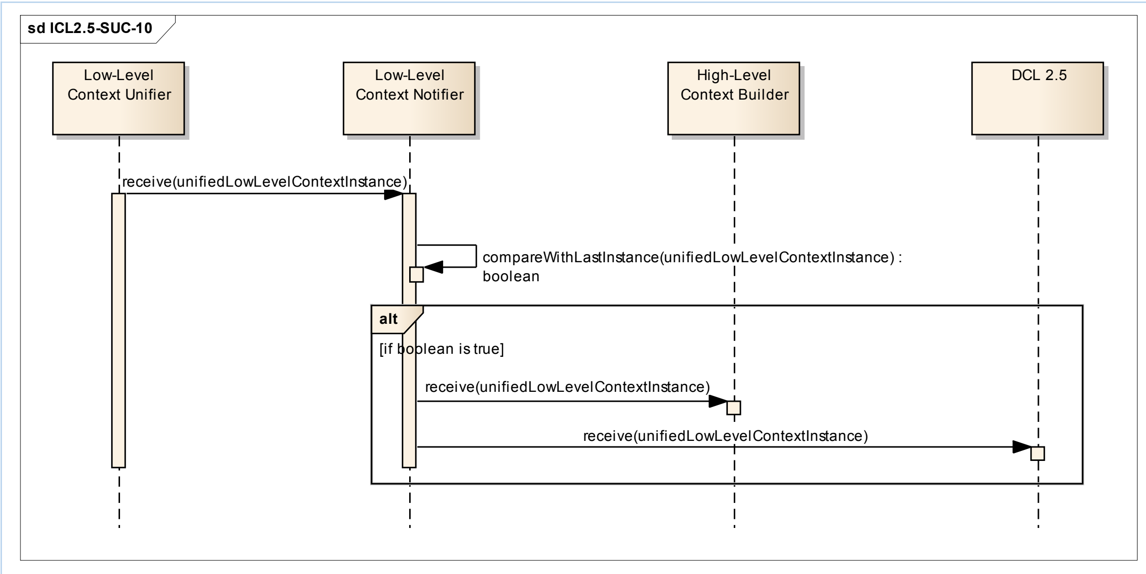
Use Case ID:	*ICL2.5-SUC-09*		
Use Case Name:	Unify low-level contexts		
FR ID:	MM-FR-11		
Created By:	Oresti Banos	**Last Updated By:**	Oresti Banos
Date Created:	20 July 2015	**Last Revision Date:**	20 July 2015
Actors:	Low-Level Context Recognizer, Low-Level Context Notifier		
Description:	Aggregation of multiple low-level context instances of the same context type (e.g., activity) corresponding to a similar period of time		
Trigger:	Receive low-level context instance		
Pre-conditions:	Low-level context instances are received from different recognizers of the same context type		
Post-conditions:	A single low-level context instance is served for notification		
Normal Flow:	1. A low-level context instance is received 2. Search for other low-level context instances of the same type valid at the same time 3. Fuse the identified low-level context instances		

	into a unified low-level context instance 4. Serve the unified low-level context instance for notification
Alternative Flows:	NA
Exceptions:	NA
Includes:	NA
Frequency of Use:	Frequent: at every reception of a low-level context label
NFR ID:	MM-NFR-05
Assumptions:	Identical labels are used to describe the same low-level context for each recognizer of the same context type (e.g., inertial activity recognizer, video activity recognizer)
Notes and Issues:	NA
Sequence Diagram:	



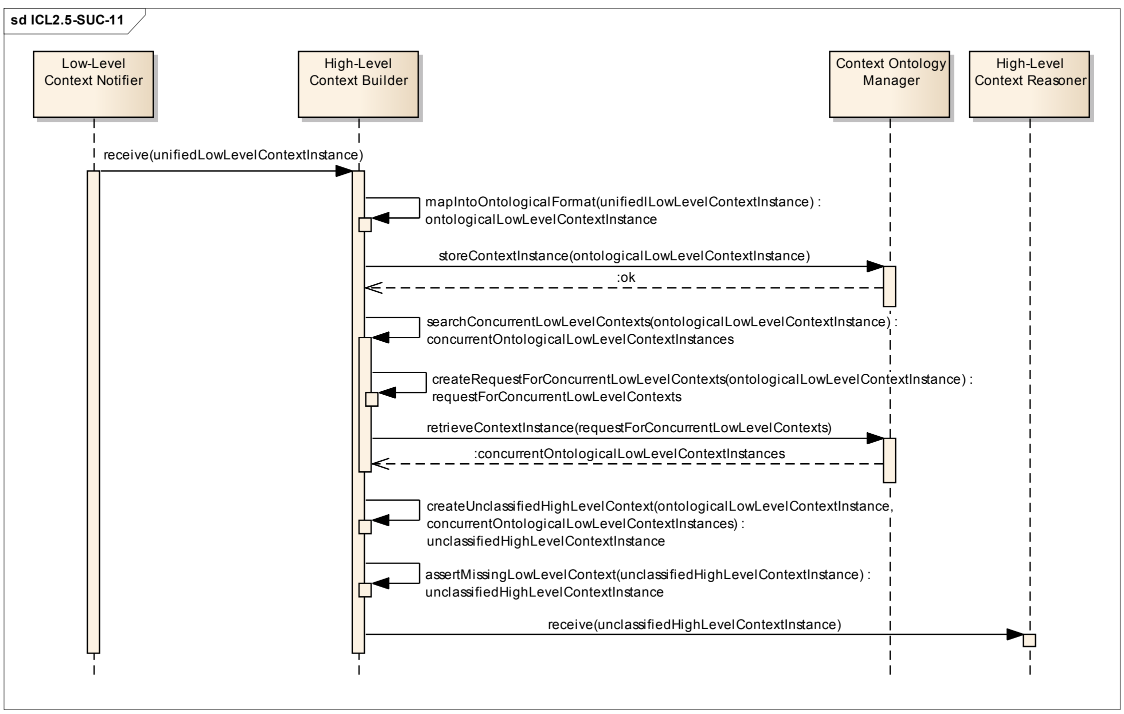
Use Case ID:	*ICL2.5-SUC-10*
Use Case Name:	Notify new low-level context
FR ID:	MM-FR-13

Created By:	Oresti Banos	**Last Updated By:**	Oresti Banos
Date Created:	14 July 2015	**Last Revision Date:**	20 July 2015
Actors:	Low-Level Context Unifier, High-Level Context Builder, DCL 2.5		
Description:	Serve the newly recognized low-level context for the identification of high-level context and also communicate it to DCL 2.5 for persistence.		
Trigger:	New low-level context is identified		
Pre-conditions:	A unified low-level context instance is received		
Post-conditions:	• The unified low-level context instance is served for the identification of the high-level context(s) • The unified low-level context instance is sent to DCL 2.5		
Normal Flow:	1. A low-level context instance is received from the low level context unifier 2. The received instance is compared with the last low-level context instance 3. The new low-level context instance is served for the identification of the high-level context 4. The new low-level context instance is sent to DCL 2.5		
Alternative Flows:	3a. If the received instance contains the same low-level context type as the previous one 1. Finalize		
Exceptions:	NA		
Includes:	NA		
Frequency of Use:	Frequent: at every reception of a low-level context instance		
NFR ID:	NA		
Assumptions:	NA		
Notes and Issues:	NA		
Sequence Diagram:			



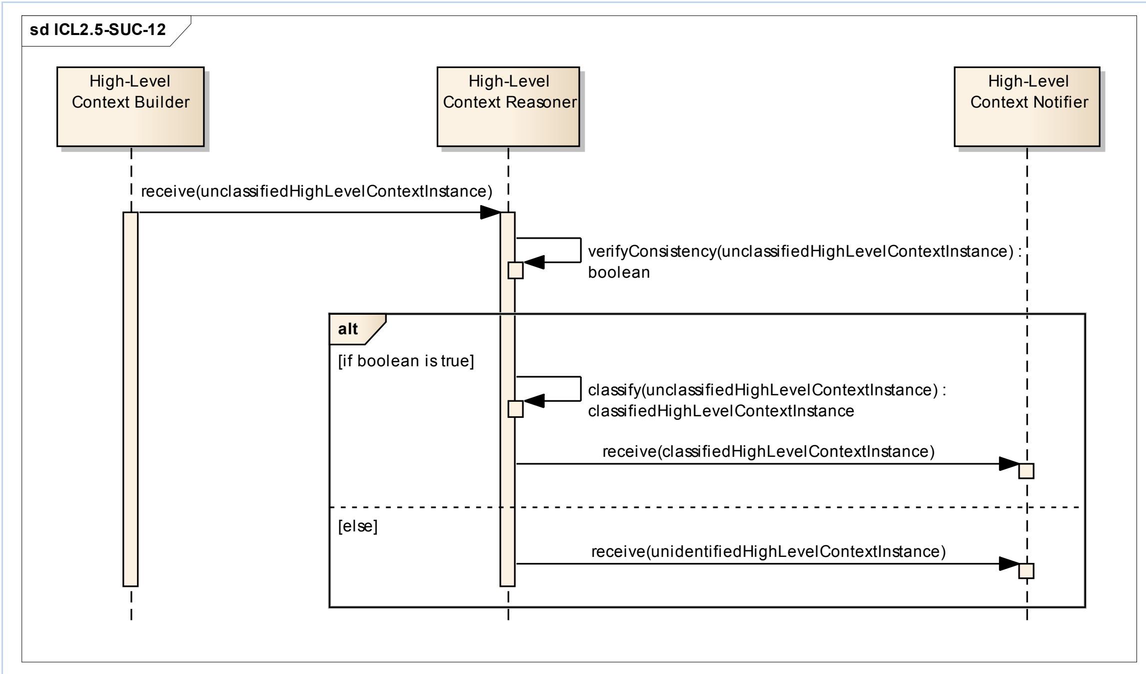
Use Case ID:	*ICL2.5-SUC-11*		
Use Case Name:	Create unclassified high-level context instance		
FR ID:	MM-FR-12		
Created By:	Claudia Villalonga	**Last Updated By:**	Oresti Banos
Date Created:	14 July 2015	**Last Revision Date:**	20 July 2015
Actors:	Low-Level Context Notifier, High-Level Context Reasoner		
Description:	Build a high-level context instance based on the identified low-level contexts		
Trigger:	Receive low-level context instance (label plus metadata)		
Pre-conditions:	A new low-level context instance is served to the high-level context builder		
Post-conditions:	The unclassified high-level context instance is created		
Normal Flow:	1. Map low-level context instance into ontological format 2. Search for other low-level context instances of different type valid at the same time 3. Create new unclassified high-level context instance which links to the available low-level		

	context instance(s) 4. Assert on the unclassified high-level context instance that the missing low-level context instances do not exist
Alternative Flows:	NA
Exceptions:	NA
Includes:	NA
Frequency of Use:	Less Frequent: whenever a new low-level context is recognized
NFR ID:	NA
Assumptions:	Low-level contexts are interpretable
Notes and Issues:	NA
Sequence Diagram:	



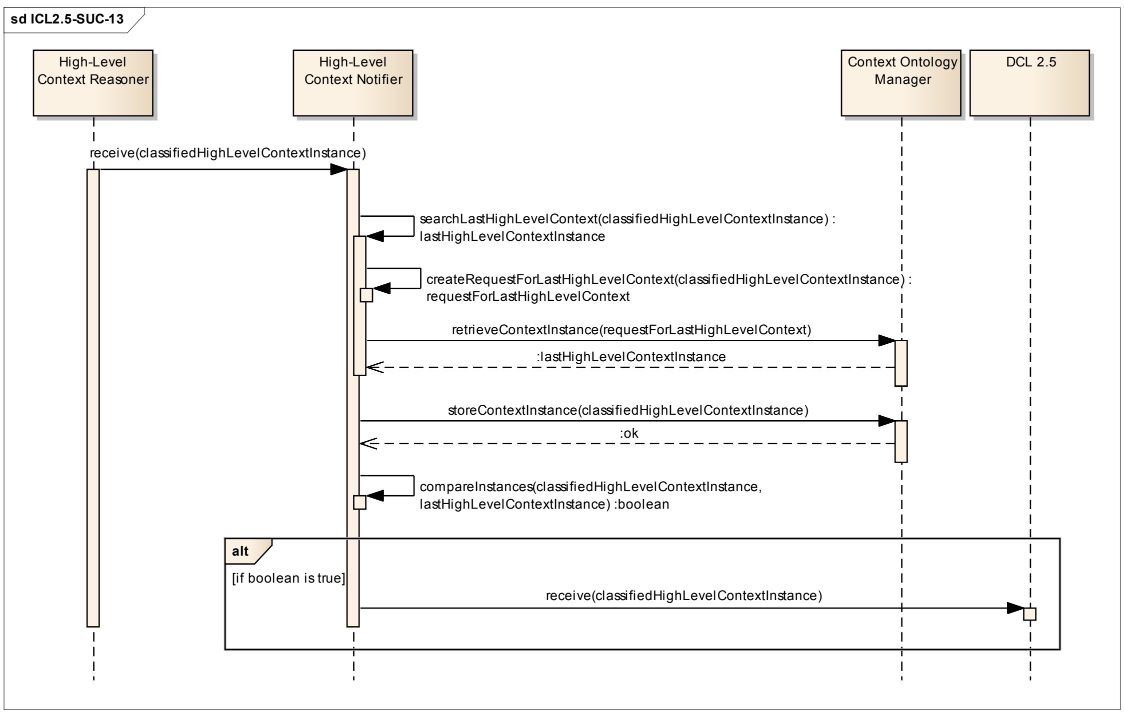
Use Case ID:	*ICL2.5-SUC-12*
Use Case Name:	Classify high-level context instance
FR ID:	MM-FR-12

Created By:	Claudia Villalonga	**Last Updated By:**	Oresti Banos
Date Created:	14 July 2015	**Last Revision Date:**	20 July 2015
Actors:	High-Level Context Builder, High-Level Context Notifier		
Description:	Classify the unclassified high-level context instance into one of the high-level context categories		
Trigger:	Creation of unclassified high-level context instance		
Pre-conditions:	The unclassified high-level context instance is created		
Post-conditions:	The classified high-level context instance is served for notification		
Normal Flow:	1. Verify the consistency of unclassified high-level context instance 2. Reason on the unclassified high-level context instance to identify the context type to which it belongs 3. Serve the classified high-level context for notification		
Alternative Flows:	1a. If the unclassified high-level context instance is not valid 1. Communicate unidentified context		
Exceptions:	NA		
Includes:	NA		
Frequency of Use:	Less Frequent: whenever an unclassified high-level context instance is created		
NFR ID:	MM-NFR-06		
Assumptions:	Low-level contexts and high-level contexts are interpretable		
Notes and Issues:	NA		
Sequence Diagram:			



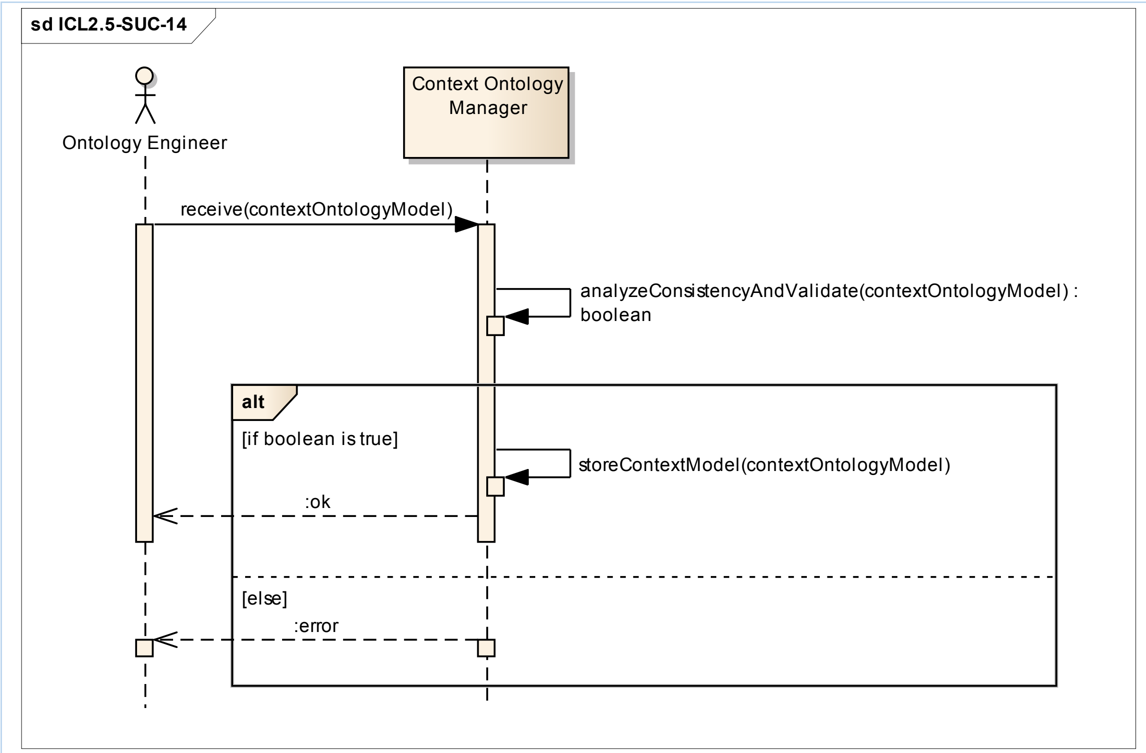
Use Case ID:	*ICL2.5-SUC-13*		
Use Case Name:	Notify new high-level context		
FR ID:	MM-FR-14		
Created By:	Oresti Banos	**Last Updated By:**	Oresti Banos
Date Created:	14 July 2015	**Last Revision Date:**	20 July 2015
Actors:	High-Level Context Reasoner, DCL 2.5		
Description:	Communicate the newly recognized high-level context to DCL 2.5 for persistence.		
Trigger:	High-level context is identified		
Pre-conditions:	A high-level context instance is received		
Post-conditions:	The new high-level context instance is sent to DCL 2.5		
Normal Flow:	1. A high-level context instance is received from the high-level context classifier 2. The received instance is compared with the last high-level context instance 3. The new high-level context instance is sent to DCL 2.5		

Alternative Flows:	2a. If the received instance contains the same high-level context type as the previous one 1. Finalize
Exceptions:	NA
Includes:	NA
Frequency of Use:	Less frequent: at every reception of a high-level context instance
NFR ID:	NA
Assumptions:	NA
Notes and Issues:	NA
Sequence Diagram:	



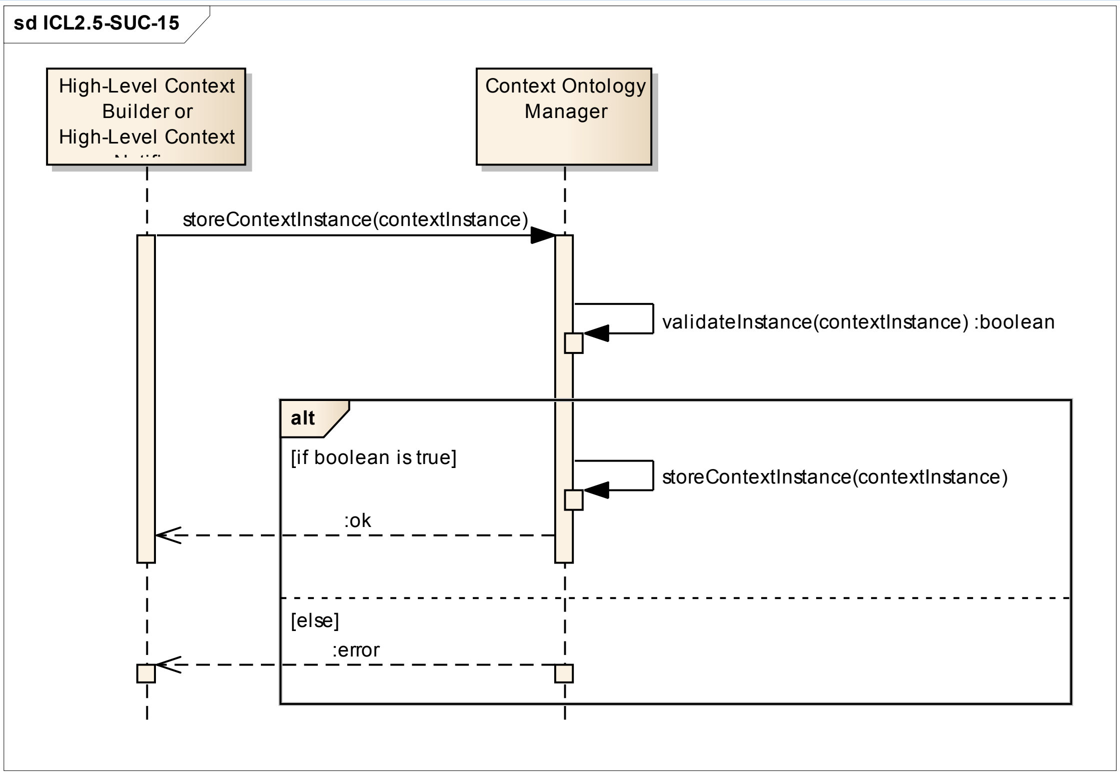
Use Case ID:	*ICL2.5-SUC-14*		
Use Case Name:	Load context ontology model		
FR ID:			
Created By:	Claudia Villalonga	**Last Updated By:**	Claudia Villalonga
Date Created:	31 August 2015	**Last Revision Date:**	3 September 2015
Actors:	Ontology Engineer		

Description:	Load and store a context ontology model that describes high-level context and its relations to low-level context into the system in order to enable the recognition of high-level context
Trigger:	The ontology engineer who has created a context ontology model loads it through the interface
Pre-conditions:	• A context ontology model that describes high-level context and its relations to low-level context has been created
Post-conditions:	• The context ontology model is stored and available for the recognition of high-level context
Normal Flow:	1. A context ontology model is received 2. The context ontology model is analyzed for its consistency and validity 3. The context ontology is stored in order to provide persistence 4. Success is notified
Alternative Flows:	2a. If the context ontology model is not valid or inconsistent 1. Error is notified 3a. If there is an error during storage 1. Repeat step 3
Exceptions:	NA
Includes:	NA
Frequency of Use:	Infrequent
NFR ID:	NA
Assumptions:	NA
Notes and Issues:	NA
Sequence Diagram:	



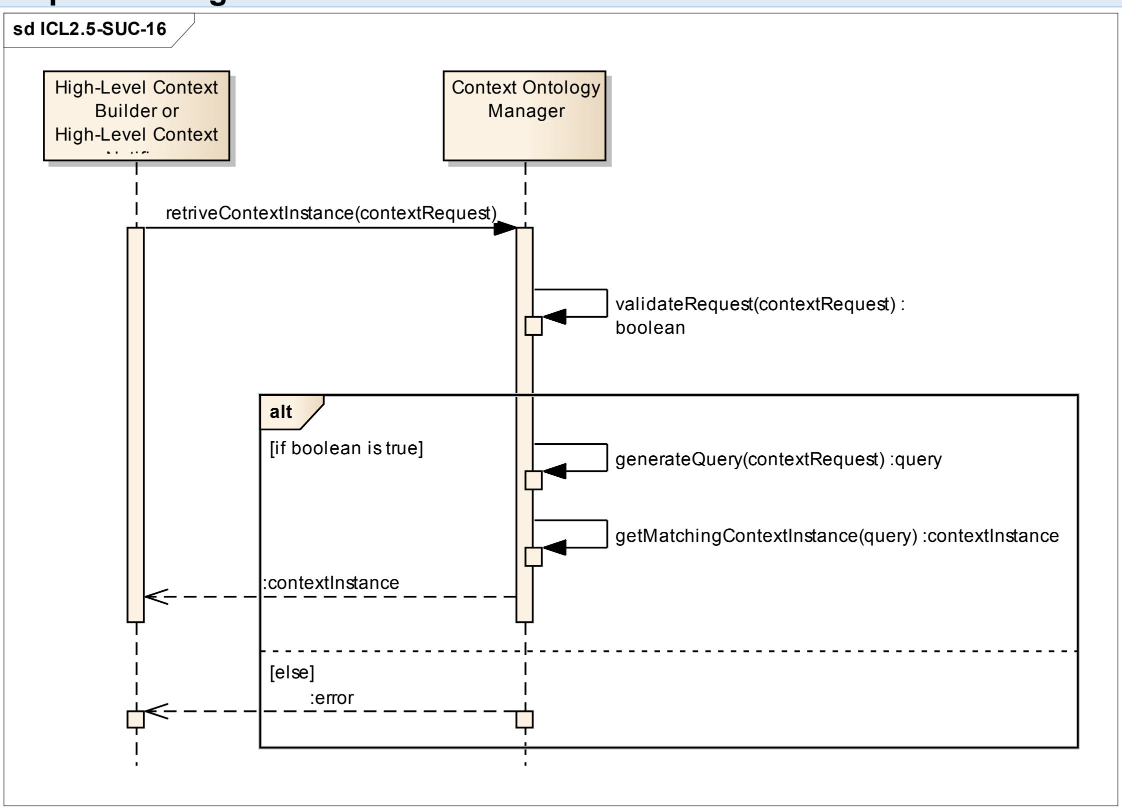
Use Case ID:	*ICL2.5-SUC-15*		
Use Case Name:	Store context instance		
FR ID:			
Created By:	Claudia Villalonga	**Last Updated By:**	Claudia Villalonga
Date Created:	31 August 2015	**Last Revision Date:**	4 September 2015
Actors:	High-Level Context Builder and High-Level Context Notifier		
Description:	Persist a context instance (high-level context instance or low-level context instance) into the system		
Trigger:	A new context instance has been created or identified		
Pre-conditions:	• A new context instance is received		
Post-conditions:	• The context instance is stored		
Normal Flow:	1. Receive context instance 2. Validate the context instance 3. Store the context instance 4. Notify success		
Alternative Flows:	2a. If the context instance is not valid 1. Error is notified 3a. If there is an error during storage		

	1. Repeat step 3
Exceptions:	NA
Includes:	NA
Frequency of Use:	Less Frequent: whenever a new low-level context instance is created, a new high-level context instance is created, or a high-level context instance is classified
NFR ID:	NA
Assumptions:	NA
Notes and Issues:	NA
Sequence Diagram:	

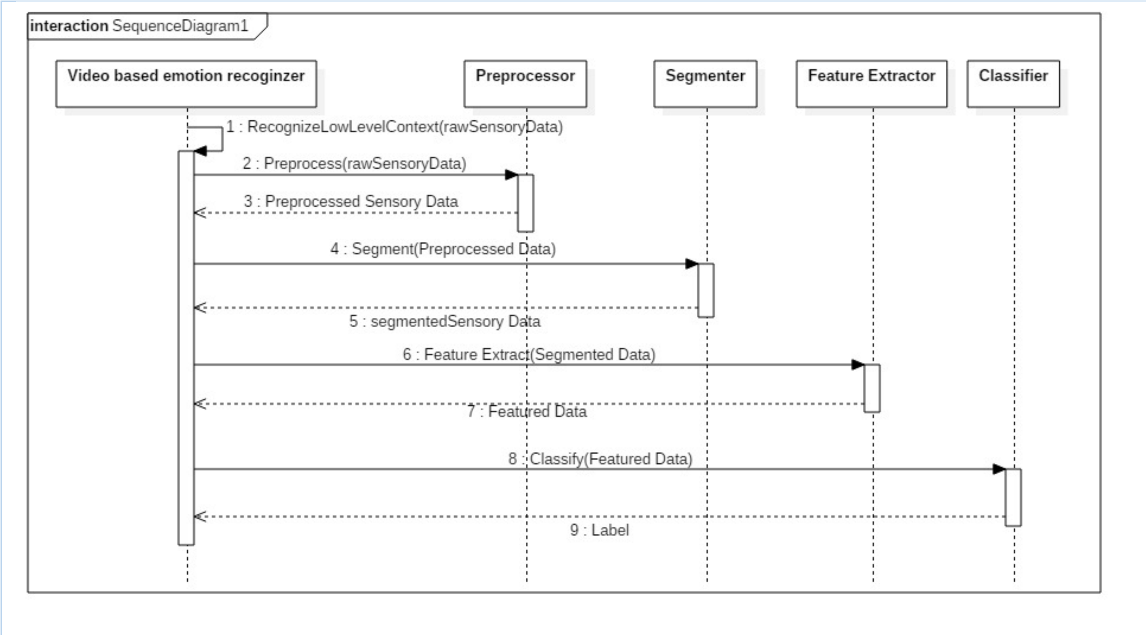


Use Case ID:	*ICL2.5-SUC-16*
Use Case Name:	Retrieve context instance
FR ID:	
Created By:	Claudia Villalonga; **Last Updated By:** Claudia Villalonga
Date Created:	31 August 2015; **Last Revision Date:** 4 September 2015
Actors:	High-Level Context Builder and High-Level Context Notifier
Description:	Provide context instances (high-level context instances or low-level context instances) that match a given request
Trigger:	A requester retrieves context instances
Pre-conditions:	• A request for context instances is received

Post-conditions:	• Matching context instances are provided to the requester
Normal Flow:	1. Receive a request for a specific context 2. Validate the request 3. Generate the query associated to the request 4. Match the query to the stored context instances 5. Return the matching context instances
Alternative Flows:	2.5a. If the request for context is not valid 1. Return error message
Exceptions:	NA
Includes:	NA
Frequency of Use:	Less frequent: whenever context instances are required in order to generate a new high-level context instance or to verify the high-level context
NFR ID:	NA
Assumptions:	NA
Notes and Issues:	NA
Sequence Diagram:	

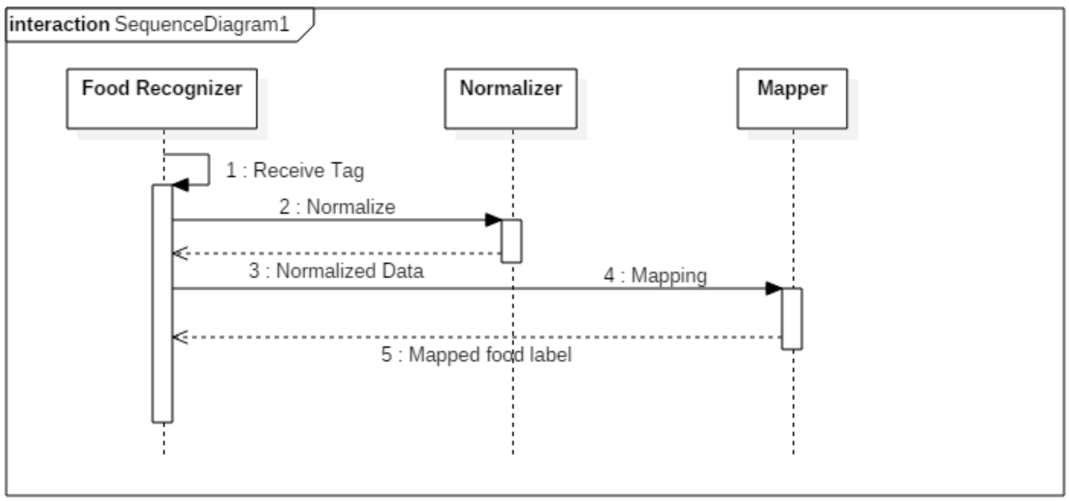


Use Case ID:	*ICL2.5-SUC-17*		
Use Case Name:	Recognize emotion based on video data		
FR ID:	MM-FR-11		
Created By:	Jae Hun, Bang	**Last Updated By:**	Jae Hun, Bang
Date Created:	18 Mar 2016	**Last Revision Date:**	18 Mar 2016
Actors:	Recognize user low-level context		
Description:	Identification of the user emotional state (e.g., “happy”) based on the processing of the video data collected from a camera while user is calling. The video data consist of user facial video collected during the call.		
Trigger:	Request for the recognition of the user emotion based on a given video data		
Pre-conditions:	• Raw sensory data is extracted from compatible sensory data (video sensory data)		
Post-conditions:	• A label corresponding to the recognized emotion is generated		
Normal Flow:	1. Video raw sensory data is received for analysis 2. The raw sensory data is preprocessed (e.g., filtered) 3. The preprocessed raw sensory data is segmented (e.g., partitioned into windows) 4. Features (Facial points) are extracted from each segment of video data (picture) 5. The extracted features are classified with SVM 6. A label identifying the corresponding user emotion is generated		
Alternative Flows:	NA		
Exceptions:	NA		
Includes:	NA		
Frequency of Use:	Frequent: at every reception of inertial raw sensory data		
NFR ID:	MM-NFR-05		
Assumptions:	The raw sensory data (video) is of the nature required by the audio emotion recognizer		
Notes and Issues:	NA		
Sequence Diagram:			



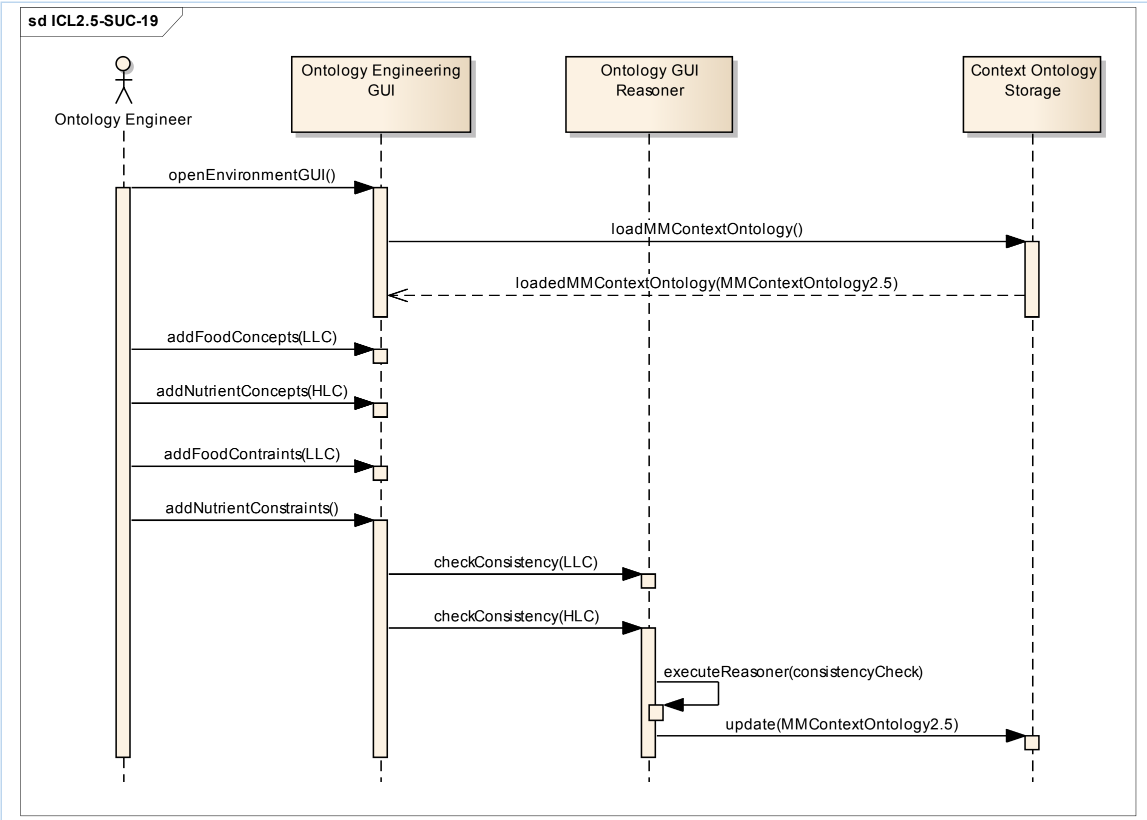
Use Case ID:	*ICL2.5-SUC-18*		
Use Case Name:	Recognize nutrition based on image data		
FR ID:	MM-FR-11		
Created By:	Dong Uk, Kang	**Last Updated By:**	Dong Uk, Kang
Date Created:	18 Mar 2016	**Last Revision Date:**	18 Mar 2016
Actors:	Recognize user low-level context		
Description:	Identification of the user food taken (e.g., “Bibimbap”) based on the tags attached to the image, the user uploaded. The tags consist of textual data, with character ‘#’ as delimiter, and are mapped to the predefined list of food.		
Trigger:	Request for the recognition of the user food taken based on a given tags with user uploaded image		
Pre-conditions:	• Tags uploaded with user food photo is extracted (textual data)		
Post-conditions:	• A label corresponding to the recognized eaten food is generated		
Normal Flow:	1. User entered tags are received for analysis 2. The tags are processed for the normalization 3. The preprocessed tags are mapped to the list of predefined food list 4. The food label is classified, if the mapping relation exists for the entered tags		
Alternative Flows:	4-1. If the mapping does not exists, generate “Tag not identified” message		
Exceptions:	NA		

Includes:	NA
Frequency of Use:	Frequent: at every reception of user tag data
NFR ID:	MM-NFR-05
Assumptions:	The food tags are entered along with the photo by the user
Notes and Issues:	NA
Sequence Diagram:	



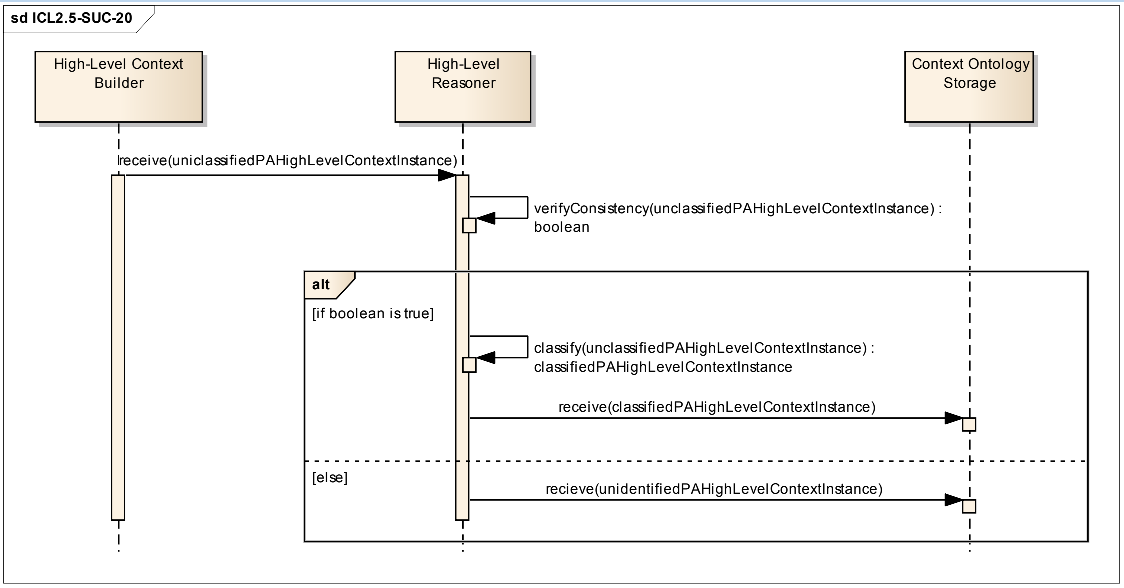
Use Case ID:	*ICL2.5-SUC-19*		
Use Case Name:	Evolve Ontology		
FR ID:	MM-FR-11		
Created By:	Wajahat Ali Khan	**Last Updated By:**	Wajahat Ali Khan
Date Created:	18 Mar 2016	**Last Revision Date:**	18 Mar 2016
Actors:	Ontology Engineer		
Description:	Evolving the already created context ontology based on the nutrition service. The low level and high level nutrition related contexts are modelled in the context ontology that only included physical activities related low level and high level entities.		
Trigger:	The ontology engineer who has created a context ontology model loads it through the interface		
Pre-conditions:	• A context ontology model that describes high-level context and its relations to low-level context has been created with physical activities and nutrition related resources		
Post-conditions:	• The modified or evolved context ontology model is stored and available for the recognition of high-level context		

Normal Flow:	1. Ontology engineer defines the low level nutrition concepts in addition to the low level physical activities concepts 2. Ontology engineer defines the high level nutrition concepts in addition to the high level physical activities concepts 3. The constraints are defined on the new concepts added to the ontology 4. Consistency of the ontology is checked with reasoner 5. The modified context ontology is stored for persistence 6. Success is notified
Alternative Flows:	4a. If the context ontology model is not valid or inconsistent 1. Error is notified 2. Check step 1,2,3
Exceptions:	NA
Includes:	NA
Frequency of Use:	Infrequent
NFR ID:	NA
Assumptions:	Context Ontology based on physical activities service already exists
Notes and Issues:	NA
Sequence Diagram:	



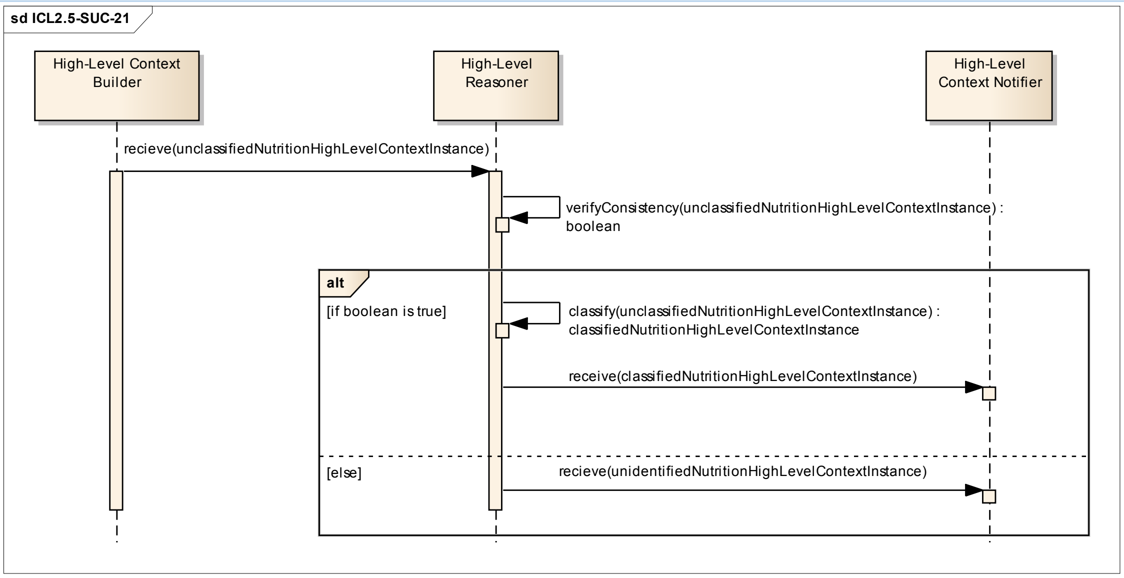
Use Case ID:	*ICL2.5-SUC-20*		
Use Case Name:	High level physical activity context		
FR ID:	MM-FR-14		
Created By:	Wajahat Ali Khan	**Last Updated By:**	Wajahat Ali Khan
Date Created:	18 March 2016	**Last Revision Date:**	18 March 2016
Actors:	High-Level Context Reasoner		
Description:	A classified high level physical activity context is recognized by the reasoner from the unclassified high level context.		
Trigger:	High-level context is identified		
Pre-conditions:	• An unclassified high-level physical activity context instance is received		
Post-conditions:	• The new high-level physical activity context instance is forwarded to Notify DCL		
Normal Flow:	1. A high-level physical activity context instance is processed by high-level context classifier 2. Reasoning is performed by the reasoner to find out classified high level physical activity context 3. The classified high level physical activity context is provided to Notify DCL		
Alternative Flows:	NA		

Exceptions:	NA
Includes:	NA
Frequency of Use:	Less frequent: at every reception of a unclassified high-level physical activity context instance
NFR ID:	NA
Assumptions:	NA
Notes and Issues:	NA
Sequence Diagram:	



Use Case ID:	*ICL2.5-SUC-21*		
Use Case Name:	High level nutrition context		
FR ID:	MM-FR-14		
Created By:	Wajahat Ali Khan	**Last Updated By:**	Wajahat Ali Khan
Date Created:	18 March 2016	**Last Revision Date:**	18 March 2016
Actors:	High-Level Context Reasoner		
Description:	A classified high level nutrition context is recognized by the reasoner from the unclassified high level context.		
Trigger:	High-level context is identified		
Pre-conditions:	• An unclassified high-level nutrition context instance is received		
Post-conditions:	• The new high-level nutrition context instance is forwarded to Notify DCL		
Normal Flow:	4. A high-level nutrition context instance is processed by high-level context classifier 5. Reasoning is performed by the reasoner to find out classified high level nutrition context 6. The classified high level nutrition context is		

	provided to Notify DCL
Alternative Flows:	NA
Exceptions:	NA
Includes:	NA
Frequency of Use:	Less frequent: at every reception of a unclassified high-level nutrition context instance
NFR ID:	NA
Assumptions:	NA
Notes and Issues:	NA
Sequence Diagram:	



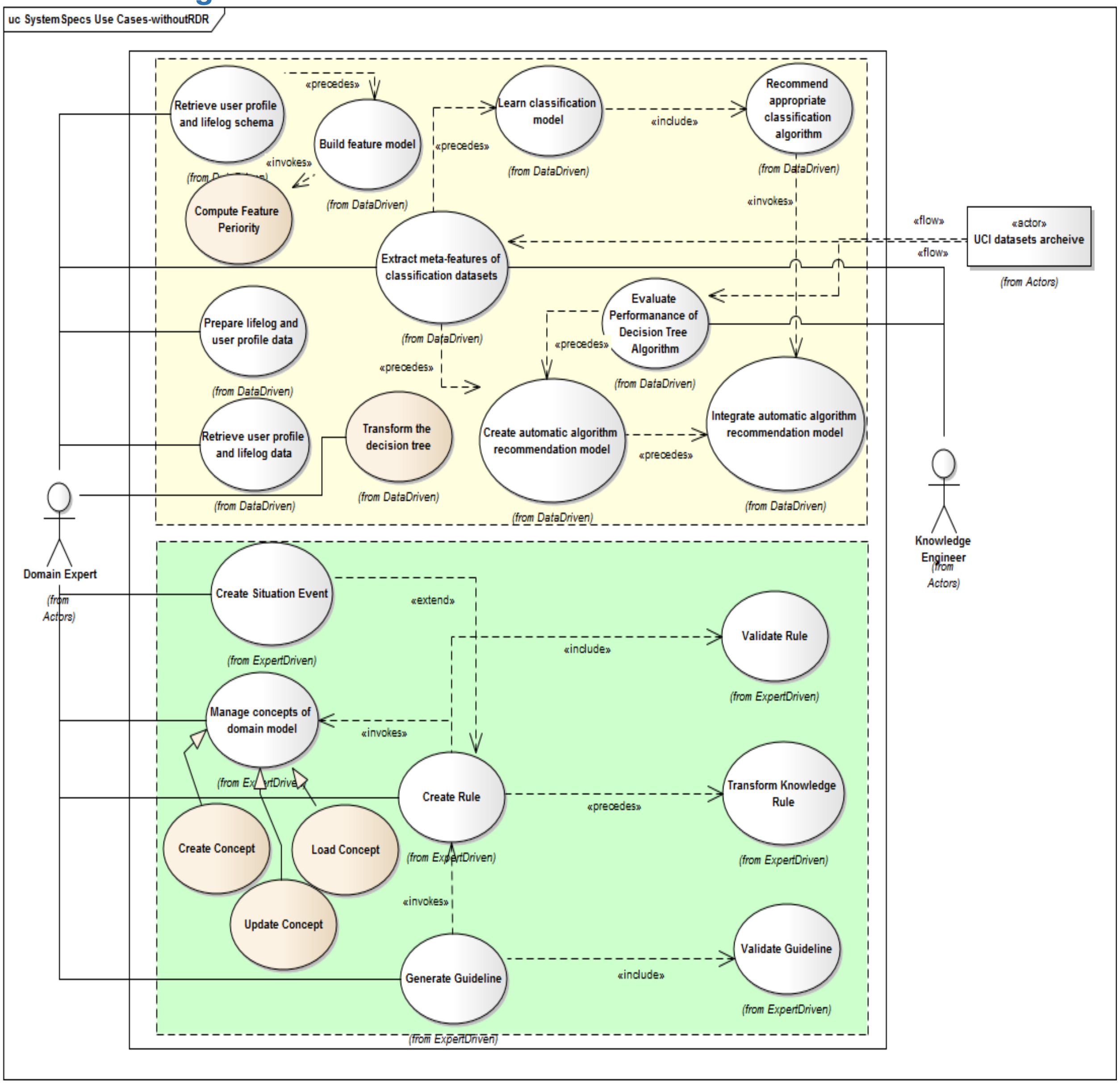
Knowledge Curation Layer (KCL Ver. 2.5)

System Level Use cases

List of Use cases

Requirements #ID	Description
KCL2.5-SUC-01	Select valid combinations of features from lifelog and user profile schema to build feature model for yielding correct classification model.
KCL2.5-SUC-02	Apply preprocessing methods on retrieved lifelog and user profile data to prepare the data for classification model learning.
KCL2.5-SUC-03	Expert generate guidelines to utilize their practices to create rules in the knowledge bases.
KCL2.5-SUC-04	System validate the guidelines in tree structure to maintain the rules.
KCL2.5-SUC-05	User profile and lifelog schema is needed to be known before feature modeling and creation of classification model.
KCL2.5-SUC-06	Retrieve user profile and lifelog data for creation of classification model.
KCL2.5-SUC-07	Extract meta-features of classification datasets.
KCL2.5-SUC-08	Evaluate performance of decision tree algorithms (i.e, f-measure)
KCL2.5-SUC-09	Create automatic algorithm recommendation model (AARM) from offline datasets. AARM will be used as recommendation model for algorithm selection.
KCL2.5-SUC-10	Create rules to enhance the knowledge base of the system to generate recommendations in easy manner.
KCL2.5-SUC-11	Rule validation avoid the duplication of rules in the knowledge base and enhance the maintainability of knowledge base.
KCL2.5-SUC-12	It integrates AARM dataset in Mining Minds Data Driven knowledge acquisition approach for recommendation of automatic algorithm on given dataset.
KCL2.5-SUC-13	It generates classification model from user profile lifelog data that can be explored by model learning mechanism with the help of learning method as well as processed data.
KCL2.5-SUC-14	The integrated AARM shall automatically recommend appropriate classification algorithm. Or domain expert can select any algorithm from available set of decision tree algorithms.
KCL2.5-SUC-15	Domain model is used in creation of rule. It manages the domain model for creating rule.
KCL2.5-SUC-16	It transforms the rules or guidelines into executable knowledge representation.
KCL2.5-SUC-17	It creates situation event and index the rule based on situation event.
KCL2.5-SUC-18	Compute features priorities to help the domain expert for selecting appropriate features from available schema
KCL2.5-SUC-19	Transform the decision tree generated from classification model to conform the rules from domain expert
KCL2.5-SUC-20	Domain model is used in creation of rule. It manages the domain model for creating rule.
KCL2.5-SUC-21	Domain model is also managed by expert. It provides updating of models.
KCL2.5-SUC-22	It provides loading model to Domain Model Manager or Rule Editor.

Use case Diagram

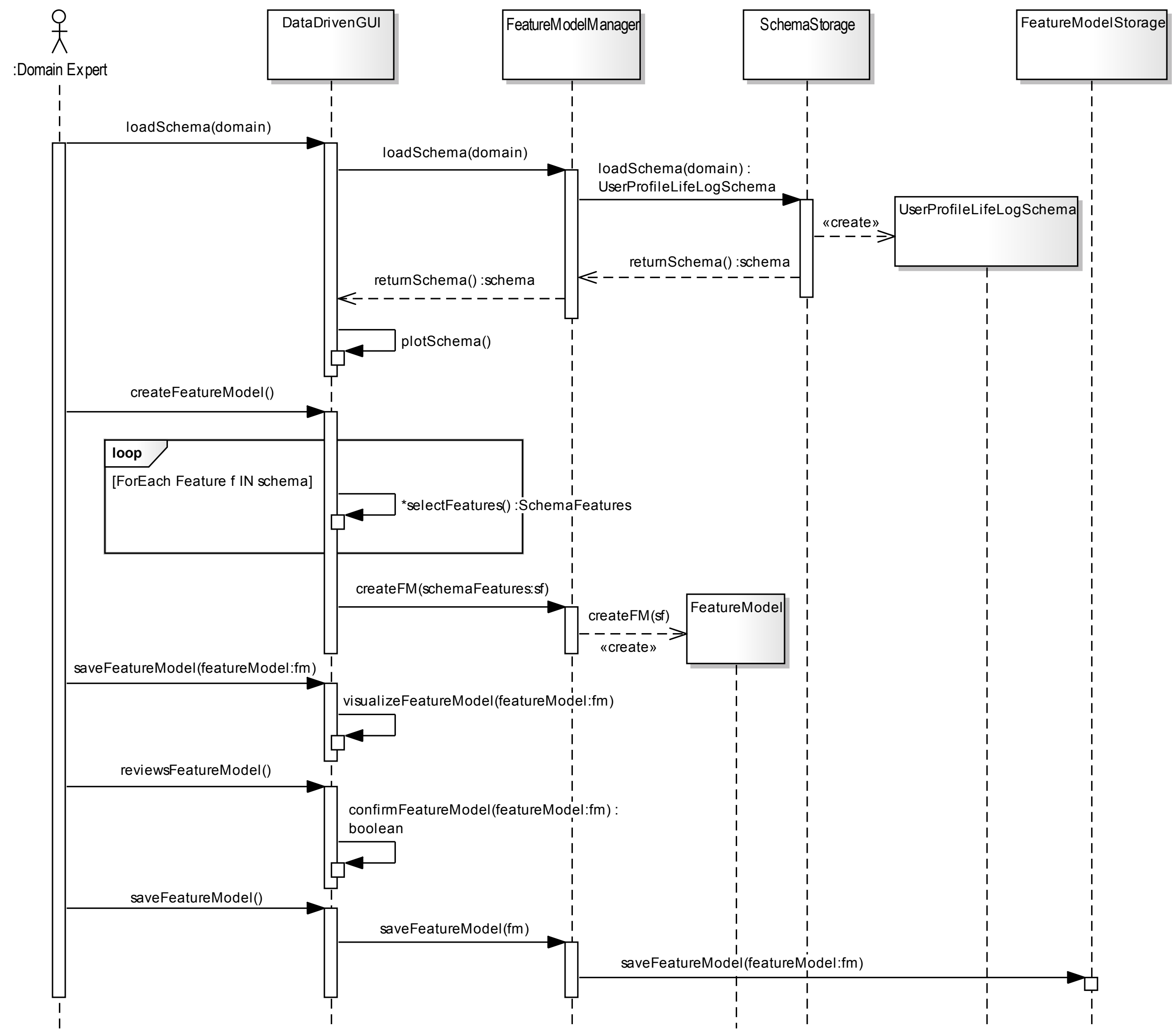


Use case Description

Use Case ID:	**KCL2.5-SUC-01**		
Use Case Name:	Build feature model		
Created By:	Maqbool Ali	**Last Updated By:**	Maqbool Hussain
Date Created:	11-07-2015	**Last Revision Date:**	15-04-2016

Actors:	Domain Expert
Description:	A feature model defines the valid combinations of features in a domain that enables capturing feature variability and interdependencies. For building feature model and its reusability, domain expert uses selected domain schema (i.e. lifelog and user profile schema) and selects the related features for final feature model.
Trigger:	Prior to classification model creation needed for required domain
Preconditions:	• System has retrieved the schema from DCL 2.5. • Domain expert has selected domain under consideration (e.g. nutrition).
Postconditions:	System will build the feature model
Normal Flow:	1. Domain expert retrieves the schema from schema storage. 2. System loads and plots the schema 3. Domain expert builds the feature models as follows: a. Select the required features for corresponding domain b. Verify the consistency of the selected features (such as concept hierarchy) c. Save the feature model 4. System creates the feature model based on selected features and visualizes it in hierarchical form 5. Domain expert reviews the feature model and confirms it for saving into repository 6. System persists the feature model into repository.
Alternative Flows:	N/A
Exceptions:	
Includes:	N/A
Frequency of Use:	When new service is required and mining mind have sufficient data for classification model creation
Special Requirements:	N/A
Assumptions:	Initially we assume that feature model is valid
Notes and Issues:	One possible candidate representation for feature model is XML.

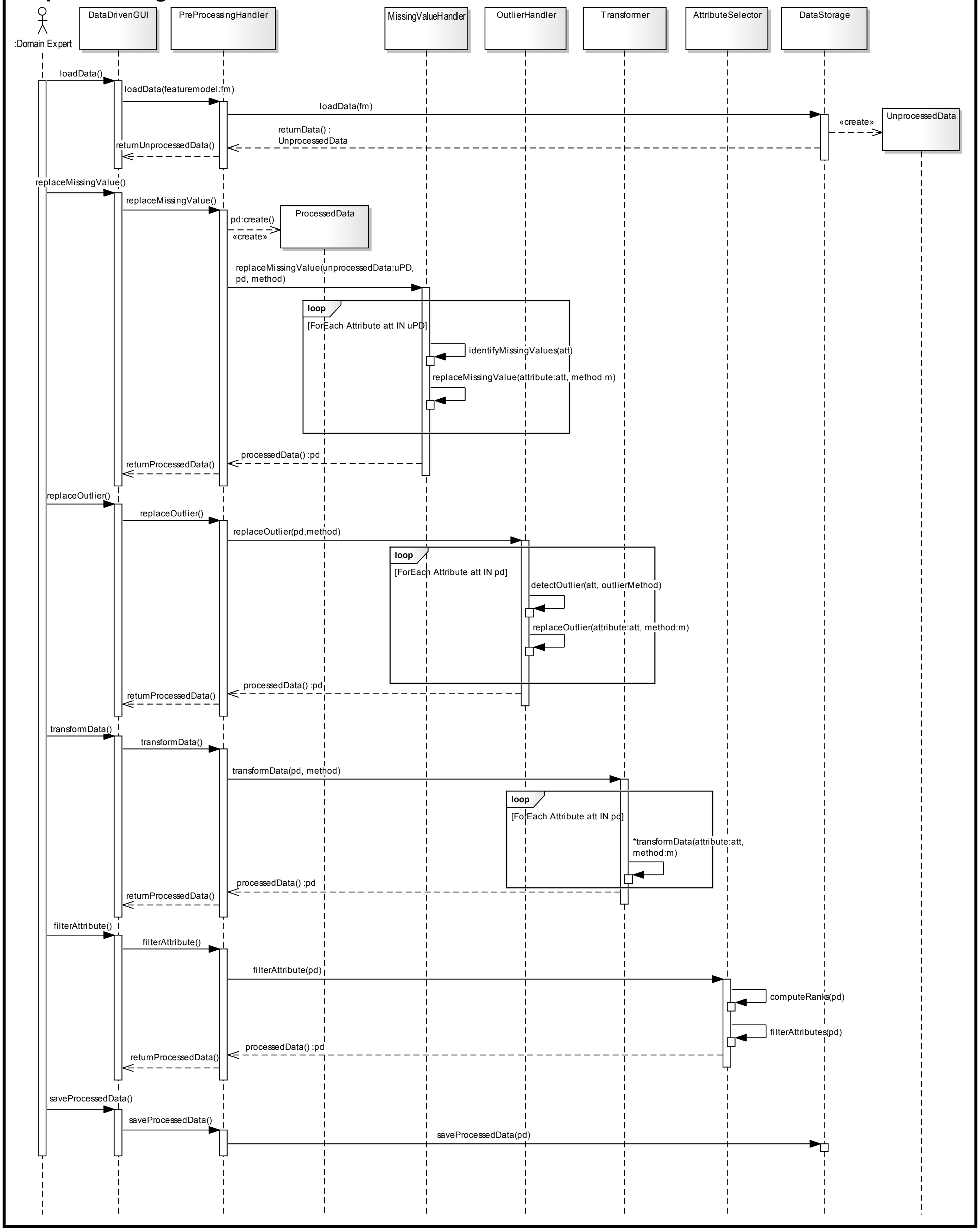
Sequence Diagram:



<table>
<tr><td>Use Case ID:</td><td colspan="3">KCL2.5-SUC-02</td></tr>
<tr><td>Use Case Name:</td><td colspan="3">Prepare lifelog and user profile data</td></tr>
<tr><td>Created By:</td><td>Maqbool Ali</td><td>Last Updated By:</td><td>Maqbool Hussain</td></tr>
<tr><td>Date Created:</td><td>10-07-2015</td><td>Last Revision Date:</td><td>15-04-2016</td></tr>
<tr><td>Actors:</td><td colspan="3">Domain Expert</td></tr>
<tr><td>Description:</td><td colspan="3">It is important to pre-process the data (i.e. lifelog and user profile data) to generate models with high accuracy. ‘Prepare lifelog and user profile data’ use case apply various pre-processing techniques such as missing value handling, outlier detection, transformation, and features selections to convert unprocessed data into processed data.</td></tr>
<tr><td>Trigger:</td><td colspan="3">Prior to classification model creation needed for high accuracy of model learning</td></tr>
<tr><td>Preconditions:</td><td colspan="3">• System has retrieved the data from DCL2, which is unprocessed data.</td></tr>
<tr><td>Postconditions:</td><td colspan="3">System will prepare and store the data</td></tr>
<tr><td>Normal Flow:</td><td colspan="3">1. Domain expert loads the unprocessed data
2. System displays the retrieved data
3. For each attribute:
a. Domain expert identifies the missing values and select appropriate method from following options for missing value replacement.
▪ Default value
▪ Mean
▪ Mode
b. System replaces the missing values using selected method.
4. For each attribute:
a. Domain expert apply outlier detection method such as interquartile range and scatterplot.
b. System display the outliers
c. Domain expert select appropriate method from following options for outlier replacement.
▪ Mean
▪ Mode
d. System replace the outlier using selected method.
5. For each attribute:
a. Domain expert identifies, normalizes the non-transformed values, and updates the dataset.
b. System modifies the values set and update the dataset
6. Domain expert applies the attributes filtration techniques (i.e ranking)
7. System computes the ranks for all attributes and displays to expert
8. Domain expert select the highly ranked attributes (i.e. rank value >= 0.8)
9. System filters the attributes based on selected attributes and displays</td></tr>
</table>

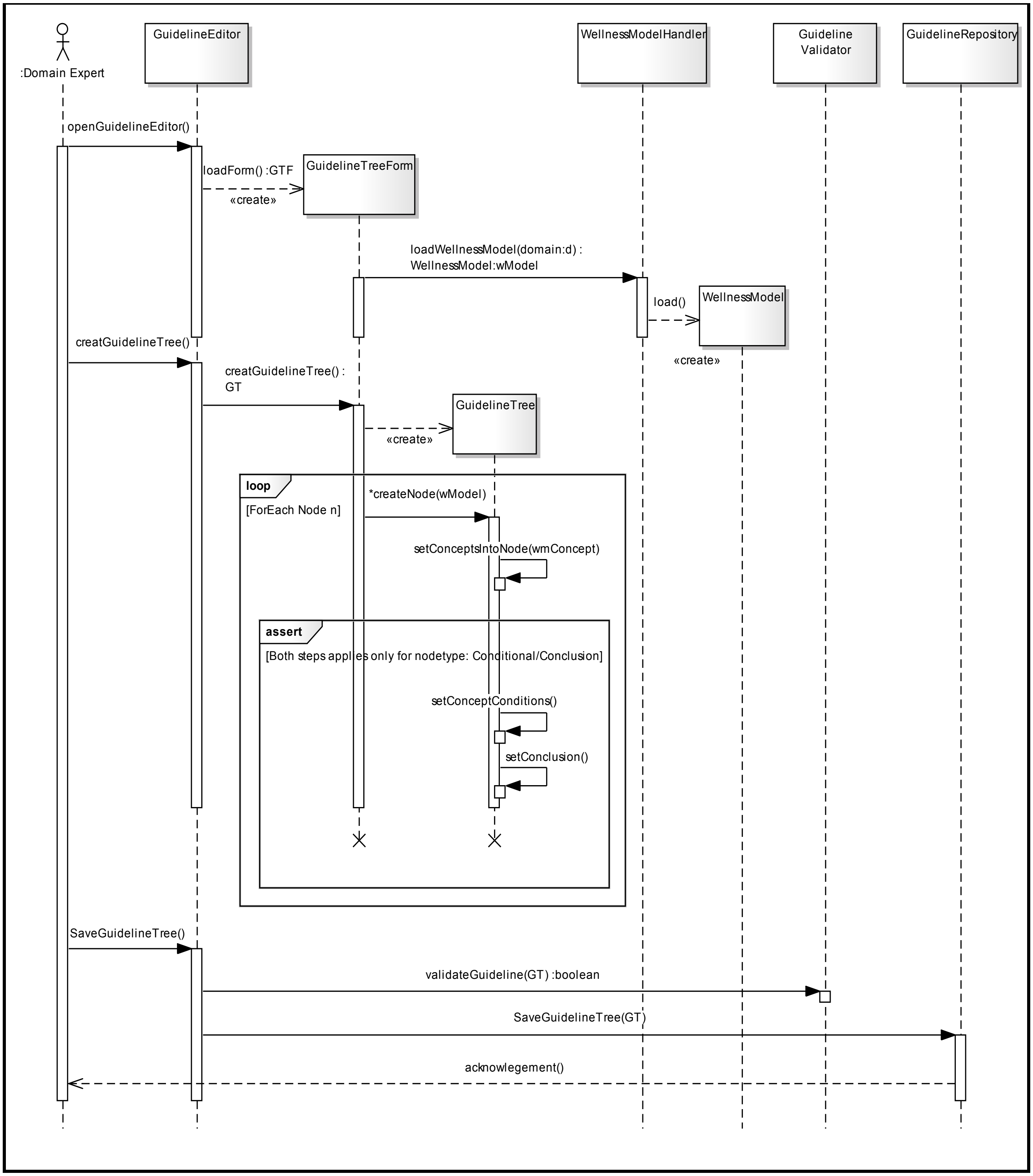
	to domain expert 10. Domain expert saves the processed data into repository 11. System persists the processed data into repository
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Frequency of Use:	When new service is required and mining mind have sufficient data for classification model creation
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	• An **outlier** is any value that is numerically distant from most of the other data points in a set of data. It can be detected by histograms, scatterplots, or interquartile range techniques. • **Data transformation** is the process to convert and normalize the data from one format to another. It can be done by Log, square root, or arcsine transformation techniques.

Sequence Diagram:



<table>
<tr><td>Use Case ID:</td><td colspan="3">KCL2.5-SUC-03</td></tr>
<tr><td>Use Case Name:</td><td colspan="3">Generate Guideline</td></tr>
<tr><td>Created By:</td><td>Taqdir Ali and Maqbool Hussain</td><td>Last Updated By:</td><td>Taqdir Ali, Maqbool Hussain</td></tr>
<tr><td>Date Created:</td><td>11-07-2015</td><td>Last Revision Date:</td><td>15-04-2016</td></tr>
<tr><td>Actors:</td><td colspan="3">Domain Expert</td></tr>
<tr><td>Description:</td><td colspan="3">Guidelines are the combination of one or multiple rules in form of decision tree. The tree format guidelines are understandable to the domain experts and it can easily interpret and execute by computer.</td></tr>
<tr><td>Trigger:</td><td colspan="3">Whenever domain expert wants to generate new guideline or update the existing one.</td></tr>
<tr><td>Preconditions:</td><td colspan="3">1. The domain expert shall be authenticated with full access on the guideline management.
2. Domain expert shall have existing guidelines as reference for generating guideline tree.</td></tr>
<tr><td>Postconditions:</td><td colspan="3">• The expert shall generate guidelines to acquire their knowledge into the system.</td></tr>
<tr><td>Normal Flow:</td><td colspan="3">1. Domain expert opens the guidelines editor.
2. System displays new guideline tree form and load the wellness model.
3. Domain expert selects/drags tree node into editor form.
4. System display the node and open the corresponding properties window, which includes;
a. Loads wellness model tree.
b. Displays operators, relationships and node type (conditional, conclusion or both) artifacts.
5. Domain expert selects concepts for the node using any of the following methods and confirm to save the node.
a. Using wellness model, dragging concepts and facts into node conditional or conclusion part.
b. Using auto pop-up Intelli-sense window to select concepts and facts into conditional or conclusion part.
6. System saves the tree node and displays as part of the guideline tree.
7. Domain expert add other nodes to guideline tree by using step repeating step 3 on ward. After completion, (s)he saves the guideline tree.
8. System validates the guideline tree using “KCL2.5-SUC-04” and save into guideline repository.</td></tr>
<tr><td>Alternative Flows:</td><td colspan="3">2a. System loads existing guideline tree for modification (modifying existing or adding new node).
a. Domain expert selects existing node in guideline tree or drag new node to appropriate place in guideline tree.
b. To modify node, step 3 onward will be invoked in Normal Flow.</td></tr>
</table>

Exceptions:	N/A
Includes:	Validate Guideline
Frequency of Use:	Whenever domain expert want to create new guideline tree or update existing guideline tree.
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	• Guidelines tree created will base on existing guidelines of corresponding domain and domain expert shall interpret textual guidelines into tree format. • Appropriate modelling of guideline is challenging task.
Sequence Diagram:	

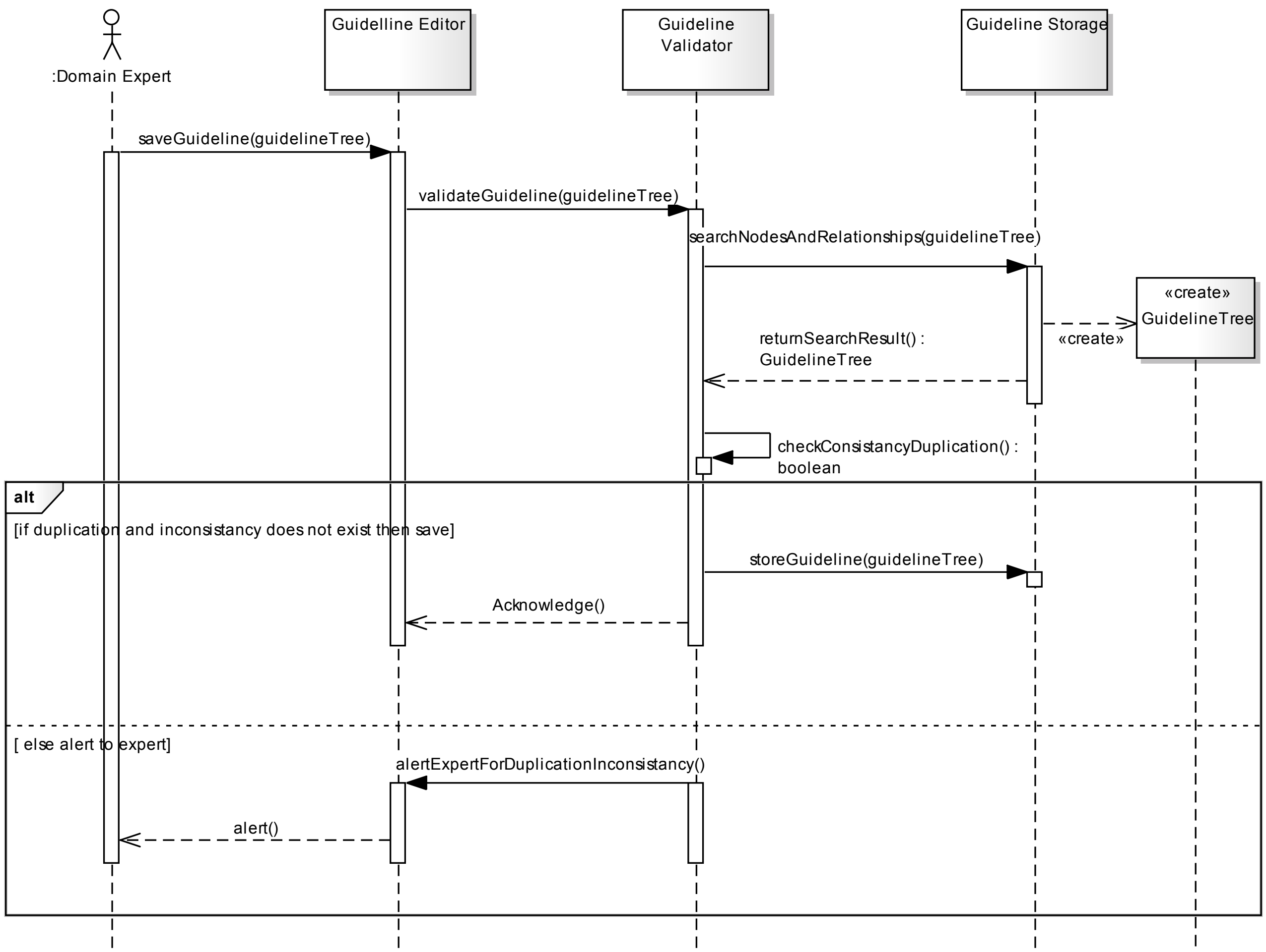


Use Case ID:	**KCL2.5-SUC-04**
Use Case Name:	Validate Guideline

Created By:	Taqdir Ali and Maqbool Hussain	**Last Updated By:**	Taqdir Ali, Maqbool Hussain
Date Created:	11-07-2015	**Last Revision Date:**	15-04-2016
Actors:	Domain Expert		
Description:	Guidelines have different facts and conclusions in form of nodes related with different relationships. The guidelines tree shall be validated for the possible duplication.		
Trigger:	Whenever domain expert wants to generate new guideline or update the existing one.		
Preconditions:	1. System shall be running 2. The domain expert shall be authenticated with full access on the guideline management.		
Postconditions:	• Validated guideline tree		
Normal Flow:	1. Domain Expert save the new guideline or update the existing guideline. 2. The system validate guideline for inconsistency and duplication as follows. a. Fetch the existing guidelines and process each node and relationship b. Guideline Tree is approved for having no inconsistency and duplication of new nodes and relationships of the facts and conclusion with the existing guidelines. c. Guideline Tree is stored into guidelines repository. d. Acknowledge the expert to save guideline successfully.		
Alternative Flows:	2b. Guideline Tree is found having inconsistency or duplication with existing guideline tree a. The system produces alert the inconsistency or duplication in guideline tree b. Domain expert review the alert message and correct the guideline tree. c. Step 1 and Step 2 of normal flow is executed.		
Exceptions:	N/A		
Includes:	N/A		
Frequency of Use:	Whenever domain expert want to create new guideline or update existing guideline.		
Special Requirements:	N/A		
Assumptions:	N/A		

Notes and Issues:	Appropriate modelling of guideline and validation is challenging task.

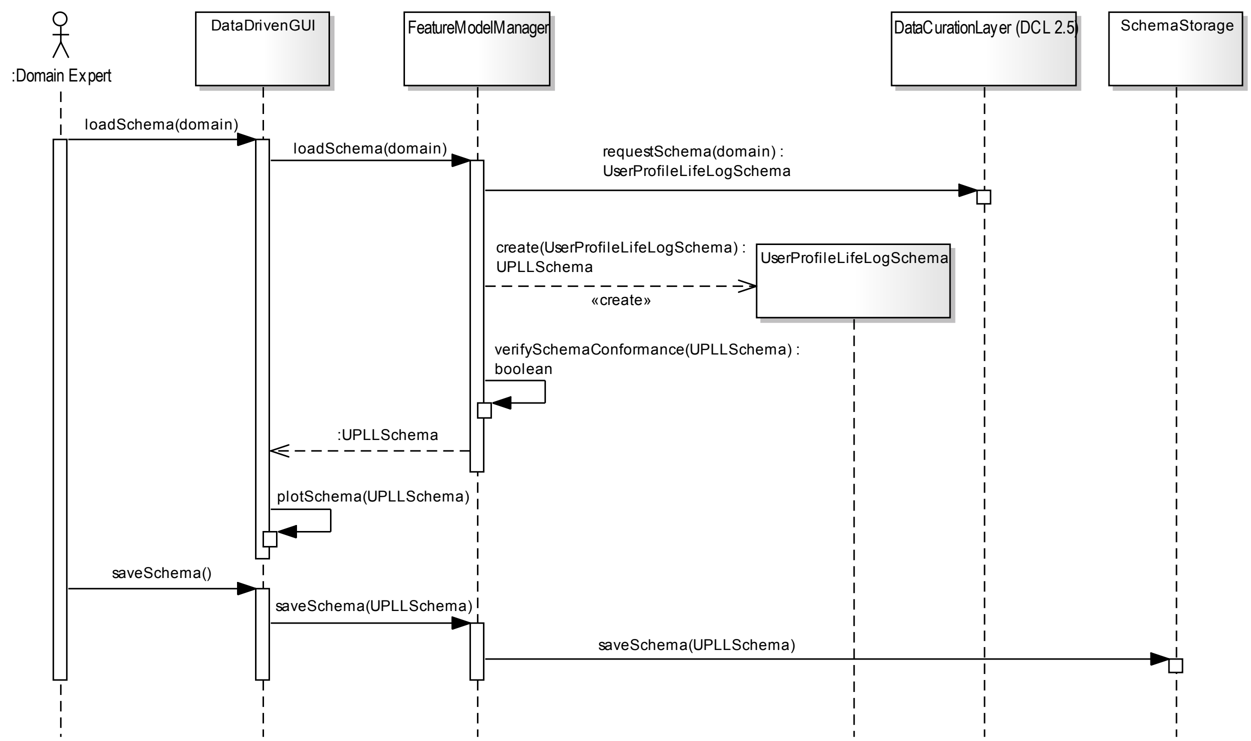
Sequence Diagram:



<table>
<tr><td>Use Case ID:</td><td colspan="3">KCL2.5-SUC-05</td></tr>
<tr><td>Use Case Name:</td><td colspan="3">Retrieve user profile and lifelog schema</td></tr>
<tr><td>Created By:</td><td>Maqbool Ali</td><td>Last Updated By:</td><td>Maqbool Hussain</td></tr>
<tr><td>Date Created:</td><td>10-07-2015</td><td>Last Revision Date:</td><td>15-04-2016</td></tr>
<tr><td>Actors:</td><td colspan="3">Domain Expert, DCL 2.5</td></tr>
<tr><td>Description:</td><td colspan="3">User profile and lifelog schema retrieval help domain expert to view all available features for building feature model.</td></tr>
<tr><td>Trigger:</td><td colspan="3">Prior to classification model creation needed for required domain</td></tr>
<tr><td>Preconditions:</td><td colspan="3">• System has access through service interface to retrieve user profile and lifelog schema from DCL 2.5
• System and DCL 2.5 has agreement on common schema representation format
• DCL 2.5 has capability to share user profile and lifelog schema in secure environment.</td></tr>
<tr><td>Postconditions:</td><td colspan="3">System will receive user profile and lifelog schema conform to its representation scheme.</td></tr>
<tr><td>Normal Flow:</td><td colspan="3">1. Domain expert selects the domain and sends requests to DCL 2.5 for user profile and lifelog schema.
2. DCL 2.5 shares the user profile and lifelog schema
3. System receives the user profile and lifelog schema
4. Domain expert uses the system and performs the following tasks;
a. Verifies the conformance of received schema
b. Plots the verified schema
c. Saves the verified schema
5. System saves the verified schema</td></tr>
<tr><td>Alternative Flows:</td><td colspan="3">N/A</td></tr>
<tr><td>Exceptions:</td><td colspan="3">1a. System unable to connect to DCL 2.5
a. System connection is failed during retrieving user profile and lifelog schema
b. System hold and will retry after sometime to connect to DCL 2.5 and retrieve the user profile and lifelog schema
4a. System unable to verify lifelog schema conformance
a. System fail to conform the schema representation from DCL 2.5
b. System will send message to DCL 2.5 about incompatible schema format</td></tr>
<tr><td>Includes:</td><td colspan="3">N/A</td></tr>
<tr><td>Frequency of Use:</td><td colspan="3">When new service is required and mining mind have sufficient data for classification model creation</td></tr>
</table>

Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	If DCL 2.5 is unable to send user profile and lifelog schema in required format, then alternate strategy has to be considered.

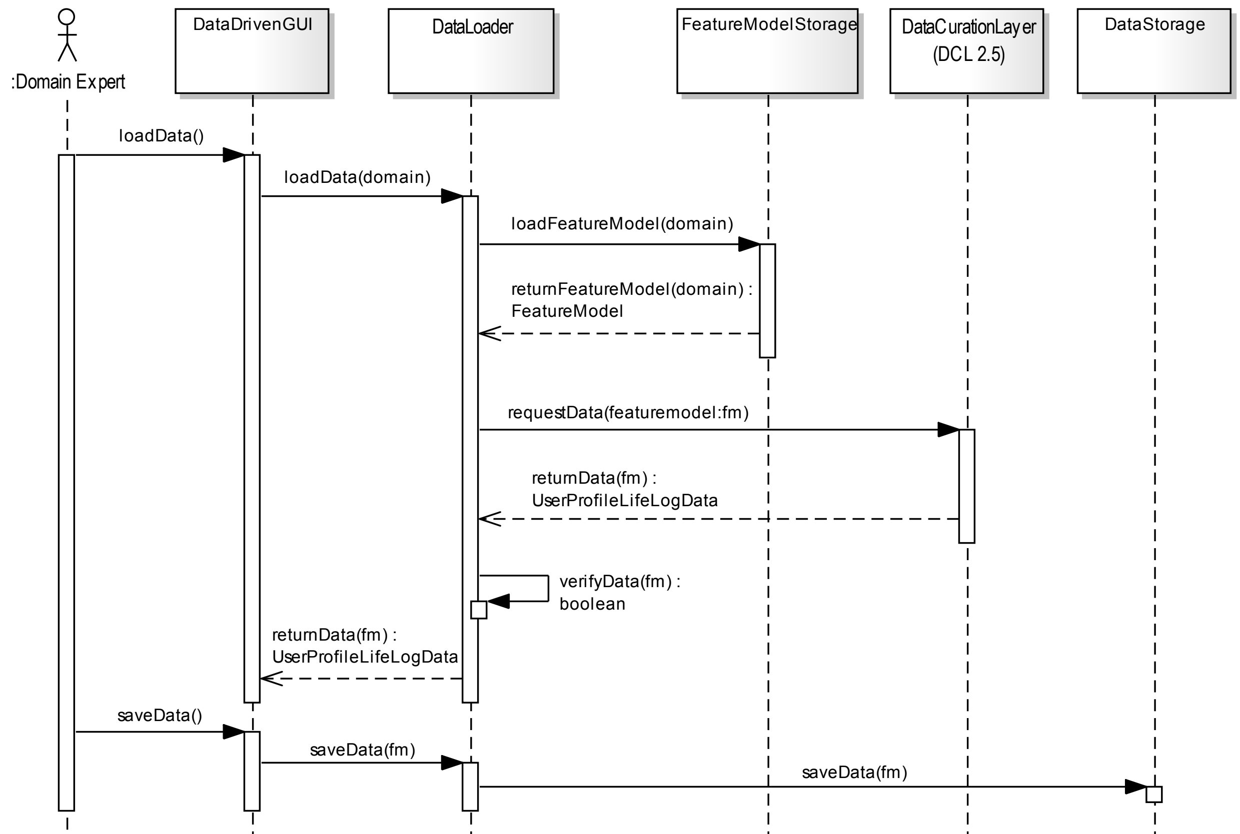
Sequence Diagram:



<table>
<tr><td>Use Case ID:</td><td colspan="3">KCL2.5-SUC-06</td></tr>
<tr><td>Use Case Name:</td><td colspan="3">Retrieve user profile and lifelog data</td></tr>
<tr><td>Created By:</td><td>Maqbool Ali</td><td>Last Updated By:</td><td>Maqbool Hussain</td></tr>
<tr><td>Date Created:</td><td>10-07-2015</td><td>Last Revision Date:</td><td>15-04-2016</td></tr>
<tr><td>Actors:</td><td colspan="3">Domain Expert, DCL 2.5</td></tr>
<tr><td>Description:</td><td colspan="3">User profile and lifelog data has hidden knowledge that can be explored after loading from data storage</td></tr>
<tr><td>Trigger:</td><td colspan="3">Prior to classification model creation needed for required data</td></tr>
<tr><td>Preconditions:</td><td colspan="3">• System has access through service interface to retrieve user profile and lifelog data from DCL 2.5
• DCL 2.5 has capability to share user profile and lifelog data in secure environment.
• System has already loaded the previously imported user profile and lifelog schema</td></tr>
<tr><td>Postconditions:</td><td colspan="3">System will receive user profile and lifelog data based on selected schema</td></tr>
<tr><td>Normal Flow:</td><td colspan="3">1. Domain expert loads the feature model for selected domain
2. System loads the corresponding feature model
3. Domain expert sends request to DCL 2.5 for user profile and lifelog data based on loaded feature model
4. DCL 2.5 shares the user profile and lifelog data
5. System receives the user profile and lifelog data
6. Domain expert uses the system and performs the following tasks;
o Verifies the user profile and lifelog data
o Saves the data after verification.
7. System saves the verified data</td></tr>
<tr><td>Alternative Flows:</td><td colspan="3"></td></tr>
<tr><td>Exceptions:</td><td colspan="3">3a. System unable to connect to DCL 2.5
a. System connection is failed during retrieving user profile and lifelog data
b. System hold and will retry after sometime to connect to DCL 2.5 and retrieve the user profile and lifelog data
6a. System receives irrelevant data
a. System detects the irrelevant data sent by DCL 2.5.
b. System request again DCL 2.5 to make sure that data received is according to feature selected.</td></tr>
<tr><td>Includes:</td><td colspan="3"></td></tr>
<tr><td>Frequency of Use:</td><td colspan="3">When new service is required and mining mind have sufficient data for classification model creation</td></tr>
</table>

Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	If DCL 2.5 is unable to send data based on dynamic feature selection from schema, then alternate strategy has to be considered.

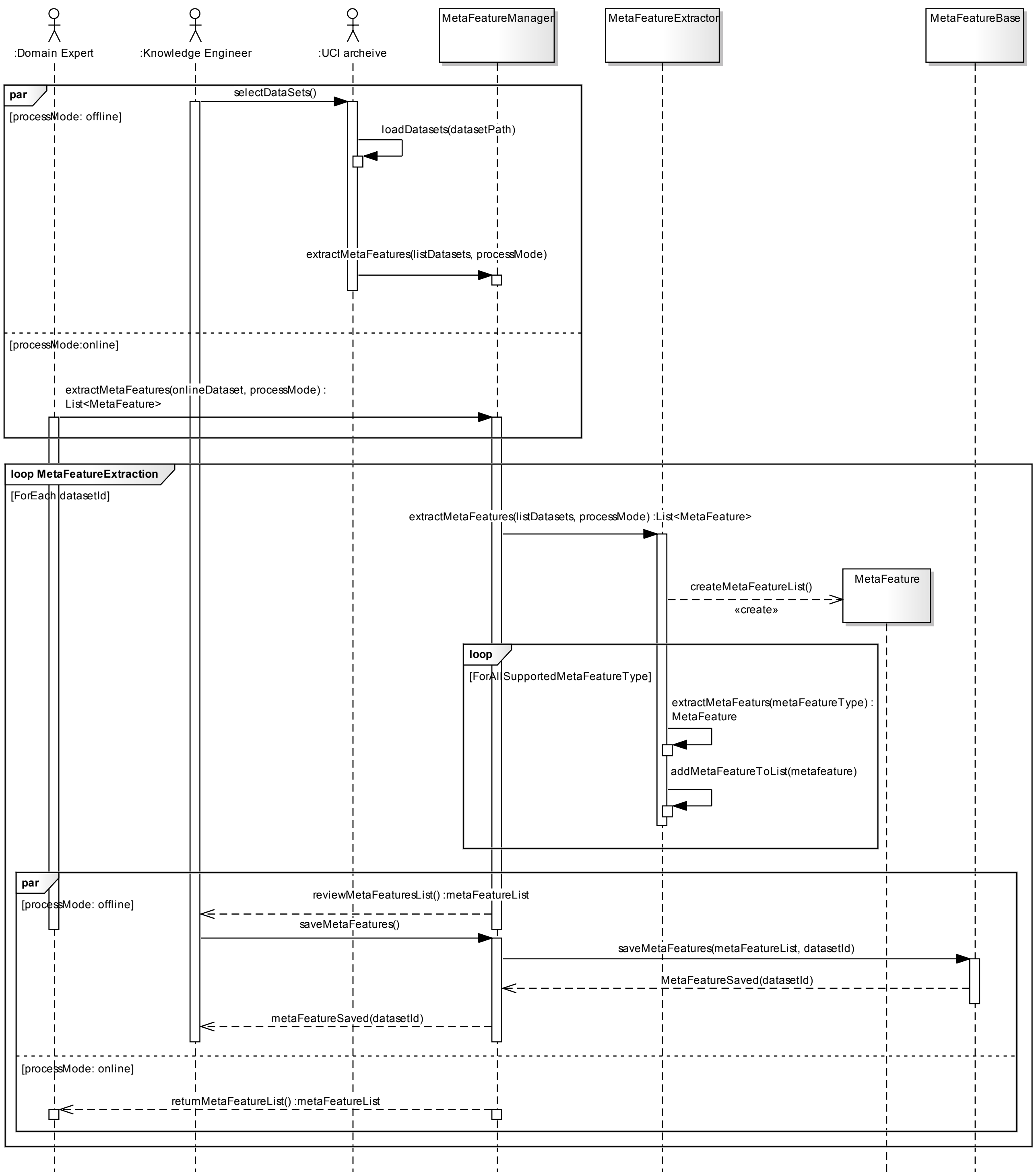
Sequence Diagram:



Use Case ID:	**KCL2.5-SUC-07**		
Use Case Name:	Extract meta-features of classification datasets		
Created By:	Rahman Ali	**Last Updated By:**	Maqbool Hussain
Date Created:	16-07-2015	**Last Revision Date:**	15-04-2016
Actors:	Knowledge Engineer/Domain Expert, UCI archive		
Description:	Datasets have simple, statistical, information theory and landmarking meta-features that can best describes nature of a dataset. These features can best used for building an algorithm selection model.		
Trigger:	In the offline process, when the algorithm selection model is build, and in the online process, when an appropriate algorithm is needed to be identified for a new query dataset		
Preconditions:	UCI archive datasets are available and are in refined format		
Postconditions:	The meta-features are ready for being used in building algorithm selection model.		
Normal Flow:	1. Knowledge Engineer selects one dataset from UCI archive. 2. System retrieves selected dataset. 3. Knowledge Engineer provides dataset to meta-feature extractor for extracting meta-features. 4. System extracts following meta-features set for selected dataset. a. basic meta-features b. statistical meta-features c. information theory meta-features d. extract landmark features 5. Knowledge Engineer reviews the extracted meta-features and saves it into meta-features base (MFB). 6. System saves meta-features into a MFB. 7. Knowledge Engineer repeats step 1-6 for each intended dataset.		
Alternative Flows:	1a. Meta-feature extraction for online dataset a. Domain Expert provides new dataset used for classification model creation. b. Step 3-4 of Normal Flow is executed for Domain expert interactions with system.		
Exceptions:	N/A		
Includes:	N/A		
Frequency of Use:	Frequently, whenever a new dataset arrives as a query dataset.		
Special Requirements:	Minimum availability of classification datasets > 60 for a reasonable accuracy		
Assumptions:	• The archived datasets are available and are in refined .arff format		

	• The meta-feature space is defined in advance
Notes and Issues:	Meta-feature extraction is time consuming task for offline process as we have to take into account more than 60 datasets.

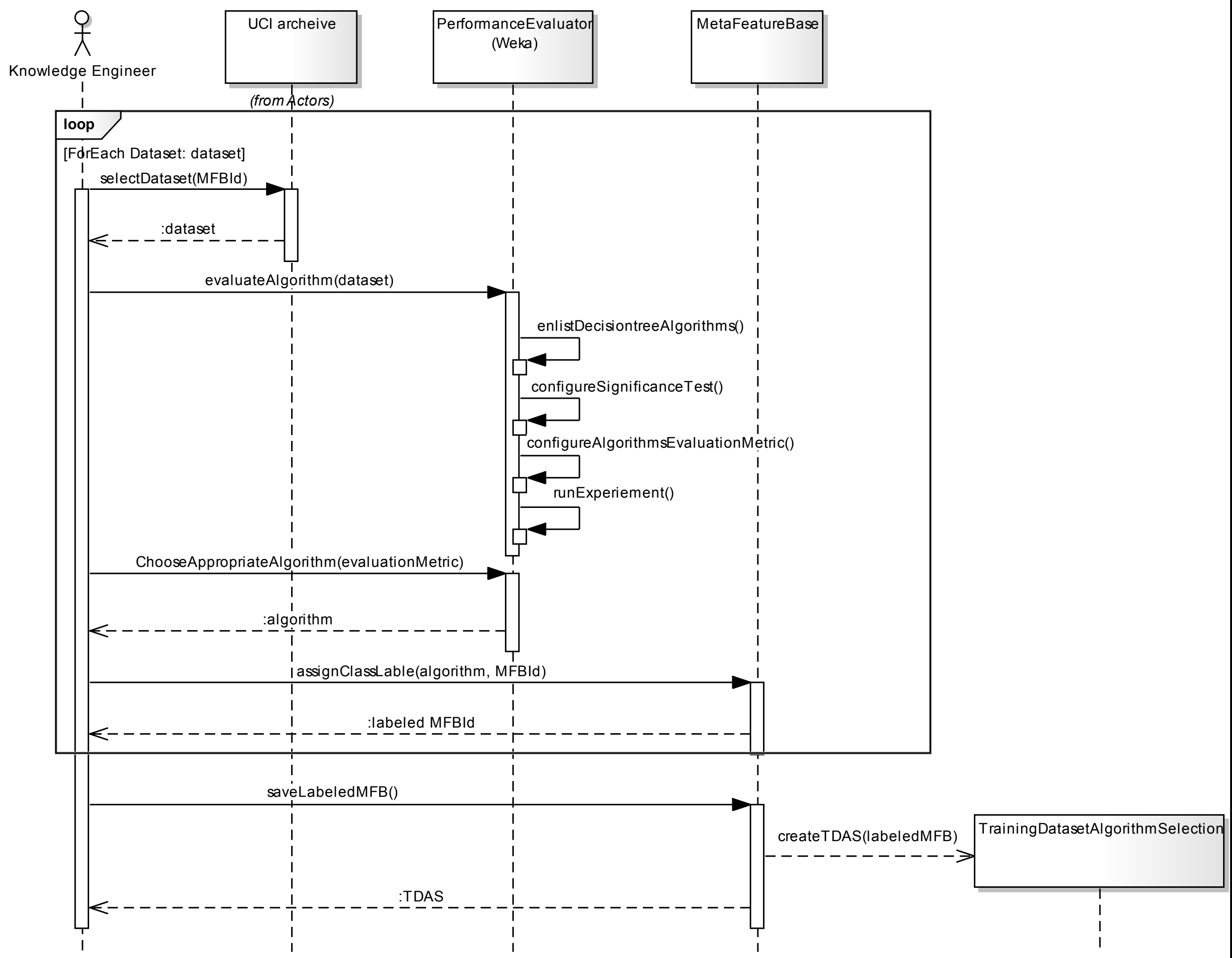
Sequence Diagram:



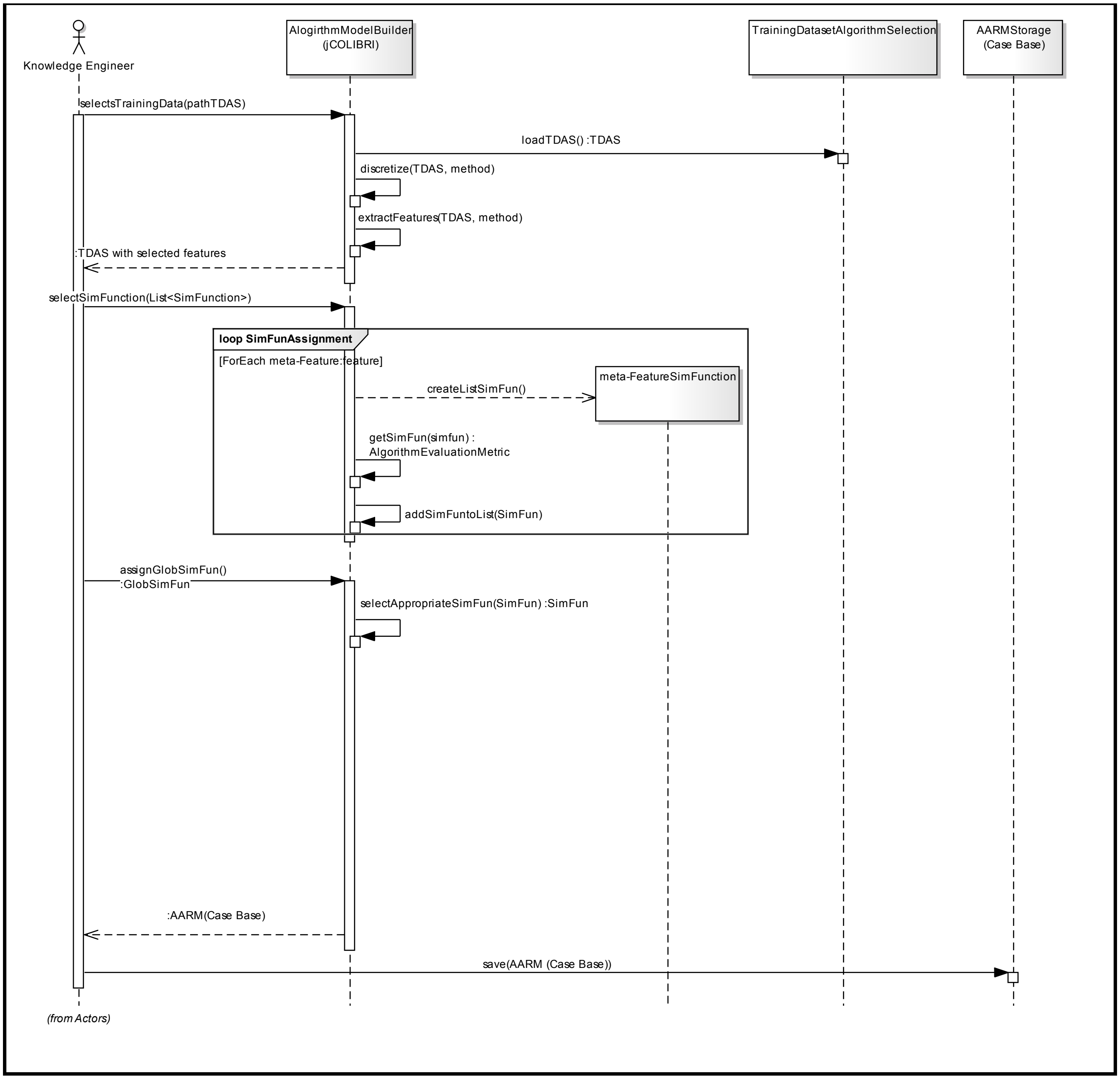
<table>
<tr><td>Use Case ID:</td><td colspan="3">KCL2.5-SUC-08</td></tr>
<tr><td>Use Case Name:</td><td colspan="3">Evaluate performance of decision tree algorithm</td></tr>
<tr><td>Created By:</td><td>Rahman Ali</td><td>Last Updated By:</td><td>Maqbool Hussain</td></tr>
<tr><td>Date Created:</td><td>16-07-2015</td><td>Last Revision Date:</td><td>15-04-2016</td></tr>
<tr><td>Actors:</td><td colspan="3">Knowledge Engineer, UCI archive</td></tr>
<tr><td>Description:</td><td colspan="3">Different algorithms have different performance score for the same dataset. To build an algorithm selection model, performance score of each algorithm needs to evaluate for choosing an appropriate one.</td></tr>
<tr><td>Trigger:</td><td colspan="3">In the offline process, when the algorithm selection model is to build the first time.</td></tr>
<tr><td>Preconditions:</td><td colspan="3">• UCI archive datasets are available and are in .arff format
• The algorithm to be considered is specified in advanced (Decision Tree algorithms implemented in Weka)
• The evaluation metric is specified (F-measure)</td></tr>
<tr><td>Postconditions:</td><td colspan="3">All datasets records in Meta-Feature Base (MFB) will be assigned with optimal decision tree algorithm class label.</td></tr>
<tr><td>Normal Flow:</td><td colspan="3">1. Knowledge Engineer selects UCI archive dataset, mentioned in MFB, for finding optimal decision tree algorithm.
2. System (Weka) loads selected datasets.
3. Knowledge Engineer setups experiment;
a. Enlists all the decision tree algorithms available in system (Weka)
b. Configure significance test (alpha=0.5)
c. Configure algorithms comparison metric (f-measure)
4. System runs experiment and produces detailed f-score for all selected algorithms.
5. Knowledge Engineer performs following tasks;
a. Records evaluation matrix.
b. Chooses algorithm with the highest f-score.
c. Assigns chosen algorithm as class label in MFB.
d. Step 1-5 are repeated for other non-labeled datasets in MFB.
e. After finishing labeling all records in MFB, saves the updated MFB as training dataset for algorithm selection (TDAS).
6. System saves the updated records in MFB as final TDAS.</td></tr>
<tr><td>Alternative Flows:</td><td colspan="3">N/A</td></tr>
<tr><td>Exceptions:</td><td colspan="3">N/A</td></tr>
<tr><td>Includes:</td><td colspan="3">N/A</td></tr>
<tr><td>Frequency of Use:</td><td colspan="3">• Rarely, once enough new datasets are added to the system</td></tr>
</table>

Special Requirements:	N/A
Assumptions:	• The decision tree-based algorithms are implemented in Weka.
Notes and Issues:	Evaluating performance of decision tree algorithms over a large number (min 60) classification dataset is a computationally complex task. For minimum 60 datasets and at least 5 decision tree algorithms, a minimum of 300 experiments are required. Moreover, 60 additional significance test experiments are needed. On average, each experiment takes times in minutes ranging from 2 minutes to 30 minutes, depending on the complexity of the dataset.

Sequence Diagram:



Use Case ID:	**KCL2.5-SUC-09**		
Use Case Name:	Create automatic algorithm recommendation model		
Created By:	Rahman Ali	**Last Updated By:**	Maqbool Hussain
Date Created:	10-07-2015	**Last Revision Date:**	15-04-2016
Actors:	Knowledge Engineer/Domain Expert		
Description:	An automatic algorithm selection model enables knowledge engineer to automatically select appropriate algorithm for building classification model for his new dataset		
Trigger:	When the training dataset comprising datasets meta-features and algorithms performance evaluation are made available.		
Preconditions:	The datasets meta-features and algorithms performance training datasets is made available.		
Postconditions:	The automatic algorithm recommendation model (AARM) is ready to integrate in Mining Minds for real time algorithm selection.		
Normal Flow:	1. Knowledge Engineer selects TDAS. 2. Knowledge Engineer performs preprocessing of the TDAS (i.e., discretization, and features selection). 3. System (jColibri) loads refined TDAS. 4. Knowledge Engineer select each meta-feature and assign appropriate local similarity function for matching. 5. Knowledge Engineer select each meta-feature and assign appropriate global similarity function for matching. 6. System (jColibri) builds Case Base Structure in the memory. 7. Knowledge Engineer saves AARM (Case Base). 8. System saves the AARM into AARM storage.		
Alternative Flows:	N/A		
Exceptions:	N/A		
Includes:	N/A		
Frequency of Use:	Rarely, when TDAS is updated with new datasets or algorithms		
Special Requirements:	Availability of records > 60 for minimum acceptable accuracy		
Assumptions:	N/A		
Notes and Issues:	N/A		
Sequence Diagram:			

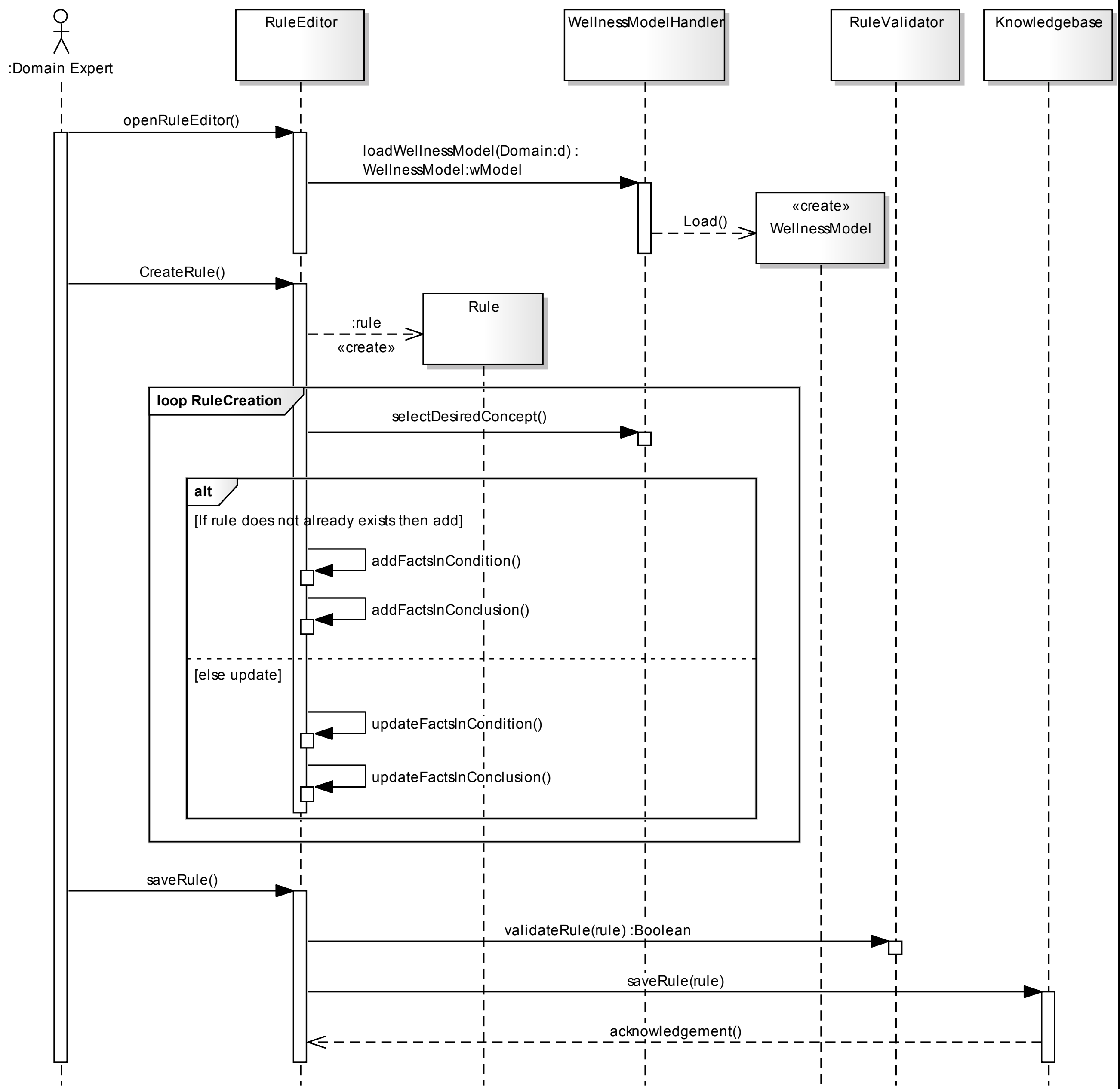


Knowledge Engineer
AlogirthmModelBuilder
(jCOLIBRI)
TrainingDatasetAlgorithmSelection
AARMStorage
(Case Base)
selectsTrainingData(pathTDAS)
loadTDAS() :TDAS
discretize(TDAS, method)
extractFeatures(TDAS, method)
:TDAS with selected features
selectSimFunction(List<SimFunction>)
loop SimFunAssignment
[ForEach meta-Feature:feature]
meta-FeatureSimFunction
createListSimFun()
getSimFun(simfun) :
AlgorithmEvaluationMetric
addSimFuntoList(SimFun)
assignGlobSimFun()
:GlobSimFun
selectAppropriateSimFun(SimFun) :SimFun
:AARM(Case Base)
save(AARM (Case Base))
(from Actors)

Use Case ID:	**KCL2.5-SUC-10**		
Use Case Name:	Create Rule		
Created By:	Taqdir Ali	**Last Updated By:**	Maqbool Hussain
Date Created:	11-07-2015	**Last Revision Date:**	15-04-2016
Actors:	Domain Expert		
Description:	Knowledge bases need to enhance with up-to-date knowledge for correct recommendation. The expert shall create rules in the knowledge base to transform their practices into executable knowledge in form of rules.		
Trigger:	Domain experts trigger it for rule creation/updating when needed.		
Preconditions:	• The domain expert shall be authenticated with full access of rule management in the knowledge base.		
Postconditions:	The correct rule shall be saved into the knowledge base.		
Normal Flow:	1. Domain expert opens the rule editor. 2. System loads wellness domain model in form of concepts tree. 3. Domain expert selects concepts for the rule conditions and conclusion using any of the following methods; o Using wellness model, dragging concepts and facts into conditional or conclusion part of the rule editor. o Using auto pop-up Intelli-sense window to select concepts and facts into conditional or conclusion part of the rule editor. 4. System checks the existing rules to add/update the rule o Add new facts of the rule in condition. o Add new conclusion according to rule facts. 5. Step 3-4 will be repeated for each new/updated concept added to rule, and domain expert finally saves the rule. 6. System save rule as follows; o The system validates the rule using “KCL2.5-SUC-11”. o The system stores the validated rule into the knowledge base		
Alternative Flows:	6a. System founds the rule is already exists in rule repository o Domain expert review the existing facts and conclusion. o Step 5-6 will be followed to change the rule.		
Exceptions:	N/A		
Includes:	Validate Rule		
Frequency of Use:	Whenever domain expert want to add rule or edit the existing rule.		
Special Requirements:	N/A		

Assumptions:	N/A
Notes and Issues:	• If knowledge base does not exist in system the administrator shall build the knowledge base first and configure with system. • After investigation, we may use unify representation for rules and guidelines.

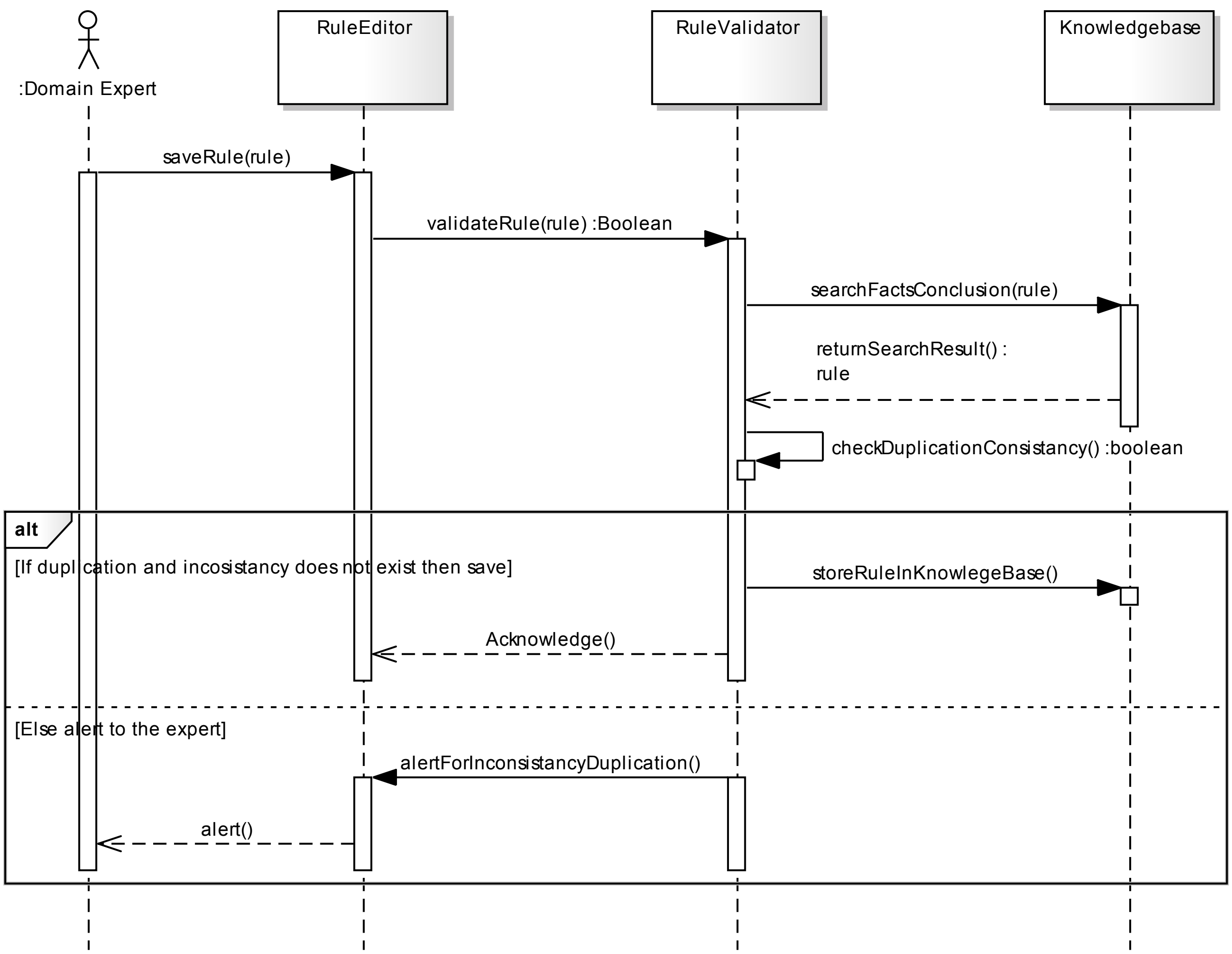
Sequence Diagram:



<table>
<tr><td>Use Case ID:</td><td colspan="3">KCL2.5-SUC-11</td></tr>
<tr><td>Use Case Name:</td><td colspan="3">Validate Rule</td></tr>
<tr><td>Created By:</td><td>Taqdir Ali</td><td>Last Updated By:</td><td>Maqbool Hussain</td></tr>
<tr><td>Date Created:</td><td>11-07-2015</td><td>Last Revision Date:</td><td>15-04-2016</td></tr>
<tr><td>Actors:</td><td colspan="3">Domain Expert</td></tr>
<tr><td>Description:</td><td colspan="3">In new rules creation and editing existing rules, duplication and inconsistency may occur. The validation is needed to validate and find the duplication and inconsistency among the rules.</td></tr>
<tr><td>Trigger:</td><td colspan="3">• When new rule is going to create.
• When existing rule is going to update</td></tr>
<tr><td>Preconditions:</td><td colspan="3">The rule creation and editing process completed by physician successfully.</td></tr>
<tr><td>Postconditions:</td><td colspan="3">The validated rule shall be saved into knowledge base.</td></tr>
<tr><td>Normal Flow:</td><td colspan="3">1. Domain expert saves the created rule.
2. System validate rule for inconsistency and duplication as follows
a. Fetch the facts and conclusion of existing rules.
b. The new or updated rule approved for having no inconsistency and duplication.
c. Created rule stores into the rules repository.
d. Acknowledge the expert to save the rule successfully.</td></tr>
<tr><td>Alternative Flows:</td><td colspan="3">2b. The created rule is found having inconsistency or duplication with existing rules in the rules repository
a. The system produces alert the inconsistency or duplication of the rule with existing rules in repository.
b. Domain expert review the alert message and correct the created rule.
c. Step 1 and Step 2 of normal flow is executed.</td></tr>
<tr><td>Exceptions:</td><td colspan="3">N/A</td></tr>
<tr><td>Includes:</td><td colspan="3">N/A</td></tr>
<tr><td>Frequency of Use:</td><td colspan="3">Whenever domain expert want to save the rule</td></tr>
<tr><td>Special Requirements:</td><td colspan="3">N/A</td></tr>
<tr><td>Assumptions:</td><td colspan="3">N/A</td></tr>
<tr><td>Notes and</td><td colspan="3">Finding duplication using facts of condition and conclusion in existing</td></tr>
</table>

Issues:	rules is challenging task.

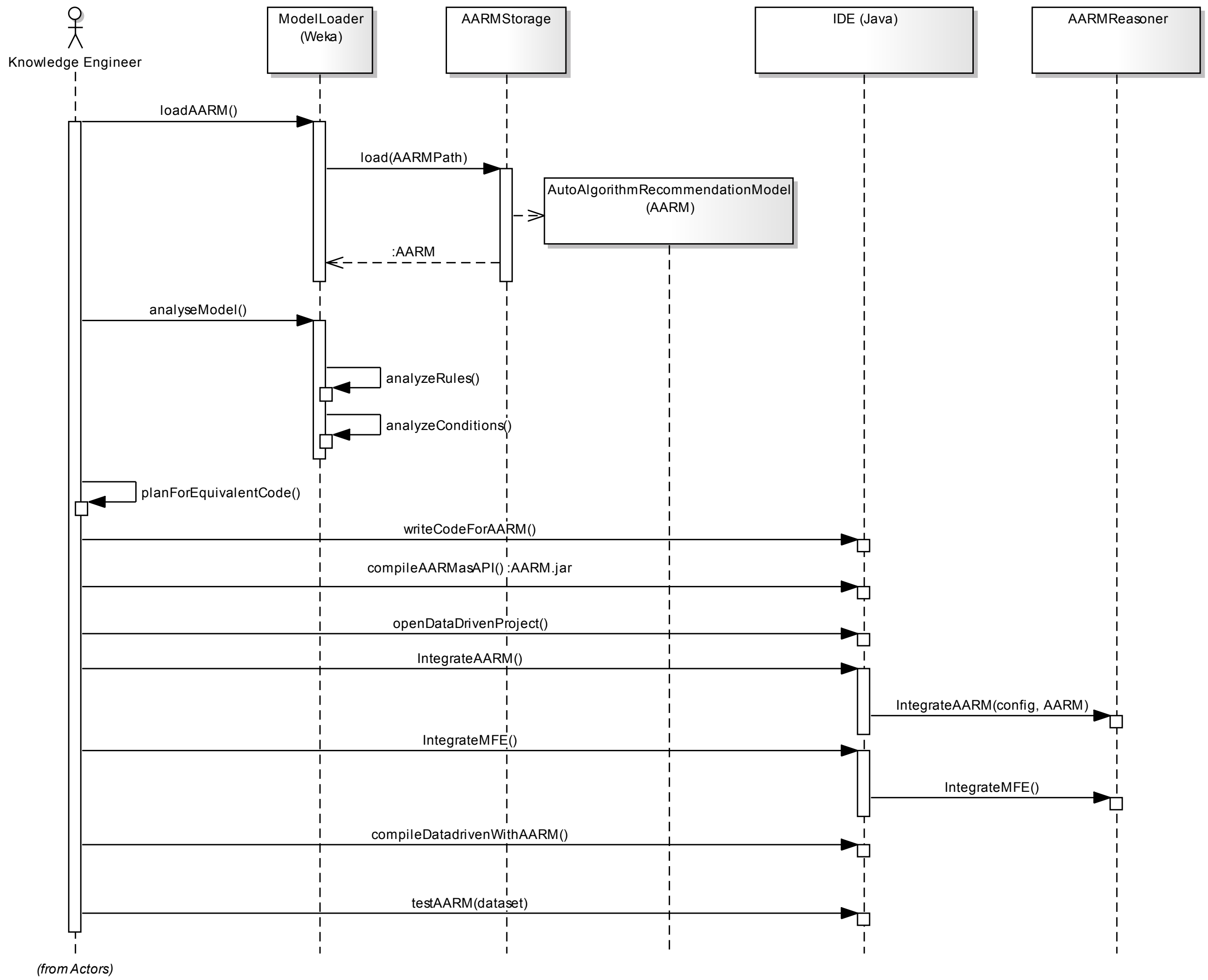
Sequence Diagram:



Use Case ID:	**KCL2.5-SUC-12**		
Use Case Name:	Integrate automatic algorithm recommendation model		
Created By:	Rahman Ali	**Last Updated By:**	Maqbool Hussain
Date Created:	10-07-2015	**Last Revision Date:**	15-04-2016
Actors:	Knowledge Engineer		
Description:	For real time recommendation of classification algorithm for a new dataset, AARM need to be integrated in Mining Minds Data Driven knowledge acquisition approach.		
Trigger:	When AARM is built.		
Preconditions:	• AARM is available • Data driven approach has a unified interface to support AARM as plugin • Data driven has unified interface for accessing Meta-Feature Extractor.		
Postconditions:	AARM is plugged into data driven environment and readily available for real time recommendation of appropriate classification algorithm.		
Normal Flow:	1. Knowledge engineer selects AARM and performs the following tasks; • Analyses number of rules in the AARM • Analyses condition attributes used in each rule of AARM • Transforms rules into executable classes (using any IDE of Java). 2. Knowledge engineer integrates the executable AARM into data driven as follows; a. Write integration code (following unified interface) into data driven source code b. Update possible configuration for newly added AARM plugin. c. Update possible configuration for accessing Meta-Feature Extractor. 3. Knowledge Engineer compile the AARM as integral part with data driven code. 4. Knowledge engineer tests AARM with sample dataset.		
Alternative Flows:	N/A		
Exceptions:	N/A		
Includes:	N/A		
Frequency of Use:	Rarely, once the AARM is updated		
Special Requirements:	AARM has acceptable accuracy (60%)		
Assumptions:	AARM is created		

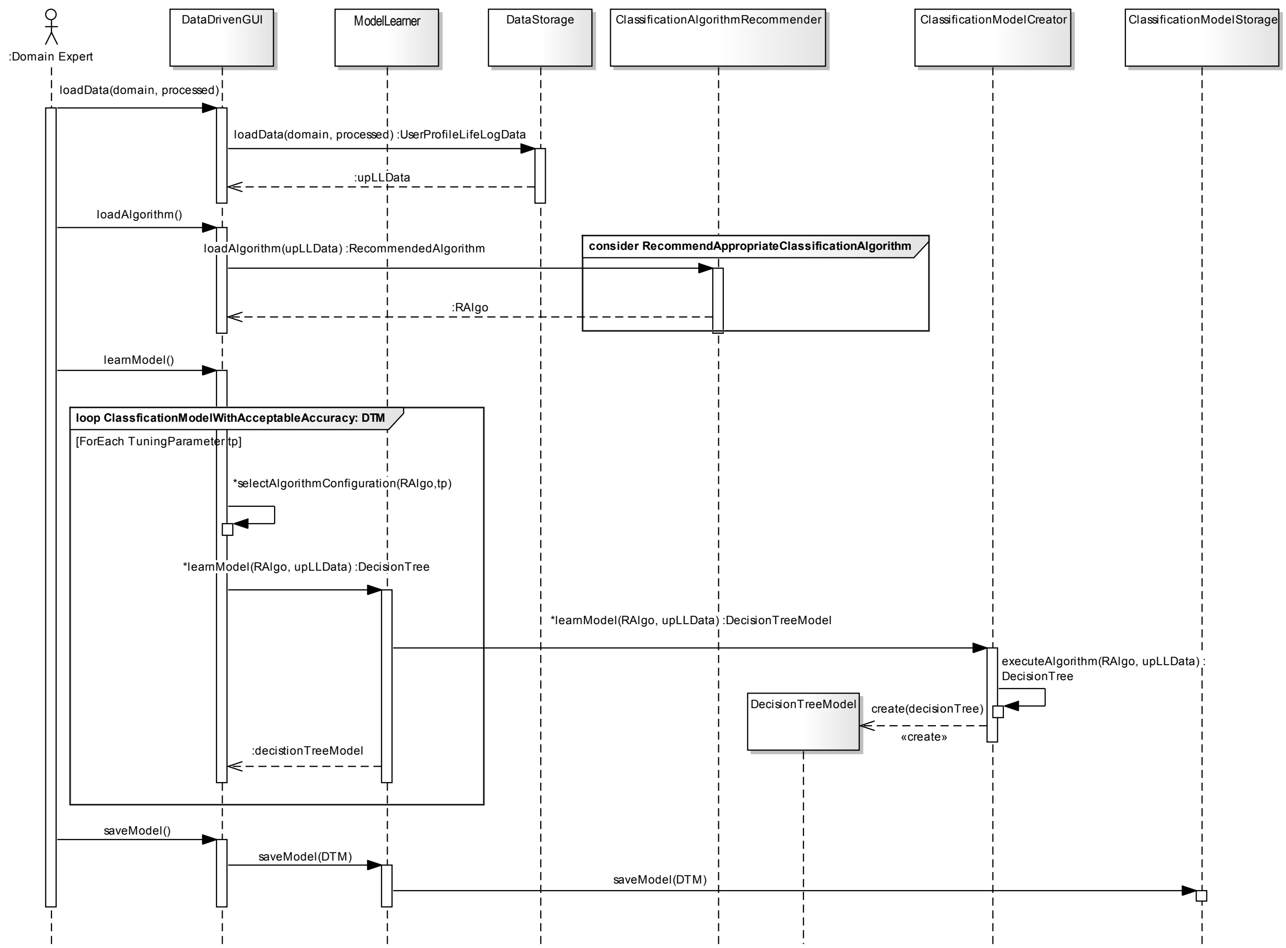
Notes and Issues:	N/A

Sequence Diagram:

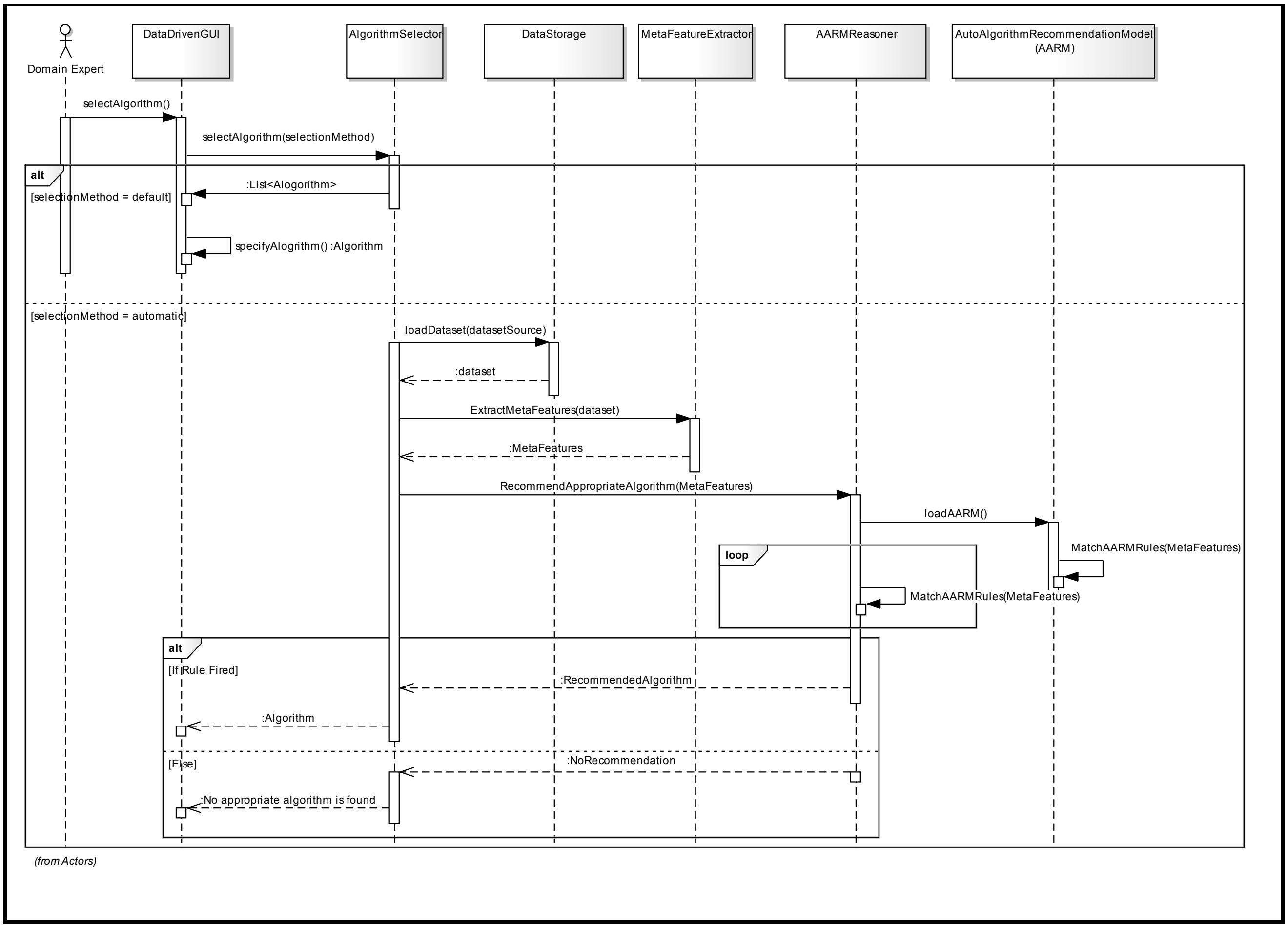


Use Case ID:	**KCL2.5-SUC-13**		
Use Case Name:	Learn classification model		
Created By:	Maqbool Ali	**Last Updated By:**	Maqbool Hussain
Date Created:	10-07-2015	**Last Revision Date:**	15-04-2016
Actors:	Domain Expert		
Description:	An expert wants to see hidden knowledge from user profile lifelog data that can be explored by model learning mechanism with the help of learning method as well as processed data.		
Trigger:	Learn model required to explore hidden knowledge		
Preconditions:	• System has loaded the prepared user profile lifelog data		
Postconditions:	System will build the classification model (decision tree)		
Normal Flow:	1. Domain expert loads the user profile lifelog processed data for selected domain 2. System loads the corresponding processed data 3. Domain expert invokes the "Recommend appropriate classification algorithm" use case by providing processed data to load the appropriate learning algorithm 4. System loads the appropriate decision tree learning algorithm 5. Domain expert select the algorithm tuning parameters of selected algorithm for further improving the results 6. System applies the tuning parameters on selected algorithm and computes the learning accuracy after learning the user profile lifelog processed data 7. Repeat the step 5-6 until required learning accuracy is achieved. 8. Domain expert finalizes the classification model with acceptable accuracy and saves the model. 9. System saves the decision tree learning model.		
Alternative Flows:	N/A		
Exceptions:	N/A		
Includes:	KCL2.5-SUC-14 (Recommend appropriate classification algorithm)		
Frequency of Use:	When new service is required and mining mind have sufficient data for classification model creation		
Special Requirements:	N/A		
Assumptions:	N/A		
Notes and Issues:	Acceptable Accuracy: Depends on criticality of the domain. For example classification model for clinical domain needs high accuracy, while for the case of nutrition domain, high accuracy is not critical.		

Sequence Diagram:

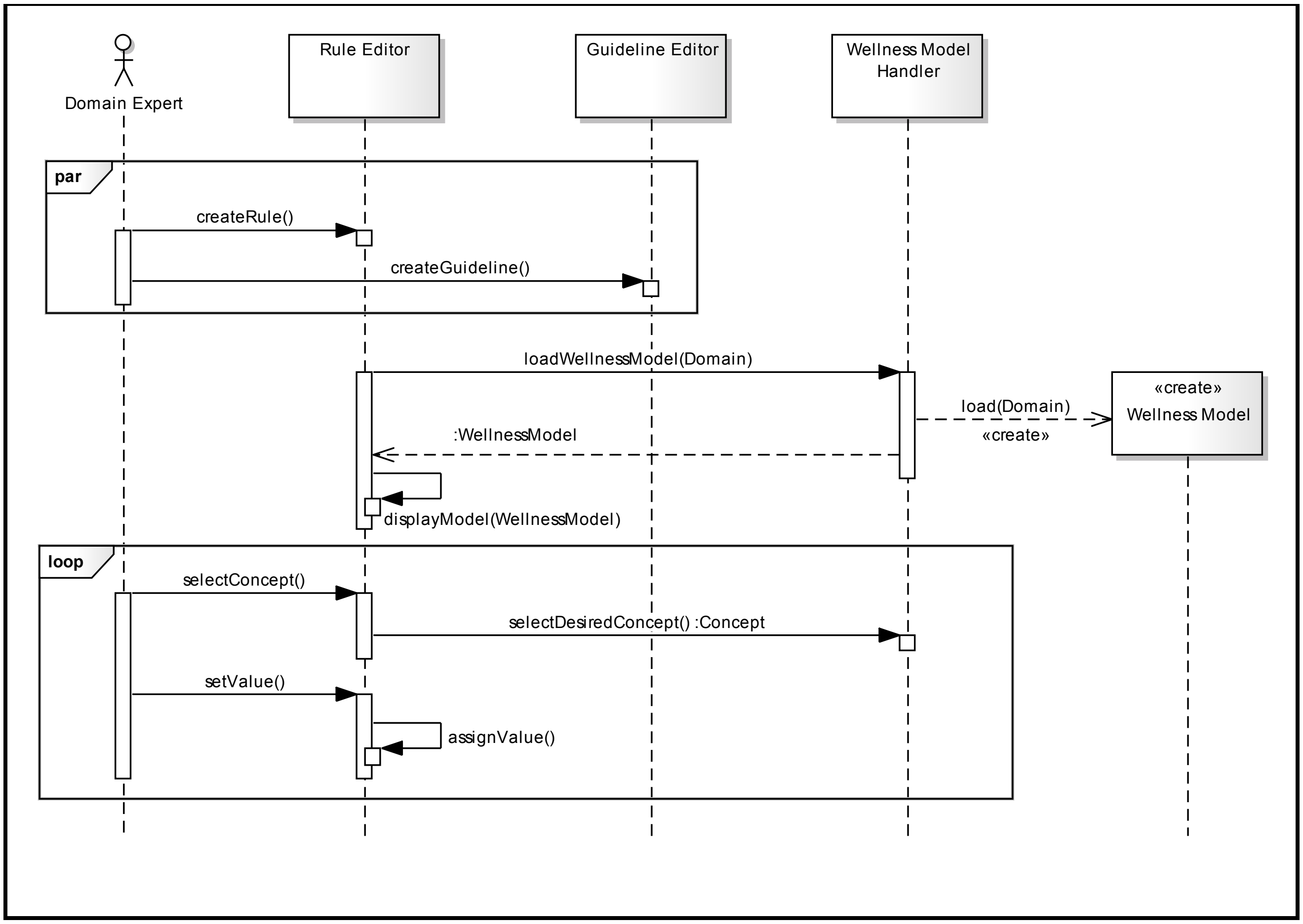


<table>
<tr><td>Use Case ID:</td><td colspan="3">KCL2.5-SUC-14</td></tr>
<tr><td>Use Case Name:</td><td colspan="3">Recommend appropriate classification algorithm</td></tr>
<tr><td>Created By:</td><td>Rahman Ali</td><td>Last Updated By:</td><td>Rahman Ali</td></tr>
<tr><td>Date Created:</td><td>16-07-2015</td><td>Last Revision Date:</td><td>15-04-2016</td></tr>
<tr><td>Actors:</td><td colspan="3">Domain expert</td></tr>
<tr><td>Description:</td><td colspan="3">For building classification model for the user new dataset, AARM shall automatically recommend appropriate classification algorithm.</td></tr>
<tr><td>Trigger:</td><td colspan="3">When domain expert wants to build a classification model.</td></tr>
<tr><td>Preconditions:</td><td colspan="3">• AARM is plugged into the data driven environment
• Meta-features extractor is plugged into the data driven environment
• New dataset is stored in local machine, structured in .arff file format</td></tr>
<tr><td>Postconditions:</td><td colspan="3">The recommended appropriate classification algorithm can be used for building classification model</td></tr>
<tr><td>Normal Flow:</td><td colspan="3">1. Domain expert loads new dataset (.arff file) from the data driven datasets storage using data driven environment.
2. System extracts meta-features of the new dataset by including KCL2.5-SUC-07 (alternate flow)
3. Domain experts provides meta-features to system for recommending appropriate classification algorithm
4. System performs meta-reasoning over integrated AARM using the following steps;
a. Starts matching each meta-feature value of the new dataset with condition attributes of each rule
b. If matched, fires the rule, recommend right hand side of the rule as the appropriate algorithm
c. Else, display a message “could not recommend”</td></tr>
<tr><td>Alternative Flows:</td><td colspan="3">1a. 4c(a) If AARM not available or have no acceptable recommendation accuracy, use Weka experimenter.</td></tr>
<tr><td>Exceptions:</td><td colspan="3">N/A</td></tr>
<tr><td>Includes:</td><td colspan="3">KCL2.5-SUC-07 (alternate flow)</td></tr>
<tr><td>Frequency of Use:</td><td colspan="3">Frequently, when domain expert needs to select appropriate algorithm for his/her dataset</td></tr>
<tr><td>Special Requirements:</td><td colspan="3">N/A</td></tr>
<tr><td>Assumptions:</td><td colspan="3">N/A</td></tr>
<tr><td>Notes and Issues:</td><td colspan="3">N/A</td></tr>
<tr><td colspan="4">Sequence Diagram:</td></tr>
</table>



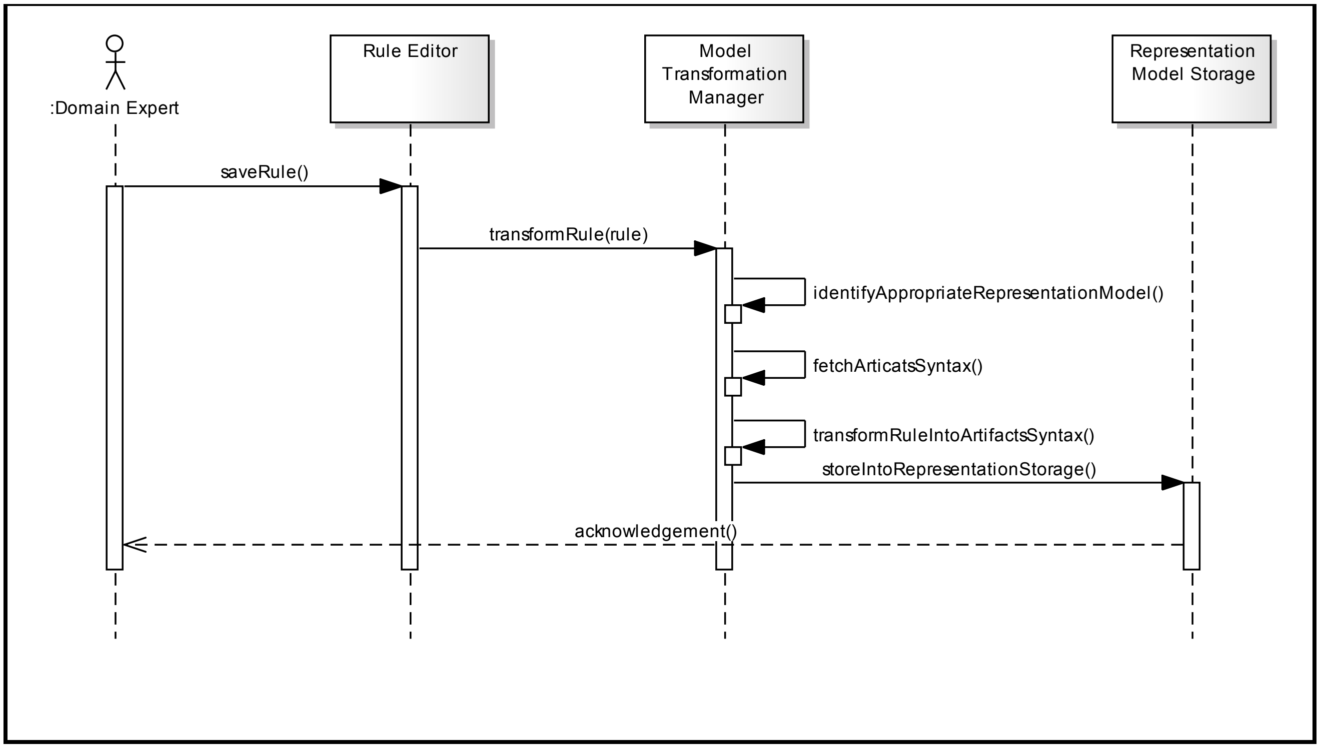
Domain Expert
DataDrivenGUI
AlgorithmSelector
DataStorage
MetaFeatureExtractor
AARMReasoner
AutoAlgorithmRecommendationModel (AARM)
selectAlgorithm()
selectAlgorithm(selectionMethod)
alt
[selectionMethod = default]
:List<Alogorithm>
specifyAlogrithm() :Algorithm
[selectionMethod = automatic]
loadDataset(datasetSource)
:dataset
ExtractMetaFeatures(dataset)
:MetaFeatures
RecommendAppropriateAlgorithm(MetaFeatures)
loadAARM()
MatchAARMRules(MetaFeatures)
loop
MatchAARMRules(MetaFeatures)
alt
[If Rule Fired]
:RecommendedAlgorithm
:Algorithm
[Else]
:NoRecommendation
:No appropriate algorithm is found
(from Actors)

Use Case ID:	**KCL2.5-SUC-15**		
Use Case Name:	Manage concepts of domain model		
Created By:	Taqdir Ali	**Last Updated By:**	Taqdir Ali, Maqbool Hussain
Date Created:	27-07-2015	**Last Revision Date:**	15-04-2016
Actors:	Domain Expert		
Description:	The concepts of wellness domain shall be used in creation of rules and generation of guidelines in tree format. The domain expert shall easily select the required concepts from wellness domain model.		
Trigger:	Domain Model will be loaded during rule creation or guideline creation.		
Preconditions:	The expert be authenticated with full access of concepts management in domain model		
Postconditions:	The right concept shall be added or edited at the right location in wellness model		
Normal Flow:	1. Domain expert creates rule (using KCL2.5-SUC-10) or creates guideline (using KCL2.5-SUC-03). 2. System loads domain model for corresponding domain. 3. Domain expert selects concepts from loaded domain model. 4. System associate domain concept to part of rule or guideline tree. 5. Domain expert assigns value to selected concept. 6. System assigns corresponding value to selected concept and show it in rule or guideline tree. 7. Step 4-6 are repeated till rule or guideline is finished.		
Alternative Flows:	N/A		
Exceptions:	N/A		
Includes:	N/A		
Frequency of Use:	Whenever domain expert want to add or edit the concepts in wellness model		
Special Requirements:	N/A		
Assumptions:	Wellness model repository in the system is exist.		
Notes and Issues:	If wellness model storage does not exist in system the administrator shall build the wellness model storage first and configure with system.		
Sequence Diagram:			

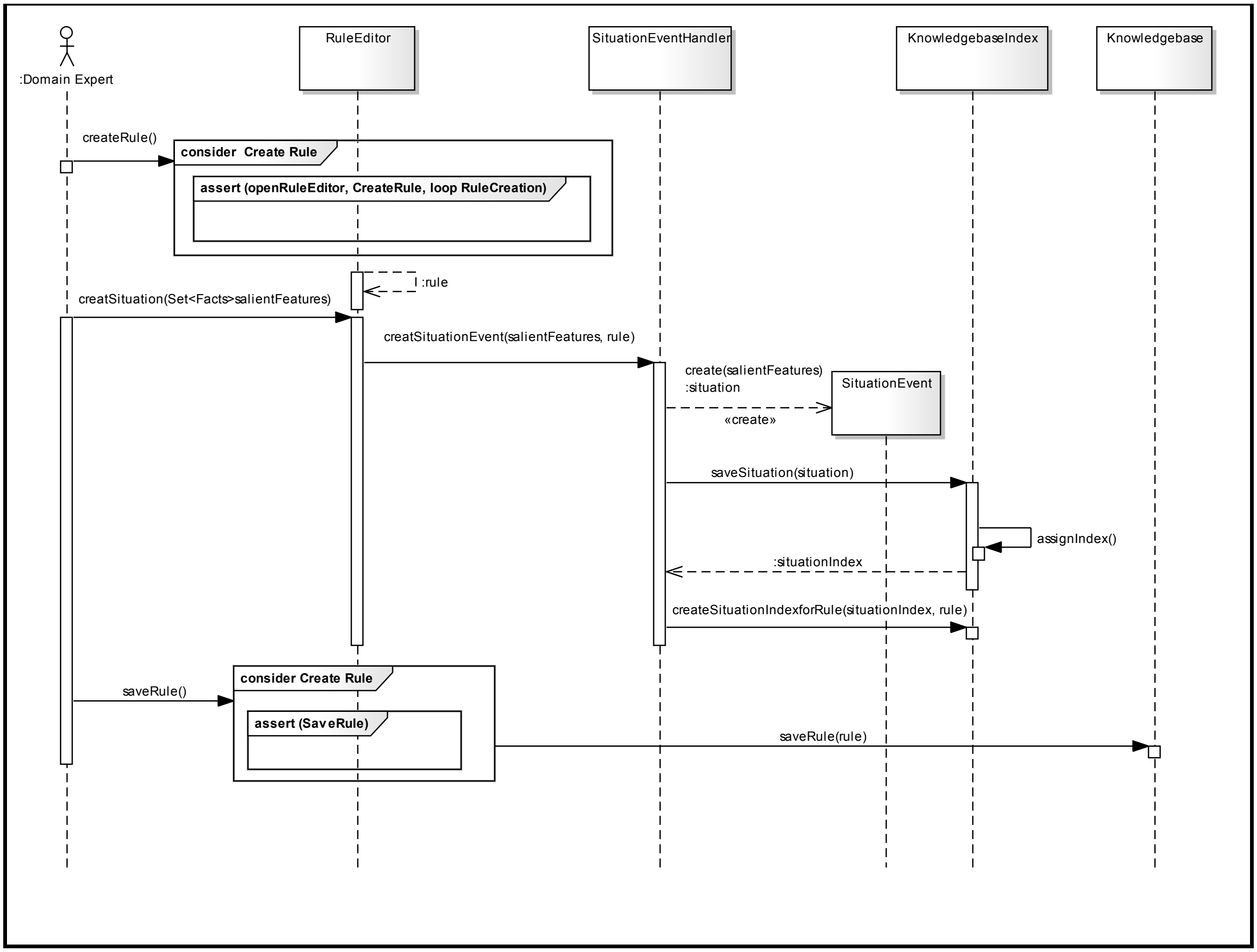


Domain Expert
Rule Editor
Guideline Editor
Wellness Model Handler
par
createRule()
createGuideline()
loadWellnessModel(Domain)
«create»
Wellness Model
load(Domain)
«create»
:WellnessModel
displayModel(WellnessModel)
loop
selectConcept()
selectDesiredConcept() :Concept
setValue()
assignValue()

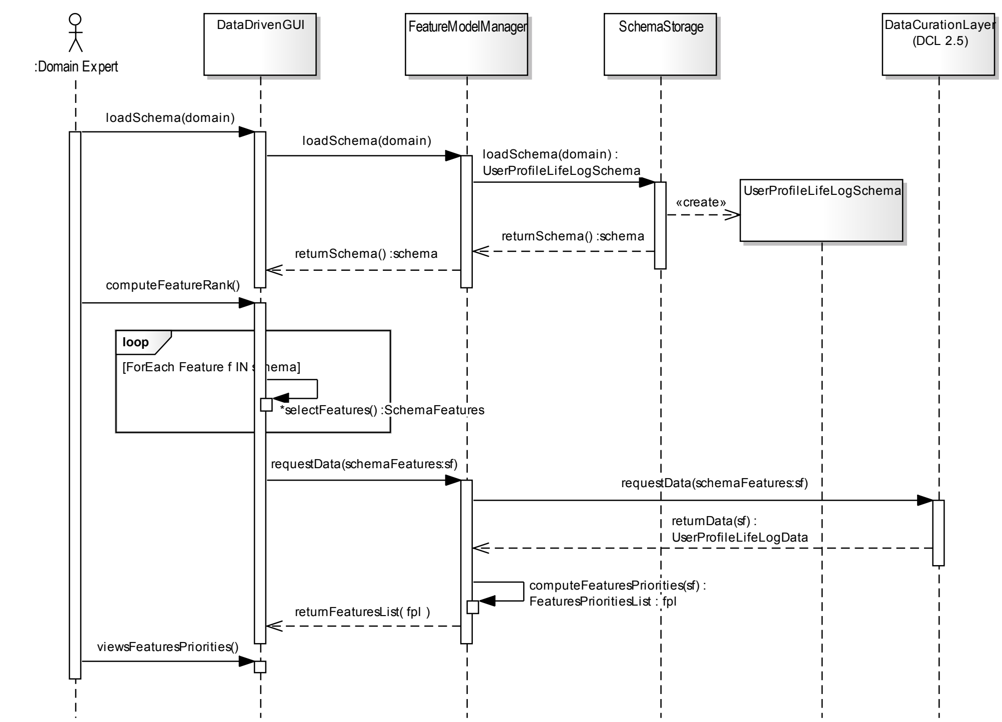
Use Case ID:	**KCL2.5-SUC-16**		
Use Case Name:	Transform Knowledge Rule		
Created By:	Taqdir Ali and Maqbool Hussain	**Last Updated By:**	Maqbool Hussain
Date Created:	27-07-2015	**Last Revision Date:**	15-04-2016
Actors:	Domain Expert		
Description:	The new created rules are needed to transform to some computer interpretable, executable format for execution as well as to shareable, standard format for maintenance and sharing with other organizations.		
Trigger:	Whenever domain expert wants to store the created or updated rule.		
Preconditions:	The expert created rule successfully and the system validated the rule.		
Postconditions:	System shall transform the created and validated rule into appropriate representation.		
Normal Flow:	1. Domain expert save the new created rule or update the existing rule. 2. The system identifies the appropriate representation model 3. Fetch the artifacts of the identified representation model 4. Transforms the rule into the artifacts and syntax of the identified representation model. 5. The rule in the representation model is stored into the repository.		
Alternative Flows:	N/A		
Exceptions:	N/A		
Includes:	N/A		
Frequency of Use:	Whenever domain expert want to save rule		
Special Requirements:	N/A		
Assumptions:	N/A		
Notes and Issues:	Appropriate representation configuration is challenging task.		
Sequence Diagram:			

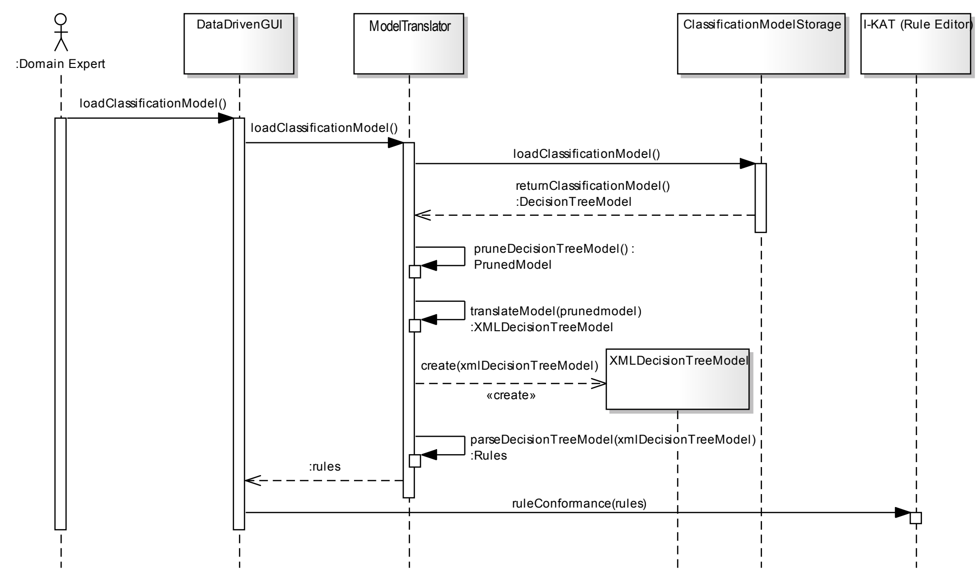

:Domain Expert
Rule Editor
Model Transformation Manager
Representation Model Storage
saveRule()
transformRule(rule)
identifyAppropriateRepresentationModel()
fetchArticatsSyntax()
transformRuleIntoArtifactsSyntax()
storeIntoRepresentationStorage()
acknowledgement()

<table>
<tr><td>Use Case ID:</td><td colspan="3">KCL2.5-SUC-17</td></tr>
<tr><td>Use Case Name:</td><td colspan="3">Create Situation Event</td></tr>
<tr><td>Created By:</td><td>Maqbool Hussain</td><td>Last Updated By:</td><td>Maqbool Hussain</td></tr>
<tr><td>Date Created:</td><td>29-07-2015</td><td>Last Revision Date:</td><td>15-04-2016</td></tr>
<tr><td>Actors:</td><td colspan="3">Domain Expert</td></tr>
<tr><td>Description:</td><td colspan="3">Situation Event is important features of mining mind which includes set of associated recommendation rules. Situation event is created and the rule is indexed in knowledgebase based on situation event.</td></tr>
<tr><td>Trigger:</td><td colspan="3">Whenever domain expert wants to store the created or updated rule.</td></tr>
<tr><td>Preconditions:</td><td colspan="3">The rule has salient features based on which the rule can be indexed.</td></tr>
<tr><td>Postconditions:</td><td colspan="3">• Rule is saved into knowledgebase
• Rule is indexed based on the created situation event</td></tr>
<tr><td>Normal Flow:</td><td colspan="3">1. Domain expert start creating rule;
a. Performs steps 1-5 in KCL2.5-SUC-10.
b. Selects salient features (indicating as event) from conditions of the rule.
2. The system performs following actions;
a. Create situation event with salient features.
b. Saves the situation event and assign index (generate if situation event is not exist in knowledgebase index).
c. Index the created rule with situation event.
3. Domain expert saves the rule by performing steps 5-6 in KCL2.5-SUC-10.
4. System saves the rule and index of the rule.</td></tr>
<tr><td>Alternative Flows:</td><td colspan="3">N/A</td></tr>
<tr><td>Exceptions:</td><td colspan="3">N/A</td></tr>
<tr><td>Extends:</td><td colspan="3">Create Rule (KCL2.5-SUC-10)</td></tr>
<tr><td>Frequency of Use:</td><td colspan="3">Whenever domain expert want to save rule</td></tr>
<tr><td>Special Requirements:</td><td colspan="3">N/A</td></tr>
<tr><td>Assumptions:</td><td colspan="3">N/A</td></tr>
<tr><td>Notes and Issues:</td><td colspan="3"></td></tr>
<tr><td colspan="4">Sequence Diagram:</td></tr>
</table>


:Domain Expert
RuleEditor
SituationEventHandler
KnowledgebaseIndex
Knowledgebase
createRule()
consider Create Rule
assert (openRuleEditor, CreateRule, loop RuleCreation)
l :rule
creatSituation(Set<Facts>salientFeatures)
creatSituationEvent(salientFeatures, rule)
create(salientFeatures)
:situation
SituationEvent
«create»
saveSituation(situation)
assignIndex()
:situationIndex
createSituationIndexforRule(situationIndex, rule)
consider Create Rule
saveRule()
assert (SaveRule)
saveRule(rule)

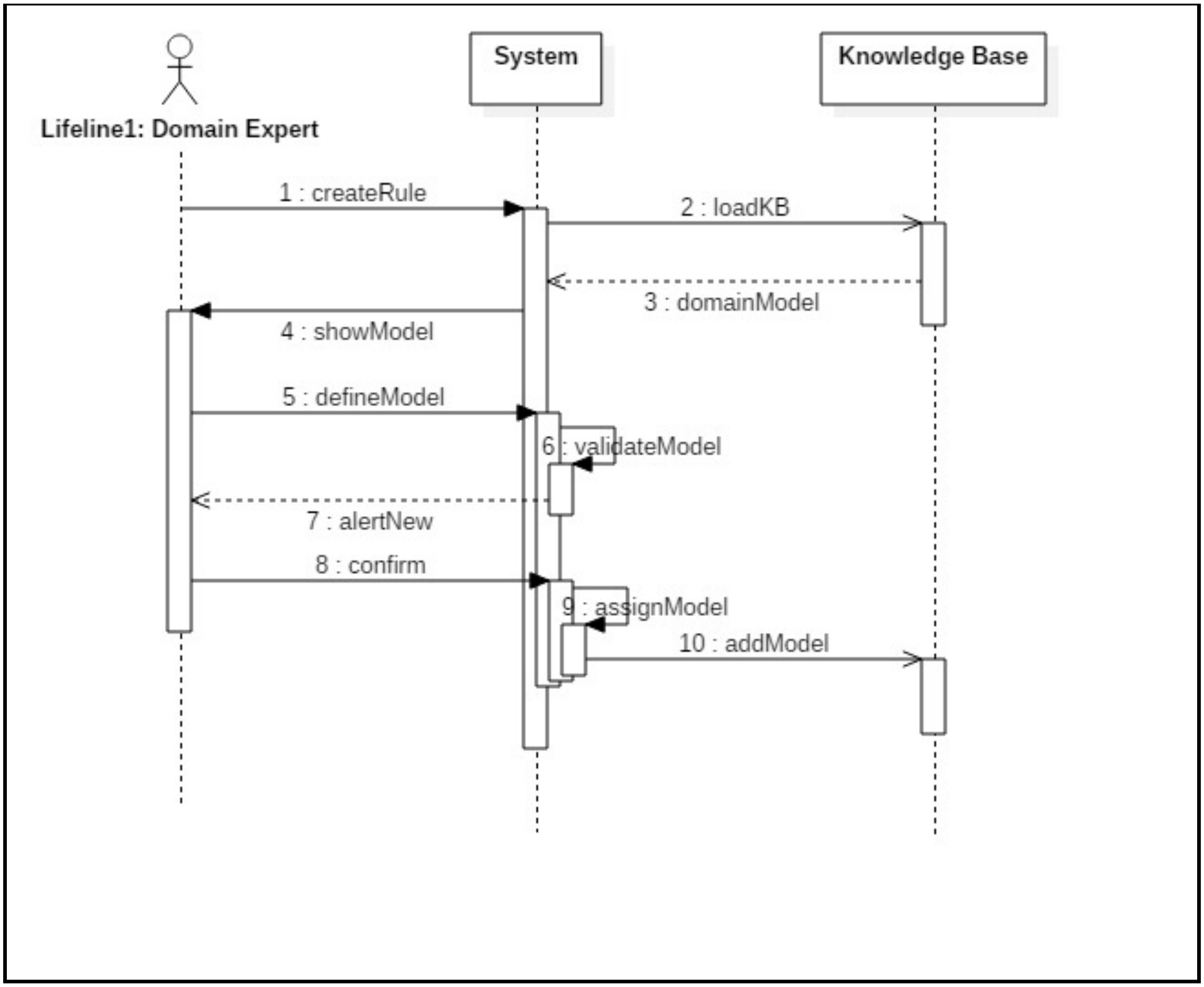
Use Case ID:	**KCL2.5-SUC-18**		
Use Case Name:	Compute features priorities		
Created By:	Maqbool Ali	**Last Updated By:**	
Date Created:	18-03-2016	**Last Revision Date:**	18-03-2016
Actors:	Domain Expert, DCL		
Description:	Features priorities computation help domain expert for selecting appropriate features from available schema		
Trigger:	Prior to data preprocessing needed for required data		
Preconditions:	• KCL has access through service interface to retrieve user profile lifelog data from DCL • KCL and DCL has agreement on common data representation format		
Postconditions:	KCL will display the features priorities list		
Normal Flow:	1. KCL connects to DCL via unified service interface and sends request for user profile lifelog data based on selected features of schema and features conditions. 2. DCL sends the required data to KCL. 3. KCL receives user profile lifelog data 4. KCL computes the features priorities 5. KCL displays the features priorities list		
Alternative Flows:	N/A		
Exceptions:	2a. KCL unable to connect to DCL a. KCL connection is failed during retrieving user profile lifelog data b. KCL hold and will retry after sometime to connect to DCL and retrieve the user profile lifelog data 3a. KCL unable to verify lifelog data conformance a. KCL fail to conform the data representation from DCL b. KCL will send message to DCL about incompatible data format		
Includes:	N/A		
Frequency of Use:	When new service is required and mining mind have sufficient data for classification model creation		
Special Requirements:	N/A		
Assumptions:	N/A		
Notes and Issues:	If DCL is unable to send data, then alternate strategy has to be considered.		

Sequence Diagram:			
Use Case ID:	**KCL2.5-SUC-19**		
Use Case Name:	Transform the decision tree		
Created By:	Maqbool Ali	**Last Updated By:**	
Date Created:	18-03-2016	**Last Revision Date:**	18-03-2016
Actors:	Domain Expert		
Description:	Transform the decision tree generated from classification model to conform the rules from domain expert		
Trigger:	Prior to rules persistence needed for rules verification		
Preconditions:	• KCL has saved the decision tree learning model		
Postconditions:	KCL will transform the decision tree into rules		
Normal Flow:	1. KCL loads the decision tree model 2. KCL prunes the decision tree model 3. KCL translates the pruned model into XML format 4. KCL parse the XML file to extract the rules from XML file 5. KCL connects to knowledge authoring tool via unified service interface 6. KCL shares the parsed rules to knowledge authoring tool for conformance from domain expert.		
Alternative Flows:	N/A		

Exceptions:	3a. KCL unable to connect to knowledge authoring tool a. KCL connection is failed during sharing of transformed rules b. KCL hold and will retry after sometime to connect to knowledge authoring tool and share the transformed rules
Includes:	
Frequency of Use:	When new service is required and mining mind have sufficient data for classification model creation
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	
Sequence Diagram:	

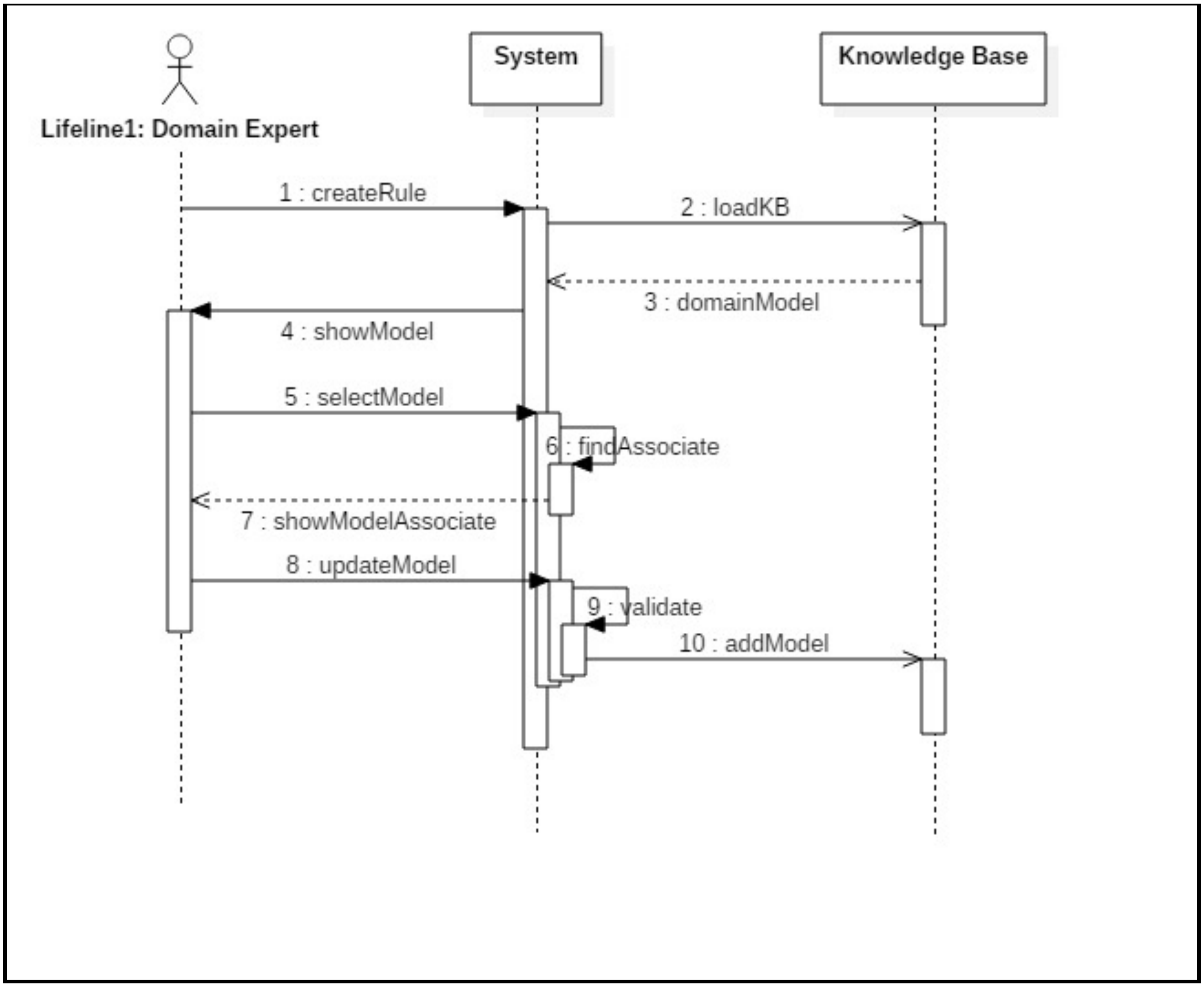
Use Case ID:	**KCL2.5-SUC-20**		
Use Case Name:	Create concept		
Created By:	Sangho Lee	**Last Updated By:**	Sangho Lee
Date Created:	18-3-2016	**Last Revision Date:**	13-4-2016
Actors:	Domain Expert		
Description:	The concepts of wellness domain shall be used in creation for rules and generation of guidelines in tree format. The domain expert shall easily select the required concept from wellness domain model. Therefore this one is provides creation of concepts of domain model.		

Trigger:	Domain expert creates concepts of domain model.
Preconditions:	The expert be authenticated with full access of concepts management in domain model.
Postconditions:	The right concept shall be added at the right location in wellness model.
Normal Flow:	1. Domain expert creates rule or creates guideline. 2. System loads domain model for corresponding domain. 3. Domain expert defines concepts from loaded domain model. 4. System validates duplicated concepts of domain model. 5. Domain expert confirms new create model. 6. System assigns corresponding value to selected concept and show it in rule or guideline tree. 7. Step 4-6 are repeated till rule or guideline is finished.
Alternative Flows:	2a. System finds out duplicated concept model a. System notices duplication alert to domain expert. b. System moves Step 2.
Exceptions:	N/A
Includes:	N/A
Frequency of Use:	Whenever domain expert want to add the concepts in wellness model
Special Requirements:	N/A
Assumptions:	Wellness model repository in the system is exist.
Notes and Issues:	If wellness model storage does not exist in system the administrator shall build the wellness model storage first and configure with system.
Sequence Diagram:	



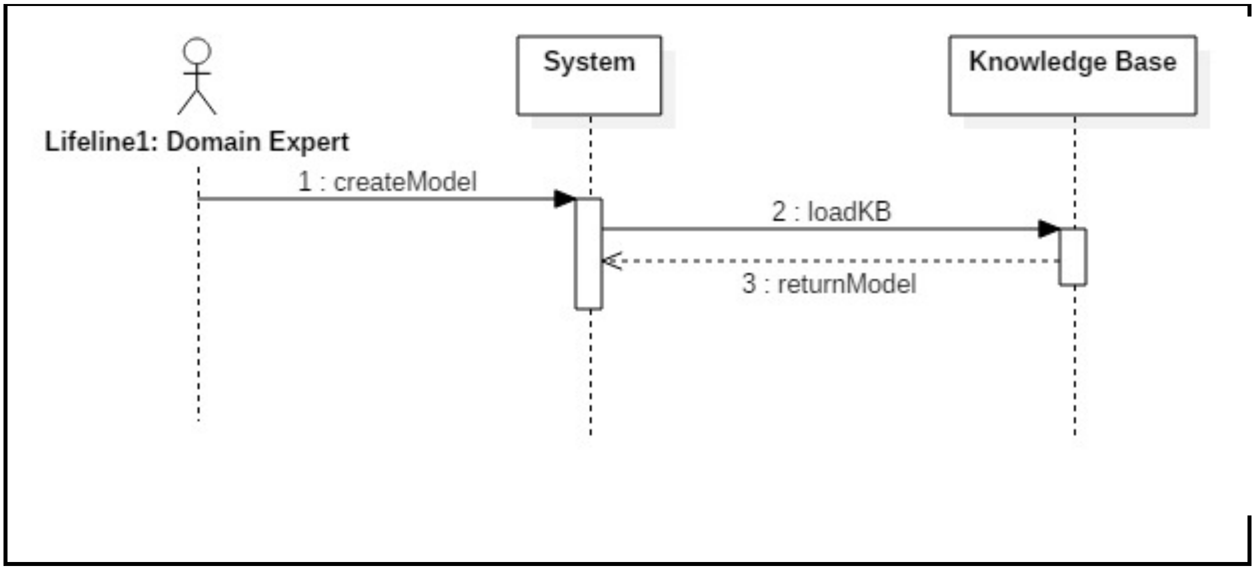
Lifeline1: Domain Expert
System
Knowledge Base
1 : createRule
2 : loadKB
3 : domainModel
4 : showModel
5 : defineModel
6 : validateModel
7 : alertNew
8 : confirm
9 : assignModel
10 : addModel

Use Case ID:	**KCL2.5-SUC-21**		
Use Case Name:	Update concept		
Created By:	Sangho Lee	**Last Updated By:**	Sangho Lee
Date Created:	18-3-2016	**Last Revision Date:**	13-4-2016
Actors:	Domain Expert		
Description:	The concepts of wellness domain shall be used in creation for rules and generation of guidelines in tree format. The domain expert shall easily select the required concept from wellness domain model. Therefore this one is provides updating of concepts of domain model.		
Trigger:	Domain expert updates concepts of domain model.		
Preconditions:	The expert be authenticated with full access of concepts management in domain model		
Postconditions:	The right concept shall be updated at the right location in wellness model		
Normal Flow:	1. Domain expert creates rule (using KCL2.5-SUC-10) or creates guideline (using KCL2.5-SUC-03). 2. System loads domain model for corresponding domain. 3. Domain expert selects concepts from loaded domain model. 4. System associate domain concept to part of rule. 5. Domain expert modifies or deletes value to selected concept. 6. System assigns corresponding value to selected concept. 7. Step 4-6 are repeated till rule or guideline is finished.		
Alternative Flows:	6a. System finds out duplicated concept model a. System notices duplication alert to domain expert. b. System moves Step 2.		
Exceptions:	N/A		
Includes:	N/A		
Frequency of Use:	Whenever domain expert want to edit the concepts in wellness model		
Special Requirements:	N/A		
Assumptions:	Wellness model repository in the system is exist.		
Notes and Issues:			
Sequence Diagram:			



Lifeline1: Domain Expert
System
Knowledge Base
1 : createRule
2 : loadKB
3 : domainModel
4 : showModel
5 : selectModel
6 : findAssociate
7 : showModelAssociate
8 : updateModel
9 : validate
10 : addModel

Use Case ID:	**KCL2.5-SUC-22**		
Use Case Name:	Load concept		
Created By:	Sangho Lee	**Last Updated By:**	Sangho Lee
Date Created:	18-3-2016	**Last Revision Date:**	13-4-2016
Actors:	Domain Expert		
Description:	The concepts of wellness domain shall be used in creation for rules and generation of guidelines in tree format. Therefore Domain Model Manager can provide concepts of domain model to rule editor and model manager.		
Trigger:	Domain Model will be loaded during rule creation or guideline creation.		
Preconditions:	The expert be authenticated with full access of concepts management in domain model		
Postconditions:	Loaded concepts of domain model is ready to use by domain model manager and rule editor		
Normal Flow:	1. Domain expert creates concepts of domain model (using KCL2.5.5-SUC-x) or updates concepts of domain model (using KCL2.5.5-SUC-x). 2. System loads associated domain model for corresponding domain.		
Alternative Flows:	N/A		
Exceptions:	N/A		
Includes:	N/A		
Frequency of Use:	Whenever domain expert access domain model manager and rule editor		
Special Requirements:	N/A		
Assumptions:	Wellness model repository in the system is exist.		
Notes and Issues:			
Sequence Diagram:			



System
Knowledge Base
Lifeline1: Domain Expert
1 : createModel
2 : loadKB
3 : returnModel

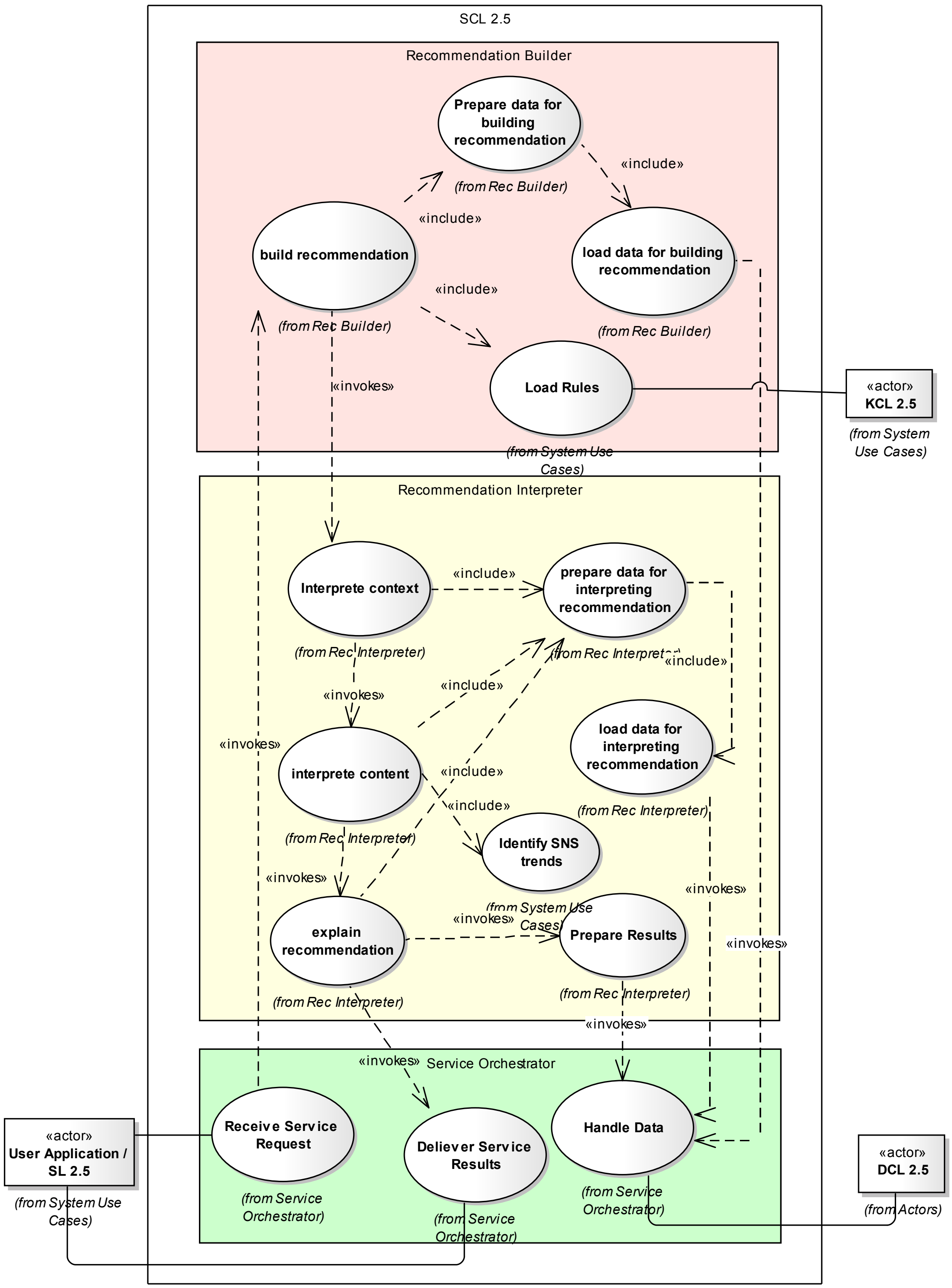
Service Curation Layer (SCL Ver. 2.5)

System Level Use cases

List of Use cases

Use Case #ID	Description
SCL2.5-SUC-01	Load data for building recommendation
SCL2.5-SUC-02	Prepare data for building recommendation
SCL2.5-SUC-03	Load Rules
SCL2.5-SUC-04	Build Recommendation
SCL2.5-SUC-05	Load data for interpreting recommendation
SCL2.5-SUC-06	Prepare data for interpreting recommendation
SCL2.5-SUC-07	Interpret Context
SCL2.5-SUC-08	Interpret Content
SCL2.5-SUC-09	Explain recommendations
SCL2.5-SUC-10	Prepare Results
SCL2.5-SUC-11	Receive service request
SCL2.5-SUC-12	Handle Data
SCL2.5-SUC-13	Deliver service results
SCL2.5-SUC-14	Identify SNS trends

Use case Diagram



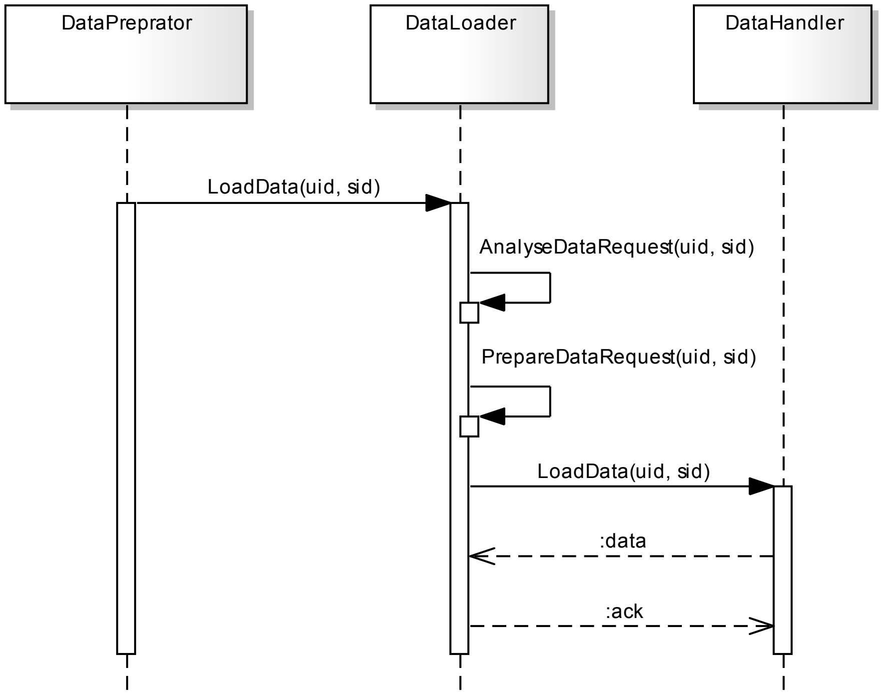
Use case Description

Use Case ID:	SCL2.5-SUC-01		
Use Case Name:	Load data for building recommendation		
Created By:	Rahman Ali	**Last Updated By:**	Rahman Ali
Date Created:	July 14, 2015	**Last Revision Date:**	April 15, 2016

Actors:	SCL2.5-SUC-02 (Prepare Data)
Description:	Retrieving user profile and lifelog data is required for reasoning to generate recommendation. This data is retrieved using Data Handler of the Service Orchestrator.
Trigger:	Triggered when a new service request is received from the user application or DCL 2.5.
Preconditions:	User profile and lifelog data is available in user lifelog.
Postconditions:	User profile and lifelog data is successfully retrieved and prepared for reasoner to process.
Normal Flow:	1. Data Preprator sends request for loading data 2. Data Loader receives the request and performs the following tasks; a. Analyses the request and user for the appropriate data loading b. Prepare separate requests for user lifelog data 3. Data Loader sends analyses request to Data Handler 4. Data Handler provides the data to Data Loader
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Frequency of Use:	Very frequent; repeated for every service request
Special Requirements:	N/A
Assumptions:	Service Orchestrator knows the required data for each registered service.

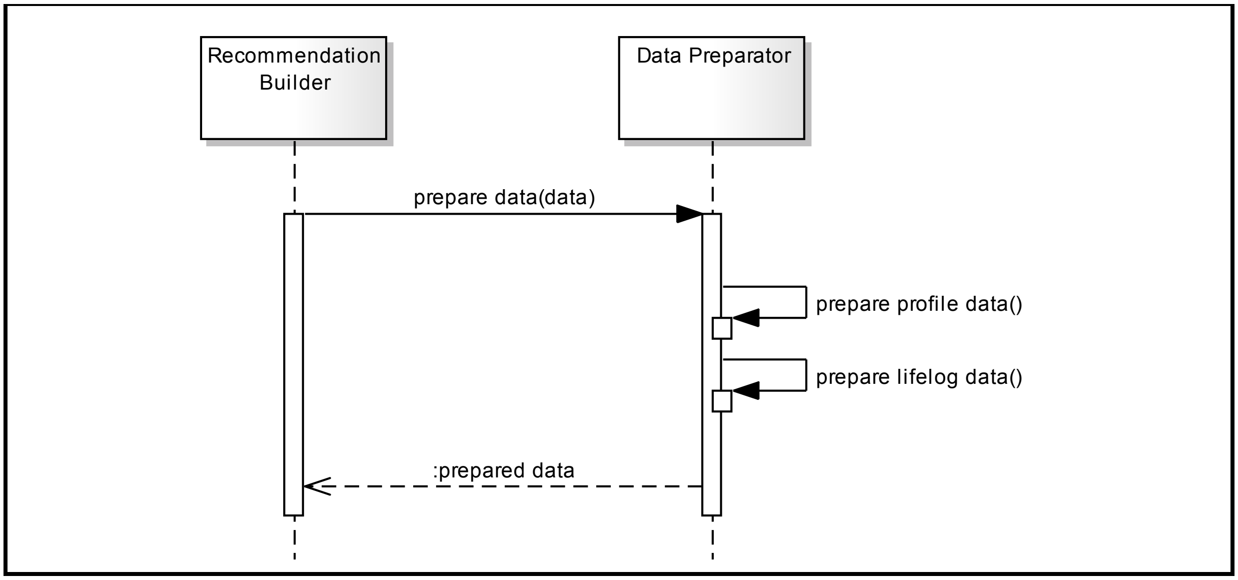
Notes and Issues:	N/A

Sequence Diagram



Use Case ID:	SCL2.5-SUC-02		
Use Case Name:	Prepare data for building recommendation		
Created By:	Rahman Ali	**Last Updated By:**	Rahman Ali
Date Created:	July 14, 2015	**Last Revision Date:**	April 15, 2016

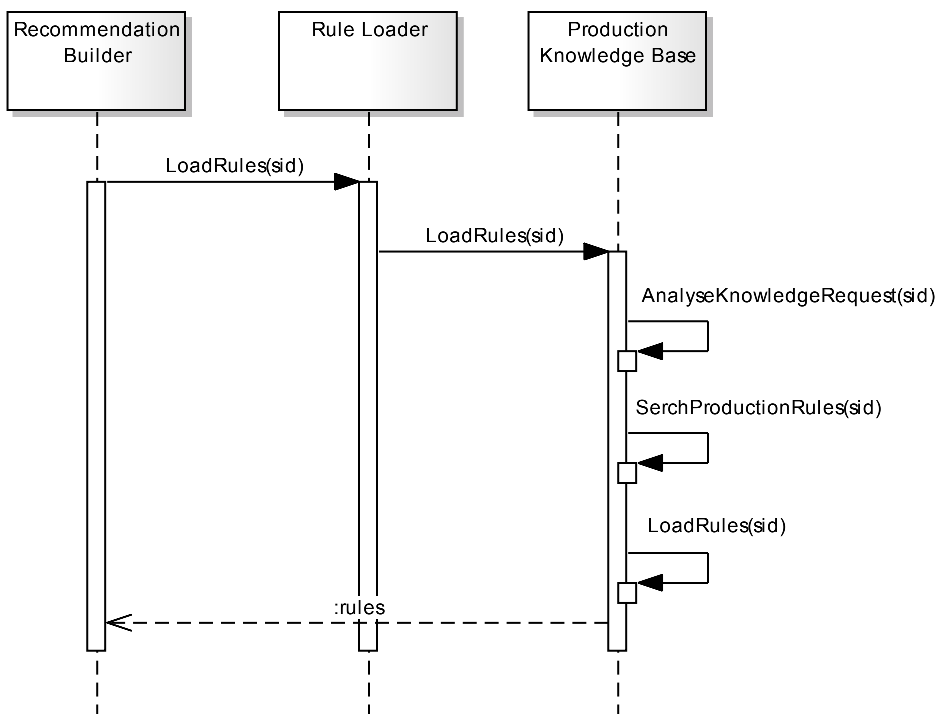
Actors:	SCL2.5-SUC-04 (Build Recommendation)
Description:	Knowledge based reasoning requires prepared data to execute the rules during the reasoning process.
Trigger:	Triggered when new service request is made for generating recommendations
Preconditions:	User profile and lifelog data is loaded into RB 2
Postconditions:	User prepared data is readily available for reasoner to process.
Normal Flow:	1. Recommendation Builder sends data preparation request to Data Preparator along with the loaded data 2. Data Preparator prepares profile data 3. Data Preparator prepares lifelog data 4. Data Preparator returns prepared data to Recommendation Builder
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	SCL2.5-SUC-01
Frequency of Use:	Very frequent; for every service request
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A
Sequence Diagram	



Use Case ID:	SCL2.5-SUC-03		
Use Case Name:	Load Rules		
Created By:	Rahman Ali	**Last Updated By:**	Rahman Ali
Date Created:	April 15, 2016	**Last Revision Date:**	April 15, 2016
Actors:	SCL2.5-SUC-04 (Build Recommendation)		
Description:	Rule-based reasoned needs knowledge rules to perform reasoning using the prepared data to generate recommendations for the service request.		
Trigger:	At the time when new service request arrives for recommendation.		
Preconditions:	• Updated knowledge is available in Production Knowledge Base. • KCL 2.5 and RB 2 agree on common format of production rules.		
Postconditions:	The reasoned is ready to execute the rules and generate recommendations.		
Normal Flow:	1. Recommendation Builders send knowledge load request to Rule Loader 2. Rule Loader sends request to Production Knowledge Base 3. System performs the following tasks;		

	a. Analyses the request knowledge b. Search production knowledge base for the requested rules c. Loads the rules d. Provides the rules back to Recommendation Builder
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Frequency of Use:	Frequent: when reasoner is invoked for new service generation.
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A

Sequence Diagram

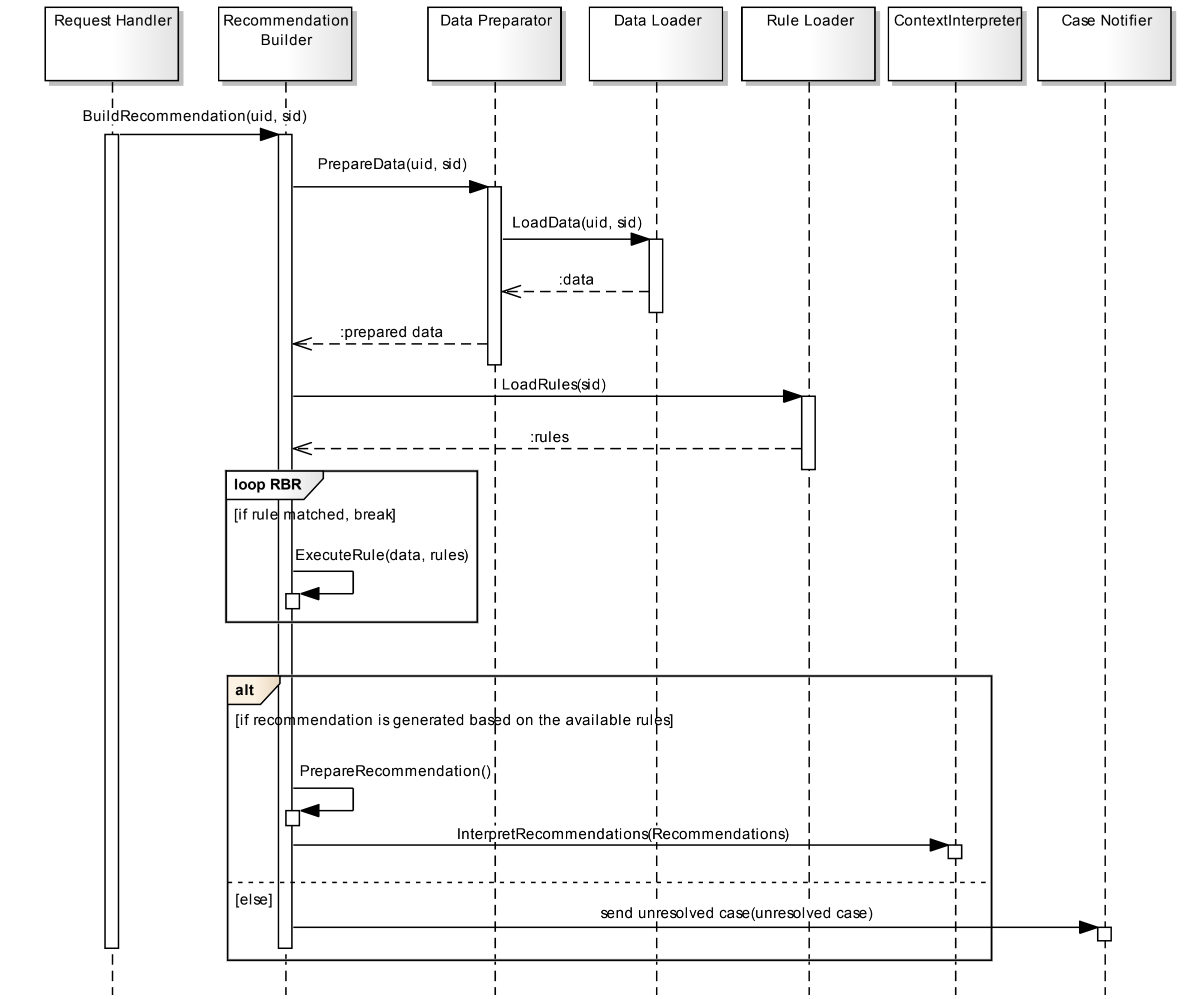


Use Case ID:	SCL2.5-SUC-04		
Use Case Name:	Build recommendations		
Created By:	Rahman Ali	**Last Updated By:**	Rahman Ali
Date Created:	July 15, 2015	**Last Revision Date:**	April 15, 2016

Actors:	SCL2.5-SUC-13 (Request Handler), SCL2.5-SUC-04, SCL2.5-SUC-02, SCL2.5-SUC-01, SCL2.5-SUC-03, SCL2.5-SUC-07 (Interpret Context)
Description:	RBR performs rule-based reasoning to generate recommendations using the production rules and prepared data.
Trigger:	At the time when new service request arrives for recommendation.
Preconditions:	Knowledge is available in Production Knowledge Base.
Postconditions:	The recommendation is reported to RI 2, if reasoning is successful, otherwise the new case is provided to Unified Knowledge Interface along with the missing rule message.
Normal Flow:	1. Request Handler invokes recommendation builder for recommendation 2. Recommendation Builder load prepared data 3. Recommendation Builder retrieves loaded rules 4. Recommendation Builder performs rule-based reasoning on the prepared data and loaded rules 5. Recommendation Builder generates recommendation and perform the following tasks; a. Prepare recommendation b. Provides recommendations to Context Interpreter for interpretation
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	SCL2.5-SUC-02, SCL2.5-SUC-03
Frequency of Use:	Frequent: when recommendation builder is invoked for generating recommendation.
Special Requirements:	N/A

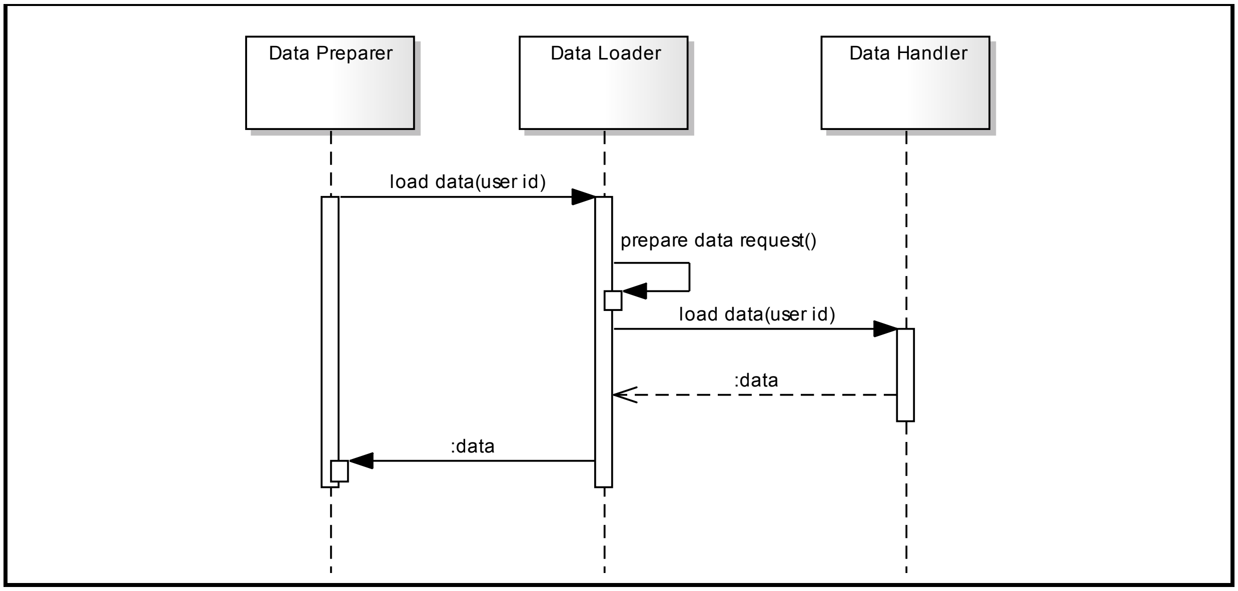
Assumptions:	KCL 2.5 and RB 2 agree on common format of production rules.
Notes and Issues:	N/A

Sequence Diagram



Use Case ID:	SCL2.5-SUC-05		
Use Case Name:	Load data for interpreting recommendations		
Created By:	Muhammad Afzal	**Last Updated By:**	Rahman Ali, Muhammad Afzal
Date Created:	July 15, 2015	**Last Revision Date:**	April 15, 2016

Actors:	Data Preparer
Description:	The data is loaded from DCL 2.5 through Service Orchestrator in order to interpret the recommendations
Trigger:	After recommendation is built
Preconditions:	• Recommendation are built • User profile is stored in lifelog • Context is recognized
Postconditions:	The user profile, lifelog, and environmental variable data is available for preparation.
Normal Flow:	1. Data Loader receives interrupt from Data Preparer 2. Data loader prepare data request 3. Data loader send request to Data Handler 4. Data loader receives data from Data Handler
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Frequency of Use:	Very frequent; at every service request
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A
Sequence Diagram	

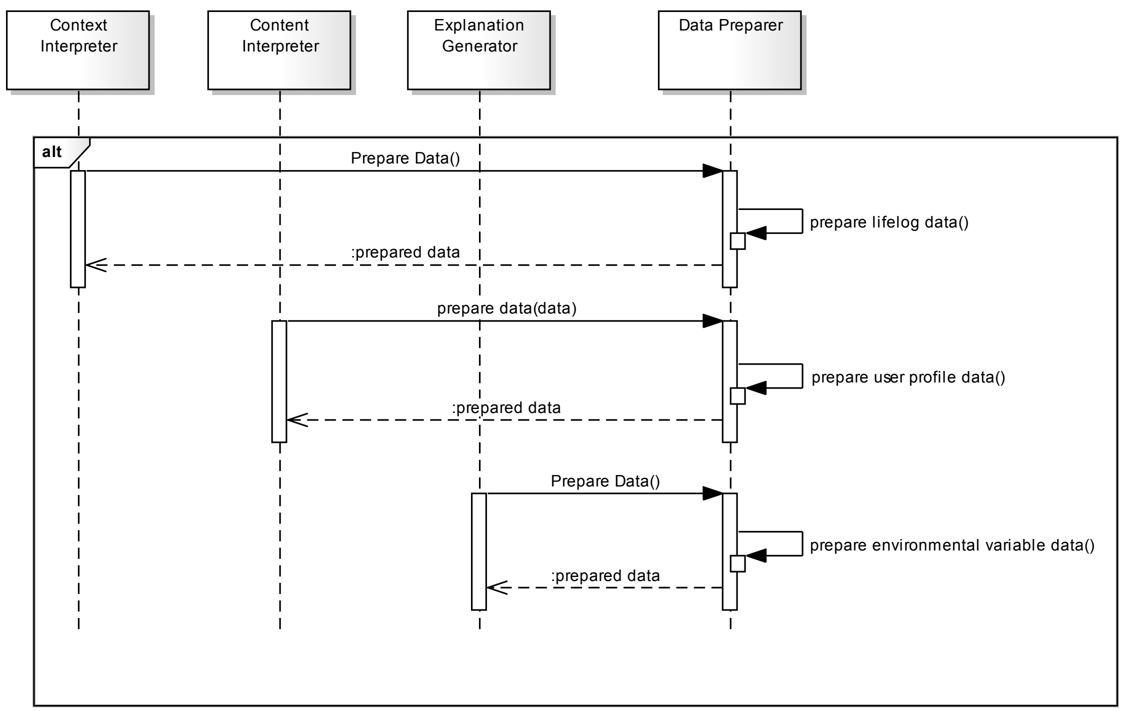


Use Case ID:	SCL2.5-SUC-06		
Use Case Name:	Prepare data for interpreting recommendations		
Created By:	Muhammad Afzal	**Last Updated By:**	Rahman Ali, Muhammad Afzal
Date Created:	July 15, 2015	**Last Revision Date:**	April 15, 2016

Actors:	Context Interpreter
Description:	The loaded data is prepared for interpretations according to different functions such as lifelog for contextual interpretations, user profile for content interpretations, and environmental variables for explanations.
Trigger:	After loading data for interpretations
Preconditions:	• Recommendation are built • Data is loaded
Postconditions:	The user profile, lifelog, and environmental variable data is prepared and is available for interpretations
Normal Flow:	1. Context Interpreter sends data to Data Preparer for preparations 2. Data Preparer prepares lifelog data 3. Content Interpreter sends request to Data Preparer for preparing profile data

	4. Data Preparer prepares profile data 5. Data Preparer prepares environmental variable data for the Explanation Generator
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	SCL2.5-SUC-05
Frequency of Use:	Very frequent: at every service request
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	

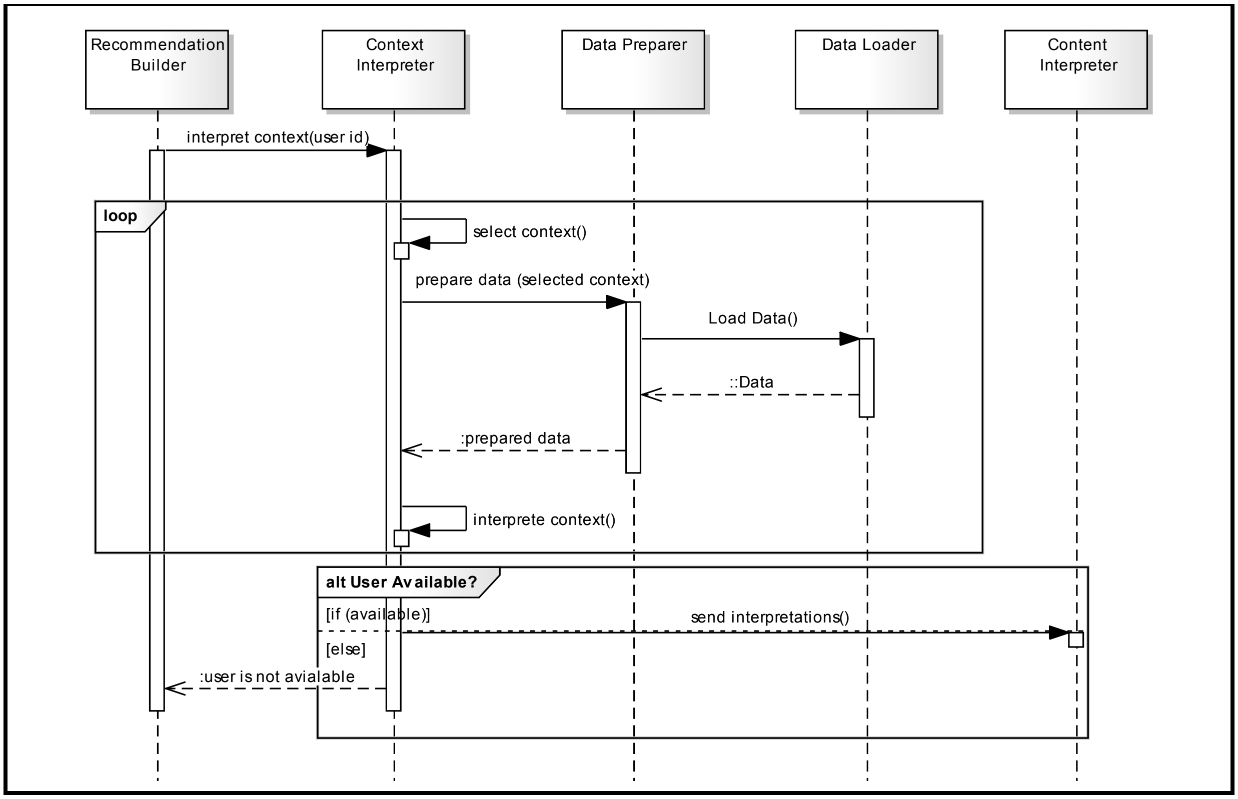
Sequence Diagram



Use Case ID:	SCL2.5-SUC-07
Use Case Name:	Interpret context

Created By:	Muhammad Afzal	**Last Updated By:**	Rahman Ali, Muhammad Afzal
Date Created:	July 15, 2015	**Last Revision Date:**	April 15, 2016

Actors:	Build Recommendation (SCL2.5-SUC-04)
Description:	The loaded data is prepared for interpretations according to different functions such as lifelog for contextual interpretations, user profile for content interpretations, and environmental variables for explanations.
Trigger:	After loading data for interpretations
Preconditions:	Recommendation are built and data is loaded
Postconditions:	The user profile, lifelog, and environmental variable data is prepared and is available for interpretations
Normal Flow:	1. Context Interpreter receives request for from Recommendation Interpreter for context interpretation 2. Context Interpreter load and prepare data lifelog data (contextual data) for interpretations. 3. Context interpreter select a context 4. Context interpreter interprets the context 5. Repeat 2-4 until all applicable contexts interpreted 6. Context Interpreter receives the interpreted context 7. Context Interpreter sends the recommendations to content interpreter for interpreting the contents
Alternative Flows:	7a. if user is not available then the process is halt and message is sent to Recommendation Builder.
Exceptions:	N/A
Includes:	SCL2.5-SUC-06
Frequency of Use:	Very frequent; at every service request
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A
Sequence Diagram	

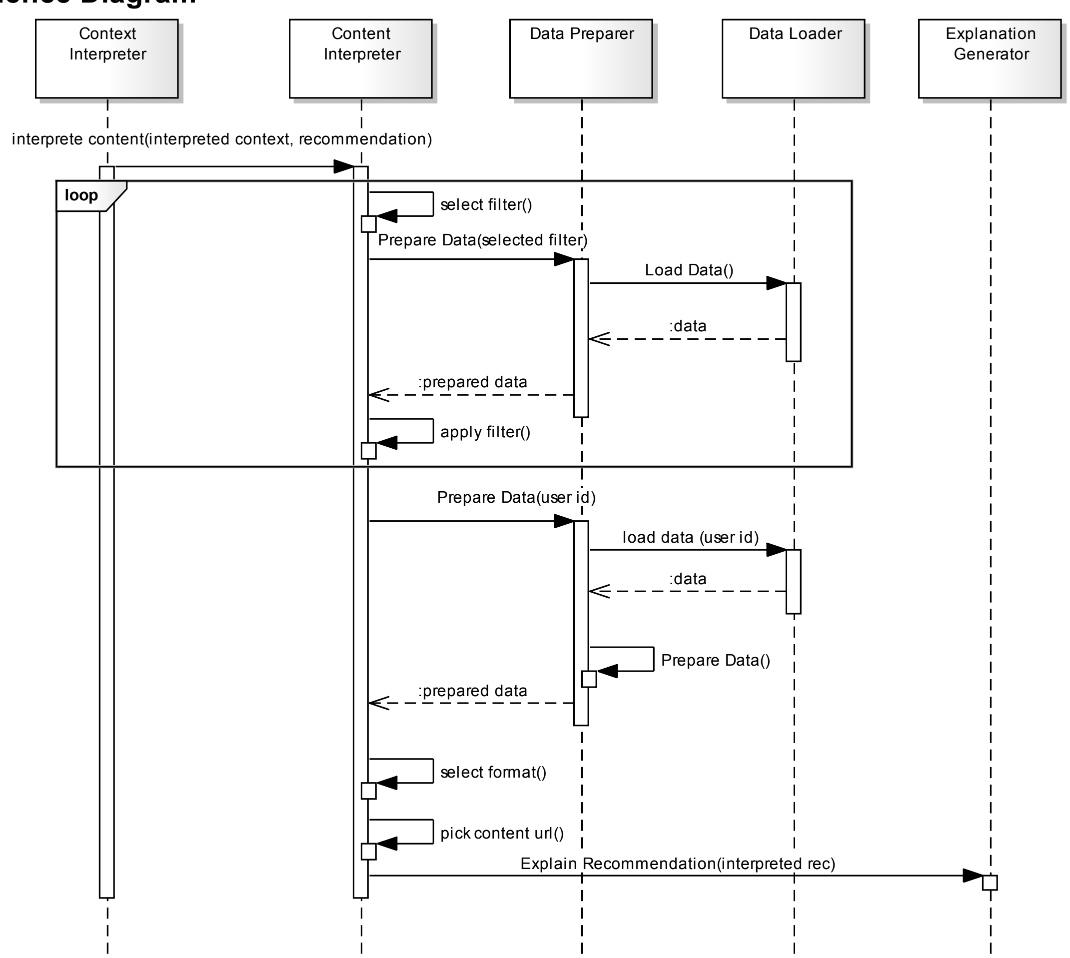


Use Case ID:	SCL2.5-SUC-08		
Use Case Name:	Interpret contents		
Created By:	Muhammad Afzal	**Last Updated By:**	Rahman Ali
Date Created:	July 15, 2015	**Last Revision Date:**	April 15, 2016

Actors:	Interpret Context (SCL2.5-SUC-07)
Description:	The recommended contents of recommendations are difficult for user to understand. These contents needs to be interpreted with support of multimedia contents.
Trigger:	After interpretation of the context
Preconditions:	Recommendations are generated and context is interpreted
Postconditions:	Recommendations are ready for explanation
Normal Flow:	1. Context Interpreter sends the contextually interpreted recommendations to the content filterer. 2. Content interpreter perform the following tasks; a. Select appropriate filter

	b. Applies the filter 3. Step 2 is repeated for all filters 4. Content interpreter selects the appropriate format 5. Content interpreter adds the relevant url 6. Content interpreter forwards the format and filtered contents to explanation generator
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	SCL2.5-SUC-06
Frequency of Use:	Very frequent: when recommendation are generated
Special Requirements:	The format should be defined in advanced based on the user special conditions
Assumptions:	N/A
Notes and Issues:	N/A

Sequence Diagram

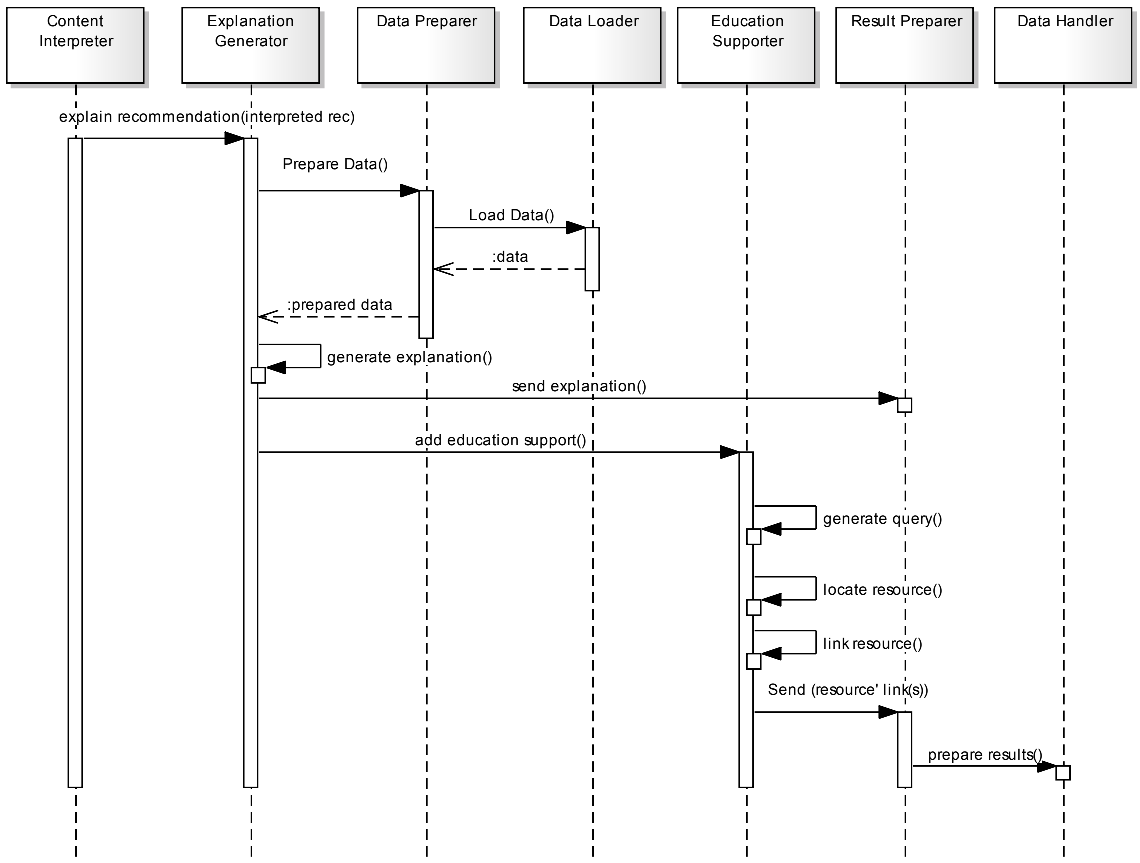


Use Case ID:	SCL2.5-SUC-09		
Use Case Name:	Explain Recommendations		
Created By:	Muhammad Afzal	**Last Updated By:**	Rahman Ali
Date Created:	July 15, 2015	**Last Revision Date:**	April 15, 2016

Actors:	Interpret content (SCL2.5-SUC-08)
Description:	Usually user don't understand the contents of recommendations. To make them understandable the interpreted recommendations needs to be explained based on the user understandability.
Trigger:	When contents are interpreted
Preconditions:	Recommendations are interpreted
Postconditions:	Recommendations are ready to deliver to the user
Normal Flow:	1. Explanation generator receives the interpreted recommendations from content interpreter. 2. Explanation generator performs the following tasks; a. Select environment variable b. Generate explanation 3. Explanation Generator sends explained recommendation to educational support handler 4. Educational support handler performs the following tasks a. generate query b. locate resource c. link resource d. send resource link to interpreter 5. System sends explanation and educational resource links to result preparer.
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	SCL2.5-SUC-06
Frequency of Use:	Very frequent: when recommendations are interpreted
Special Requirements:	N/A
Assumptions:	N/A

Notes and Issues:	N/A

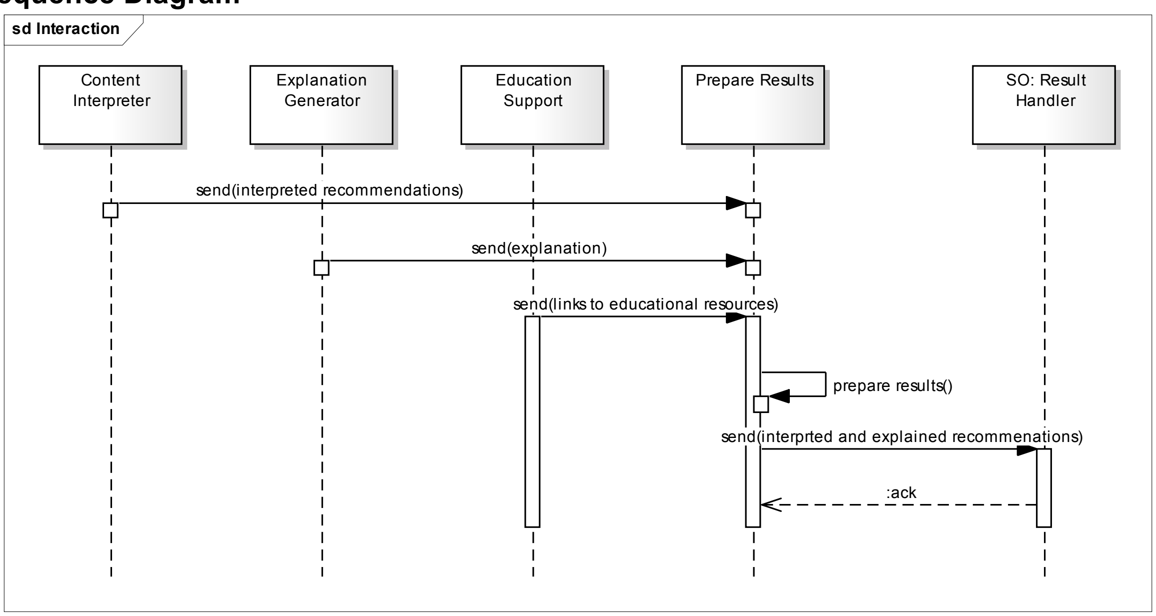
Sequence Diagram



Use Case ID:	SCL2.5-SUC-10		
Use Case Name:	Prepare Results		
Created By:	Muhammad Afzal	**Last Updated By:**	Rahman Ali, Muhammad Afzal
Date Created:	July 5, 2015	**Last Revision Date:**	April 15, 2016
Actors:	Explain Recommendation (SCL2.5-SUC-09), Interpret Contents (SCL2.5-SUC-08)		
Description:	This use case prepare the results accumulated from explanation generator and content interpreter and forwards to results handler of service orchestrator.		

Trigger:	When recommendation are interpreted and explained
Preconditions:	The recommendation are interpreted and explained
Postconditions:	The results are forwarded to service orchestrator.
Normal Flow:	1. Result Preparer receives outputs from content interpreter and/or explanation generator as well as education support. 2. Result Preparer combines the received results 3. Result Preparer sends the results to result handler of service orchestrator
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Frequency of Use:	Frequent
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A

Sequence Diagram

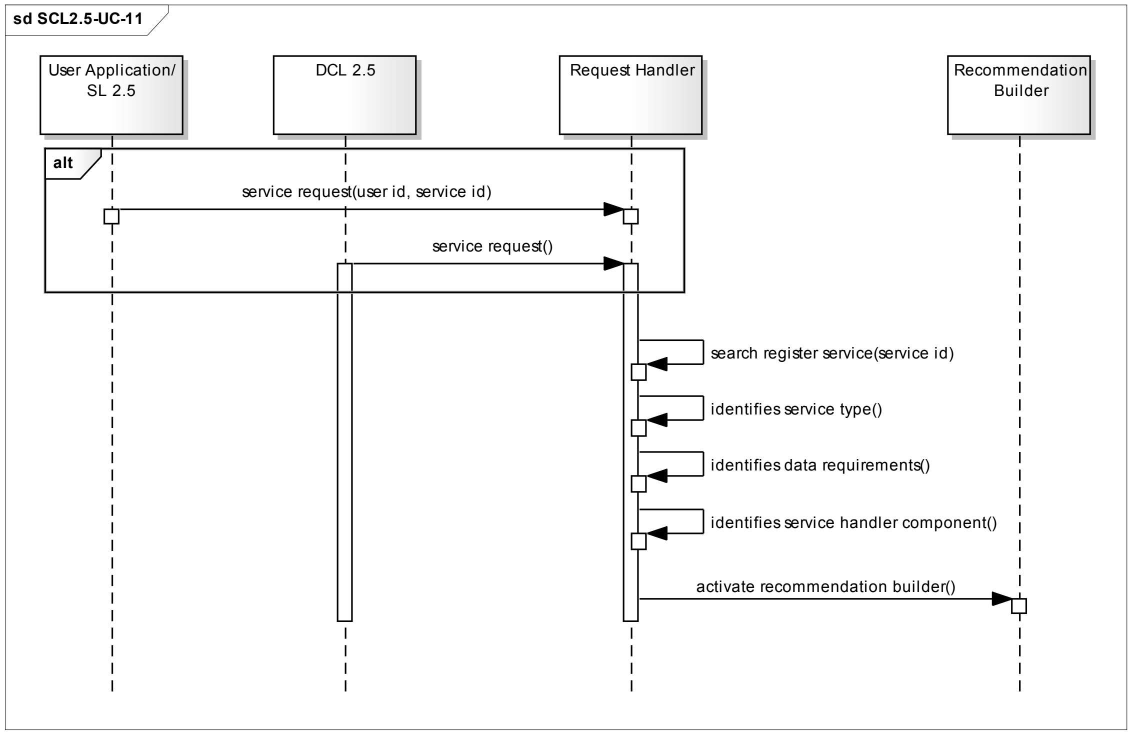


Use Case ID:	SCL2.5-SUC-11
Use Case	Receive service request

<table>
<tr><td>Name:</td><td colspan="3"></td></tr>
<tr><td>Created By:</td><td>Muhammad Afzal</td><td>Last Updated By:</td><td>Rahman Ali,
Muhammad Afzal</td></tr>
<tr><td>Date Created:</td><td>July 5, 2015</td><td>Last Revision Date:</td><td>April 15, 2016</td></tr>
<tr><td>Actors:</td><td colspan="3">User Application / SL 2.5</td></tr>
<tr><td>Description:</td><td colspan="3">Service orchestrator receives request from user application, or DCL 2.5 for recommendation. Orchestrator parses the request and invokes required service of Mining Mind for responding.</td></tr>
<tr><td>Trigger:</td><td colspan="3">At the time of a request from the user application, or from mining mind generated events.</td></tr>
<tr><td>Preconditions:</td><td colspan="3">• User is registered with Mining Minds
• Service is registered as mining Minds valid service
• Service-data binding is specified in advance</td></tr>
<tr><td>Postconditions:</td><td colspan="3">The request is received and recommendations are generated</td></tr>
<tr><td>Normal Flow:</td><td colspan="3">1. Service orchestrator receives the service request from user application
2. System parses the request
a. Search for the registered service
b. Identifies the service type
c. Identifies data requirements of the service
d. Identifies the appropriate handling module
3. Service Orchestrator passes the request to recommendation builder of SCL 2.5 to build the recommendation</td></tr>
<tr><td>Alternative Flows:</td><td colspan="3">1.a Event handler of service orchestrator receives the request as an interrupt from DCL 2.5, whenever a situation occurs
4.a Step 2-4 of the normal flow are executed.</td></tr>
<tr><td>Exceptions:</td><td colspan="3">N/A</td></tr>
<tr><td>Includes:</td><td colspan="3">N/A</td></tr>
<tr><td>Frequency of Use:</td><td colspan="3">Very frequent: at every service request</td></tr>
<tr><td>Special Requirements:</td><td colspan="3">N/A</td></tr>
<tr><td>Assumptions:</td><td colspan="3">Service orchestrator and DCL 2.5 agreed on service</td></tr>
</table>

	contract.
Notes and Issues:	

Sequence Diagram

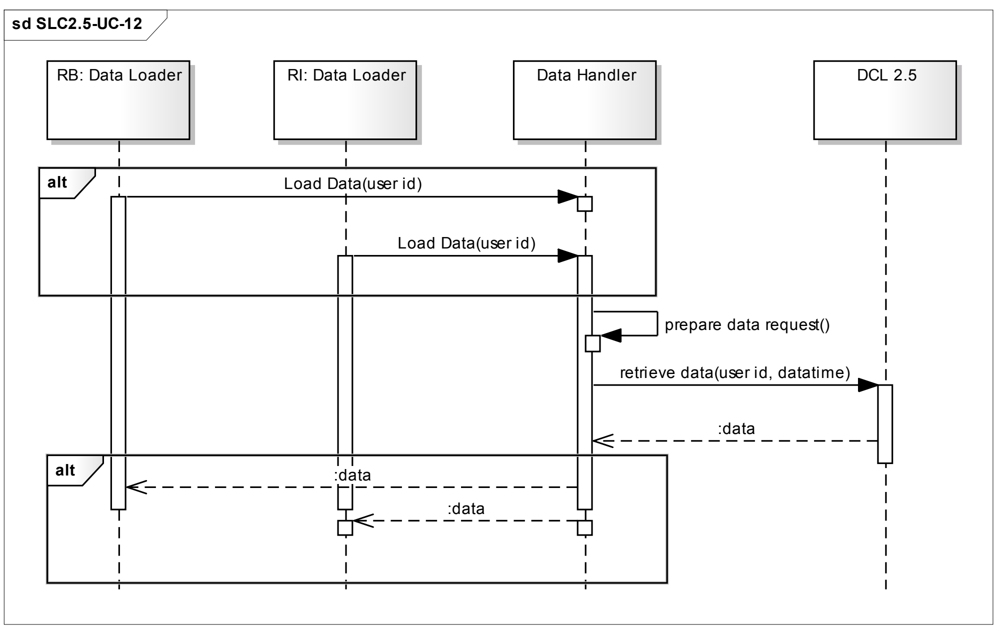


Use Case ID:	SCL2.5-SUC-12		
Use Case Name:	Handle Data		
Created By:	Muhammad Afzal	**Last Updated By:**	Muhammad Afzal
Date Created:	April 15, 2016	**Last Revision Date:**	April 15, 2016

Actors:	DCL 2.5
Description:	This use case receives data request from recommendation builder and recommendation interpreter. It makes request from DCL 2.5 to get the data for requester.
Trigger:	At data request time
Preconditions:	Service Request has been received to service orchestrator
Postconditions:	Data has been provided to requester
Normal Flow:	1. Data Handler in Service Orchestrator received data

	loading request from recommendation builder 2. Prepare data request 3. Retrieve data from DCL 2.5 4. Send data to RB: Data Loader
Alternative Flows:	1a. Data Handler in Service Orchestrator received data loading request from recommendation interpreter Step 2-3 of normal flow 4a. Send data to RI: Data Loader
Exceptions:	N/A
Includes:	N/A
Frequency of Use:	Very Frequent: At every service request
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A

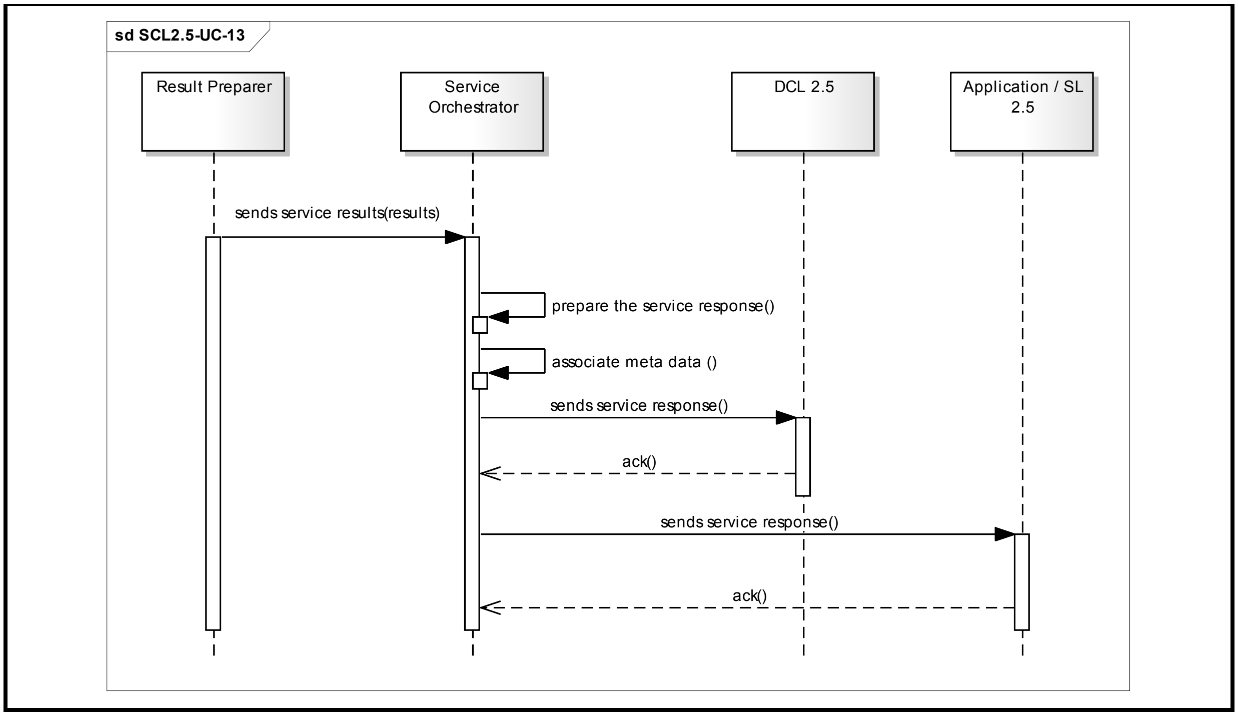
Sequence Diagram



Use Case ID:	SCL2.5-SUC-13		
Use Case Name:	Deliver service results		
Created By:	Muhammad Afzal	**Last Updated By:**	Rahman Ali,

			Muhammad Afzal
Date Created:	July 5, 2015	**Last Revision Date:**	April 15, 2016

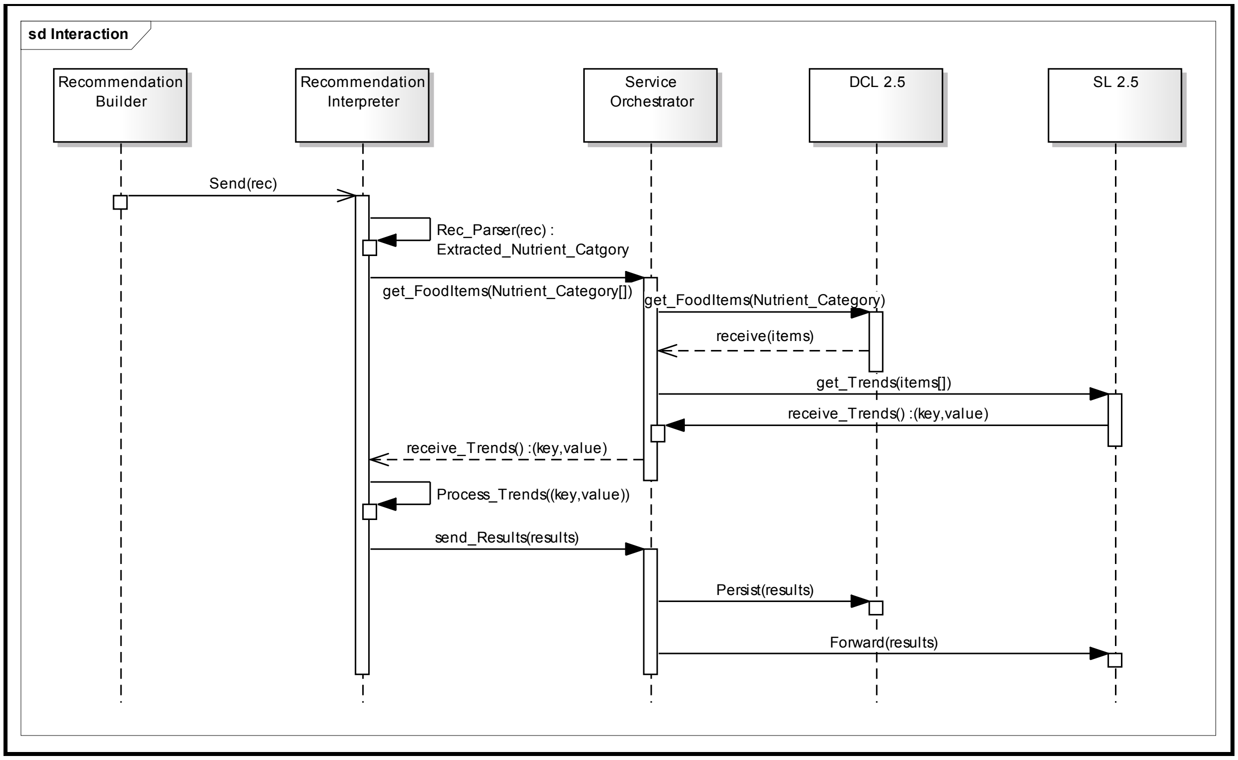
Actors:	Application / SL 2.5
Description:	It is required to send request response to the service requester in the form of recommendation. Service orchestrator delivers the interpreted recommendation to user.
Trigger:	At the time of completion of interpretations
Preconditions:	Recommendations are generated and interpreted
Postconditions:	Service results are successfully delivered to the requester and DCL 2.5 for persistence
Normal Flow:	1. Service orchestrator receives results from recommendation interpreter 2. System perform the following tasks; a. Prepares the response message b. Associate recommendations with service meta-data 3. Service Orchestrator sends recommendations to DCL 2.5 for persistence 4. Service Orchestrator receives acknowledgement of storage 5. Service Orchestrator sends interpreted recommendations to SL 2.5 6. Service Orchestrator receives acknowledgement of receipt
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Frequency of Use:	Very frequent: at every service request completion
Special Requirements:	N/A
Assumptions:	Service orchestrator agreed on service contract with DCL 2.5 and user application.
Notes and Issues:	N/A
Sequence Diagram	



Use Case ID:	SCL2.5-SUC-14		
Use Case Name:	Identify SNS trends		
Created By:	Muhammad Afzal	**Last Updated By:**	Imran Ali
Date Created:	April 12, 2016	**Last Revision Date:**	April 15, 2016

Actors:	SCL2.5-SUC-10, SL 2.5
Description:	For trend analysis, it is required to extract keywords from the nutrition recommendation and provide to supporting layer for retrieving food trends from external entity (TAPACROSS).
Trigger:	At the time of result generation
Preconditions:	Recommendations are generated and interpreted
Postconditions:	SNS trends are appended to the interpreted recommendation
Normal Flow:	1. Recommendation Interpreter receives the recommendation from Recommendation Builder 2. Recommendation Interpreter extracts nutrient category from the recommendation 3. Recommendation Interpreter sends nutrient category

	to Service Orchestrator 4. Service Orchestrator gets nutrient items from DCL 2.5 5. Service Orchestrator sends nutrient items to SL 2.5 6. System receives the SNS trends from SL 2.5 7. Recommendation Interpreter receives SNS trends from Service Orchestrator 8. Recommendation Interpreter process the received trends with respect to user preferences and select the matched or top trends from the list. 9. Recommendation Interpreter forwards the trends to result preparer to prepare the final recommendations. 10. Recommendation Interpreter sends the final recommendation to Service Orchestrator 11. Service Orchestrator sends the recommendation in DCL 2.5 for persistence. 12. Service Orchestrator forwards the recommendation to SL 2.5
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Frequency of Use:	Frequent: every time when the recommendation are interpreted.
Special Requirements:	N/A
Assumptions:	SCL 2.5 agreed on service contract with SL 2.5
Notes and Issues:	N/A
Sequence Diagram	



sd Interaction
Recommendation Builder
Recommendation Interpreter
Service Orchestrator
DCL 2.5
SL 2.5
Send(rec)
Rec_Parser(rec) : Extracted_Nutrient_Catgory
get_FoodItems(Nutrient_Category[])
get_FoodItems(Nutrient_Category)
receive(items)
get_Trends(items[])
receive_Trends() :(key,value)
receive_Trends() :(key,value)
Process_Trends((key,value))
send_Results(results)
Persist(results)
Forward(results)

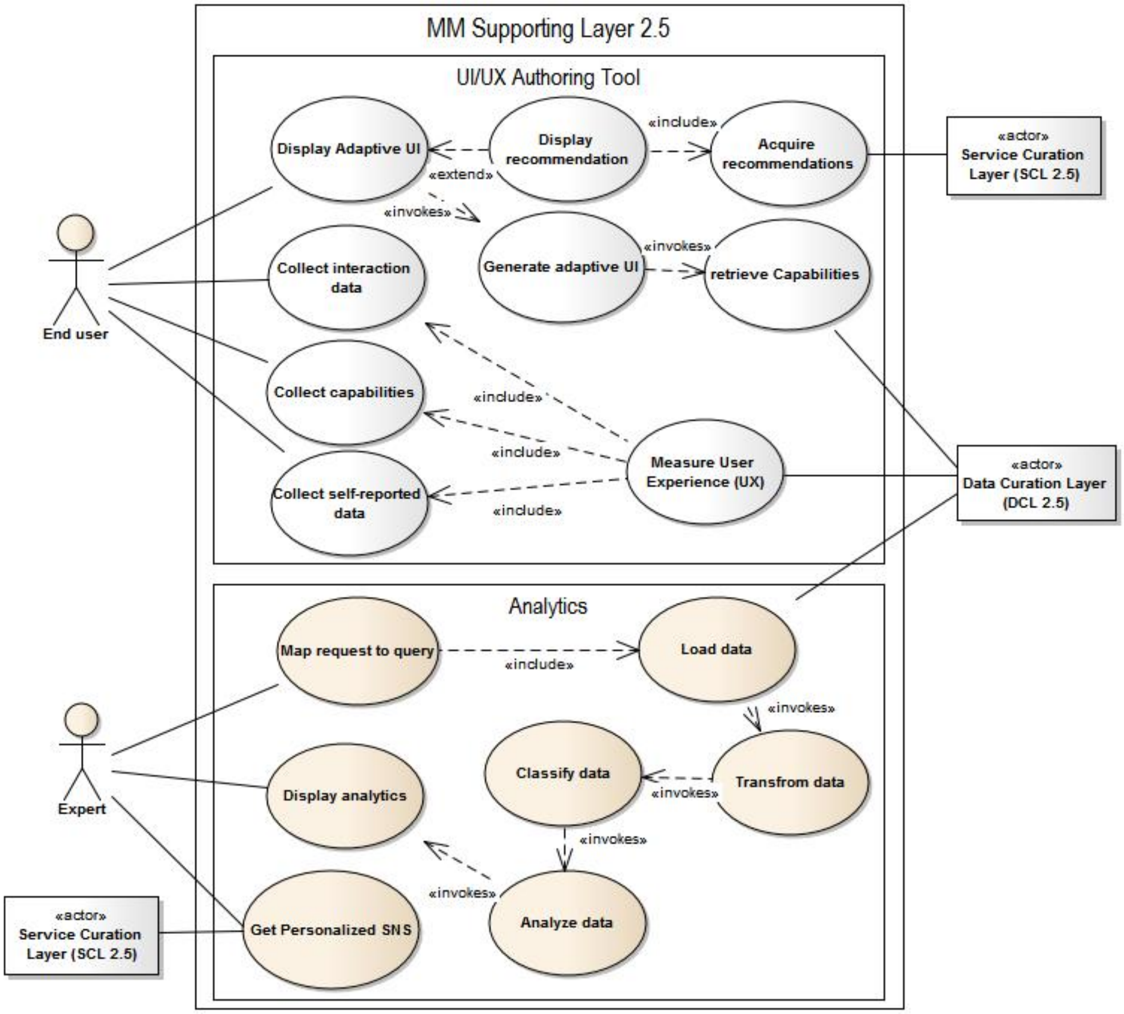
Supporting Layer (SL Ver. 2.5)

System Level Use cases

List of Use cases

Use case ID#	**Description**
SL2.5-UCS-01	User registration
SL2.5-UCS-02	Retrieve capabilities for user interface adaptation
SL2.5-UCS-03	Mapping the user capability information into model
SL2.5-UCS-04	Adapt user interface based on user profile, context and device
SL2.5-UCS-07	Collect and analyze observational data
SL2.5-UCS-08	Acquire Recommendations for displaying to end user
SL2.5-UCS-09	User capabilities collection
SL2.5-UCS-10	Self-reporting user experience measurement
SL2.5-UCS-11	Map Request to Query
SL2.5-UCS-12	Transform Data
SL2.5-UCS-13	Classify Data
SL2.5-UCS-14	Analyze Data
SL2.5-UCS-15	Display Analytics
SL2.5-UCS-16	Get Personalized SNS
SL2.5-UCS-17	Inference attack detection
SL2.5-UCS-18	Recommendation Integrity Check

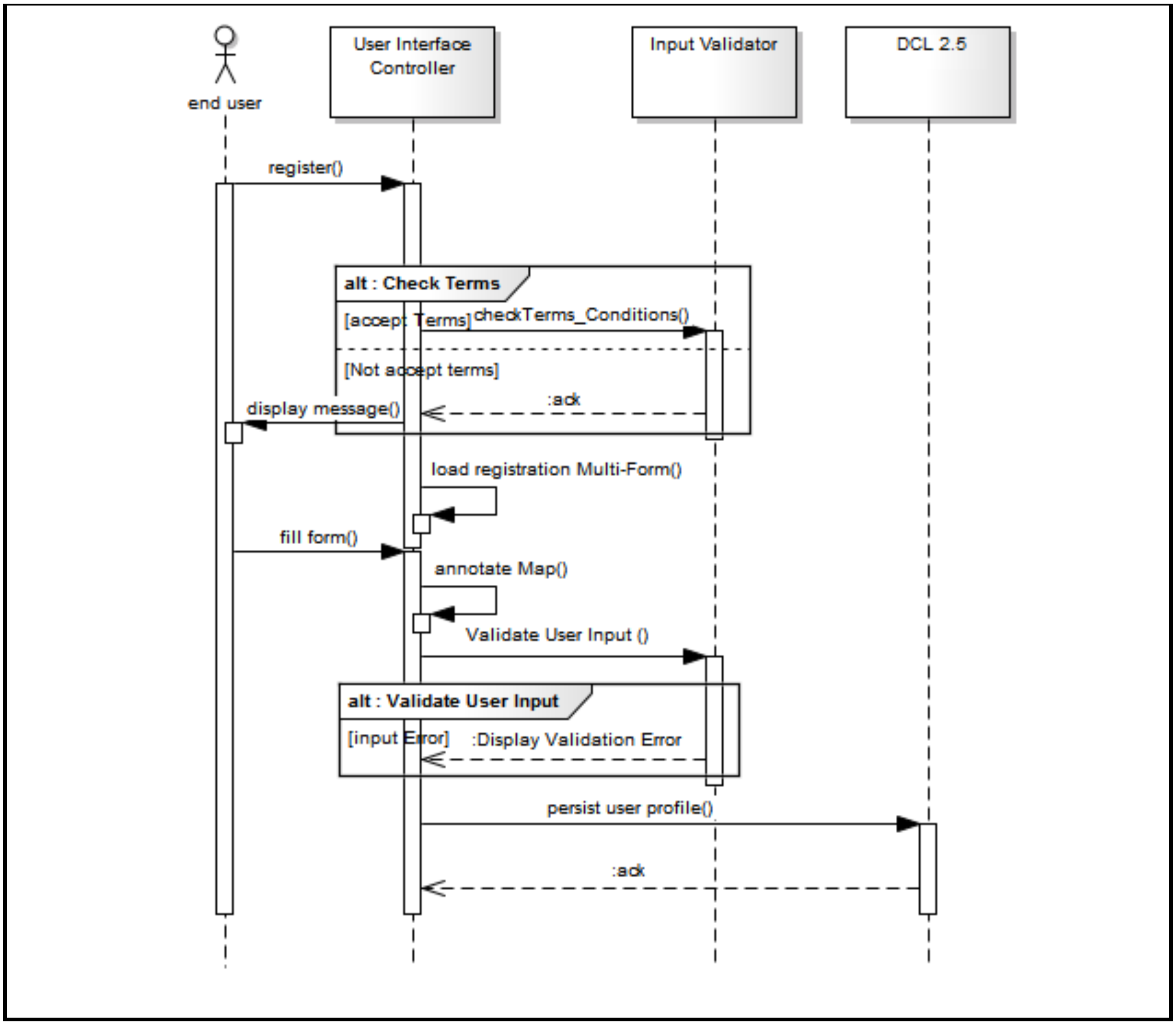
Use case Diagram



Use case Description

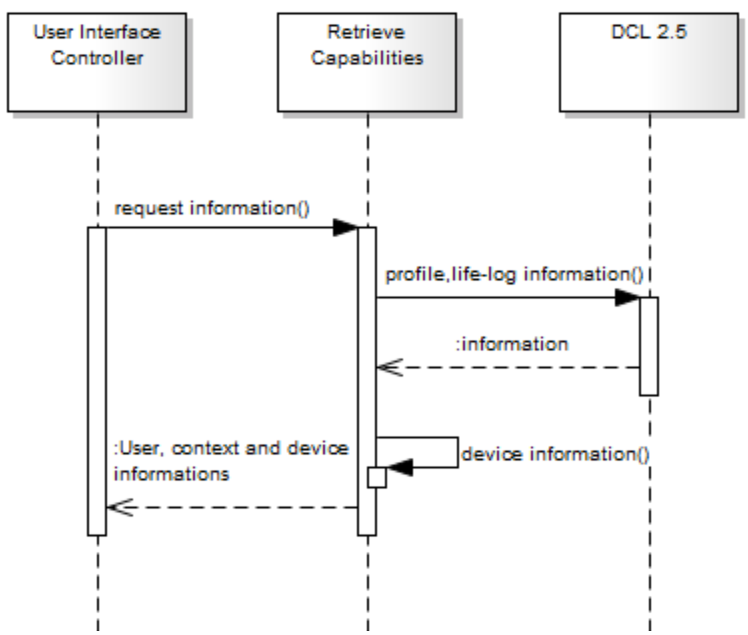
Use Case ID:	SL2.5-UCS-01		
Use Case Name:	User registration		
Created By:	Jamil Husain	**Last Updated By:**	Jamil Husain
Date Created:	14 July 2015	**Last Revision Date:**	15 April 2016
Actors:	Primary: End-user Secondary: DCL 2.5		
Description:	This use case is for the user registration. A user must register with the MM app before they are able to use it. Registration primarily consists of entering an email address for verification and creating a password. All basic		

	demographics, account, activity level, user interest information, and personalized map information shall be collect from user and persist in DCL 2.5.
Trigger:	End user
Preconditions:	The non-register user asked the application to register to it.
Postconditions:	The user successfully registered to the application and can access its functionality
Normal Flow:	1. The user start registration of the new account by pressing the “Sig up” button on the application first screen. 2. Then **Terms & Conditions** page displayed 3. The application will display the multi-step registration form with empty fields for the account and user profile. 4. The user annotates the map for personalization by selecting different locations of his interest. 5. Validate User Input • The application will automatically validate all the user input for all the required fields • The user cannot proceed until providing the correct data. 6. The user can press “Submit” button and the new account data will be persisted to the DCL 2.5.
Alternative Flows:	2a. In step 2. If the user Agree with Terms & Conditions • User is allowed to the next step by click on agreed term & condition checkbox. 2b. In step 2. If user not agree with terms & conditions then • User is redirect to the first screen. 3a. in step 3. Display Validation Error • If the validation failed, then the validation icon will be displayed nearby the wrong field and there will be validation message.
Exceptions:	N/A
Includes:	N/A
Frequency of Use:	When user first time use the system [low]
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A
Sequence Diagram	



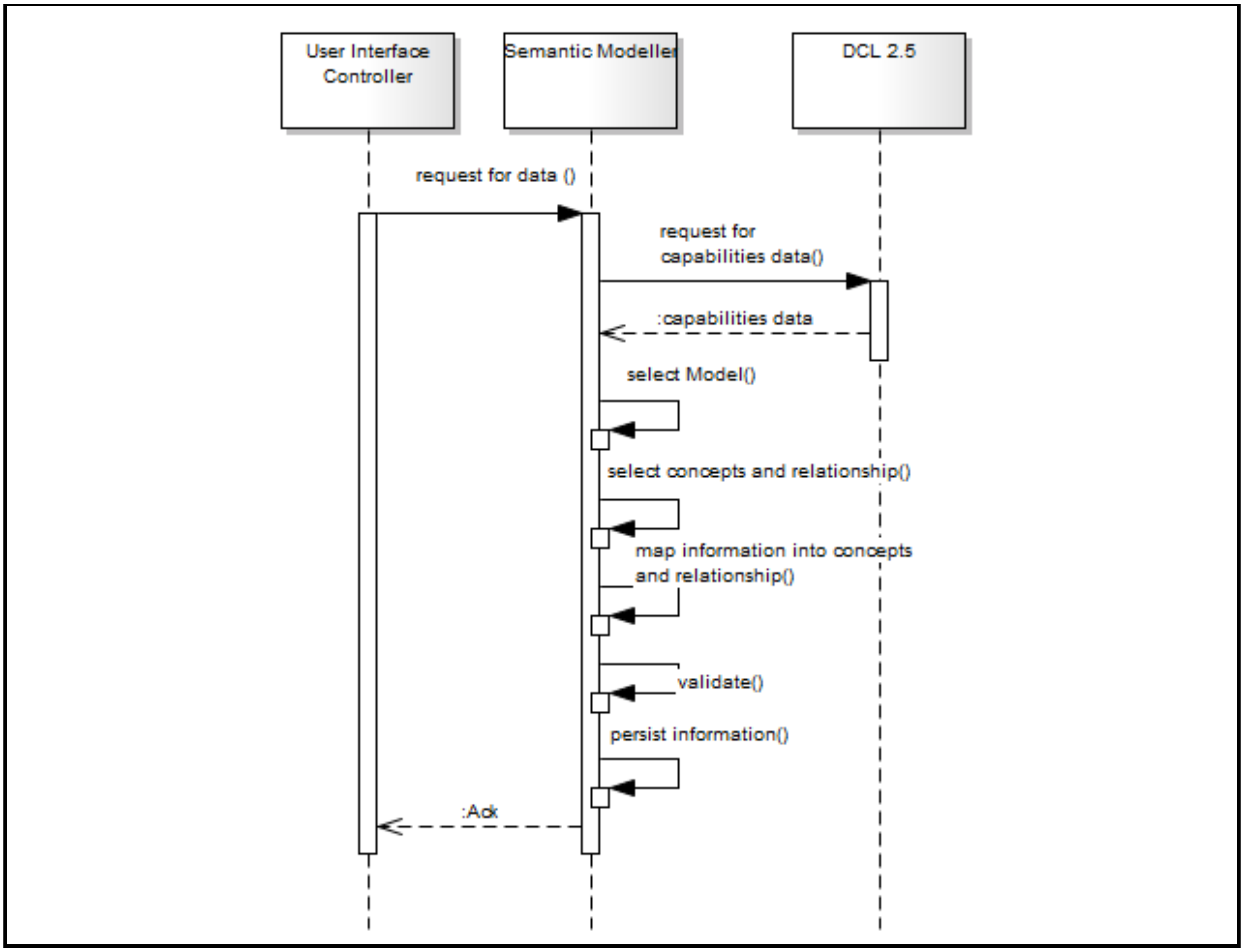
Use Case ID:	SL2.5-UCS-02		
Use Case Name:	Retrieve capabilities for user interface adaptation		
Created By:	Jamil Hussain	**Last Updated By:**	Jamil Husain
Date Created:	14 July 2015	**Last Revision Date:**	15 April 2016
Actors:	DCL 2.5		
Description:	This use case focuses on the retrieval of the capabilities for user interface		

	adaptation. The capabilities include user profile information, context information and device information. It is utilized for adaptation based on changes or observational data.
Trigger:	
Preconditions:	The DCL 2.5 provide the access to required information
Postconditions:	All required capabilities are successfully collected.
Normal Flow:	1. SL generates request for user, device, and context information collection from DCL 2.5 2. This information is utilized for the adaptation of the user interfaces 3. The adaptation is based on changes in user profile, context information or collected observational data
Alternative Flows:	N/A
Exceptions:	If there is not capabilities information then the default user interfaces will be displayed.
Includes:	N/A
Frequency of Use:	Always when the application is running [High]
Special Requirements:	N/A
Assumptions:	The capabilities information should be available with the DCL.
Notes and Issues:	N/A
Sequence Diagram	



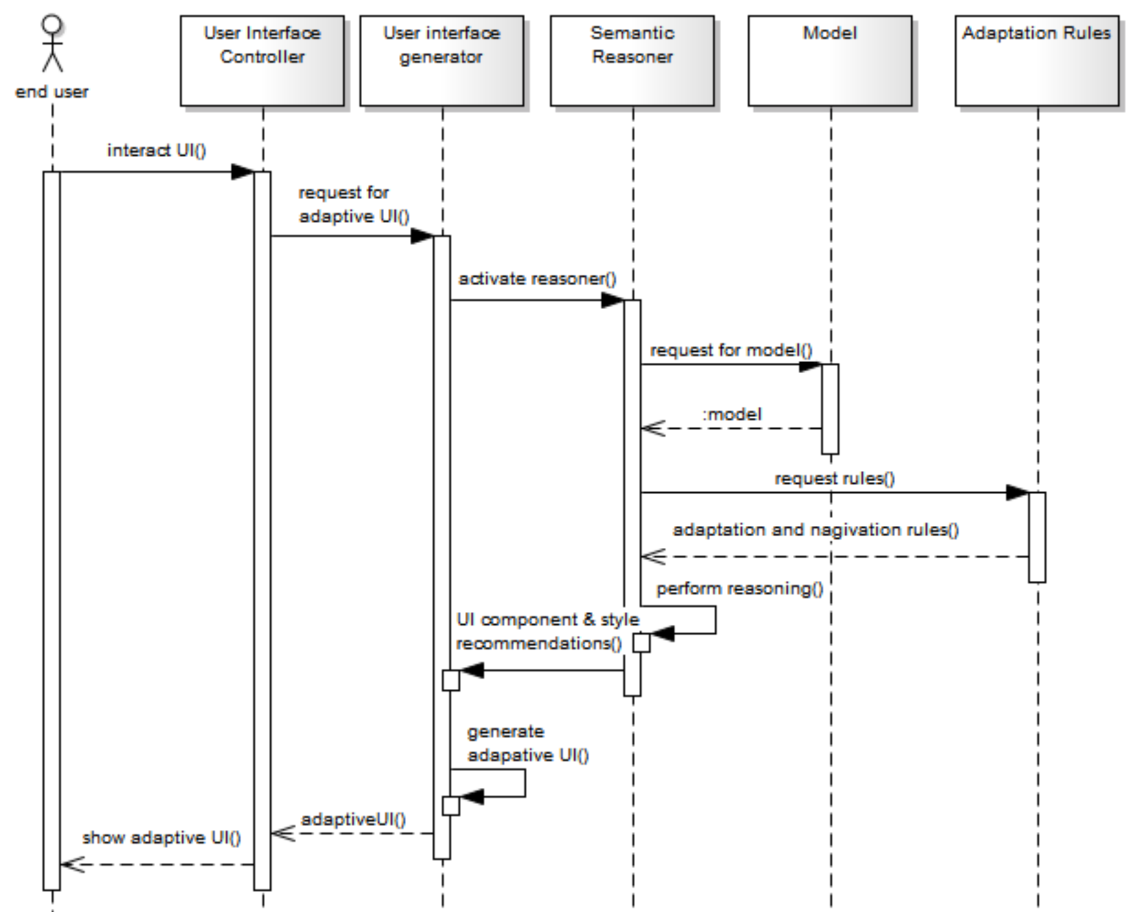
Use Case ID:	SL2.5-UCS-03		
Use Case Name:	Mapping the user capability information into model		
Created By:	Jamil Hussain	**Last Updated By:**	Jamil Husain
Date Created:	14 July 2015	**Last Revision Date:**	15 April 2016

Actors:	DCL 2.5
Description:	The collected capabilities information from DCL 2.5 shall be mapped against the hierarchical structure of the model
Trigger:	SL 2.5 initiate communication with DCL 2.5.
Pre-conditions:	4. User is a registered client of MM platform 5. Updated user profile must be available
Post-conditions:	1. User profile and environmental variables are received by UI/UX 2. All collected variables are successfully mapped and validated
Normal Flow:	1. UI/UX send request to DCL 2.5 for environmental variables (e.g., temperature, weather, time, noise, light level etc.) and user profile variables (e.g. uid, name, age, perceptual information) 2. DCL 2.5 sent back the requested variables. 3. The semantic modeller maps the data to model 4. The mapped information is persisted in model.
Alternative Flows:	N/A
Exceptions:	
Includes:	**SL2.5-UCS-01**
Frequency of Use:	Always when the application is running [High]
Assumptions:	N/A
Notes and Issues:	N/A
Sequence Diagram	



Use Case ID:	SL2.5-UCS-04		
Use Case Name:	Adapt user interface based on user profile, context and device		
Created By:	Jamil Hussain	**Last Updated By:**	Jamil Husain
Date Created:	14 July 2015	**Last Revision Date:**	15 April 2016
Actors:	DCL 2.5, end user		
Description:	The collected information of user profile, context of use and device characteristics from DCL 2.5 results in adaption of the user interface		
Trigger:	End user start interacting with user interface		
Pre-conditions:	• The user profile and context of use and device data has been collected by UI/UX Authoring tool		
Post-conditions:	• Adaptive UI rendered/generated based on collected information		

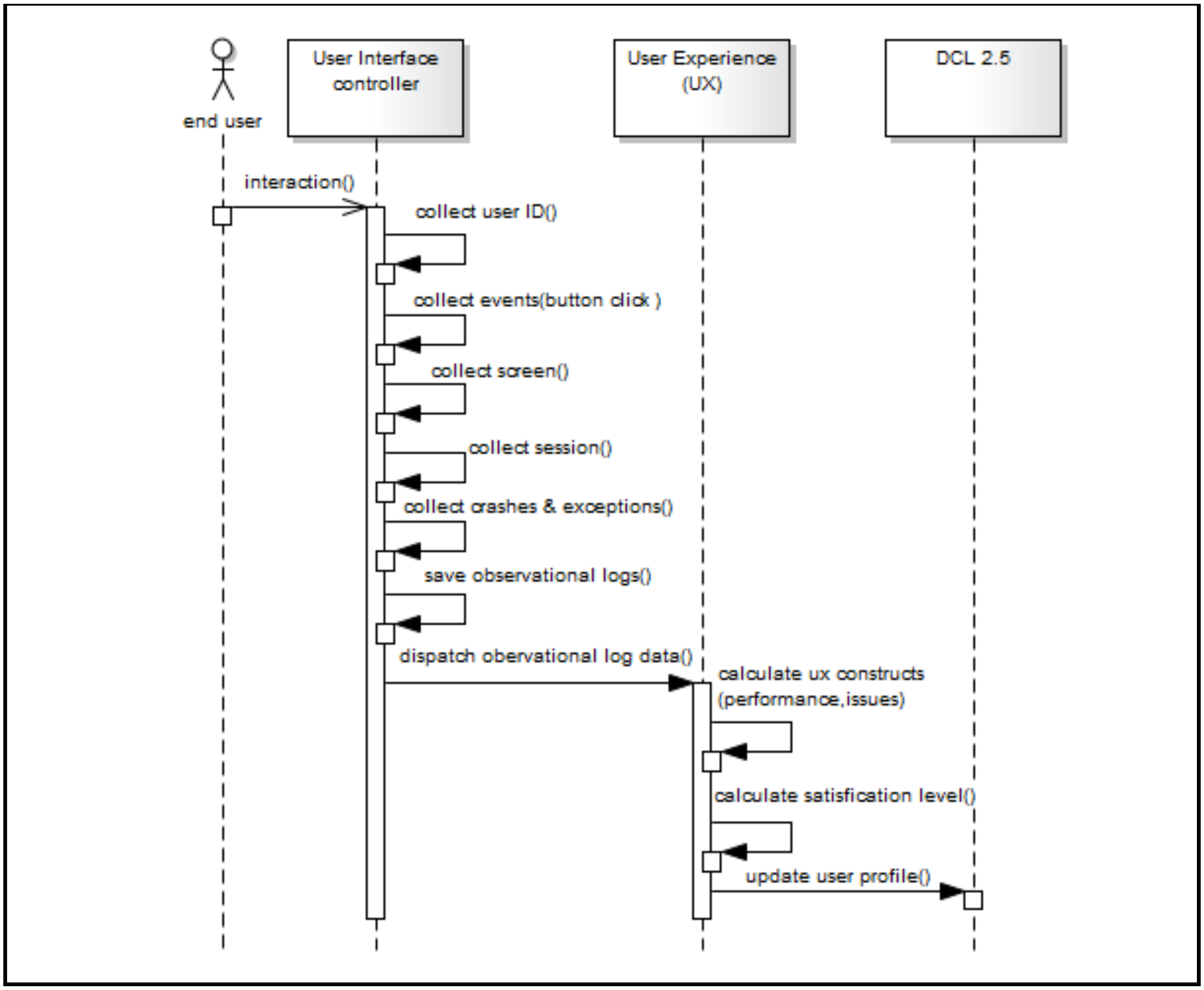
Normal Flow:	1. User request for personalize user interface to UI/UX. 2. UI/UX send request to adaptation engine for generating the personalized user interface based on user, context and device information. 3. Adaptation engine perform reasoning based on pre-defined adaption and navigations rules. 4. Reasoner recommend the user interface components and its styles to generate the UI. 5. Then personalized generated UI is displayed to end user.
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Frequency of Use:	When user interact with the system
Assumptions:	N/A
Notes and Issues:	N/A
Sequence Diagram	



Use Case ID:	SL2.5-UCS-05

Use Case Name:	Collect and analyze observational data		
Created By:	Jamil Hussain	**Last Updated By:**	Jamil Husain
Date Created:	14 July 2015	**Last Revision Date:**	15 April 2016

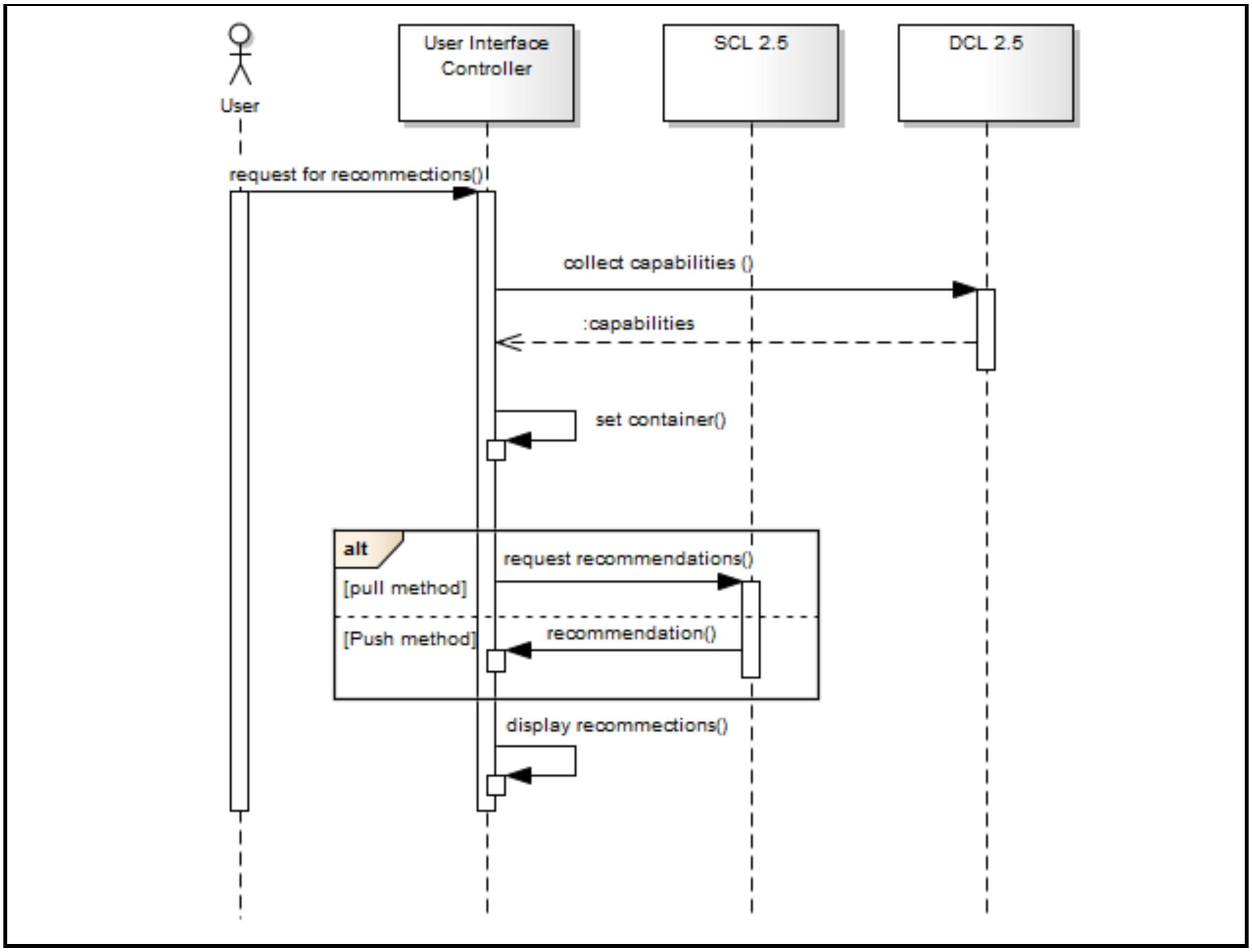
Actors:	End user, DCL 2.5
Description:	The UI/UX shall identify areas of improvement and maximize the user interaction by analyzing the user interaction with app.
Trigger:	Initiated by end user
Pre-conditions:	• Analytics tracker is already installed
Post-conditions:	• Observational data are successful collected and analyzed for user experience measurement
Normal Flow:	1. Analytics collector collect the user interaction data such as user ID, event, session, screen, crashes & exceptions, and user timings 2. The collected data is stored locally before being dispatched 3. Data is dispatched for user experience measurement from the app for every 30 minutes 4. the pragmatic quality such as usability-(e.g. performance, issues) are calculated in order to find the user experience (UX) 5. UX quality variables are sent to DCL 2.5 for storage/updating in user profile.
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Frequency of Use:	Frequent, request by SL 2.5
Special Requirements:	N/A
Assumptions:	Service contract between DCL 2.5 and SL 2.5 is defined
Notes and Issues:	N/A
Sequence Diagram	



Use Case ID:	SL2.5-UCS-08		
Use Case Name:	Acquire Recommendations for displaying to end user		
Created By:	Jamil Husain	**Last Updated By:**	Jamil Husain
Date Created:	14 July 2015	**Last Revision Date:**	15 April 2016

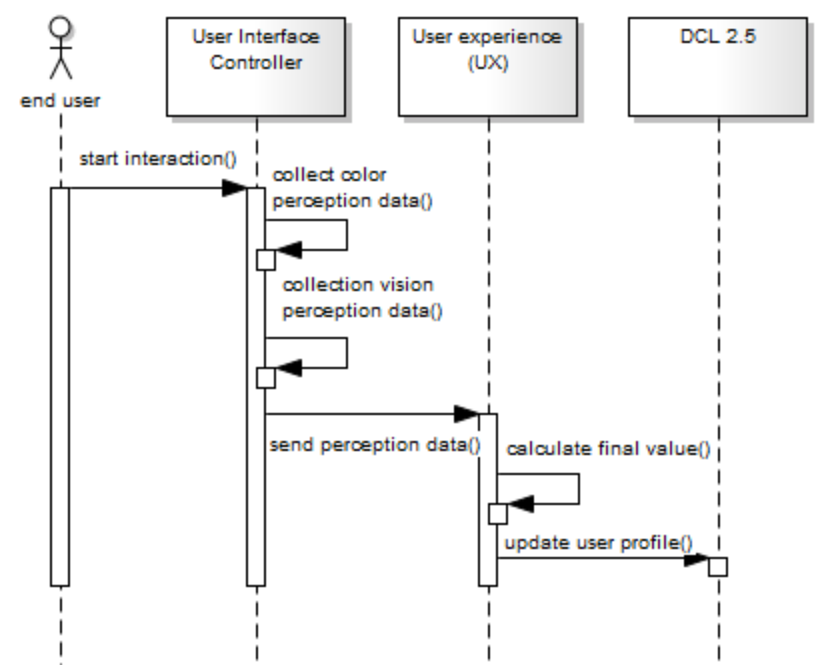
Actors:	Primary: End-user Secondary: SCL 2.5, DCL 2.5
Description:	This use case collects the recommendations generated by SCL 2.5 and displays it on the user interface for the end users. The provided recommendations are displayed according to user capabilities, context of use, and device characteristics. This information is obtained from the DCL 2.5.
Trigger:	SCL 2.5 push the recommendations to the App or end-user send request

	for recommendations
Preconditions:	End-user subscribes to particular services
Postconditions:	All recommendations are successfully displayed according to user capabilities, context, and device characteristics.
Normal Flow:	1. SCL 2.5 generate the recommendations and provide it to user interface 2. The SCL 2.5 recommendations are acquired by the SL 2.5 3. SL investigates the user capabilities, context of use, and device characteristics by obtaining from DCL 2.5 4. The recommendation are displayed in graphical user interface based on collected capabilities of user, context and device information.
Alternative Flows:	2a. In step 2. The SCL 2.5 recommendations are acquired by the SL 2.5 1. user request for recommendations (pull method) 2b. In step 2. The SCL 2.5 recommendations are acquired by the SL 2.5 1. SL 2.5 push recommendations to App based on situations
Exceptions:	N/A
Includes:	N/A
Frequency of Use:	Whenever the recommendations are generated by SCL 2.5 [Medium]
Special Requirements:	N/A
Assumptions:	The user profile data and context information should exist in the DCL 2.5
Notes and Issues:	N/A
Sequence Diagram	



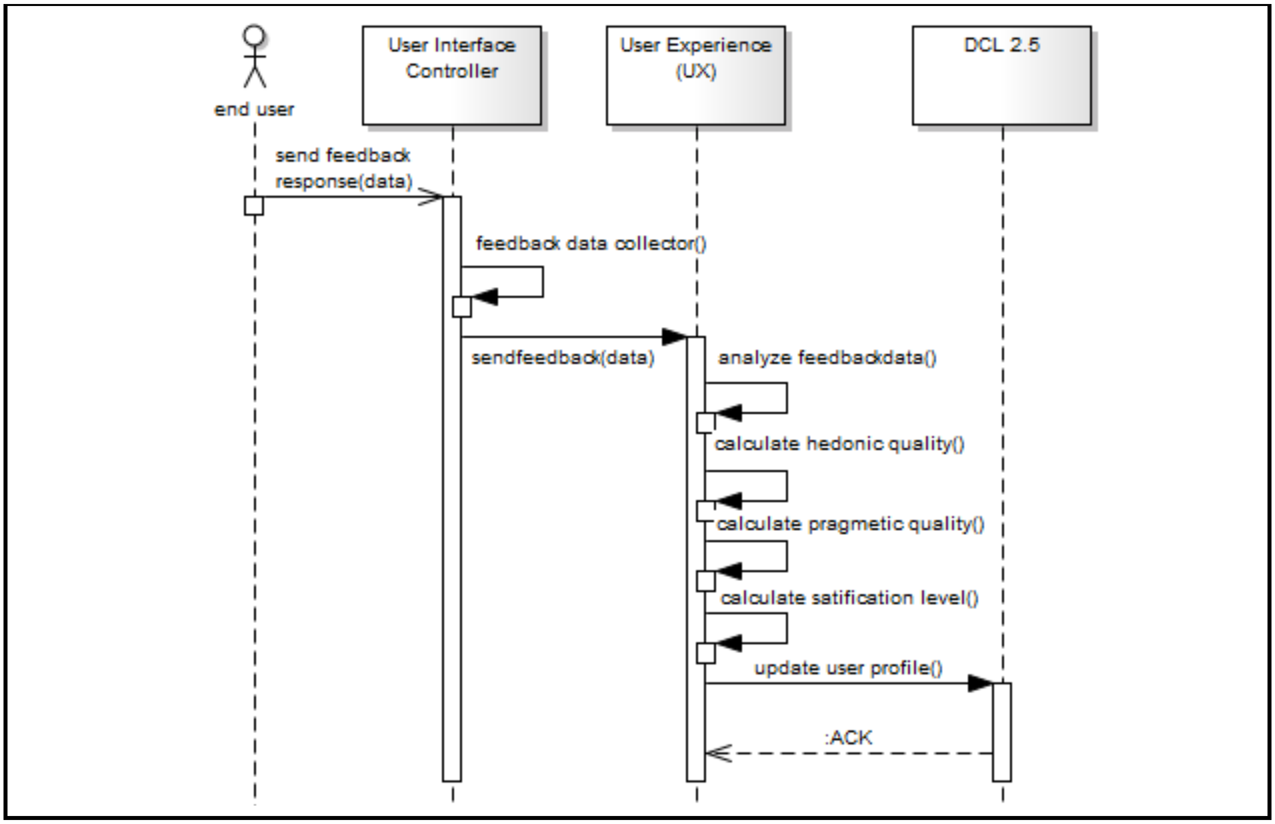
Use Case ID:	SL2.5-UCS-9		
Use Case Name:	User capabilities collection		
Created By:	Jamil Hussain	**Last Updated By:**	Jamil Hussain
Date Created:	18 March 2016	**Last Revision Date:**	15 April 2016
Actors:	End user, DCL 2.5		
Description:	The UI/UX collects the user capabilities information's by analyzing the user perception such as user visual and color perception		
Trigger:	User uses the tools for collection		
Pre-conditions:	• Perception collection tools are installed		
Post-conditions:	• User perceptual information successfully collected and update information in user profile DCL 2.5		

Normal Flow:	1. User select the tools for color and visual perception and interact with it accordingly. 2. Tools acquire its interaction data in order to find the user perceptions 3. User experience calculate its final value. 4. Final values are stored in user profile DCL 2.5.
Alternative Flows:	
Exceptions:	N/A
Includes:	N/A
Frequency of Use:	When user interact with the system
Special Requirements:	
Assumptions:	Service contract between SL 2.5 and DCL 2.5 is defined
Notes and Issues:	N/A
Sequence Diagram	



Use Case ID:	SL2.5-UCS-10		
Use Case Name:	Self-reporting user experience measurement		
Created By:	Jamil Hussain	**Last Updated By:**	Jamil Hussain
Date Created:	18 March 2016	**Last Revision Date:**	15 April 2016
Actors:	End user, DCL 2.5		

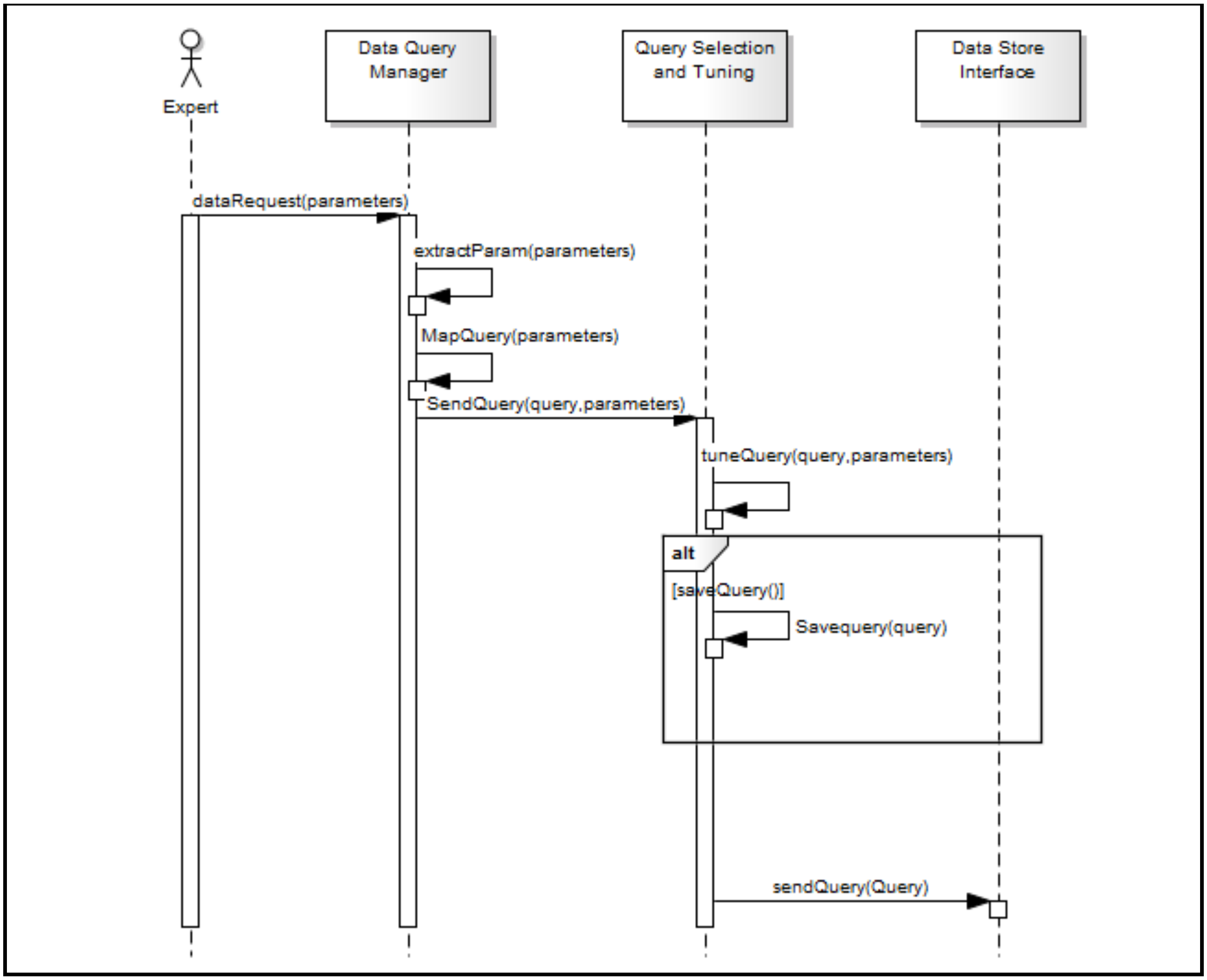
Description:	The UI/UX shall collect feedback about how users feel about the system during or after use by self-reporting method.
Trigger:	End user
Pre-conditions:	• Self-reporting questionnaire already exist
Post-conditions:	• The feedback is successfully collected
Normal Flow:	1. The end user provide feedback using the questionnaire. 2. The feedback is sent to user experience in order to evaluate the user response. 3. user experience variables such as usability, pleasure, beauty are calculated based on filled questions 4. The UI/UX update the calculated variables values in user profile by sending request to DCL 2.5
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Frequency of Use:	When user interact with the system
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A
Sequence Diagram	



Use Case ID:	SL2.5-UCS-11		
Use Case Name:	Map Request to Query		
Created By:	Shujaat Hussain	**Last Updated By:**	Shujaat Hussain
Date Created:	14 July 2015	**Last Revision Date:**	15 April 2016

Actors:	Primary: Expert
Description:	This use case focuses on mapping the expert request to the query library for data store interface.
Trigger:	The request from the expert panel for analytics
Preconditions:	A predefined query library for retrieving the big data
Postconditions:	The query is sent to the data store interface and the data is fetched.

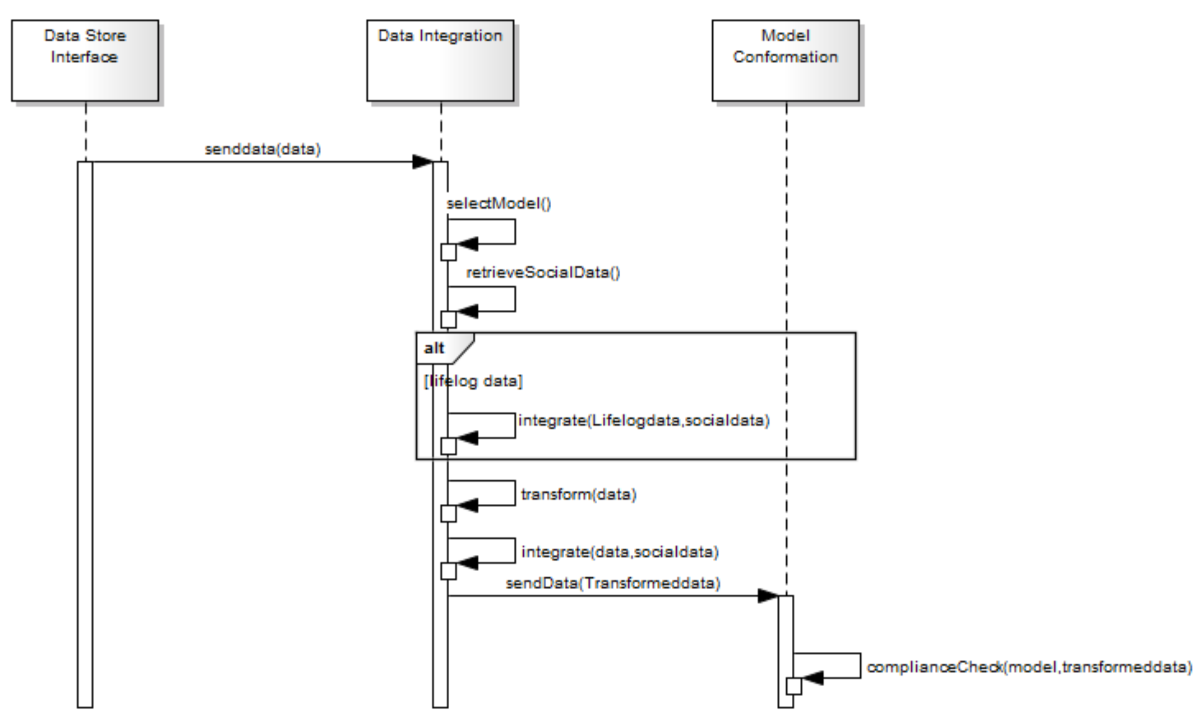
Normal Flow:	1. The expert requests the analytics for a specific context. 2. The parameters of the request is extracted and sent to query manager. 3. The query manager matches the parameters with the predefined queries in the query library. 4. The Query is selected and tuned according to the duration of the data to be extracted.
Alternative Flows:	4a. In step 4 of the normal flow, if there is a more tuning done than the query 1. The query is saved in the library for future calls.
Exceptions:	N/A
Includes:	N/A
Frequency of Use:	This use case can be used by the domain expert about 5-10 times based on the volume of data. [Low]
Special Requirements:	N/A
Assumptions:	For this use case the assumption is a query library.
Notes and Issues:	1. How many queries are there in the query library?
Sequence Diagram	



Use Case ID:	SL2.5-UCS-12		
Use Case Name:	Transform Data		
Created By:	Shujaat Hussain	**Last Updated By:**	Shujaat Hussain
Date Created:	14 July 2015	**Last Revision Date:**	15 April 2016

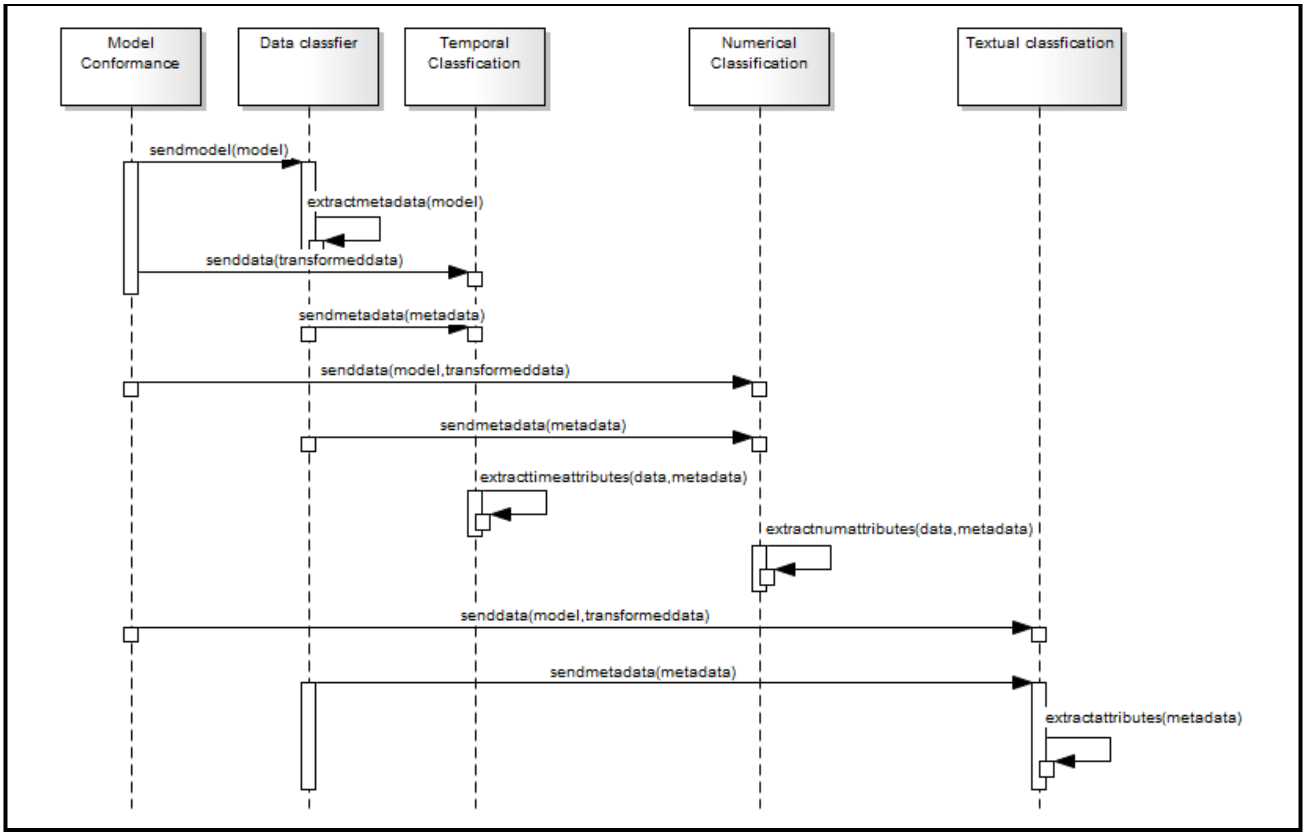
Actors:	Primary: Expert
Description:	The mapping query is transformed to specific model structure for trend analysis.
Trigger:	The data store interface initiates the data transformation process

Preconditions:	The data is sent from the data store interface.
Postconditions:	The transformed data is sent to trend analyzer.
Normal Flow:	1. The unstructured data from the big data repository is sent to the data integration component. 2. The data is transformed in an object model or a table depending on the requirements. 3. The social network data is than additionally integrated which is retrieved through a web service. 4. The transformed data is then checked for compliance with the model template.
Alternative Flows:	2a. In step 2 of the normal flow, if the data is retrieved from the life log then it is sent directly to the integration component.
Exceptions:	If the transformed data does not pass the compliance check, step 2 is started again.
Includes:	N/A
Frequency of Use:	This use case is used when the data comes from the big data and requires social data integration. [Low]
Special Requirements:	N/A
Assumptions:	
Notes and Issues:	1. How many models can the data be transformed in?
Sequence Diagram	



Use Case ID:	SL2.5-UCS-13		
Use Case Name:	Classify Data		
Created By:	Shujaat Hussain	**Last Updated By:**	Shujaat Hussain
Date Created:	14 July 2015	**Last Revision Date:**	15 April 2016

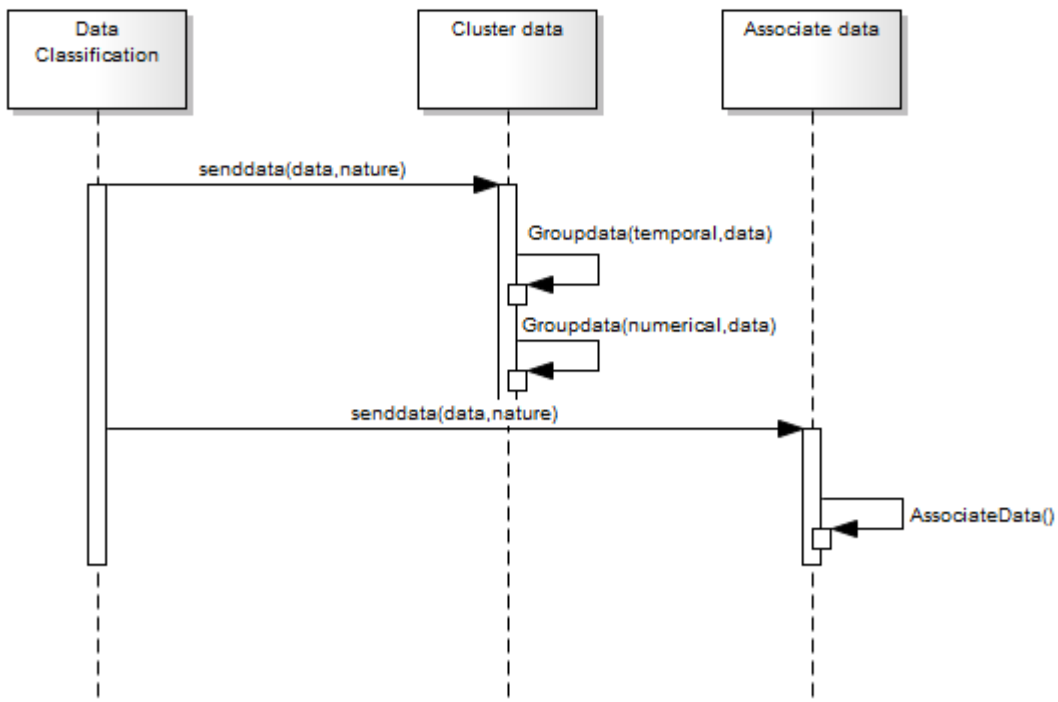
Actors:	Primary: Expert
Description:	The transformed data is further classified and clustered to identify and analyze trends.
Trigger:	The transformed data is sent for trend analysis.
Preconditions:	The data is structured into a particular model.
Postconditions:	The data is classified into temporal, numerical and textual categories
Normal Flow:	1. The model is passed for the classification. 2. Metadata is extracted from the model. 3. The data is categorized based on the extracted metadata. 4. The temporal, numerical and textual data is extracted from the transformed data.
Alternative Flows:	
Exceptions:	1. There is no temporal data to be classified. 2. There is no numerical data to be classified. 3. There is no textual data to be classified.
Includes:	
Frequency of Use:	This use case is used when the transformed data comes from model transformation module. [Low]
Special Requirements:	
Assumptions:	
Notes and Issues:	
Sequence Diagram	



Use Case ID:	SL2.5-UCS-14		
Use Case Name:	Analyze Data		
Created By:	Shujaat Hussain	**Last Updated By:**	Shujaat Hussain
Date Created:	14 July 2015	**Last Revision Date:**	15 April 2016

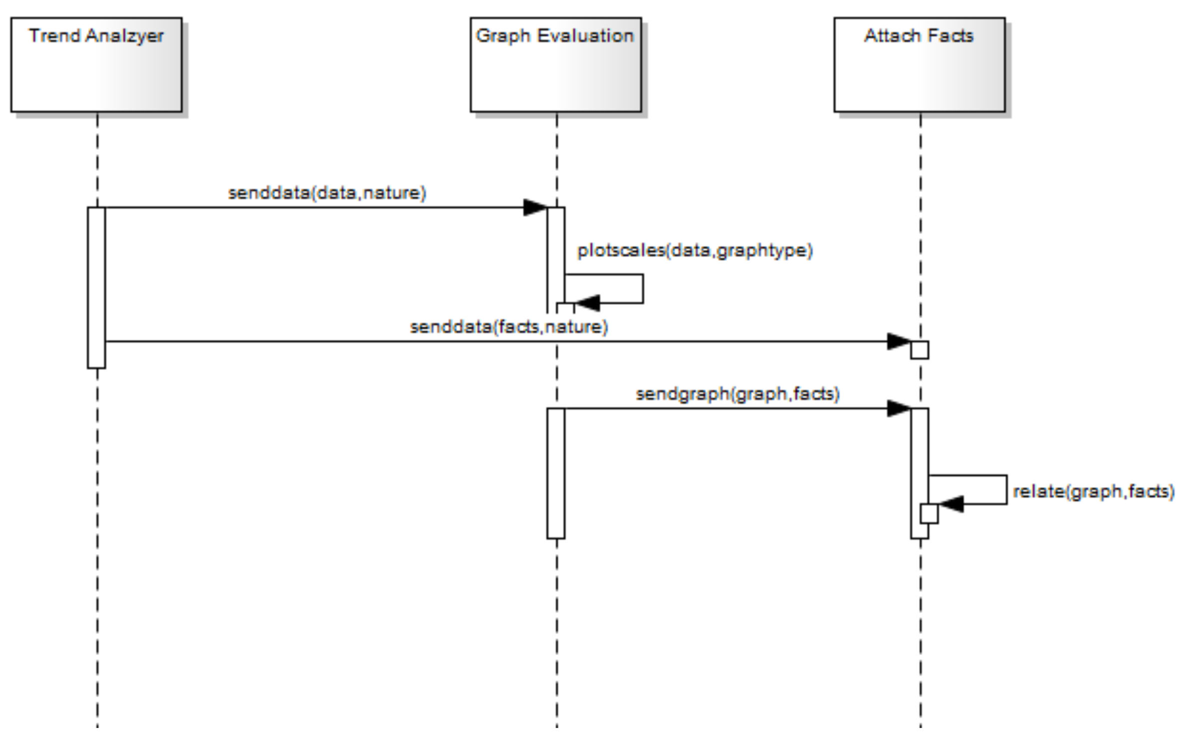
Actors:	Primary: Expert
Description:	The classified data is analyzed through association and clustering techniques for visualization and analytics.
Trigger:	The classified data is passed to association clustering for finding analytics and trends.
Preconditions:	The numerical, temporal and textual data is classified separately so that association could be applied.
Postconditions:	The association is done with the data for analytics and data to be plotted is sent for visualization.
Normal Flow:	1. The data classifier passes the data for association clustering. 2. The temporal and numerical data is analyzed for clustering. 3. The data is clustered into a group for graph plotting. 4. The textual data is associated with each other to create analytics based on the textual attribute and their corresponding facts.

Alternative Flows:	
Exceptions:	1. Clustering could not be done due to multiple outliers.
Includes:	
Frequency of Use:	This use case is used when the classified data is sent for grouping and association. [Low]
Special Requirements:	
Assumptions:	
Notes and Issues:	
Sequence Diagram	



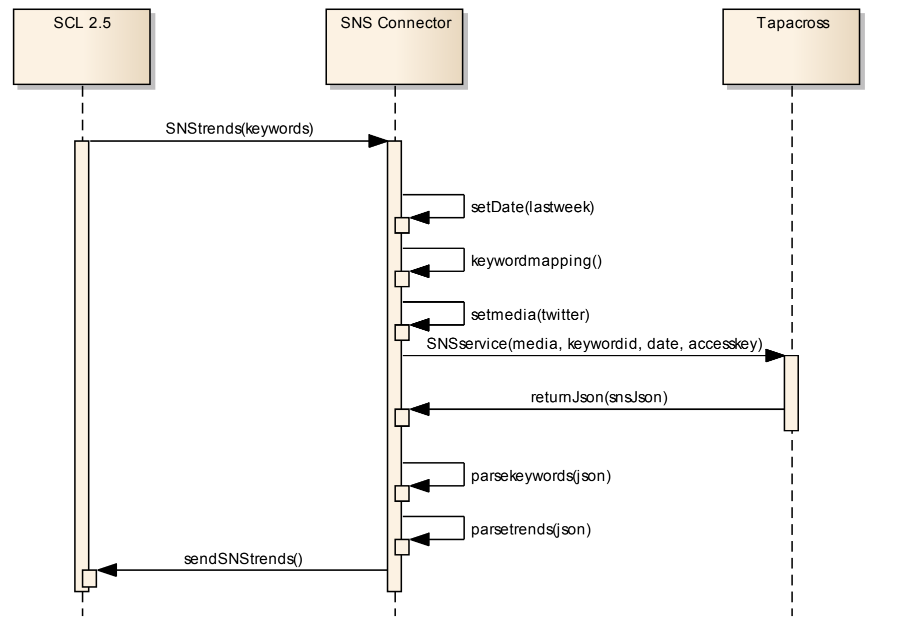
Use Case ID:	SL2.5-UCS-15		
Use Case Name:	Display Analytics		
Created By:	Shujaat Hussain	**Last Updated By:**	Shujaat Hussain
Date Created:	14 July 2015	**Last Revision Date:**	15 April 2016
Actors:	Primary: Expert		
Description:	The grouped data and relevant analytics is passed to visualization enabler so that the graphs are plotted and displayed.		
Trigger:	The trend analyzer sends the data for graph visualization and plotting.		

Preconditions:	The data is sent to visualization enabler distinguishable by their attributes and association.
Postconditions:	The analytics and relevant visualization is sent to the user interface.
Normal Flow:	1. The data is categorized according to the graph templates for visualization. 2. The scales are defined for the grouped data to be plotted on the coordinates. 3. The association text and the relevant facts about the data is also attached to the graph as analytics.
Alternative Flows:	
Exceptions:	
Includes:	
Frequency of Use:	This use case is used when the grouped data is sent for display in graph and analytics form. [Low]
Special Requirements:	
Assumptions:	
Notes and Issues:	
Sequence Diagram	



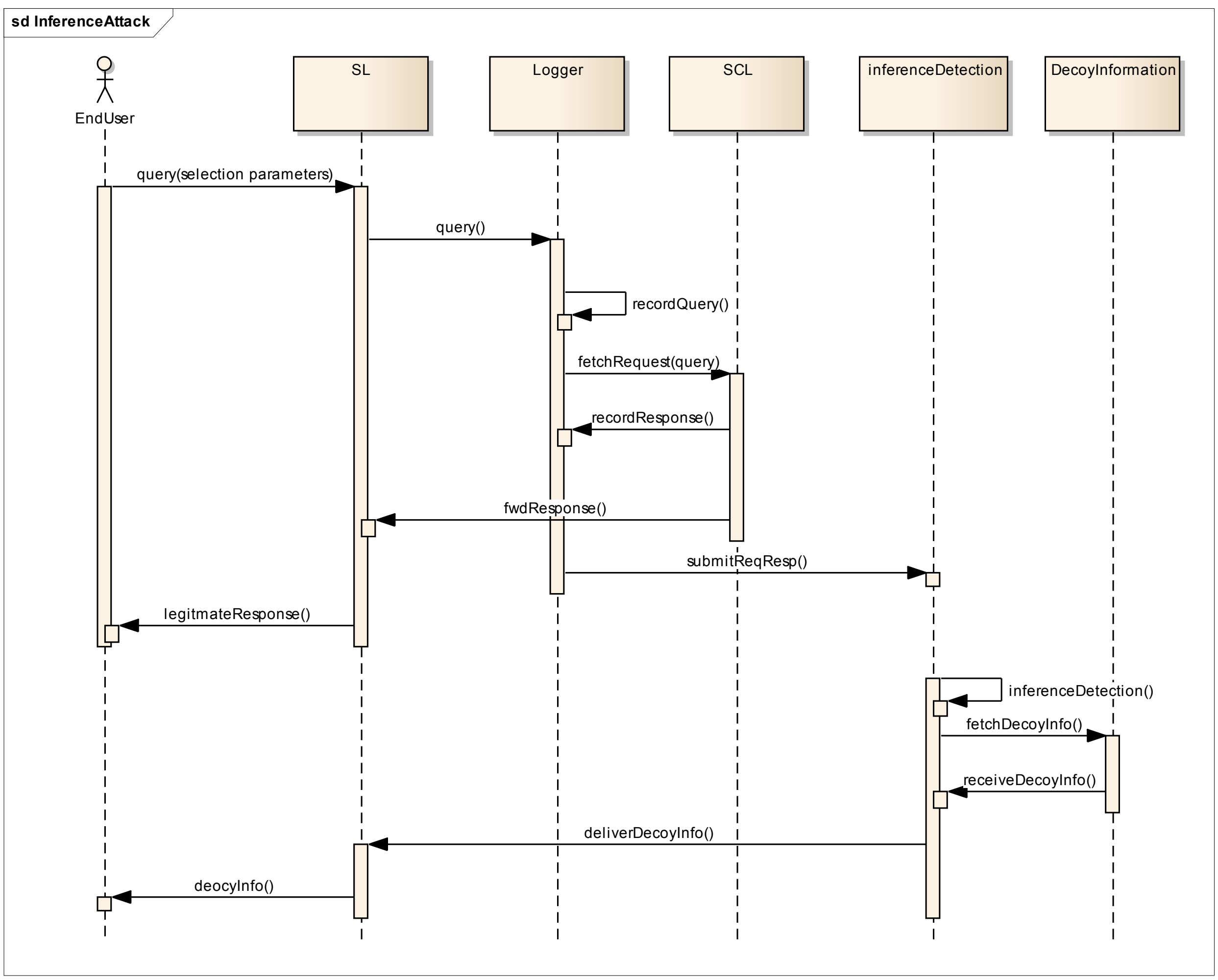
Use Case ID:	SL2.5-UCS-16		
Use Case Name:	Get Personalized SNS		
Created By:	Shujaat Hussain	**Last Updated By:**	Shujaat Hussain
Date Created:	18 March 2016	**Last Revision Date:**	15 April 2016

Actors:	Primary: Expert
Description:	The SNS trends are given to service curation layer (SCL) from tapacross.
Trigger:	SCL intiates the process by sending keywords to analytics module.
Preconditions:	SCL sends keywords to supporting layer
Postconditions:	The SNS trends of keywords are sent back to SCL
Normal Flow:	1. The descriptive analytics module receives keywords from SCL. 2. The keywords are appended in the webservice to get the trends from tapacross. 3. Tapacross sends the trends of the keywords in terms of frequency. 4. The trends and keywords are transformed into the JSON format agreed with SCL. 5. The json is forwarded to the SCL
Alternative Flows:	3a. There is no trend sent or the trends are not relevant. In that case no trends will be sent to SCL.
Exceptions:	If the transformed data does not pass the compliance check, step 4 is started again.
Includes:	N/A
Frequency of Use:	This use case is used when the recommendation requires personalized SNS. [High]
Special Requirements:	N/A
Assumptions:	
Notes and Issues:	

Sequence Diagram	

Use Case ID:	SL2.5-UCS-17		
Use Case Name:	Inference attack detection		
Created By:	Mahmood Ahmad	**Last Updated By:**	Mahmood Ahmad
Date Created:	14 July 2015	**Last Revision Date:**	23 April 2016
Actors:	Primary: End-user Secondary: SCL 2.5		
Description:	This use case is used for detecting the inference attack. A user (malicious user) in possession of valid credentials (compromised credentials) logs into the system and tries to learn the information which is unauthorized. The queries of users are logged and if found suspicious with intention to discover the identity or information of unauthorized resource, a decoy information is sent in return. A legitimate user receiving the decoy information in that case will be prompted for challenge question to release the normal reply of the application.		
Trigger:	End user		
Preconditions:	User is in possession of login credentials		
Postconditions:	Susceptibility of Inference attack is prepared with decoy information.		
Normal Flow:	1. The user interacts with the application and inquires information 2. The inquiring queries and respective response are logged through SCL 3. The inference detector using the logged information estimates the likelihood of inference attack 4. Disclosure of unauthorized information or discovery of unauthorized individual identity, the response is coupled with decoy information		

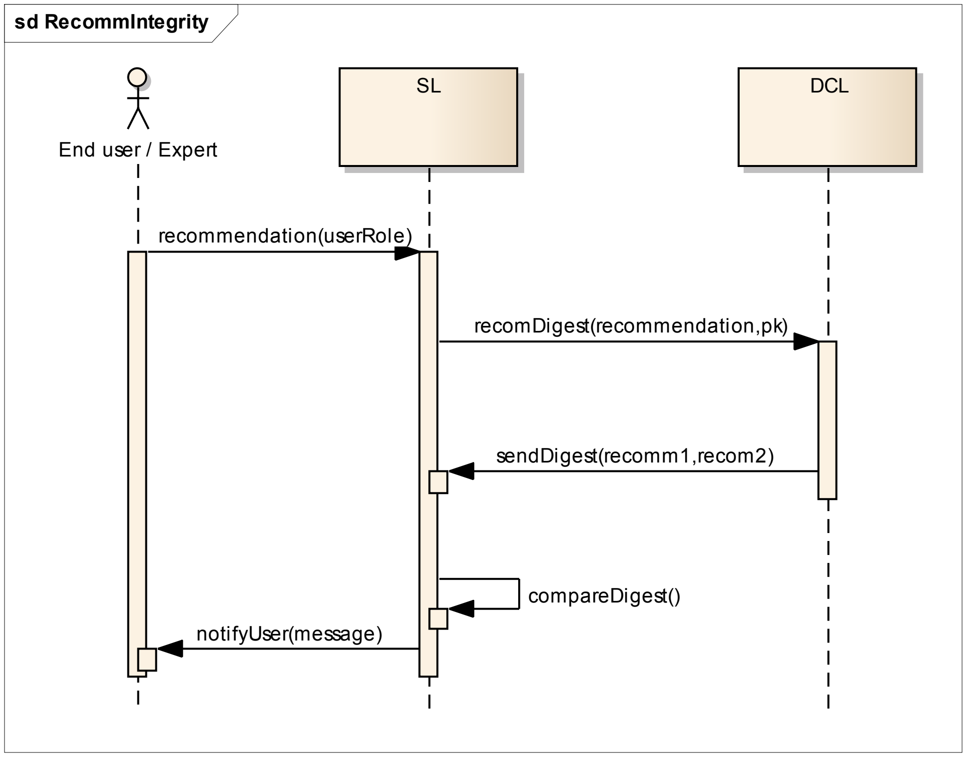
Alternative Flows:	NA
Exceptions:	N/A
Includes:	N/A
Frequency of Use:	When compromised credentials are used
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A
Sequence Diagram	



Use Case ID:	SL2.5-UCS-18		
Use Case Name:	Recommendation Integrity Check		
Created By:	Mahmood Ahmad	**Last Updated By:**	Mahmood Ahmad
Date Created:	14 July 2015	**Last Revision Date:**	23 April 2016

Actors:	Primary: End-user Secondary: DCL 2.5
Description:	The actor role which is shown as user and expert receives and generates the recommendations respectively. The message digest of these recommendations is encrypted under the public key of mining minds and is stored into the DCL. The last saved recommendations in DCL are fetched into the SL and compared with their digest values. In case of mismatch appropriate message is delivered to the user and to the expert. 1: Notification to the end user- Discard the last received recommendations and wait for the new ones 2: Notification to the expert – Reset the password and generate new recommendations
Trigger:	End user
Preconditions:	A malicious user has modified the recommendations before it is delivered to the end user
Postconditions:	Notification of recommendation regeneration to the expert
Normal Flow:	1. The expert generates the recommendation and send to the end user and also send its digest to the application 2. Assuming the malicious user (man in the middle), acquires the recommendations and modifies it 3. User receives the modified recommendations and send the recommendation digest to the application 4. SL compares the message digest and notifies the user and expert with appropriate notifications
Alternative Flows:	NA
Exceptions:	N/A
Includes:	N/A
Frequency of Use:	When recommendations are modified before delivery to the end user
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A

Sequence Diagram	



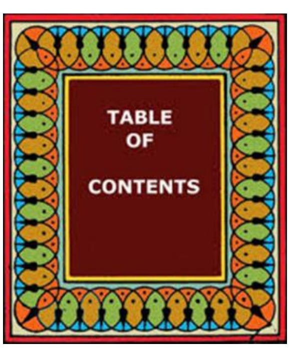
Section 4-C

Implementation Details

Mining Minds Platform Version 2.5

Principal Investigator:
Professor Sungyoung Lee

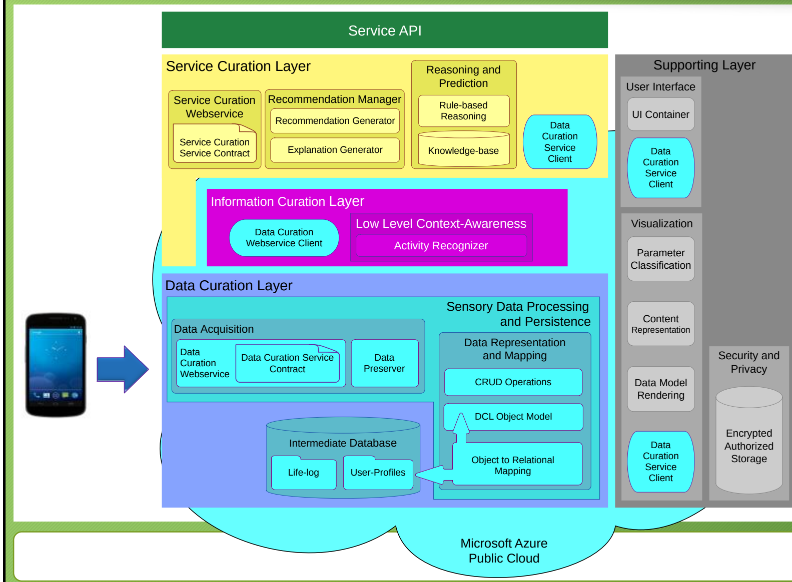
Contents



- Introduction
- Data Curation Layer
- Information Curation Layer
- Knowledge Curation Layer
- Service Curation Layer
- Supporting Layer

MM Ver. 1.0 (Release Jan 2015)

3

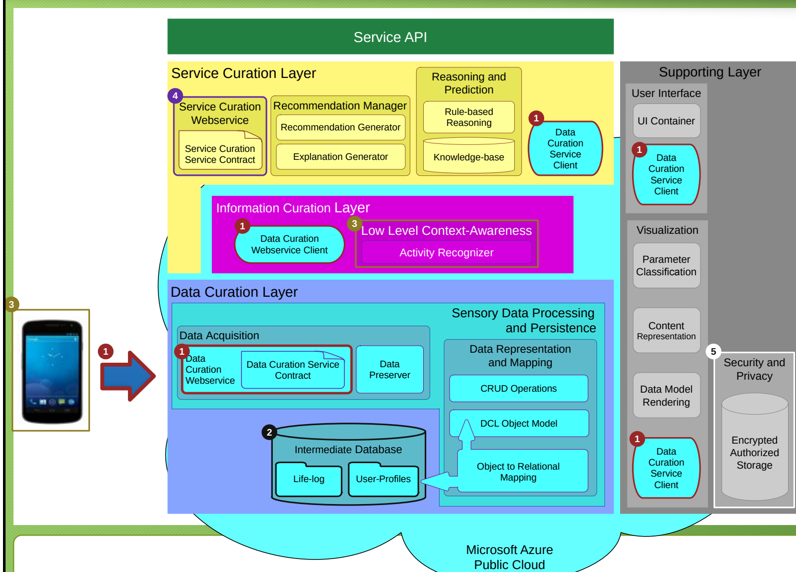


1. Input Data: Sensors (Smartphone-based Accelerometer and GPS)
2. Services : Calories-based Physical Activity recognition (5 Activities Identified)

1. Business : B2C
2. Technical Contributions:
   1. Curation
      i. DCL : Data Representation and Mapping
      ii. SCL : Rule-based Reasoning
   2. Token-based Authentication
   3. UI
   4. Admin View
   5. Communication
      i. Request-based
      ii. SOAP Web services

MM Ver. 1.0 (Release Jan 2015)

4

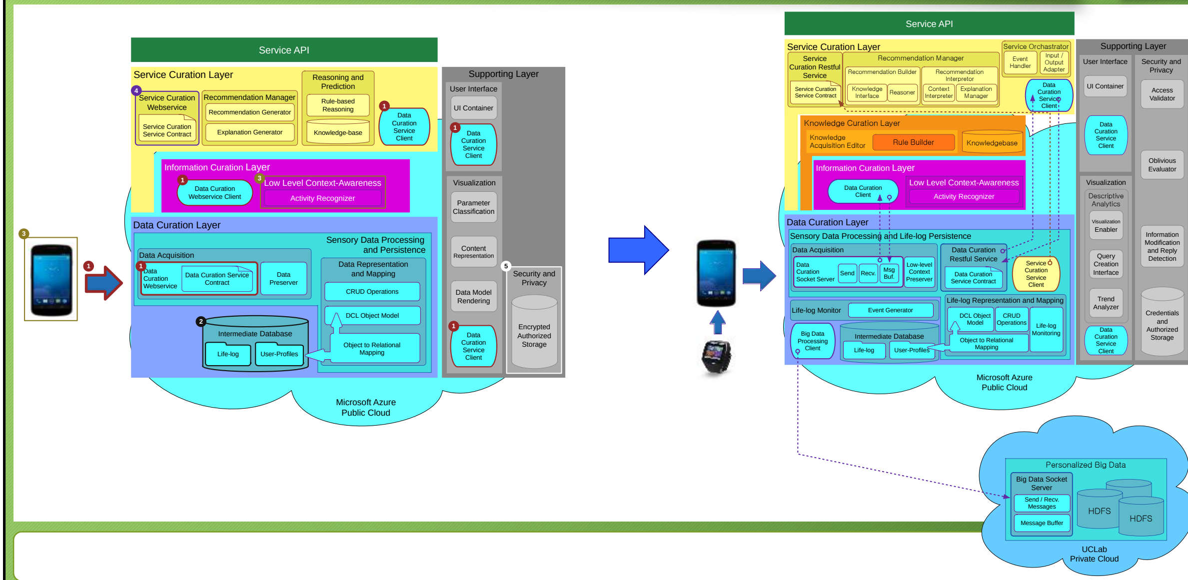


Technological Limitations

1. Poor Communication Performance due to SOAP-based Web services
2. Stress on RDBMS due to excess of data storage and persistence
3. Primitive amount of Activities were identified (5 only)
4. Request-based Recommendation generation
5. Message integrity can be compromised during communication

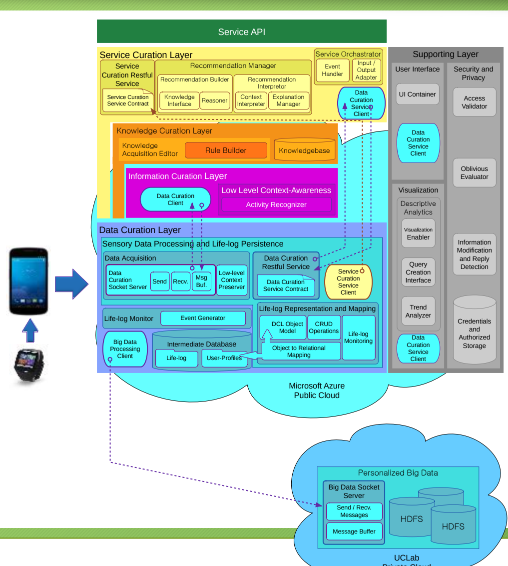
MM Ver. 1.0 (Release Jan 2015) to Ver. 1.5

5



MM Ver. 1.5 (Release Apr 2015)

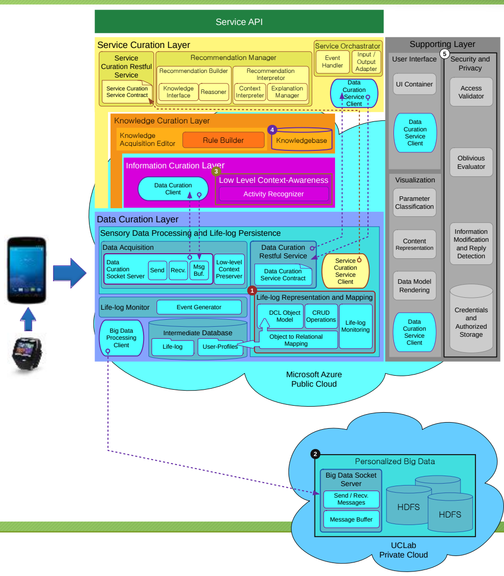
6



1. Input Data : Sensors (Smartphone-based Accelerometer & GPS, and Wearable device)
2. Services :
   1. Calories-based Physical Activity recognition 3 (15 Activities Identified)
   2. Educational Facts
   3. Healthy Habits Induction
3. Business : B2C
4. Technical Contributions:
   1. Curation
      i. DCL : Big data and Life-log Representation & Mapping 2
      ii. KCL : Rule-authoring
      iii. SCL : Generalized and Personalized Recommendations
   2. Security and Privacy
      i. Access model for domain experts
      ii. Data integrity check during communication 5
   3. UI
      - Expert View
   4. Analytics
   5. Communication
      i. Real-time socket based communication 1
      ii. Restful services with cloud push model 4

MM Ver. 1.5 (Release Apr 2015)

7

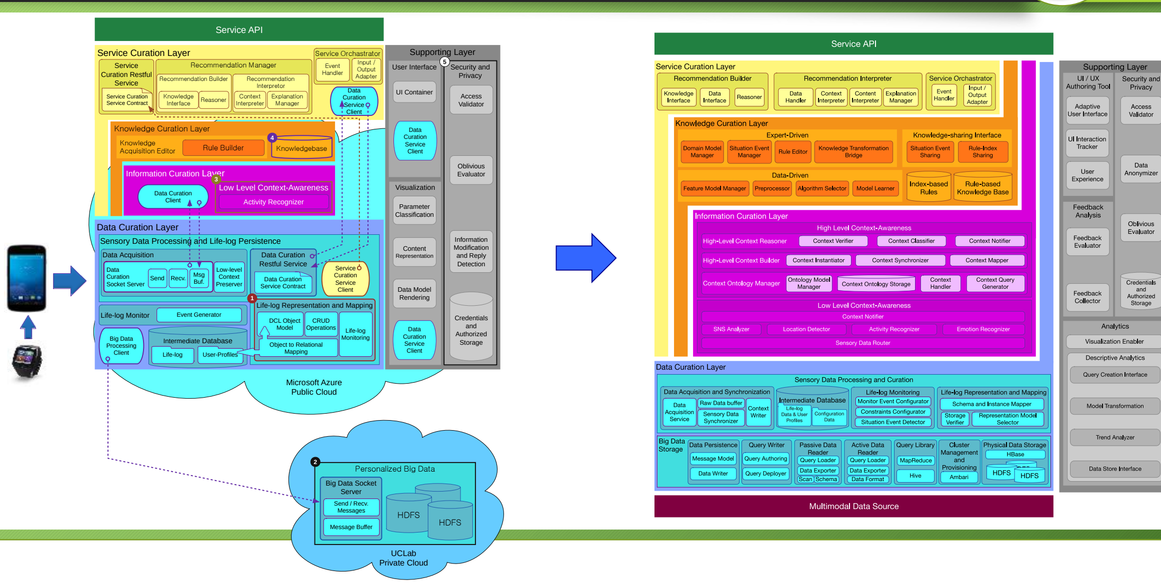


Limitations

1. Object Oriented and Relational life-log models
2. Big data is only used for persistence
3. Lack of High-level context determination
4. Only static rules as knowledge
5. Anonymized Access

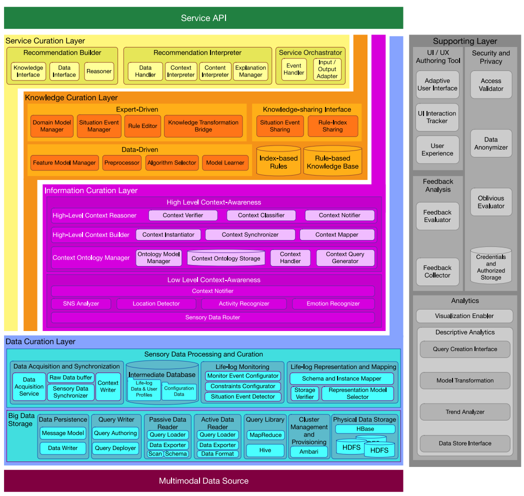
MM Ver. 1.5 (Release Apr 2015) to Ver. 2.0

8



MM Ver. 2.0 (Release Dec 2015)

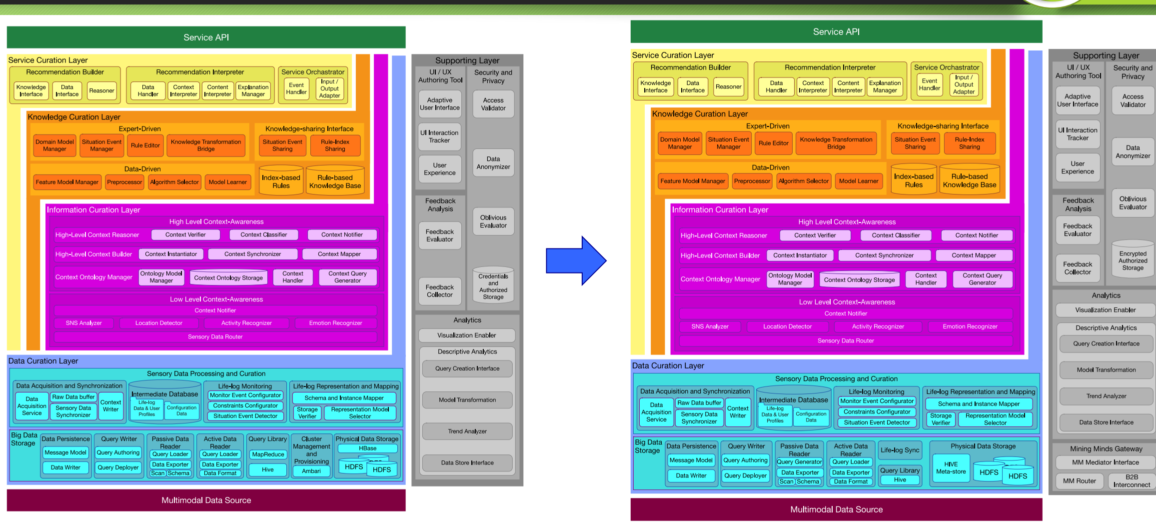
9



1. Input Data : Sensors (Smartphone-based Accelerometer & GPS, Wearable device) + Kinect-based Video + SNS
2. Services :
   1. Physical Activity recognition
   2. Educational Facts
   3. Healthy Habits Induction
3. Business : B2B, B2C
4. Technical Contributions:
   1. Curation
      - i. DCL : Life-log Monitoring 1
      - ii. ICL : Context and Fusioning 3
      - iii. KCL : Executable Knowledge, Wellness model 4
      - iv. SCL : Service Orchestration, Recommendation Builder
   2. Cloud Synchronization and Intermediate Data Generation 2
   3. Descriptive Analytics
   4. User Experience (UX)

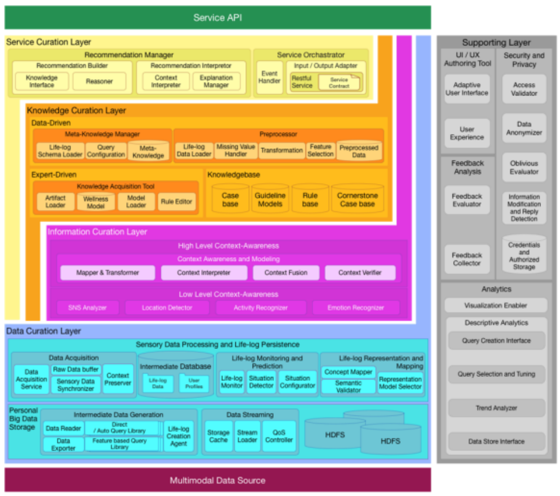
MM Ver. 2.0 (Dec 2015) to Ver. 2.5 (April 2016)

10



MM Ver. 2.5 (Release Apr 2016)

11



1. Input Data : Sensors (Smartphone-based Accelerometer & GPS, Wearable device) + Kinect-based Video + SNS
2. Services :
   1. Physical Activity recognition
   2. Educational Facts
   3. Healthy Habits Induction
   4. Nutrition-based Execution scenarios
3. Business : B2B, B2C
4. Technical Contributions:
   1. Curation
      i. DCL : Life-log Monitoring
      ii. ICL : Context and Fusioning
      iii. KCL : Executable Knowledge, Wellness model
      iv. SCL : Service Orchestration, Recommendation Builder
   2. Cloud Synchronization and Intermediate Data Generation
   3. Descriptive Analytics
   4. User Experience (UX)

Data Curation Layer (DCL 2.0)

경희대학교 KYUNG HEE UNIVERSITY KITECH 한국생산기술연구원 POSTECH 포항공과대학교 UTAS 녹십자헬스케어 TAPACROSS U2 system User & Usability system

DCL 2.5 Responsibilities

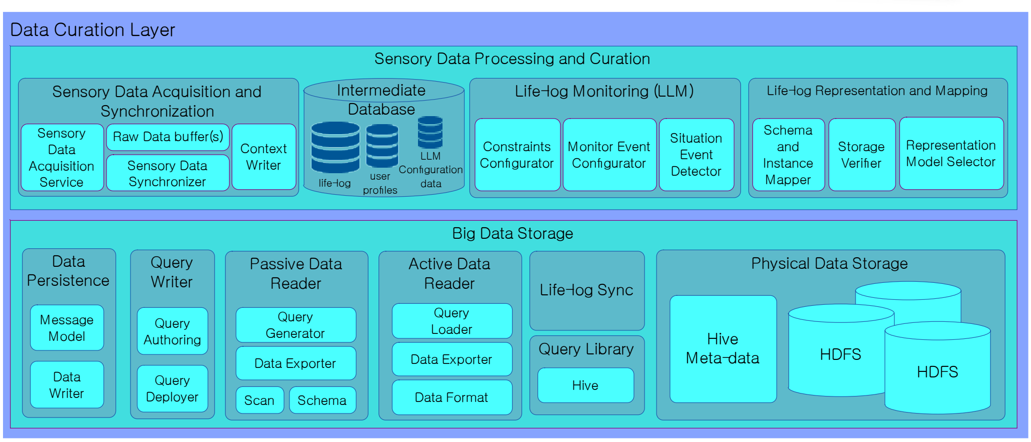
13

1. Acquisition of raw sensory, environmental variables and depth video camera data in real time (3 sec)
2. Synchronization of heterogeneous data from multiple data sources for Context determination
3. Curation of Context data to user life-log
4. Recording of user daily activities as user life-log with CRUD Operations
5. Curation of Context, raw sensory, environmental variables and depth video camera data in a large-scale non-volatile persistence (Big Data) with CRUD operations
6. Monitoring of Life-log data for static and dynamic situations
7. High Performance inter-layer and intra-layer communication

경희대학교 KYUNG HEE UNIVERSITY

DCL V2.5 Architecture

14

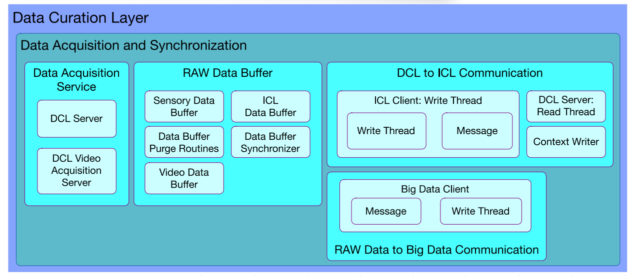


경희대학교 KYUNG HEE UNIVERSITY

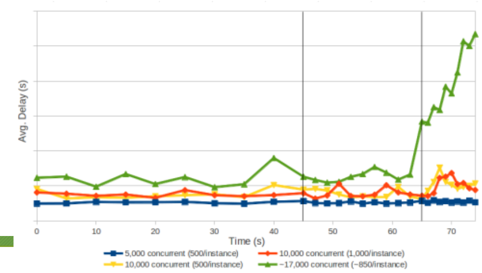
1. Data Acquisition and Synchronization Overview

15

- Role
  - Acquisition of raw sensory and video data at real time from data sources
  - Synchronization of heterogeneous data
  - Real-time communication between DCL and DCL-ICL components



- How is it different from Ver. 1.5 & Ver. 2.0 ?
  - More scalable and Concurrent with state-of-the-art Non-blocking Communication IO
  - Substantially less communication overhead with improved resource consumption
  - Data Synchronization and Buffer for heterogeneous data streams (audio, video, activities data, environmental variables)

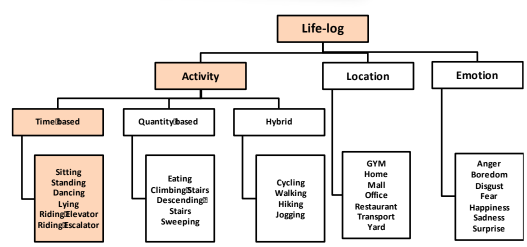


경희대학교 KYUNG HEE UNIVERSITY

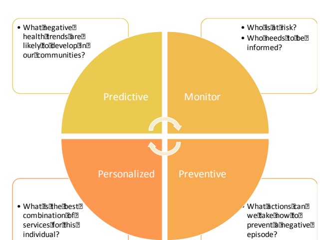
2. Life-log Monitoring (LLM) Overview

16

- Role
  - Monitoring of user life-log for situations to respond
  - Hosting and execution of static situations
  - Incorporation and execution of dynamic situations



- How is it different from Ver. 1.5 & Ver. 2.0 ?
  - Agile Static Situation: Static situations are not hard-coded, they are represented, hosted and executed
  - Dynamic Situations can now be incorporated in the LLM Execution
  - Nutrition-based situations can now be monitored

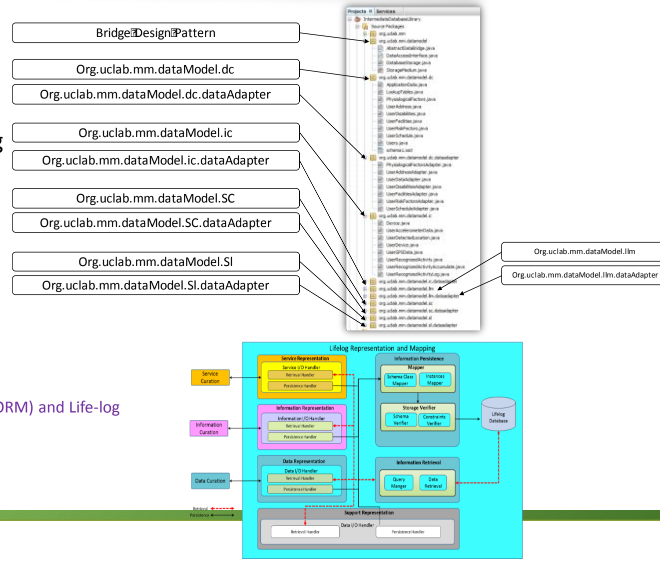


경희대학교 KYUNG HEE UNIVERSITY

3. Life-log Representation and Mapping Overview

17

- Role
  - Curation and persistence of user context in the form of user daily life routines as life-log
  - CRUD operations to user life-log data via Restful service



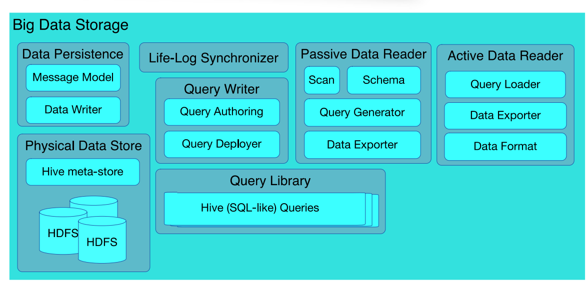
- How is it different from Ver. 1.5 & Ver. 2.0 ?
  - **"Models and constructs from Ver. 1.5 are re-used"**
  - More objects and properties at Object-to-Relational Mapping (ORM) and Life-log database (Data Level)
  - Data Curation Restful Service handles larger set of queries

경희대학교 KYUNG HEE UNIVERSITY

4. Big Data Storage

18

- Role
  - Non-Volatile storage of data from heterogeneous sources with CRUDS operations.
  - Allow active and passive data read operations



- How is it different from Ver. 1.5 & Ver. 2.0 ?
  - Read access to Big data is now enabled
  - Life-log data can now be synchronized with big data storage

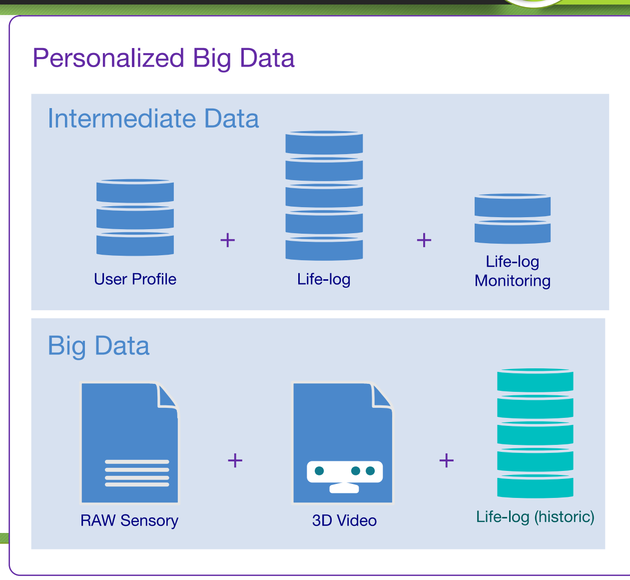
경희대학교 KYUNG HEE UNIVERSITY

Distributed Databases

M. Bilal Amin
&
Taqdir Ali

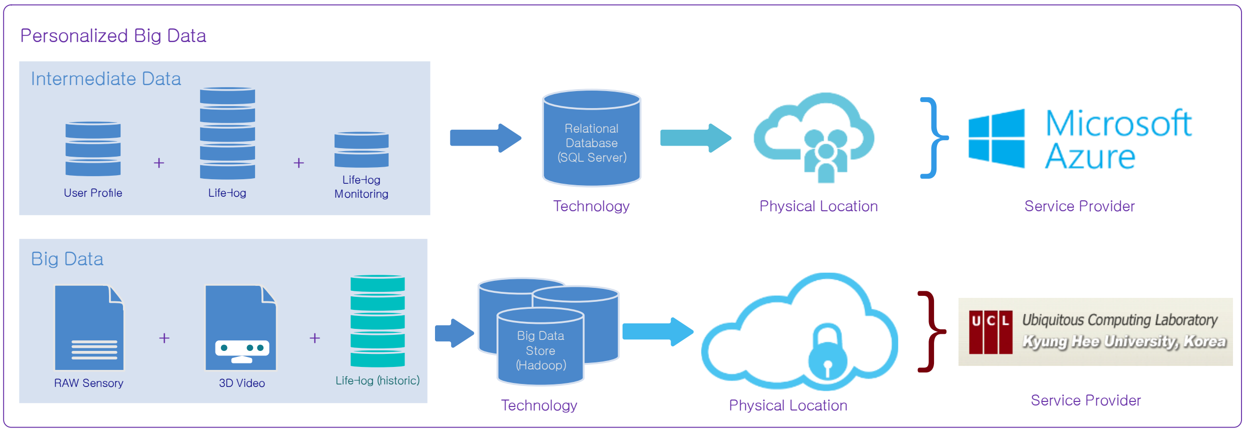
Mining Minds Distributed Database Terminologies

- Terminologies
  - Life-log Data
  - Intermediate Data
  - Raw Sensory Data
  - Big Data
  - Personalized Big Data



Mining Minds Personalized Big Data Deployment

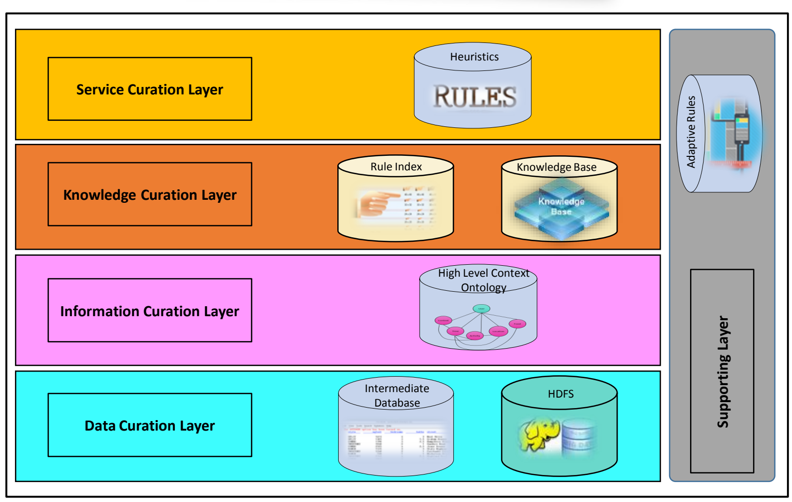
21



Mining Minds Distributed Database stacks

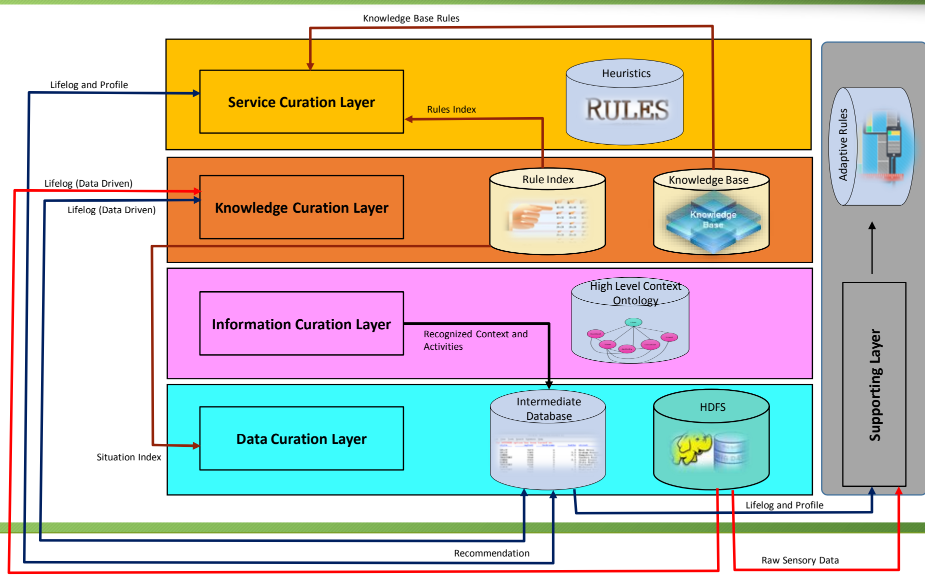
22

- Each Layer defines its own abstraction on data
- Native abstraction is fully encapsulated
- Abstractions are shared if necessary via web services based on pre-defined service contracts
- Abstraction on data is irrespective of core functionalities of other layers



Distributed Database Stacks and Layer Communication

23



Distributed Databases

24

- Pros
  - Data abstractions reduces the overall coupling between the Layers
  - Each layer is responsible for its own persistence encouraging encapsulation
  - Evolution of context ontology is independent of Life-log data
  - Reduced Communication between ICL and DCL
- Cons
  - Each layer manages its own data store
  - High Level Context:ICL and Life-log Data:DCL may have replication

Data Acquisition and Communication

M. Bilal Amin

경희대학교 KYUNG HEE UNIVERSITY · KITECH · POSTECH · UTAS · 녹십자헬스케어 · TAPACROSS · U2 system

Introduction

- Acquisition of heterogeneous data from Multi-modal data sources in *Real-time* is a must for data curation layer
- This acquisition of data must be *dynamic, parallel* and of *high performance* to support the influx of multimodal data at real-time
- For reliable acquisition of data at real-time, *optimal resource utilization* at the layer needs to be implemented such that the *performance lag is bare minimum* with low packet loss

경희대학교 KYUNG HEE UNIVERSITY

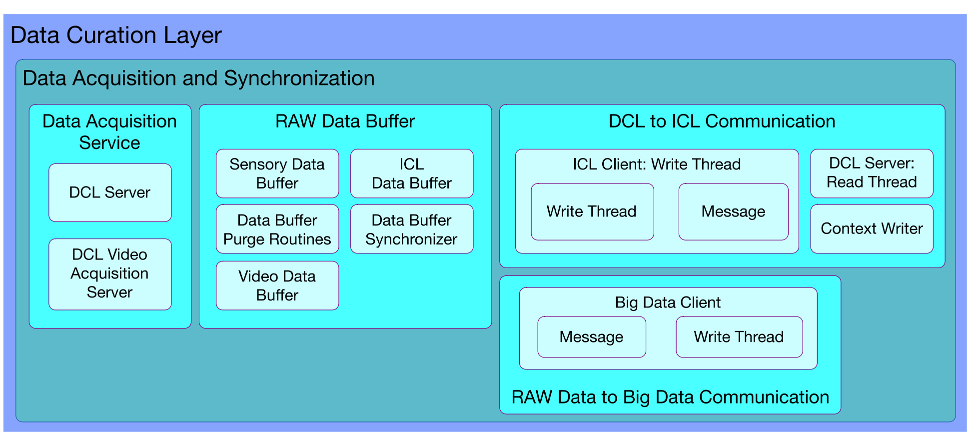
Motivations

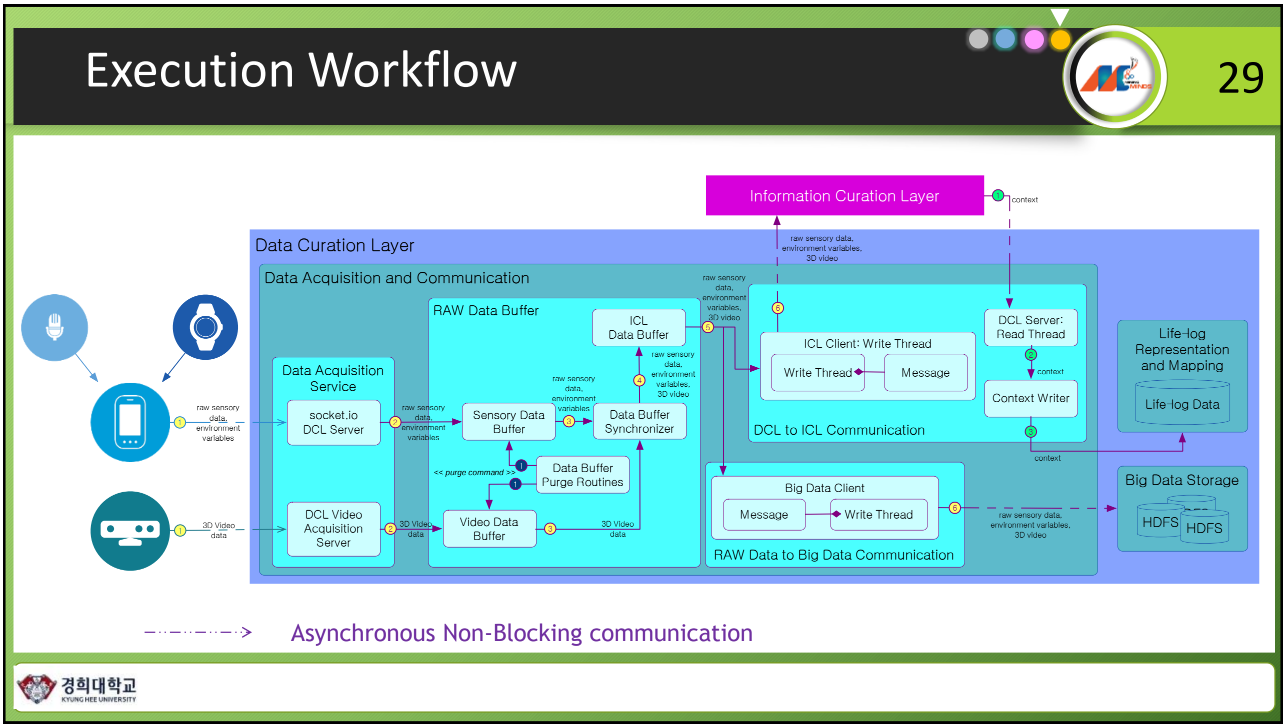
27

- Sensory data is generated after every 3 seconds from data sources
  - *Real-time data acquisition*
- MMV 2.5 is using data from multiple data sources for context determination
  - *Data needs to be synchronized prior to context determination*
- Context determination is non-real-time process
  - *Regulation and pipelining of ICL data preparation*
- DCL 2.5 is a distributed layer with parallel execution
  - *Optimal resource utilization with no communication bottlenecks*

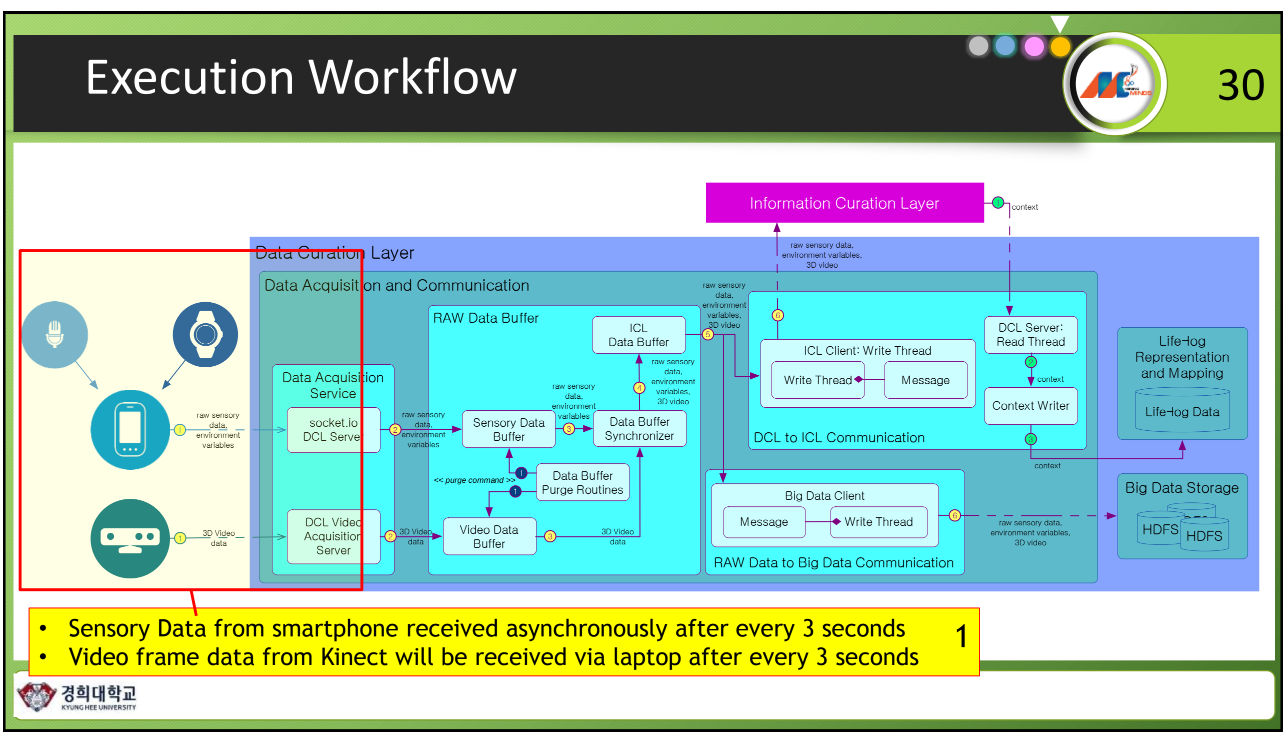
High-level Architecture

28





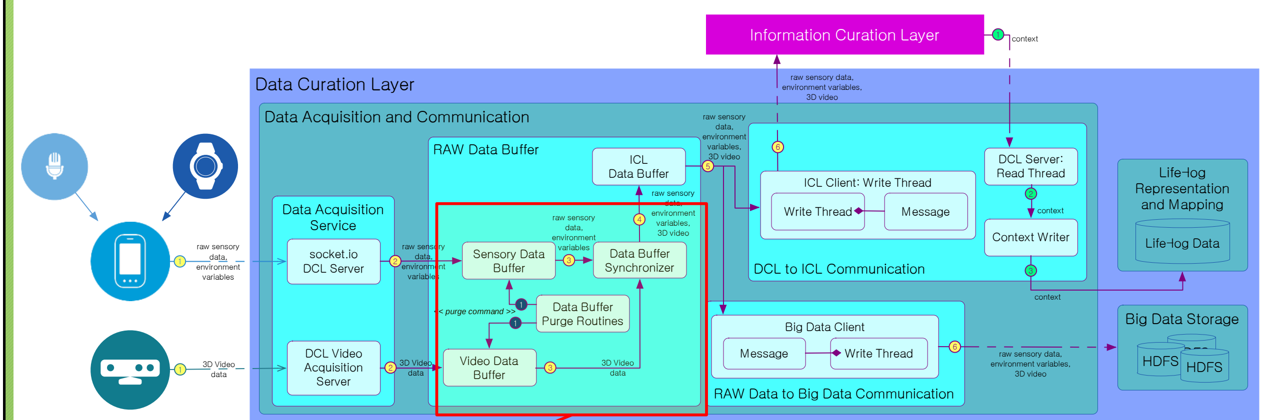
Execution Workflow
29
Information Curation Layer
context
Data Curation Layer
raw sensory data, environment variables, 3D video
Data Acquisition and Communication
RAW Data Buffer
ICL Data Buffer
Data Acquisition Service
socket.io DCL Server
Sensory Data Buffer
Data Buffer Synchronizer
Data Buffer Purge Routines
<< purge command >>
DCL Video Acquisition Server
Video Data Buffer
3D Video data
ICL Client: Write Thread
Write Thread
Message
DCL Server: Read Thread
Context Writer
DCL to ICL Communication
Big Data Client
Message
Write Thread
RAW Data to Big Data Communication
Life-log Representation and Mapping
Life-log Data
Big Data Storage
Asynchronous Non-Blocking communication
경희대학교



Execution Workflow
30
Information Curation Layer
context
Data Curation Layer
raw sensory data, environment variables, 3D video
Data Acquisition and Communication
RAW Data Buffer
ICL Data Buffer
Data Acquisition Service
socket.io DCL Server
Sensory Data Buffer
Data Buffer Synchronizer
Data Buffer Purge Routines
<< purge command >>
DCL Video Acquisition Server
Video Data Buffer
3D Video data
ICL Client: Write Thread
Write Thread
Message
DCL Server: Read Thread
Context Writer
DCL to ICL Communication
Big Data Client
Message
Write Thread
RAW Data to Big Data Communication
Life-log Representation and Mapping
Life-log Data
Big Data Storage
• Sensory Data from smartphone received asynchronously after every 3 seconds
• Video frame data from Kinect will be received via laptop after every 3 seconds
1
경희대학교

Execution Workflow

31



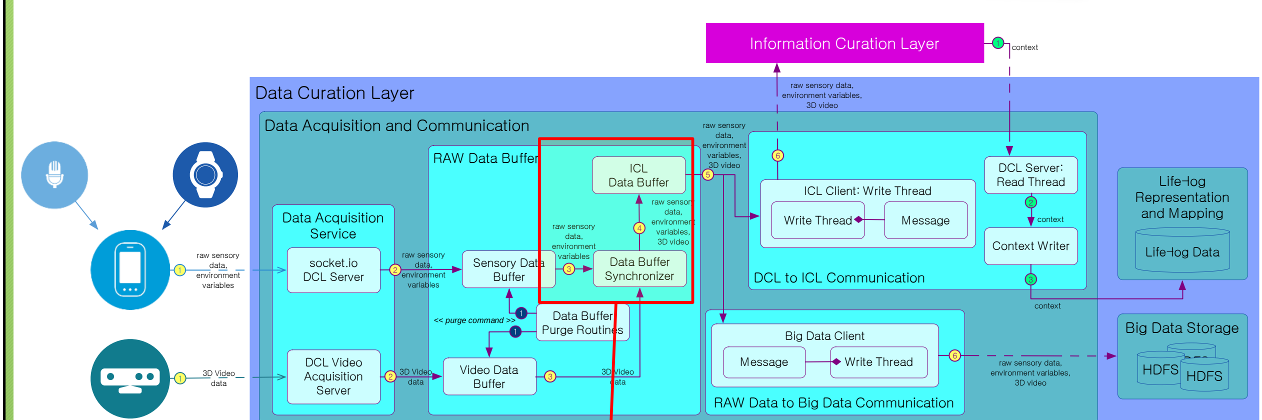
- Sensory data buffer for smartphone-based sensory data
- Video data buffer for for Kinect-based video frames data
- Buffers are synchronized by time-stamp and user id

2

경희대학교 KYUNG HEE UNIVERSITY

Execution Workflow

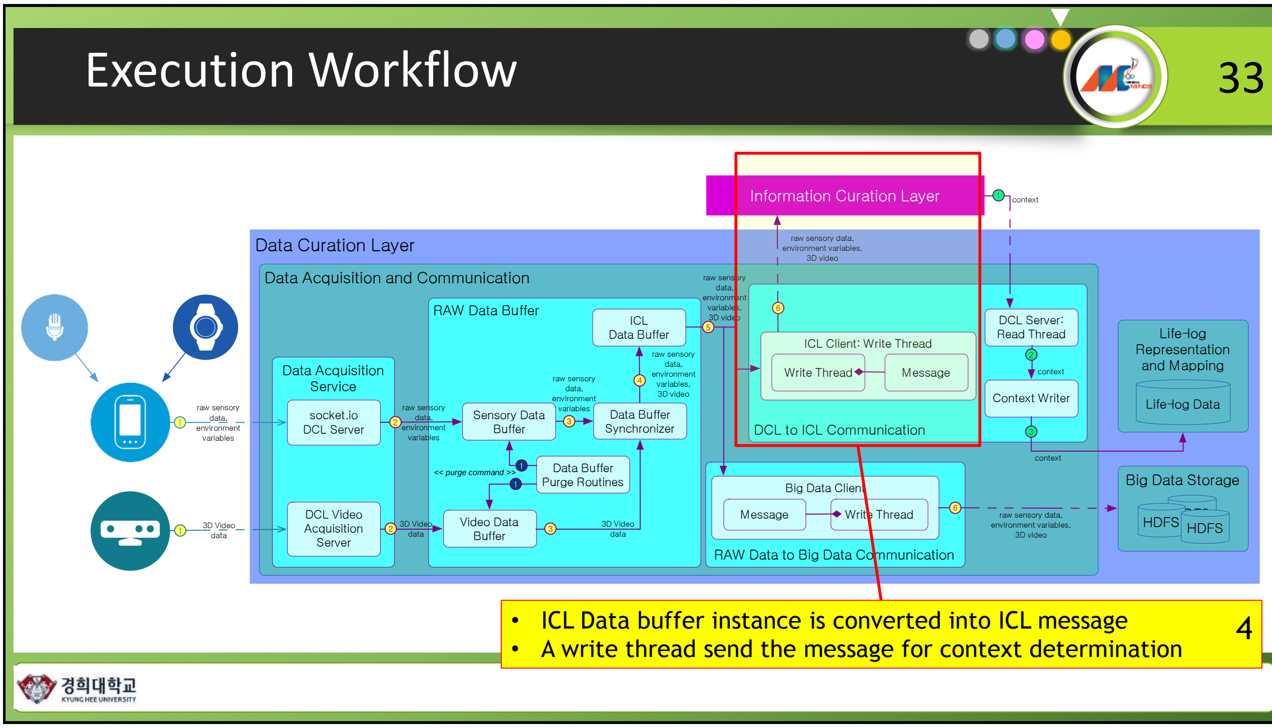
32

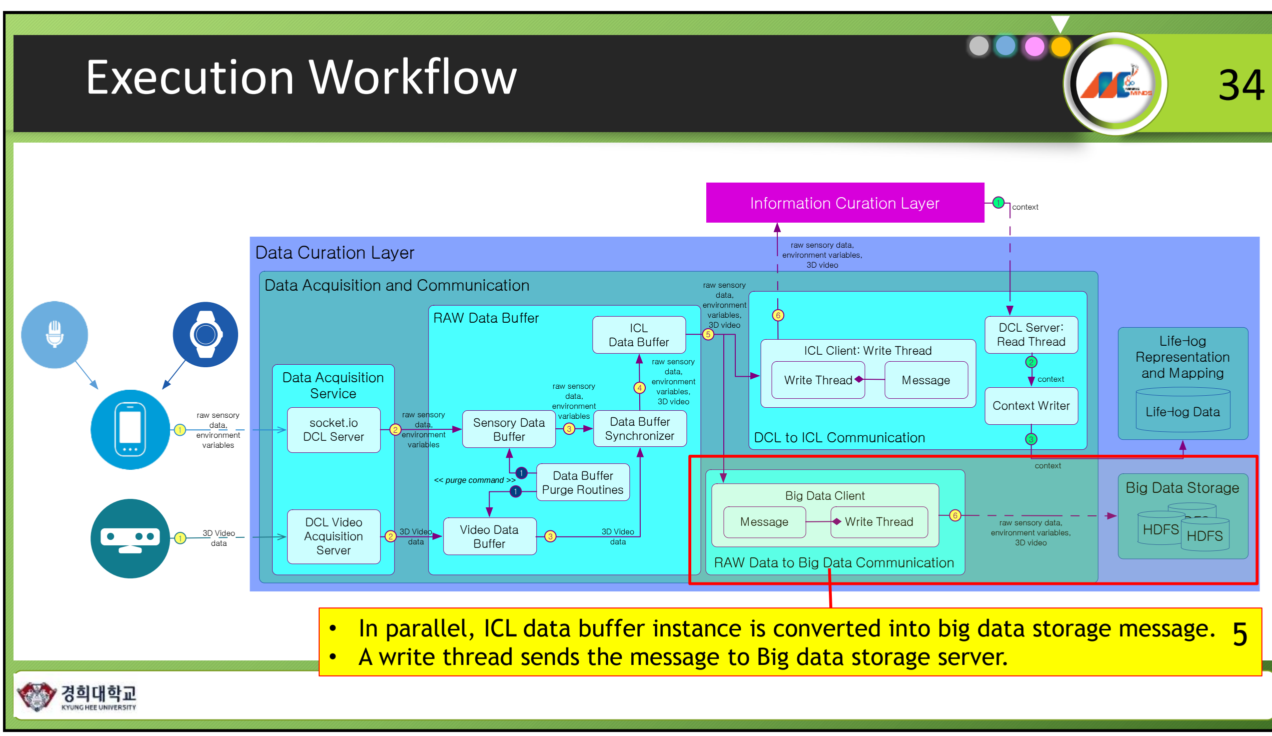


- Data synchronized by time-stamp is moved to ICL buffer
- Data items are purged from Sensory and Video Data buffers

3

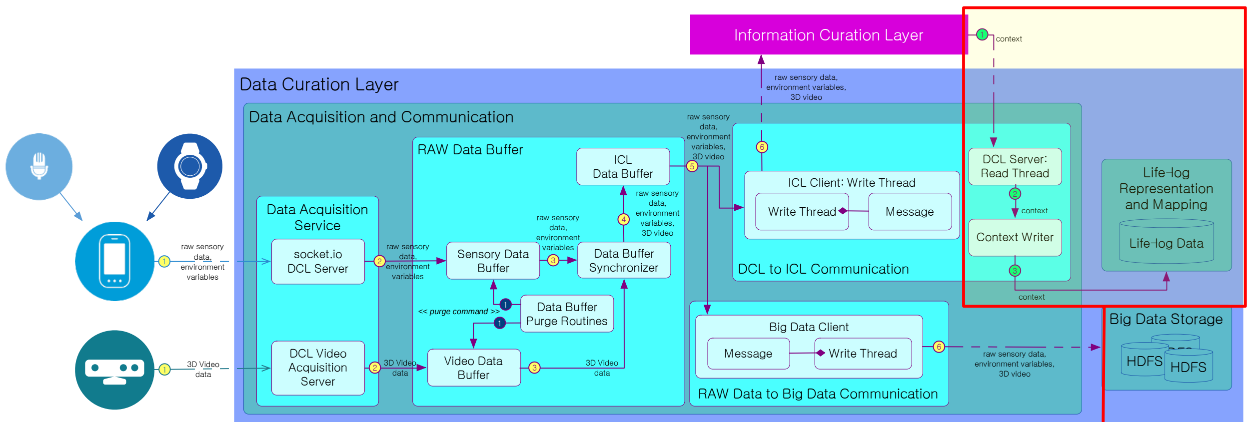
경희대학교 KYUNG HEE UNIVERSITY


Execution Workflow
33
Information Curation Layer
context
raw sensory data, environment variables, 3D video
Data Curation Layer
Data Acquisition and Communication
RAW Data Buffer
ICL Data Buffer
Data Acquisition Service
socket.io DCL Server
Sensory Data Buffer
Data Buffer Synchronizer
Data Buffer Purge Routines
<< purge command >>
DCL Video Acquisition Server
Video Data Buffer
3D Video data
raw sensory data, environment variables
ICL Client: Write Thread
Write Thread
Message
DCL Server: Read Thread
Context Writer
DCL to ICL Communication
Life-log Representation and Mapping
Life-log Data
Big Data Client
Message
Write Thread
RAW Data to Big Data Communication
Big Data Storage
HDFS
HDFS
• ICL Data buffer instance is converted into ICL message
• A write thread send the message for context determination
4
경희대학교
KYUNG HEE UNIVERSITY


Execution Workflow
34
Information Curation Layer
context
raw sensory data, environment variables, 3D video
Data Curation Layer
Data Acquisition and Communication
RAW Data Buffer
ICL Data Buffer
Data Acquisition Service
socket.io DCL Server
Sensory Data Buffer
Data Buffer Synchronizer
Data Buffer Purge Routines
<< purge command >>
DCL Video Acquisition Server
Video Data Buffer
3D Video data
raw sensory data, environment variables
ICL Client: Write Thread
Write Thread
Message
DCL Server: Read Thread
Context Writer
DCL to ICL Communication
Life-log Representation and Mapping
Life-log Data
Big Data Client
Message
Write Thread
RAW Data to Big Data Communication
Big Data Storage
HDFS
HDFS
• In parallel, ICL data buffer instance is converted into big data storage message.
• A write thread sends the message to Big data storage server.
5
경희대학교
KYUNG HEE UNIVERSITY

Execution Workflow

35



- With change in context, Read thread will receive the new context from ICL 6
- Context is mapped to life-log object model for life-log persistence

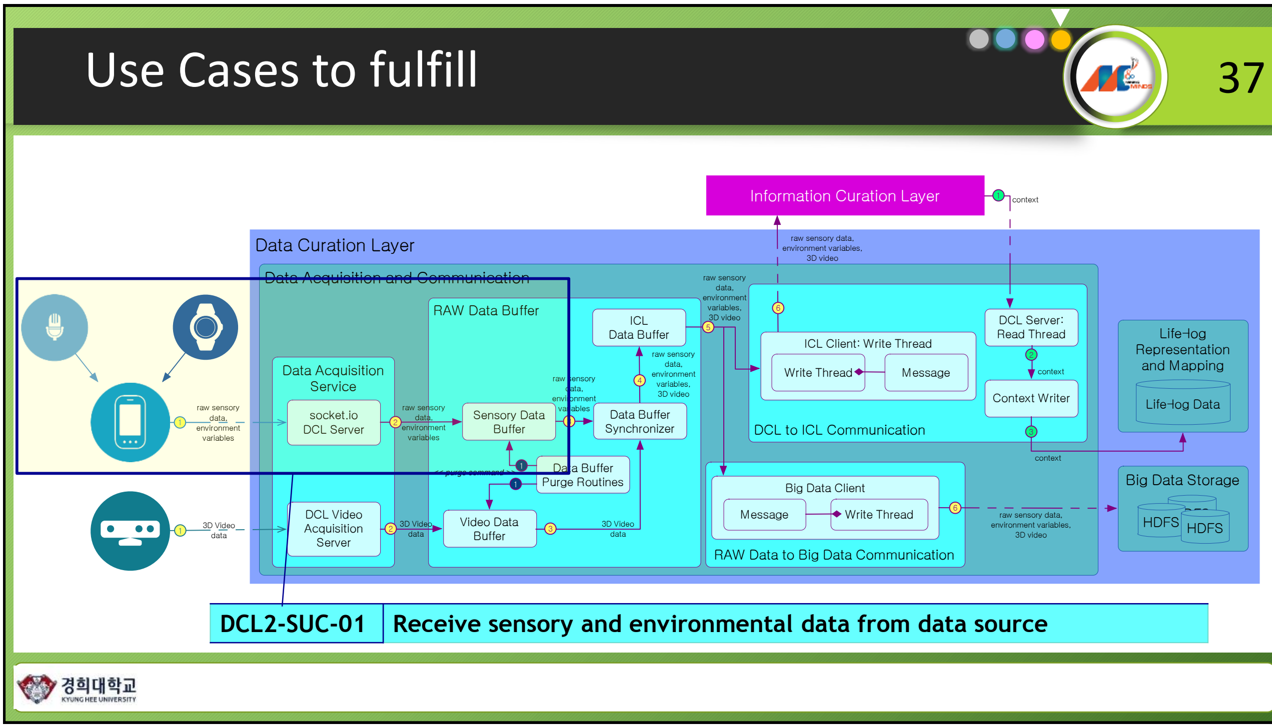
경희대학교 KYUNG HEE UNIVERSITY

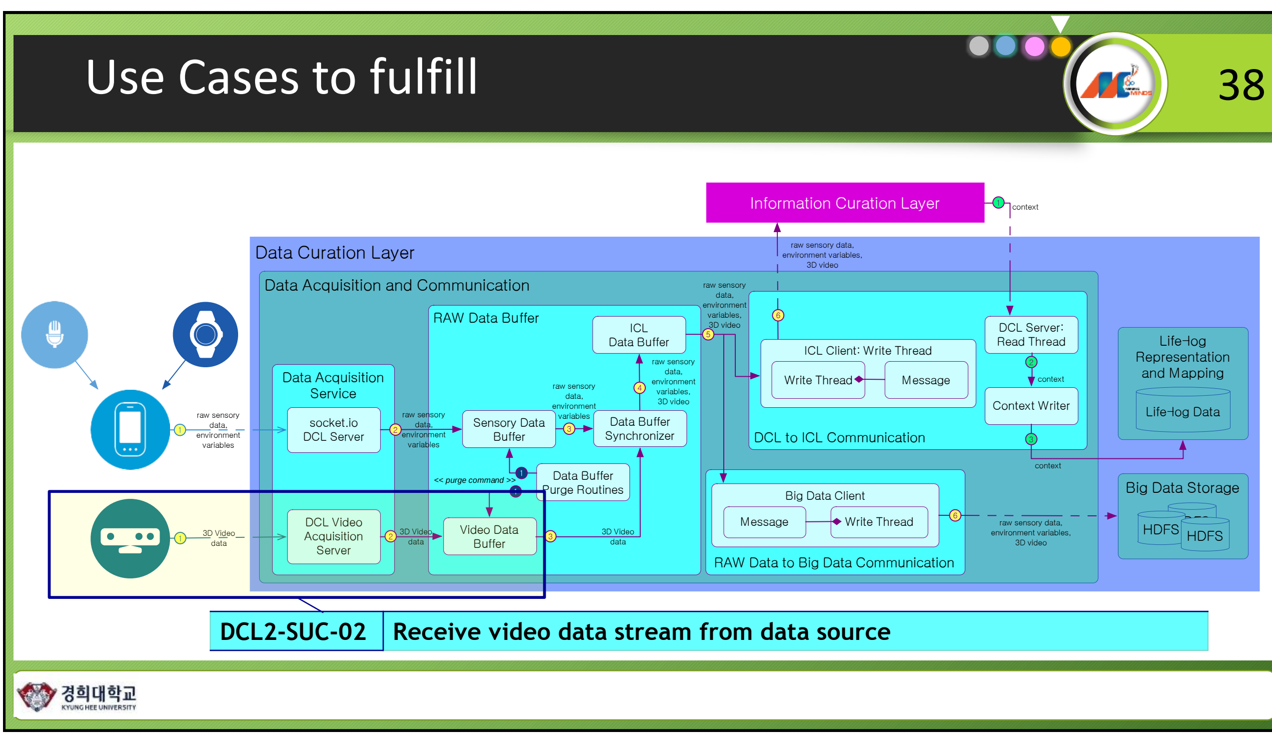
DCLv2.5 Use cases

36

Use case ID	Use Case Description
DCL2.5-SUC-01	Receive sensory and environmental data from data source
DCL2.5-SUC-02	Receive video data stream from data source
DCL2.5-SUC-03	Synchronize heterogeneous user data
DCL2.5-SUC-04	Send data for context determination
DCL2.5-SUC-05	Receive context data
DCL2.5-SUC-15	Persist sensory data in non-volatile storage

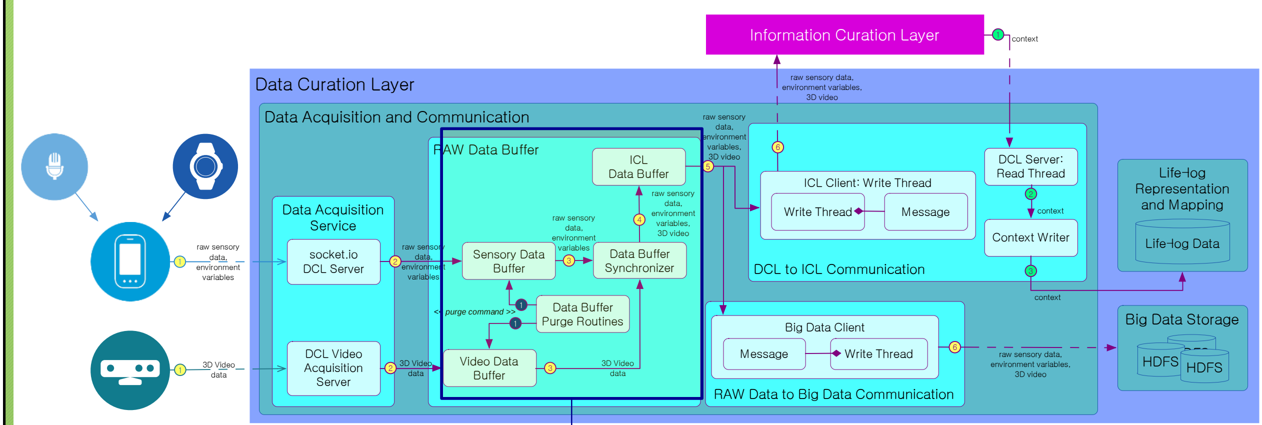
경희대학교 KYUNG HEE UNIVERSITY


Use Cases to fulfill
37
Information Curation Layer
context
Data Curation Layer
raw sensory data, environment variables, 3D video
Data Acquisition and Communication
RAW Data Buffer
ICL Data Buffer
DCL Server: Read Thread
ICL Client: Write Thread
Write Thread
Message
Context Writer
Data Acquisition Service
socket.io DCL Server
Sensory Data Buffer
Data Buffer Synchronizer
Data Buffer Purge Routines
<< purge command >>
DCL to ICL Communication
Life-log Representation and Mapping
Life-log Data
Big Data Storage
HDFS
Big Data Client
DCL Video Acquisition Server
Video Data Buffer
3D Video data
RAW Data to Big Data Communication
DCL2-SUC-01
Receive sensory and environmental data from data source
경희대학교
KYUNG HEE UNIVERSITY


Use Cases to fulfill
38
Information Curation Layer
context
Data Curation Layer
raw sensory data, environment variables, 3D video
Data Acquisition and Communication
RAW Data Buffer
ICL Data Buffer
DCL Server: Read Thread
ICL Client: Write Thread
Write Thread
Message
Context Writer
Data Acquisition Service
socket.io DCL Server
Sensory Data Buffer
Data Buffer Synchronizer
Data Buffer Purge Routines
<< purge command >>
DCL to ICL Communication
Life-log Representation and Mapping
Life-log Data
Big Data Storage
HDFS
Big Data Client
DCL Video Acquisition Server
Video Data Buffer
3D Video data
RAW Data to Big Data Communication
DCL2-SUC-02
Receive video data stream from data source
경희대학교
KYUNG HEE UNIVERSITY

Use Cases to fulfill

39

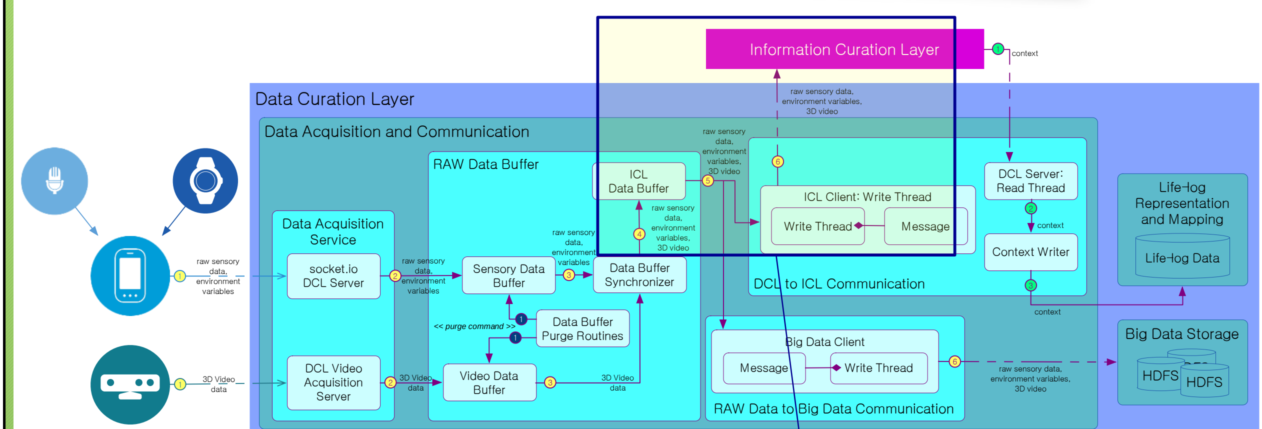


DCL2-SUC-03 **Synchronize heterogeneous user data**

경희대학교 KYUNG HEE UNIVERSITY

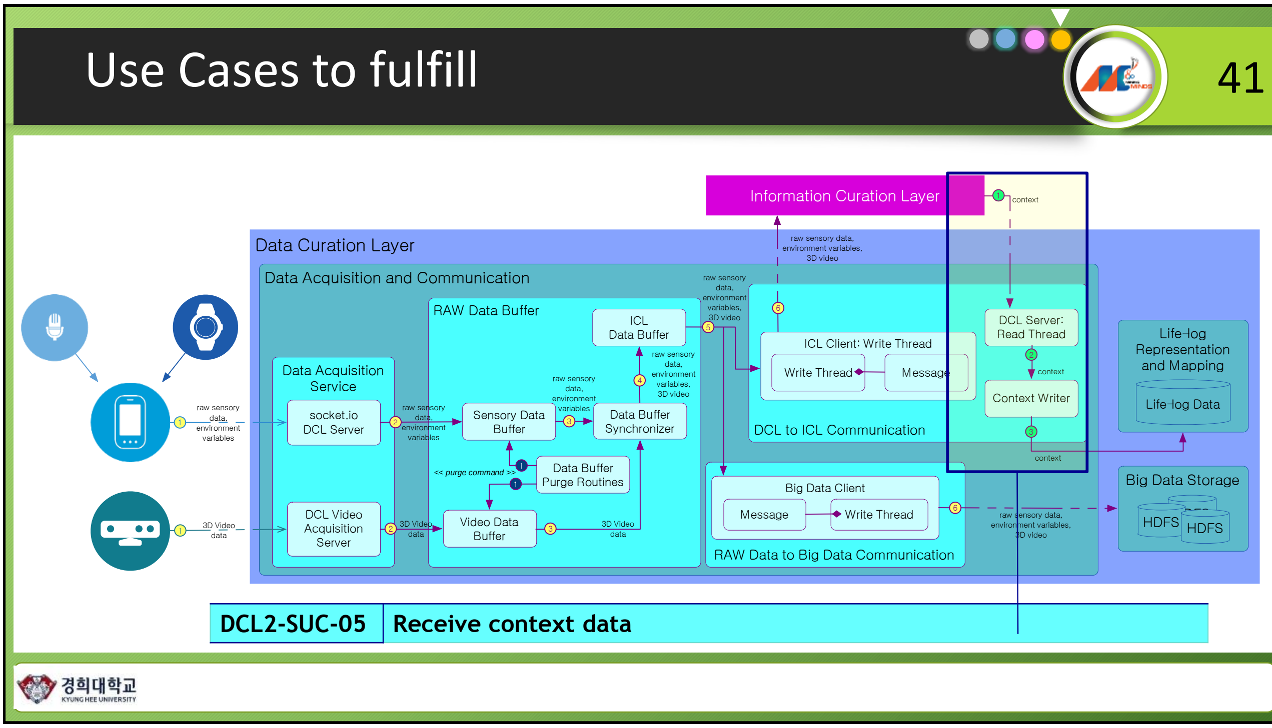
Use Cases to fulfill

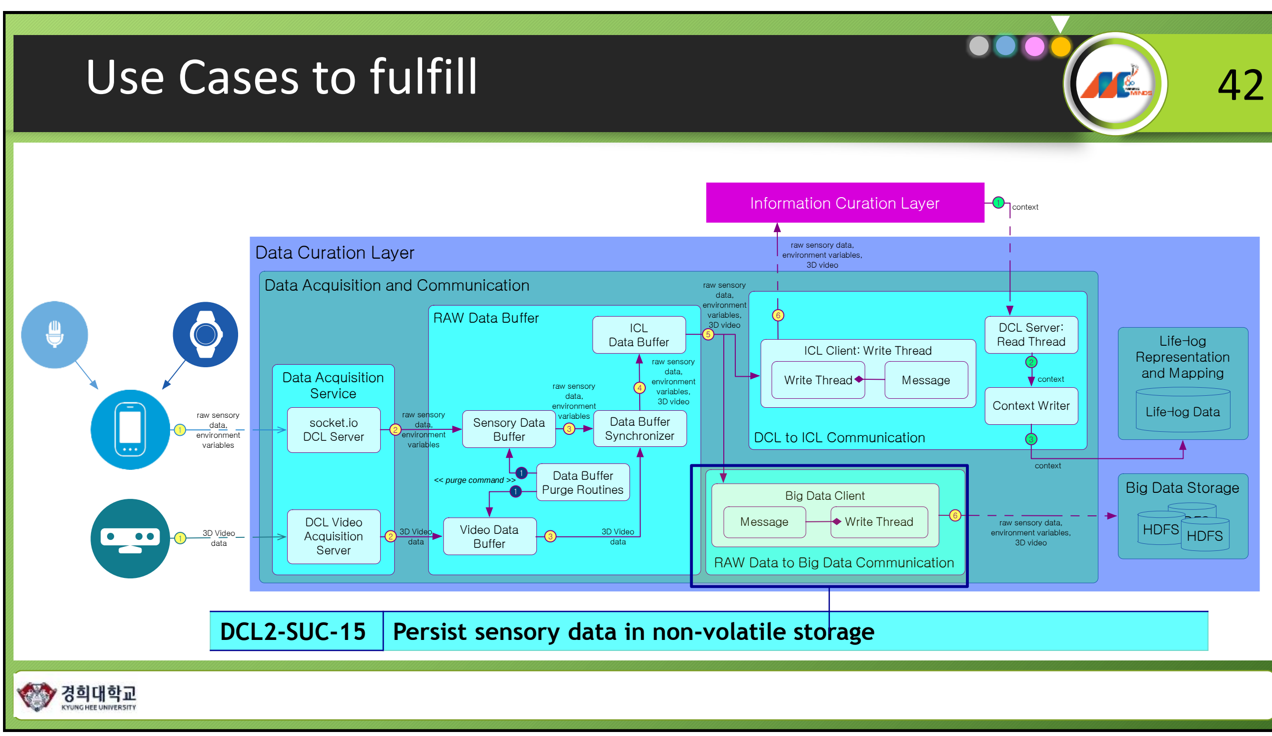
40



DCL2-SUC-04 **Send data for context determination**

경희대학교 KYUNG HEE UNIVERSITY


Use Cases to fulfill
41
Information Curation Layer
context
Data Curation Layer
raw sensory data, environment variables, 3D video
Data Acquisition and Communication
RAW Data Buffer
ICL Data Buffer
DCL Server: Read Thread
Context Writer
ICL Client: Write Thread
Write Thread
Message
DCL to ICL Communication
Data Acquisition Service
socket.io DCL Server
Sensory Data Buffer
Data Buffer Synchronizer
<< purge command >>
Data Buffer Purge Routines
DCL Video Acquisition Server
Video Data Buffer
3D Video data
Big Data Client
Message
Write Thread
RAW Data to Big Data Communication
Life-log Representation and Mapping
Life-log Data
Big Data Storage
HDFS
HDFS
DCL2-SUC-05
Receive context data
경희대학교
KYUNG HEE UNIVERSITY


Use Cases to fulfill
42
Information Curation Layer
context
Data Curation Layer
raw sensory data, environment variables, 3D video
Data Acquisition and Communication
RAW Data Buffer
ICL Data Buffer
DCL Server: Read Thread
Context Writer
ICL Client: Write Thread
Write Thread
Message
DCL to ICL Communication
Data Acquisition Service
socket.io DCL Server
Sensory Data Buffer
Data Buffer Synchronizer
<< purge command >>
Data Buffer Purge Routines
DCL Video Acquisition Server
Video Data Buffer
3D Video data
Big Data Client
Message
Write Thread
RAW Data to Big Data Communication
Life-log Representation and Mapping
Life-log Data
Big Data Storage
HDFS
HDFS
DCL2-SUC-15
Persist sensory data in non-volatile storage
경희대학교
KYUNG HEE UNIVERSITY

Implementation Details

- Actionable Items
  1. Application to DCL Real-time Communication
  2. Regulation of Data Influx from DCL to ICL for Context Determination
  3. DCL to ICL Communication
  4. Non-volatile Data Persistence

1. Application to DCL Real-time Communication

Requirements

○ 1. Non-Blocking Async Recv.

✕ 2. Real-time Communication

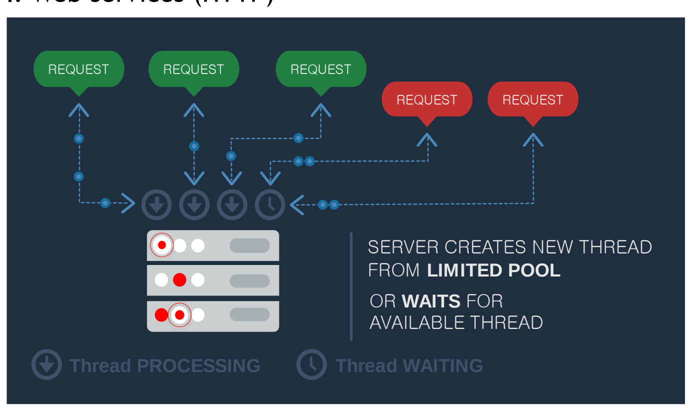
✕ 3. Scalable

Properties

1. Built for Web Applications
2. No Real-time Support
3. Less scalable

Candidate Solutions

I. Web services (HTTP)



1. Application to DCL Real-time Communication

45

Requirements

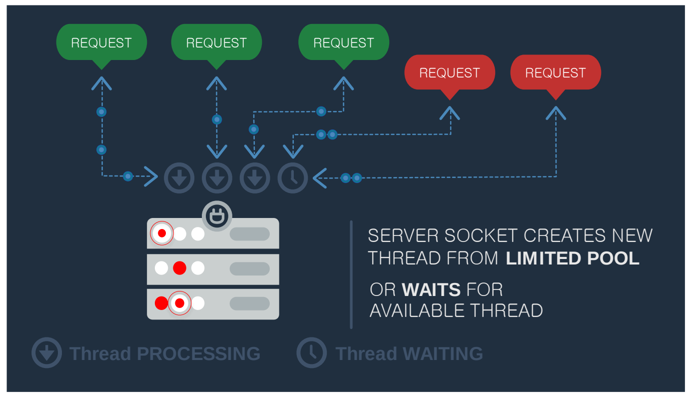
- ✗ 1. Non-Blocking Async Recv.
- ✔ 2. Real-time Communication
- ✗ 3. Scalable

Properties

1. Built for peer-to-peer Network Applications
2. Support for Real-time Communication
3. Blocking IO
4. Less scalable

Candidate Solutions

II. Socket-based Communication (TCP)



경희대학교
KYUNG HEE UNIVERSITY

1. Application to DCL Real-time Communication

46

Requirements

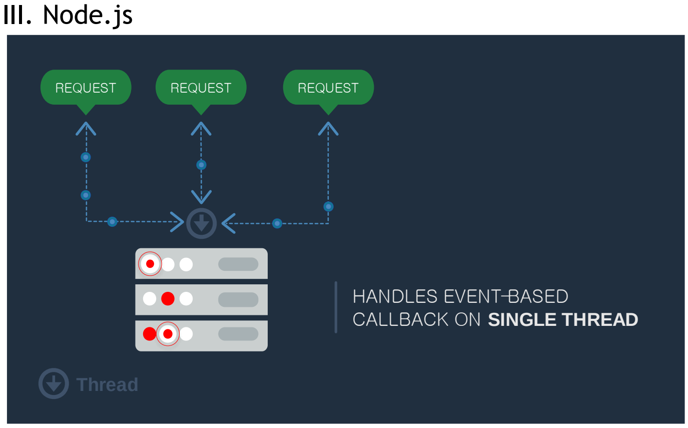
- ✔ 1. Non-Blocking Async Recv.
- ✔ 2. Real-time Communication
- ✔ 3. Scalable

Properties

1. Built for Network Applications
2. Support for Real-time Communication
3. Non-Blocking IO
4. Highly scalable by event-based callback

Candidate Solutions

III. Node.js



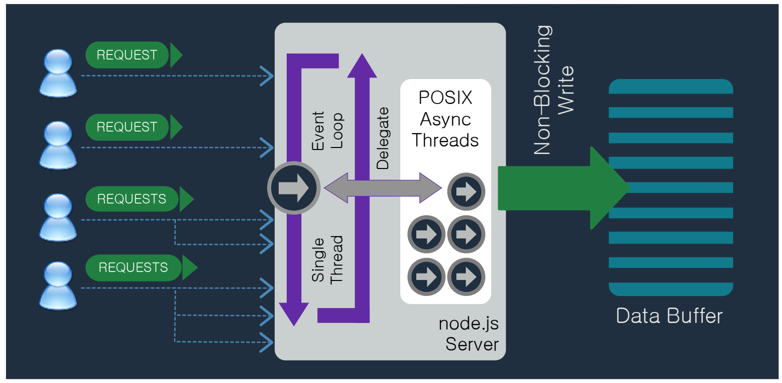
경희대학교
KYUNG HEE UNIVERSITY

1. Application to DCL Real-time Communication

47

Node.js

1. Built for Network Applications
2. Support for Real-time Communication
3. Non-Blocking IO
4. Highly scalable by event-based callback



JavaScript

socket.io

Azure! App! Services!

경희대학교 KYUNG HEE UNIVERSITY

2. Regulation of Data Influx from DCL to ICL

48

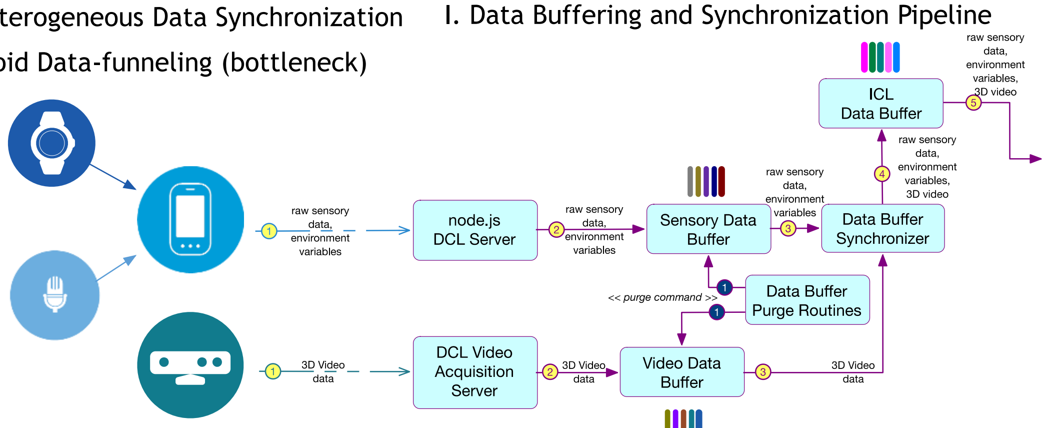
Requirements

✔1. Heterogeneous Data Synchronization

✔2. Avoid Data-funneling (bottleneck)

Candidate Solution

I. Data Buffering and Synchronization Pipeline



경희대학교 KYUNG HEE UNIVERSITY

3. DCL to ICL Communication

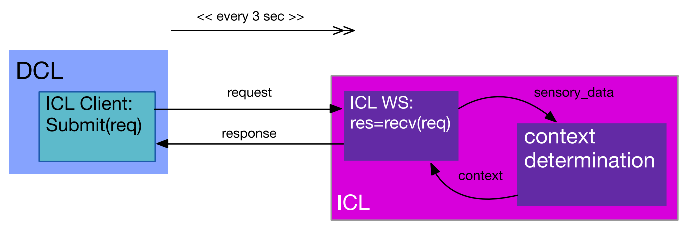
49

Requirements

1. Non-Blocking Communication
2. Reduced Communication Overhead

Candidate Solutions

I. Web services (HTTP)



Limitations

1. Slow
2. Large Communication Overhead

경희대학교
KYUNG HEE UNIVERSITY

3. DCL to ICL Communication

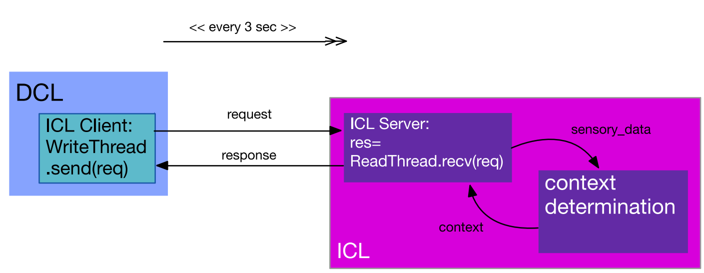
50

Requirements

1. Non-Blocking Communication
2. Reduced Communication Overhead

Candidate Solutions

II. Java sockets (Request-Response)



Limitations

1. Blocking IO
2. Large Communication Overhead

경희대학교
KYUNG HEE UNIVERSITY

3. DCL to ICL Communication

51

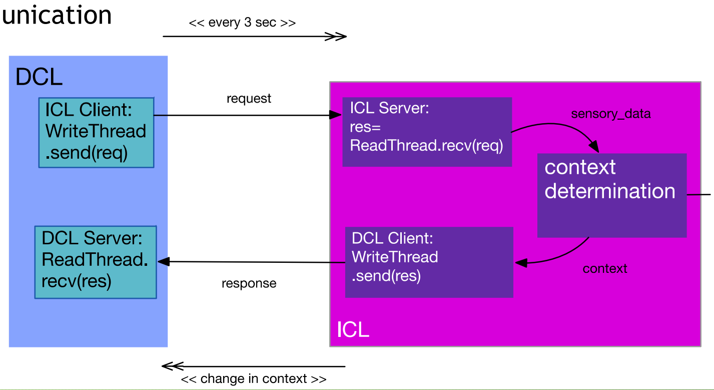
Requirements

✔ 1. Non-Blocking Communication

✔ 2. Reduced Communication Overhead

Candidate Solutions

III. Java sockets (Non-Blocking)




```
if (context_changed == True){
 dclClient = new Thread(DCL);
  dclClient.write(context);
}
```

경희대학교 KYUNG HEE UNIVERSITY

4. Actionable Item : Non-volatile Data Persistence

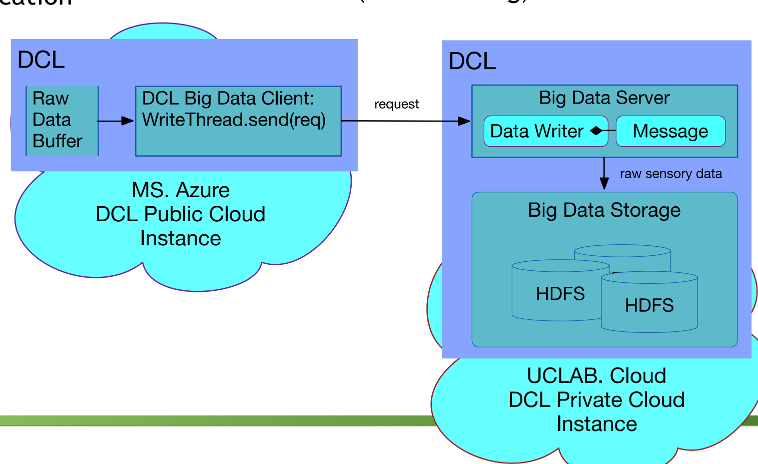
52

Requirements

✔ 1. Non-Blocking Communication

Candidate Solutions

I. Java sockets (Non-Blocking)



경희대학교 KYUNG HEE UNIVERSITY

Contributions

53

- *Acquisition* of Heterogeneous data at *real-time*
- *Synchronization* of Heterogeneous data per user and timestamp
- Buffered pipe-lining of data to ICL for context determination to *avoid ICL data stress*
- Non-blocking IO to *avoid Communication bottlenecks*

경희대학교
KYUNG HEE UNIVERSITY

Life-log Monitoring (LLM)

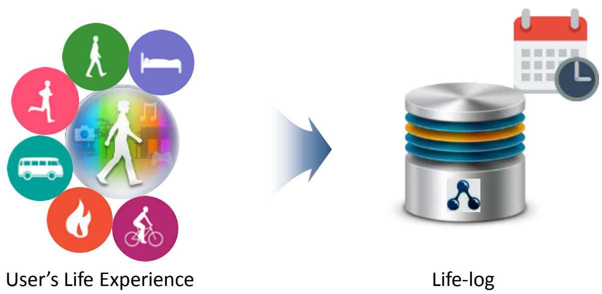
Bilal Ali

경희대학교 KYUNG HEE UNIVERSITY KITECH 한국생산기술연구원 POSTECH 포항공과대학교 UTAS 녹십자헬스케어 TAPACROSS U2 system User & Usability system

Introduction

55

Life-log persistently record and archive some informational dimension of *user's* (user life-log) *life experience* in a particular data category.

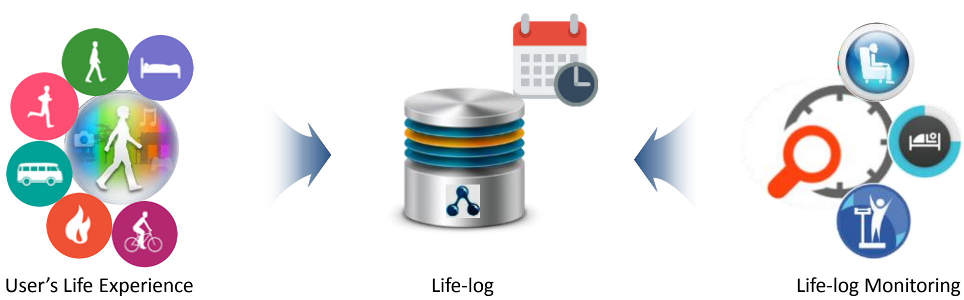


https://en.wikipedia.org/wiki/DARPA_LifeLog

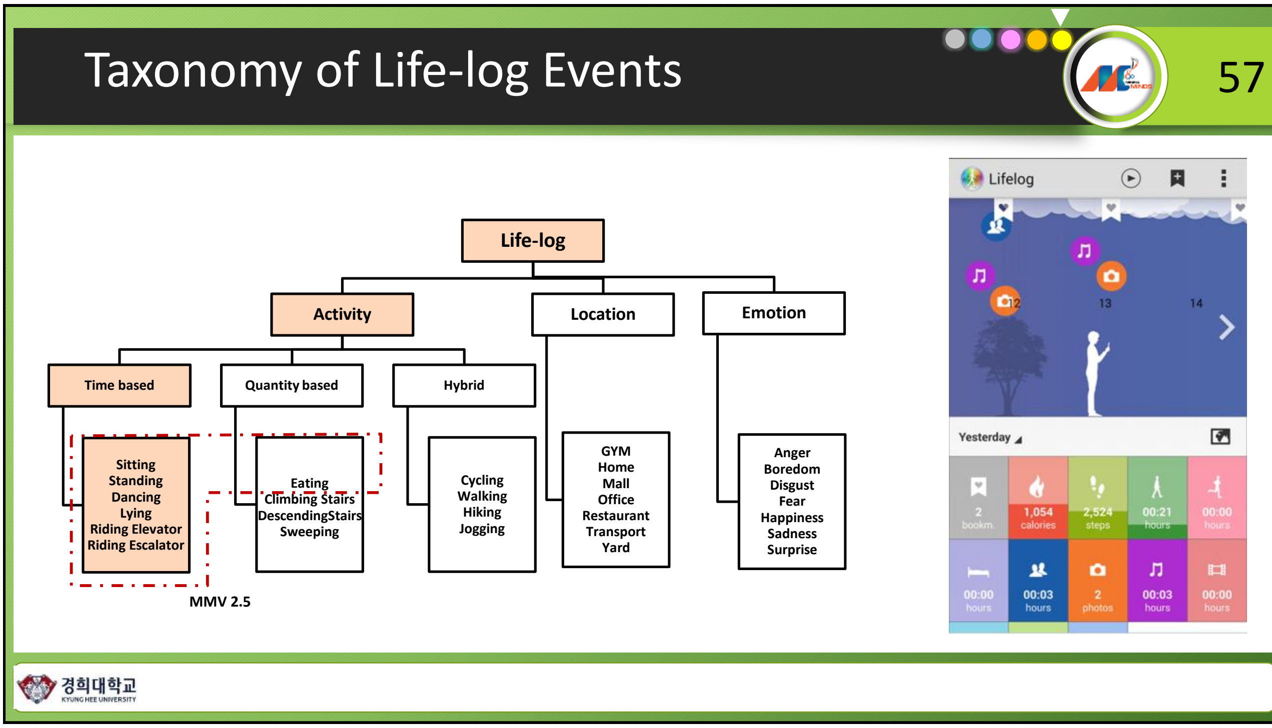
Introduction

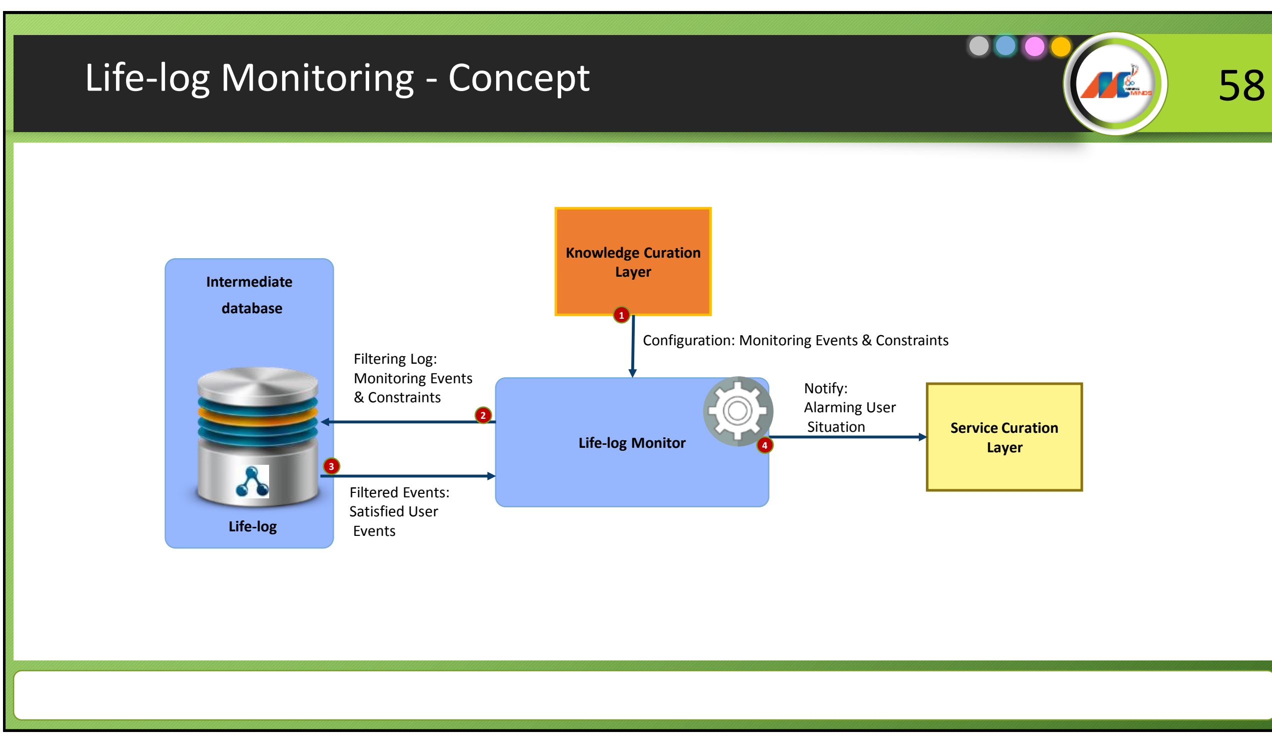
56

Life-log monitoring is to *trace* the 'threads' of an individual's life in terms of events to observe a *situation* over time.



https://en.wikipedia.org/wiki/DARPA_LifeLog


Taxonomy of Life-log Events
57
Life-log
Activity
Location
Emotion
Time based
Quantity based
Hybrid
Sitting
Standing
Dancing
Lying
Riding Elevator
Riding Escalator
Eating
Climbing Stairs
DescendingStairs
Sweeping
Cycling
Walking
Hiking
Jogging
GYM
Home
Mall
Office
Restaurant
Transport
Yard
Anger
Boredom
Disgust
Fear
Happiness
Sadness
Surprise
MMV 2.5
Lifelog
Yesterday
경희대학교
KYUNG HEE UNIVERSITY


Life-log Monitoring - Concept
58
Knowledge Curation Layer
Intermediate database
Life-log
Configuration: Monitoring Events & Constraints
Filtering Log: Monitoring Events & Constraints
Filtered Events: Satisfied User Events
Life-log Monitor
Notify: Alarming User Situation
Service Curation Layer

Why Life-log Monitoring in Mining Minds

59

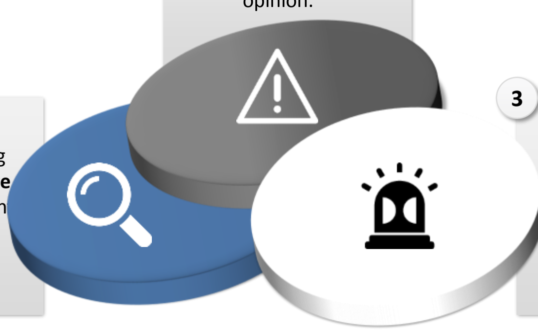
- Recognize the event from stream of life-log
  - Identification of *target* events for efficient monitoring
- Detect the alarming condition from stream of life-log to notify
  - *When* a *user* crossing the threshold of an *activity* which is abnormal
- Dynamically evolve to monitor events
  - To accommodate *when*, *who* and *which* for efficient and effective actions
    - When: Time
    - Who: User
    - Which : Activity

Motivations

60

2
Dynamically accommodate the critical situation on the basis of expert's opinion.

1
Monitor the Life-log in terms of **attractive attributes** to highligh the **atypical** conditions.



3
Evaluate the **alarming** condition in **real time** and generate alarm for producing recommendation.

경희대학교
KYUNG HEE UNIVERSITY

Challenges, Solutions, and Methods

Motivation	Challenges	Solution	Methods
Life-log Monitoring Variables	Selection of Targeted contributing attributes	Monitoring Configuration	Querying target variables based data from Life-log data , preprocess to transform for selecting essential attributes.
Dynamic Situation	Implementation of experts opinion for the identification of rules related to situation.	Constraints Configuration	Evaluation of quantified personal life to identify alarming situation on the basis of experts' decided rule.
Real-time Situation Detection	Identification of current activities form life-log to generate the alarm notification.	Active log Monitoring	Trigger base log management to maintain current running activities for monitoring process after a fixed interval of time.

Related Work

Study	Domain	Service	Methodology	Techniques
(Zini 2015) Life-logs Aggregation for Quality of Life Monitoring	Health Care	Reporting of patient log	An architecture for the acquisition of life-logs, their fusion, and their storage. A prototype GUI to generate report on quality of life indicators	Aggregation of Log
(Kwon 2014) Lifelog Agent for Human Activity Pattern Analysis on Health Avatar Platform	Health Care	Analyze human activity pattern by using lifelog agent cooperating with the Health Avatar platform.	Using the lifelog measured by accelerometer and gyroscope in a smartphone at a 50 Hz rate.	Hourly summarization
(Yang 2015) Life Record:A smartphone based daily Activity Monitoring System	Wellness	Summarize report of daily activities	Two layered architecture to identify the daily activities and provide visualization on smart phone	Daily summarization, Classification of activities.
(Rabbi 2015) Automated Personalized Feedback for Physical Activity and Dietary Behavior Change With Mobile Phones	Wellness	Imitated Feedback and suggestion	Automatic and manual logging to track physical activity	Decision-making algorithm, called multi-armed bandit (MAB)

경희대학교 KYUNG HEE UNIVERSITY

Limitations of Exiting Work

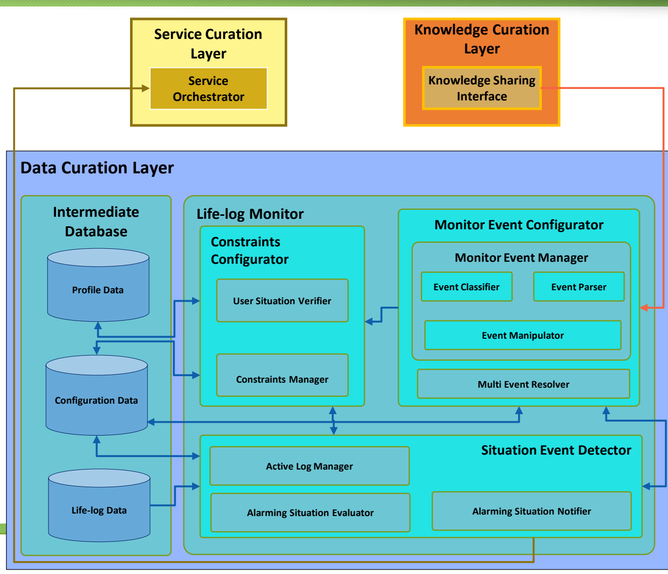
63

- Life-log is only used for *identification* of activities.
- *Limited reporting* on user demand of recorded activities , hourly and daily basis.
- *Small* set of activities are accommodated.
- *Manual* entries of activities for logging.

경희대학교
KYUNG HEE UNIVERSITY

Life-log Monitor – High Level Architecture

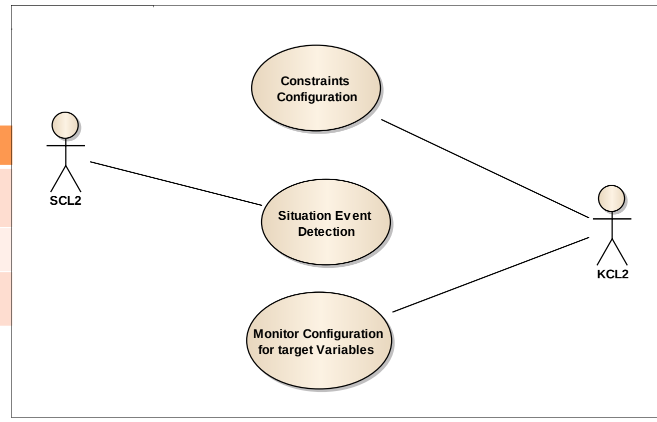
64



Use Case Model- Implementation

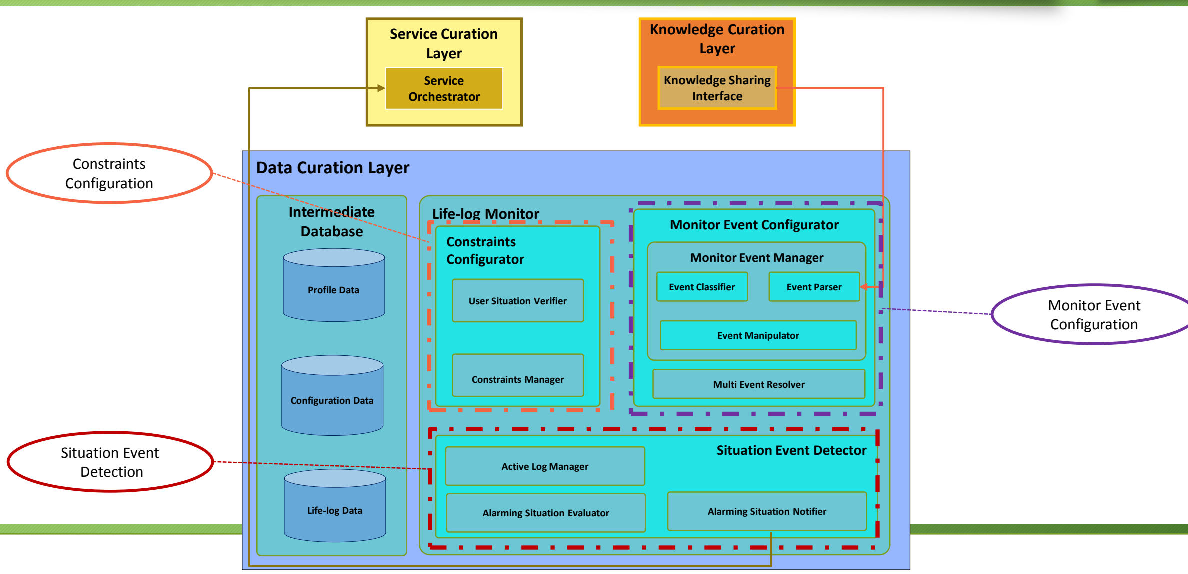
65

Use Case	Description
DCL2-SUC-01	Monitor Event Configuration for target Variables
DCL2-SUC-02	Constraints Configuration
DCL2-SUC-03	Situation Event Detection



Life-log Monitor - Architecture

66

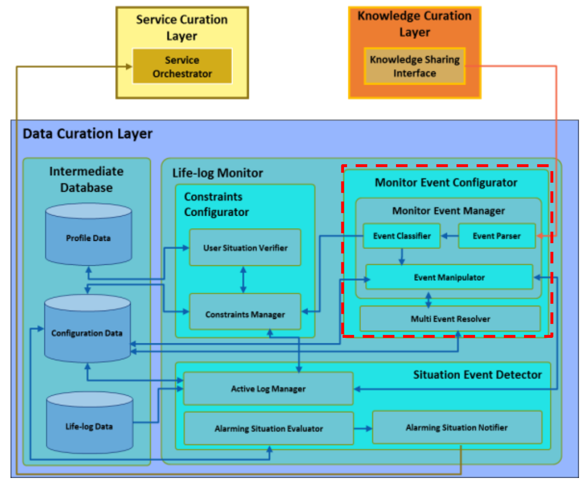


DCL2-SUC-10: Monitor Configuration for target Variables

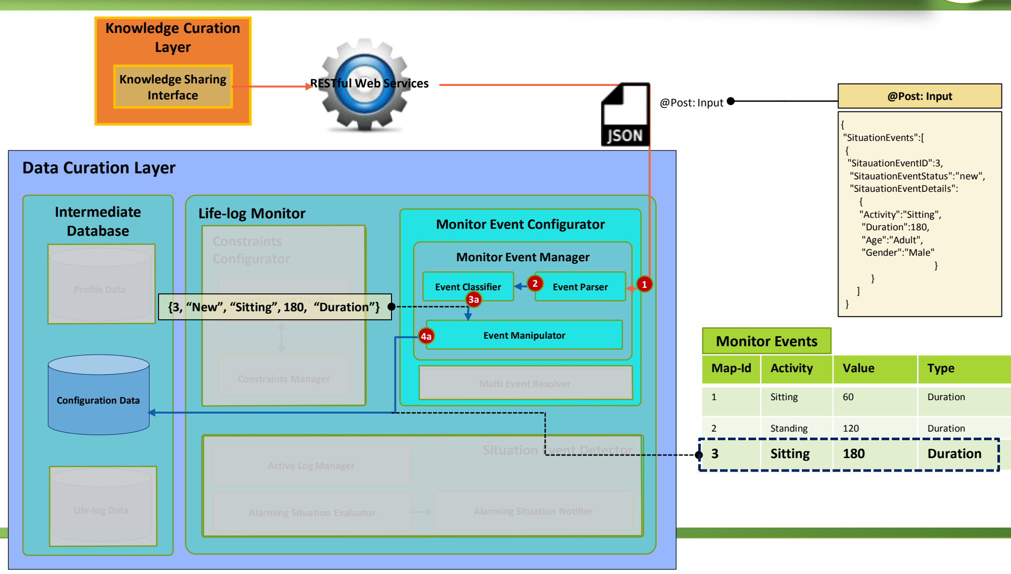
Methodology Description

Use-case (DCL2-UCS-10): *Monitor Configuration for target variables.*

- **Description**
  - Configure the Life-log monitor for the screening of the target variable from Life-log data
- **Input**
  - Target variable in common configuration format
- **Process**
  - KCL 2 will share the target variables in common configured format created by expert.
  - Evaluation and parsing of common configuration format into components.
  - Events' components are classified into constraints and monitoring events.
  - Life-log monitor event is configured on the basis of the shared target variable.
- **Output**
  - Target variable based monitor event configuration.



DCL2-SUC-10: Monitor Configuration for target Variables



Monitor Events

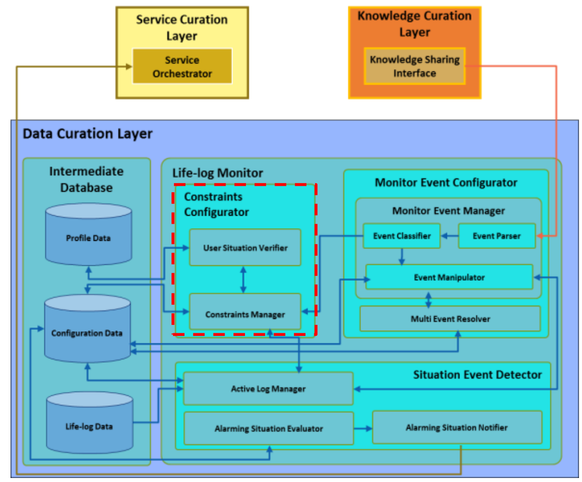
Map-Id	Activity	Value	Type
1	Sitting	60	Duration
2	Standing	120	Duration
3	Sitting	180	Duration

DCL2-SUC-11: Constraints Configuration

69

Methodology Description

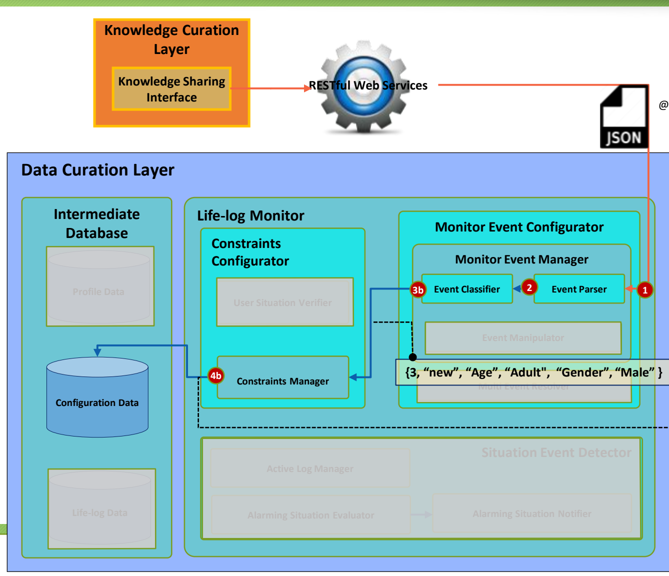
- DCL2-SUC-11: *Constraints Configuration*
- **Description**
  - Situation is determined by experts and is communicated to DCL 2 for monitoring the Life-log against constraints.
- **Input**
  - Classified constraints of a event
- **Process**
  - KCL 2 will share the target variables in common configured format created by expert.
  - Evaluation and parsing of common configuration format into components.
  - Events' components are classified into constraints and monitoring events.
  - Parsed constraints are updated in persistent storage as per categories.
- **Output**
  - *Monitor events based configured constraints.*



DCL2-SUC-11: Constraints Configuration

Implementation

70



@Post: Input


```
{
 "SituationEvents":[
  {
   "SitauationEventID":3,
   "SitauationEventStatus":"new",
   "SitauationEventDetails":
     {
     "Activity":"Sitting",
      "Duration":180,
      "Age":"Adult",
      "Gender":"Male"
                  }
       }
    ]
 }
```


Constraints

S-Id	Key	Value	Map-Id
1	Age	Adult	3
2	Gender	Male	3

DCL2-SUC-12: Situation Detection

Step-1 | Step-2 | Step-3 | Step-4 | Step-5

71

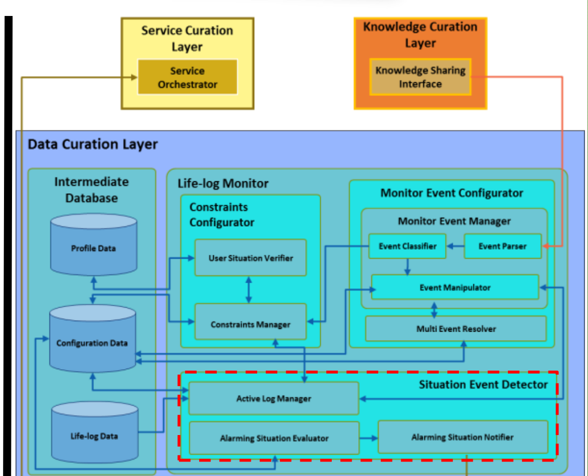
1. Persist Current Active Log
2. Filter Monitoring Attribute
3. Verifies Situation Attributes
4. Manipulate Status for Monitoring
5. Evaluate Activity for Threshold

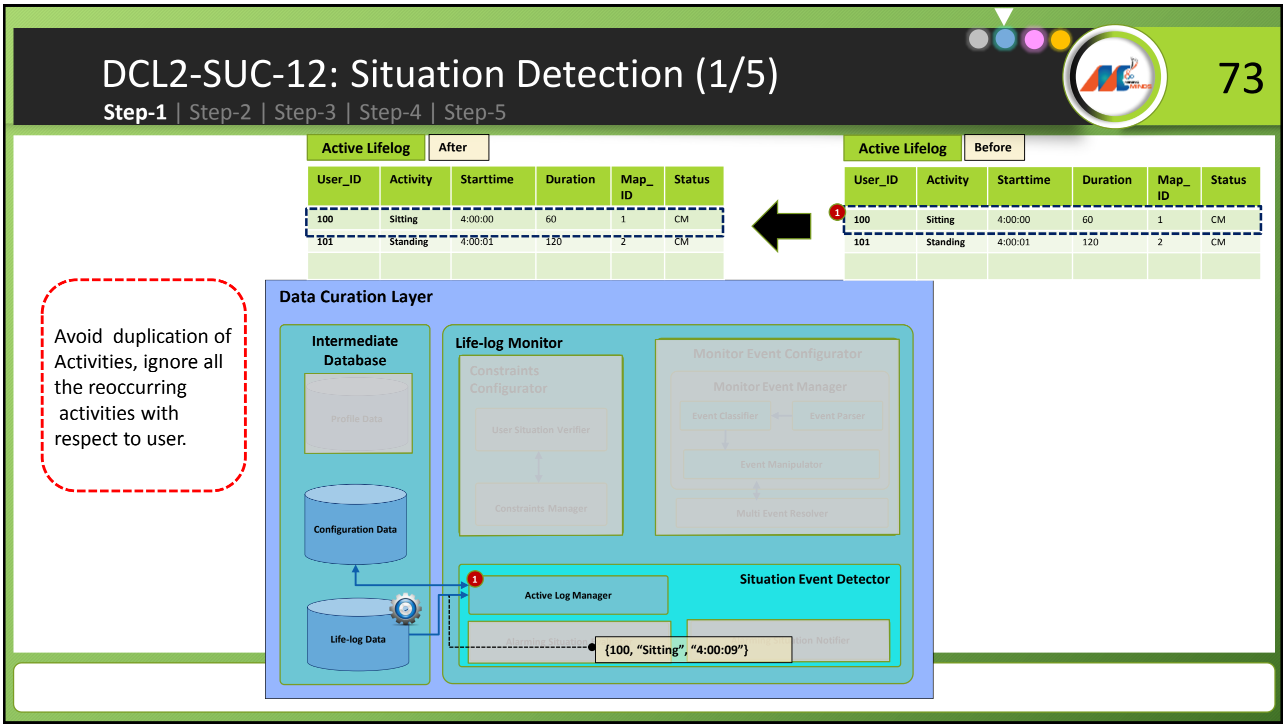
DCL2-SUC-12: Situation Detection

72

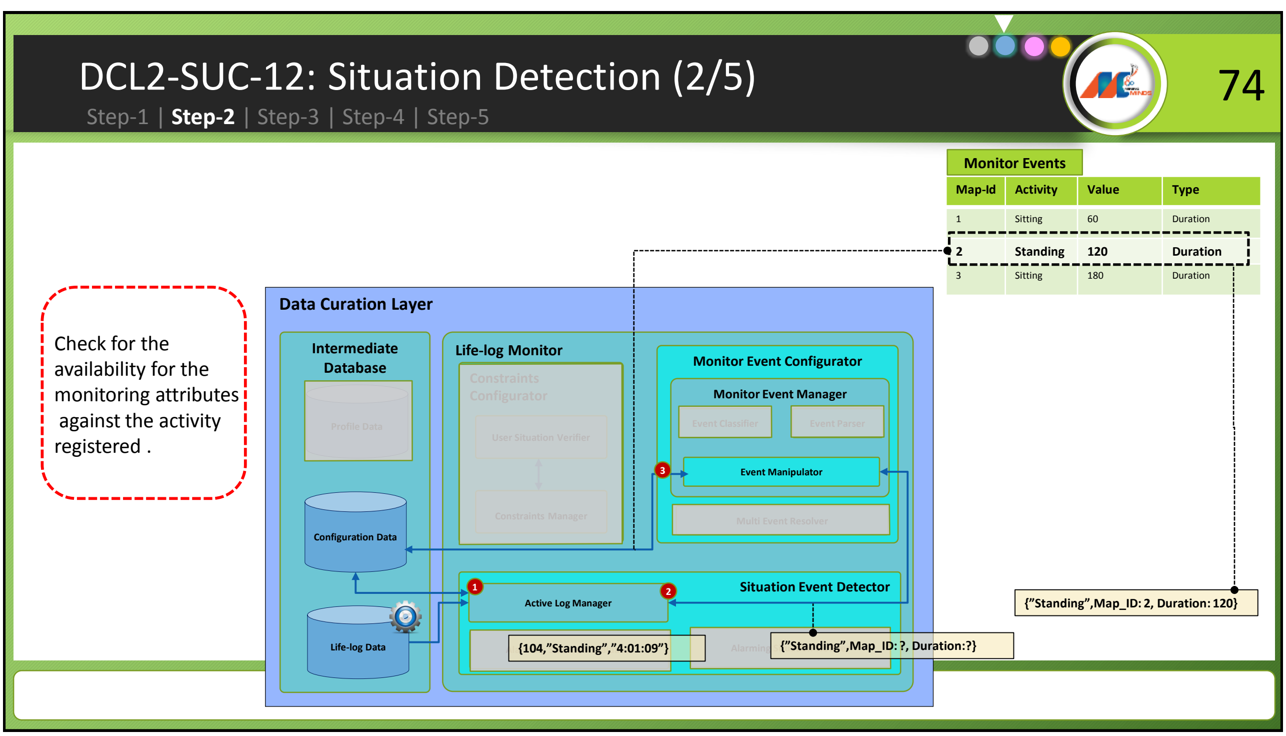
Methodology Description

- **Description**
  - Identification of the existence of a condition in user activities to highlight the alarming situation as per experts' understanding.
- **Input**
  - Activity of a user, when started.
- **Process**
  - ICL 2 recognizes activity and sends to Life-log.
  - Life-log monitor trigger identify the target activity.
  - Retrieve associated situation with the activity.
  - Continuous access that activity log.
  - Aggregate the interval/duration of activity.
  - Remove the irregularity in activity as per situation.
  - Evaluate the duration of activity against the situation.
    - If situation condition meets then send message to SCL 2 to inform about the occurrence of a situation along with user information.
    - If situation condition does not occur, don't send message to SCL 2.
- **Output**
  - Alarming situation of a particular user.

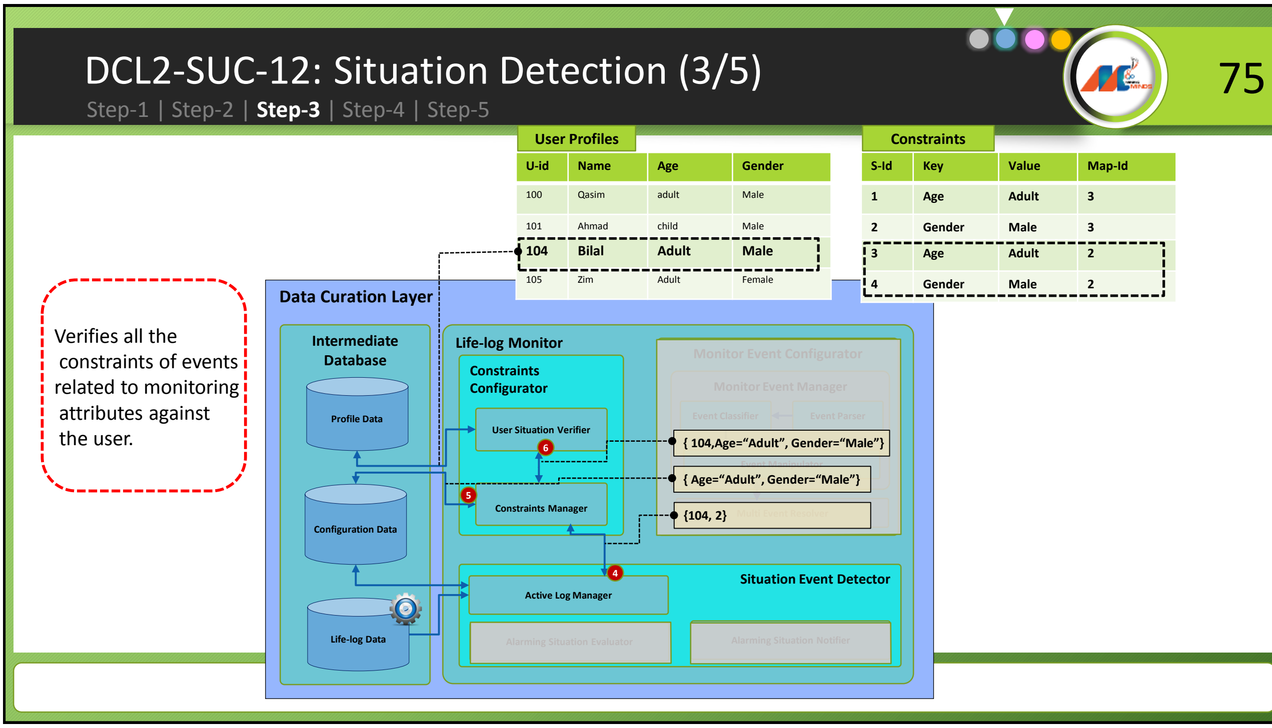


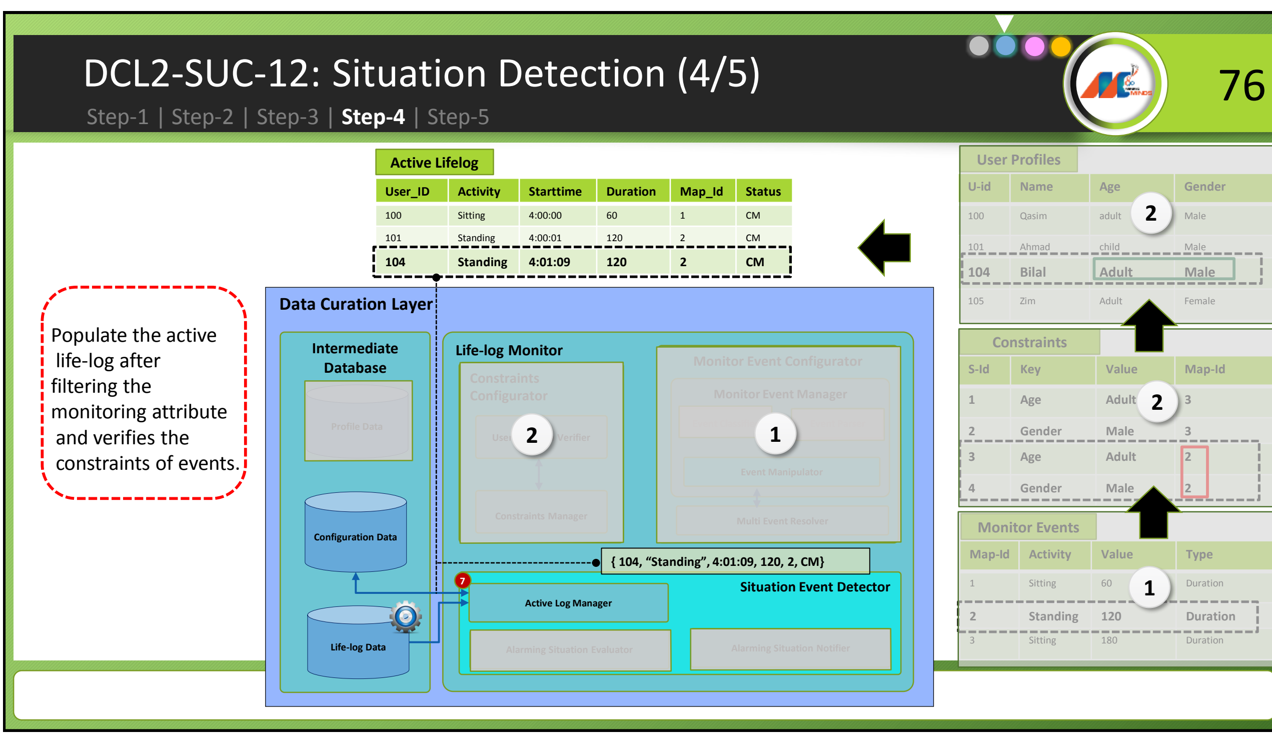


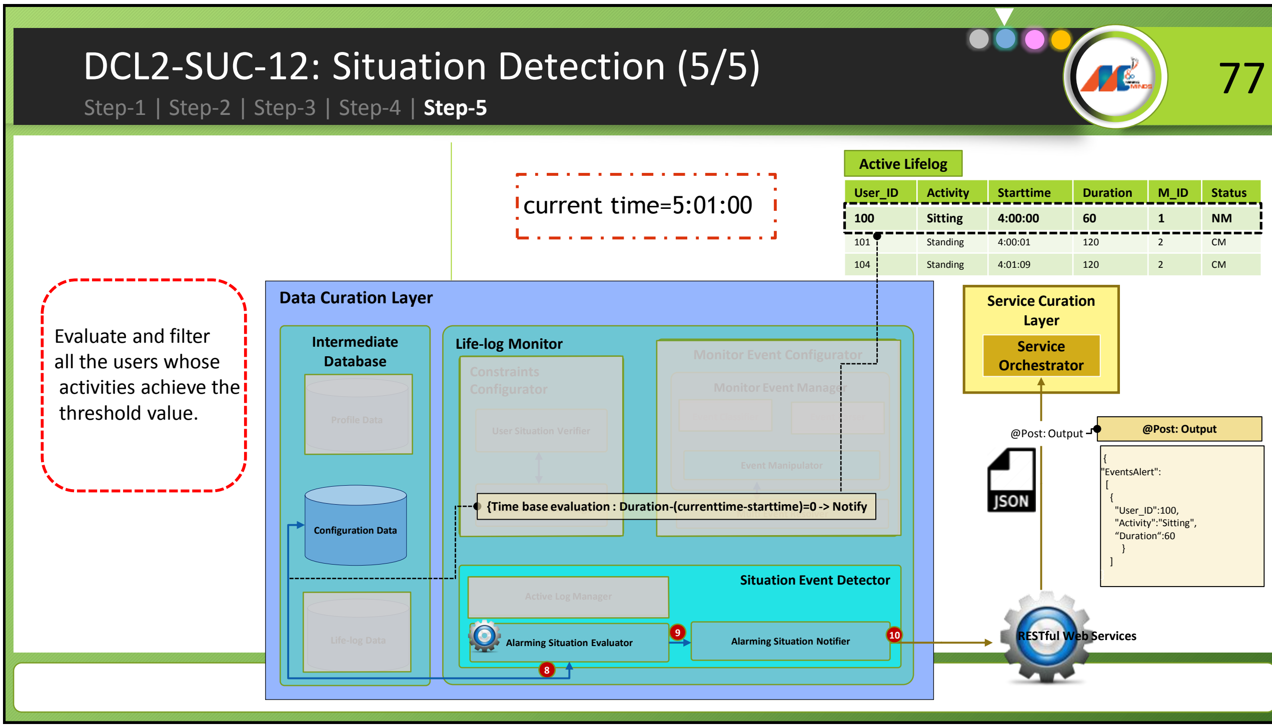
DCL2-SUC-12: Situation Detection (1/5)
Step-1 | Step-2 | Step-3 | Step-4 | Step-5
73
Active Lifelog
After
User_ID Activity Starttime Duration Map_ID Status
100 Sitting 4:00:00 60 1 CM
101 Standing 4:00:01 120 2 CM
Active Lifelog
Before
User_ID Activity Starttime Duration Map_ID Status
100 Sitting 4:00:00 60 1 CM
101 Standing 4:00:01 120 2 CM
Avoid duplication of Activities, ignore all the reoccurring activities with respect to user.
Data Curation Layer
Intermediate Database
Profile Data
Configuration Data
Life-log Data
Life-log Monitor
Constraints Configurator
User Situation Verifier
Constraints Manager
Monitor Event Configurator
Monitor Event Manager
Event Classifier
Event Parser
Event Manipulator
Multi Event Resolver
Situation Event Detector
Active Log Manager
{100, "Sitting", "4:00:09"}

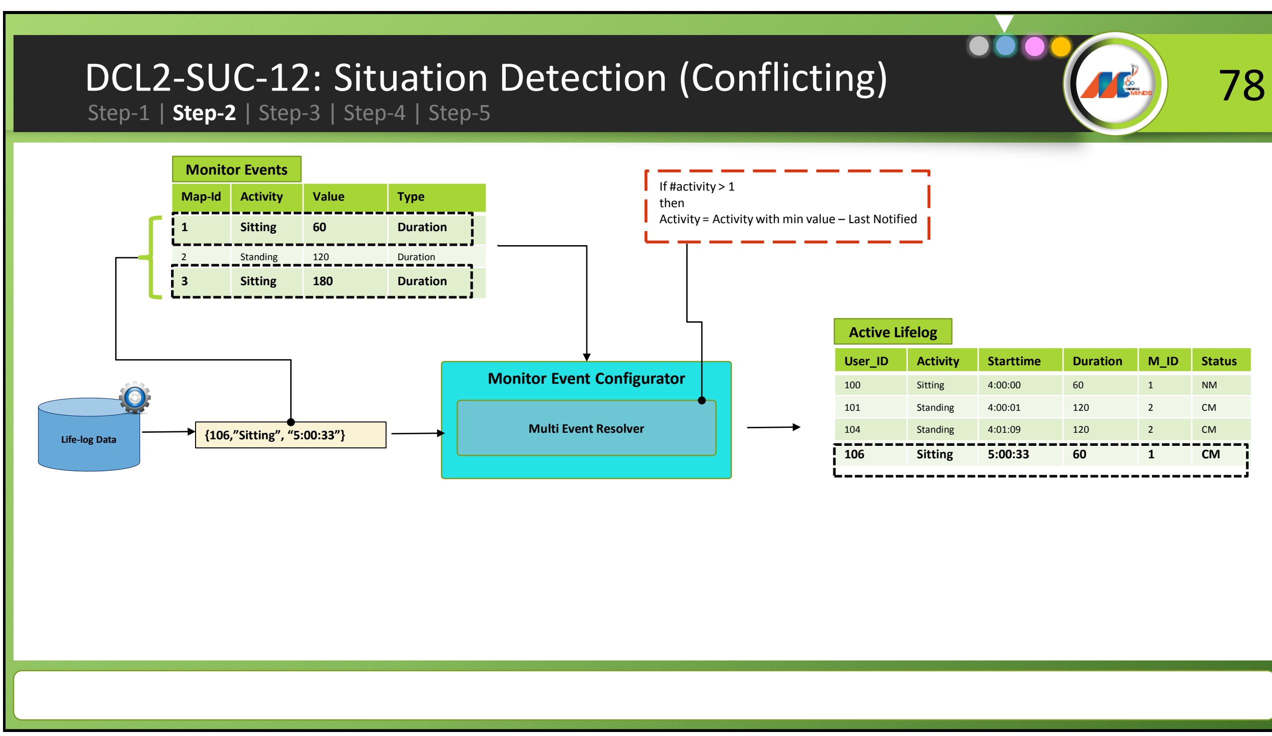


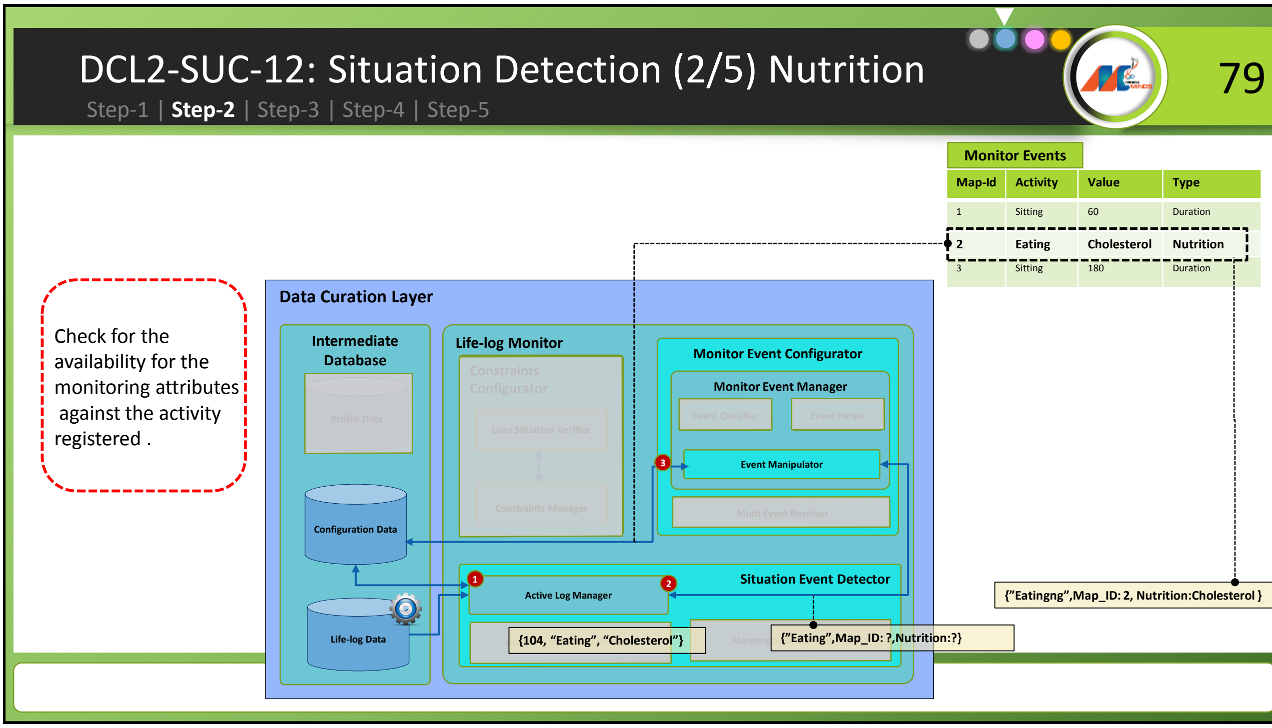
DCL2-SUC-12: Situation Detection (2/5)
Step-1 | Step-2 | Step-3 | Step-4 | Step-5
74
Monitor Events
Map-Id Activity Value Type
1 Sitting 60 Duration
2 Standing 120 Duration
3 Sitting 180 Duration
Check for the availability for the monitoring attributes against the activity registered .
Data Curation Layer
Intermediate Database
Profile Data
Configuration Data
Life-log Data
Life-log Monitor
Constraints Configurator
User Situation Verifier
Constraints Manager
Monitor Event Configurator
Monitor Event Manager
Event Classifier
Event Parser
Event Manipulator
Multi Event Resolver
Situation Event Detector
Active Log Manager
{104,"Standing","4:01:09"}
{"Standing",Map_ID:?, Duration:?}
{"Standing",Map_ID: 2, Duration: 120}

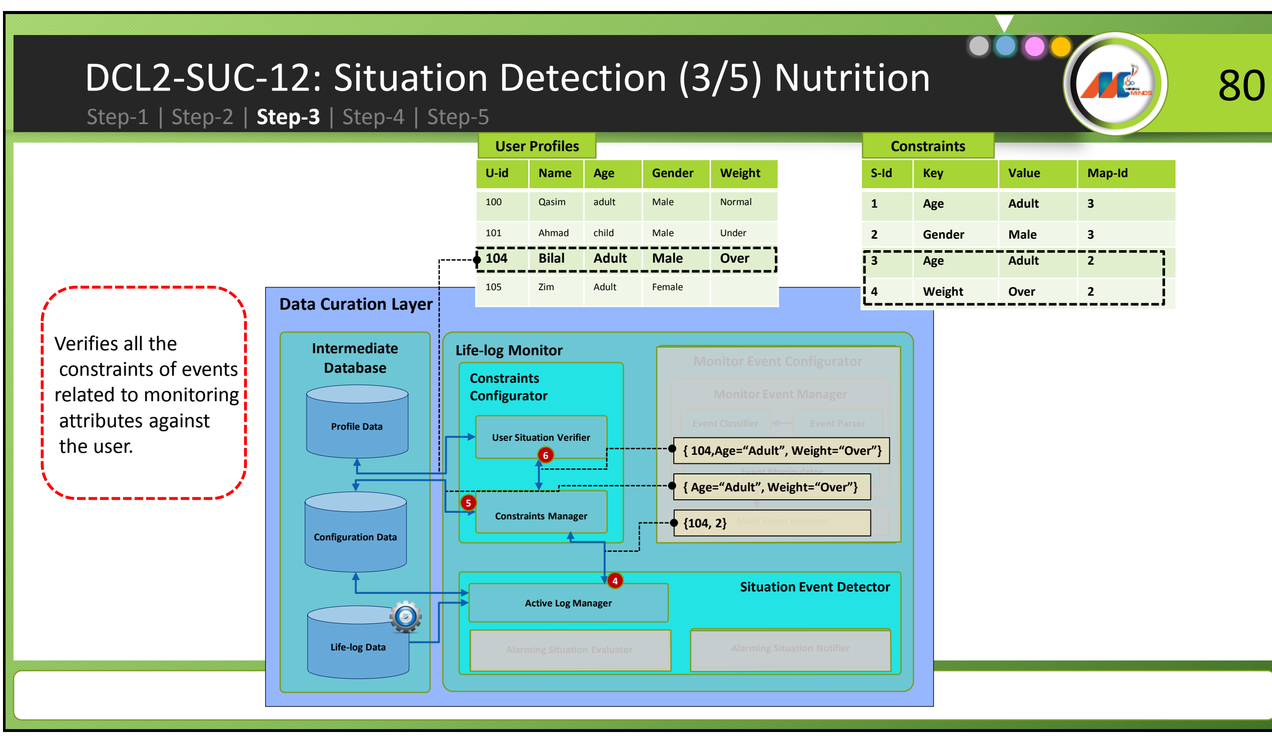

DCL2-SUC-12: Situation Detection (3/5)
Step-1 | Step-2 | Step-3 | Step-4 | Step-5
75
User Profiles
U-id | Name | Age | Gender
100 | Qasim | adult | Male
101 | Ahmad | child | Male
104 | Bilal | Adult | Male
105 | Zim | Adult | Female
Constraints
S-Id | Key | Value | Map-Id
1 | Age | Adult | 3
2 | Gender | Male | 3
3 | Age | Adult | 2
4 | Gender | Male | 2
Verifies all the constraints of events related to monitoring attributes against the user.
Data Curation Layer
Intermediate Database
Profile Data
Configuration Data
Life-log Data
Life-log Monitor
Constraints Configurator
User Situation Verifier
Constraints Manager
Monitor Event Configurator
Monitor Event Manager
{ 104,Age="Adult", Gender="Male"}
{ Age="Adult", Gender="Male"}
{104, 2}
Situation Event Detector
Active Log Manager
Alarming Situation Evaluator
Alarming Situation Notifier

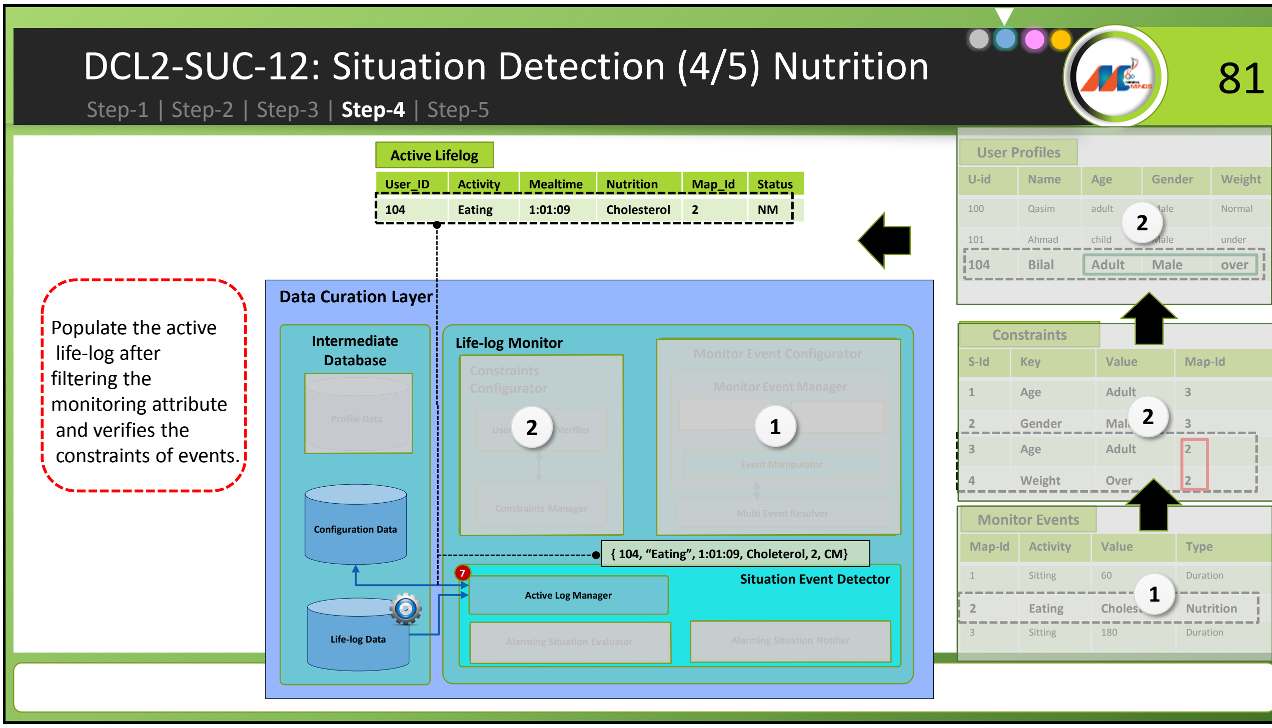

DCL2-SUC-12: Situation Detection (4/5)
Step-1 | Step-2 | Step-3 | Step-4 | Step-5
76
Active Lifelog
User_ID | Activity | Starttime | Duration | Map_Id | Status
100 | Sitting | 4:00:00 | 60 | 1 | CM
101 | Standing | 4:00:01 | 120 | 2 | CM
104 | Standing | 4:01:09 | 120 | 2 | CM
User Profiles
U-id | Name | Age | Gender
100 | Qasim | adult | Male
101 | Ahmad | child | Male
104 | Bilal | Adult | Male
105 | Zim | Adult | Female
Constraints
S-Id | Key | Value | Map-Id
1 | Age | Adult | 3
2 | Gender | Male | 3
3 | Age | Adult | 2
4 | Gender | Male | 2
Monitor Events
Map-Id | Activity | Value | Type
1 | Sitting | 60 | Duration
2 | Standing | 120 | Duration
3 | Sitting | 180 | Duration
Populate the active life-log after filtering the monitoring attribute and verifies the constraints of events.
Data Curation Layer
Intermediate Database
Configuration Data
Life-log Data
Life-log Monitor
Monitor Event Configurator
Monitor Event Manager
{ 104, "Standing", 4:01:09, 120, 2, CM}
Situation Event Detector
Active Log Manager
1
2

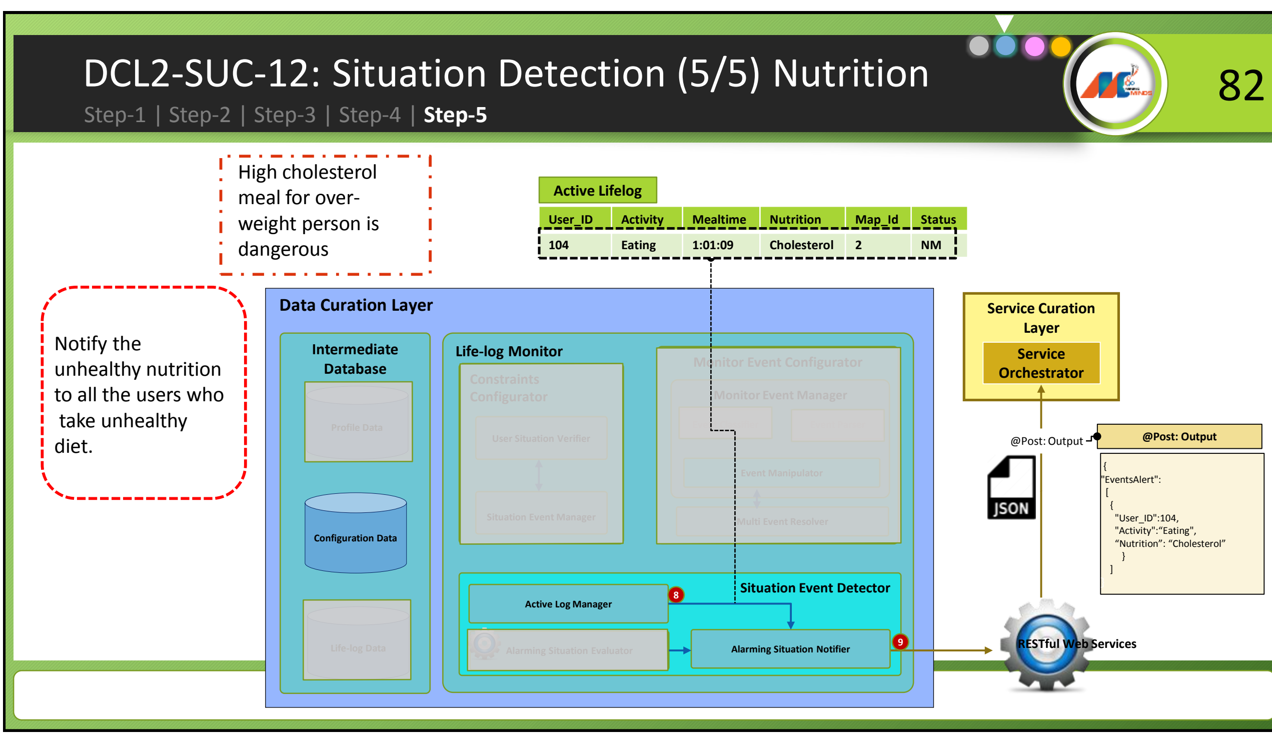

DCL2-SUC-12: Situation Detection (5/5)
Step-1 | Step-2 | Step-3 | Step-4 | Step-5
77
current time=5:01:00
Active Lifelog
User_ID Activity Starttime Duration M_ID Status
100 Sitting 4:00:00 60 1 NM
101 Standing 4:00:01 120 2 CM
104 Standing 4:01:09 120 2 CM
Evaluate and filter all the users whose activities achieve the threshold value.
Data Curation Layer
Intermediate Database
Configuration Data
Life-log Monitor
{Time base evaluation : Duration-(currenttime-starttime)=0 -> Notify
Situation Event Detector
Alarming Situation Evaluator
Alarming Situation Notifier
Service Curation Layer
Service Orchestrator
@Post: Output
@Post: Output
JSON
{
"EventsAlert":
[
{
"User_ID":100,
"Activity":"Sitting",
"Duration":60
}
]
RESTful Web Services


DCL2-SUC-12: Situation Detection (Conflicting)
Step-1 | Step-2 | Step-3 | Step-4 | Step-5
78
Monitor Events
Map-Id Activity Value Type
1 Sitting 60 Duration
2 Standing 120 Duration
3 Sitting 180 Duration
If #activity > 1
then
Activity = Activity with min value – Last Notified
Life-log Data
{106,"Sitting", "5:00:33"}
Monitor Event Configurator
Multi Event Resolver
Active Lifelog
User_ID Activity Starttime Duration M_ID Status
100 Sitting 4:00:00 60 1 NM
101 Standing 4:00:01 120 2 CM
104 Standing 4:01:09 120 2 CM
106 Sitting 5:00:33 60 1 CM


DCL2-SUC-12: Situation Detection (2/5) Nutrition
Step-1 | Step-2 | Step-3 | Step-4 | Step-5
79
Monitor Events
Map-Id Activity Value Type
1 Sitting 60 Duration
2 Eating Cholesterol Nutrition
3 Sitting 180 Duration
Check for the availability for the monitoring attributes against the activity registered .
Data Curation Layer
Intermediate Database
Profile Data
Configuration Data
Life-log Data
Life-log Monitor
Constraints Configurator
User Situation Verifier
Constraints Manager
Monitor Event Configurator
Monitor Event Manager
Event Classifier
Event Parser
Event Manipulator
Multi Event Resolver
Situation Event Detector
Active Log Manager
{104, "Eating", "Cholesterol"}
{"Eating",Map_ID: ?,Nutrition:?}
{"Eatingng",Map_ID: 2, Nutrition:Cholesterol }

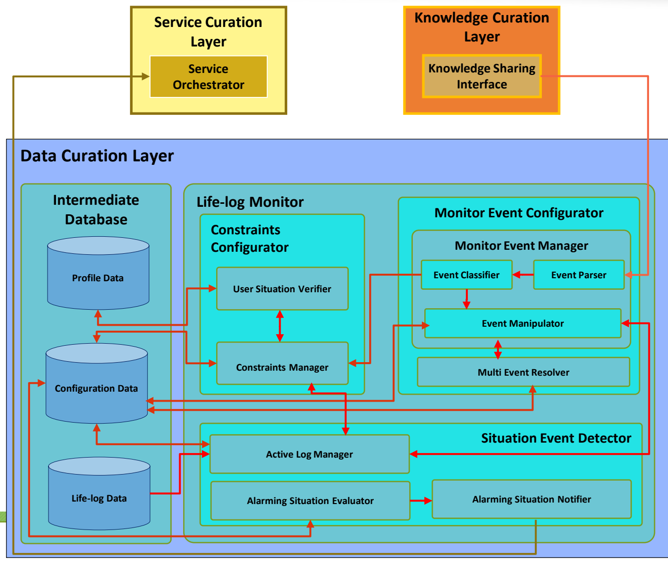

DCL2-SUC-12: Situation Detection (3/5) Nutrition
Step-1 | Step-2 | Step-3 | Step-4 | Step-5
80
User Profiles
U-id Name Age Gender Weight
100 Qasim adult Male Normal
101 Ahmad child Male Under
104 Bilal Adult Male Over
105 Zim Adult Female
Constraints
S-Id Key Value Map-Id
1 Age Adult 3
2 Gender Male 3
3 Age Adult 2
4 Weight Over 2
Verifies all the constraints of events related to monitoring attributes against the user.
Data Curation Layer
Intermediate Database
Profile Data
Configuration Data
Life-log Data
Life-log Monitor
Constraints Configurator
User Situation Verifier
Constraints Manager
Monitor Event Configurator
Monitor Event Manager
{ 104,Age="Adult", Weight="Over"}
{ Age="Adult", Weight="Over"}
{104, 2}
Situation Event Detector
Active Log Manager
Alarming Situation Evaluator
Alarming Situation Notifier


DCL2-SUC-12: Situation Detection (4/5) Nutrition
Step-1 | Step-2 | Step-3 | Step-4 | Step-5
81
Active Lifelog
User_ID | Activity | Mealtime | Nutrition | Map_Id | Status
104 | Eating | 1:01:09 | Cholesterol | 2 | NM
Populate the active life-log after filtering the monitoring attribute and verifies the constraints of events.
Data Curation Layer
Intermediate Database
Profile Data
Configuration Data
Life-log Data
Life-log Monitor
Constraints Configurator
Constraints Manager
Monitor Event Configurator
Monitor Event Manager
Event Manipulator
Multi Event Resolver
{ 104, "Eating", 1:01:09, Choleterol, 2, CM}
Situation Event Detector
Active Log Manager
Alarming Situation Evaluator
Alarming Situation Notifier
User Profiles
U-id | Name | Age | Gender | Weight
100 | Qasim | adult | Male | Normal
101 | Ahmad | child | Male | under
104 | Bilal | Adult | Male | over
Constraints
S-Id | Key | Value | Map-Id
1 | Age | Adult | 3
2 | Gender | Male | 3
3 | Age | Adult | 2
4 | Weight | Over | 2
Monitor Events
Map-Id | Activity | Value | Type
1 | Sitting | 60 | Duration
2 | Eating | Cholesterol | Nutrition
3 | Sitting | 180 | Duration


DCL2-SUC-12: Situation Detection (5/5) Nutrition
Step-1 | Step-2 | Step-3 | Step-4 | Step-5
82
High cholesterol meal for over-weight person is dangerous
Notify the unhealthy nutrition to all the users who take unhealthy diet.
Active Lifelog
User_ID | Activity | Mealtime | Nutrition | Map_Id | Status
104 | Eating | 1:01:09 | Cholesterol | 2 | NM
Data Curation Layer
Intermediate Database
Profile Data
Configuration Data
Life-log Data
Life-log Monitor
Constraints Configurator
User Situation Verifier
Situation Event Manager
Monitor Event Configurator
Monitor Event Manager
Event Manipulator
Multi Event Resolver
Situation Event Detector
Active Log Manager
Alarming Situation Evaluator
Alarming Situation Notifier
Service Curation Layer
Service Orchestrator
@Post: Output
@Post: Output
JSON
{
"EventsAlert":
[
{
"User_ID":104,
"Activity":"Eating",
"Nutrition": "Cholesterol"
}
]
RESTful Web Services

Life-log Monitor – Over All Flow MMV 2.5

83



Selected Tools and Techniques

84

- **Language**
  - Java
- **Database**:
  - MS SQL Server
- **IDEs**:
  - NetBeans

Java

NetBeans

Microsoft SQL Server

경희대학교 KYUNG HEE UNIVERSITY

Contribution

85

- Monitoring of Life-log for occurrence of *alarming situation* in personal life in real time manner.
- Accommodate dynamically *constraints of events* on the recommendation of experts.
- Configuration of Life-log monitor with *contributing factors* / *target variables*.

Life-log Representation and Mapping

Taqdir Ali

Introduction

87

- **Representation of Lifelog:** Daily Activities, Goals, Recommendations and feedback have Continuous Semantics.

Total 9 representation models in literature

Related Work

88

Graphical Representation [1], [2]	Logic Based Representation [3]	Ontological Representation [4], [5], [6], [7], [16], [17]
<u>Features</u> ➢ Diagrammatic Representation at design time ➢ Focus on issues of formality and generality ➢ Cover diversity, quality and complex relationships	<u>Features</u> ➢ Adding data as facts and using expressions or rules ➢ Sensed data represent in First Order Predicate Logic ➢ Also represent the meta-propositional properties	<u>Features</u> ➢ Ontology based models for data representation ➢ Focus on interoperability and uniform format of data ➢ Store two types of knowledge, Domain and generic knowledge ➢ Ontological model using case-based reasoning ➢ Feedback system for verification and falsification
<u>Limitations</u> ➢ Needs to translate into usable format ➢ Limitation of privacy and distribution of data	<u>Limitations</u> ➢ Model does not support design and documentation of design process ➢ The sensed data cannot be used by software tool	<u>Limitations</u> ➢ Face challenges of abductive reasoning ➢ No logical reasoning on different hypothesis ➢ Facing challenges to exploring temporal and spatial inferences ➢ No scalability in design model

경희대학교 KYUNG HEE UNIVERSITY

Related Work

89

Tuple-Based Representation [8], [9]

Features

- Using key-value pair storage system
- Contains number of attributes and corresponding values as a single record
- System gave special emphasis to group activity and data access rights
- Data stores as tuple {ID, Subject, State, Time, Lag, Interval}

Limitations

- Limitation of capturing variety of data
- No Consistency checking
- No reasoning support

Hierarchical Representation [10]

Features

- Represent data in structure like tree
- Enhance the efficiency of searching data
- Stores data of Identity, Location, Status and Time

Limitations

- Static data model for storing specific type of data
- Cannot store the sensed data from environment directly

Domain-Focused Representation [11]

Features

- General data model to represent the facts
- Provides ease of querying, processing and adaptation of incomplete data
- Stored data in 4Ws (Who, What, Where, When)

Limitations

- Less expressiveness
- Less flexibility
- Needs to improve robustness and adaptability

경희대학교 KYUNG HEE UNIVERSITY

Related Work

90

Spatial Representation [12]

Features

- Stored the spatial data for Geographical Information System
- Five tiers of ontology was proposed for GIS
- handle the occurrence of exceptional situations in consistency constraints

Limitations

- It needs more efforts to gather the location data
- It is hard to keep the location data up to date

Hybrid Representation [13]

Features

- Integrate different existing representation model
- Integration of ontological, tuple based and logic based representation
- Due to ontological model, it has clear advantages of interoperability and heterogeneity

Limitations

- Inappropriate choices that arise from reliance on flawed data that reflect the user's requirements
- Inappropriate privacy
- Preferences exposed to users

Object Oriented Representation [14], [15]

Features

- Data stores as object oriented principles
- It followed object's modelling entities like People, Places and Objects
- Person, place, thing and patient are the main modelled entities

Limitations

- Limitation of interoperability
- Use third party system to overcome this limitation

경희대학교 KYUNG HEE UNIVERSITY

Comparison b/w Object Oriented and Ontological Model

91

Object Oriented Representation

- Pros
  - ➢ Data stores as object oriented principals
  - ➢ Allows "Real World" to be modeled more closely
  - ➢ High extensibility and scalability
  - ➢ Support for schema evolution
  - ➢ High Performance (Time)
- Cons
  - ➢ Lack of universal data model
  - ➢ Lack of standards
  - ➢ Highly complex to manage
  - ➢ Less expressive

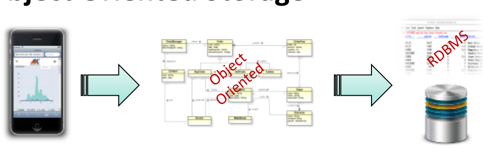
Ontological Representation

- Pros
  - ➢ Data stores as classes, attributes and individuals with semantics
  - ➢ Improve reusability and interoperability
  - ➢ More expressive
  - ➢ Improvement on searches
  - ➢ Permit inferences
  - ➢ High extensibility and scalability
- Cons
  - ➢ Ontology creation is difficult
  - ➢ Low performance (Time)
  - ➢ Challenges in abductive reasoning

Comparison b/w Object Oriented and Ontological Storage

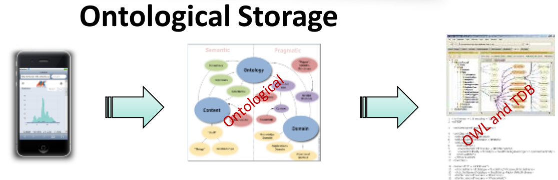
92

Object Oriented Storage



- Consists of DB Schema and DB Data
- Close world assumption (CWA)
  - Missing information treated as false
- Unique name assumption (UNA)
  - Each individual has a unique name
- Schema behaves as constraints on structure of data

Ontological Storage



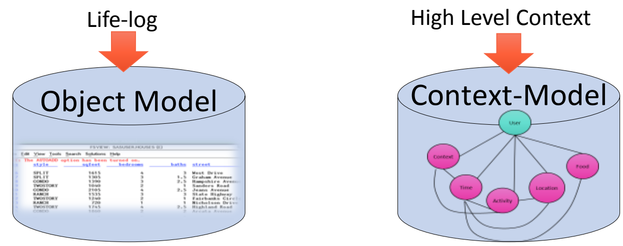
- Consists of axioms (Structure of model) and facts (Concrete Situation)
- Open world assumption (OWA)
  - Missing information treated as unknown
- No-Unique name assumption (No-UNA)
  - Individuals may have more than one name
- Ontology axioms behave like implications (inference rules)

Discussion: Trade off between Object model and ontological model

93

- It has decided that due to performance issue we will represent and store only the context information (High Level Context) in ontological format while the remaining whole Life-log will be represent and store in the object model format using relational databases.

Solution to performance issue



- There is trade off between performance of reasoning and scalability of heterogeneous source data.
- In ontological format the performance will be low and scalability will be high.
- In object model the performance will be high and scalability will be low.

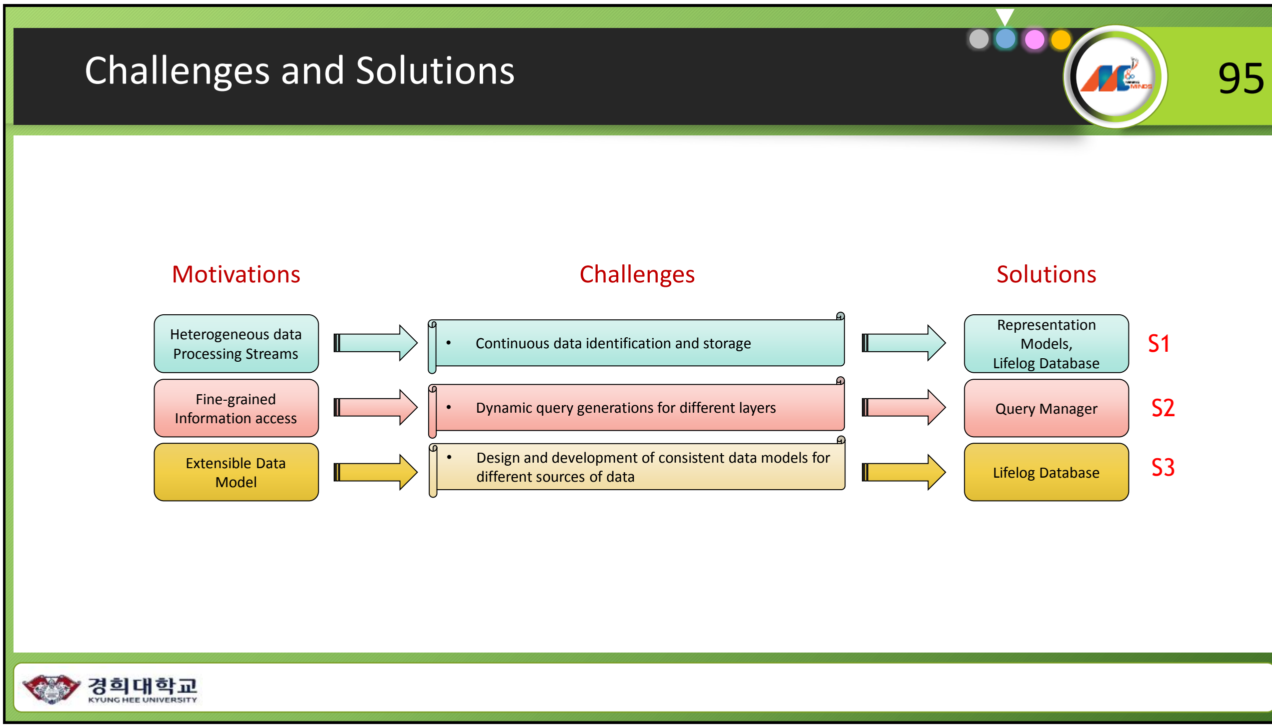
경희대학교 KYUNG HEE UNIVERSITY

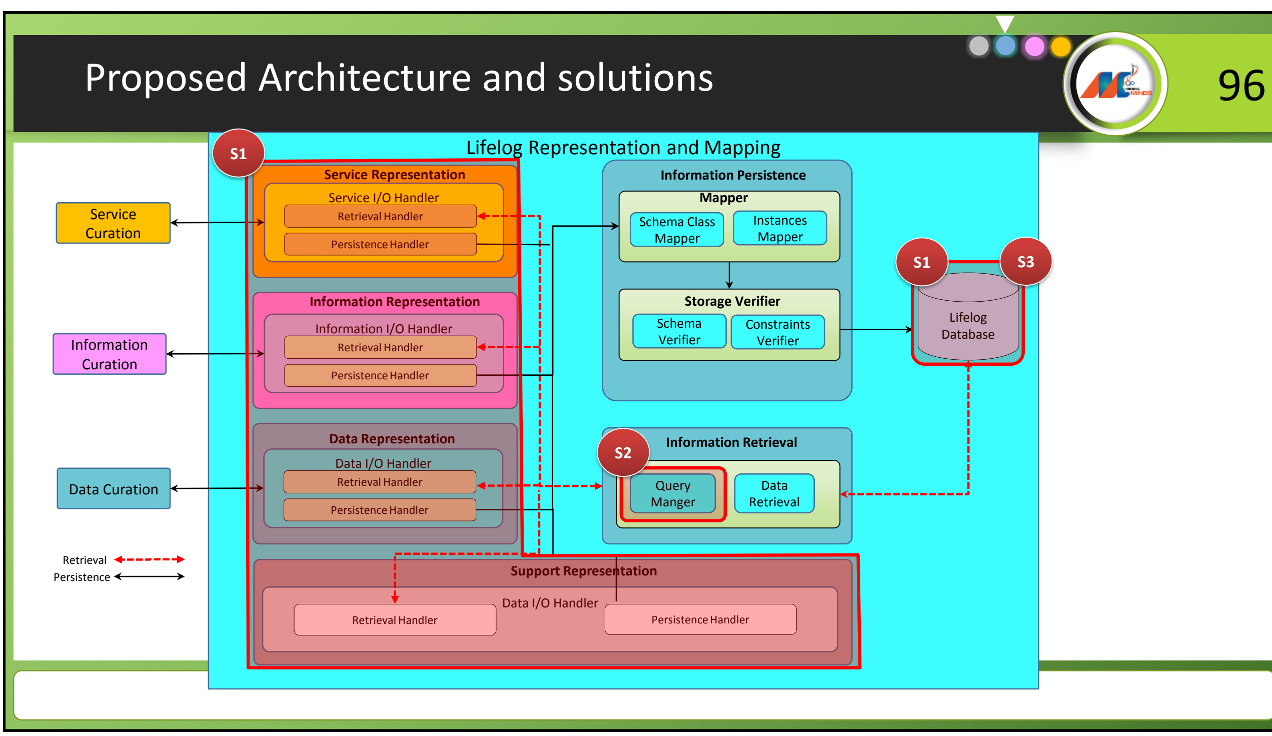
Motivation

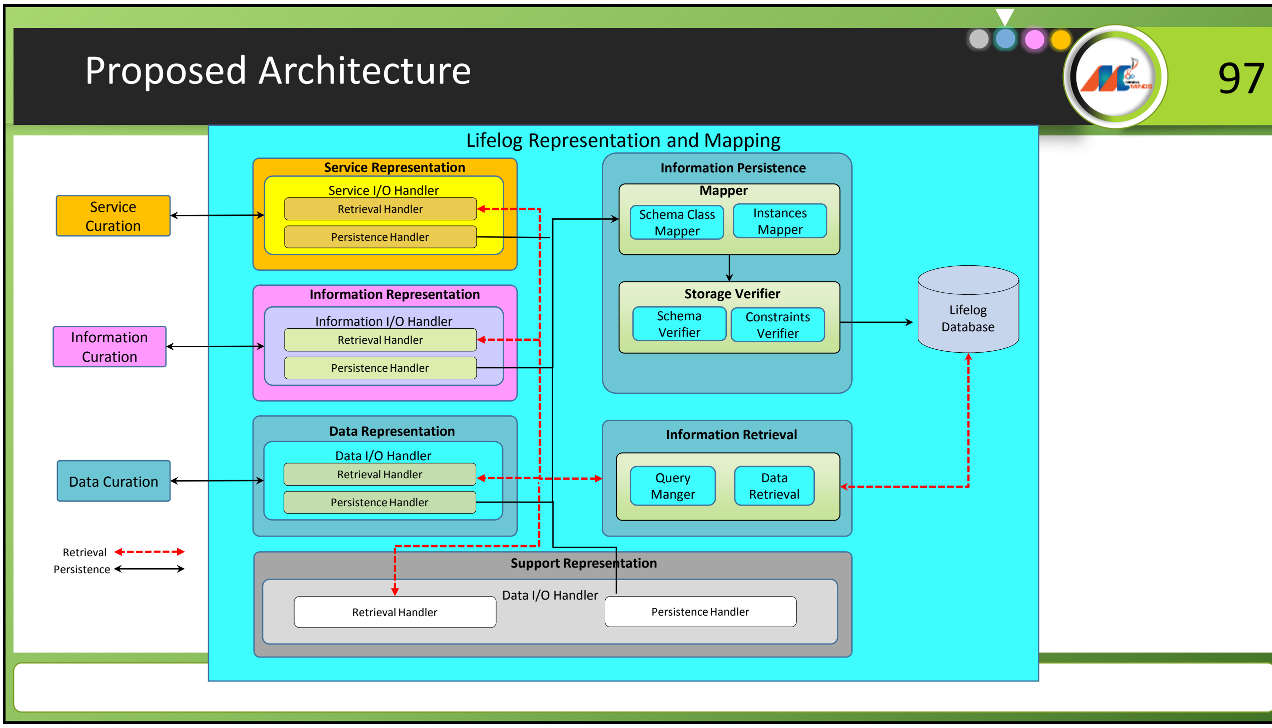
94

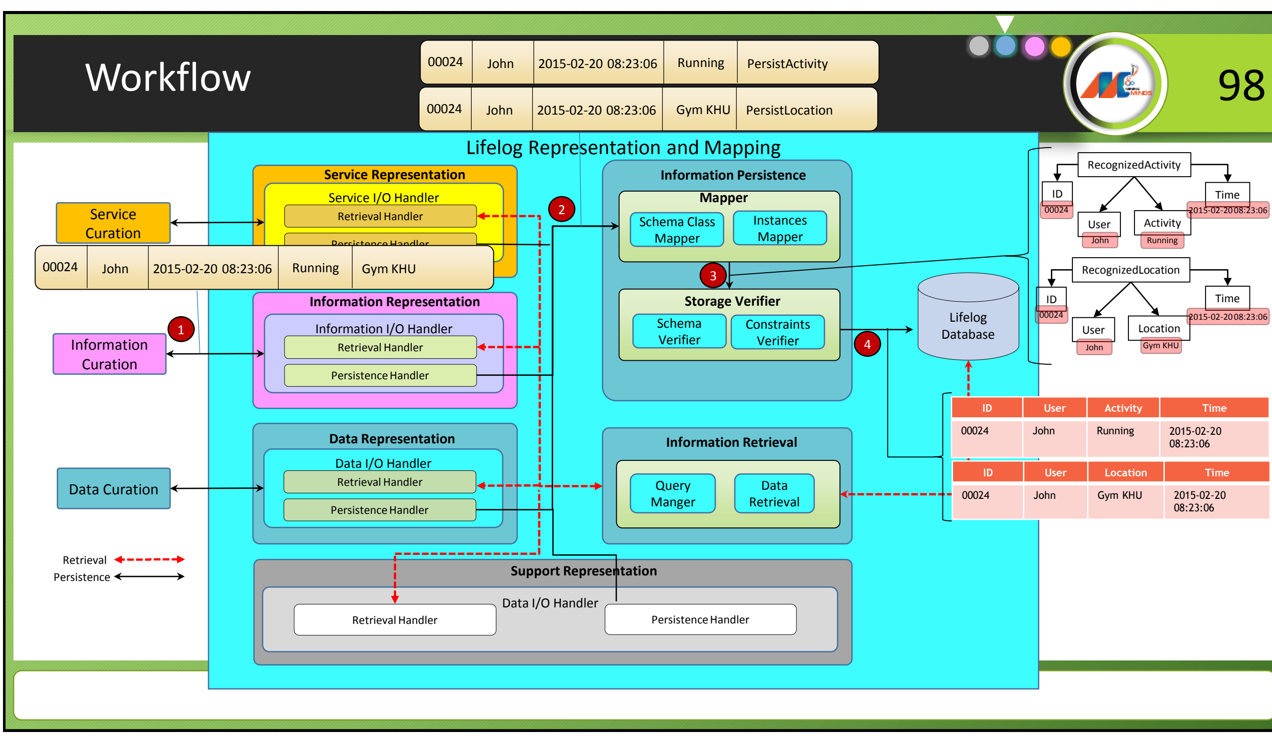
Heterogeneous data Processing Streams	Providing extensible and understandable structured format to variety of data.
Fine-grained Information access	Processable and procedural format to all layers and modules of MM Platform
Extensible Data Model	Integratable new models based on the different input data sources

경희대학교 KYUNG HEE UNIVERSITY


Challenges and Solutions
95
Motivations
Challenges
Solutions
Heterogeneous data Processing Streams
Continuous data identification and storage
Representation Models, Lifelog Database
S1
Fine-grained Information access
Dynamic query generations for different layers
Query Manager
S2
Extensible Data Model
Design and development of consistent data models for different sources of data
Lifelog Database
S3
경희대학교
KYUNG HEE UNIVERSITY

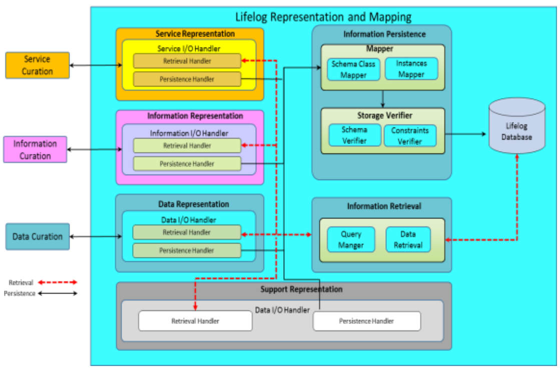

Proposed Architecture and solutions
96
Lifelog Representation and Mapping
S1
Service Representation
Service I/O Handler
Retrieval Handler
Persistence Handler
Information Representation
Information I/O Handler
Retrieval Handler
Persistence Handler
Data Representation
Data I/O Handler
Retrieval Handler
Persistence Handler
Support Representation
Data I/O Handler
Retrieval Handler
Persistence Handler
Information Persistence
Mapper
Schema Class Mapper
Instances Mapper
Storage Verifier
Schema Verifier
Constraints Verifier
S1
S3
Lifelog Database
S2
Information Retrieval
Query Manger
Data Retrieval
Service Curation
Information Curation
Data Curation
Retrieval
Persistence


Proposed Architecture
97
Lifelog Representation and Mapping
Service Curation
Information Curation
Data Curation
Retrieval
Persistence
Service Representation
Service I/O Handler
Retrieval Handler
Persistence Handler
Information Representation
Information I/O Handler
Retrieval Handler
Persistence Handler
Data Representation
Data I/O Handler
Retrieval Handler
Persistence Handler
Information Persistence
Mapper
Schema Class Mapper
Instances Mapper
Storage Verifier
Schema Verifier
Constraints Verifier
Lifelog Database
Information Retrieval
Query Manger
Data Retrieval
Support Representation
Data I/O Handler
Retrieval Handler
Persistence Handler

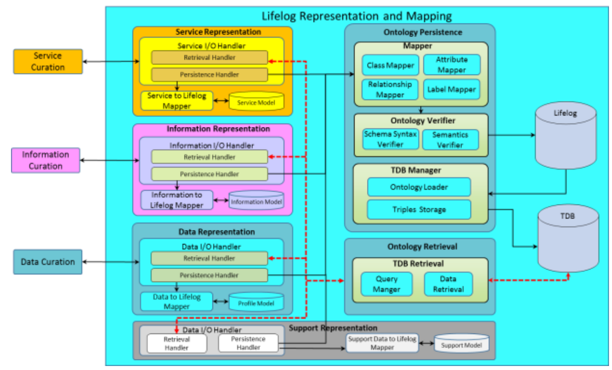

Workflow
98
00024 John 2015-02-20 08:23:06 Running PersistActivity
00024 John 2015-02-20 08:23:06 Gym KHU PersistLocation
Lifelog Representation and Mapping
Service Curation
00024 John 2015-02-20 08:23:06 Running Gym KHU
Information Curation
Data Curation
Retrieval
Persistence
Service Representation
Service I/O Handler
Retrieval Handler
Information Representation
Information I/O Handler
Retrieval Handler
Persistence Handler
Data Representation
Data I/O Handler
Retrieval Handler
Persistence Handler
Information Persistence
Mapper
Schema Class Mapper
Instances Mapper
Storage Verifier
Schema Verifier
Constraints Verifier
Lifelog Database
Information Retrieval
Query Manger
Data Retrieval
Support Representation
Data I/O Handler
Retrieval Handler
Persistence Handler
1
2
3
4
RecognizedActivity
ID
00024
Time
2015-02-2008:23:06
User
John
Activity
Running
RecognizedLocation
ID
00024
Time
2015-02-2008:23:06
User
John
Location
Gym KHU
ID User Activity Time
00024 John Running 2015-02-20 08:23:06
ID User Location Time
00024 John Gym KHU 2015-02-20 08:23:06

Objected Oriented vs Ontological Representation

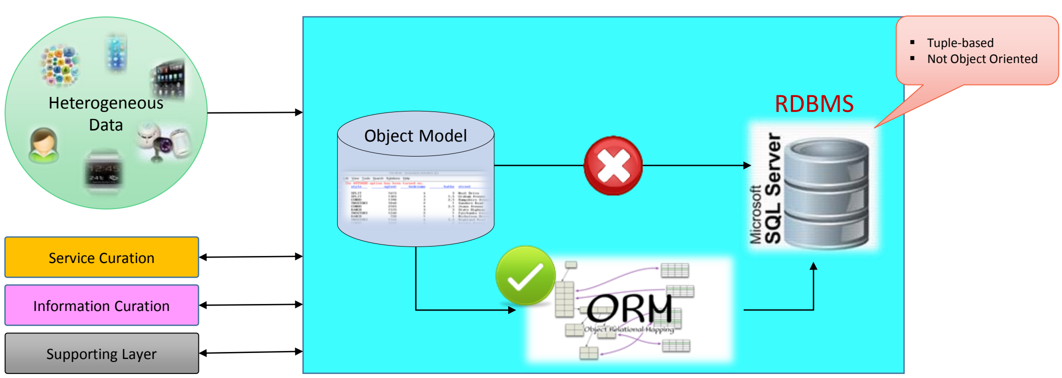
Object Oriented Representation



Ontological Representation

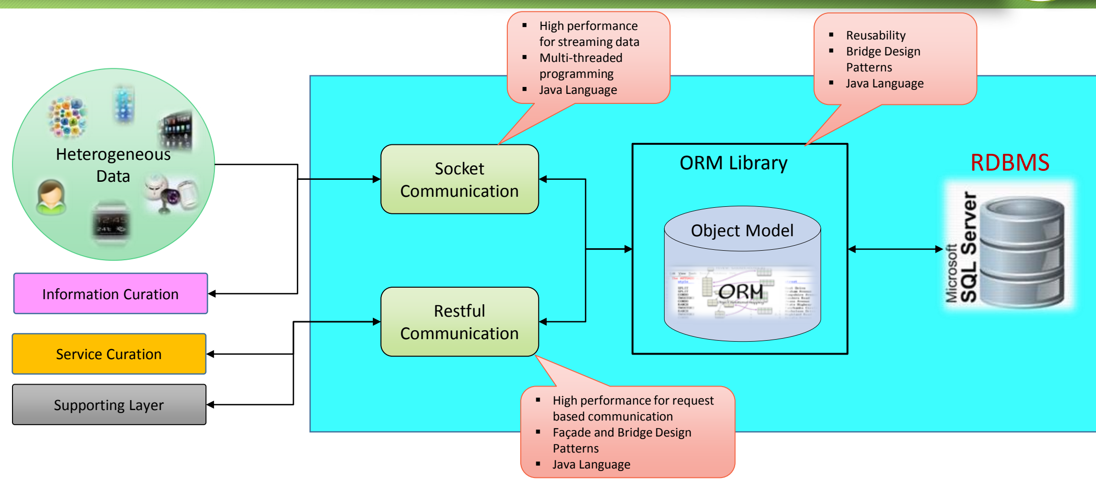


Implementation Details



Use of ORM (Object-Relational Mapping)

101



Object Oriented Design Patterns

102

Façade Design Pattern

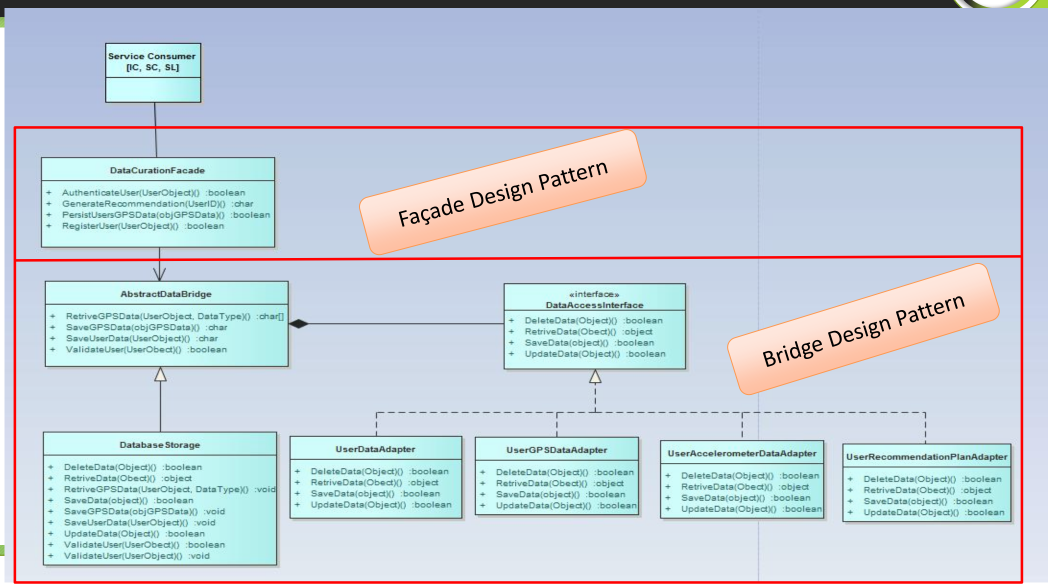
- Provides a unified interface to multiple inner interfaces
- Software Library easy to use, understand and test
- Library become more readable
- Reduce the dependencies of outer code on inner library classes
- Encapsulate the inner objects and methods of library
- Wrap a complicated sub systems with unified interface

Bridge Design Pattern

- Scalability for data storage
- Decouple the abstraction from its implementations
- Divide the responsibilities among different classes
- Hide details from client

Object Oriented Design Patterns

103



Big Data Storage

Usman Akhtar

Introduction

105

- To store unstructured data from heterogeneous sources in *real–time* is must for big data storage.
- This storage must be non–volatile and allow *CRUDS operations* on data.
- Big data storage should handle very large amounts of data and keep *scaling* to keep up with growth.

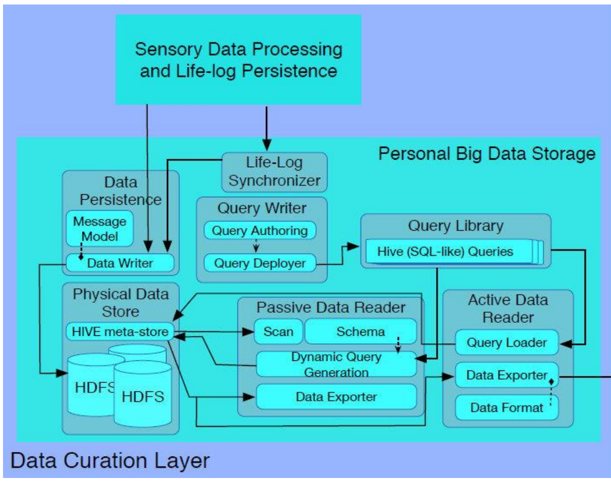
Motivation

106

- To store raw sensory, environmental variables in a large-scale non-volatile persistence (Big Data) with CRUD operations.
  - *Real-time data storage*
- For model training and rule generation, KCL requires interface to selected historic sensory data
  - *Passive data read operations*
- For Analytics and Visualization, SL required real-time data read based on provided parameters
  - *Active data read operations*

Big Data Storage - Tools

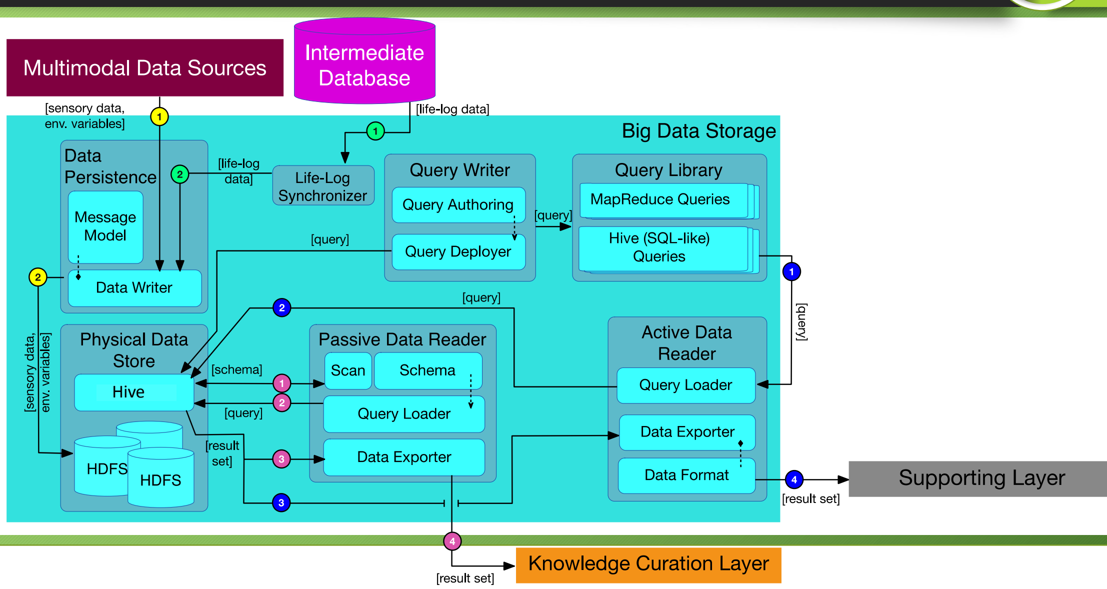
107



Tools and Technologies

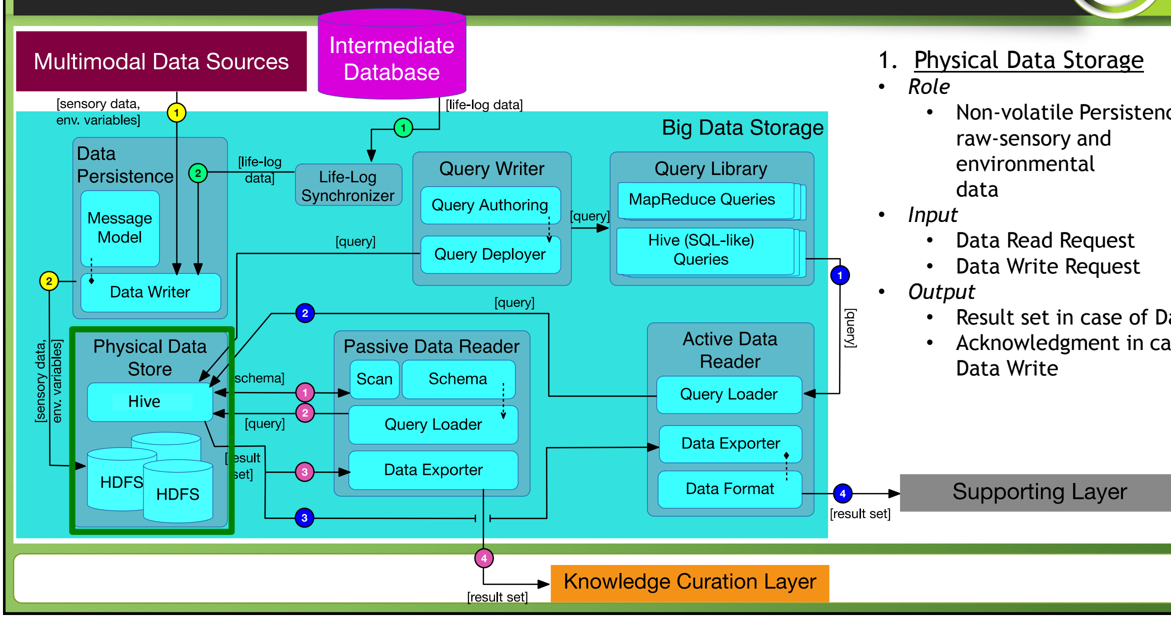
Functional Diagram

108



Big Data Storage Components

109

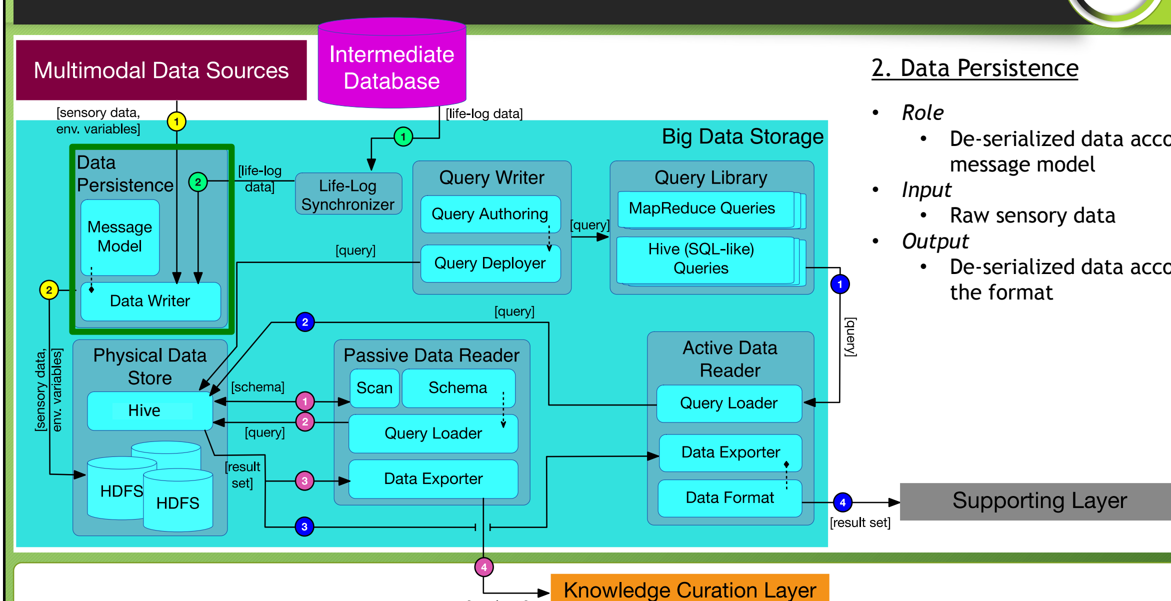


1. Physical Data Storage

- *Role*
  - Non-volatile Persistence to raw-sensory and environmental data
- *Input*
  - Data Read Request
  - Data Write Request
- *Output*
  - Result set in case of Data Read
  - Acknowledgment in case of Data Write

Big Data Storage Components

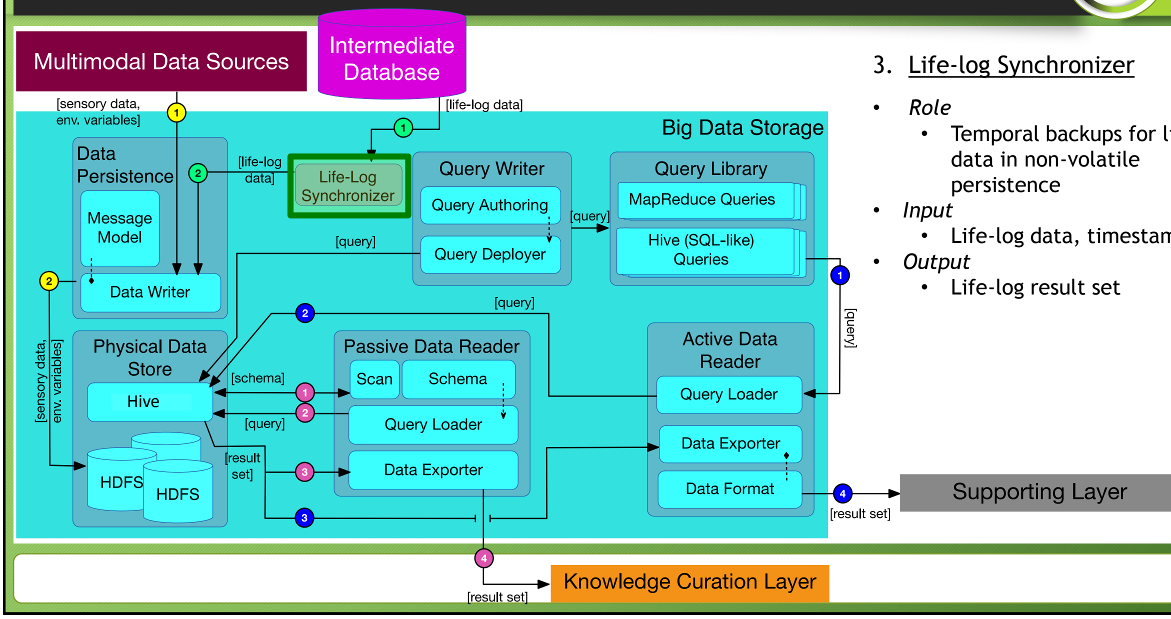
110



2. Data Persistence

- *Role*
  - De-serialized data according to message model
- *Input*
  - Raw sensory data
- *Output*
  - De-serialized data according to the format

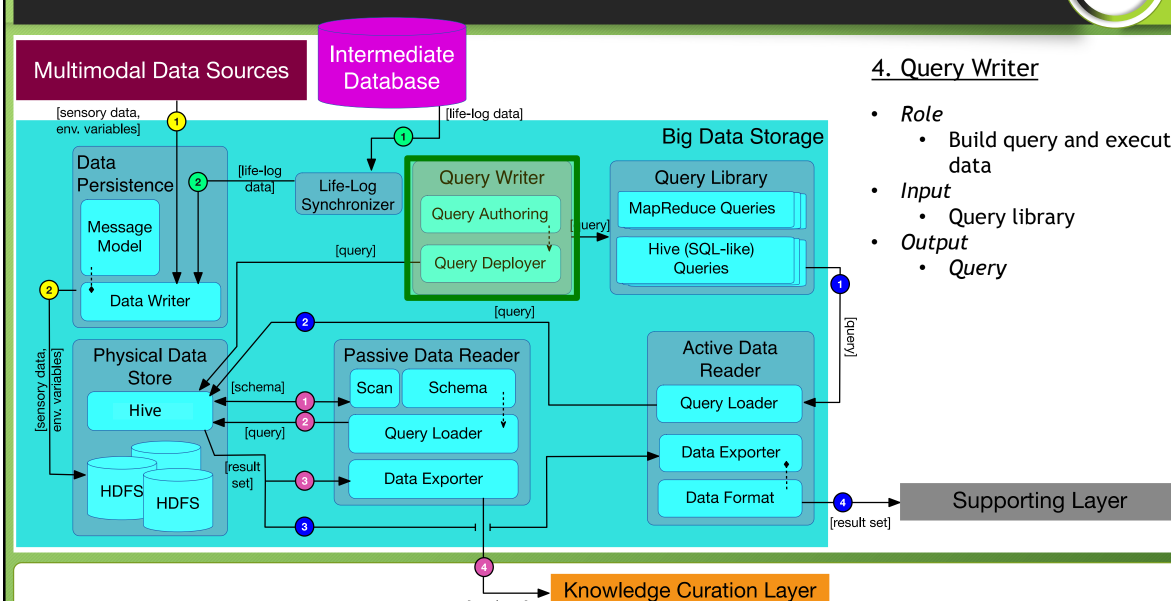
Big Data Storage Components



3. Life-log Synchronizer

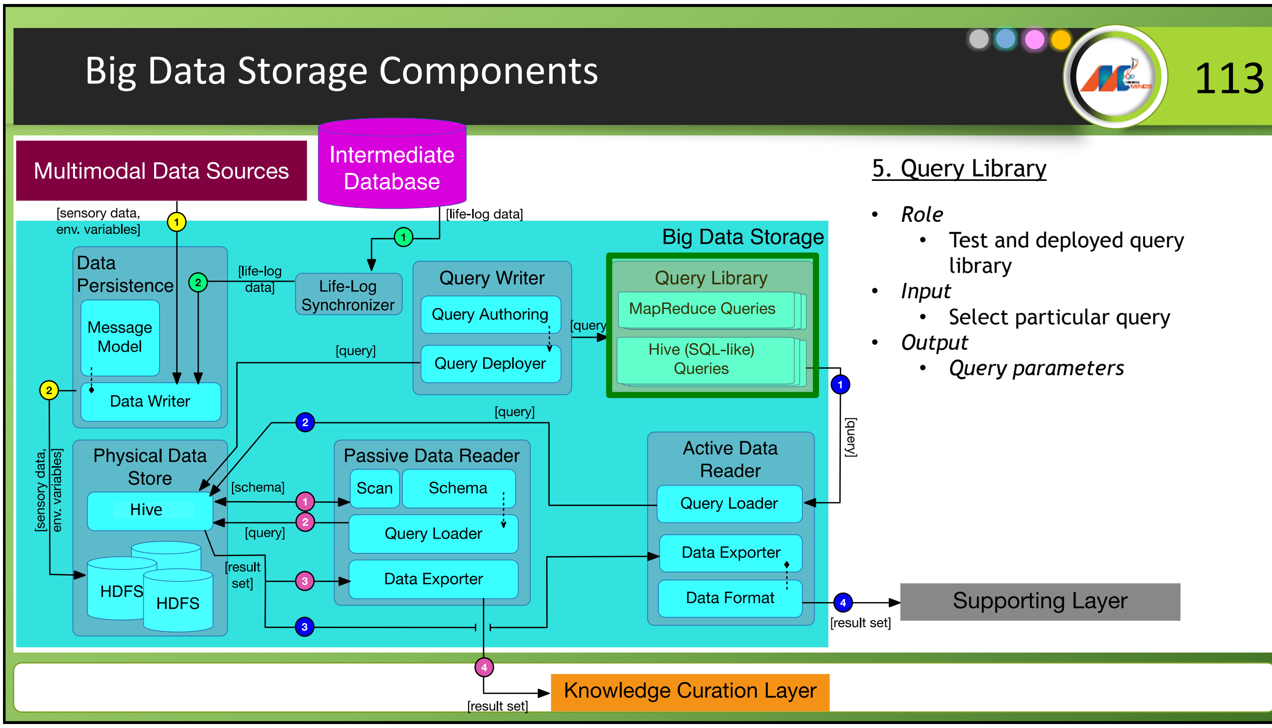
- *Role*
  - Temporal backups for life-log data in non-volatile persistence
- *Input*
  - Life-log data, timestamps etc.
- *Output*
  - Life-log result set

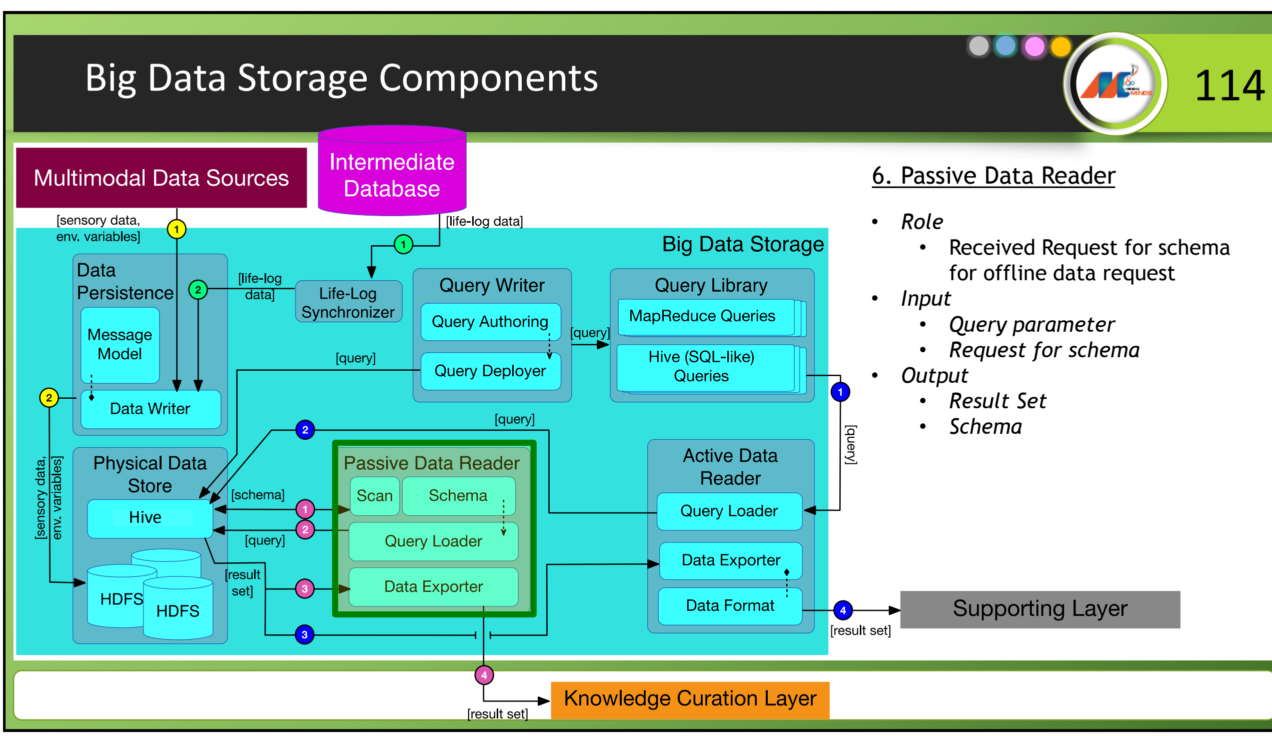
Big Data Storage Components



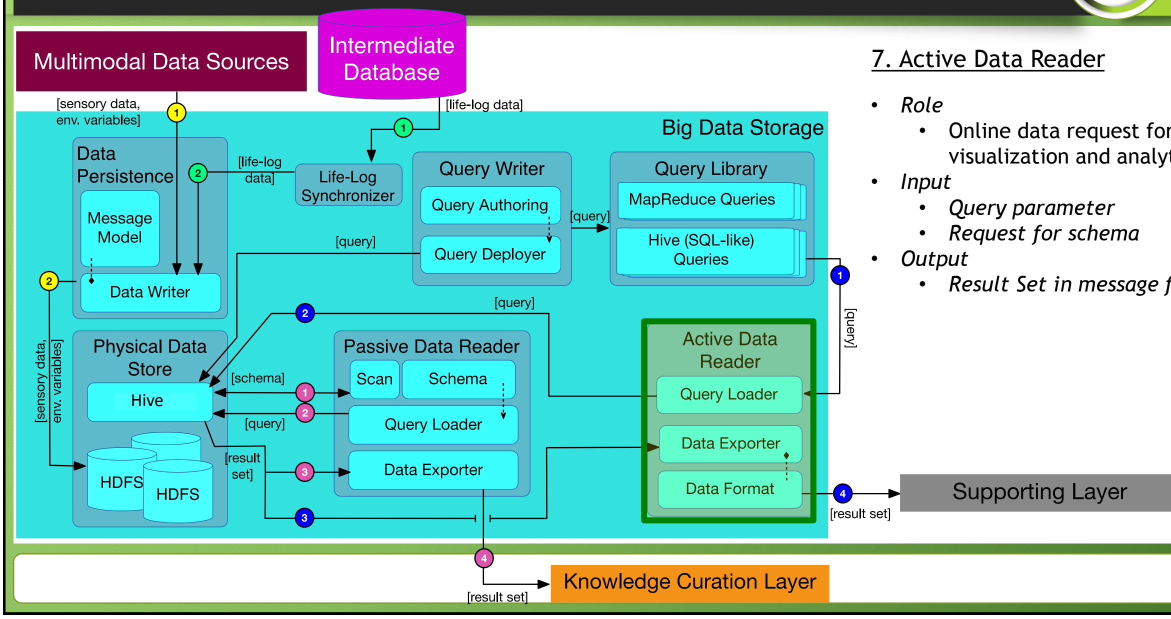
4. Query Writer

- *Role*
  - Build query and execute on data
- *Input*
  - Query library
- *Output*
  - *Query*


Big Data Storage Components
113
5. Query Library
• Role
• Test and deployed query library
• Input
• Select particular query
• Output
• Query parameters
Multimodal Data Sources
Intermediate Database
[sensory data, env. variables]
[life-log data]
Big Data Storage
Data Persistence
Message Model
Data Writer
Life-Log Synchronizer
Query Writer
Query Authoring
Query Deployer
Query Library
MapReduce Queries
Hive (SQL-like) Queries
[query]
[sensory data, env. variables]
Physical Data Store
Hive
HDFS
HDFS
[schema]
[result set]
Passive Data Reader
Scan
Schema
Query Loader
Data Exporter
Active Data Reader
Query Loader
Data Exporter
Data Format
Supporting Layer
[result set]
Knowledge Curation Layer


Big Data Storage Components
114
6. Passive Data Reader
• Role
• Received Request for schema for offline data request
• Input
• Query parameter
• Request for schema
• Output
• Result Set
• Schema
Multimodal Data Sources
Intermediate Database
[sensory data, env. variables]
[life-log data]
Big Data Storage
Data Persistence
Message Model
Data Writer
Life-Log Synchronizer
Query Writer
Query Authoring
Query Deployer
Query Library
MapReduce Queries
Hive (SQL-like) Queries
[query]
[sensory data, env. variables]
Physical Data Store
Hive
HDFS
HDFS
[schema]
[result set]
Passive Data Reader
Scan
Schema
Query Loader
Data Exporter
Active Data Reader
Query Loader
Data Exporter
Data Format
Supporting Layer
[result set]
Knowledge Curation Layer

Big Data Storage Components



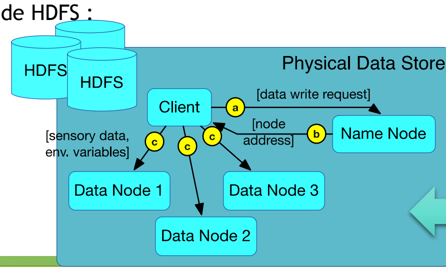
7. Active Data Reader

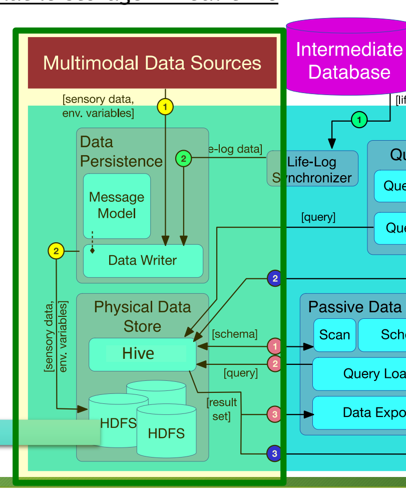
- Role
  - Online data request for data visualization and analytics
- *Input*
  - *Query parameter*
  - *Request for schema*
- *Output*
  - *Result Set in message format*

Execution Scenario #1

Writing Raw Sensory Data and environment variables into Non-volatile storage in real-time

1. Raw Sensory data is received by Data Writer of Data Persistence Component
   i. Data is de-serialized according to the Message Model
2. De-serialized message is sent to HDFS for persistence
3. Data gets written inside HDFS :





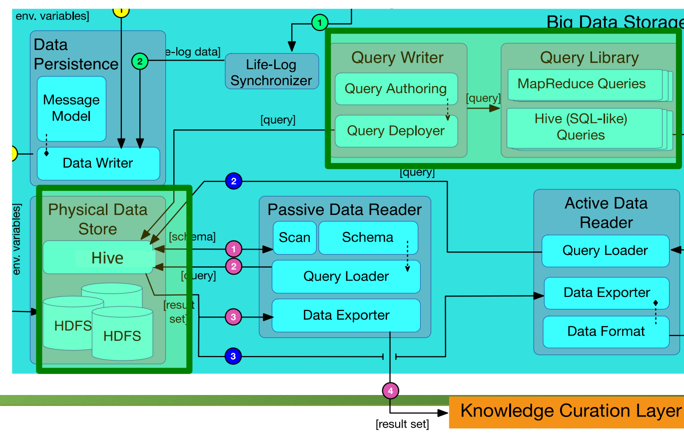
경희대학교
KYUNG HEE UNIVERSITY

Execution Scenario #2

117

Offline Query Authoring

1. Big Data Storage Requirements are converted into MapReduce or Hive Queries
2. Queries are written using Eclipse as an IDE
3. Query is tested and deployed as part of Query Library in Big Data Storage



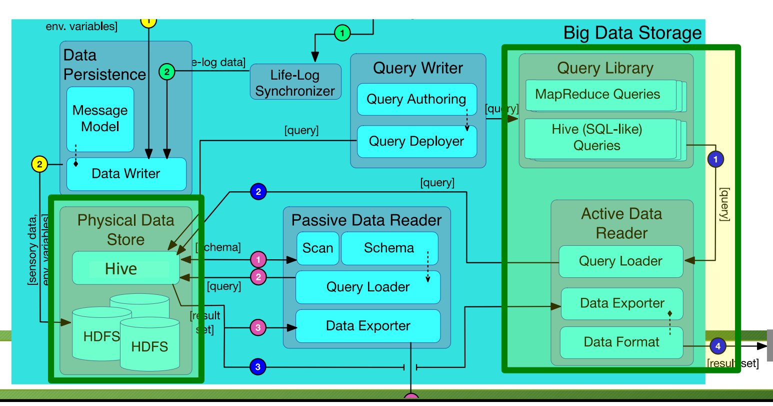
경희대학교 KYUNG HEE UNIVERSITY

Execution Scenario #3

118

Online Data Request for Data Visualization and Analytics

1. Data read request is generated by Visualization and Analytics Components of SL
   I. Query parameters are sent for query selection and execution
2. Active data reader selects the particular query depending upon the query parameters
3. Selected Query is sent to Hbase for execution
4. Required data is returned as a result set to Data exporter
5. Result-set is converted into data message per defined data format

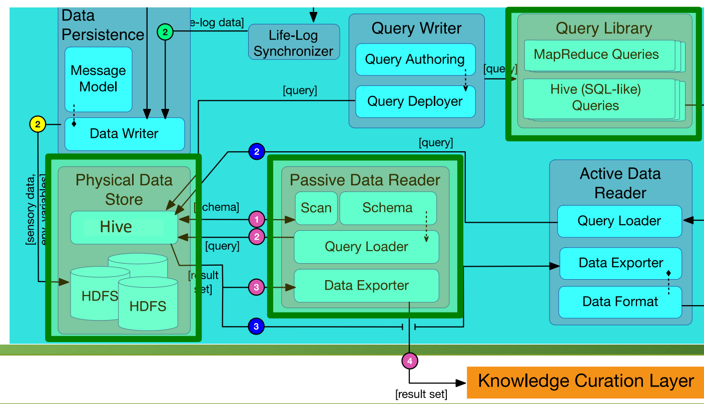


경희대학교 KYUNG HEE UNIVERSITY

Execution Scenario #4

Offline Data Request by KCL for Model Training and Rules Generation

1. Request for Schema of the persisted data is received by the Passive Data Reader
2. Scanned and most updated schema from Non-volatile storage is returned to KCL
3. KCL selects the parameters from the schema to generate a query and submits the request to Passive Data Reader
4. Passive Data Reader selects the query from Query Library
5. Selected Query is sent to Hbase for execution
6. Required data is returned as a result set to Data exporter
7. Result-set is converted into data message and returned to KCL



Contribution

- *Storage* of Heterogeneous data at *real-time*
- *Stream-based soft real-time* data read for *Analytics and Visualization*
- Schema-based query selection and execution over Big Data Storage
  - Availability to the most updated schema of persisted Data
- Temporal backups of Life-log data for non-volatile storage
- Able to build the big data ecosystem that facilitate request from the other layers.

Information Curation Layer (ICL 2.5)

Inertial Activity Recognition

Tae Ho Hur
Sang Beom Park

Introduction

123

- Life management system provides health related information and services to the user
  - User itself is the key factor of the system
  - System will collect the data related to the user
- User Context awareness is the fundamental part in this regard
  - Various valuable information can be acquired
    - Location, Situation, activity, etc.
  - Activity recognition is the cornerstone of context awareness
    - User context can be inferred based on user's activity information




Problem Statement

124

- Motivation
  - Inertial sensor based activity recognition has long been used, but works well only in limited condition with few designated activities.
  - Robustness and reliability must be settled within the range of diverse activities
- Goal
  - Create a model for recognizing user's activities in a highly accurate and robust manner
- Objectives
  - Accurately recognize several diverse and commonplace activities
  - Make two separate AR models for position dependency and independency
  - Achieve the robustness and reliability based on fusion technique


Related Works

125

Authors	Published year	Sensor placement	Sensor type	Techniques	Limitation
Chun zhu et al. [1]	2009	Waist, Foot	ACC, Gyro	• Two feed-forward neural networks fusion • Heuristic segmentation	• Few basic activities • Offline evaluation
Jun-Ki Min et al. [3]	2011	Head, two arms and two wrists, fingers	ACC, Gyro, skin temperature, heat flux, galvanic skin	• Dynamic feature selection • Outputs of classifiers are combined and compared	• Device is to bulky to use in real life
Lei Gao et al. [4]	2011	Waist, chest, thigh, side	ACC	• Considered the difference of sensor orientation change using estimate of constant gravity vector • Sensor fault is considered	• Only used ACC
Ming Zeng et al. [2]	2014	Free	ACC, GPS, Speed, Ambient light	• Build separate models for each activity • Feature transformation	• Heavy weighted system
Muhammad Shoaib et al. [5]	2014	Upper arm, wrist, waist, two pockets on pants	ACC, Gyro	• Considered orientation independency • Compared the difference of sensor types and feature sets	• Few basic activities

Challenges

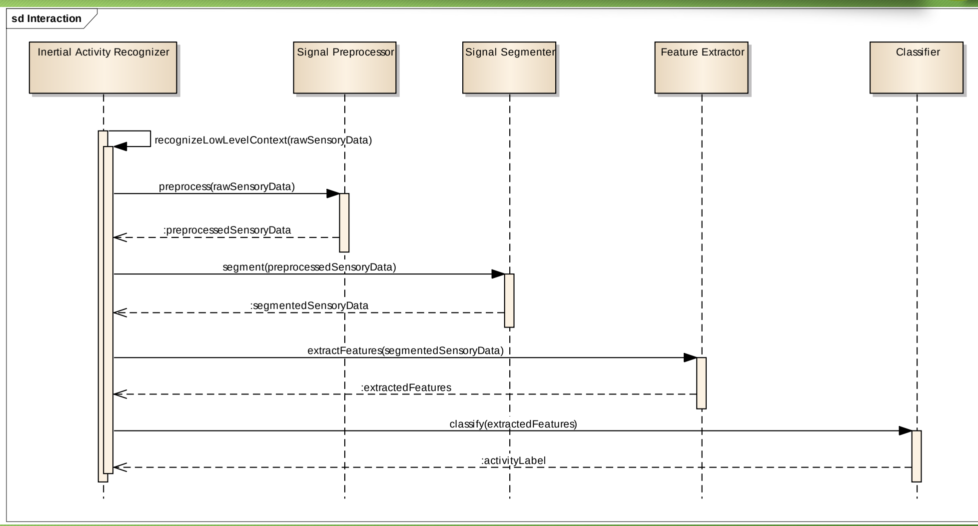
126

- Position dependency
  - Some of the sensor devices are attached in fixed position while some doesn't such as a smartphone inside a pocket.
  - Satisfying these two different characteristics into one AR model is hard to accomplish

- Achieve Reliability
  - AR model is usually made based on limited condition which does not show expected performance in the real environment
- Achieve Robustness
  - In multiple sensor based AR, It does not work fine in a single AR model based recognizer using all the sensor values at the same time if one of the sensor does not work properly

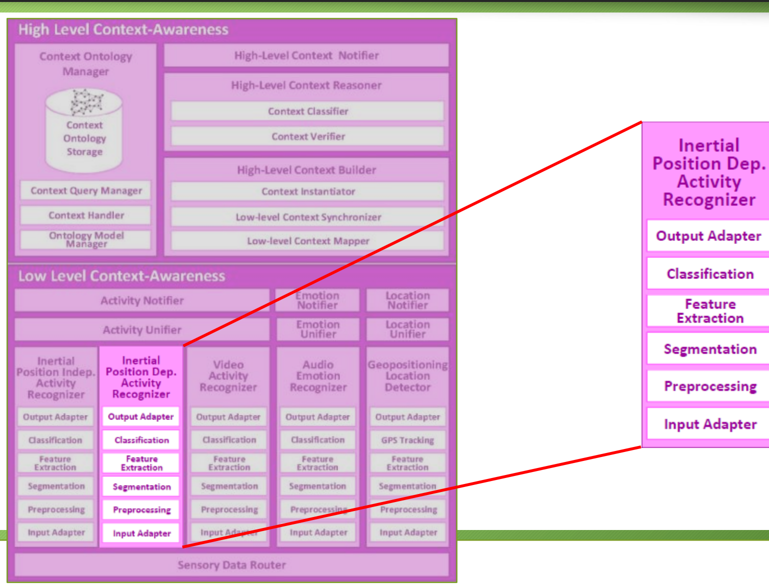
Sequence Diagram for inertial sensor based AR

127



Architecture in Mining Minds

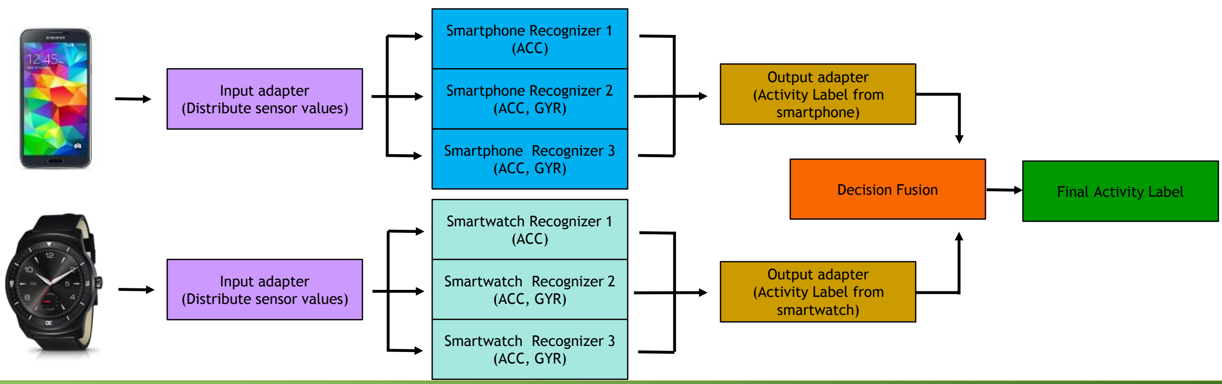
128



Proposed method

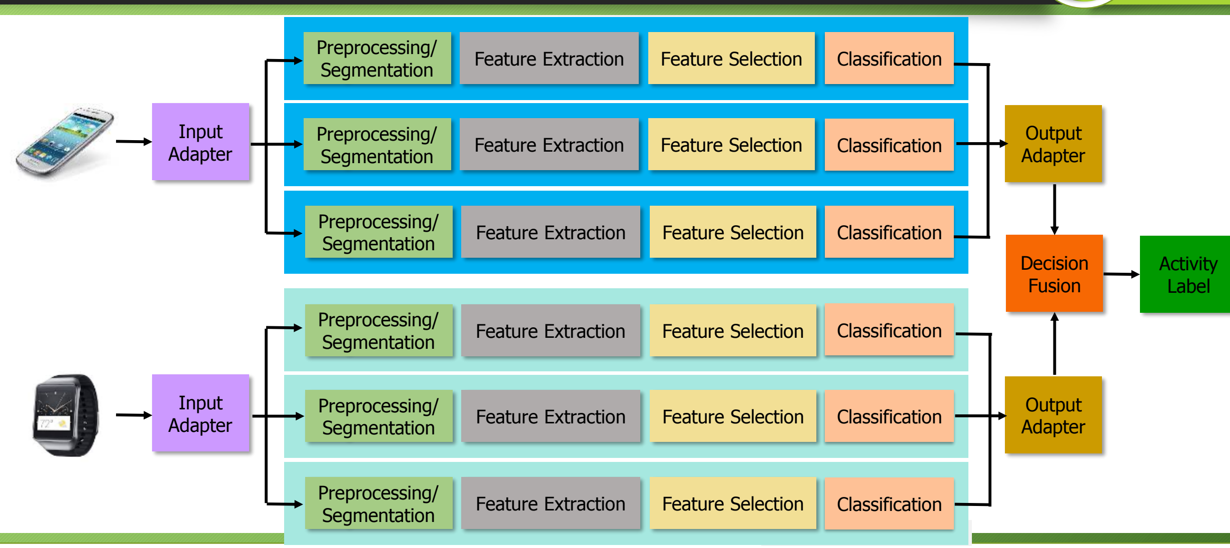
129

- Use Multiple Activity Recognizer
  - Solve the position problem
  - Strong on sensor failure
- Final decision will be made by decision fusion
  - Achieve high accuracy with the highest possibility



Flow chart (Abstract)

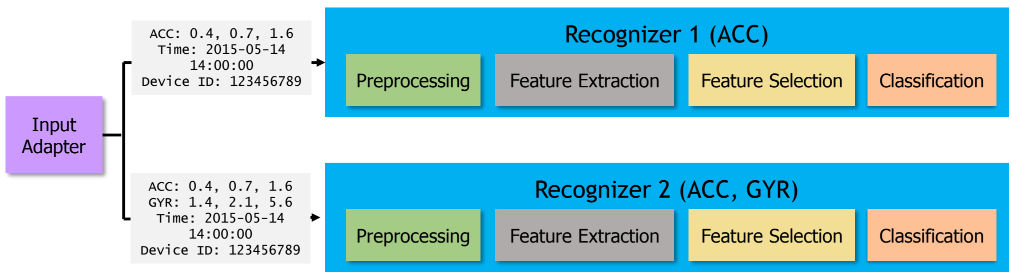
130



Input Adapter

131

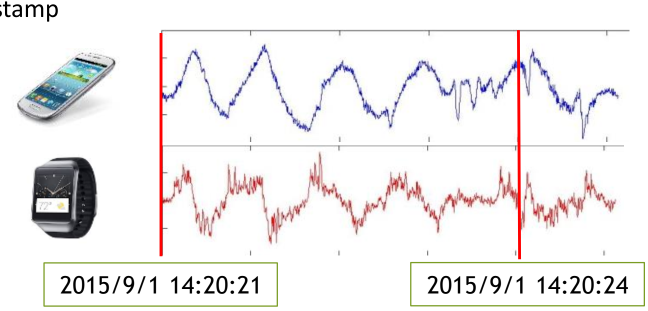
- Distribute sensor values to each recognizer



Preprocessing & Segmentation

132

- Data synchronization
  - Based on timestamp



- Segmentation
  - Sliding window based segmentation
  - 3 seconds window
  - Half-Overlapping

Feature Extraction

133

- Each devices are specialized to recognize the activities. Smartwatch is specialized to recognize movement of user's arm(e.g. direction of arm, movement of arm). And Smartphone is specialized to recognize user's entire movement(e.g. walking, running, etc.).
- Integrated model is made for higher accuracy. It amends the misclassified activities.
- The number of feature of device model is 42, and the number of integrated model is 49. The feature which is used at each model is the most widely used in time domain feature.

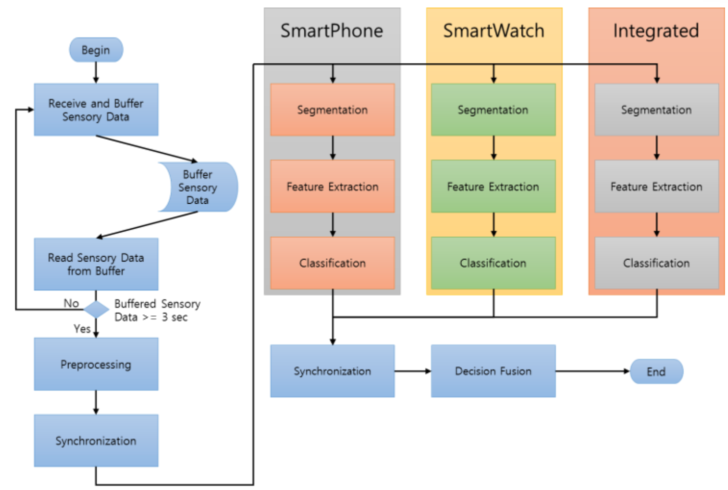
Classification

134

- Combine different classifiers
  - To improve the performance of individual classifiers by combining their output using different techniques
- Decision Fusion
  - Decide final activity with Weighted voting Method.
  - The reason why we use Weighted Voting is WV is the most widely used method for decision making system and it has advantage in correction of misdecided activity.

Flowchart

135



Devices and Tools

136

- Devices
  - Galaxy S5
  - G Watch R
- Tools
  - WEKA
  - Android Studio
- Language
  - Java

WEKA The University of Waikato

Android Studio

Powered by IntelliJ Platform

Experiment Result

137

- Experiment Result(Confusion Matrix)

	Eating	Running	Sitting	Standing	Walking	Stretching	Sweeping	Lyingdown
Eating	0.92	0.01	0.06	0	0	0	0	0
Running	0	0.96	0	0	0.03	0	0	0
Sitting	0.02	0	0.89	0.03	0	0	0	0.05
Standing	0	0	0.01	0.97	0	0.02	0	0
Walking	0	0	0.01	0.01	0.97	0.01	0.01	0
Stretching	0	0	0.01	0.02	0.02	0.88	0.07	0
Sweeping	0	0	0.02	0.06	0.12	0.26	0.54	0
Lyingdown	0	0	0.5	0	0	0	0	0.5

Conclusion & Future work

138

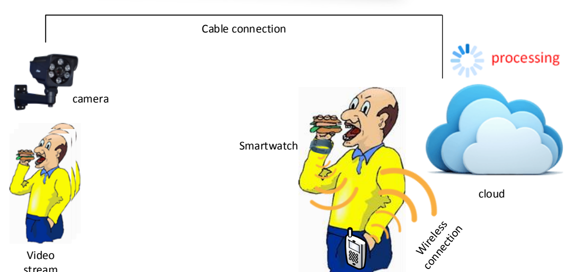
- Conclusion
  - MM v1.0 recognized 5 basic activities with smartphone
    - ✓ High recognition rate but few activities recognized
  - MM v1.5 recognized 8 daily activities with smartphone & smartwatch
    - ✓ Increased types of activities with additional sensor
    - ✓ Weak on robustness & reliability
  - MM v2.0 recognized more activities with higher accuracy
    - ✓ Use of Multiple Activity Recognizer will guarantee robustness and reliability
    - ✓ Decision fusion on inertial sensor and video increased the accuracy
  - MM v2.5 reduced the number of activities but focused on different eating styles
    - ✓ Food recognition is added on MM v2.5 which requires the recognition of eating to be improved
    - ✓ While recognizing eating had poorest accuracy among all, recognizing different styles of eating increased the accuracy of eating
- Future work
  - Multiple activity recognition
    - ✓ As more than one kind of sensor devices are used, one activity can be combined into different kind of motions
      - Ex) Walking while eating
    - ✓ Different kind of sensor/device (Inertial/video, smartphone/smartwatch/wearable) based multiple activity recognition will be performed

Video based Activity Recognition

Thien Huynh-The

Introduction

- Activity recognition module is one of the most important components in ICL layer and MM platform.
- Provide much expensive information about activities to deeply understand user behaviors.
- Activity recognition can be performed on data collected from different sources:
  - Wearable device
  - Camera device



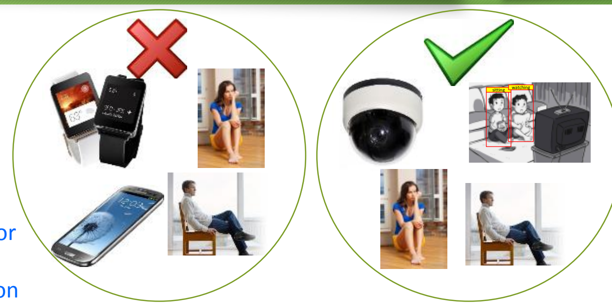
Motivation

141

ICL ver 1.5

- Collect physical sensor data to detect popular dynamic activities
- Poor accuracy with activities having different postures
- Impossible for posture detection

Why camera for activity recognition ?



- Activities are detected and recognized based on human posture.
- Deeply visual understand the posture of activities
- Provide some information of the scenes.

Goal & Objectives

142

- **Goal**: Improvement of the offline and online activity recognize module

- **Objectives**
  - Investigate the algorithm with different parameter configurations
  - Implement the recognizer with the highest appropriated parameter pattern
  - Improve the online recognition accuracy from 80% to 85%.

Related Works

Authors	Year	Dimension	No. Activity	Activity types	Accuracy	Key points	Limitation
Dollar	2005	2D	6	Single action	80%	Interest point features	Low classification accuracy
Wang	2010	2D	4	Daily activity	90%	Pose estimation	Highly computational cost Limited in the number of activities
Chen	2012	2D	6	Single action	94%	Hybrid interest point detection	Highly computational cost
Liu	2013	2D	8	Single action	95%	Key pose estimation	High complexity
Cai	2014	2D	8	Single action	96%	Pose dictionary learning	Depend on the pose dictionary
Liu	2014	2D	8	Single action	93%	Hybrid interest point detection	High complexity and computational cost

- Single action: walking, jogging, running, hand waving…
- Daily activity: answering phone, open refrigerator…

Related Works

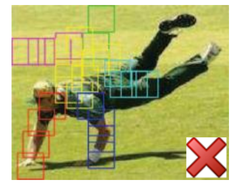
Authors	Year	Dimension	No. Activity	Activity types	Accuracy	Key points	Limitation
Gu	2010	3D	8	Single action	94%	Hidden Markov Mode	High complexity
Ofli	2013	3D	12	Single action	80%	Annotated joint feature	Low recognition rate
Vantigodi	2013	3D	12	Single action	96%	Temporal joint distance feature	
Kruthiventi	2014	3D	12	Single action	97%	Dynamic time warping	
Wang	2014	3D	12	Single action	95%	Actionlet ensemble model	High complexity and computational cost

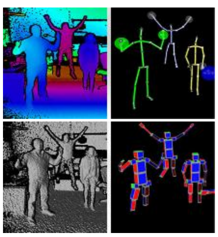
- Single action: hand catching, forward punching, two hand waving, ward kicking, high throwing …

Limitation of Existing Works

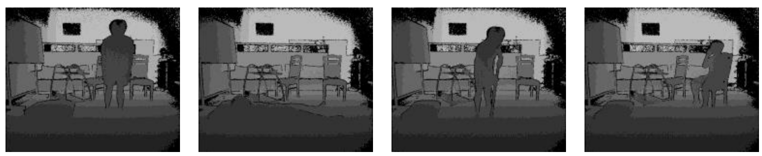
145

- Using RGB color videos
  - Non-robust in the dynamic scenes
  - Body-component occlusion issue
  - Human pose estimation
  - Pose tracking in high-speed motion.
  - Highly computational cost





→ Consider depth camera for video-based activity recognition in MM 2.0



MS Kinect device

146

Why Kinect ?

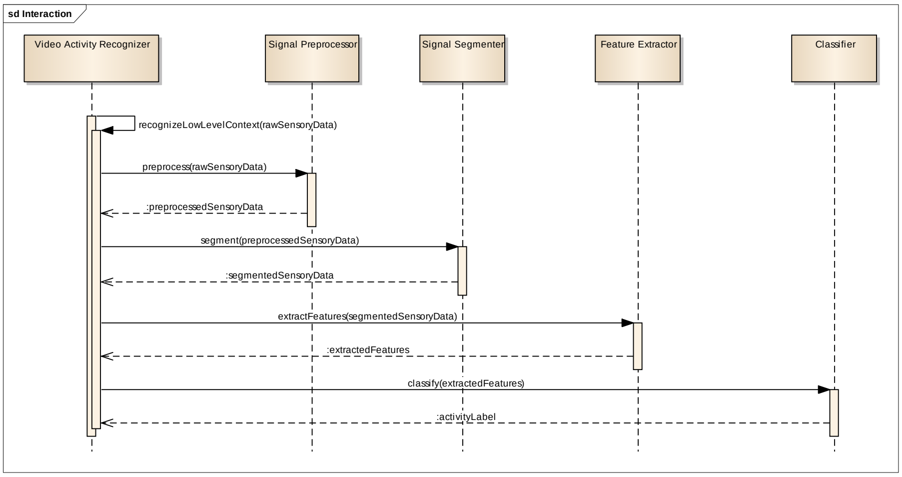
- Kinect is a line of motion sensing input devices by Microsoft for Xbox One video game consoles and Windows PCs
- Provide depth frame channel besides RGB channel
- Provide body frame with output is 3D position of each body joint
- Kinect is receiving strong support by Microsoft and Community to be easy publish apps on Windows Store.
- Has a wide range of potential applications: video gaming, education, medical, and so on

Limitation

- Only set up in the indoor environment
- Require high requirement hardware

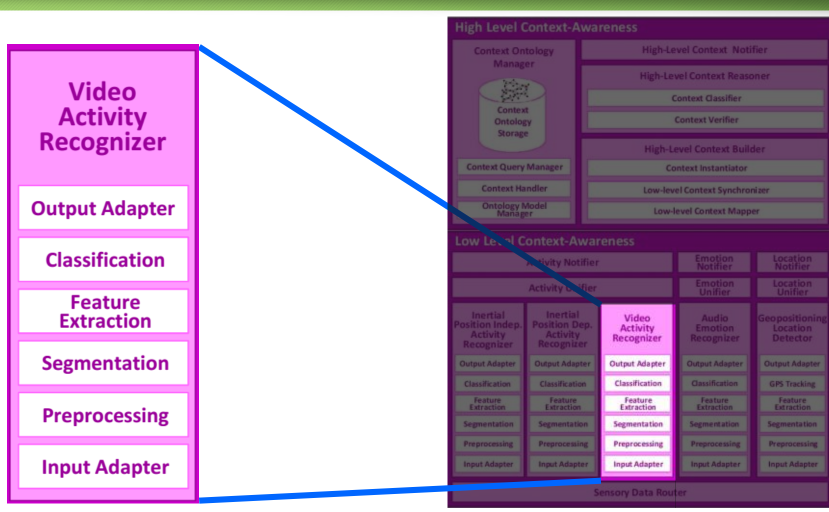
Sequence diagram for VAR

147



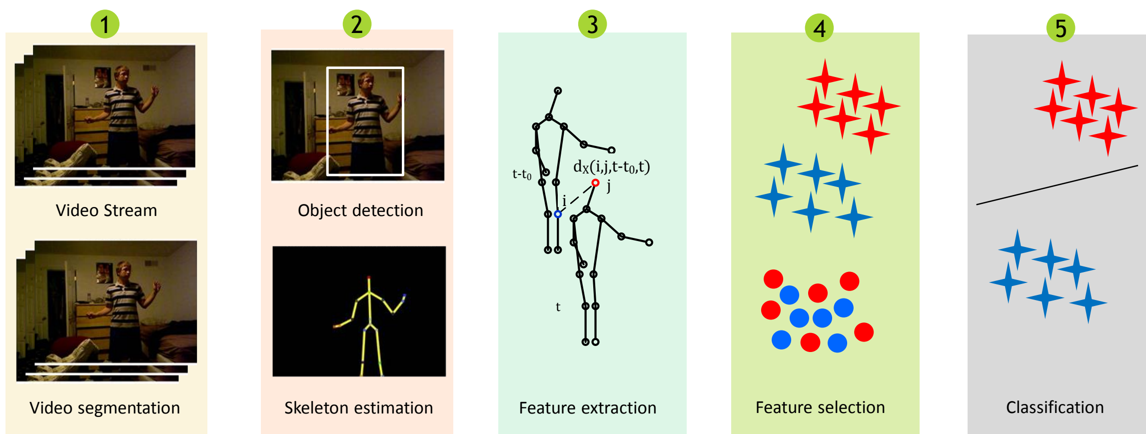
Architecture –ICL 2.5

148



Work Flow

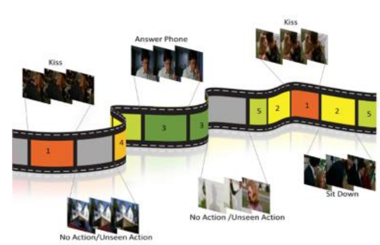
149

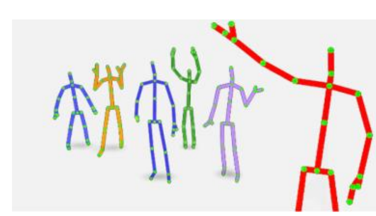


Video segmentation

150

- Skeleton data provided by Kinect device for activity recognition:
  - 25-joint skeleton
  - Each joint includes 3 values of x, y, and z
  - 30 frames per second
- Segment the body frame stream from DCL into short windows with length of 3s (=90 frames)
- Recognize activities corresponding to the data of each window





Preprocessing

151

- The output from Microsoft Kinect device to DCL will be:
  - Body joint coordinate (25 joints with 3 values xyz of each joint)

→ Do not use color stream because of privacy issue

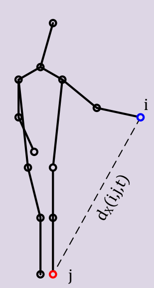
- Object Detection and Skeleton Estimation are performed by using Microsoft API functions

KINECT FOR DEVELOPERS

Feature extraction

152

Spatial Joint Distance Feature



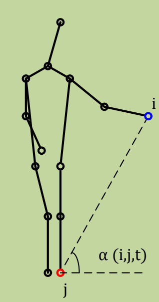
Spatial Joint Distance Feature is measure Euclidean distance between 2 joints of an object in the current frame.

$$d(i,j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Spatial Joint Angle Feature is measure angle between 2 vector $\overrightarrow{ij}$ and $\overrightarrow{Ox}$.

$$\alpha(i,j) = \tan^{-1}\left(\frac{|y_i - y_j|}{|x_i - x_j|}\right)$$

Spatial Joint Angle Feature



Feature selection

153

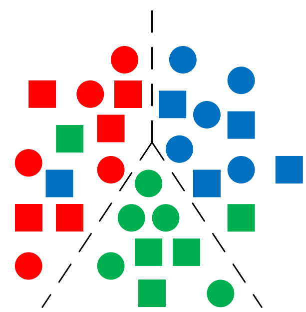
Why we need feature selection ?

- Select the highest discriminative features
- Reduce the computational cost in feature extraction and classification

Limitation: Accuracy will be reduced based on the number of used features

Feature ranking

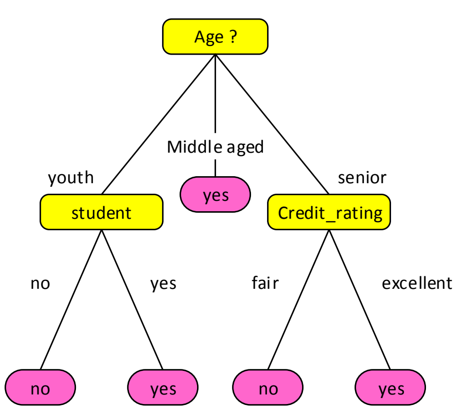
- Rank the features by class reparability criteria
- Use an independent evaluation criterion for binary classification.
- Absolute value two-sample t-test with pooled variance estimate.



Square or circle is better ?

Decision tree

154

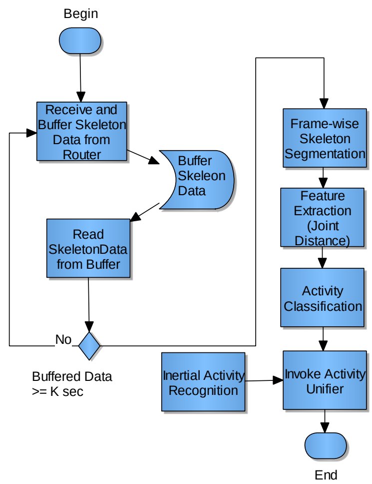


Why Decision Tree ?

- Relatively fast compared to other classification models
- Obtain similar and sometimes better accuracy compared to other models
- Simple and easy to understand
- Can be converted into simple and easy to understand classification rules

Flowchart

155



경희대학교
KYUNG HEE UNIVERSITY

Tools/Technologies

156

Windows 8

Java

ORACLE

MATLAB & SIMULINK

Dataset collection

157

Scenario:

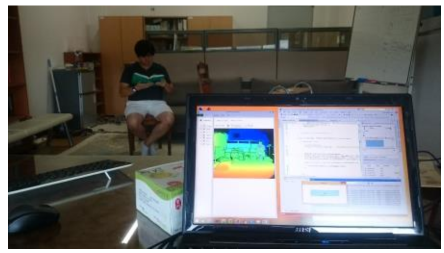
- Collect daily indoor activities: stretching, sweeping, sitting, lying, standing, and eating.
- Number of collected candidates: 10 (age: 21-30, height: 1m65-1m75)

Device:

- Laptop (Windows 8.1, USB 3.0)
- MS Kinect device

Output data:

- Body frame: 30 frames/second



Evaluation results

158

- Benchmark the algorithm with different types and number of feature
  - Number of feature
  - Frame rate
- The parameter configuration is set up:
  - Frame rate: 10 frame per second (30 body frames for each window)
  - Number of feature: 70 joint distance and angle
- Achieve over 90% of accuracy for offline recognition and over 84% of accuracy for online recognition

	Stretching	Lying	Sweeping	Sitting	Eating	Standing
Stretching	**0.86**	0.00	0.06	0.00	0.02	0.06
Lying	0.02	**0.86**	0.00	0.06	0.00	0.06
Sweeping	0.12	0.00	**0.84**	0.00	0.00	0.04
Sitting	0.04	0.04	0.00	**0.88**	0.04	0.00
Eating	0.00	0.04	0.02	0.12	**0.82**	0.00
Standing	0.08	0.08	0.02	0.02	0.00	**0.80**

Classification result using decision tree on the testing dataset

Contribution

159

- Collect a new dataset for evaluating the video-based activity recognition
- Develop a method for video-based activity recognition
- Investigate the proposed method on Matlab with different parameters.
- Prove a method to be efficient with a high performance in accuracy and computational cost

Future plan

- Video-based activity recognition demo
- Activity recognition demo
- Activity recognition video demo preparation

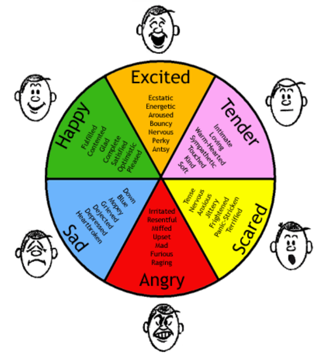
Audio-based Emotion Recognition

Le Ba Vui

Introduction

161

- Emotion is a person’s state of mind and instinctive responses
- Representation of emotion
  - Emotional speech
  - Facial expression
  - Physiological signals
- Speech is the most commonly used and natural method of human communication
- Affected by various factors: language, gender, age, etc.



Motivation

162

- Automatic understanding emotional speech will help smart systems to interact with user in emotional ways.
  - Enrich the content of personality
  - Provide appropriate recommendation depend on the realistic situations
  - Help psychologists to understand users' mental states

Goal and Objectives

163

- **Goal**: Design and implement a methodology that is able to recognize emotional states from users' speech
- **Objectives:**
  - Achieve acceptable accuracy
  - Flexible to apply on different emotions
  - Can be easily integrated in different environments: call center, personal computer, smartphone, …
  - Collect a real-world and diverse dataset

경희대학교 KYUNG HEE UNIVERSITY

Related Works – Datasets

164

Database	Modalities	Elicitation Method	Emotional Content	Size
AIBO database (Batliner et al., 2004) [1]	Audio	Natural: children interaction with robot	anger, bored, emphatic, helpless, ironic, joyful, motherese, reprimanding, rest, surprise, touchy	110 dialogues, 29200 words
Berlin Database (Burkhardt et al., 2005) [2]	Audio	Acted	anger, boredom, disgust, fear, happiness, sadness, neutral	493 sentences; 5 actors & 5 actresses
ISL meeting corpus (Burger et al., 2002)	Audio	Natural: meeting corpus	negative, positive, neutral	18 meetings; average 5 persons per meeting
Adult Attachment Interview database (Roisman, 2004) [3]	Audio-Visual	Natural: subjects were interviewed to describe the childhood experience	6 basic emotions, contempt, embarrassment, shame, general positive and negative emotions	60 adults: each interview was 30-60 minutes long
Belfast database (Douglas-Cowie et al., 2003) [4]	Audio-Visual	Natural: clips taken from television and realistic interviews with research team	Dimensional labeling/categorical labeling	125 subjects; 209 clips from TV and 30 from interviews
Busso-Narayanan database (Busso et al., 2007) [5]	Audio-Visual	Acted	anger, happiness, sadness, neutral	612 sentences; an actress
Haq-Jackson database (Haq & Jackson, 2009) [6]	Audio-Visual	Acted: emotion stimuli were shown on screen	6 basic emotions, neutral	480 sentences; 4 male subjects

경희대학교 KYUNG HEE UNIVERSITY

Related Works – Methodologies

165

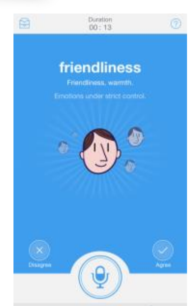
Reference	Data	Features	Classifier	Classes	Accuracy
Borchert et al., 2005 [7]	EmoDB	prosody, quality	SVM	7	70%
Luengo et al., 2005 [8]	97 samples per emotion; 21 number, 21 words, 55 sentences; single actress	prosody, MFCC	SVM, GMM	7	92%(SVM), 87%(GMM)
Lee et al., 2011 [9]	AIBO Dataset	Prosody, MFCC + Statistical Functions	Hierarchical Bayesian Logistic	5	48.2%
Schuller et al., 2009 [10]	EmoDB, eNTERFACE	Prosody, MFCC + Statistical Functions	SVM	7 (EmoDB), 6 (eNTERFACE)	84.6% (EmoDB), 72.5% (eNTERFACE)
Wang et al., 2015 [11]	EmoDB, CASIA	MFCC, Fourier Parameters	SVM	6	88.9% (EmoDB), 79% (CASIA)
Schuller et al., 2007 [12]	Audio-Visual, 10.5 hours of spontaneous conversation; 11 male and 10 female	prosody, articulatory, voice quality and linguistic information	SVM	3	64%
Haq & Jackson, 2009 [6]	Audio-Visual; 480 sentences; four male subjects	prosody, MFCC, 60 facial marker	GMM	7	56%(Audio), 95%(Video)
Poria et al., 2015 [13]	Audio-Visual; eNTERFACE	V: characteristic points, distances; A: MFCC, spectral features	SVM	6	81.2%(V), 78.6%(A), 87.95% (AV)

경희대학교 KYUNG HEE UNIVERSITY

Related Works

166

- Public framework, software
  - EmoVoice [14]
  - EMOSpeech [15]
  - Moodies Emotions Analytics [16]



emospeech
Speech Emotion Recognition

경희대학교 KYUNG HEE UNIVERSITY

Limitations of existing works

167

- Just focus on a particular problem
- Lack of preprocessing: noise reduction, voice activation detection, segmentation
- No consideration of personality
- There is no services utilizing emotion information: context modeling, recommendation, etc.

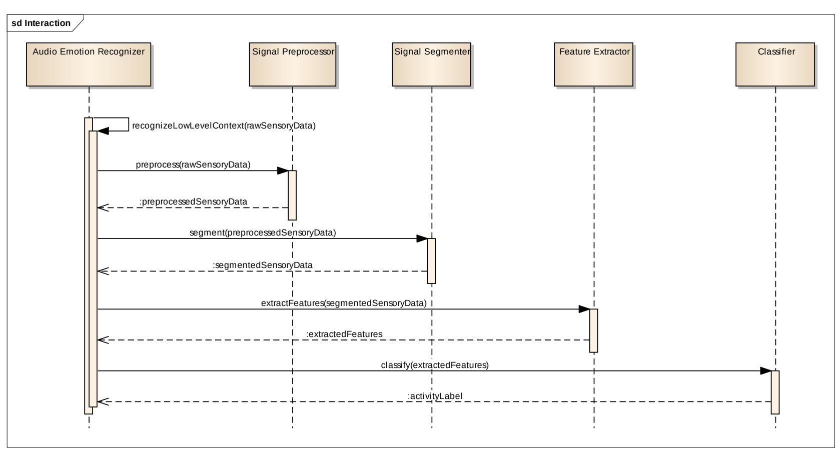
경희대학교 KYUNG HEE UNIVERSITY

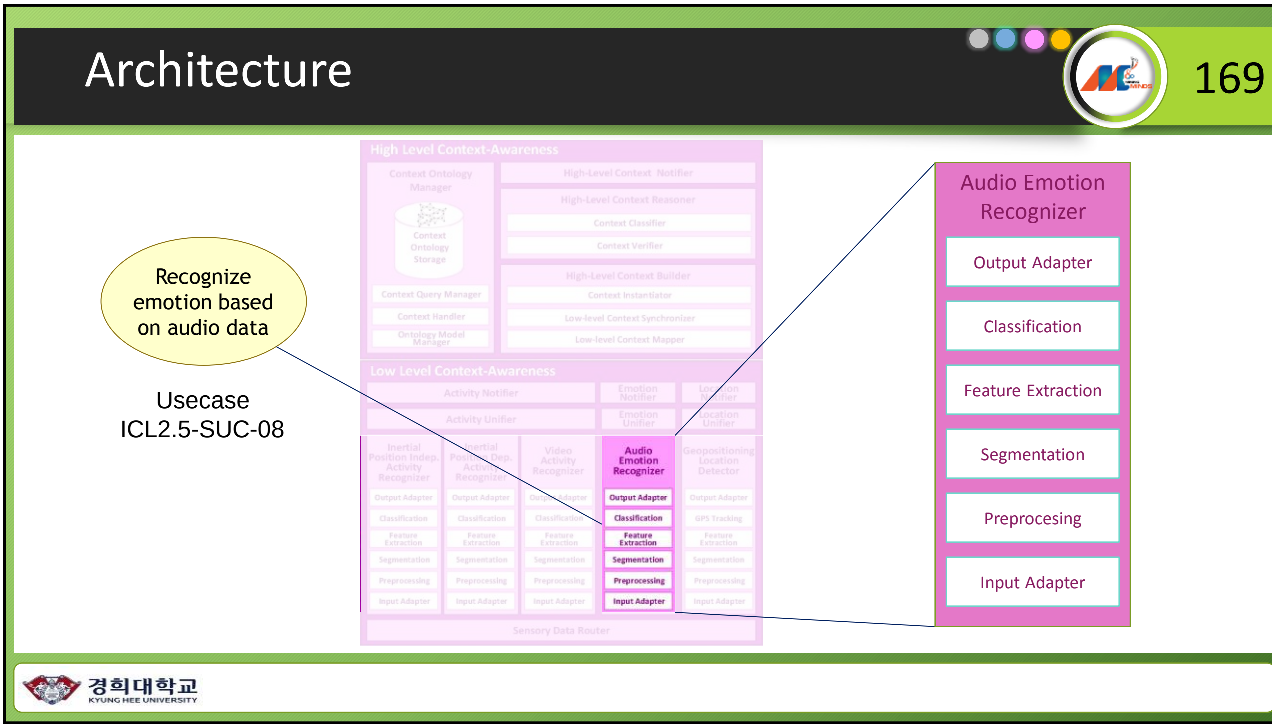
ICL System Specification

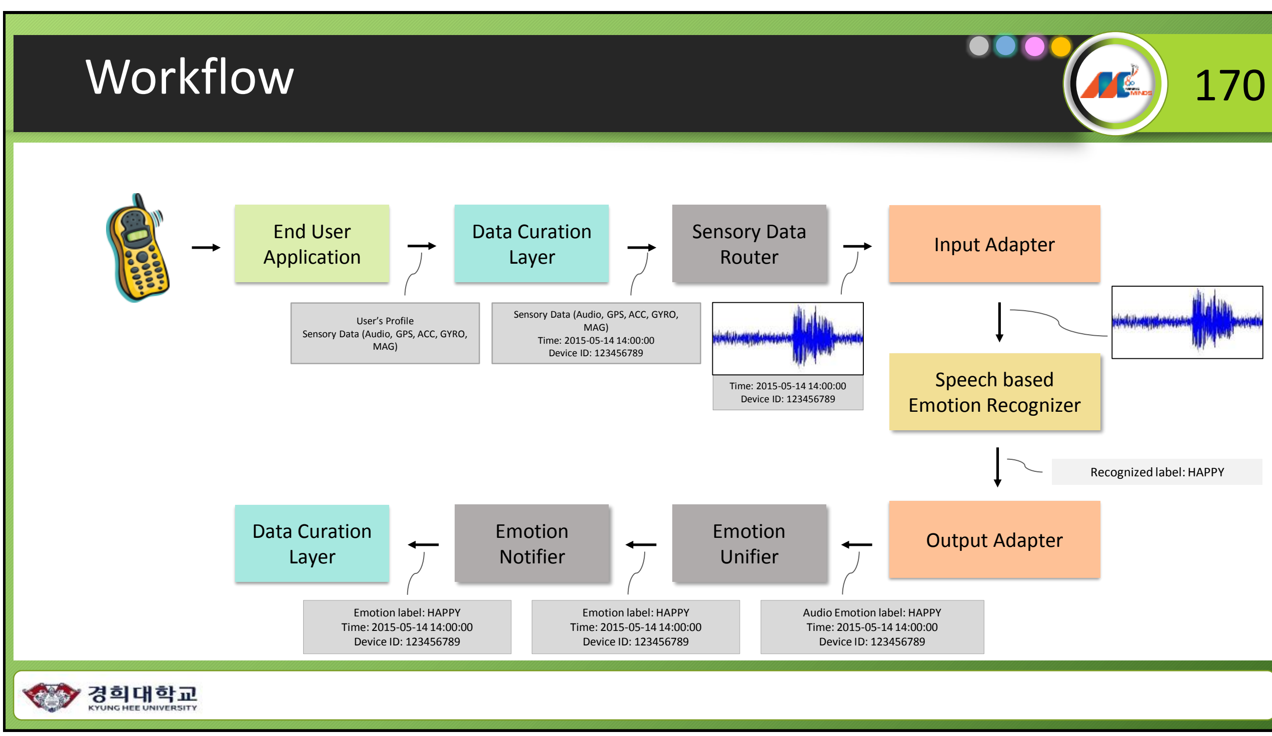
168

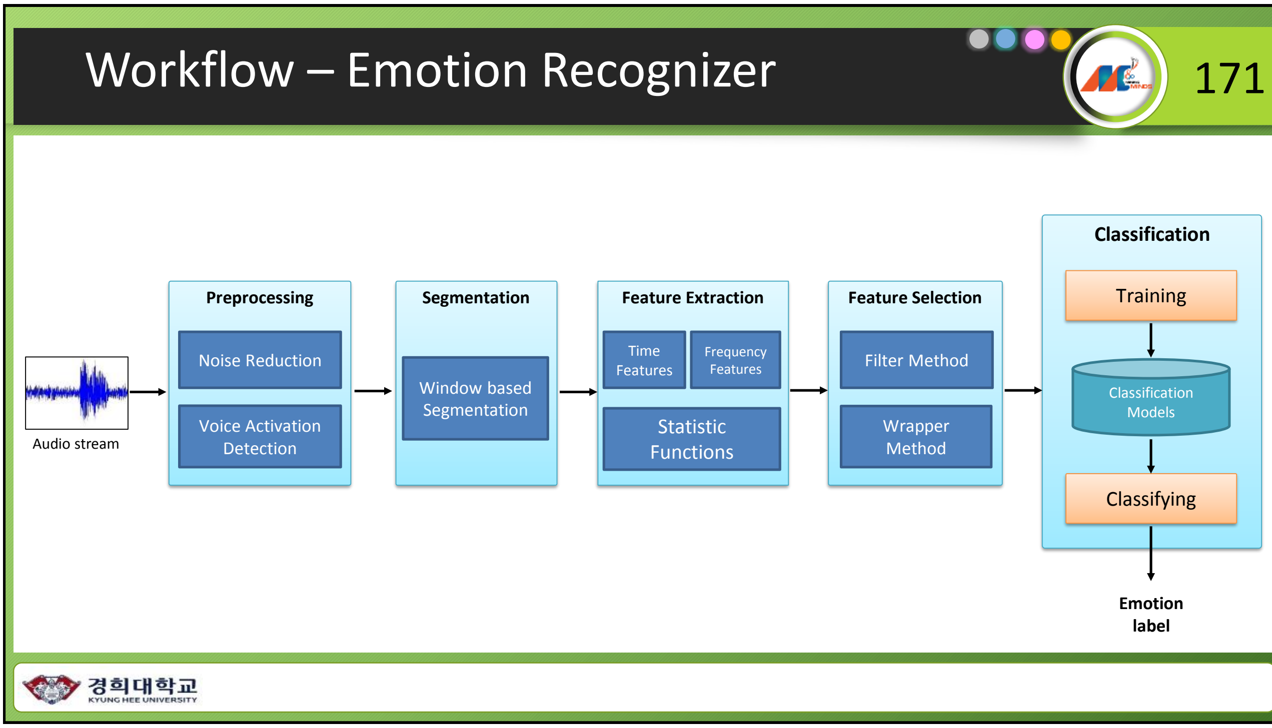
Usecase
ICL2.5-SUC-08

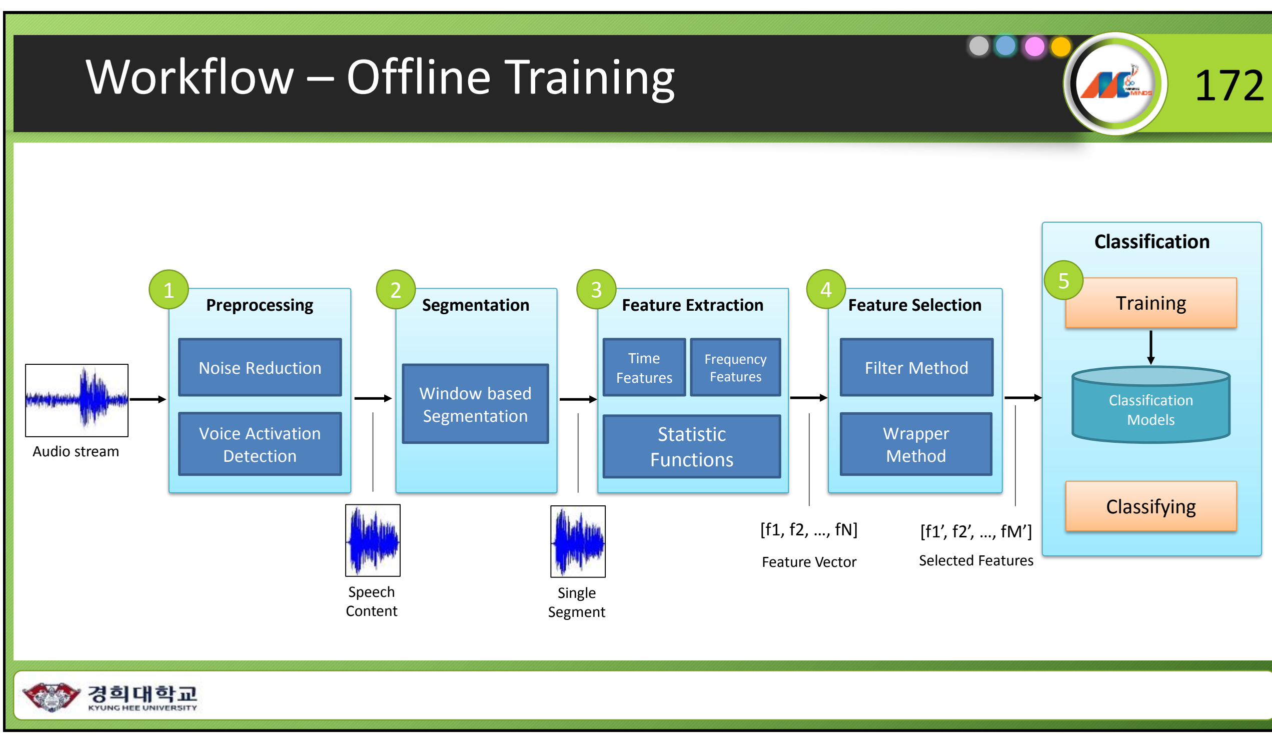
Sequence Diagram




Architecture
169
Recognize emotion based on audio data
Usecase ICL2.5-SUC-08
High Level Context-Awareness
Low Level Context-Awareness
Audio Emotion Recognizer
Output Adapter
Classification
Feature Extraction
Segmentation
Preprocesing
Input Adapter
Sensory Data Router
경희대학교
KYUNG HEE UNIVERSITY

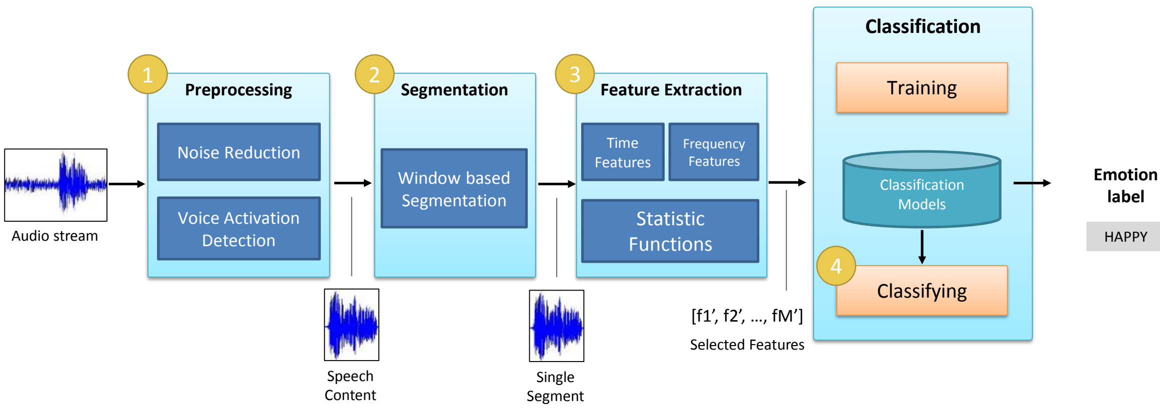

Workflow
170
End User Application
Data Curation Layer
Sensory Data Router
Input Adapter
User's Profile
Sensory Data (Audio, GPS, ACC, GYRO, MAG)
Sensory Data (Audio, GPS, ACC, GYRO, MAG)
Time: 2015-05-14 14:00:00
Device ID: 123456789
Time: 2015-05-14 14:00:00
Device ID: 123456789
Speech based Emotion Recognizer
Recognized label: HAPPY
Data Curation Layer
Emotion Notifier
Emotion Unifier
Output Adapter
Emotion label: HAPPY
Time: 2015-05-14 14:00:00
Device ID: 123456789
Emotion label: HAPPY
Time: 2015-05-14 14:00:00
Device ID: 123456789
Audio Emotion label: HAPPY
Time: 2015-05-14 14:00:00
Device ID: 123456789
경희대학교
KYUNG HEE UNIVERSITY


Workflow – Emotion Recognizer
171
Audio stream
Preprocessing
Noise Reduction
Voice Activation Detection
Segmentation
Window based Segmentation
Feature Extraction
Time Features
Frequency Features
Statistic Functions
Feature Selection
Filter Method
Wrapper Method
Classification
Training
Classification Models
Classifying
Emotion label
경희대학교
KYUNG HEE UNIVERSITY


Workflow – Offline Training
172
Audio stream
1
Preprocessing
Noise Reduction
Voice Activation Detection
2
Segmentation
Window based Segmentation
3
Feature Extraction
Time Features
Frequency Features
Statistic Functions
4
Feature Selection
Filter Method
Wrapper Method
5
Classification
Training
Classification Models
Classifying
Speech Content
Single Segment
[f1, f2, …, fN]
Feature Vector
[f1’, f2’, …, fM’]
Selected Features
경희대학교
KYUNG HEE UNIVERSITY

Workflow – Online Classifying

173



경희대학교 KYUNG HEE UNIVERSITY

Noise Reduction using Spectral Subtraction

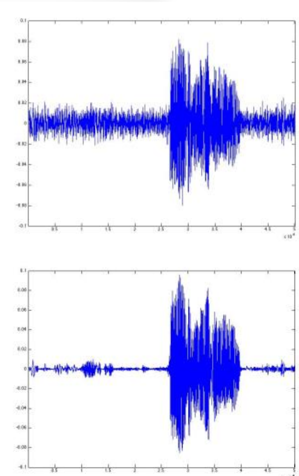
174

- Reduce the background noise in recording signal using spectral subtraction method

Time domain $y(t) = x(t) + n(t)$

Frequency domain $Y(w) = X(w) + N(w)$

From recording signal, we have Y(w) and approximation of N(w), then we can estimate X(w)



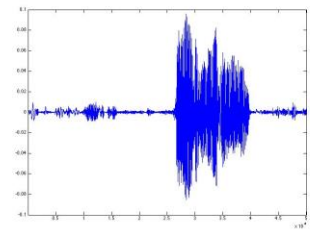
경희대학교 KYUNG HEE UNIVERSITY

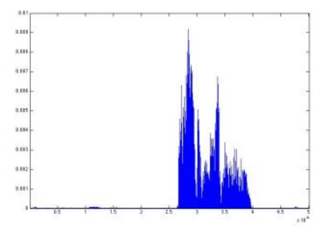
Voice Activation Detection

- Detect the segment of signal that contains the speech content only using energy threshold

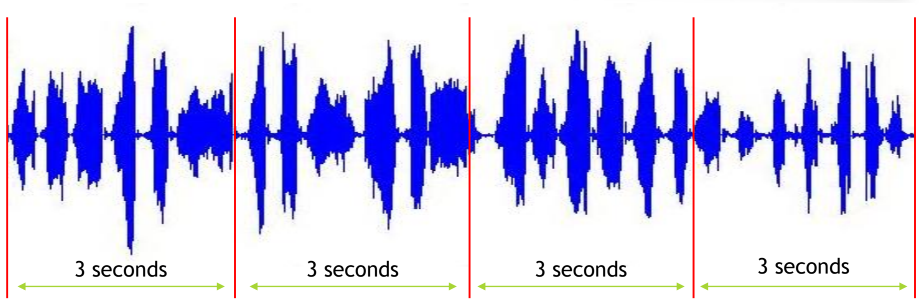
$$E = \sum x^2$$

- The average energy of sliding window is compared with a predefined threshold to consider the voice signal





Segmentation

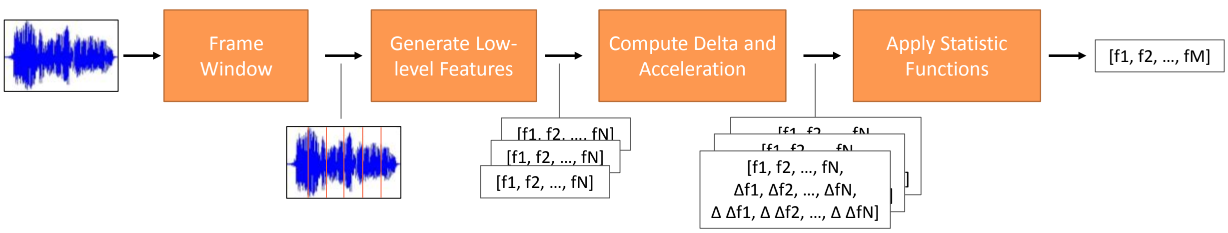


- Sliding window with 3-second length

Feature Extraction

177

- Generate low-level descriptors for each frame: pitch, root mean square energy, zero crossing rate, harmonic to noise ratio, mel-frequency cepstral coefficients
- Compute delta and acceleration coefficients of each descriptor
- Apply statistic functions: mean, standard deviation, kurtosis, skewness, maximum, minimum, etc.

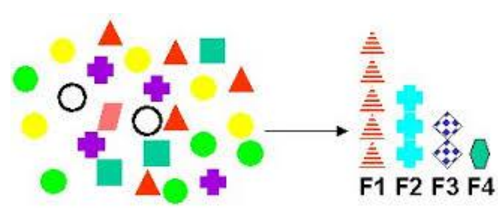


경희대학교
KYUNG HEE UNIVERSITY

Feature Selection

178

- Select prominent features, reduce redundancy
- Reduce computation
- Sequential forward feature selection
  - Starting from empty set
  - Select the next best feature based on given criteria
    - Filter approach: mutual information
    - Wrapper approach: accuracy
  - Update feature set with selected feature
  - Repeat to select the next feature

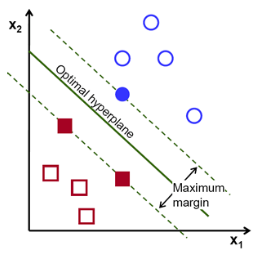


경희대학교
KYUNG HEE UNIVERSITY

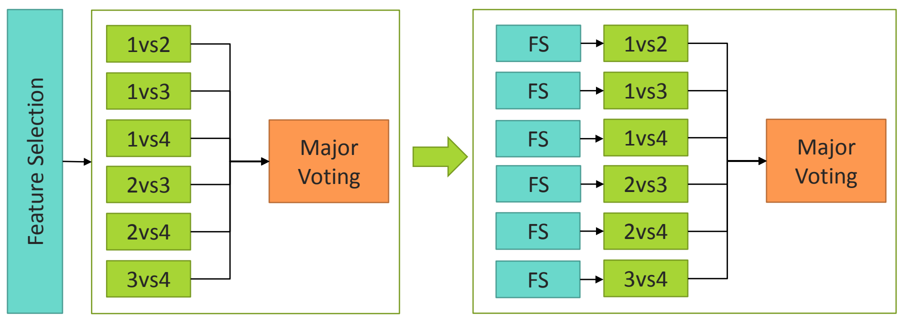
Classification

179

- Support Vector Machines



Find the optimal separating hyperplane that has the largest margin.



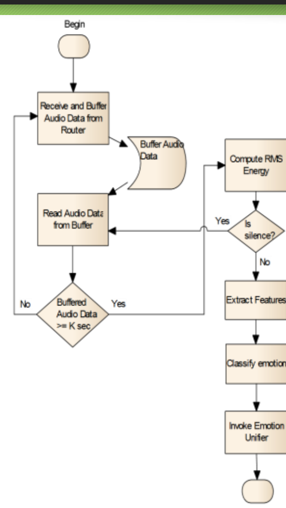
Original SVM for multiclass

Modified SVM for multiclass

경희대학교 KYUNG HEE UNIVERSITY

Flowchart

180



경희대학교 KYUNG HEE UNIVERSITY

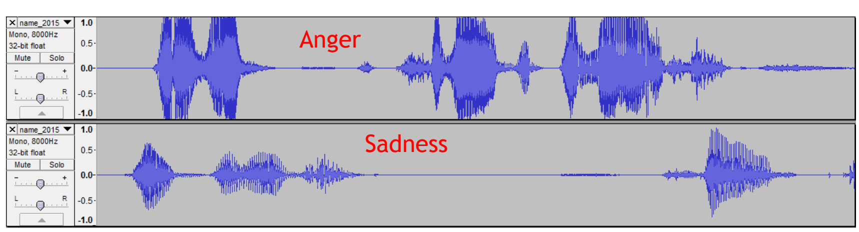
Tools and Technologies

- Implementation Tools and Technologies
  - Weka – Classification [17]
  - openSmile – Feature Extraction [18]
  - Eclipse
  - Programming Language: Java

Dataset Collection

- Scenario: Phone call
- Acted emotions: users try to simulate emotion by reading scripts
- 7 emotions: surprise, anger, sadness, neutral, boredom, fear, happiness

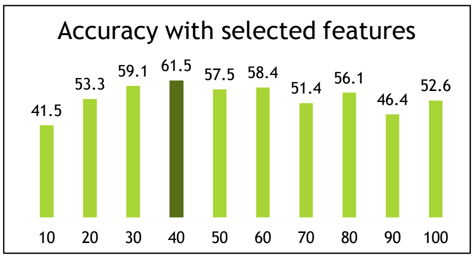
Number of emotions	7
Number of users	10
Language	Korean
Duration	1 min/emotion/person
Device	Galaxy S5



Evaluation Results

183

- 10-fold cross-validation
- Average accuracy: 57%
- Feature Selection



	a	b	c	d	e	f	g
a=anger	**15**	0	1	5	0	11	2
b=boredom	1	**32**	0	2	1	0	5
c=fear	1	0	**31**	7	1	0	1
d=happiness	6	0	5	**26**	0	0	5
e=sadness	0	9	1	2	**13**	1	2
f=surprise	9	2	1	2	2	**8**	1
g=neural	0	9	3	2	5	1	**19**

Implementation Plan

184

- Collection of multimodal real-world dataset (audio, video, GPS, inertial sensors) using smartphone with predefined scenarios
- Evaluation of the proposed methodology on the collected dataset
- Implementation and Evaluation for ICLv2.5
  - Designing class, sequence diagrams
  - Writing source code
  - Offline evaluating with collected data
  - Online evaluating with real-time data

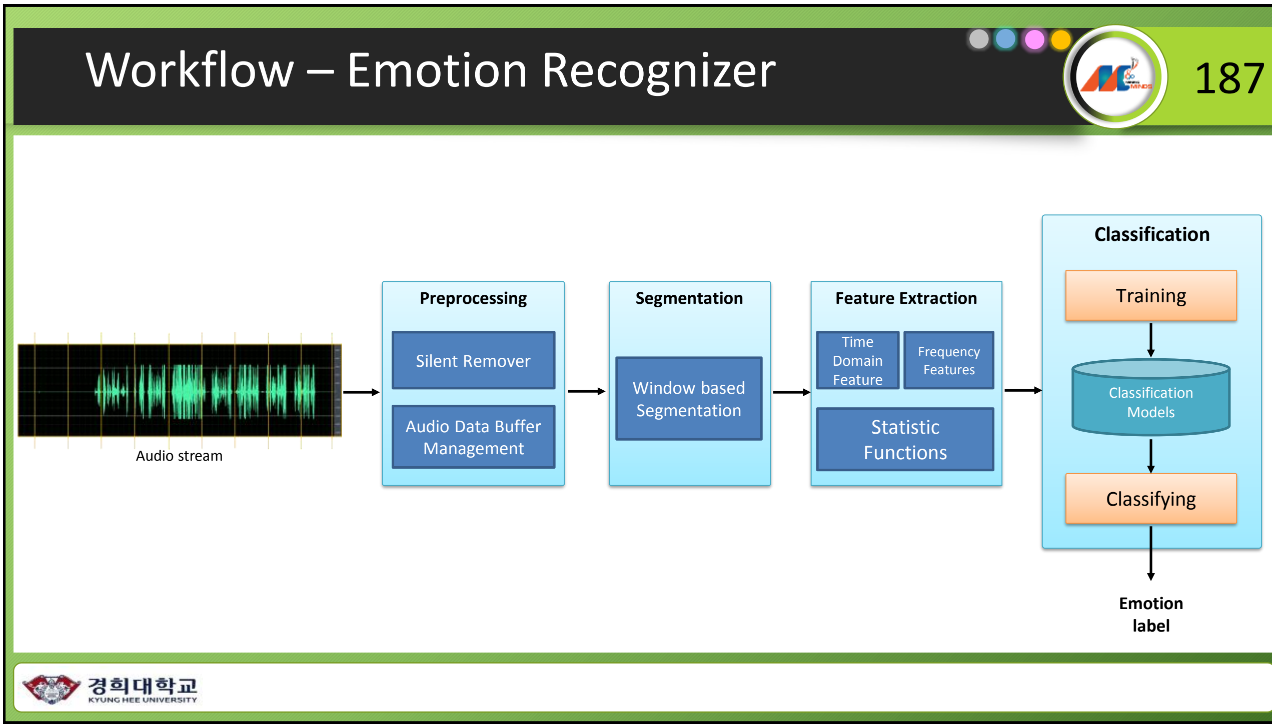
경희대학교 KYUNG HEE UNIVERSITY

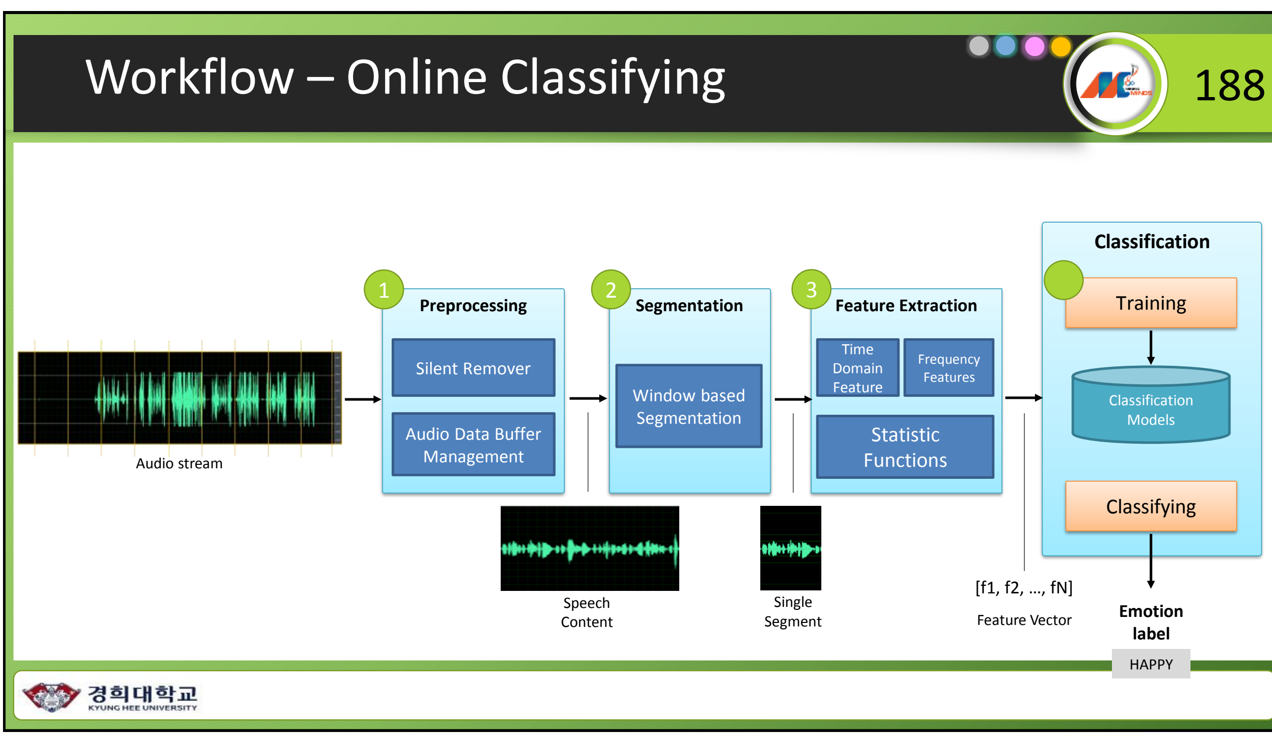
Contributions

- Proposing a methodology for speech based emotion recognition
- Creation of an emotion set based on requirements of services to be delivered
- Collection of a real-world emotional speech dataset
- Offline evaluation and validation of various emotion recognition models
- Implementation for ICL version 2.5
- Online validation

Speech Emotion Recognition

Jaehun Bang

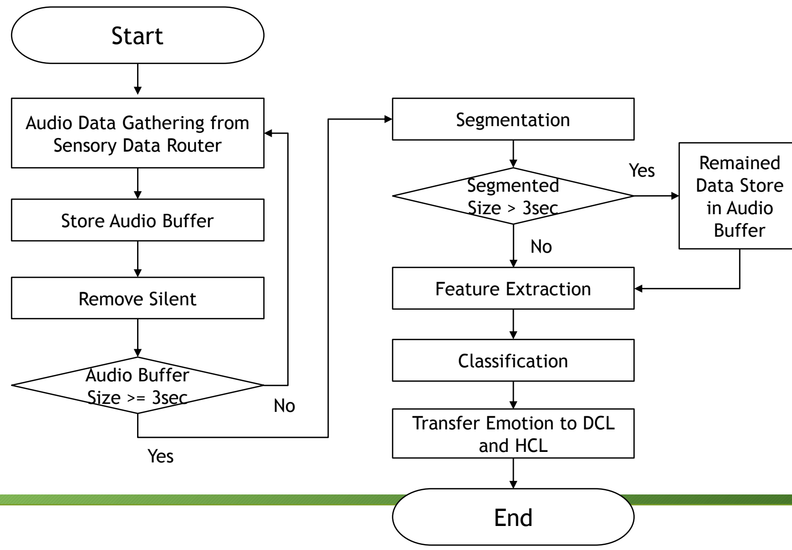

Workflow – Emotion Recognizer
187
Audio stream
Preprocessing
Silent Remover
Audio Data Buffer Management
Segmentation
Window based Segmentation
Feature Extraction
Time Domain Feature
Frequency Features
Statistic Functions
Classification
Training
Classification Models
Classifying
Emotion label
경희대학교
KYUNG HEE UNIVERSITY


Workflow – Online Classifying
188
Audio stream
1
Preprocessing
Silent Remover
Audio Data Buffer Management
2
Segmentation
Window based Segmentation
3
Feature Extraction
Time Domain Feature
Frequency Features
Statistic Functions
Classification
Training
Classification Models
Classifying
Speech Content
Single Segment
[f1, f2, ..., fN]
Feature Vector
Emotion label
HAPPY
경희대학교
KYUNG HEE UNIVERSITY

Flow Chart

189

1. Gather Audio Raw Sensory Data from Sensory Data Router Module
2. Stored Audio Buffer for Collecting Sufficient Data
3. Remove Silent in Audio
4. Check Audio Buffer Size, If Audio Buffer Size More Than 3sec, Begin Recognize Process. If Not, Return to Audio Data Gathering
5. Segment Data based on 3sec
6. If Segmented Data Size More than 3sec, Remained Data store in Audio Buffer. If Not, Begin Feature Extraction
7. Extract Features based on Time Domain Features, Statistical Feature and Frequency Features
8. Classification based on SVM
9. Transfer Emotion Label to DCL And HCL Server



Evaluation Result

190

- Data: 9 person data
- Evaluation Method : 10-fold cross validation
- Classifier: SVM
- Evaluation Tool : Weka

7 emotions – 44.1176%

	Angry	Bored	Fear	Happy	Normal	Sad	Surprise
Angry	46	1	7	2	2	3	6
Bored	1	30	4	1	1	9	1
Fear	10	12	27	5	2	5	1
Happy	12	3	5	31	1	2	7
Normal	3	4	8	3	15	9	1
Sad	0	22	6	0	6	8	1
Surprise	31	1	5	3	3	0	8

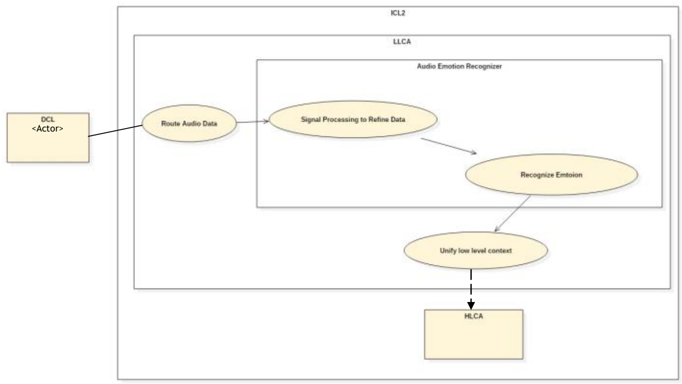
4 emotions – 70.0935%

	Angry	Happy	Normal	Sad
Angry	52	6	6	3
Happy	13	42	2	4
Normal	4	3	27	9
Sad	2	0	12	29

Use Case Diagram

Recognize emotion based on audio data

Usecase ICL2.5-SUC-08

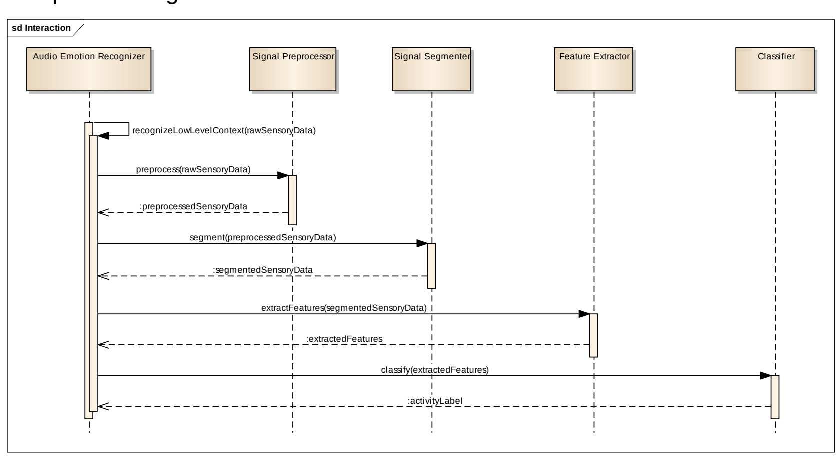


ICL System Specification

Recognize emotion based on audio data

Usecase ICL2.5-SUC-08

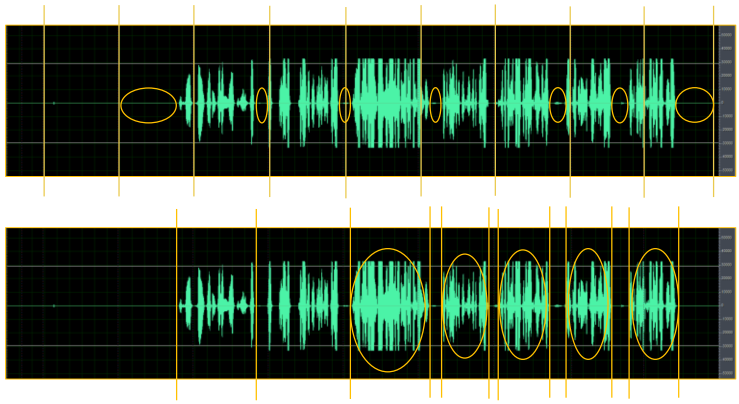
Sequence Diagram



Future works

193

- Real-time Recognition Windowing Problem
  - Call Speech data always contain blank speech
    - In the Speech Emotion Recognition, the blank area has not meaningful

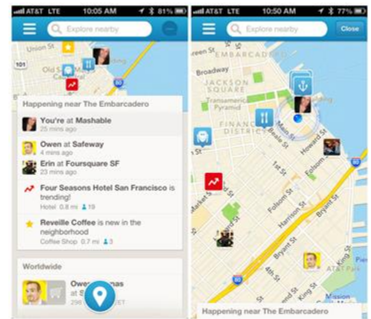


Location Detection

San Kim

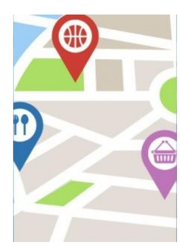
Introduction

- The identification of people location is of much interest to support diverse type of services
- Location-based services have been widely explored during the last years, mainly exploited in social networks such as Foursquare, TripAdvisor or Twitter among many others
- Location detection techniques can be essentially categorized into outdoor or geopositioning methods and indoor positioning approaches



Problem Statement

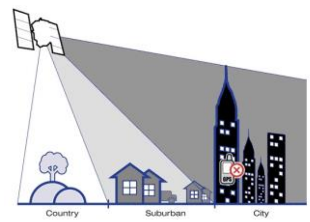
- Motivation
  - Users location is a primary source of information to infer context
  - Tracking of people locations can also serve to identify behaviors and routines
- Goal
  - Define mechanisms to seamlessly identify and register the user location by using technologies readily available to every user
- Objectives
  - Tag based user location detection
  - Creation of personalized maps with general and user-centric points of interest
  - New tagging available in run-time



Challenges

197

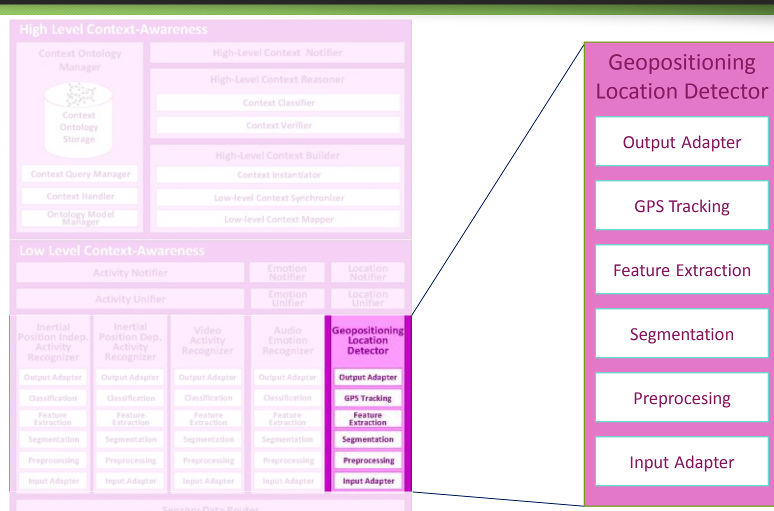
- Geolocation mechanisms are subject to GPS signal availability; thus, no detection is possible in places without coverage (e.g., indoor spaces)
- There exist diverse sort of map APIs; however, they present limitations in their use for some particular countries (e.g., Google Maps in Korea)
- A very location may have different meanings for different users (e.g., "Peter's fitness center" is in the same building where "David's favourite restaurant" is located)



경희대학교 KYUNG HEE UNIVERSITY

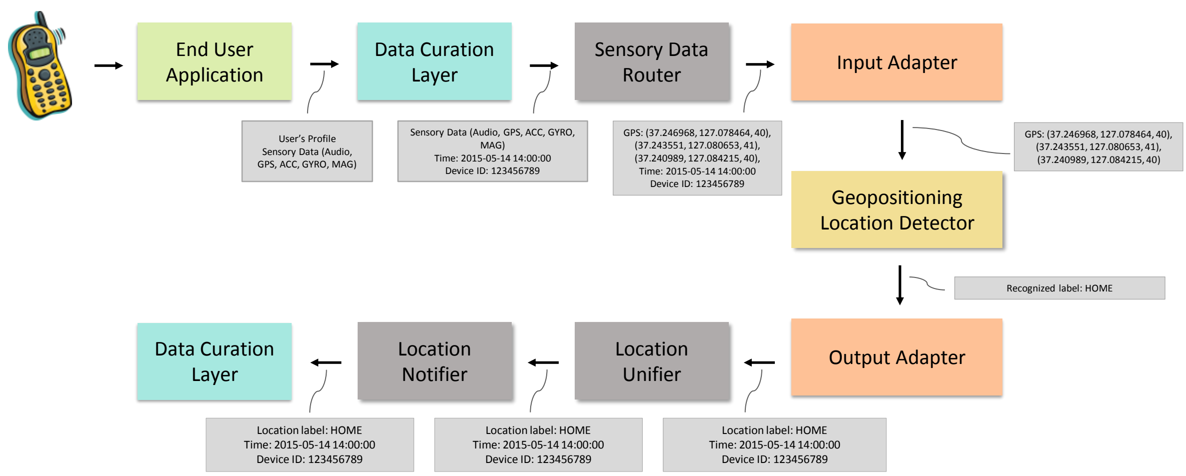
Architecture

198



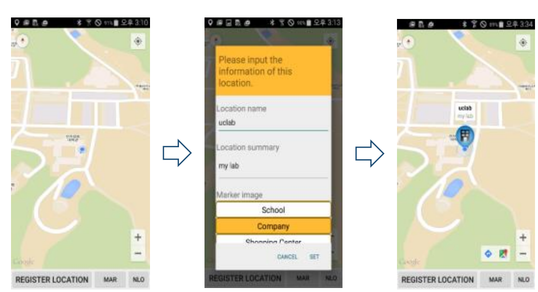
경희대학교 KYUNG HEE UNIVERSITY

Workflow



Location Detection

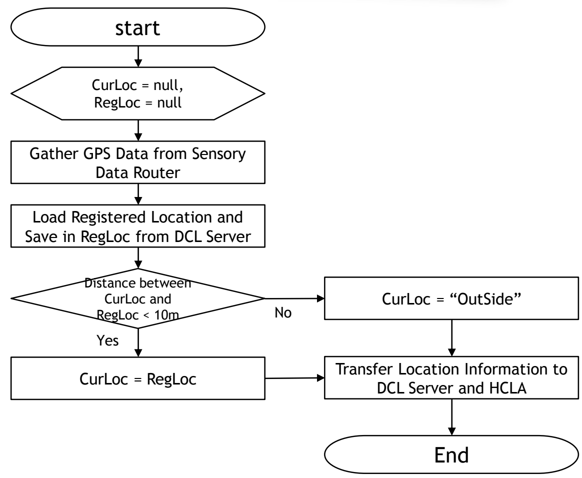
- Implementation of user-centered location registation module
- Implementation of classfying general POI (e.g., "City Hall"), user-centered POI (e.g., "home") and Frequency POI (e.g., "Restuarants")
- New features on MM v2.5
  - If user stays at the same location over than 15 minutes, system automatically detects the context and new location
  - The system execute the new location tagging window on the screen
  - The user inputs the specification of the location
  - The application sends the new location data (Name, explanation, latitude, longitude) to MM server for new location registration



Designed Location Detector Flowchart

201

1. Initialize Current Location and Registered Location Variable.
2. Gather User GPS Data from Sensory Data Router in LLCA
3. Load Registered Location and Save in RegLoc Variable form DCL Server
4. Calculate Distance Between CurLoc and RegLoc.
5. If Distance < 10m, the CurLoc has changed RegLoc
6. If Distance > 10m, Put “OutSide” in Cur Loc
7. Transfer Extracted Location to DCL Server and HCLA



Conclusion & Future Work

8

- Conclusion
  - The identification of users location is of primal necessity in order to fairly identify the user context
  - Location tracking can be also used to better determine user behavior and routines
  - GPS-based location detection is explored in MMv2.0 and MMv2.5
  - Both user-centric and generic maps are considered to personalize the information acquired from the user location
- Future Work
  - Automatic location detection function will be added while current version is tag based

Tag Based Food Recognizer

Tae ho hur

Introduction

204

- Main factor for healthy lifestyle is habit, exercise and food
  - Alcohol and smoking will deteriorate user's body functions
  - Exercise will increase metabolism and depletes stress
  - Balanced food intake will prevent malnutrition and obesity
- Food is the most complex factor to be handled be individual
  - Number of bad habits and exercise are well known while the number of food is not comparable
    - There are thousands of food all over the world
  - Ordinary people do not know what kind and proportion of nutrition is included in the food
    - The help of nutritionist is required

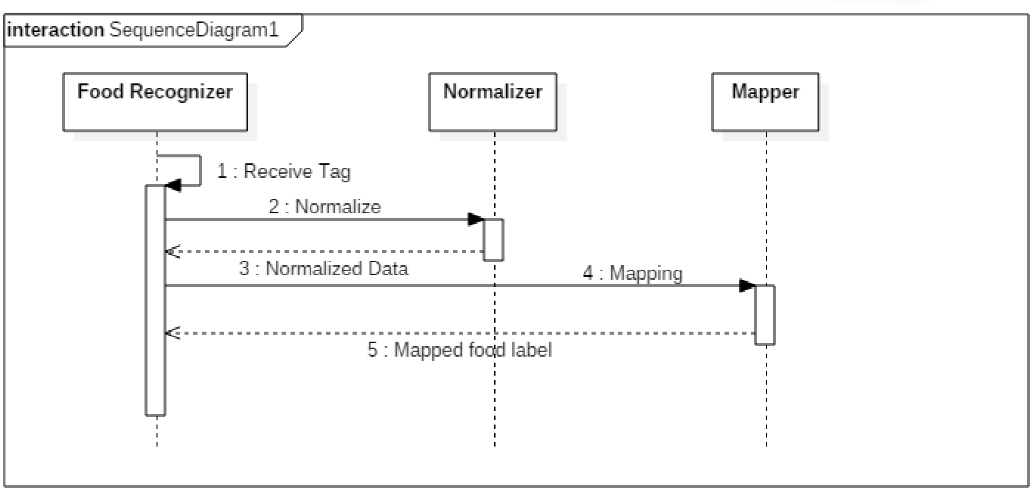
Problem Statement

205

- Motivation
  - Knowing which kind of food the user has eaten is a challenging task using the sensors
  - Image is the only option while analyzing the image to know the kind of food needs advanced machine learning
- Goal
  - Recognize what kind of food the user has eaten by tagging
- Objectives
  - Take the picture of the food and tag the label of it
  - Send the picture and food label

Sequence diagram for Food recognizer

206



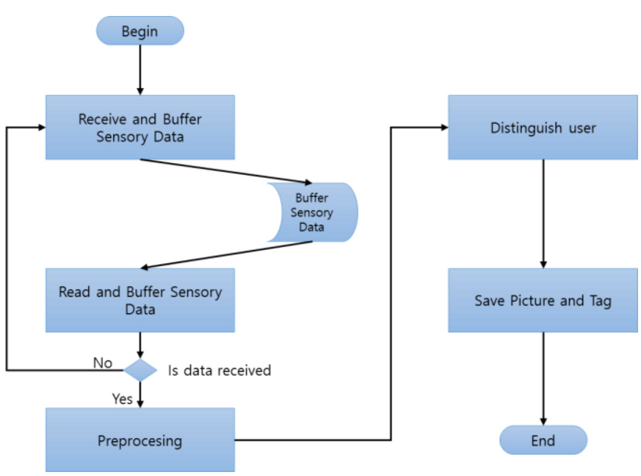
Architecture

207

경희대학교
KYUNG HEE UNIVERSITY

Flowchart

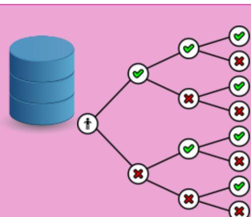
208



경희대학교
KYUNG HEE UNIVERSITY

TAG Based Food Recognizer

Tagging



High level context awareness

Send

TAG Based Food Recognizer

	Food Items
1	Rice
2	Noodle
3	Oats
4	Barley
5	Quinoa
6	Brown
7	Rye
8	Bread

Send

Cloud Services

Food Categories		Food Items	
1	Grain	1	Rice
		2	Noodle
		3	Oats
		4	Barley
		5	Quinoa
		6	Brown
		7	Rye
		8	Bread

Conclusion & Future work

- Conclusion
  - Purpose of this module is to check user’s food intake habit and give appropriate recommendation of nutrition intake
  - User’s nutrition intake proportion and consumed calories will be analyzed by the expert
- Future Work
  - Image processing based food recognition
    - ✓ Automatic food recognition by machine learning without user based tagging

경희대학교 KYUNG HEE UNIVERSITY

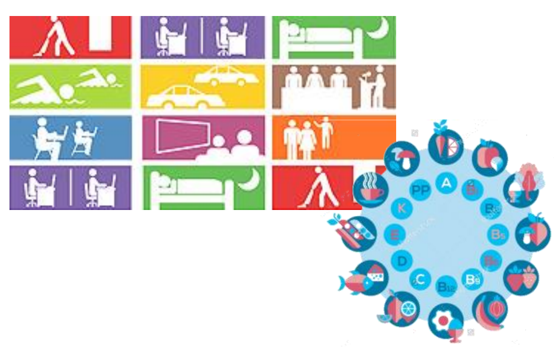
High Level Context Awareness

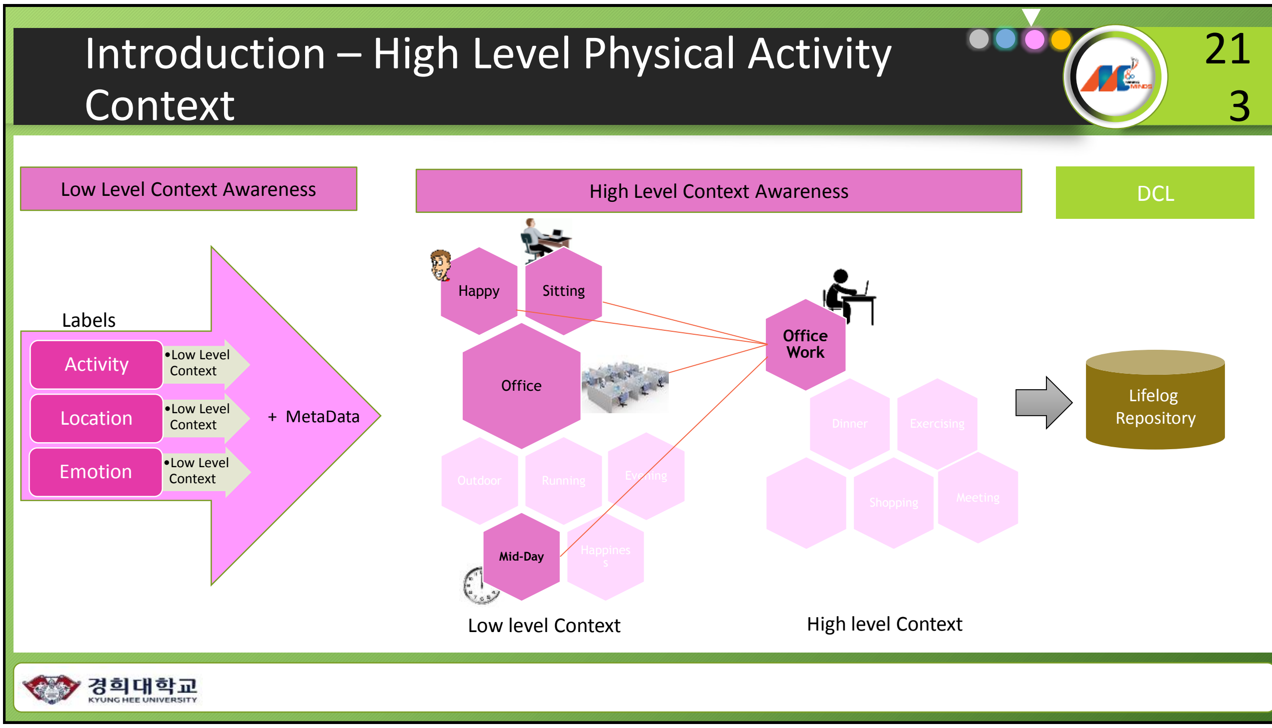
Muhammad Asif Razzaq
Dr. Wajahat Ali Khan

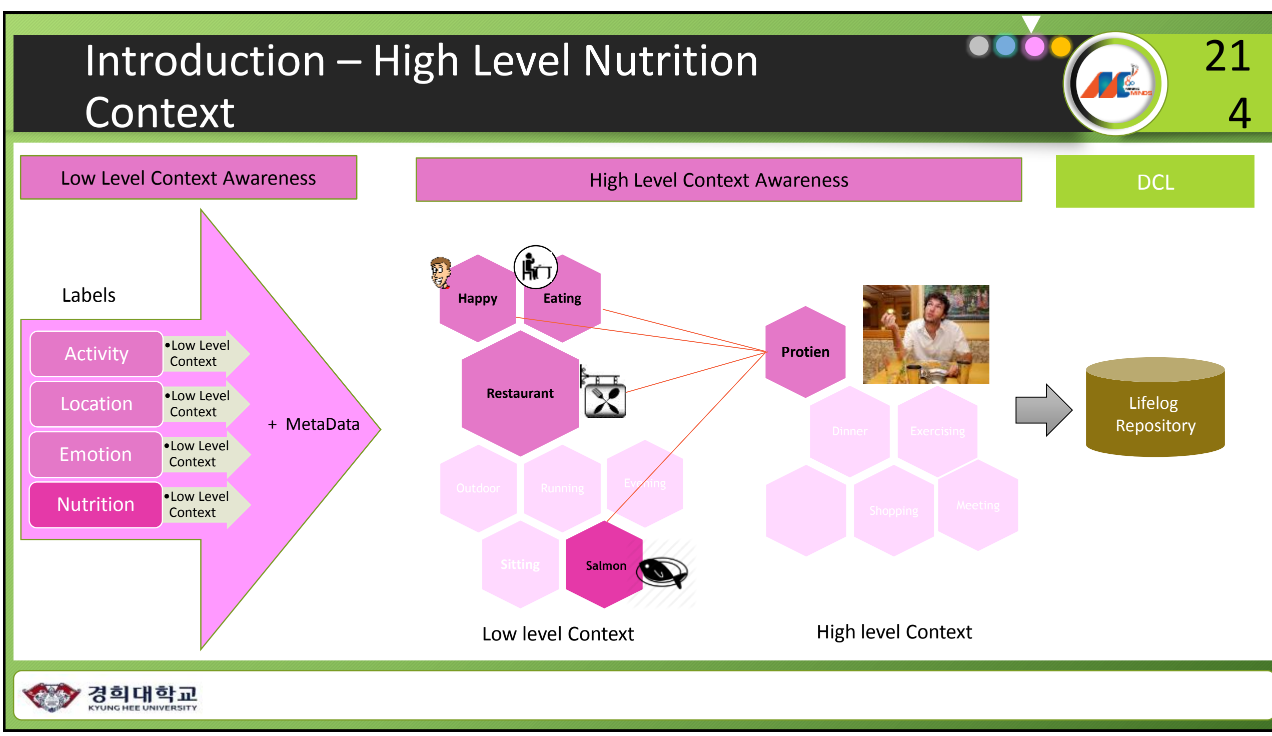
Introduction

212

- High Level Context is obtained from Processing & Reasoning of Low Level Context based on user's
  - Activity
  - Location
  - Emotion
  - Nutrition




Introduction – High Level Physical Activity Context
21
3
Low Level Context Awareness
High Level Context Awareness
DCL
Labels
Activity
Location
Emotion
•Low Level Context
•Low Level Context
•Low Level Context
+ MetaData
Happy
Sitting
Office
Outdoor
Running
Evening
Mid-Day
Office Work
Dinner
Exercising
Shopping
Meeting
Lifelog Repository
Low level Context
High level Context
경희대학교
KYUNG HEE UNIVERSITY


Introduction – High Level Nutrition Context
21
4
Low Level Context Awareness
High Level Context Awareness
DCL
Labels
Activity
Location
Emotion
Nutrition
•Low Level Context
•Low Level Context
•Low Level Context
•Low Level Context
+ MetaData
Happy
Eating
Restaurant
Outdoor
Running
Evening
Sitting
Salmon
Protien
Dinner
Exercising
Shopping
Meeting
Lifelog Repository
Low level Context
High level Context
경희대학교
KYUNG HEE UNIVERSITY

Motivation

215

- Abstract description of user’s context
- Extraction of High level context from low level context for better understandability of user’s context.
- Identification of High Level Context for decision making by upper layers:
  - Personalized recommendations
  - Behavior modeling
  - Personalized predictions

경희대학교
KYUNG HEE UNIVERSITY

Goal and Objectives

216

- **Goal**: Design and implementation of methodology for high level context recognition.
- **Objectives:**
  - Achieve acceptable accuracy for identifying HLCA
  - Proposal and implementation of Low level context synchronization technique
  - Deployment of Triple storage for ontology persistence
  - Modeling of High-level and Low-level context
  - Reasoning in order to derive High-level context from Low-level context
  - Development of a simulation tool to generate low-level context instances

경희대학교
KYUNG HEE UNIVERSITY

Related Work

217

Authors	Domain	Methodology	Features	Limitations
Perera 2014	IoT	Context Aware Computing	• Survey w.r.t IoT • Comprehensive Analysis and Evaluation of Context Aware Techniques	• No Implementation • No Practical Implementation with Results.
Bellavista 2013	Ubiquitous Systems	Unified Architectural model	• Context Data Distribution • Classification of Context • Runtime Adaptation Support	• Context Aggregation and Filtering • Adaptive Context
Khattak 2014	Context Aware Systems	Context Fusion	• Context Fusioning Methods • Survey of Context Representation Schemes	• No Implementation • Evaluation and Proof of Concepts Missing
Moen 2015	Mobile Computing	Activity Recognition Algorithm	• Future Research Methodologies	• Activity Recognition without considering Emotions.
Perera 2013	IoT	Component Level Architecture	• Sensor Selection • Context Aware Architecture	• Semantic and Quantitative Reasoning Missing
Gerhard 2012	Context Aware Systems	Context Aware Framework	• Context Acquisition • Context Representation • Context Utilization	• No Implementation • Evaluation and Proof of Concepts Missing

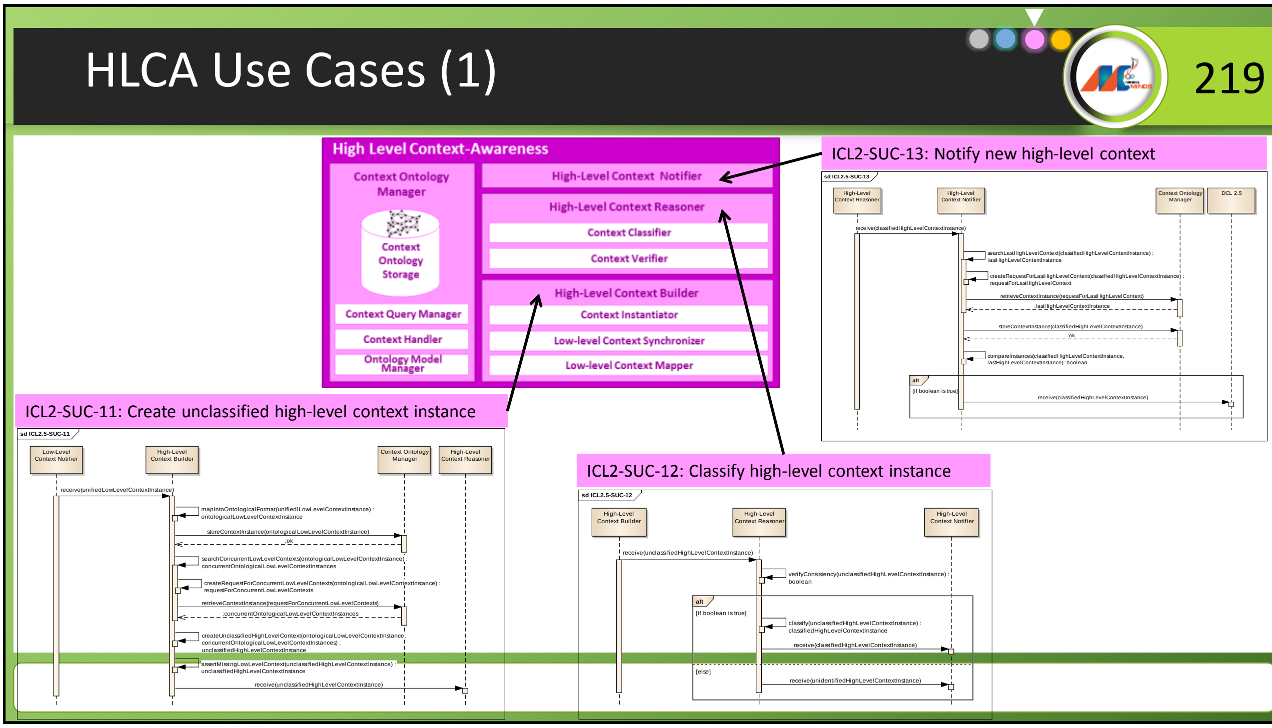
경희대학교 KYUNG HEE UNIVERSITY

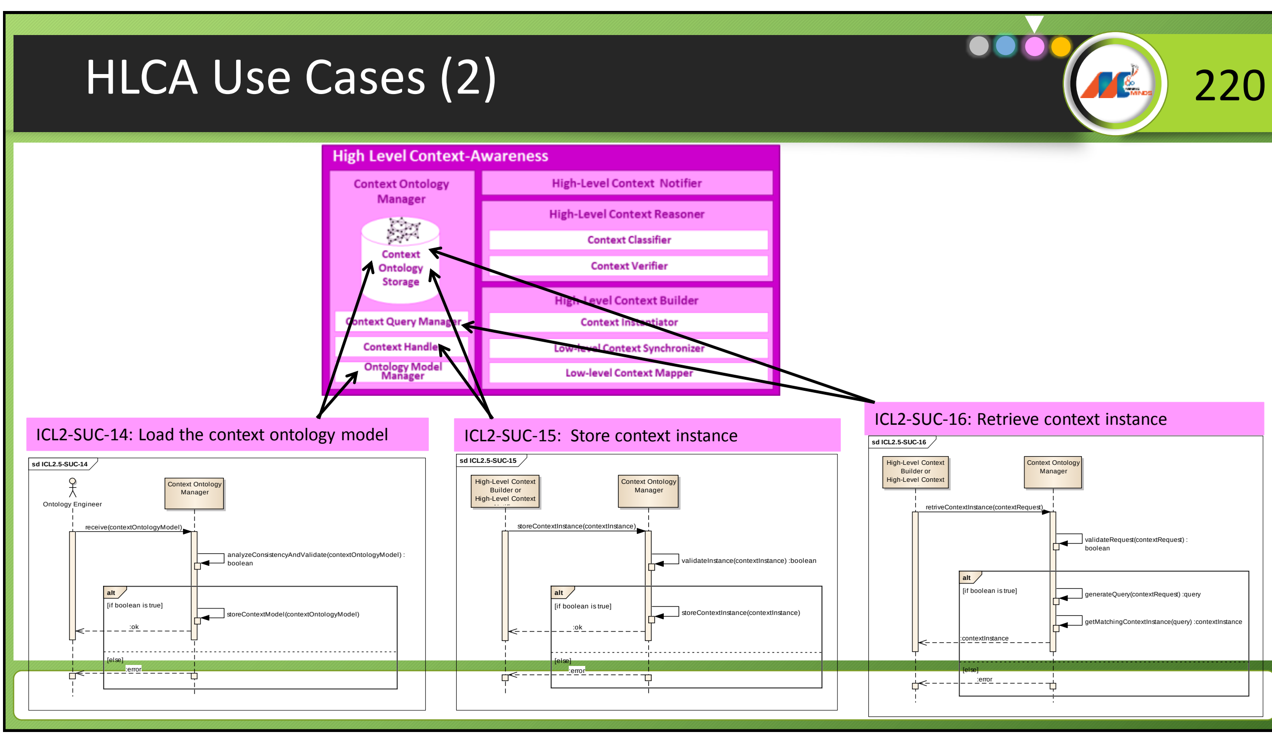
Limitations of existing Work

218

- Lack of Implementation
- Context Aggregation and Abstraction
- Activity Recognition without Emotion Detection in High Level Context Modelling
- Evaluation and Proof of Concept Missing
- Semantic Reasoning Missing

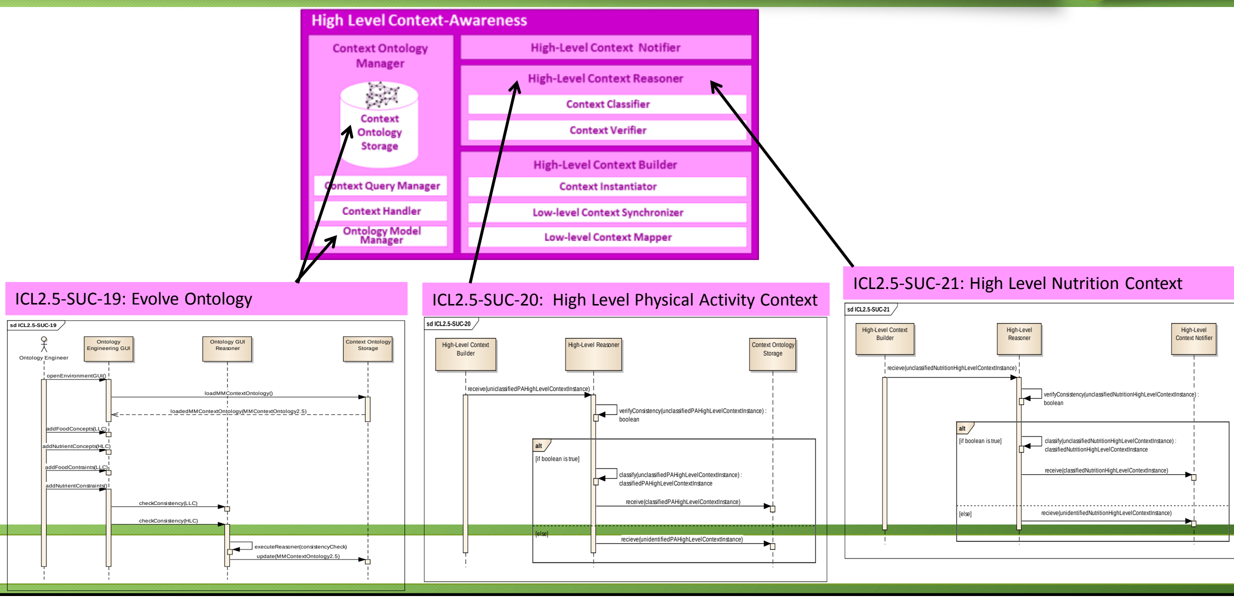
경희대학교 KYUNG HEE UNIVERSITY


HLCA Use Cases (1)
219
High Level Context-Awareness
Context Ontology Manager
Context Ontology Storage
Context Query Manager
Context Handler
Ontology Model Manager
High-Level Context Notifier
High-Level Context Reasoner
Context Classifier
Context Verifier
High-Level Context Builder
Context Instantiator
Low-level Context Synchronizer
Low-level Context Mapper
ICL2-SUC-13: Notify new high-level context
ICL2-SUC-11: Create unclassified high-level context instance
ICL2-SUC-12: Classify high-level context instance


HLCA Use Cases (2)
220
High Level Context-Awareness
Context Ontology Manager
Context Ontology Storage
Context Query Manager
Context Handler
Ontology Model Manager
High-Level Context Notifier
High-Level Context Reasoner
Context Classifier
Context Verifier
High-Level Context Builder
Context Instantiator
Low-level Context Synchronizer
Low-level Context Mapper
ICL2-SUC-14: Load the context ontology model
ICL2-SUC-15: Store context instance
ICL2-SUC-16: Retrieve context instance

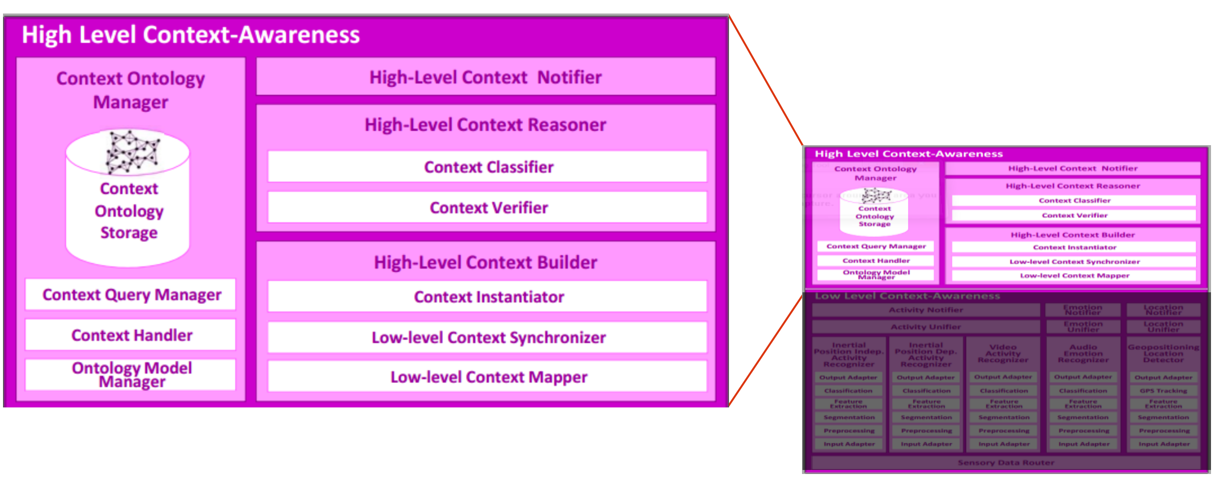
HLCA Use Cases (2)

221

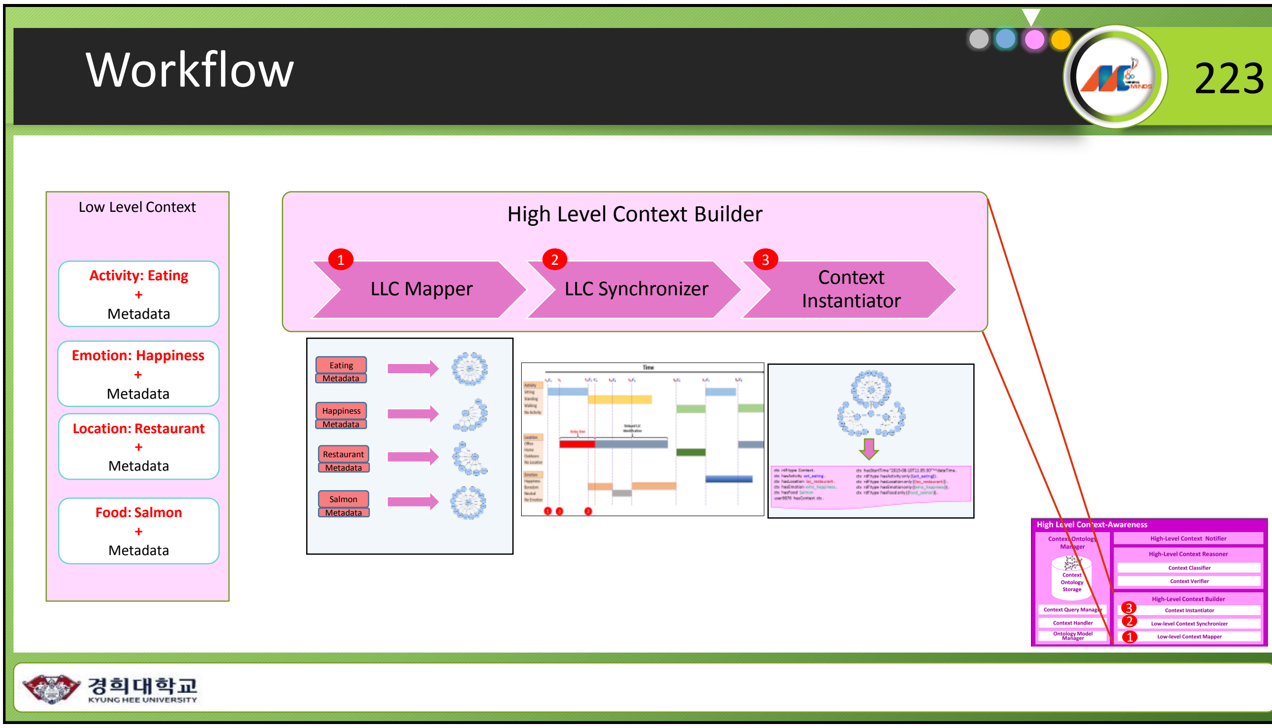


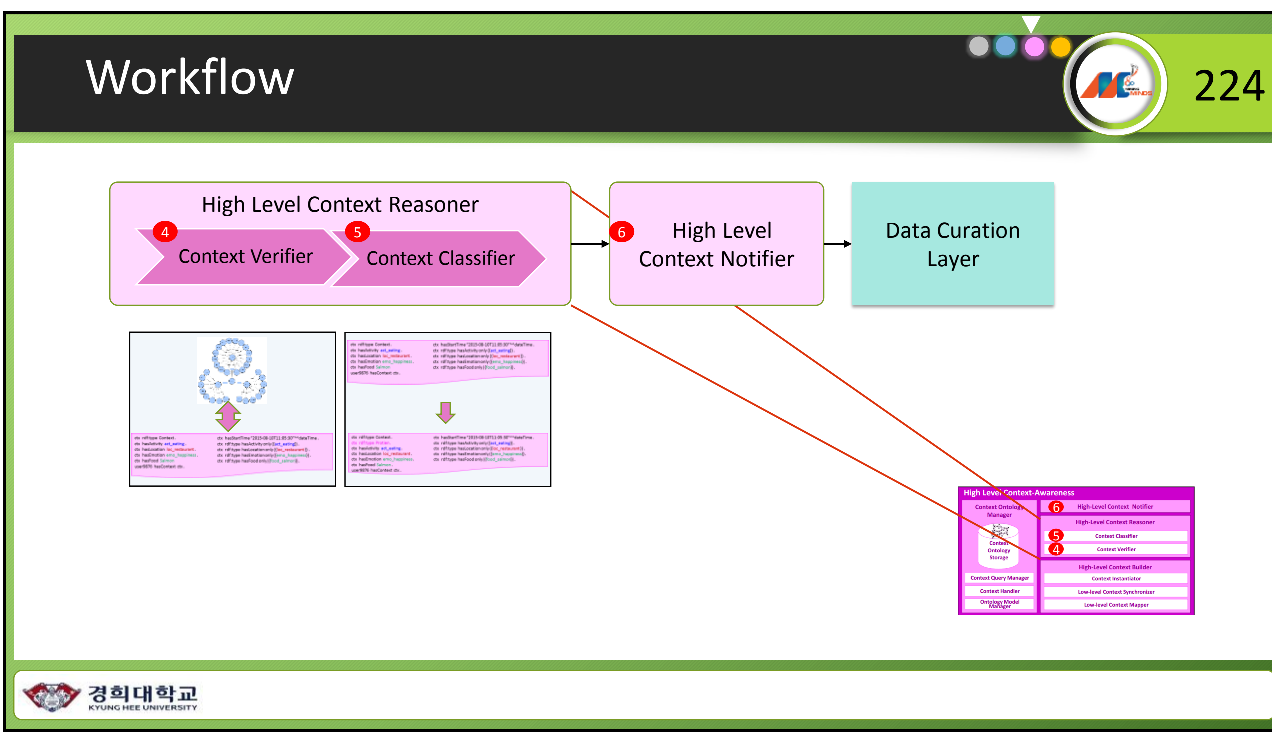
HLCA Architecture

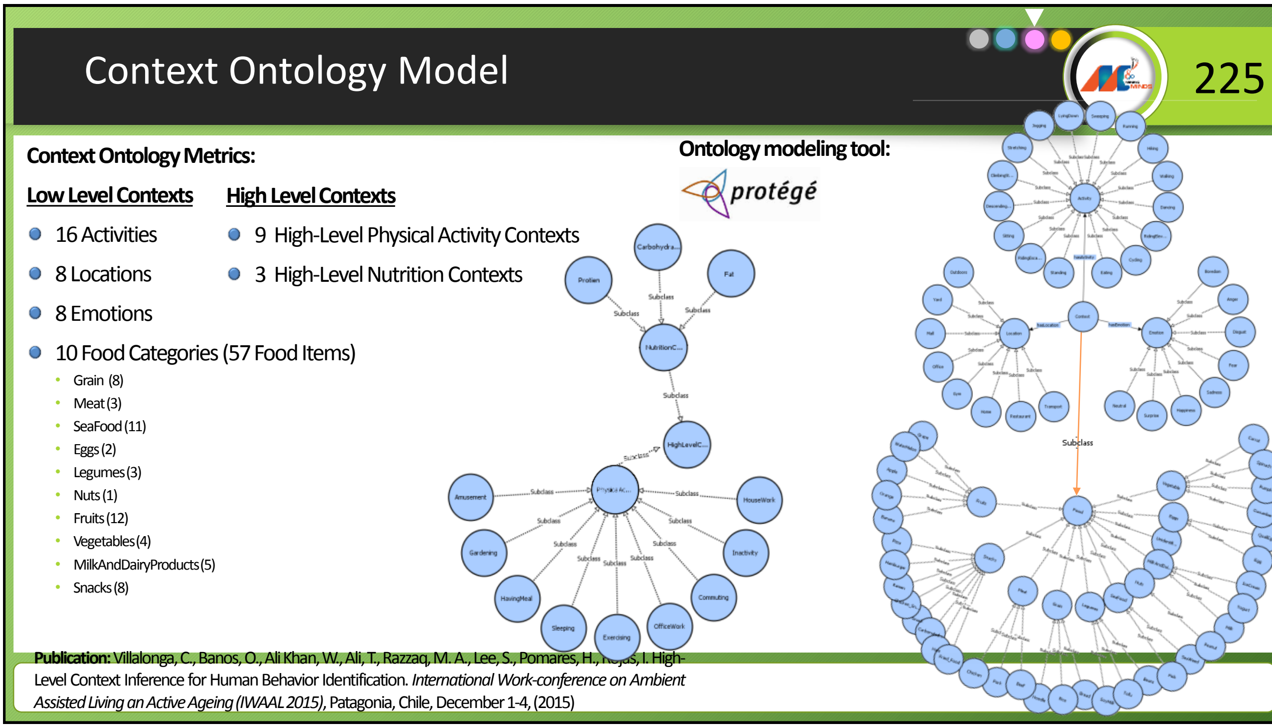
222

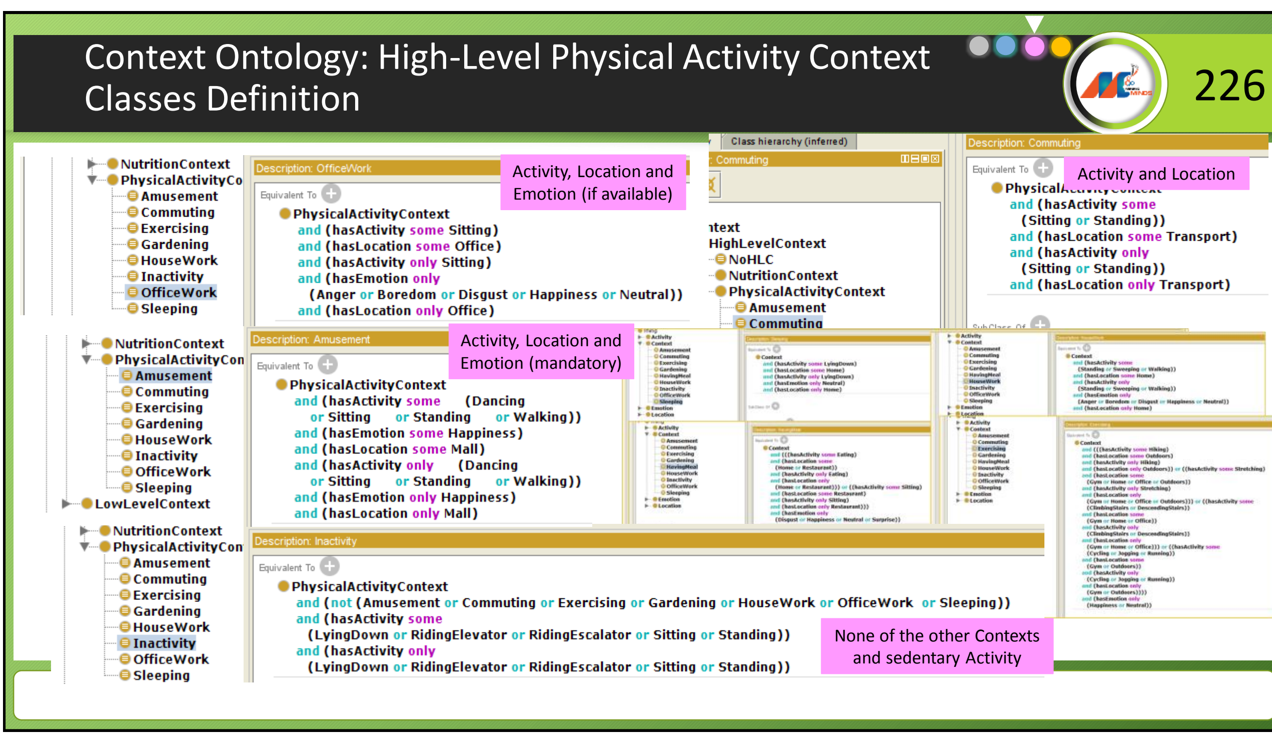


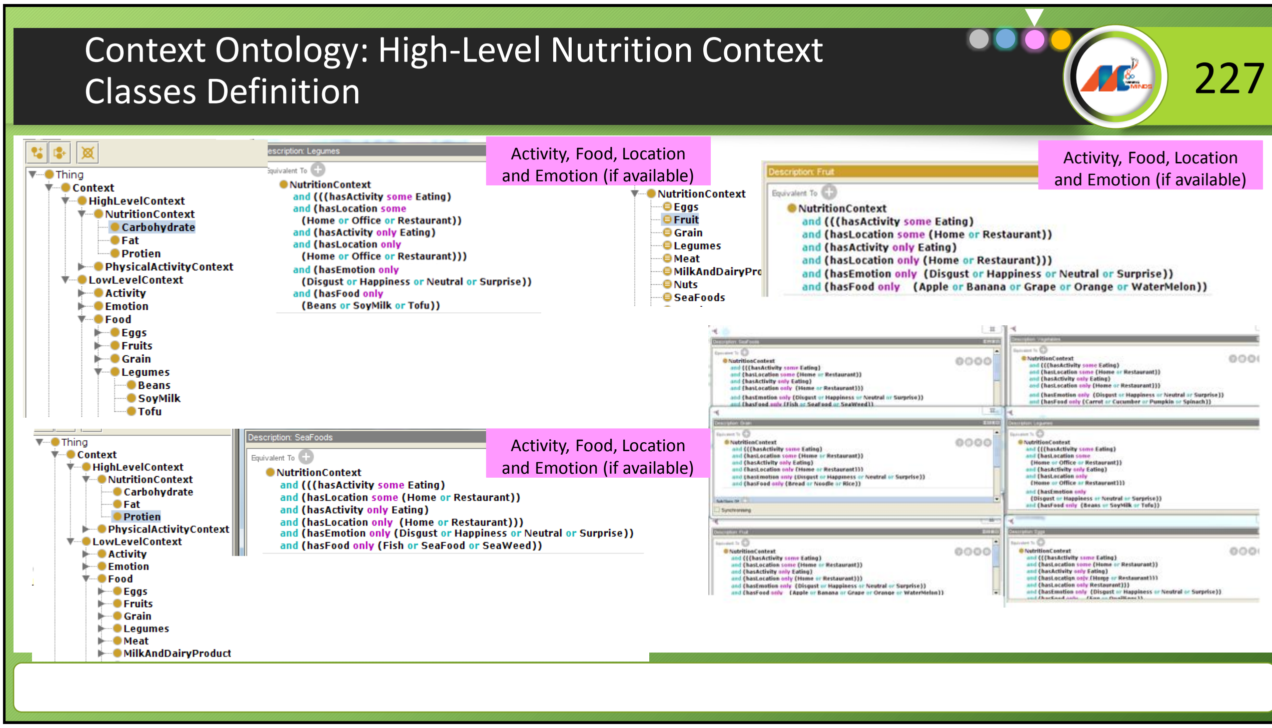
경희대학교
KYUNG HEE UNIVERSITY

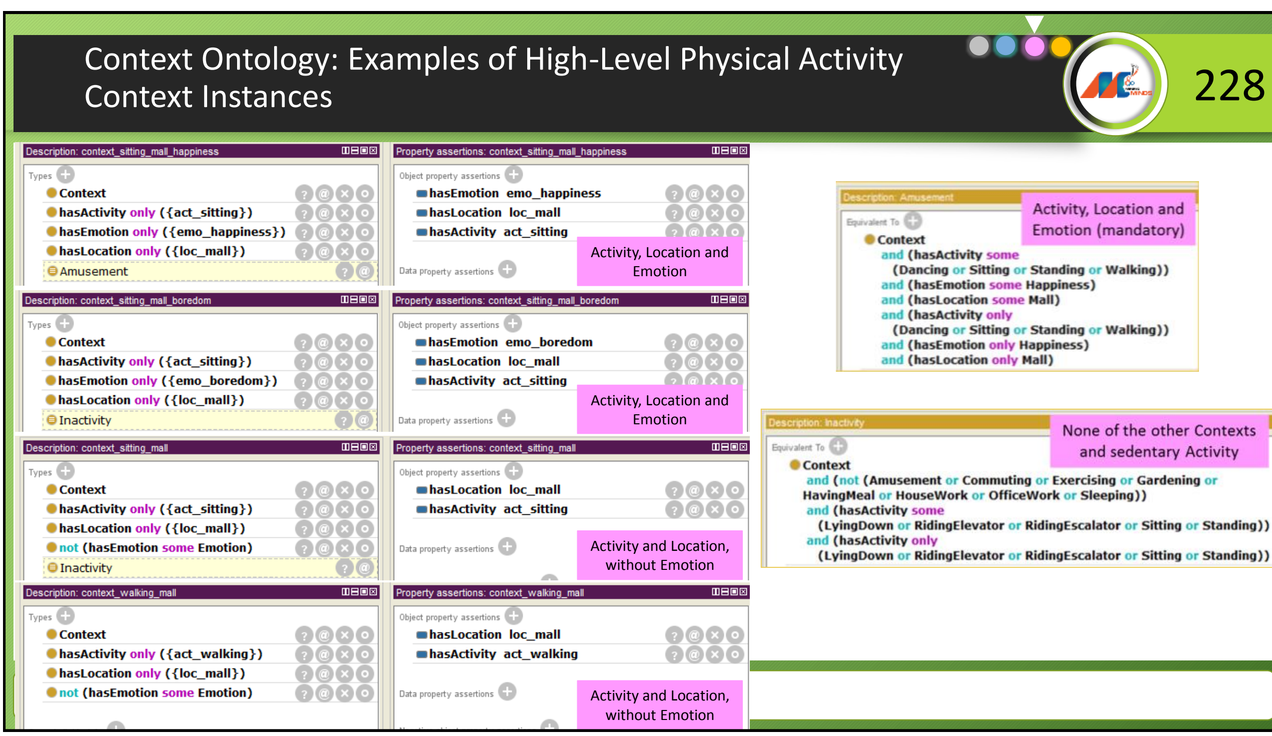

Workflow
223
Low Level Context
Activity: Eating
+
Metadata
Emotion: Happiness
+
Metadata
Location: Restaurant
+
Metadata
Food: Salmon
+
Metadata
High Level Context Builder
1
LLC Mapper
2
LLC Synchronizer
3
Context
Instantiator
Eating
Metadata
Happiness
Metadata
Restaurant
Metadata
Salmon
Metadata
Time
High Level Context-Awareness
High-Level Context Notifier
High-Level Context Reasoner
Context Classifier
Context Verifier
High-Level Context Builder
Context Instantiator
Low-level Context Synchronizer
Low-level Context Mapper
Context Query Manager
Context Handler
경희대학교
KYUNG HEE UNIVERSITY


Workflow
224
High Level Context Reasoner
4
Context Verifier
5
Context Classifier
6
High Level
Context Notifier
Data Curation
Layer
High Level Context-Awareness
High-Level Context Notifier
High-Level Context Reasoner
Context Classifier
Context Verifier
High-Level Context Builder
Context Instantiator
Low-level Context Synchronizer
Low-level Context Mapper
Context Query Manager
Context Handler
경희대학교
KYUNG HEE UNIVERSITY


Context Ontology Model
225
Context Ontology Metrics:
Low Level Contexts
16 Activities
8 Locations
8 Emotions
10 Food Categories (57 Food Items)
Grain (8)
Meat (3)
SeaFood (11)
Eggs (2)
Legumes (3)
Nuts (1)
Fruits (12)
Vegetables (4)
MilkAndDairyProducts (5)
Snacks (8)
High Level Contexts
9 High-Level Physical Activity Contexts
3 High-Level Nutrition Contexts
Ontology modeling tool:
protégé
Protien
Fat
Amusement
HouseWork
Gardening
Inactivity
HavingMeal
Commuting
Sleeping
Exercising
OfficeWork
Subclass
Publication: Villalonga, C., Banos, O., Ali Khan, W., Ali, T., Razzaq, M. A., Lee, S., Pomares, H., Rojas, I. High-Level Context Inference for Human Behavior Identification. International Work-conference on Ambient Assisted Living an Active Ageing (IWAAL 2015), Patagonia, Chile, December 1-4, (2015)

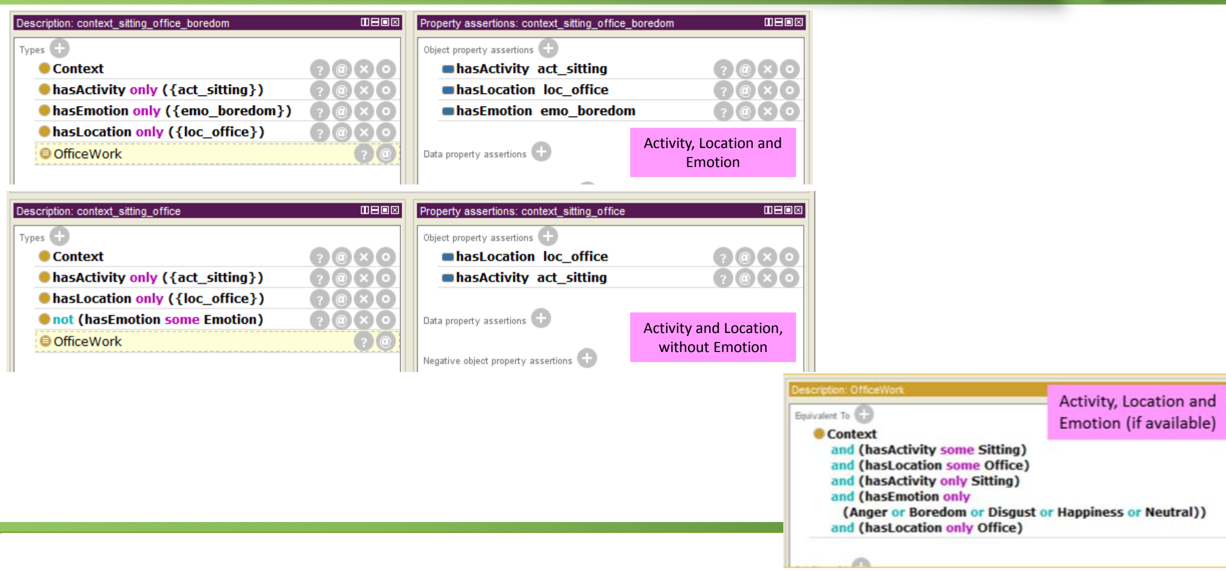

Context Ontology: High-Level Physical Activity Context Classes Definition
226
NutritionContext
PhysicalActivityContext
Amusement
Commuting
Exercising
Gardening
HouseWork
Inactivity
OfficeWork
Sleeping
LowLevelContext
Description: OfficeWork
Equivalent To
PhysicalActivityContext
and (hasActivity some Sitting)
and (hasLocation some Office)
and (hasActivity only Sitting)
and (hasEmotion only
(Anger or Boredom or Disgust or Happiness or Neutral))
and (hasLocation only Office)
Activity, Location and Emotion (if available)
Description: Amusement
Equivalent To
PhysicalActivityContext
and (hasActivity some (Dancing or Sitting or Standing or Walking))
and (hasEmotion some Happiness)
and (hasLocation some Mall)
and (hasActivity only (Dancing or Sitting or Standing or Walking))
and (hasEmotion only Happiness)
and (hasLocation only Mall)
Activity, Location and Emotion (mandatory)
Class hierarchy (inferred)
Commuting
HighLevelContext
NoHLC
NutritionContext
PhysicalActivityContext
Amusement
Commuting
Description: Commuting
Equivalent To
PhysicalActivityContext
and (hasActivity some (Sitting or Standing))
and (hasLocation some Transport)
and (hasActivity only (Sitting or Standing))
and (hasLocation only Transport)
Activity and Location
Description: Inactivity
Equivalent To
PhysicalActivityContext
and (not (Amusement or Commuting or Exercising or Gardening or HouseWork or OfficeWork or Sleeping))
and (hasActivity some
(LyingDown or RidingElevator or RidingEscalator or Sitting or Standing))
and (hasActivity only
(LyingDown or RidingElevator or RidingEscalator or Sitting or Standing))
None of the other Contexts and sedentary Activity


Context Ontology: High-Level Nutrition Context Classes Definition
227
Activity, Food, Location and Emotion (if available)
Activity, Food, Location and Emotion (if available)
Activity, Food, Location and Emotion (if available)
Thing
Context
HighLevelContext
NutritionContext
Carbohydrate
Fat
Protien
PhysicalActivityContext
LowLevelContext
Activity
Emotion
Food
Eggs
Fruits
Grain
Legumes
Beans
SoyMilk
Tofu
Meat
MilkAndDairyProduct
Description: Legumes
NutritionContext
and (((hasActivity some Eating)
and (hasLocation some
(Home or Office or Restaurant))
and (hasActivity only Eating)
and (hasLocation only
(Home or Office or Restaurant)))
and (hasEmotion only
(Disgust or Happiness or Neutral or Surprise))
and (hasFood only
(Beans or SoyMilk or Tofu))
Description: Fruit
NutritionContext
and (((hasActivity some Eating)
and (hasLocation some (Home or Restaurant))
and (hasActivity only Eating)
and (hasLocation only (Home or Restaurant)))
and (hasEmotion only (Disgust or Happiness or Neutral or Surprise))
and (hasFood only (Apple or Banana or Grape or Orange or WaterMelon))
Description: SeaFoods
NutritionContext
and (((hasActivity some Eating)
and (hasLocation some (Home or Restaurant))
and (hasActivity only Eating)
and (hasLocation only (Home or Restaurant)))
and (hasEmotion only (Disgust or Happiness or Neutral or Surprise))
and (hasFood only (Fish or SeaFood or SeaWeed))


Context Ontology: Examples of High-Level Physical Activity Context Instances
228
Description: context_sitting_mall_happiness
Context
hasActivity only ({act_sitting})
hasEmotion only ({emo_happiness})
hasLocation only ({loc_mall})
Amusement
Property assertions: context_sitting_mall_happiness
hasEmotion emo_happiness
hasLocation loc_mall
hasActivity act_sitting
Activity, Location and Emotion
Description: context_sitting_mall_boredom
Context
hasActivity only ({act_sitting})
hasEmotion only ({emo_boredom})
hasLocation only ({loc_mall})
Inactivity
Property assertions: context_sitting_mall_boredom
hasEmotion emo_boredom
hasLocation loc_mall
hasActivity act_sitting
Activity, Location and Emotion
Description: context_sitting_mall
Context
hasActivity only ({act_sitting})
hasLocation only ({loc_mall})
not (hasEmotion some Emotion)
Inactivity
Property assertions: context_sitting_mall
hasLocation loc_mall
hasActivity act_sitting
Activity and Location, without Emotion
Description: context_walking_mall
Context
hasActivity only ({act_walking})
hasLocation only ({loc_mall})
not (hasEmotion some Emotion)
Property assertions: context_walking_mall
hasLocation loc_mall
hasActivity act_walking
Activity and Location, without Emotion
Description: Amusement
Activity, Location and Emotion (mandatory)
Context
and (hasActivity some
(Dancing or Sitting or Standing or Walking))
and (hasEmotion some Happiness)
and (hasLocation some Mall)
and (hasActivity only
(Dancing or Sitting or Standing or Walking))
and (hasEmotion only Happiness)
and (hasLocation only Mall)
Description: Inactivity
None of the other Contexts and sedentary Activity
Context
and (not (Amusement or Commuting or Exercising or Gardening or HavingMeal or HouseWork or OfficeWork or Sleeping))
and (hasActivity some
(LyingDown or RidingElevator or RidingEscalator or Sitting or Standing))
and (hasActivity only
(LyingDown or RidingElevator or RidingEscalator or Sitting or Standing))

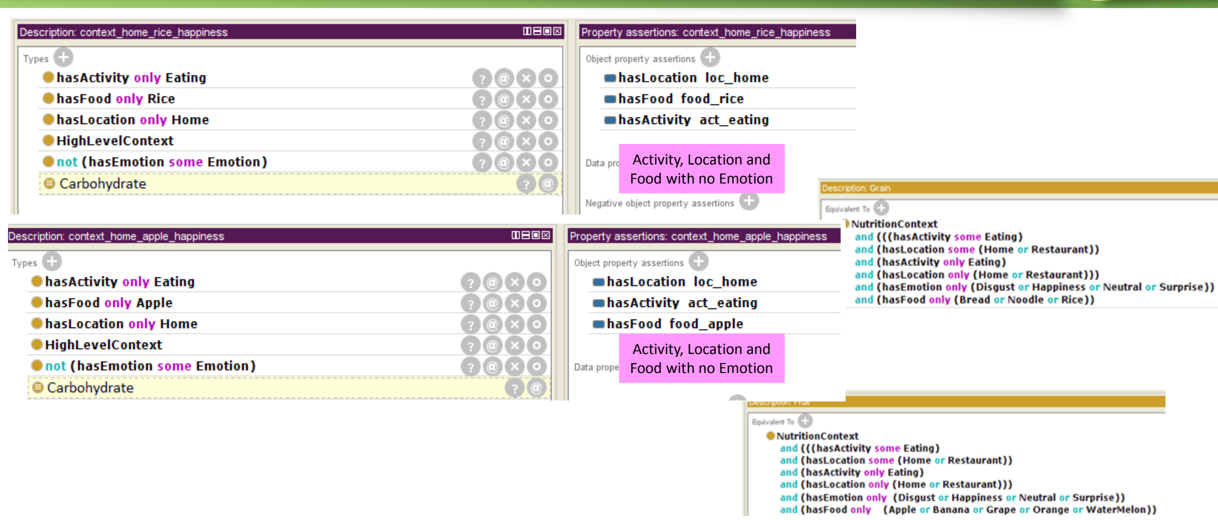
Context Ontology: Examples of High-Level Physical Activity Context Instances

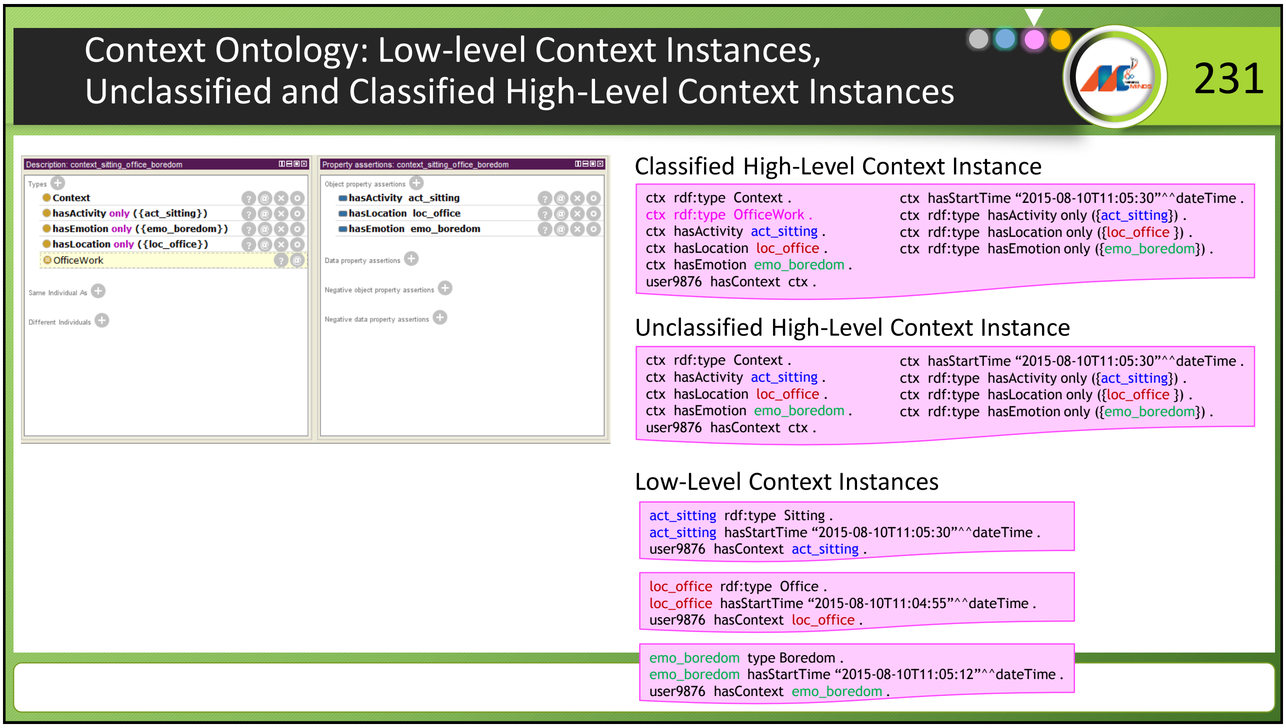
229

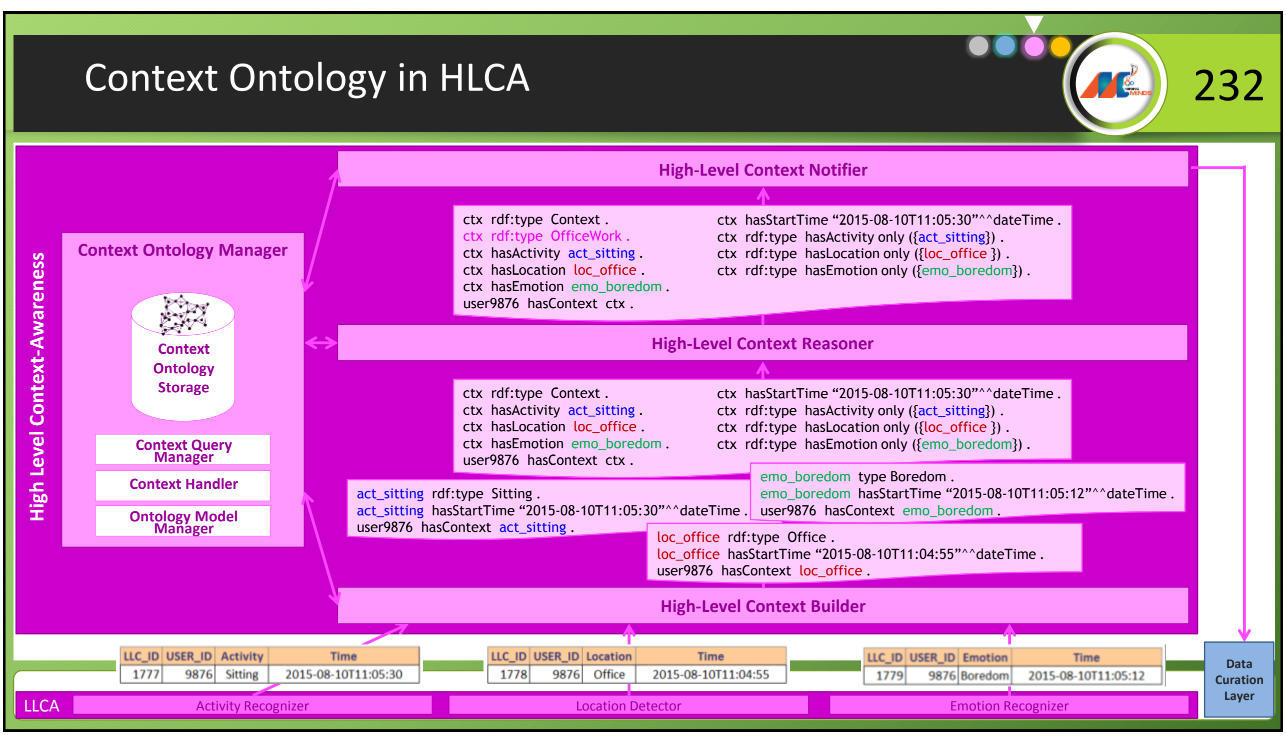


Context Ontology: Examples of High-Level Nutrition Context Instances

230

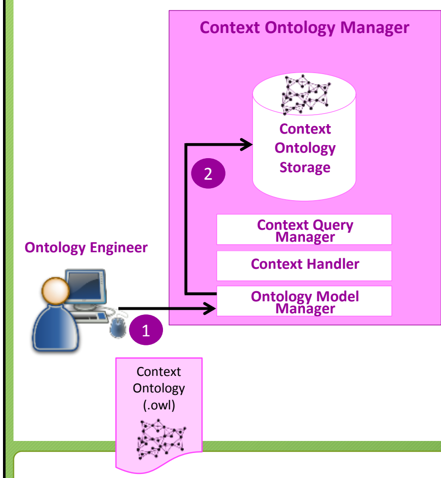



Context Ontology: Low-level Context Instances, Unclassified and Classified High-Level Context Instances
231
Description: context_sitting_office_boredom
Types
Context
hasActivity only ({act_sitting})
hasEmotion only ({emo_boredom})
hasLocation only ({loc_office})
OfficeWork
Same Individual As
Different Individuals
Property assertions: context_sitting_office_boredom
Object property assertions
hasActivity act_sitting
hasLocation loc_office
hasEmotion emo_boredom
Data property assertions
Negative object property assertions
Negative data property assertions
Classified High-Level Context Instance
ctx rdf:type Context .
ctx rdf:type OfficeWork .
ctx hasActivity act_sitting .
ctx hasLocation loc_office .
ctx hasEmotion emo_boredom .
user9876 hasContext ctx .
ctx hasStartTime “2015-08-10T11:05:30”^^dateTime .
ctx rdf:type hasActivity only ({act_sitting}) .
ctx rdf:type hasLocation only ({loc_office }) .
ctx rdf:type hasEmotion only ({emo_boredom}) .
Unclassified High-Level Context Instance
ctx rdf:type Context .
ctx hasActivity act_sitting .
ctx hasLocation loc_office .
ctx hasEmotion emo_boredom .
user9876 hasContext ctx .
ctx hasStartTime “2015-08-10T11:05:30”^^dateTime .
ctx rdf:type hasActivity only ({act_sitting}) .
ctx rdf:type hasLocation only ({loc_office }) .
ctx rdf:type hasEmotion only ({emo_boredom}) .
Low-Level Context Instances
act_sitting rdf:type Sitting .
act_sitting hasStartTime “2015-08-10T11:05:30”^^dateTime .
user9876 hasContext act_sitting .
loc_office rdf:type Office .
loc_office hasStartTime “2015-08-10T11:04:55”^^dateTime .
user9876 hasContext loc_office .
emo_boredom type Boredom .
emo_boredom hasStartTime “2015-08-10T11:05:12”^^dateTime .
user9876 hasContext emo_boredom .


Context Ontology in HLCA
232
High Level Context-Awareness
Context Ontology Manager
Context Ontology Storage
Context Query Manager
Context Handler
Ontology Model Manager
High-Level Context Notifier
ctx rdf:type Context .
ctx rdf:type OfficeWork .
ctx hasActivity act_sitting .
ctx hasLocation loc_office .
ctx hasEmotion emo_boredom .
user9876 hasContext ctx .
ctx hasStartTime “2015-08-10T11:05:30”^^dateTime .
ctx rdf:type hasActivity only ({act_sitting}) .
ctx rdf:type hasLocation only ({loc_office }) .
ctx rdf:type hasEmotion only ({emo_boredom}) .
High-Level Context Reasoner
ctx rdf:type Context .
ctx hasActivity act_sitting .
ctx hasLocation loc_office .
ctx hasEmotion emo_boredom .
user9876 hasContext ctx .
ctx hasStartTime “2015-08-10T11:05:30”^^dateTime .
ctx rdf:type hasActivity only ({act_sitting}) .
ctx rdf:type hasLocation only ({loc_office }) .
ctx rdf:type hasEmotion only ({emo_boredom}) .
emo_boredom type Boredom .
emo_boredom hasStartTime “2015-08-10T11:05:12”^^dateTime .
user9876 hasContext emo_boredom .
act_sitting rdf:type Sitting .
act_sitting hasStartTime “2015-08-10T11:05:30”^^dateTime .
user9876 hasContext act_sitting .
loc_office rdf:type Office .
loc_office hasStartTime “2015-08-10T11:04:55”^^dateTime .
user9876 hasContext loc_office .
High-Level Context Builder
LLC_ID USER_ID Activity Time
1777 9876 Sitting 2015-08-10T11:05:30
LLC_ID USER_ID Location Time
1778 9876 Office 2015-08-10T11:04:55
LLC_ID USER_ID Emotion Time
1779 9876 Boredom 2015-08-10T11:05:12
Data Curation Layer
LLCA
Activity Recognizer
Location Detector
Emotion Recognizer

Context Ontology Manager

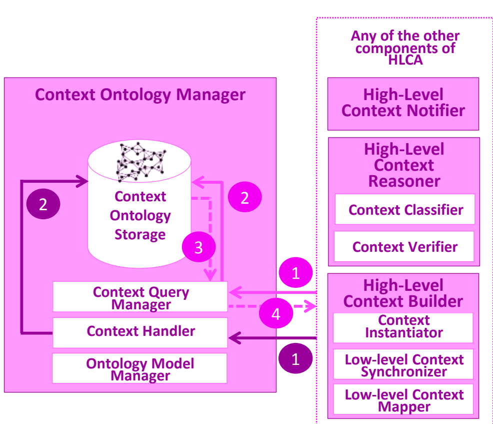
233



- **Context Ontology Storage**
  - Persist the Context Ontology Model and Context Instances
  - Technology: Triplestore
  - Tool (to be confirmed) : TDB of Jena
- **Ontology Model Manager**
  - Load into MM the Context Ontology Model
  - Workflow:
    - 1 Receive Context Ontology Model (in OWL format) generated by the ontology engineer
    - 2 Store the Context Ontology Model into the triplestore
  - Tool: Jena (including RDF API and Ontology API)

Context Ontology Manager

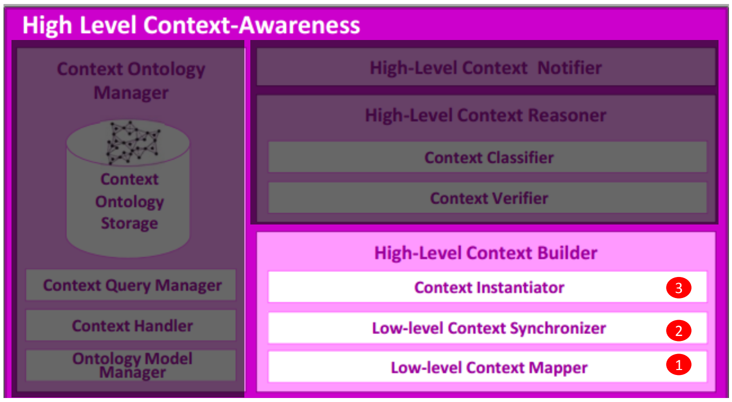
234



- **Context Handler**
  - Add/Delete/Update Context Instances
  - Workflow:
    - 1 Receive Context Instance
    - 2 Store the Context Ontology Model into the triplestore
  - Tool: Jena (including Ontology API)
- **Context Query Manager**
  - Retrieve Context Ontology Model and Context Instances
  - Workflow:
    - 1 Receive query for specific context
    - 2 Generate SPARQL query and pose it to the triplestore
    - 3 4 Return results
  - Tool: Jena (including ARQ)

High Level Context Builder

235



Low Level Context Mapper

236

Low Level Context Instances

LLC_ID	USER_ID	Activity	Time
1777	9876	Sitting	2015-08-10T11:05:30

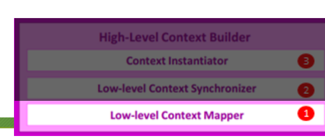
LLC_ID	USER_ID	Location	Time
1778	9876	Office	2015-08-10T11:04:55

LLC_ID	USER_ID	Emotion	Time
1779	9876	Boredom	2015-08-10T11:05:12

act_sitting rdf:type Sitting .
act_sitting hasStartTime “2015-08-10T11:05:30”^^dateTime .
user9876 hasContext act_sitting .

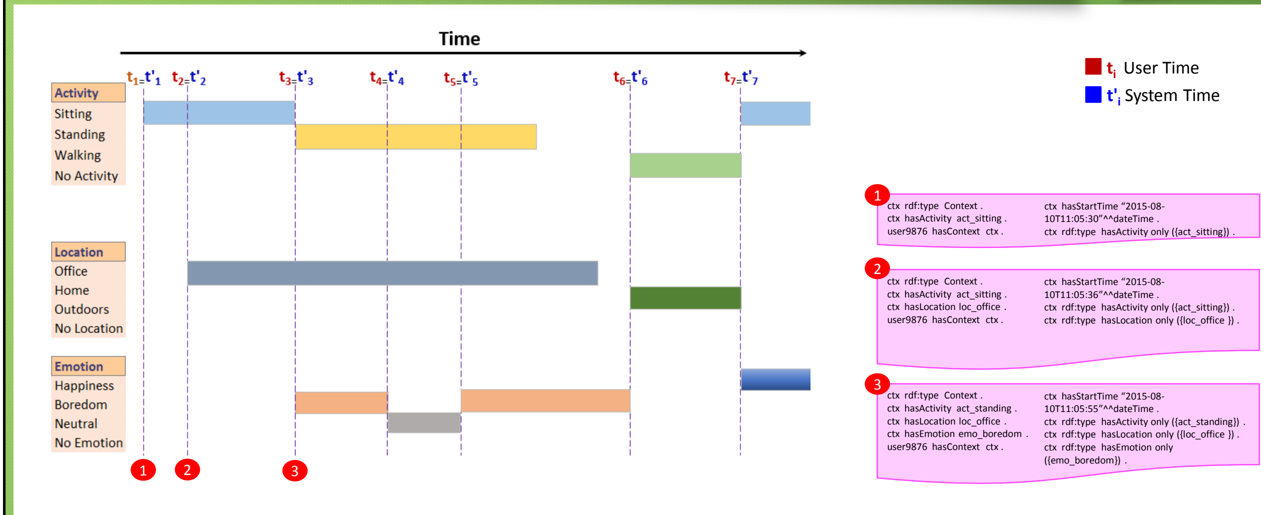
loc_office rdf:type Office .
loc_office hasStartTime “2015-08-10T11:04:55”^^dateTime .
user9876 hasContext loc_office .

emo_boredom type Boredom .
emo_boredom hasStartTime “2015-08-10T11:05:12”^^dateTime .
user9876 hasContext emo_boredom .



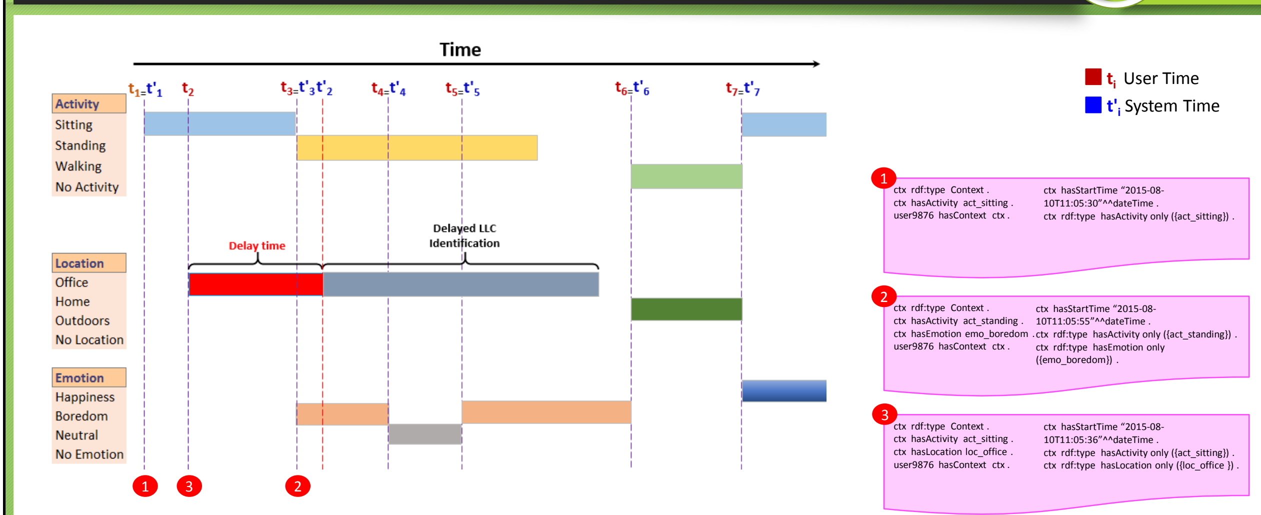
Low Level Context Synchronization

237



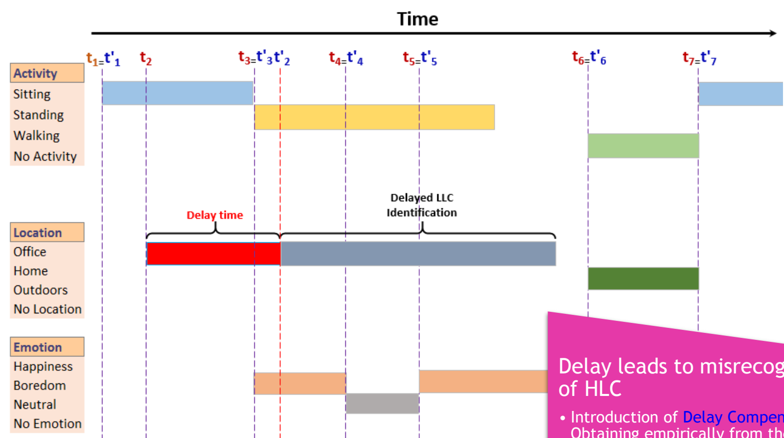
Low Level Context Synchronization

238



Low Level Context Synchronization

239



t_i User Time

t'_i System Time

Delay leads to misrecognition of HLC

- Introduction of Delay Compensation by Obtaining empirically from the analysis of the LLC Estimated identification time

Rearrangement of unidentified Low level context for Unclassified HLC

- Queuing Mechanism

Context Instantiator

240

Low Level Context Instances

LLC_ID	USER_ID	Activity	Time
1777	9876	Sitting	2015-08-10T11:05:30

LLC_ID	USER_ID	Location	Time
1778	9876	Office	2015-08-10T11:04:55

LLC_ID	USER_ID	Emotion	Time
1779	9876	Boredom	2015-08-10T11:05:12

act_sitting rdf:type Sitting .
act_sitting hasStartTime “2015-08-10T11:05:30”^^dateTime .
user9876 hasContext act_sitting .

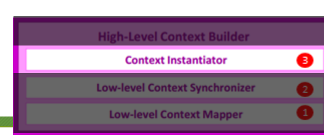
loc_office rdf:type Office .
loc_office hasStartTime “2015-08-10T11:04:55”^^dateTime .
user9876 hasContext loc_office .

emo_boredom type Boredom .
emo_boredom hasStartTime “2015-08-10T11:05:12”^^dateTime .
user9876 hasContext emo_boredom .

Unclassified High Level Context Instance

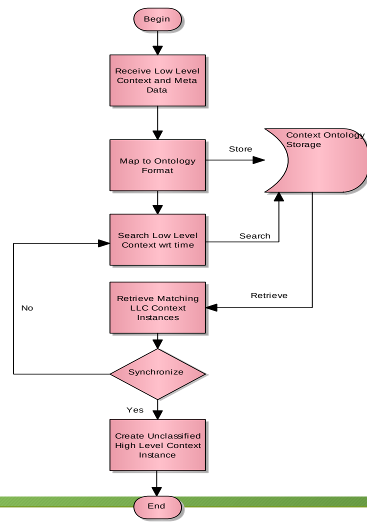
ctx rdf:type Context .
ctx hasActivity act_sitting .
ctx hasLocation loc_office .
ctx hasEmotion emo_boredom .
user9876 hasContext ctx .

ctx hasStartTime “2015-08-10T11:05:30”^^dateTime .
ctx rdf:type hasActivity only ({act_sitting}) .
ctx rdf:type hasLocation only ({loc_office }) .
ctx rdf:type hasEmotion only ({emo_boredom}) .



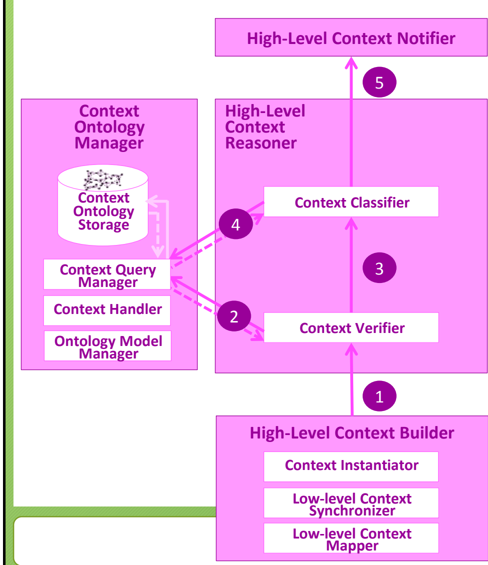
High-Level Context Builder Flowchart

241



High-level Context Reasoner

242



- **Context Verifier**
  - Check the consistency of the unclassified high-level context instance versus the context ontology model
  - Workflow:
    1. Receive an unclassified high-level context instance

       ctx rdf:type Context .
       ctx hasActivity act_sitting .
       ctx hasLocation loc_office .
       ctx hasEmotion emo_boredom .
       user9876 hasContext ctx .
       ctx hasStartTime "2015-08-10T11:05:30"^^dateTime .
       ctx rdf:type hasActivity only ({act_sitting}) .
       ctx rdf:type hasLocation only ({loc_office }) .
       ctx rdf:type hasEmotion only ({emo_boredom}) .

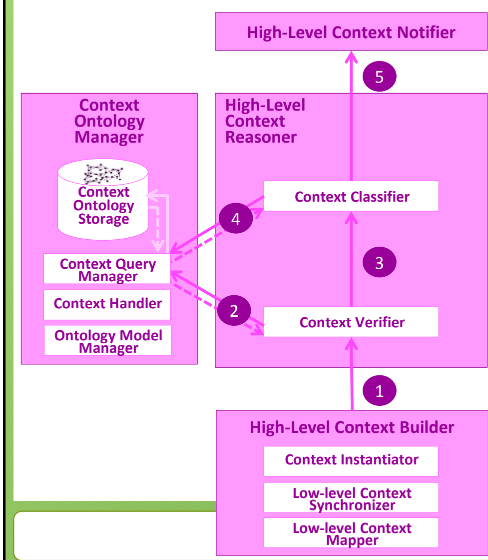
    2. Retrieve the context ontology model
    3. Verify the consistency of unclassified high-level context instance. If the unclassified high-level context instance is valid, serve it to the Context Classifier

       ctx rdf:type Context .
       ctx hasActivity act_sitting .
       ctx hasLocation loc_office .
       ctx hasEmotion emo_boredom .
       user9876 hasContext ctx .
       ctx hasStartTime "2015-08-10T11:05:30"^^dateTime .
       ctx rdf:type hasActivity only ({act_sitting}) .
       ctx rdf:type hasLocation only ({loc_office }) .
       ctx rdf:type hasEmotion only ({emo_boredom}) .

- Tools: Jena (including Inference API) & Pellet Reasoner

High-level Context Reasoner

243



- **Context Classifier**
  - Identify the context type to which the unclassified high-level context instance belongs
  - Workflow:

3 Receive a verified unclassified high-level context instance

```
ctx rdf:type Context .
ctx hasActivity act_sitting .
ctx hasLocation loc_office .
ctx hasEmotion emo_boredom .
user9876 hasContext ctx .
ctx hasStartTime "2015-08-10T11:05:30"^^dateTime .
ctx rdf:type hasActivity only ({act_sitting}) .
ctx rdf:type hasLocation only ({loc_office }) .
ctx rdf:type hasEmotion only ({emo_boredom}) .
```

4 Retrieve the context ontology model

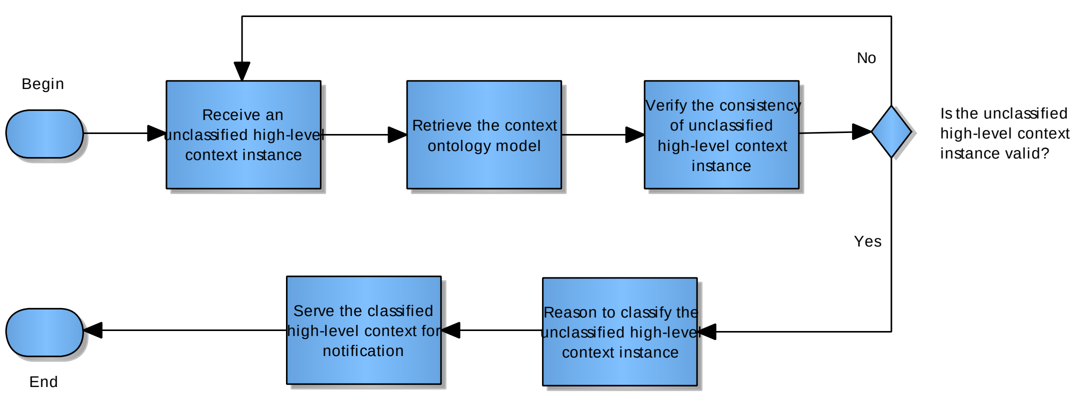
5 Reason to classify the unclassified high-level context instance and serve the classified high-level context for notification

```
ctx rdf:type Context .
ctx rdf:type OfficeWork .
ctx hasActivity act_sitting .
ctx hasLocation loc_office .
ctx hasEmotion emo_boredom .
user9876 hasContext ctx .
ctx hasStartTime "2015-08-10T11:05:30"^^dateTime .
ctx rdf:type hasActivity only ({act_sitting}) .
ctx rdf:type hasLocation only ({loc_office }) .
ctx rdf:type hasEmotion only ({emo_boredom}) .
```

- Tools: Jena (including Inference API) & Pellet Reasoner

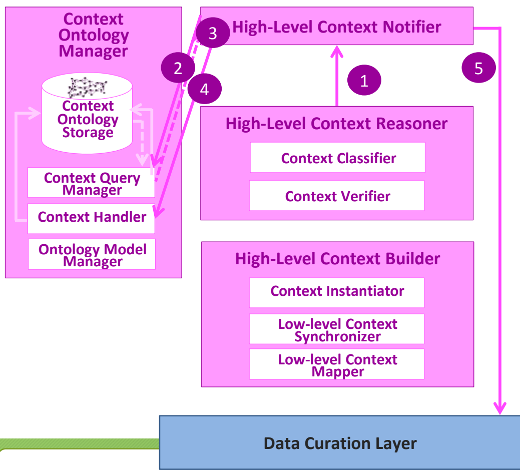
High-level Context Reasoner Flowchart

244



High-Level Context Notifier

245



- **Context Notifier**
  - Communicate the newly recognized high-level context to Data Curation Layer for storage into the LifeLog
  - Workflow:
    1. Receive a classified high-level context instance

```
ctx rdf:type Context .
ctx rdf:type OfficeWork .
ctx hasActivity act_sitting .
ctx hasLocation loc_office .
ctx hasEmotion emo_boredom .
user9876 hasContext ctx .
ctx hasStartTime "2015-08-10T11:05:30"^^dateTime .
ctx rdf:type hasActivity only ({act_sitting}) .
ctx rdf:type hasLocation only ({loc_office }) .
ctx rdf:type hasEmotion only ({emo_boredom}) .
```

    2. Retrieve previous high-level context instance

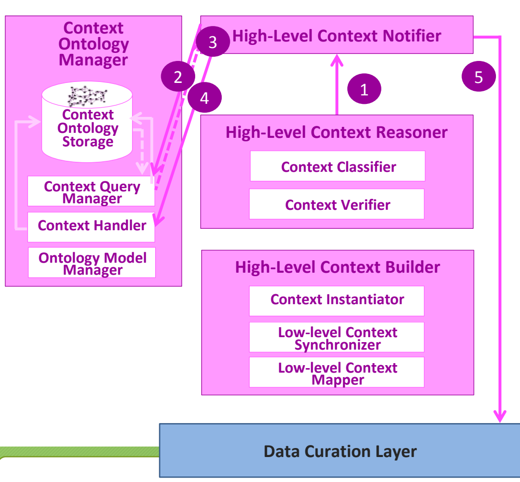
```
ctx0 rdf:type Context .
ctx0 rdf:type Inactivity .
ctx0 hasActivity act_lyingdown .
ctx0 hasLocation loc_office .
ctx0 hasEmotion emo_boredom .
user9876 hasContext ctx0 .
ctx0 hasStartTime "2015-08-10T11:04:12"^^dateTime .
ctx0 rdf:type hasActivity only ({act_lyingdown}) .
ctx0 rdf:type hasLocation only ({loc_office }) .
ctx0 rdf:type hasEmotion only ({emo_boredom}) .
```

    3. Add end time to previous high-level context instance

```
ctx0 rdf:type Context .
ctx0 rdf:type Inactivity .
ctx0 hasActivity act_lyingdown .
ctx0 hasLocation loc_office .
ctx0 hasEmotion emo_boredom .
user9876 hasContext ctx0 .
ctx0 hasStartTime "2015-08-10T11:04:12"^^dateTime .
ctx0 hasEndTime "2015-08-10T11:05:30"^^dateTime .
ctx0 rdf:type hasActivity only ({act_lyingdown}) .
ctx0 rdf:type hasLocation only ({loc_office }) .
ctx0 rdf:type hasEmotion only ({emo_boredom}) .
```

High-Level Context Notifier

246



- **Context Notifier**
  - Communicate the newly recognized high-level context to Data Curation Layer for storage into the LifeLog
  - Workflow:
    4. Store the two high-level context instances

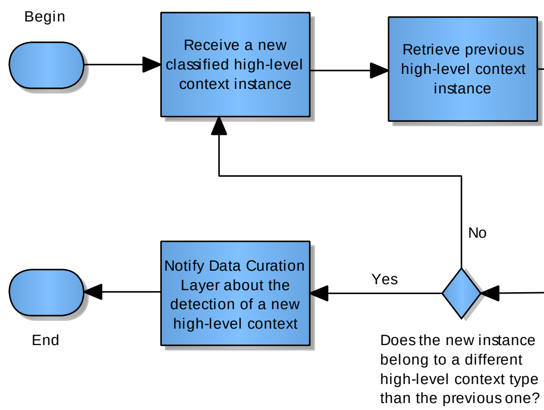
```
ctx rdf:type Context .
ctx rdf:type OfficeWork .
ctx hasActivity act_sitting .
ctx hasLocation loc_office .
ctx hasEmotion emo_boredom .
user9876 hasContext ctx .
ctx hasStartTime "2015-08-10T11:05:30"^^dateTime .
ctx rdf:type hasActivity only ({act_sitting}) .
ctx rdf:type hasLocation only ({loc_office }) .
ctx rdf:type hasEmotion only ({emo_boredom}) .
```

```
ctx0 rdf:type Context .
ctx0 rdf:type Inactivity .
ctx0 hasActivity act_lyingdown .
ctx0 hasLocation loc_office .
ctx0 hasEmotion emo_boredom .
user9876 hasContext ctx0 .
ctx0 hasStartTime "2015-08-10T11:04:12"^^dateTime .
ctx0 hasEndTime "2015-08-10T11:05:30"^^dateTime .
ctx0 rdf:type hasActivity only ({act_lyingdown}) .
ctx0 rdf:type hasLocation only ({loc_office }) .
ctx0 rdf:type hasEmotion only ({emo_boredom}) .
```

    5. If the new instance belongs to a different high-level context type than the previous one, notify Data Curation Layer about the detection of a new high-level context
- Tool: Jena (including RDF API and Ontology API)

High-Level Context Notifier Flowchart

247



Technologies for Data Representation

248

- Ontologies are **formalized vocabularies** of terms, often covering a specific **domain** and **shared** by a community of users
  - Vocabulary of terms plus explicit characterizations of the assumptions made in interpreting those terms
  - Include some notion of hierarchical classification (is-a)
  - Definition of classes through description of their characteristics
  - Possibility of using inference to help in management and deployment of the knowledge
- **Why ontologies?**
  - The semantics of a language can tell us precisely how to interpret a complex expression
  - Ontologies have formal semantics which provide an **unambiguous** interpretation of the descriptions

Technologies for Data Representation

- Standard Ontology Languages
  - **RDF** is a framework for representing information in the Web.
    RDF graphs are sets of subject-predicate-object triples used to express descriptions of resources.
  - **RDF Schema** is a semantic extension of RDF.
    RDFS provides mechanisms for describing groups of related resources and the relationships between these resources.
  - **OWL** is an ontology language for the Semantic Web with formally defined meaning.
    OWL adds an additional layer of semantics on top of RDF

Standard	Features	Limitations or disadvantages
RDF 1.1	Assert statements (rdf:Statement and rdf:subject, rdf:predicate, rdf:object)	Very, very restricted vocabulary No inference
RDF Schema	Define classes (rdfs:Class) and their hierarchy (rdfs:subClassOf) Define properties (rdfs:Property) and their hierarchy (rdfs:subPropertyOf)	Restricted vocabulary No rigid structure, i.e., no constraints
OWL 2	Describe data in terms of set operations (owl:unionOf) Define equivalences (owl:sameAs) Restrict property values (owl:allValuesFrom) Define annotations or meta-meta-data (owl:deprecatedProperty)	More complex ontology

RDF 1.1 (Resource Description Framework) http://www.w3.org/TR/rdf11-concepts/
RDFS (RDF Schema) http://www.w3.org/TR/rdf-schema/
OWL 2 (OWL 2 Web Ontology Language) http://www.w3.org/TR/owl2-overview/

Technologies for Data Representation

- OWL 2 Syntaxes: needed in order to store OWL 2 ontologies and to exchange them among tools and applications

Syntax	Standard Status	Purpose
RDF/XML	Mandatory	Interoperability among all OWL 2 tools
OWL/XML	Optional	Easier to process using XML tools
Functional Syntax	Optional	Easier to see the formal structure of ontologies
Manchester Syntax	Optional	Easier to read/write DL Ontologies
Turtle	Optional	Easier to read/write RDF triples

- OWL 2 Semantics: ways of assigning meaning to OWL 2 ontologies, which are used by reasoners and other tools

Semantics	Name	Advantage	Disadvantage
Direct Semantics	OWL 2 DL	Compatible with the semantics of SROIQ Description Logic (FOL) Decidable	Restrictions on some ontology structures Less expressiveness
RDF-Based Semantics	OWL 2 Full	No restrictions Expressiveness	Undecidable

OWL 2 (OWL 2 Web Ontology Language) http://www.w3.org/TR/owl2-overview/

Technologies for Data Representation

- OWL 2 Profiles: subset of the structural elements in an ontology

Profile	Supported Features	Suitable for …	Benefits
OWL 2 EL	Polynomial time algorithms for standard reasoning	Very large ontologies	Higher performance as a tradeoff for the lower expressive power
OWL 2 QL	Conjunctive queries using standard relational DB technology	Lightweight ontologies with large numbers of individuals	Access the data directly via relational queries (e.g., SQL)
OWL 2 RL	Polynomial time algorithms for reasoning using rule-extended database technologies operating on RDF triples	Lightweight ontologies with large numbers of individuals	Operate directly on data in the form of RDF triples

To be decided!

(!) OWL 2 EL, OWL 2 QL, and OWL 2 RL are even more restrictive than OWL DL

- Standard Ontology Query Language
  - **SPARQL** is a set of specifications that provide languages and protocols to query and manipulate RDF graph content on the Web or in an RDF store

OWL 2 (OWL 2 Web Ontology Language) http://www.w3.org/TR/owl2-overview/
SPARQL 1.1 (SPARQL Query Language for RDF) http://www.w3.org/TR/sparql11-overview/

Tools for Ontology Engineering

Tool	Description
protégé	Protégé 4 supports OWL 2 on top of the OWL API. It enables users to load and save OWL ontologies, edit and visualize classes and properties, and check the ontology using an OWL reasoner.
Swoop	SWOOP is a tool for creating, editing, and debugging OWL ontologies. It is an open source project.
NeOn Toolkit	The NeOn Toolkit is an Open Source ontology engineering suite developed in the popular Eclipse environment. Thanks to its modular design and a rich choice of plug-ins, the NeOn Toolkit not only allows editing of ontologies but also provides a variety of leading-edge functionalities, including support for modularization, consistency checking and debugging, alignments and mapping, DB integration, as well as several novel means for visualizing and navigating large ontologies and ontology networks. In addition, it has a unique built-in support for deploying ontology design patterns and for managing ontology development projects, in accordance with the procedures and methods specified in the NeOn Methodology.

protégé

http://www.w3.org/2001/sw/wiki/OWL/Implementations
http://www.w3.org/wiki/Ontology_editors

Ontology Frameworks and APIs

253

API	Description
OWL API	The OWL API is an API for working with OWL 2 ontologies. It comes with a Java based reference implementation. The OWL API reference implementation includes parsers and renderers for RDF/XML, OWL/XML, Turtle, and Manchester OWL Syntax. It provides a standard interface to various OWL reasoners

Framework	Description
Jena	Jena, a Java RDF API and toolkit (triple store, programming environment, reasoner, rule reasoner, owl reasoner, rdfs reasoner, parser). Directly usable from Java
Sesame	Sesame is a powerful Java framework for processing and handling RDF data. This includes creating, parsing, storing, inferencing and querying over such data. It offers an easy-to-use API that can be connected to all leading RDF storage solutions

Tools for Ontological Reasoning

254

Tool	Native Profiles	Semantics	(Non-) Conformance	Description
CEL	OWL EL	Direct	Lacks support for nominals (ObjectHasValue and ObjectOneOf) and datatypes/values.	CEL is an open-source polynomial-time Classifier for the OWL 2 EL profile. It has demonstrated scalability and proved well suited for several biomedical ontologies.
FaCT++	OWL DL	Direct	Fully conformant except for keys and some datatypes (coming soon).	FaCT++ is an open-source tableaux-based OWL 2 DL reasoner. It is implemented in C++ and shows exceptional performance on expressive ontologies.
HermiT	OWL DL	Direct	Fully conformant	Based on a novel "hypertableau" algorithm, HermiT can determine whether or not the input ontology is consistent, identify subsumption relationships between classes, and much more.
Pellet	OWL DL, EL	Direct	Fully conformant	Pellet is an open source reasoner for OWL 2 DL in Java. It provides standard and cutting-edge reasoning services for OWL ontologies.
RacerPro	OWL DL	Direct		RacerPro is a commercial (but free for research) OWL reasoner and inference server.

Since it supports SWRL Built-in atoms

http://www.w3.org/2001/sw/wiki/OWL/Implementations

Tools and Technologies

255

- Data Representation
  - Ontological representation (OWL2)
- Ontology Query
  - SPARQL Queries
- Ontology Engineering
  - Protégé
- Ontological Reasoner
  - Pellet
- Ontology Storage
  - TDB (Triplestore)
- Implementation Tools and Technologies
  - Jena API (Semantic Web Framework)
  - Netbeans (Java)
  - SQL Server

경희대학교
KYUNG HEE UNIVERSITY

Contributions

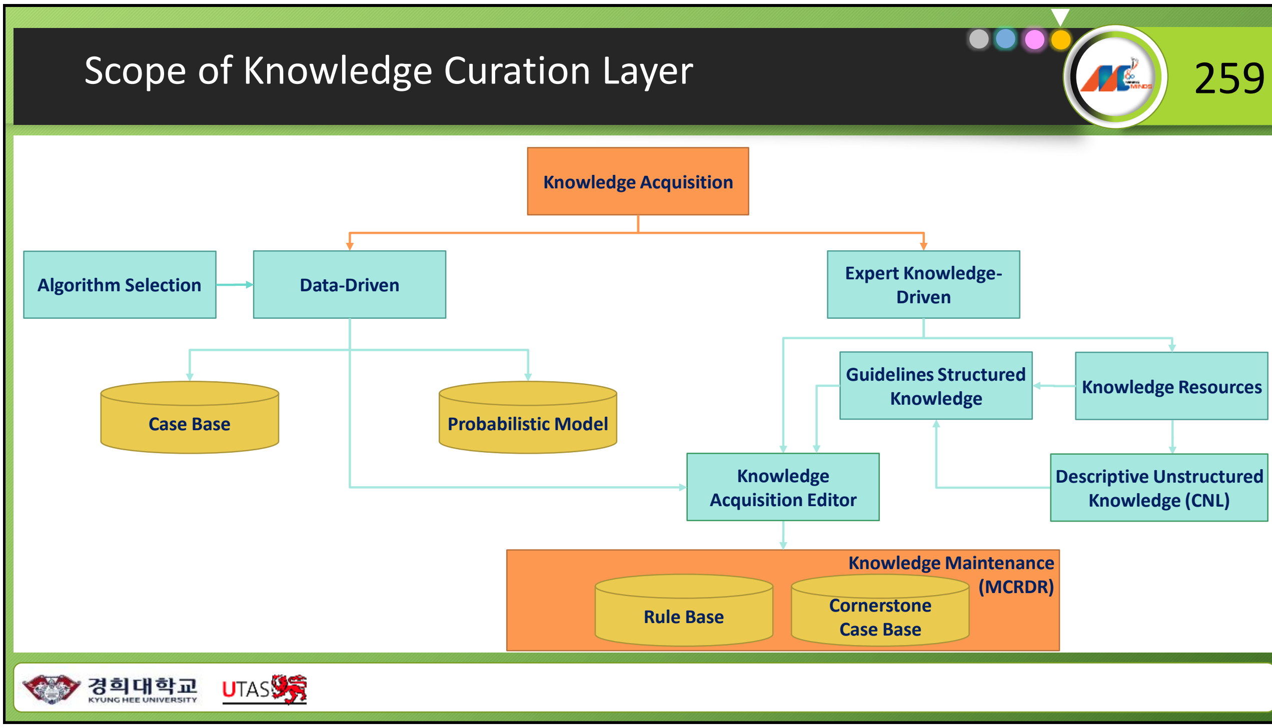
256

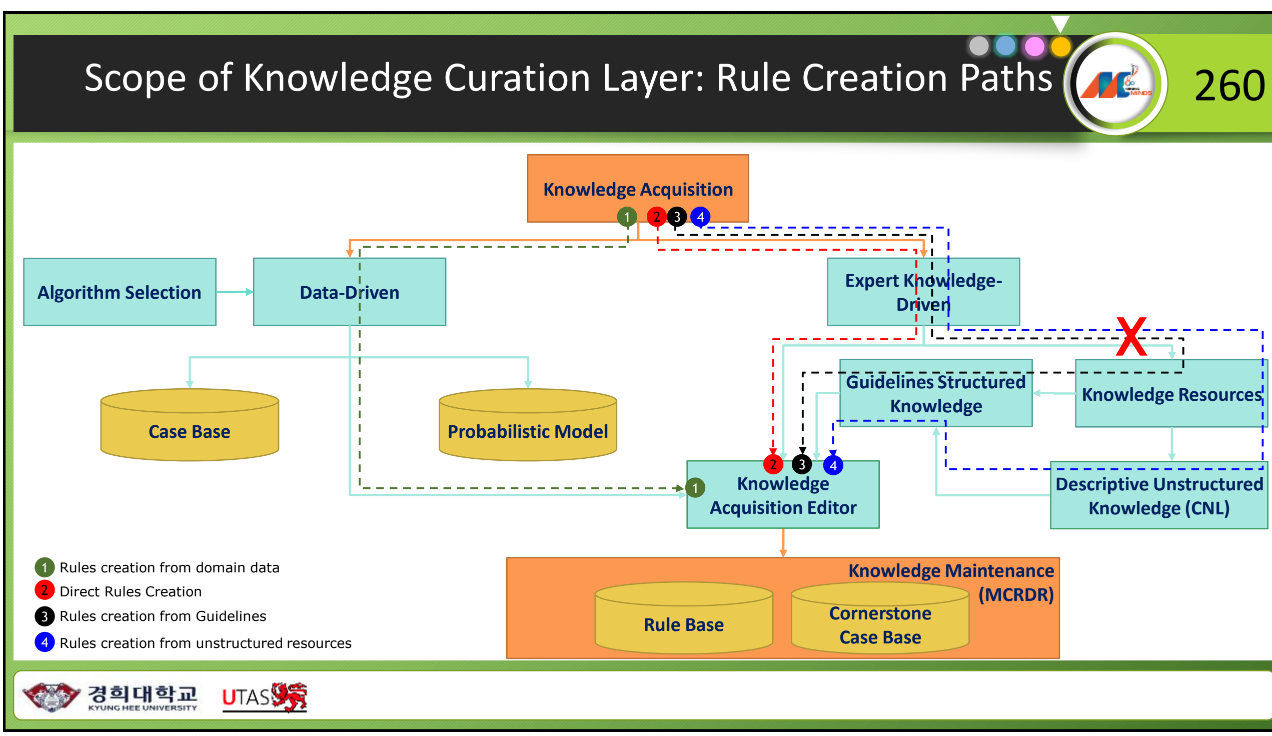
- Engineering of an ontology for context definition
- Analysis of the best modeling principles
- Investigation of ontological reasoners
- Identification of the issues associated to low-level and high-level context synchronization
- Analysis and identification of the technologies required to implement high-level context

경희대학교
KYUNG HEE UNIVERSITY

Knowledge Curation Layer (KCL 2.5)

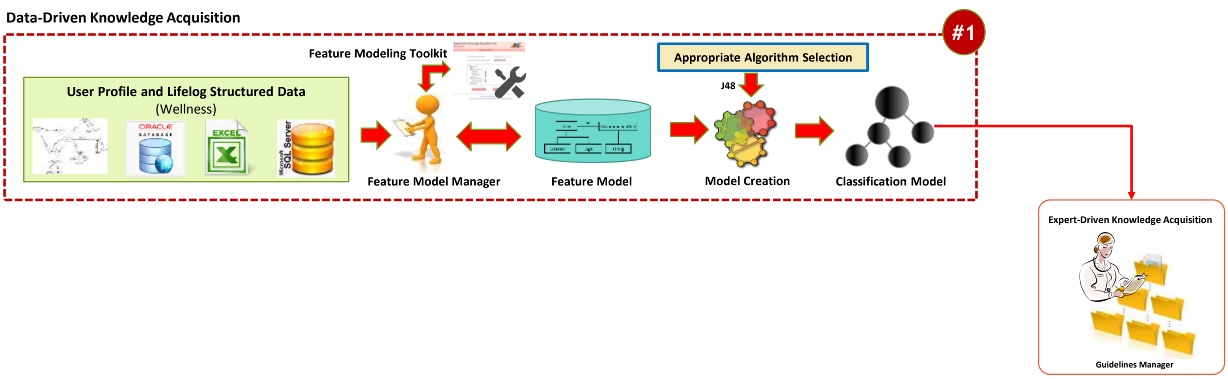
Knowledge Creation & Evolution


Scope of Knowledge Curation Layer
259
Knowledge Acquisition
Algorithm Selection
Data-Driven
Expert Knowledge-Driven
Case Base
Probabilistic Model
Guidelines Structured Knowledge
Knowledge Resources
Knowledge Acquisition Editor
Descriptive Unstructured Knowledge (CNL)
Knowledge Maintenance (MCRDR)
Rule Base
Cornerstone Case Base
경희대학교
KYUNG HEE UNIVERSITY
UTAS


Scope of Knowledge Curation Layer: Rule Creation Paths
260
Knowledge Acquisition
1 2 3 4
Algorithm Selection
Data-Driven
Expert Knowledge-Driven
Case Base
Probabilistic Model
Guidelines Structured Knowledge
Knowledge Resources
X
2 3 4
1 Knowledge Acquisition Editor
Descriptive Unstructured Knowledge (CNL)
1 Rules creation from domain data
2 Direct Rules Creation
3 Rules creation from Guidelines
4 Rules creation from unstructured resources
Knowledge Maintenance (MCRDR)
Rule Base
Cornerstone Case Base
경희대학교
KYUNG HEE UNIVERSITY
UTAS

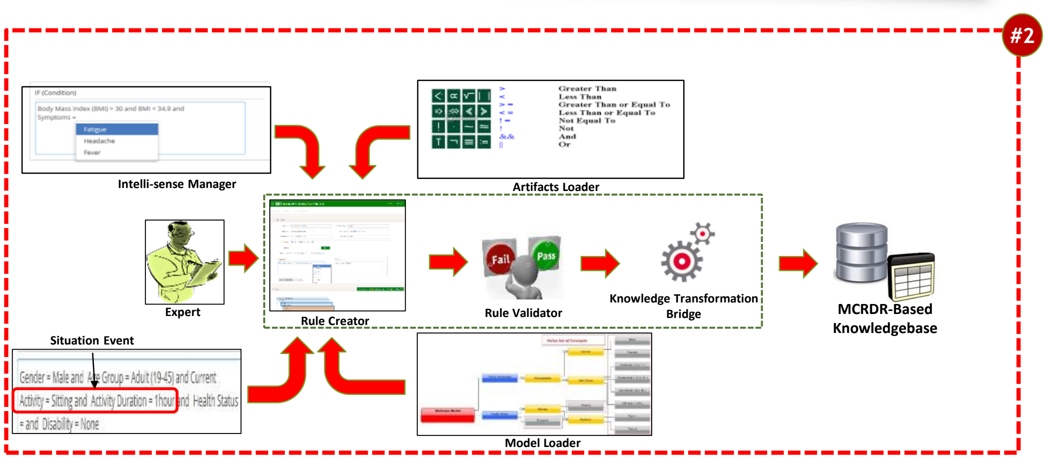
Data-Driven Knowledge Acquisition (Path 1)

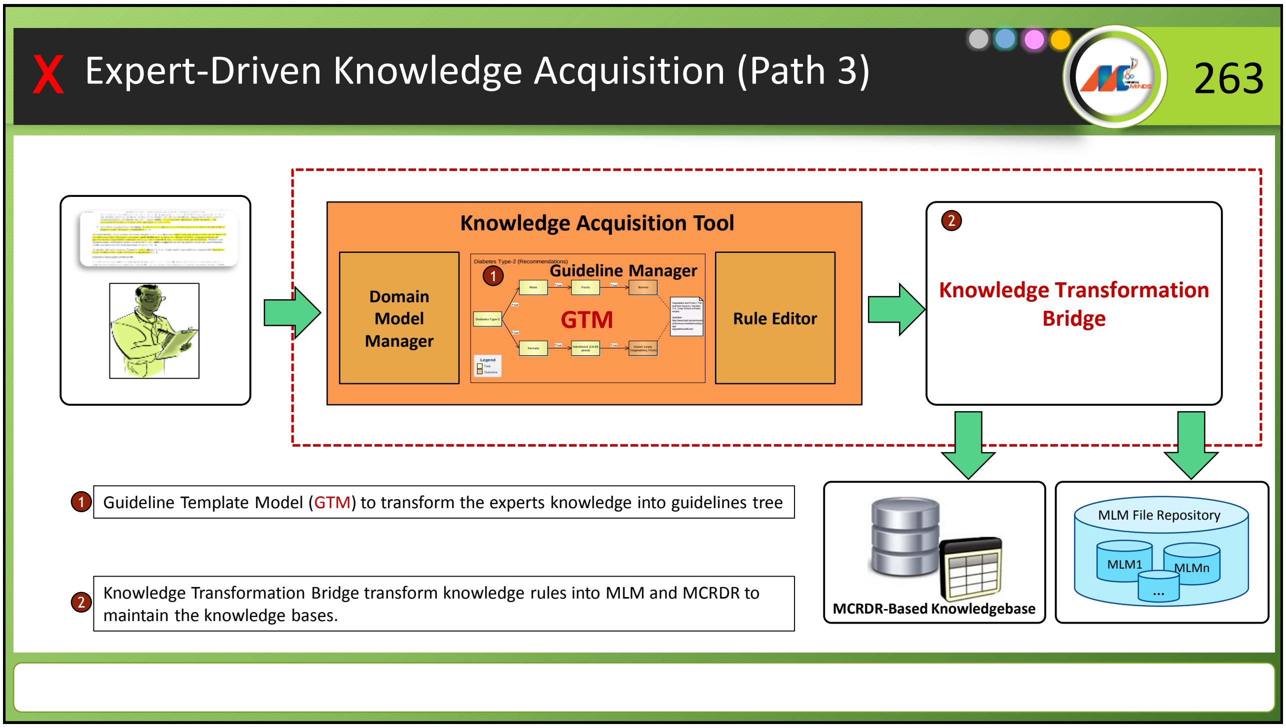
261

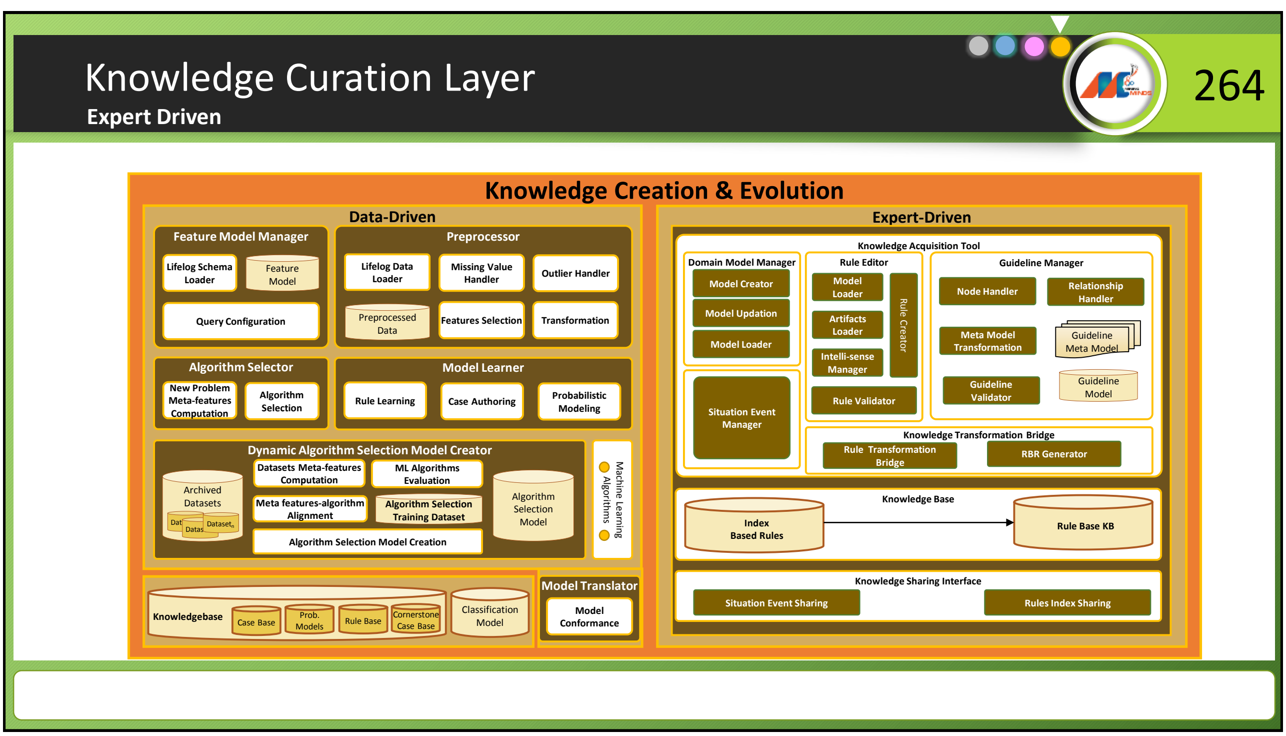


Expert-Driven Knowledge Acquisition (Path 2)

262



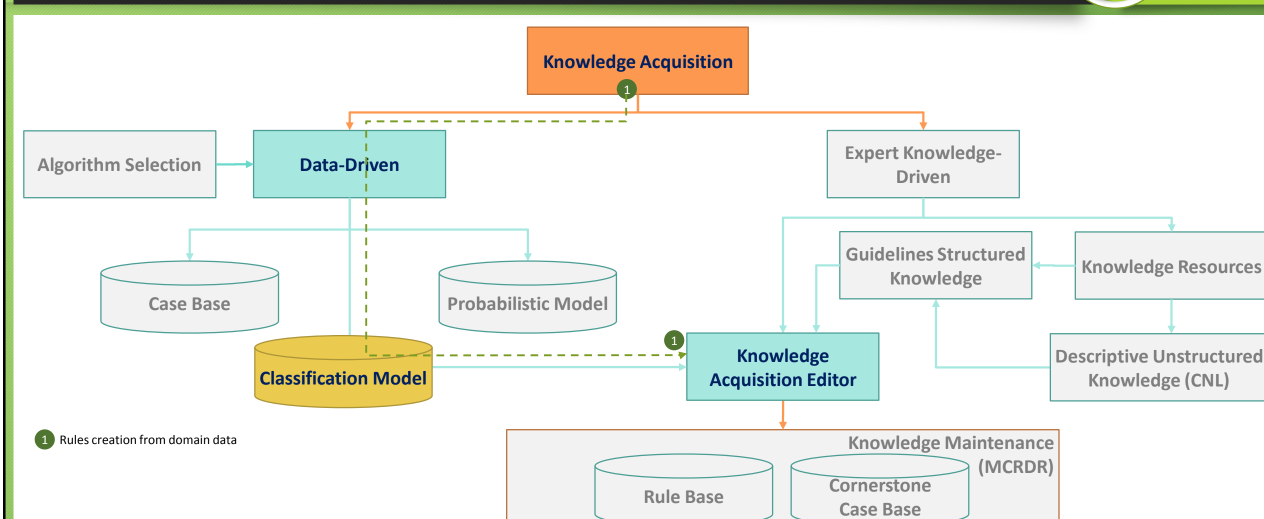

Expert-Driven Knowledge Acquisition (Path 3)
263
Knowledge Acquisition Tool
Domain Model Manager
Guideline Manager
GTM
Rule Editor
Knowledge Transformation Bridge
1 Guideline Template Model (GTM) to transform the experts knowledge into guidelines tree
2 Knowledge Transformation Bridge transform knowledge rules into MLM and MCRDR to maintain the knowledge bases.
MCRDR-Based Knowledgebase
MLM File Repository
MLM1
MLMn
...


Knowledge Curation Layer
Expert Driven
264
Knowledge Creation & Evolution
Data-Driven
Feature Model Manager
Lifelog Schema Loader
Feature Model
Query Configuration
Preprocessor
Lifelog Data Loader
Missing Value Handler
Outlier Handler
Preprocessed Data
Features Selection
Transformation
Algorithm Selector
New Problem Meta-features Computation
Algorithm Selection
Model Learner
Rule Learning
Case Authoring
Probabilistic Modeling
Dynamic Algorithm Selection Model Creator
Archived Datasets
Datasets Meta-features Computation
ML Algorithms Evaluation
Meta features-algorithm Alignment
Algorithm Selection Training Dataset
Algorithm Selection Model
Algorithm Selection Model Creation
Machine Learning Algorithms
Knowledgebase
Case Base
Prob. Models
Rule Base
Cornerstone Case Base
Classification Model
Model Translator
Model Conformance
Expert-Driven
Knowledge Acquisition Tool
Domain Model Manager
Model Creator
Model Updation
Model Loader
Rule Editor
Model Loader
Artifacts Loader
Intelli-sense Manager
Rule Validator
Rule Creator
Guideline Manager
Node Handler
Relationship Handler
Meta Model Transformation
Guideline Meta Model
Guideline Validator
Guideline Model
Situation Event Manager
Knowledge Transformation Bridge
Rule Transformation Bridge
RBR Generator
Knowledge Base
Index Based Rules
Rule Base KB
Knowledge Sharing Interface
Situation Event Sharing
Rules Index Sharing

Data Driven Knowledge Acquisition

Scope of Data Driven Knowledge Acquisition for MM Ver. 2.5

266



What is Knowledge Acquisition?

- **Knowledge acquisition** is the process of *extracting*, *structuring* and *organizing* knowledge from one **source**
- **Source** can be *Structured* or *Unstructured* Data

Data Driven: Improving Business and Society Through Data

Relatively easy to analyze

Structured Data

Unstructured Data

What is Data Driven?

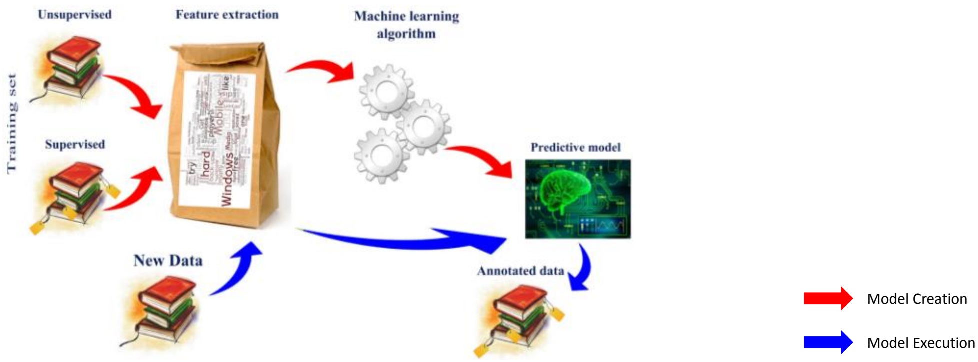
- Data driven is a powerful approach, which
  - process existing data and then create new ones from them.
  - requires less up-front knowledge, but a lot more back-end computation and experimentation

http://www.cad.zju.edu.cn/home/zhx/csmath/lib/exe/fetch.php?media=2013:csmath-01-data-driven.pdf

Machine Learning and Data Driven

269

- Machine learning follows data driven approach
  - Extracts hidden patterns of large data using different computerized algorithms



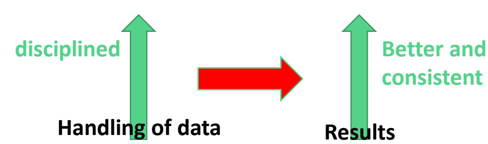
경희대학교 KYUNG HEE UNIVERSITY UTAS

http://www.computervisionblog.com/2015/03/deep-learning-vs-machine-learning-vs.html

Machine Learning and Data Preprocessing

270

- Machine learning algorithms learn from data that should
  - be right
  - be in a useful scale and format
  - includes meaningful features



- **Data Preprocessing** describes any type of process performed on raw data to prepare it into an understandable format

경희대학교 KYUNG HEE UNIVERSITY UTAS

http://www.mimuw.edu.pl/~son/datamining/DM/4-preprocess.pdf

Data Preprocessing

271

- Real world data is usually in raw structured format
  - incomplete: lacking attribute values, containing only aggregate data
    - e.g., occupation=“ ”
  - noisy: containing errors or outliers
    - e.g., Salary=“-10”
  - inconsistent: containing discrepancies in codes or names
    - e.g., Age=“42” Birthday=“03/07/1997”
    - e.g., Was rating “1,2,3”, now rating “A, B, C”

- No quality data, no quality mining results!
  - Quality decisions must be based on quality data

- Data preprocessing lacks the support of Expert-Driven Feature Modeling.

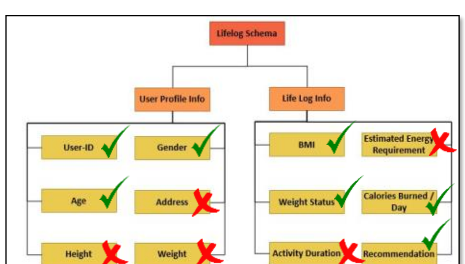
경희대학교 KYUNG HEE UNIVERSITY UTAS

http://www.mimuw.edu.pl/~son/datamining/DM/4-preprocess.pdf

What is Feature Modeling?

272

- A feature model defines the valid combinations of features in a domain.
- Used widely in domain engineering and product line Approaches [1,2]



경희대학교 KYUNG HEE UNIVERSITY UTAS

www2.fiit.stuba.sk/~vranic/pub/FMM.pdf

Related Works

Category	Study	Features	Limitations
Feature Modeling	Sánchez [1], Vranić [2]	• Addresses quality attributes at run time by means of feature models • Context based feature selection • Used in the field of Software Product Lines (SPL)	• Requires extra engineering efforts at development time for designing and mapping the model
Data Preprocessing	Kwiatkowska [3], Dimitriadis [4],	• Prepared consistent and calibrated baseline dataset • Handle noisy, highly variable data	• Knowledge engineer dependency
Algorithm Selection	Smith [5], Ali and Smith [6], Song [7], Wang [8]	• Finds best classification algorithm • Considered multiple datasets • Considered multiple algorithms	• Use a sub-set of meta-features • Use single-metric evaluation criteria • Use black-box learning techniques for model creation
Model Learning	Dimitriadis [4], Bachman [9]	• Extract new knowledge • Create effective set of decision rules • Worked on Time, space, and medical domain	• Used fixed machine learning methods (ZeroR, NaiveBayes, J48, SVM) • Knowledge engineer dependency

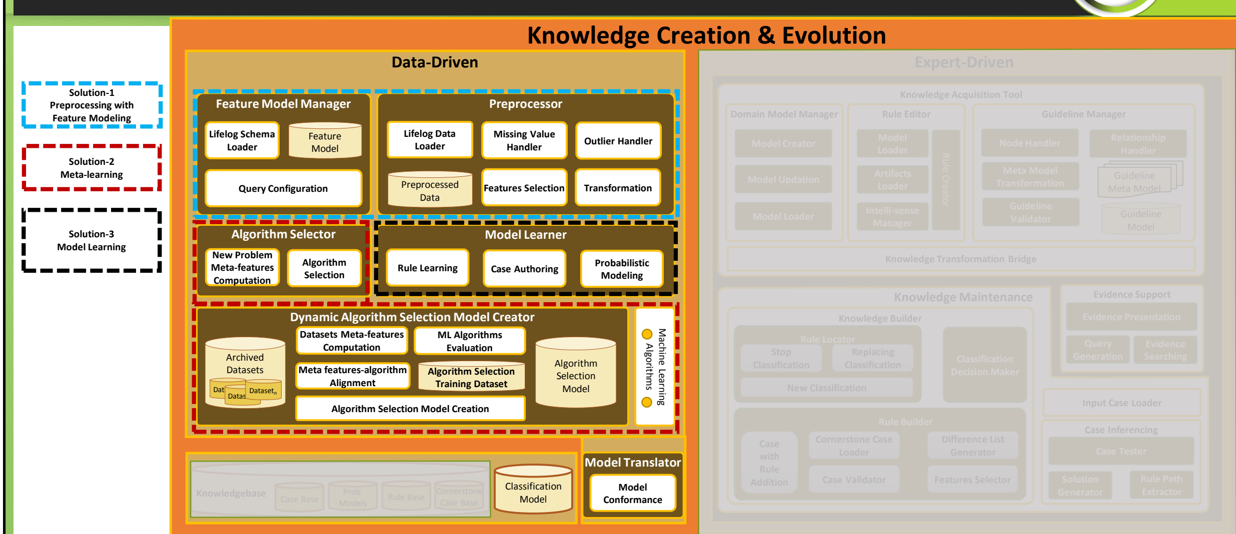
Limitations

- Expert manually select machine learning algorithm for building classification model
- Use a sub-set of meta-features
- Lack of feature modeling for preprocessing of data

Motivations – Challenges - Solutions

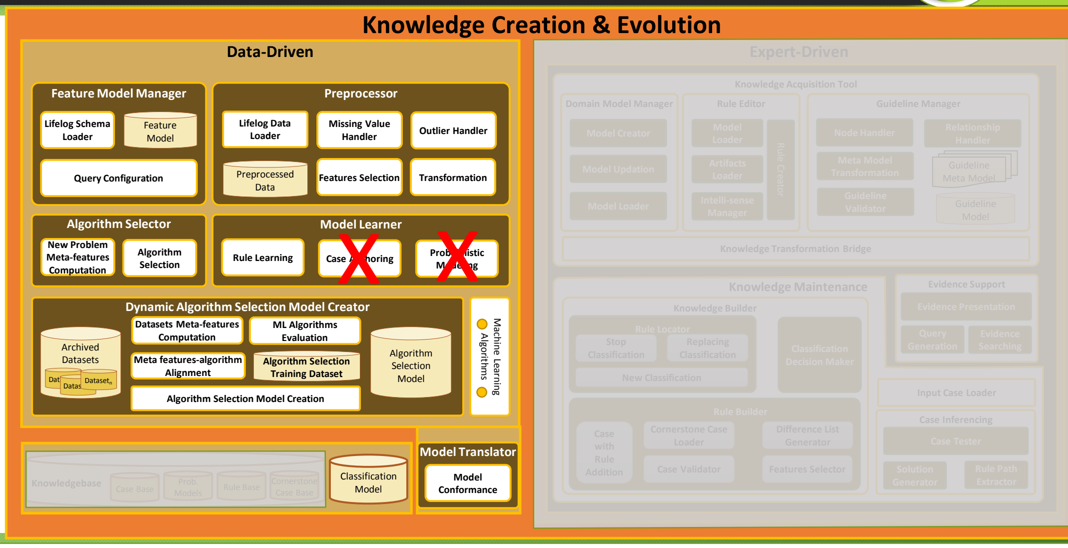
Motivations	Challenges	Solutions
For yielding correct classification model in any domain, selection of quality data requires domain configuration/engineering as well as data preprocessing	Ch-1 How to select, manage, and process quality attributes?	Sol-1 Integration of Data Pre-Processing with Feature Modeling
There exist lot of ML methods that can be utilized to build classification model for large amount of data	Ch-2 How to automatically select an appropriate ML method for learning classification model appropriately?	Sol-2 Meta-learning
A classification model can auto generate recommendation for different services	Ch-3 How to build a classification model?	Sol-3 Model Learning

Data Driven Knowledge Acquisition



Data Driven Knowledge Acquisition Ver. 2.5

277



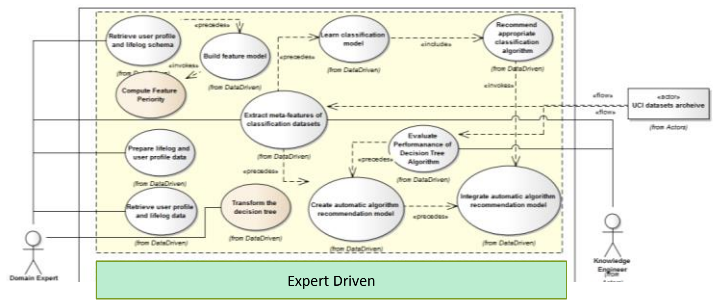
Goals of Data Driven Knowledge Acquisition Ver. 2.5

278

- Design and Development of Feature Model (Partial Implementation)
- Preparing raw data into processed form
- Meta-features extraction from UCI and OpenML datasets
- Classification algorithms performance evaluation (Offline Process)
- Classification Model Learning
- Rules extraction from classification model (Initial Implementation)

Data Driven Knowledge Acquisition Ver. 2.5 – Use cases

Use Case #ID	Use Case Name
KCL2.5-SUC-05	Retrieve user profile and lifelog schema
KCL2.5-SUC-01	Build feature model
KCL2.5-SUC-06	Retrieve user profile and lifelog data
KCL2.5-SUC-02	Prepare lifelog and user profile data
KCL2.5-SUC-13	Learn classification model
KCL2.5-SUC-18	Compute features priorities
KCL2.5-SUC-19	Transform the decision tree



Data Driven Knowledge Acquisition Ver. 2.5 – Use cases

Retrieve user profile and lifelog schema

Compute features priorities

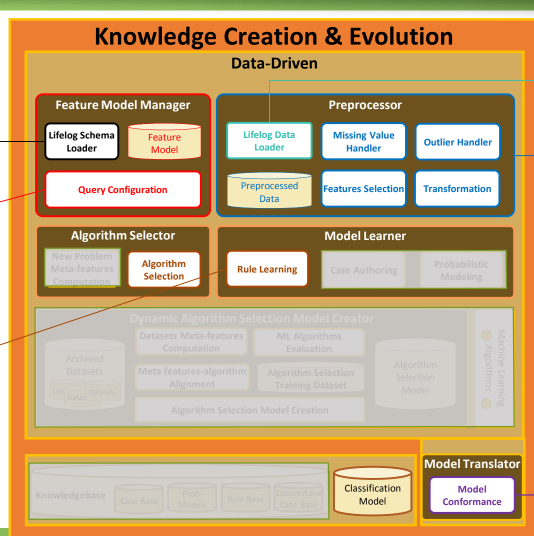
extends

Build feature model

- Feature Modeling
  - Select the attributes

Learn classification model

- Model Learning
  - Generate the decision tree



Retrieve user profile and lifelog data

Prepare lifelog and user profile data

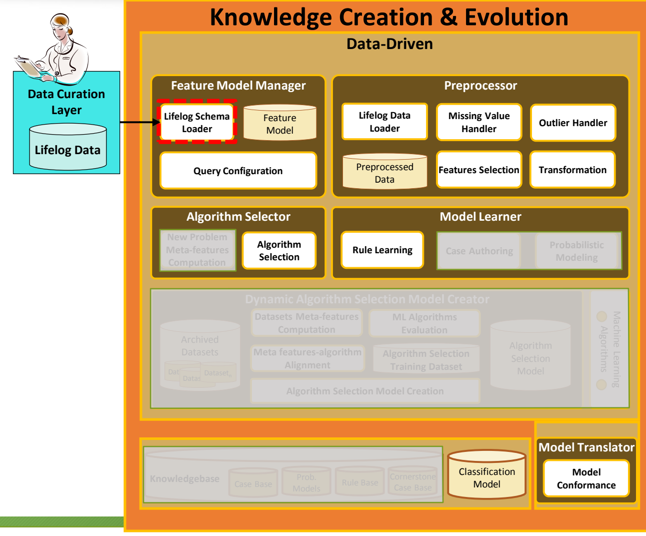
- Data Cleaning
  - Fill in missing values, identify or remove outliers
- Data Transformation
  - Generalization: concept hierarchy climbing
- Data Reduction
  - Obtains reduced representation in volume

Transform the decision tree

KCL2.5-SUC-05: Retrieve user profile and lifelog schema 281

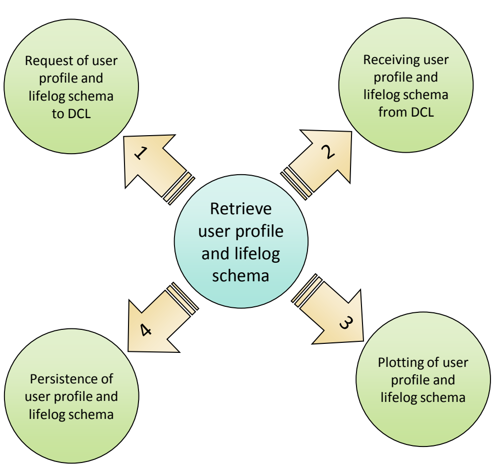
Use Case

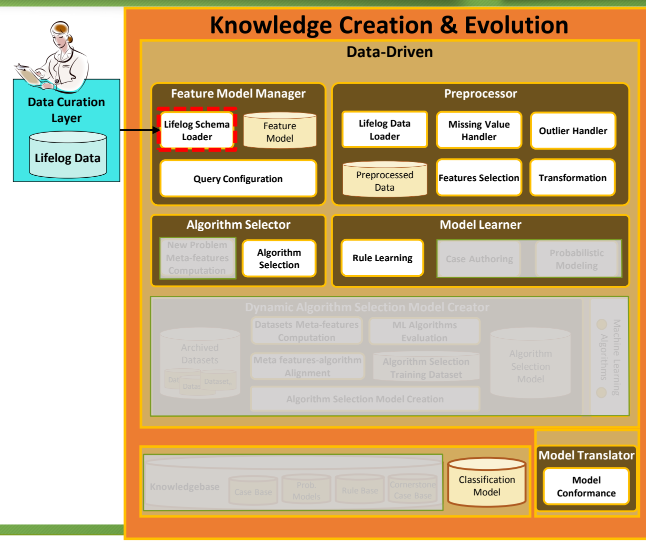
- **Objective**
  - To help domain expert to view all available features for building feature model
- **Methodology**
  - ***Communication***: Restful Service between *Feature Model Manager* and *DCL*
  - ***Lifelog Schema Loader*** *(operation)*
    1. **Input**: Required configuration of domain
    2. **Processing**:
       a. Domain expert selects the domain and sends requests to DCL for user profile and lifelog schema.
       b. DCL shares the user profile and lifelog schema
       c. System receives the user profile and lifelog schema
       d. System saves the received schema into schema storage
    3. **Output**: forwards the received schema to *Query Configuration*

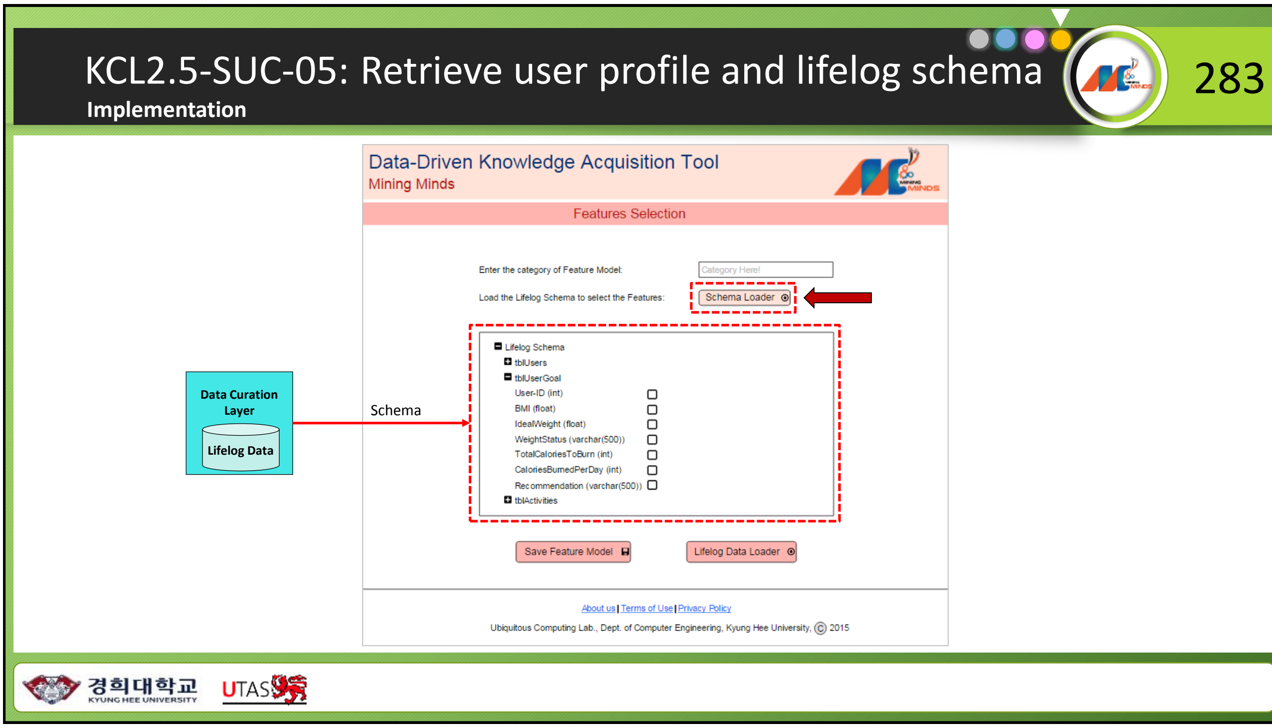


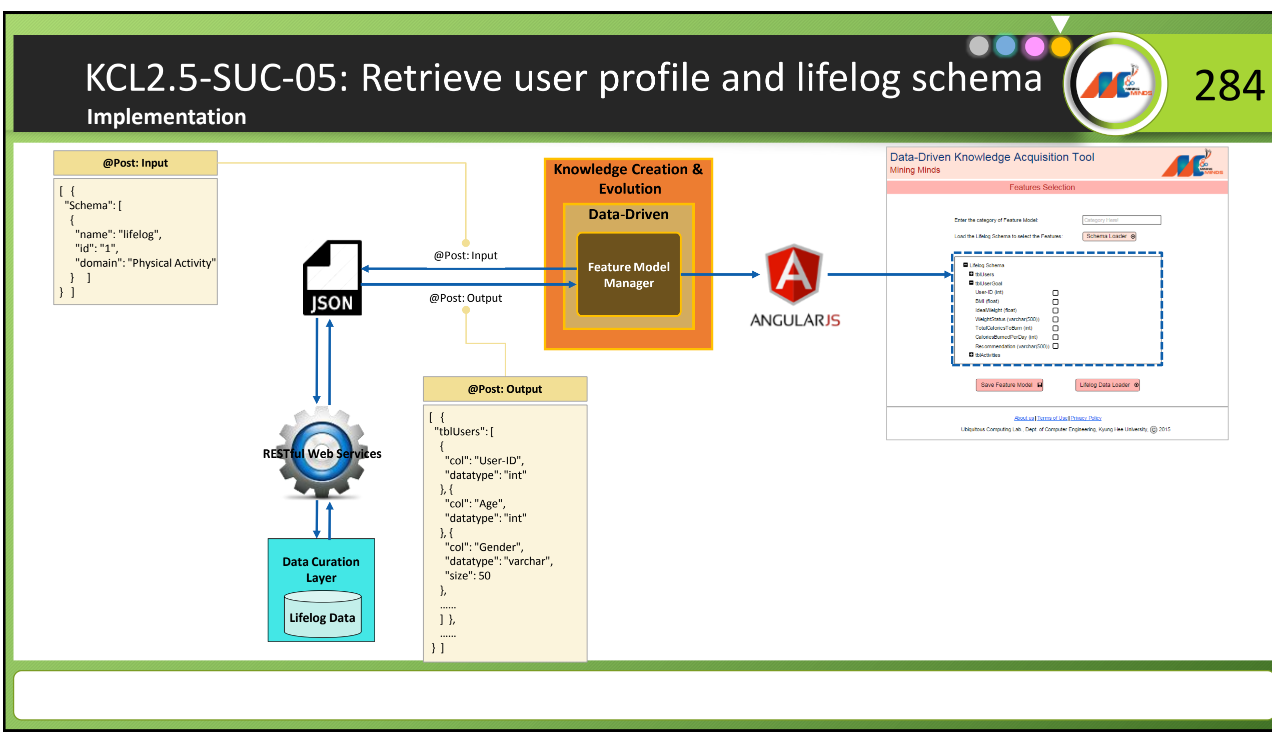
KCL2.5-SUC-05: Retrieve user profile and lifelog schema 282

Use Case Functions






KCL2.5-SUC-05: Retrieve user profile and lifelog schema
Implementation
283
Data-Driven Knowledge Acquisition Tool
Mining Minds
Features Selection
Enter the category of Feature Model:
Category Here!
Load the Lifelog Schema to select the Features:
Schema Loader
Lifelog Schema
tblUsers
tblUserGoal
User-ID (int)
BMI (float)
IdealWeight (float)
WeightStatus (varchar(500))
TotalCaloriesToBurn (int)
CaloriesBurnedPerDay (int)
Recommendation (varchar(500))
tblActivities
Save Feature Model
Lifelog Data Loader
About us | Terms of Use | Privacy Policy
Ubiquitous Computing Lab., Dept. of Computer Engineering, Kyung Hee University, © 2015
Data Curation Layer
Lifelog Data
Schema
경희대학교
KYUNG HEE UNIVERSITY
UTAS

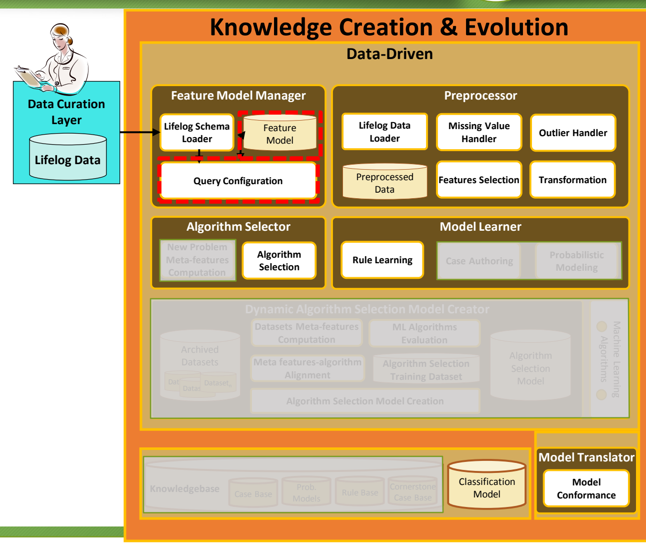

KCL2.5-SUC-05: Retrieve user profile and lifelog schema
Implementation
284
@Post: Input
[{
"Schema": [
{
"name": "lifelog",
"id": "1",
"domain": "Physical Activity"
}]
}]
JSON
@Post: Input
@Post: Output
Knowledge Creation & Evolution
Data-Driven
Feature Model Manager
ANGULARJS
RESTful Web Services
Data Curation Layer
Lifelog Data
@Post: Output
[{
"tblUsers": [
{
"col": "User-ID",
"datatype": "int"
}, {
"col": "Age",
"datatype": "int"
}, {
"col": "Gender",
"datatype": "varchar",
"size": 50
},
......
] },
......
}]
Data-Driven Knowledge Acquisition Tool
Mining Minds
Features Selection

KCL2.5-SUC-01: Build feature model

Use Case

285

- **Objective**
  - To define the valid combinations of features in a domain for building feature model
- **Methodology**
  - ***Query Configuration*** *(operation)*
  1. **Input**: Lifelog schema
  2. **Processing**:
     a. Domain expert retrieves the schema from schema storage.
     b. System loads and plots the schema
     c. Domain expert builds the feature models as follows:
        a. Select the required features for corresponding domain
        b. Verify the consistency of the selected features (such as concept hierarchy)
        c. Save the feature model
     d. System persists the feature model into repository.
  3. **Output**: builds the feature model and forwards that model to *Lifelog Data Loader*

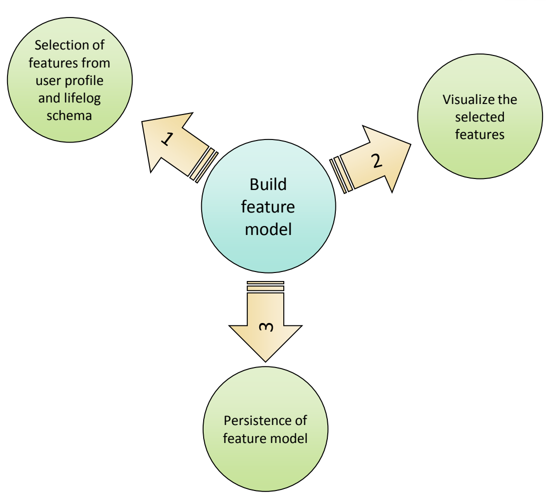


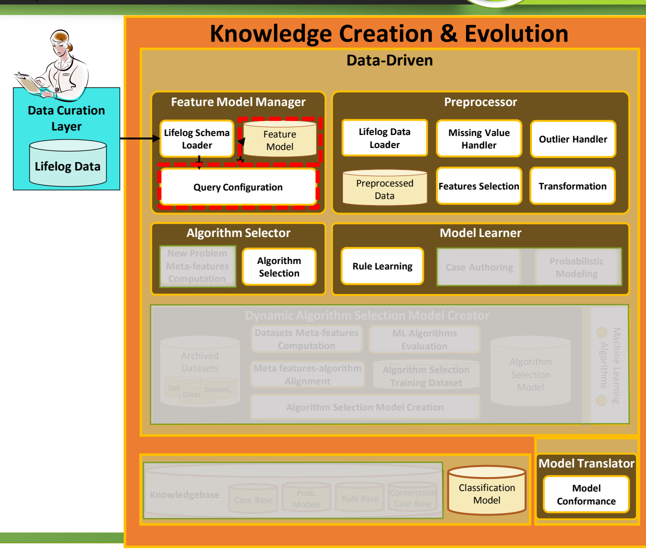
경희대학교 KYUNG HEE UNIVERSITY UTAS

KCL2.5-SUC-01: Build feature model

Use Case Functions

286



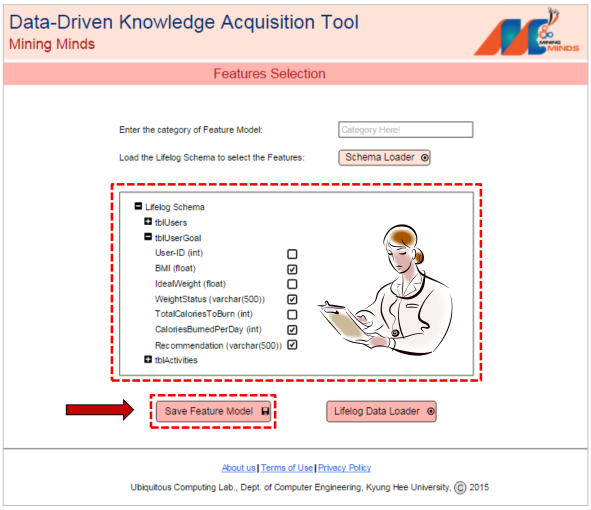


경희대학교 KYUNG HEE UNIVERSITY UTAS

KCL2.5-SUC-01: Build feature model

Implementation

287



<<Physical Activity Schema>>
```
<?xml version="1.0" encoding="UTF-8"?>
<xs:element name="tblUsers">
 <xs:complexType>
  <xs:sequence>
        <xs:element name="User-ID" type="xs:int"/>
        <xs:element name="Age" type="xs:int"/>
        <xs:element name="Gender" type="xs:string"/>
  </xs:sequence>
 </xs:complexType>
 <xs:key name="tblUserKey" msdata:PrimaryKey="true">
  <xs:selector xpath=".//tblUser"/>
  <xs:field xpath="User-ID"/>
 </xs:key>
</xs:element>
<xs:element name="tblUserGoal">
 <xs:complexType>
  <xs:sequence>
        <xs:element name="BMI" type="xs:float"/>
        <xs:element name="WeightStatus" type="xs:string"/>
        <xs:element name="CaloriesBurnedPerDay" type="xs:int"/>
        <xs:element name="Recommendation" type="xs:string"/>
  </xs:sequence>
 </xs:complexType>
<xs:keyref name="tblUserGoalFKey" refer="tblUserKey">
  <xs:selector xpath=".//tblUserGoal" />
  <xs:field xpath="User-ID" />
</xs:keyref>
</xs:element>
```


Feature Model

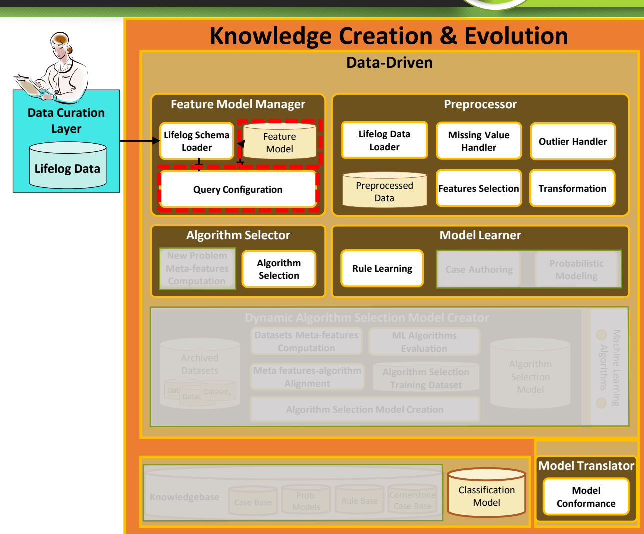
경희대학교 KYUNG HEE UNIVERSITY UTAS

KCL2.5-SUC-18: Compute features priorities

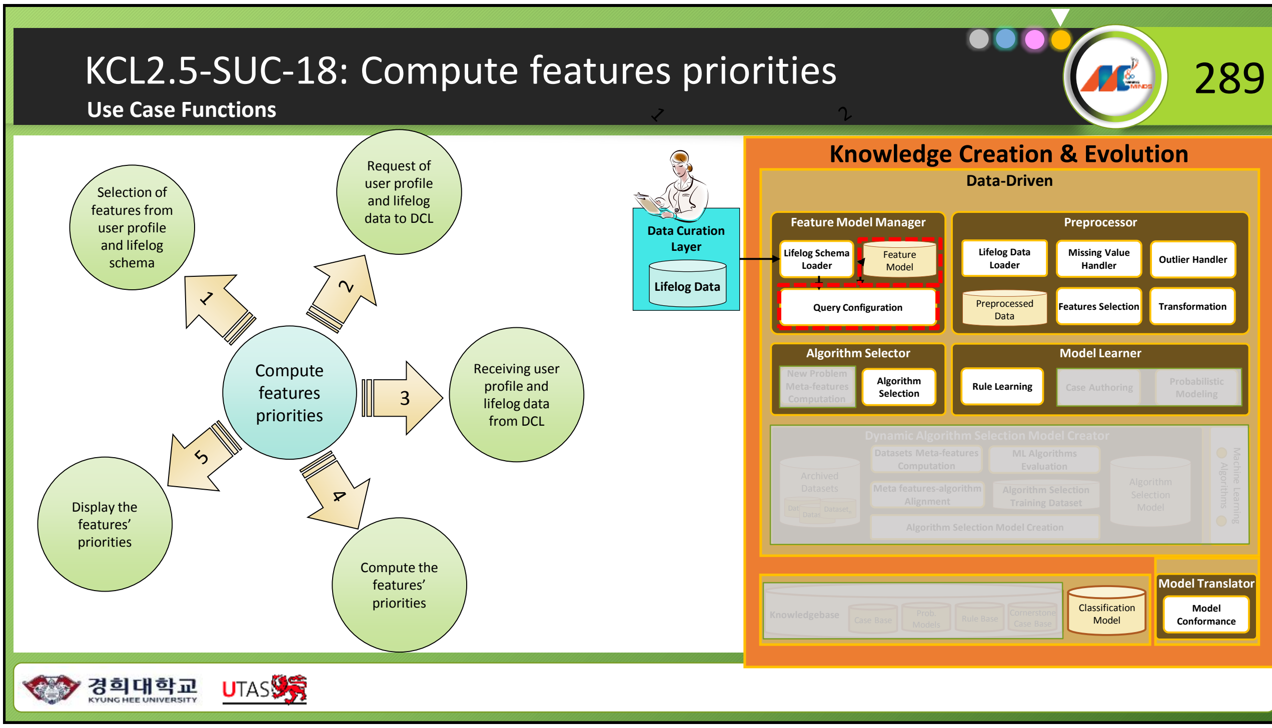
Use Case

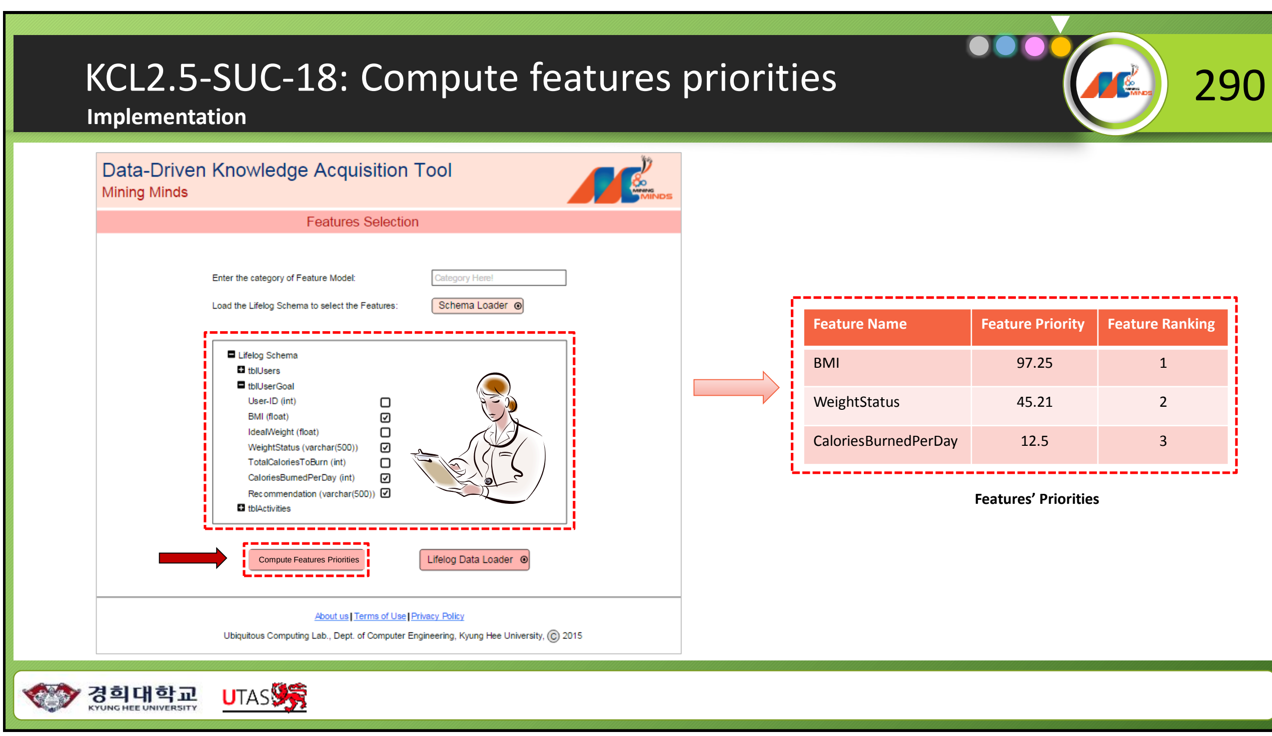
288

- **Objective**
  - ➢ To compute the features priorities for helping domain expert to select appropriate features from available schema
- **Methodology**
  - ➢ ***Query Configuration*** *(operation)*
    1. **Input**: Lifelog schema
    2. **Processing**:
       a. Domain expert selects the features
       b. System sends request to DCL2.5 for user profile lifelog data based on selected features of schema and features conditions.
       c. DCL2.5 sends the required data
       d. System receives user profile lifelog data to compute the features priorities
       e. System displays the features priorities list
    3. **Output**: displays the features priorities list to domain expert for showing importance of each selected feature.



경희대학교 KYUNG HEE UNIVERSITY UTAS





Feature Name	Feature Priority	Feature Ranking
BMI	97.25	1
WeightStatus	45.21	2
CaloriesBurnedPerDay	12.5	3

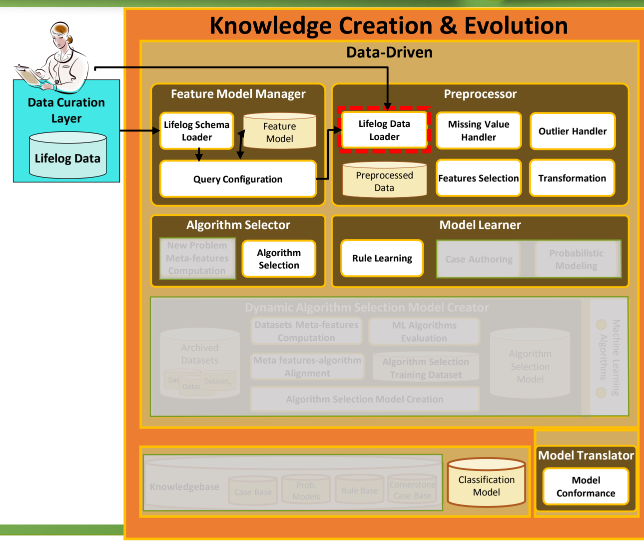
Features' Priorities

KCL2.0-SUC-06: Retrieve user profile and lifelog data

Use Case

291

- **Objective**
  - To help domain expert to view unprocessed user profile and lifelog data
- **Methodology**
  - ***Communication***: Restful Service between *Preprocessor* and *DCL*
  - ***Lifelog Data Loader*** *(operation)*
    1. **Input**: Feature model and lifelog data
    2. **Processing**:
       a. Domain expert loads the feature model for selected domain
       b. System loads the corresponding feature model
       c. Domain expert sends request to DCL for user profile and lifelog data based on loaded feature model
       d. DCL shares the user profile and lifelog data
       e. System receives the user profile and lifelog data
       f. System saves the received lifelog data
    3. **Output**: forwards the lifelog data to *Missing Value Handler*

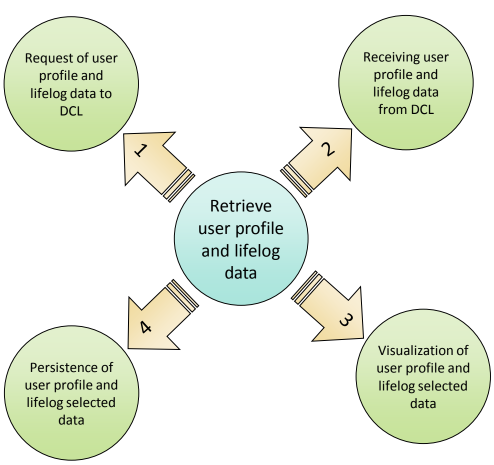


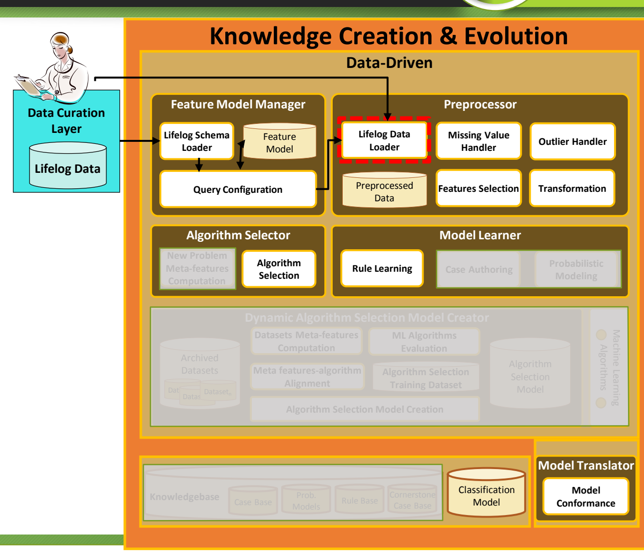
경희대학교 KYUNG HEE UNIVERSITY UTAS

KCL2.5-SUC-06: Retrieve user profile and lifelog data

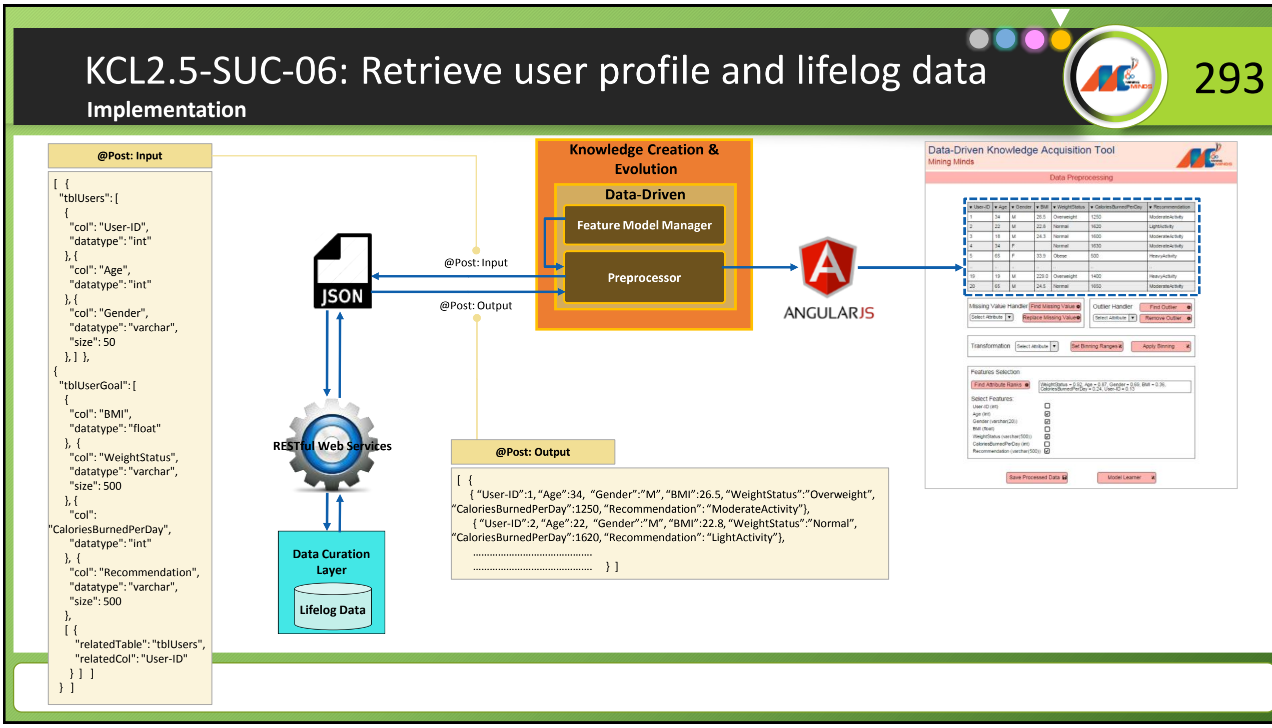
Use Case Functions

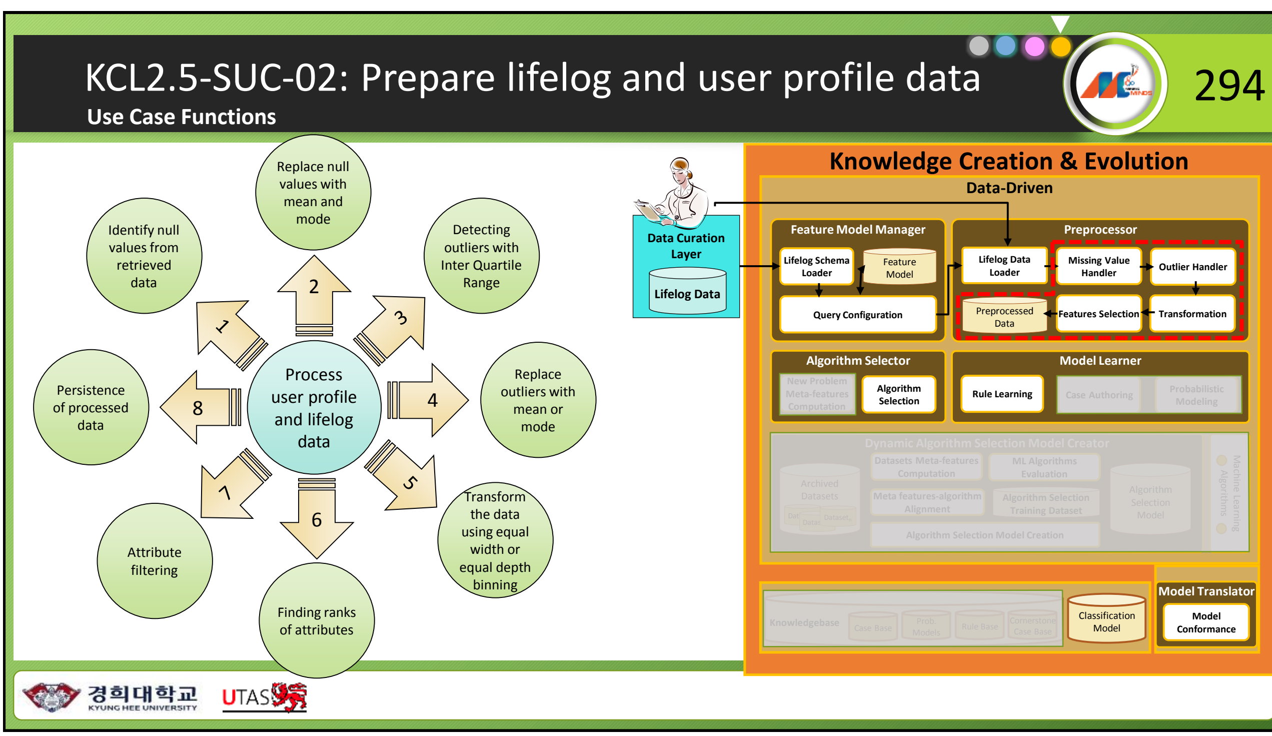
292





경희대학교 KYUNG HEE UNIVERSITY UTAS


KCL2.5-SUC-06: Retrieve user profile and lifelog data
Implementation
293
@Post: Input
[{
"tblUsers": [
{
"col": "User-ID",
"datatype": "int"
}, {
"col": "Age",
"datatype": "int"
}, {
"col": "Gender",
"datatype": "varchar",
"size": 50
},] },
{
"tblUserGoal": [
{
"col": "BMI",
"datatype": "float"
}, {
"col": "WeightStatus",
"datatype": "varchar",
"size": 500
}, {
"col":
"CaloriesBurnedPerDay",
"datatype": "int"
}, {
"col": "Recommendation",
"datatype": "varchar",
"size": 500
},
[{
"relatedTable": "tblUsers",
"relatedCol": "User-ID"
}]]
}]
JSON
@Post: Input
@Post: Output
RESTful Web Services
Data Curation Layer
Lifelog Data
Knowledge Creation & Evolution
Data-Driven
Feature Model Manager
Preprocessor
ANGULARJS
@Post: Output
[{
{ "User-ID":1, "Age":34, "Gender":"M", "BMI":26.5, "WeightStatus":"Overweight", "CaloriesBurnedPerDay":1250, "Recommendation":"ModerateActivity"},
{ "User-ID":2, "Age":22, "Gender":"M", "BMI":22.8, "WeightStatus":"Normal", "CaloriesBurnedPerDay":1620, "Recommendation": "LightActivity"},
……………………………………
…………………………………… }]
Data-Driven Knowledge Acquisition Tool
Mining Minds
Data Preprocessing

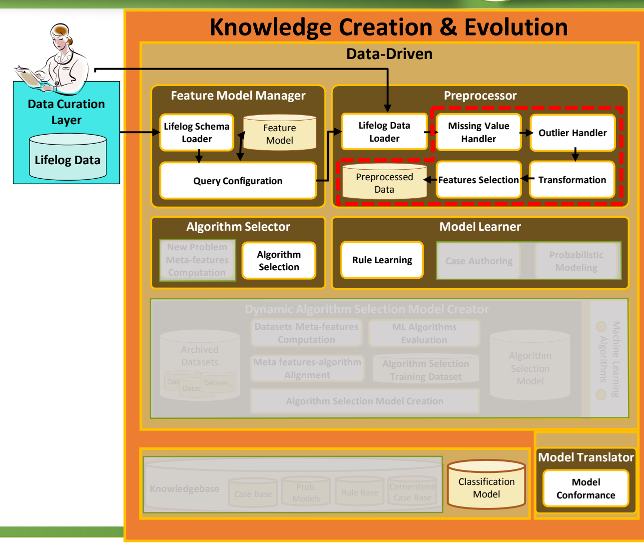

KCL2.5-SUC-02: Prepare lifelog and user profile data
Use Case Functions
294
Process user profile and lifelog data
1
Identify null values with retrieved data
2
Replace null values with mean and mode
3
Detecting outliers with Inter Quartile Range
4
Replace outliers with mean or mode
5
Transform the data using equal width or equal depth binning
6
Finding ranks of attributes
7
Attribute filtering
8
Persistence of processed data
Data Curation Layer
Lifelog Data
Knowledge Creation & Evolution
Data-Driven
Feature Model Manager
Lifelog Schema Loader
Feature Model
Query Configuration
Preprocessor
Lifelog Data Loader
Missing Value Handler
Outlier Handler
Preprocessed Data
Features Selection
Transformation
Algorithm Selector
New Problem Meta-features Computation
Algorithm Selection
Model Learner
Rule Learning
Case Authoring
Probabilistic Modeling
Dynamic Algorithm Selection Model Creator
Knowledgebase
Classification Model
Model Translator
Model Conformance
경희대학교
KYUNG HEE UNIVERSITY
UTAS

KCL2.5-SUC-02: Prepare lifelog and user profile data (1/4)

Missing Value Handler | Outlier Handler | Transformation | Features Selection

295

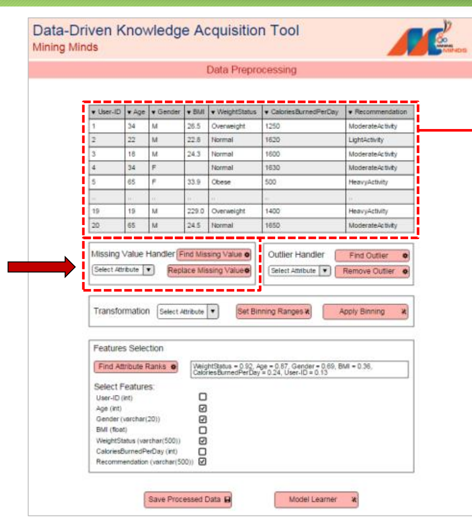
- **Objective**
  - To preprocess the data to generate models with high accuracy.
- **Methodology**
  - ***Missing Value Handler*** *(operation)*
    1. **Input**: unprocessed data
    2. **Processing**:
       a. Domain expert loads the unprocessed data
       b. System displays the retrieved data
       c. For each attribute:
          i. Domain expert identifies the missing values and select appropriate method from following options for missing value replacement.
             - Mean
             - Mode
          ii. System replaces the missing values using selected method.
       d. Domain expert saves the processed data into repository
    3. **Output**: forwards the processed data to *Outlier Handler*



KCL2.5-SUC-02: Prepare lifelog and user profile data (1/4)

Missing Value Handler | Outlier Handler | Transformation | Features Selection

296



Unprocessed Data

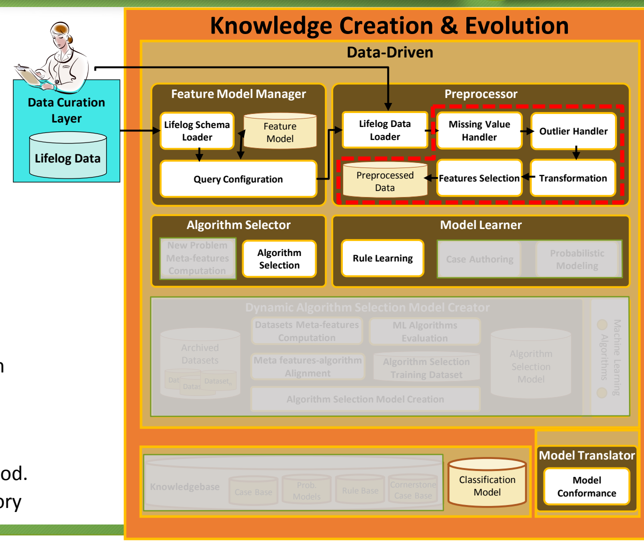
User-ID	Age	Gender	BMI	WeightStatus	CaloriesBurnedPerDay	Recommendation
1	34	M	26.5	Overweight	1250	ModerateActivity
2	22	M	22.8	Normal	1620	LightActivity
3	18	M	24.3	Normal	1600	ModerateActivity
4	34	F		Normal	1630	ModerateActivity
5	65	F	33.9	Obese	500	HeavyActivity
..	..	..	..	..	...	..
19	19	M	229.0	Overweight	1400	HeavyActivity
20	65	M	24.5	Normal	1650	ModerateActivity

- **Mean/Mode substitution**
  - Replace missing value with sample mean or mode, which are two kinds of "averages".
  - The "**mean**" is the "average" you're used to, where you add up all the numbers and then divide by the number of numbers.
  - The "**mode**" is the value that occurs most often. If no number is repeated, then there is no mode for the list.
- Dummy Variable Adjustment
  - Create an indicator for missing value (1=value is missing for observation; 0=value is observed for observation)
  - Impute missing values to a constant (such as the mean)
  - Include missing indicator in regression
- Regression Imputation
  - Replaces missing values with predicted score from a regression equation.

KCL2.5-SUC-02: Prepare lifelog and user profile data (2/4)

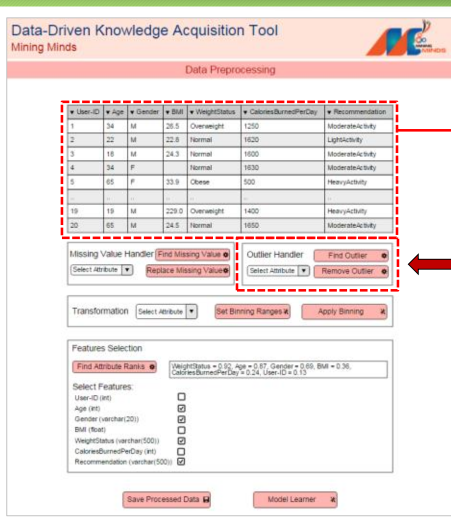
Missing Value Handler | **Outlier Handler** | Transformation | Features Selection

- **Objective**
  - To preprocess the data to generate models with high accuracy.
- **Methodology**
  - ***Outlier Handler*** *(operation)*
    1. **Input**: processed data
    2. **Processing**:
       a. Domain expert loads the processed data
       b. System displays the retrieved data
       c. For each attribute:
          i. Domain expert apply outlier detection method such as interquartile range and scatterplot.
          ii. System display the outliers
          iii. Domain expert select appropriate method from following options for outlier replacement.
             - Mean
             - Mode
          iv. System replace the outlier using selected method.
       d. Domain expert saves the processed data into repository
    3. **Output**: forwards the processed data to *Transformation*



KCL2.5-SUC-02: Prepare lifelog and user profile data (2/4)

Missing Value Handler | **Outlier Handler** | Transformation | Features Selection

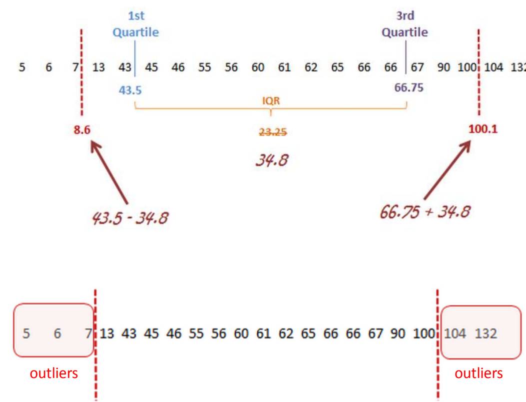


Unprocessed Data

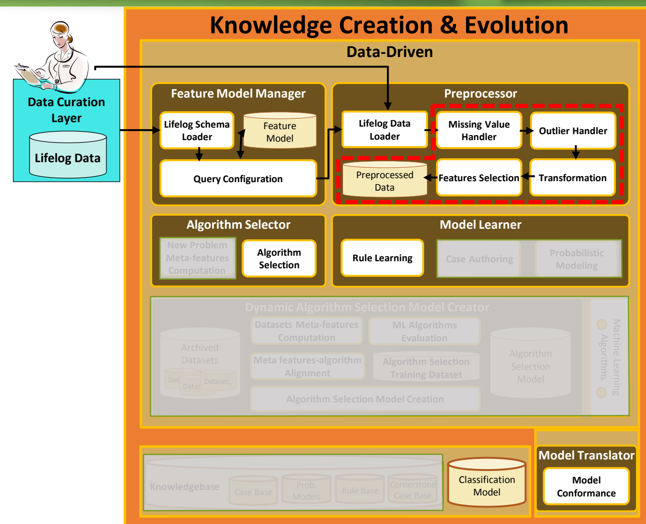
User-ID	Age	Gender	BMI	WeightStatus	CaloriesBurnedPerDay	Recommendation
1	34	M	26.5	Overweight	1250	ModerateActivity
2	22	M	22.8	Normal	1620	LightActivity
3	18	M	24.3	Normal	1600	ModerateActivity
4	34	F		Normal	1630	ModerateActivity
5	65	F	33.9	Obese	500	HeavyActivity
..	..	..	..	..	...	..
19	19	M	229.0	Overweight	1400	HeavyActivity
20	65	M	24.5	Normal	1650	ModerateActivity

- Outlier
  - An outlier is an observation which deviates so much from the other observations
- Detection Methods
  - **Statistical Methods**
    - **Parametric**
    - **Non-Parametric**
      - **Histogram**
      - **Inter Quartile Ranges (IQR)**
  - Proximity-Based Methods
  - Clustering-Based Methods

- **Inter Quartile Ranges (IQR)**
  - Calculate the interquartile range (IQR) of data
    - IQR = Quartile 3 – Quartile 1
  - Multiply the interquartile range (IQR) by the number 1.5
  - Add 1.5 x (IQR) to the third quartile. Any number greater than this is a suspected outlier.
    - (Quartile 3) + (1.5 × IQR)
  - Subtract 1.5 x (IQR) from the first quartile. Any number less than this is a suspected outlier.
    - (Quartile 1) – (1.5 × IQR)



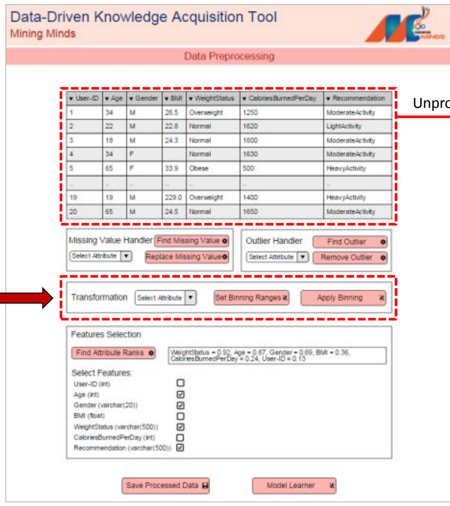
- **Objective**
  - To preprocess the data to generate models with high accuracy.
- **Methodology**
  - ***Transformation*** *(operation)*
    1. **Input**: processed data
    2. **Processing**:
       a. Domain expert loads the processed data
       b. System displays the retrieved data
       c. For each attribute:
          i. Domain expert identifies, normalizes the non-transformed values, and updates the dataset.
          ii. System modifies the values set and update the dataset
       d. Domain expert saves the processed data into repository
    3. **Output**: forwards the processed data to *Feature Selection*



KCL2.5-SUC-02: Prepare lifelog and user profile data (3/4)

Missing Value Handler | Outlier Handler | **Transformation** | Features Selection

301



Unprocessed Data

User-ID	Age	Gender	BMI	WeightStatus	CaloriesBurnedPerDay	Recommendation
1	34	M	26.5	Overweight	1250	ModerateActivity
2	22	M	22.8	Normal	1620	LightActivity
3	18	M	24.3	Normal	1600	ModerateActivity
4	34	F		Normal	1630	ModerateActivity
5	65	F	33.9	Obese	500	HeavyActivity
..	..	..	..	..	...	.
19	19	M	229.0	Overweight	1400	HeavyActivity
20	65	M	24.5	Normal	1650	ModerateActivity

> 1650 → High
1600 – 1650 → Normal
< 1600 → Low

CaloriesBurnedPerDay
Low
Normal
Low
Low
Low
...
Low
Normal

- Transformation
  - A function that maps the entire set of values of a given attribute to a new set of replacement values such that each old value can be identified with one of the new values
- Methods
  - **Discretization: Concept hierarchy climbing**
  - Smoothing: Remove noise from data
  - Attribute/feature construction
    - New attributes constructed from the given ones
  - Aggregation: Summarization, data cube construction
  - Normalization: Scaled to fall within a smaller, specified range

경희대학교 KYUNG HEE UNIVERSITY UTAS

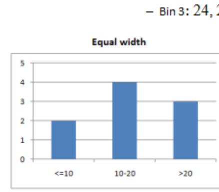
KCL2.5-SUC-02: Prepare lifelog and user profile data (3/4)

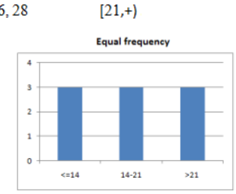
Missing Value Handler | Outlier Handler | **Transformation** | Features Selection

302

- **Discretization:** Concept hierarchy climbing
  - Divide the range of a continuous attribute into intervals
  - Interval labels can then be used to replace actual data values
  - Discretization can be performed recursively on an attribute
- Typical methods:
  - **Binning**
    - Equal-width (distance) partitioning
      - Divides the range into N intervals of equal size: uniform grid
      - if A and B are the lowest and highest values of the attribute, the width of intervals will be: W = (B –A)/N.
    - Equal-depth (frequency) partitioning
      - Divides the range into N intervals, each containing approximately same number of samples
  - Histogram analysis
  - Clustering analysis
  - Correlation analysis

- **Data :** 0, 4, 12, 16, 16, 18, 24, 26, 28
- **Equal width**
  - Bin 1: 0, 4 [-,10)
  - Bin 2: 12, 16, 16, 18 [10,20)
  - Bin 3: 24, 26, 28 [20,+)
- **Equal frequency**
  - Bin 1: 0, 4, 12 [-, 14)
  - Bin 2: 16, 16, 18 [14, 21)
  - Bin 3: 24, 26, 28 [21,+)

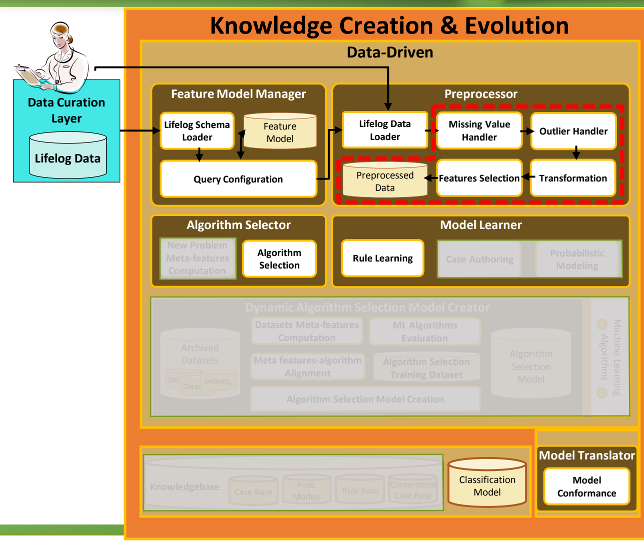




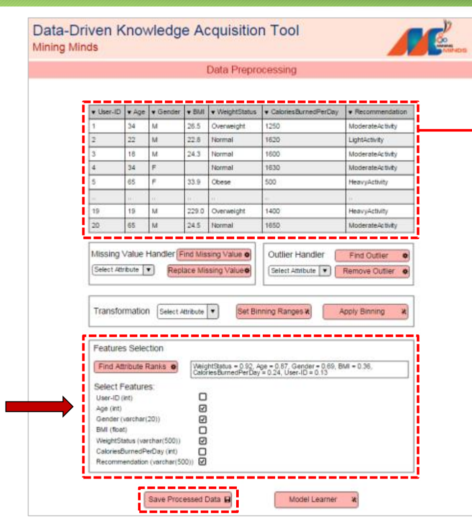
경희대학교 KYUNG HEE UNIVERSITY UTAS

Missing Value Handler | Outlier Handler | Transformation | **Features Selection**

- **Objective**
  - To preprocess the data to generate models with high accuracy.
- **Methodology**
  - ***Feature Selection*** *(operation)*
    1. **Input**: processed data
    2. **Processing**:
       a. Domain expert loads the processed data
       b. System displays the retrieved data
       c. Domain expert applies the attributes filtration techniques (i.e. ranking)
       d. System computes the ranks for all attributes and displays to expert
       e. Domain expert select the highly ranked attributes (i.e. rank value >= 0.8)
       f. System filters the attributes based on selected attributes and displays to domain expert
       g. Domain expert saves the processed data into repository
    3. **Output**: processed data



Missing Value Handler | Outlier Handler | Transformation | **Features Selection**



Unprocessed Data

User-ID	Age	Gender	BMI	WeightStatus	CaloriesBurnedPerDay	Recommendation
1	34	M	26.5	Overweight	1250	ModerateActivity
2	22	M	22.8	Normal	1620	LightActivity
3	18	M	24.3	Normal	1600	ModerateActivity
4	34	F		Normal	1630	ModerateActivity
5	65	F	33.9	Obese	500	HeavyActivity
..	..	..	..	..	...	..
19	19	M	229.0	Overweight	1400	HeavyActivity
20	65	M	24.5	Normal	1650	ModerateActivity

- Feature Selection
  - It is the process of selecting a subset of relevant features for use in model construction.
- Feature Selection Algorithms
  - Filter Methods
    - Chi Squared test
    - Information Gain score
    - Correlation Coefficient score
  - Wrapper Methods
  - Embedded Methods

$$Information\ Gain(S,F_i) = Entropy(S) - \sum_{v_i \epsilon V_{Fj}} \frac{|S_{vi}|}{|S|} * Entropy(S_{vi})$$

$$Entropy(S) = \sum_{i=1}^{c} -p_i * log_2(p_i)$$

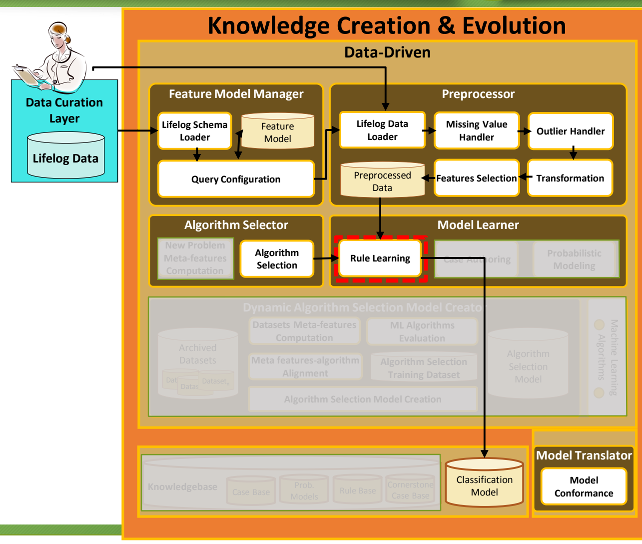
Where: p_i - proportion of the examples belonging to the i-th class.
S_{vi} – subset of S, for which feature F_j has value vi
V_{Fj} – set of all possible values of feature F_j

KCL2.5-SUC-13: Learn classification model

Use Case

305

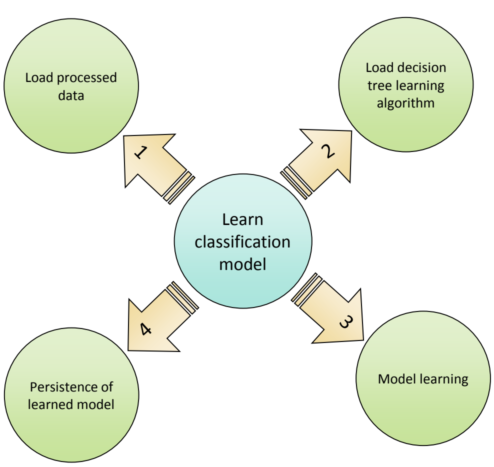
- **Objective**
  - To generate the classification model.
- **Methodology**
  - ***Rule Learning*** *(operation)*
    1. **Input**: processed data
    2. **Processing**:
       a. Domain expert loads the user profile lifelog processed data for selected domain
       b. System loads the corresponding processed data
       c. Domain expert selects the J48 learning algorithm
       d. System loads the J48 decision tree algorithm and generate the classification model
       e. Domain expert saves the model.
    3. **Output**: classification model

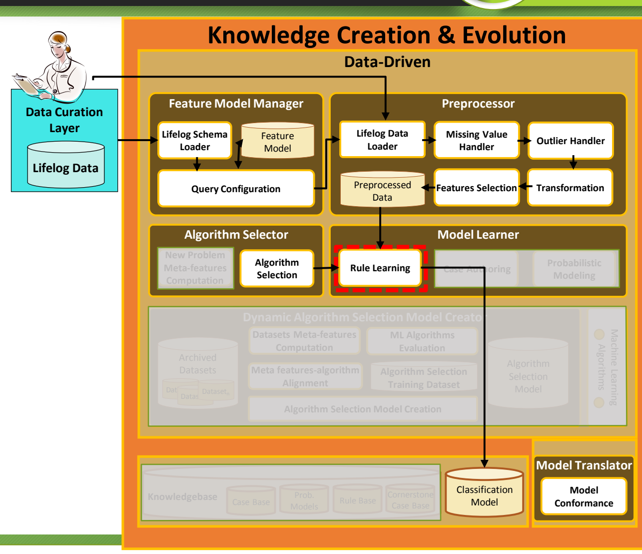


KCL2.5-SUC-13: Learn classification model

Use Case Functions

306

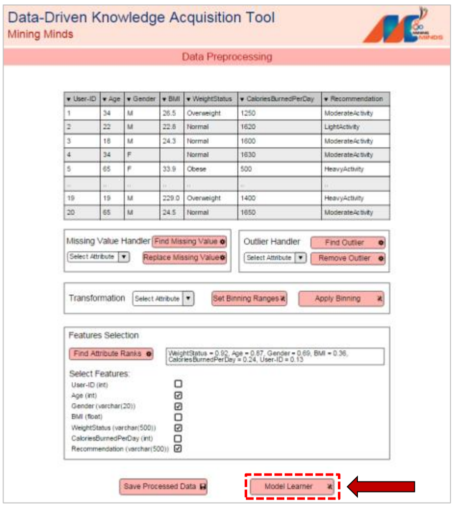


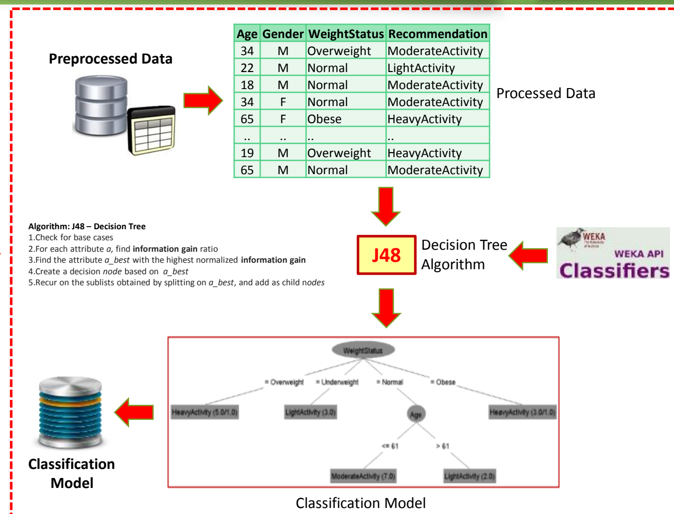


KCL2.5-SUC-13: Learn classification model

Implementation

307



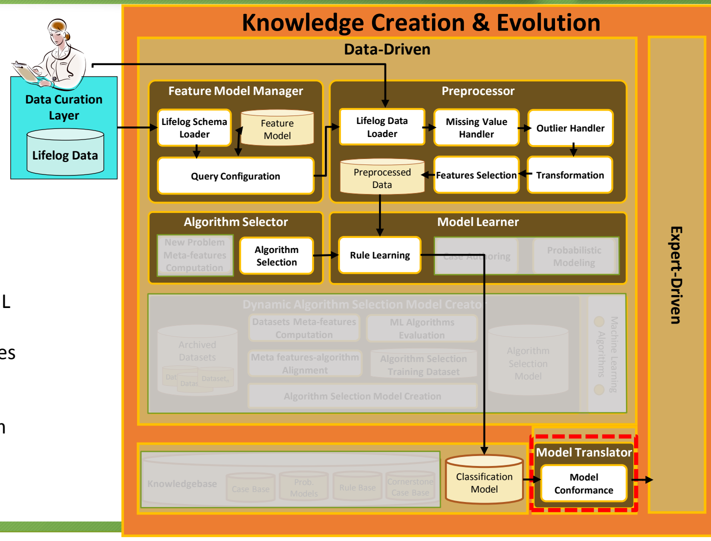


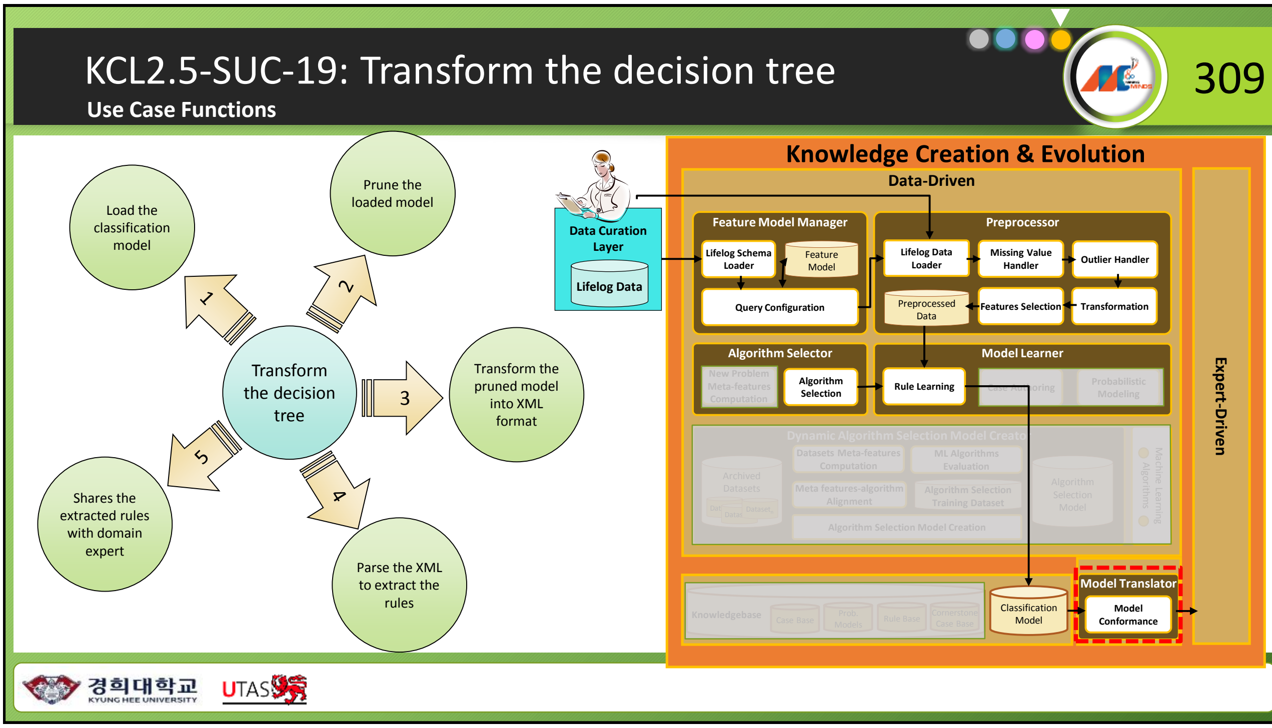
KCL2.5-SUC-19: Transform the decision tree

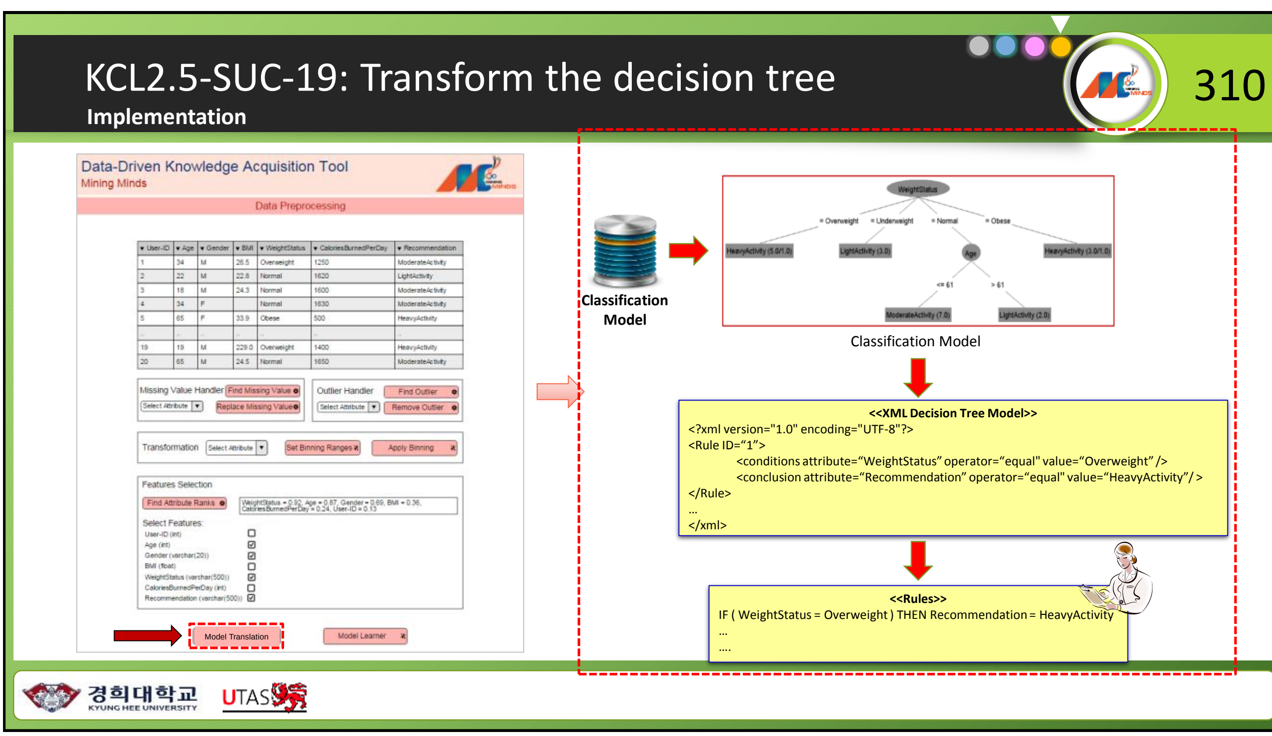
Use Case

308

- **Objective**
  - To transform the decision tree into rules to conform them from domain expert
- **Methodology**
  - ***Model Conformance*** *(operation)*
    1. **Input**: classification model
    2. **Processing**:
       a. System loads the classification model called decision tree model
       b. System prunes the decision tree model
       c. System translates the pruned model into XML format
       d. System parses the XML file to extract the rules from XML file
       e. System connects to Expert-Driven editor to share the parsed rules to conform them from domain expert
    3. **Output**: rules

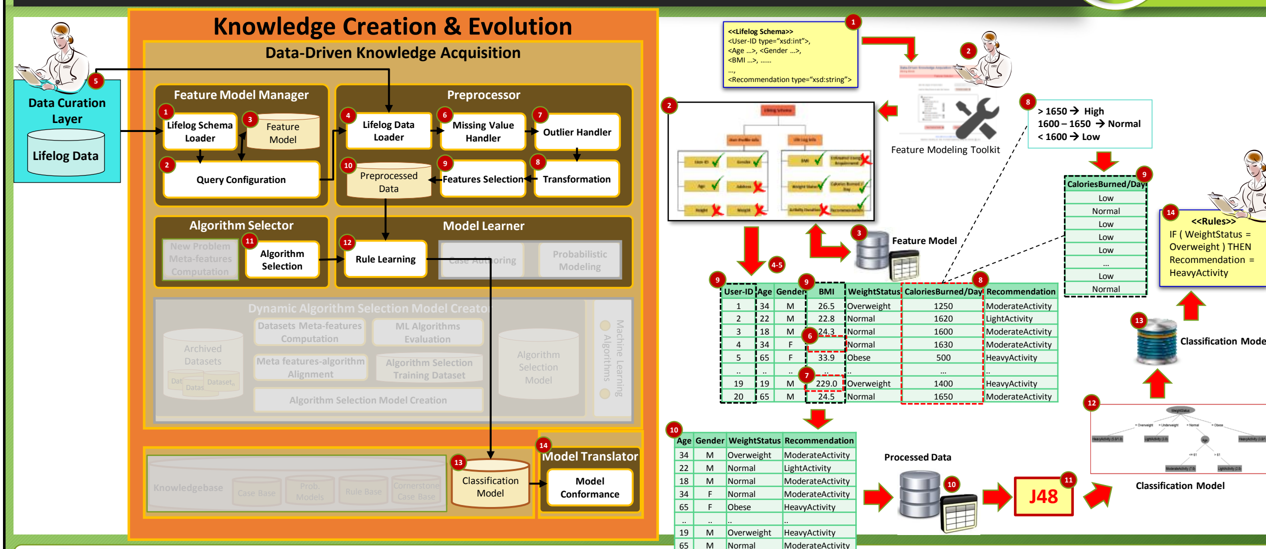



KCL2.5-SUC-19: Transform the decision tree
Use Case Functions
309
Load the classification model
Prune the loaded model
Transform the decision tree
Transform the pruned model into XML format
Parse the XML to extract the rules
Shares the extracted rules with domain expert
1
2
3
4
5
Data Curation Layer
Lifelog Data
Knowledge Creation & Evolution
Data-Driven
Expert-Driven
Feature Model Manager
Lifelog Schema Loader
Feature Model
Query Configuration
Preprocessor
Lifelog Data Loader
Missing Value Handler
Outlier Handler
Preprocessed Data
Features Selection
Transformation
Algorithm Selector
Algorithm Selection
Model Learner
Rule Learning
Model Translator
Classification Model
Model Conformance
경희대학교
KYUNG HEE UNIVERSITY
UTAS


KCL2.5-SUC-19: Transform the decision tree
Implementation
310
Data-Driven Knowledge Acquisition Tool
Mining Minds
Data Preprocessing
Missing Value Handler
Find Missing Value
Replace Missing Value
Outlier Handler
Find Outlier
Remove Outlier
Transformation
Set Binning Ranges
Apply Binning
Features Selection
Find Attribute Ranks
Select Features:
Model Translation
Model Learner
Classification Model
WeightStatus
Classification Model
<<XML Decision Tree Model>>
<?xml version="1.0" encoding="UTF-8"?>
<Rule ID="1">
<conditions attribute="WeightStatus" operator="equal" value="Overweight" />
<conclusion attribute="Recommendation" operator="equal" value="HeavyActivity"/ >
</Rule>
…
</xml>
<<Rules>>
IF (WeightStatus = Overweight) THEN Recommendation = HeavyActivity
…
….
경희대학교
KYUNG HEE UNIVERSITY
UTAS

Workflow

311



Dependencies, Issues and Challenges

312

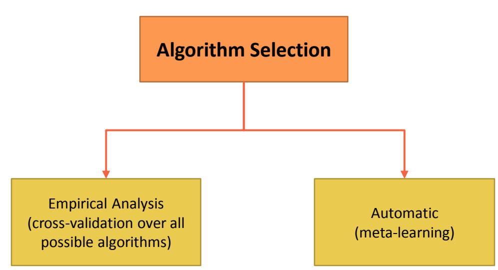
- The user profile and life log schema and data retrieval from DCL
- Feature model development
- Data preprocessing
- Model learning after configuration with available machine learning algorithms API's
- Model Translation

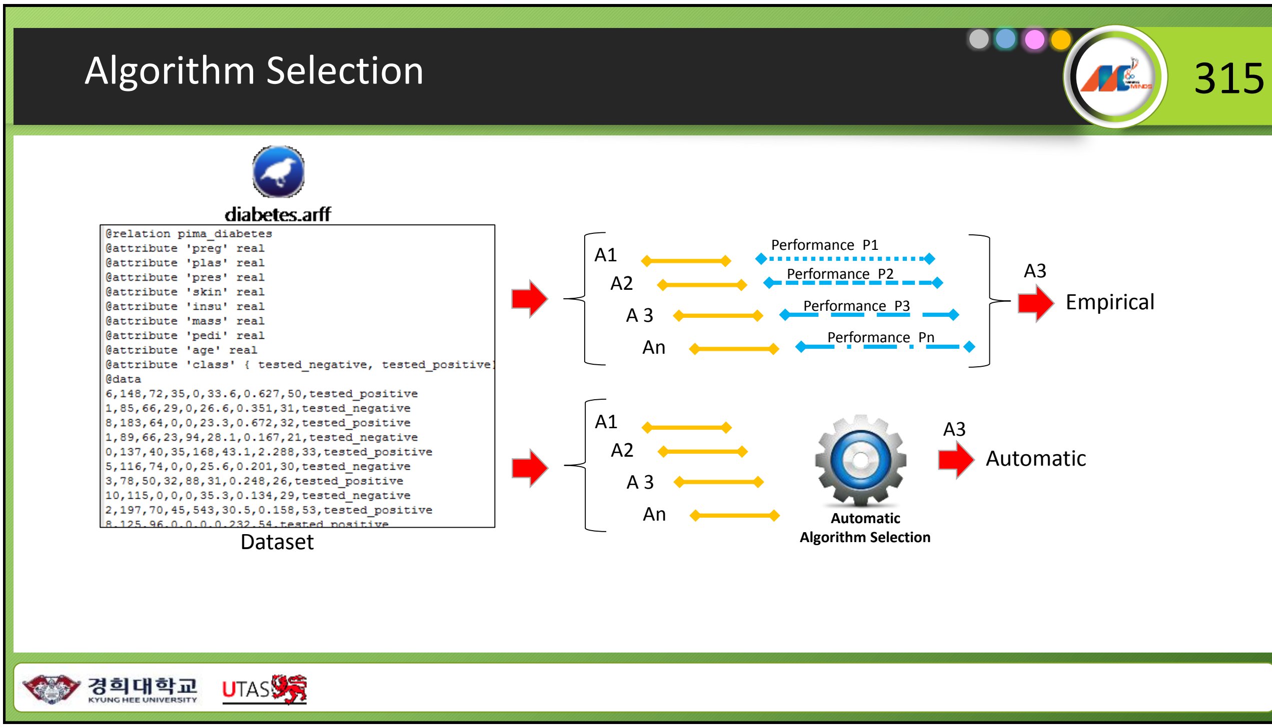
Selected Tools and Techniques

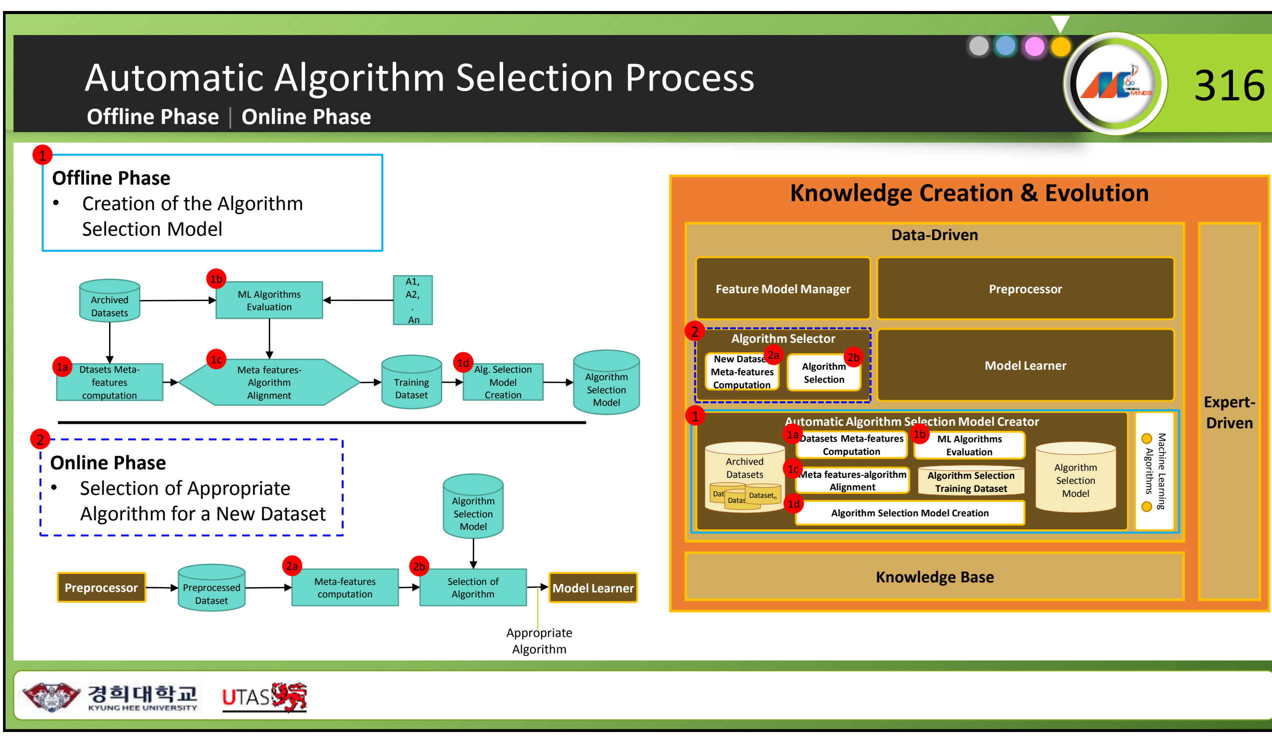
- Language
  - Java
- Frameworks:
  - Spring MVC
  - AJAX - jQuery
- APIs:
  - Weka
- IDEs:
  - Eclipse

Algorithm Selection

- The process of selecting an appropriate algorithm for learning a dataset
- Types of selecting appropriate algorithms
  - Empirical analysis
  - Automatic recommendation



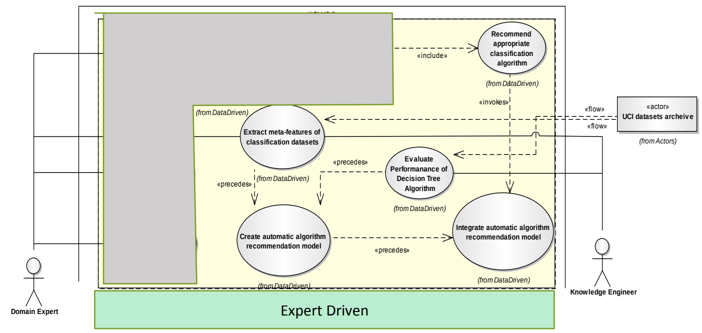

Algorithm Selection
315
diabetes.arff
@relation pima_diabetes
@attribute 'preg' real
@attribute 'plas' real
@attribute 'pres' real
@attribute 'skin' real
@attribute 'insu' real
@attribute 'mass' real
@attribute 'pedi' real
@attribute 'age' real
@attribute 'class' { tested_negative, tested_positive}
@data
6,148,72,35,0,33.6,0.627,50,tested_positive
1,85,66,29,0,26.6,0.351,31,tested_negative
8,183,64,0,0,23.3,0.672,32,tested_positive
1,89,66,23,94,28.1,0.167,21,tested_negative
0,137,40,35,168,43.1,2.288,33,tested_positive
5,116,74,0,0,25.6,0.201,30,tested_negative
3,78,50,32,88,31,0.248,26,tested_positive
10,115,0,0,0,35.3,0.134,29,tested_negative
2,197,70,45,543,30.5,0.158,53,tested_positive
Dataset
A1
A2
A 3
An
Performance P1
Performance P2
Performance P3
Performance Pn
A3
Empirical
A1
A2
A 3
An
Automatic
Algorithm Selection
A3
Automatic
경희대학교
KYUNG HEE UNIVERSITY
UTAS


Automatic Algorithm Selection Process
Offline Phase | Online Phase
316
1
Offline Phase
• Creation of the Algorithm Selection Model
Archived Datasets
1b
ML Algorithms Evaluation
A1, A2, . An
1a
Dtasets Meta-features computation
1c
Meta features-Algorithm Alignment
Training Dataset
1d
Alg. Selection Model Creation
Algorithm Selection Model
2
Online Phase
• Selection of Appropriate Algorithm for a New Dataset
Algorithm Selection Model
Preprocessor
Preprocessed Dataset
2a
Meta-features computation
2b
Selection of Algorithm
Model Learner
Appropriate Algorithm
Knowledge Creation & Evolution
Data-Driven
Feature Model Manager
Preprocessor
2
Algorithm Selector
New Dataset Meta-features Computation
2a
2b
Algorithm Selection
Model Learner
1
Automatic Algorithm Selection Model Creator
Archived Datasets
1a
Datasets Meta-features Computation
1b
ML Algorithms Evaluation
1c
Meta features-algorithm Alignment
Algorithm Selection Training Dataset
Algorithm Selection Model
1d
Algorithm Selection Model Creation
Machine Learning Algorithms
Expert-Driven
Knowledge Base
경희대학교
KYUNG HEE UNIVERSITY
UTAS

Use-case diagram of automatic algorithm selection

317

Use Case #ID	Use Case Name
KCL2.5-SUC-07	Extract meta-features of classification datasets
KCL2.5-SUC-08	Evaluate performance of classification algorithms
KCL2.5-SUC-09	Create automatic algorithm selection Model
KCL2.5-SUC-10	Integrate automatic algorithm recommendation model
KCL2.5-SUC-11	Recommend appropriate classification algorithm

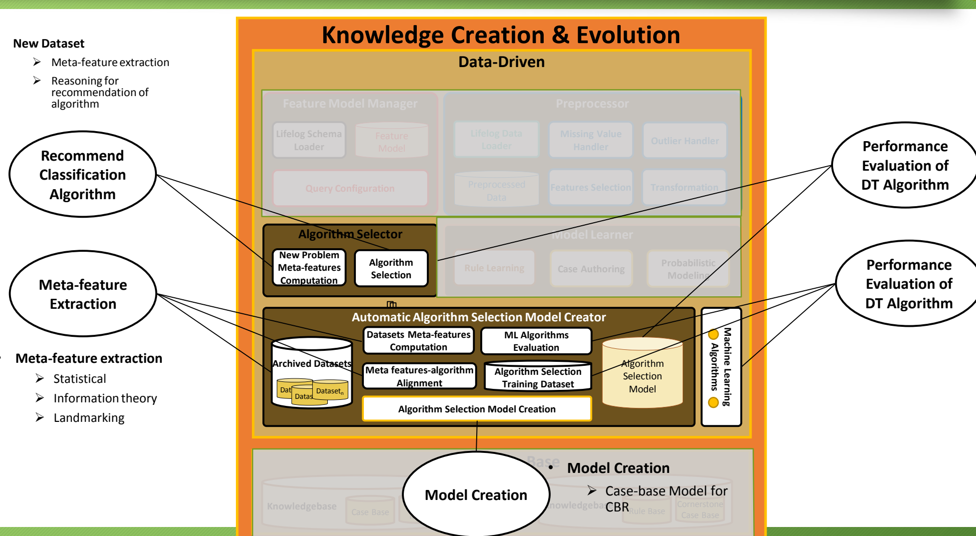


Use-case diagram of automatic algorithm selection

318

- New Dataset
  - Meta-feature extraction
  - Reasoning for recommendation of algorithm

- Meta-feature extraction
  - Statistical
  - Information theory
  - Landmarking



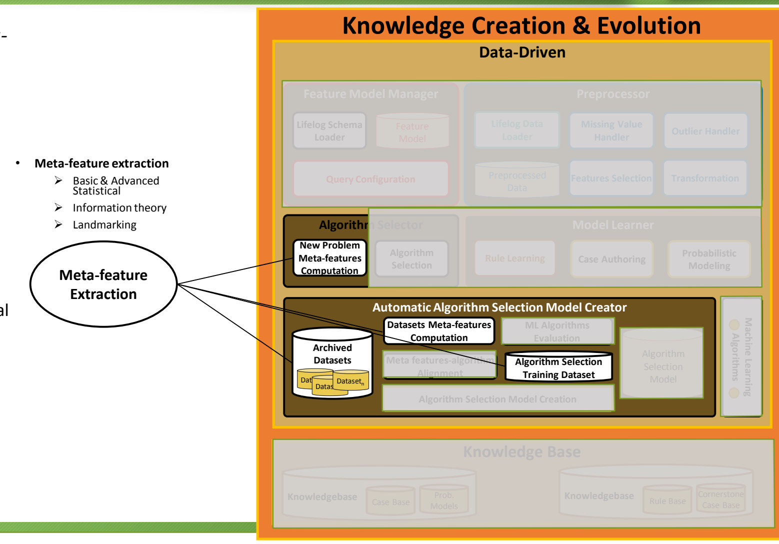
Meta-Features Extraction

Use Case

319

Use-case (KCL2.5-SUC-07): *Extract meta-features of classification dataset*

- **Objective**
  - Extraction of meta-characteristics
- **Methodology**
  - *Library used*: OpenML dataset characterization library is used
  - ***Dataset Meta-features computation*** *(operation)*
    1. **Input**: UCI and OpenMl classification datasets
    2. **Processing**:
       a. Takes each dataset form the local copy of the datasets
       b. Extracts basic and advanced statistical meta-feature and information theory features
       c. Stores the extracted features to meta-feature base
    3. **Output**:
       a. A set of meta-features

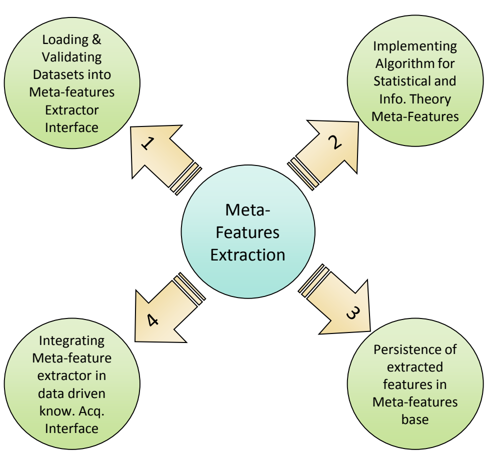


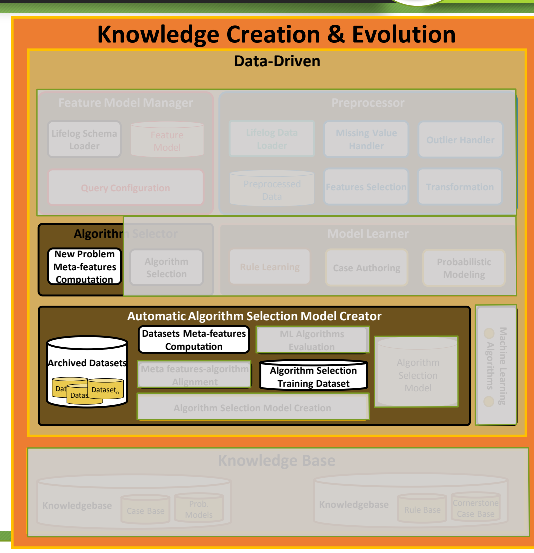
경희대학교 KYUNG HEE UNIVERSITY UTAS

Meta-Features Extraction

Use Case Features Functions

320





경희대학교 KYUNG HEE UNIVERSITY UTAS

List of Meta-features

321

No.	Basic Statistical Features
Simp. Statistic 1	InstanceCount
Simp. Statistic 2	NumAttributes
Simp. Statistic 3	ClassCount
Simp. Statistic 4	PercentageOfBinaryAtts
Simp. Statistic 5	PercentageOfNominalAtts
Simp. Statistic 6	PercentageOfNumericAtts
Simp. Statistic 7	AttrWithOutlier.Prop
Simp. Statistic 8	MeanSkewnessOfNumericAtts
Simp. Statistic 9	MeanKurtosisOfNumericAtts
Simp. Statistic 10	MeanAbsCoef
Simp. Statistic 11	Dimensionality
Simp. Statistic 12	NumBinaryAtts
Simp. Statistic 13	NumNominalAtts
Simp. Statistic 14	NumNumericAtts
Simp. Statistic 15	NumMissingValues

No.	Advanced Statistical Features
Advanced Statistic 1	MeanStdDevOfNumericAtts
Advanced Statistic 2	MeanMeansOfNumericAtts
Advanced Statistic 3	NegativePercentage
Advanced Statistic 4	PositivePercentage
Advanced Statistic 5	DefaultAccuracy
Advanced Statistic 6	IncompleteInstanceCount
Advanced Statistic 7	PercentageOfMissingValues
Advanced Statistic 8	MinNominalAttDistinctValues
Advanced Statistic 9	MaxNominalAttDistinctValues
Advanced Statistic 10	StdvNominalAttDistinctValues
Advanced Statistic 11	MeanNominalAttDistinctValues

No.	Information Theory Features
InfTheory 1	ClassEntropy
InfTheory 2	MeanAttributeEntropy
InfTheory 3	MeanMutualInformation
InfTheory 4	EquivalentNumberOfAtts
InfTheory 5	NoiseToSignalRatio

Total 31 Meta-features

경희대학교 KYUNG HEE UNIVERSITY UTAS

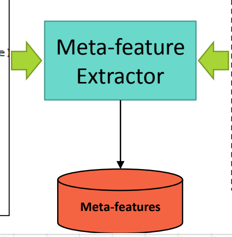
Meta-features Extraction (One Dataset)

322

diabetes.arff

```
@relation pima_diabetes
@attribute 'preg' real
@attribute 'plas' real
@attribute 'pres' real
@attribute 'skin' real
@attribute 'insu' real
@attribute 'mass' real
@attribute 'pedi' real
@attribute 'age' real
@attribute 'class' { tested_negative, tested_positive}
@data
6,148,72,35,0,33.6,0.627,50,tested_positive
1,85,66,29,0,26.6,0.351,31,tested_negative
8,183,64,0,0,23.3,0.672,32,tested_positive
1,89,66,23,94,28.1,0.167,21,tested_negative
0,137,40,35,168,43.1,2.288,33,tested_positive
5,116,74,0,0,25.6,0.201,30,tested_negative
3,78,50,32,88,31,0.248,26,tested_positive
10,115,0,0,0,35.3,0.134,29,tested_negative
2,197,70,45,543,30.5,0.158,53,tested_positive
8,125,96,0,0,0.232,54,tested_positive
```

Dataset



Mean Absolute Deviation
- The sum of the differences between data values and the mean, divided by the count;
- [(x1 - mean) + (x2 - mean) + (x3 - mean) + ... + (xn - mean)] / n

Skewness
- The sum of the cubed differences between data values and the mean, divided by the count minus 1 times the cubed standard deviation;
- [(x1 - mean)3 + (x2 - mean)3 + (x3 - mean)3 + ... + (xn - mean)3] / [(n - 1) * s3]

Kurtosis
- The sum of the fourth power of differences between data values and the mean, divided by the count minus 1 times the fourth power of the standard standard deviation;
- [(x1 - mean)4 + (x2 - mean)4 + (x3 - mean)4 + ... + (xn - mean)4] / [(n - 1) * s4]

Mean Standard Deviation (s)
- The square root of the variance;
- 2√variance or variance = s2

Root Mean Square (RMS)
-
-

Dataset N	MeanSkev	MeanKurt	MeanStdD	MeanMea	NumAttril	Dimensio	Percentag	Percentag	NumNom	Percentag	Nu
2dplanes.	-0.00382	-1.5503	0.834942	0.002006	11	2.70E-04	0	0	0	0.909091	
abalone-t	0.281157	1.154309	0.175304	0.37763	9	0.007183	0	0.111111	1	0.777778	
abalone-t	0.742422	14.44895	0.174304	0.384878	9	0.003078	0	0.111111	1	0.777778	
abalone.a	0.620072	11.01516	0.174618	0.382703	9	0.002155	0	0.111111	1	0.777778	
abe_148.a	-0.72778	8.382742	68.41312	28.97861	6	0.090909	0	0	0	0.833333	
acute-infl	0.110633	-1.5449	1.819132	38.72417	7	0.058333	0.714286	0.714286	5	0.142857	
ada_agno:	7.149791	257.7275	14.34559	34.15562	49	0.010741	0	0	0	0.979592	
ada_prior	2.830627	34.60191	19481.17	33078.85	15	0.003288	0.066667	0.533333	8	0.4	

경희대학교 KYUNG HEE UNIVERSITY UTAS

Meta-features Extraction (For All Datasets)

- Types of meta-features
  - Basic statistical (13)
  - Advanced statistical (11)
  - Information theory (5)
  - Complexity characteristics
  - Landmarking features
  - Model-based meta-features
- Types datasets
  - Classification datasets (60-100)
  - Clustering

OpenML Dataset Characterization Library

UCI and OpenML Classification Datasets → Meta-features

	mf_1	mf_2	mf_3	....	mf_{m-1}	mf_m
	Mean Absolute Deviation	Skewness	Mean Standard Deviation (s)		Class Entropy	Mean Mutul Inform
	0.750390634	2.231884058	13.209362	....	0.814765885	0.02238773
d_1	-1	-1	-1	....	0.991231	-1
d_2	-1	-1	-1	....	0.791645	-1
⋮	.	.	.	....	.	.
d_{n-1}	2.059351	3.21875	-13.2185	....	-1	-1
d_n	2.070336	2.3125	-23.5571	....	-1	-1

$$\#\,Exp.\,for\,meta-features\,computation = n\,(no\,datasets) * m(no\,metafetures)$$

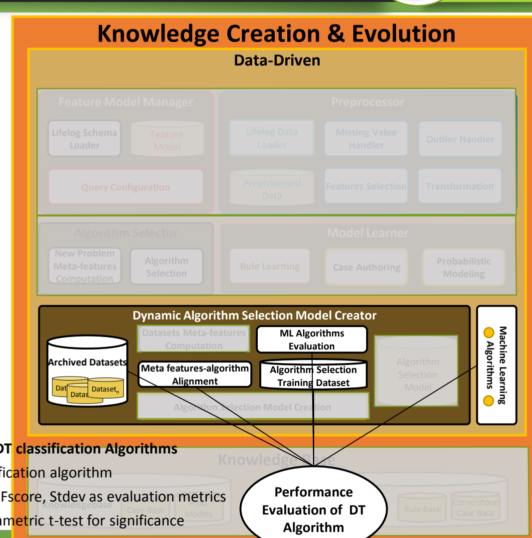
Performance Evaluation of Classification Algorithm

Use Case

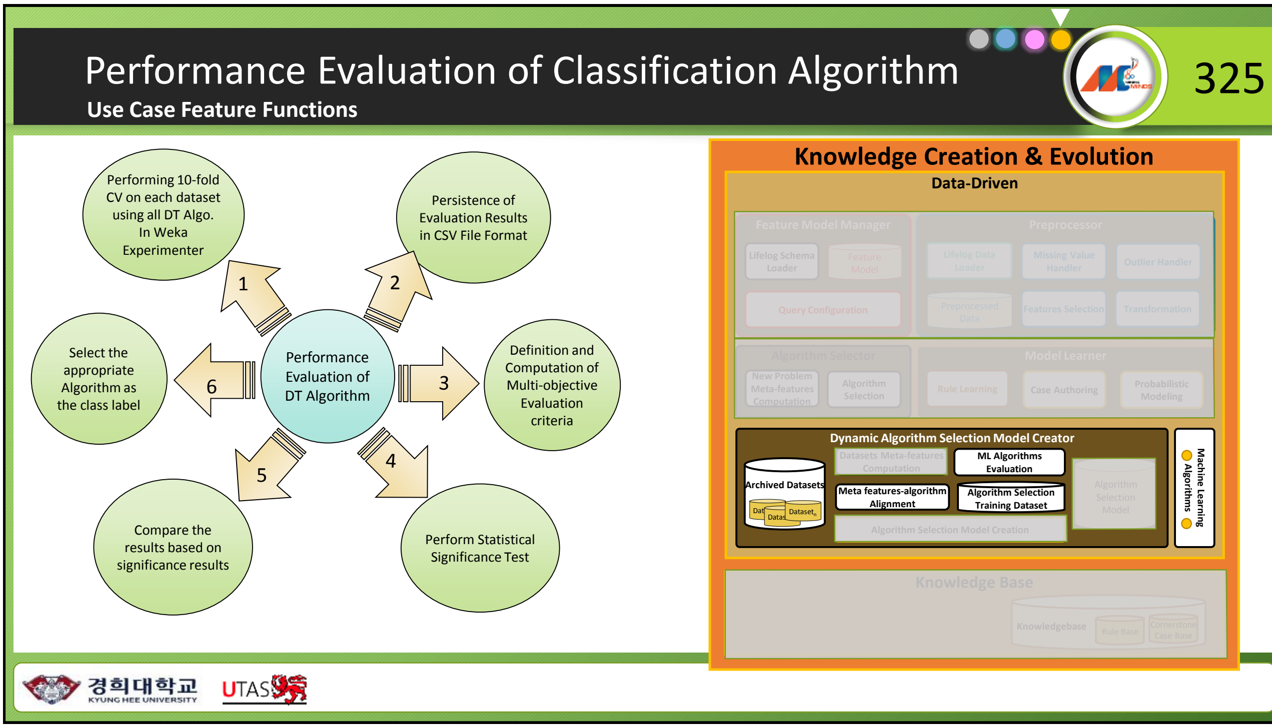
Use-case (KCL2.5-SUC-08): *Evaluate decision tree classification algorithms*

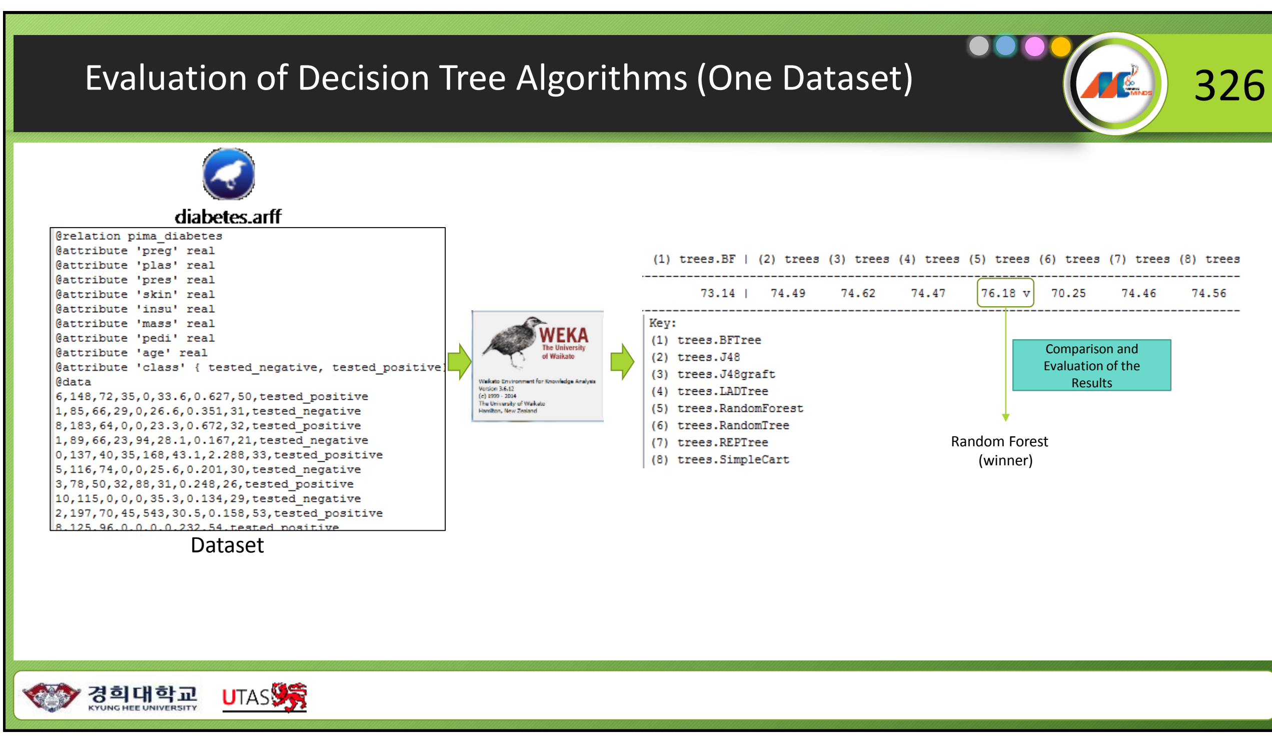
- **Objective**
  - To find the most appropriate DT algorithm for the datasets
- **Methodology**
  - *Library used*: Weka, Excel Statistician Tool
  - ***ML algorithm evaluation*** *(operation)*

1. **Input**: UCI and OpenMl classification datasets, Weka DT algorithms
2. **Processing**:
   a. Takes each dataset form the local copy of the archeived datasets
   b. Use Weka experimenter and test the DT algorithm on the loaded dataset
   c. Obtain all the evaluation metrics
   d. Compute Balanced accuracy
   e. Perform parametric test for the significantly better DT algorithm
   f. Select the appropriate algorithm
   g. Align selected algorithm with the meta-features of that dataset
   h. Select appropriate algorithm as a class label
3. **Output**: Appropriate algorithm



- **Evaluation of DT classification Algorithms**
  - DT classification algorithm
  - Wgt.Avg.Fscore, Stdev as evaluation metrics
  - Non-parametric t-test for significance

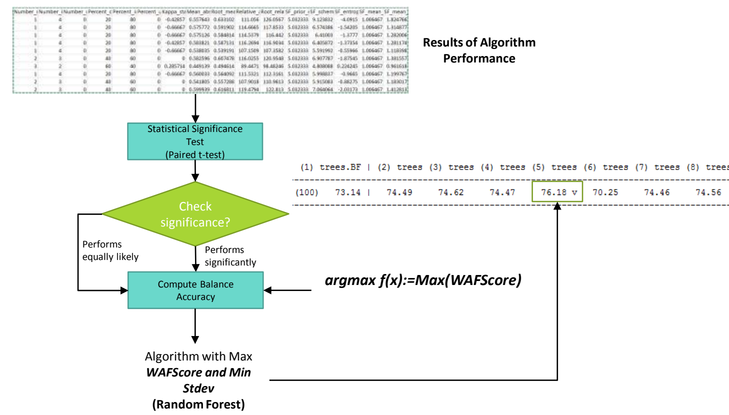

Performance Evaluation of Classification Algorithm
Use Case Feature Functions
325
Performing 10-fold CV on each dataset using all DT Algo. In Weka Experimenter
Persistence of Evaluation Results in CSV File Format
Select the appropriate Algorithm as the class label
Performance Evaluation of DT Algorithm
Definition and Computation of Multi-objective Evaluation criteria
Compare the results based on significance results
Perform Statistical Significance Test
1
2
3
4
5
6
Knowledge Creation & Evolution
Data-Driven
Feature Model Manager
Preprocessor
Algorithm Selector
Model Learner
Dynamic Algorithm Selection Model Creator
ML Algorithms Evaluation
Archived Datasets
Meta features-algorithm Alignment
Algorithm Selection Training Dataset
Algorithm Selection Model Creation
Machine Learning Algorithms
Knowledge Base
경희대학교
KYUNG HEE UNIVERSITY
UTAS


Evaluation of Decision Tree Algorithms (One Dataset)
326
diabetes.arff
@relation pima_diabetes
@attribute 'preg' real
@attribute 'plas' real
@attribute 'pres' real
@attribute 'skin' real
@attribute 'insu' real
@attribute 'mass' real
@attribute 'pedi' real
@attribute 'age' real
@attribute 'class' { tested_negative, tested_positive}
@data
6,148,72,35,0,33.6,0.627,50,tested_positive
1,85,66,29,0,26.6,0.351,31,tested_negative
8,183,64,0,0,23.3,0.672,32,tested_positive
1,89,66,23,94,28.1,0.167,21,tested_negative
0,137,40,35,168,43.1,2.288,33,tested_positive
5,116,74,0,0,25.6,0.201,30,tested_negative
3,78,50,32,88,31,0.248,26,tested_positive
10,115,0,0,0,35.3,0.134,29,tested_negative
2,197,70,45,543,30.5,0.158,53,tested_positive
Dataset
WEKA
The University of Waikato
(1) trees.BF | (2) trees (3) trees (4) trees (5) trees (6) trees (7) trees (8) trees
73.14 | 74.49 74.62 74.47 76.18 v 70.25 74.46 74.56
Key:
(1) trees.BFTree
(2) trees.J48
(3) trees.J48graft
(4) trees.LADTree
(5) trees.RandomForest
(6) trees.RandomTree
(7) trees.REPTree
(8) trees.SimpleCart
Comparison and Evaluation of the Results
Random Forest (winner)
경희대학교
KYUNG HEE UNIVERSITY
UTAS

Comparison and Evaluation of Results (One Dataset)

327

- Perform statistical significance test (Paired t-test)
  1. Perform significantly better
  2. Perform equally likely, cannot decide
  3. Perform significantly poor
- Drops algorithms that performs significantly poor
- Perform evaluation for balanced accuracy
  - Wgt.Avg.F-Score, Stdev
- Select the algorithm with maximum highest Wgt.Avg.F-Score value and minimum Stdev

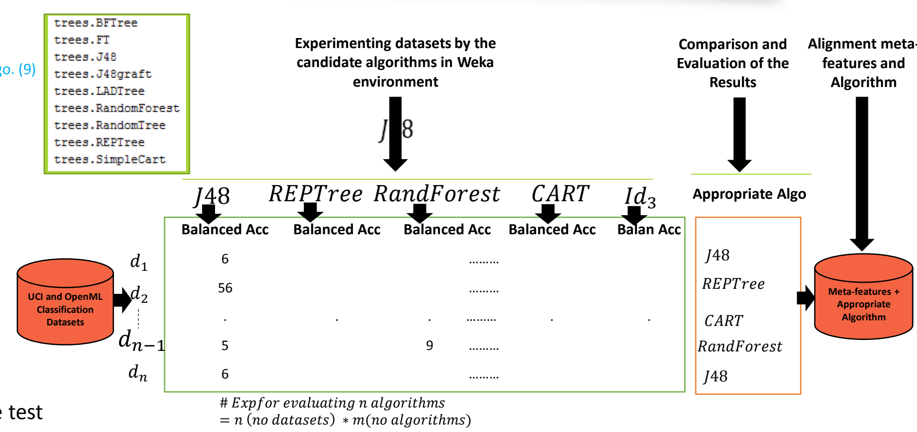


경희대학교 KYUNG HEE UNIVERSITY UTAS

Comparison and Evaluation of Results (All Datasets)

328

- Types algorithms
  - Decision tree-based algo. (9)
  - Rule-based algo.
  - Probabilistic algo.
  - Distance-based algo.
  - Meta-learning algo.
  - Function-based algo.
- Evaluation criteria
  - Accuracy
  - Wgt.Avg.F-score
  - Stdev
  - Balanced accuracy
  - Precision
  - Error rate
  - Time-complexity
  - .
  - .
- Statistical significance test
  - Parametric
  - Non-parametric



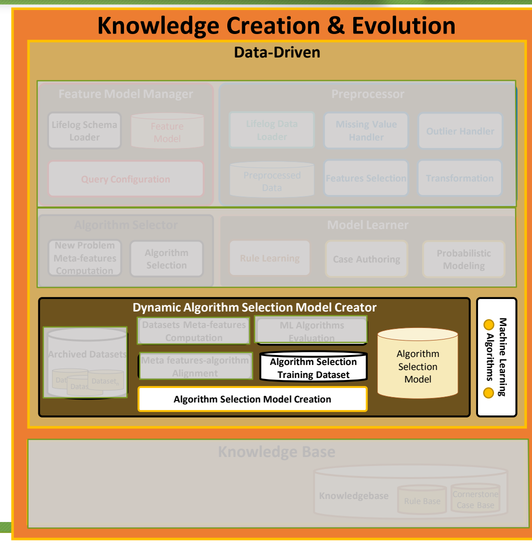
경희대학교 KYUNG HEE UNIVERSITY UTAS

Algorithm Selection Model Creation

Use Case

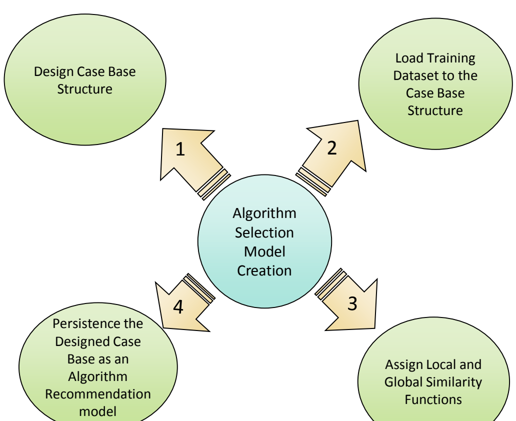
Use-case (KCL2.5-SUC-09): *Create Algorithm Selection Model (Algorithm Selection Case Base)*

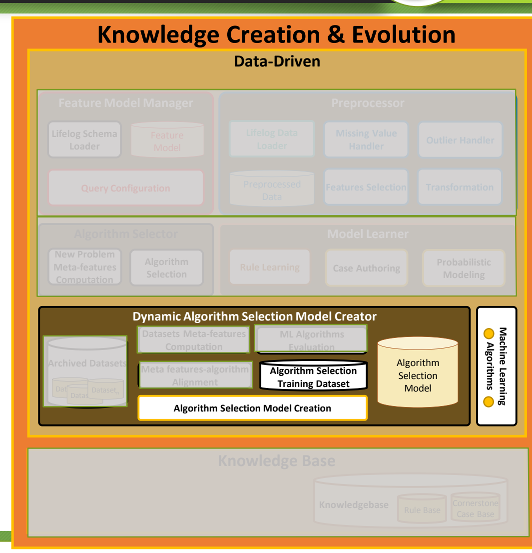
- **Objective**
  - To Create the Case base for recommending best algorithm
- **Methodology**
  - *Library used*: jColibri, Excel Statistician Tool
  - ***Algorithm Selection Case Base Creation*** *(operation)*
    1. **Input**: Algorithm Selection Training Dataset, Similarity Functions
    2. **Processing**:
       a. Define Case base structure in jColibri
       b. Takes Algorithm Selection Training Dataset and load to Case base structure in jColibri
       c. Assign local similarity functions to each meta-feature
       d. Assign global similarity functions to each meta-feature
       e. Save the Algorithm Selection Case Base as a CBR Model
    3. **Output**: Algorithm Selection Model (CBR Case Base)



Algorithm Selection Model Creation

Use Case Feature Functions



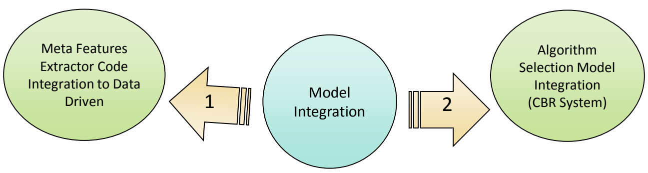


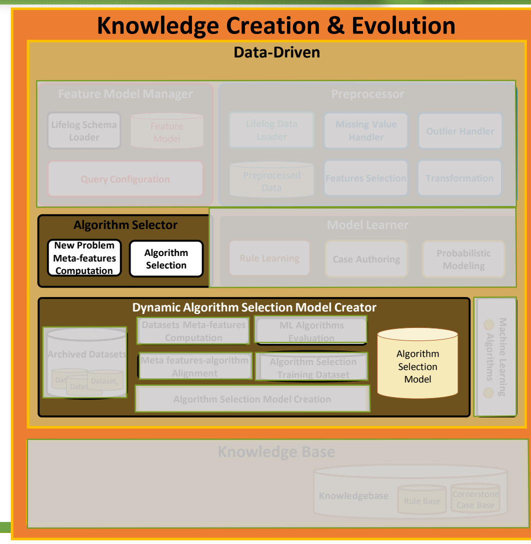
Algorithm Selection Model Integration

Use Case Feature Functions

Use-case (KCL2.5-SUC-10): *Integrate Algorithm Selection Model*

- **Objective**
  - To Select Best Algorithm from The Data Driven Environment
- **Methodology**
  - *Library used*: Net Beans or Eclipse
  - ***Algorithm Selection Model Integration****(operation)*
    1. **Input**: Algorithm Selection Model
    2. **Processing**:
       a. Integrate meta-feature Extractor
       b. Integrate CBR algorithm selection model
    3. **Output**: Integrated Algorithm Selection Model

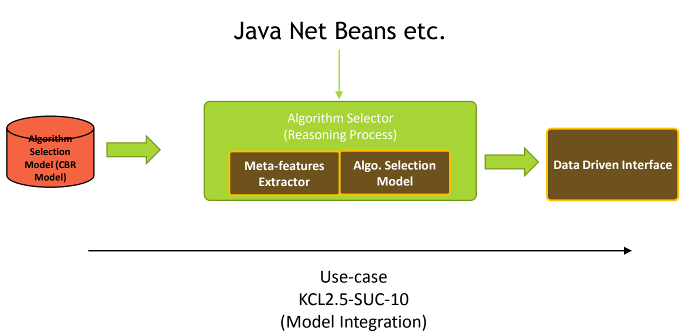




경희대학교 KYUNG HEE UNIVERSITY UTAS

Algorithm Selection Model Integration

Tasks Flow



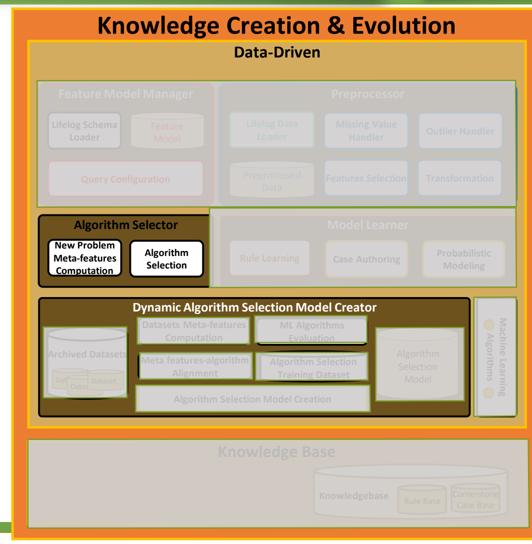
Algorithm Recommendation

Use Case

333

Use-case (KCL2.5-SUC-11): *Recommendation of best Algorithm*

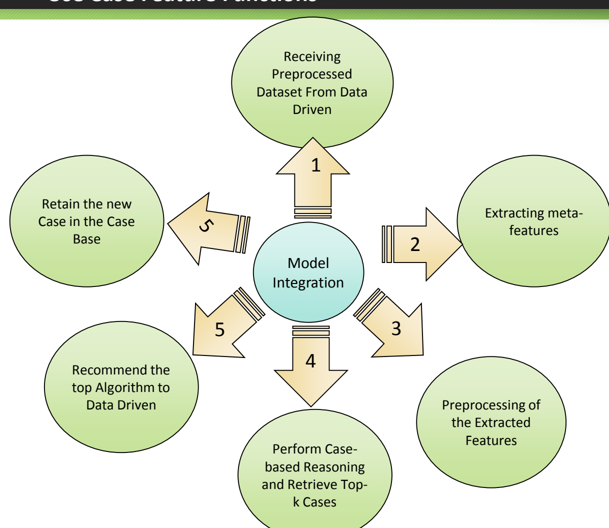
- **Objective**
  - To select best algorithm
- **Methodology**
  - *Library used*: jColibri
  - ***ML algorithm Recommendation*** *(operation)*
    1. **Input**: new preprocessed datasets, CBR model
    2. **Processing**:
       a. Takes preprocessed dataset form data driven environment
       b. Extract meta-features through meta-feature extractor
       c. Apply Retrieve step of the CBR methodology
       d. Select top-k similar cases
       e. Select the top case as the recommended algorithm
       f. Provide the recommendation algorithm to the data driven environment
       g. Retain the new case in Case Base
    3. **Output**: the best algorithm

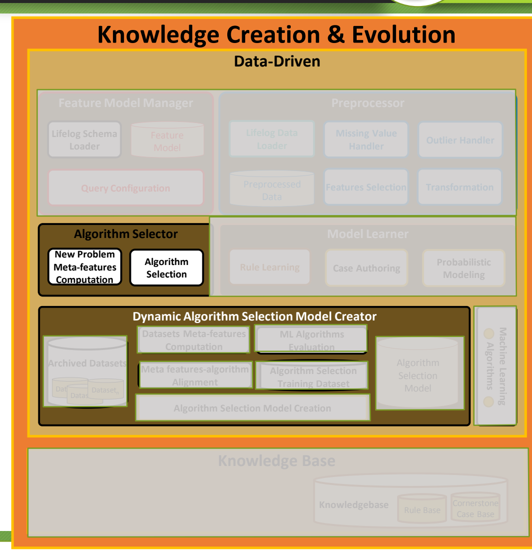


Algorithm Recommendation

Use Case Feature Functions

334

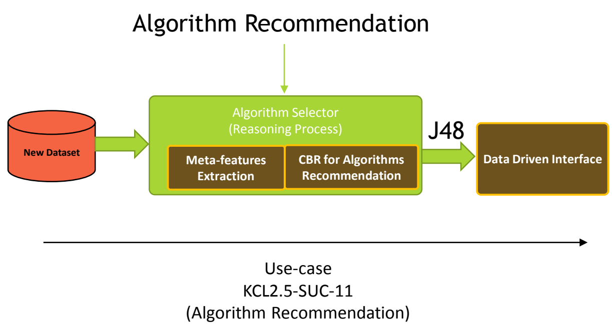




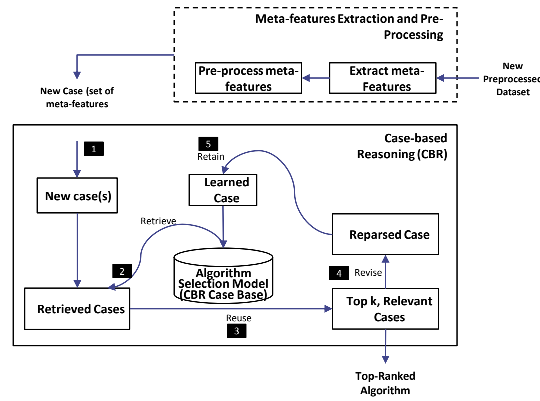
Algorithm Recommendation

335

Task WorkFlow



CBR Algorithm for Algorithm Recommendation



Selected Tools and Techniques

336

- APIs:
  - Weka
  - OpenML
- Tools:
  - Weka
  - RapidMiner
  - jColibriv2.0

WEKA The University of Waikato

jCOLIBRI 2 Case-Based Reasoning Framework

OpenML

rapidminer

경희대학교 KYUNG HEE UNIVERSITY UTAS

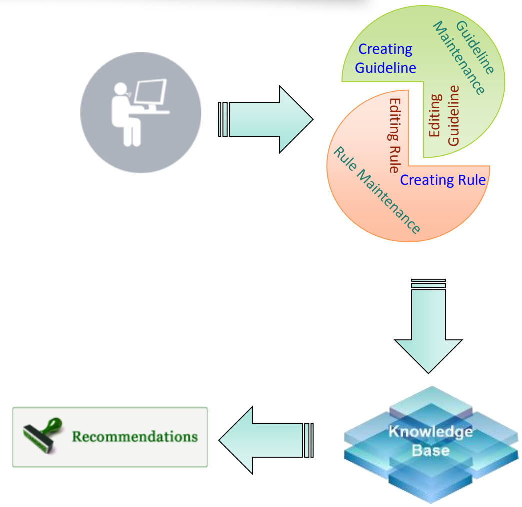
Contributions

- Support of *feature modeling* to select valid combination of features for optimal classification model
- Generating *classification model*
- Use of *multi meta-features learning* to exploit intrinsic behaviors of datasets for improved performance of the algorithm selection model
- Use of *multi-metric objective* function to evaluate algorithms performance and recommend appropriate algorithm
- *Automatic recommendation* of optimum performance algorithm for a new dataset at run time

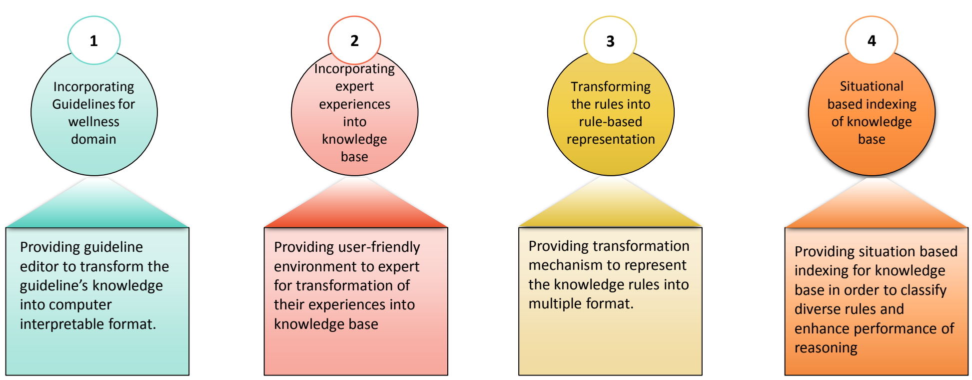
Expert Driven Knowledge Acquisition

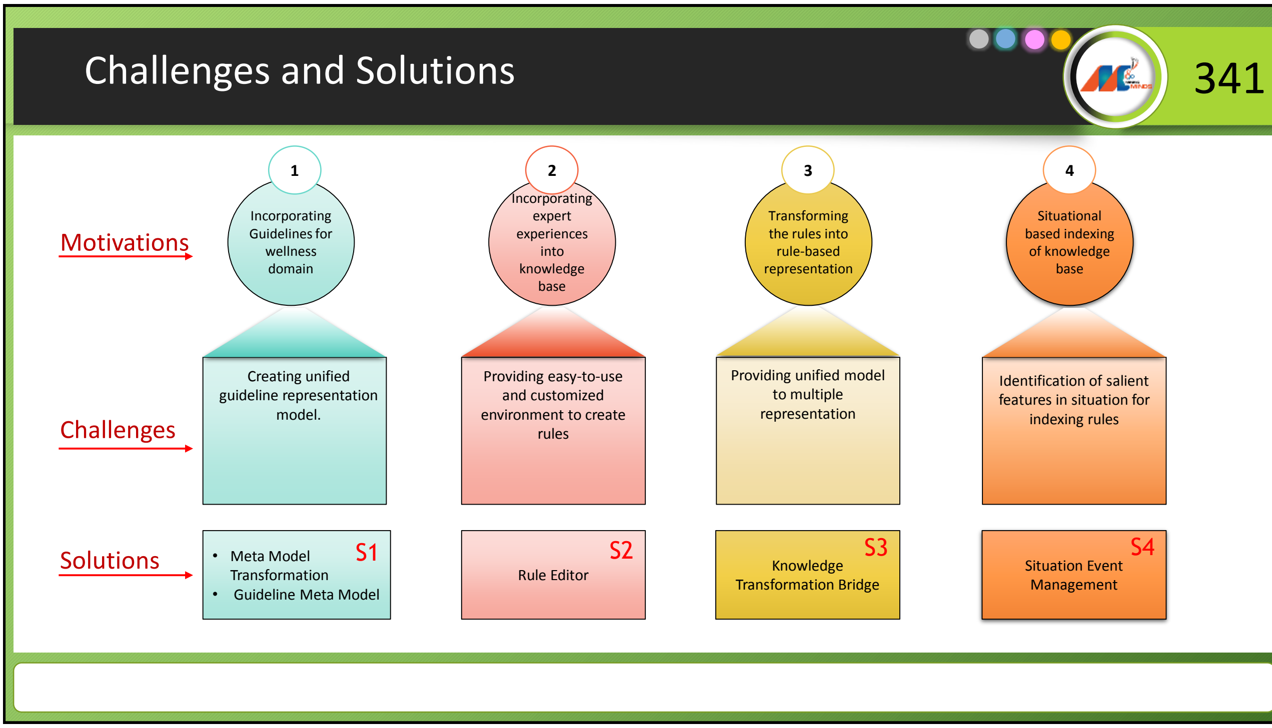
Goal and Objectives

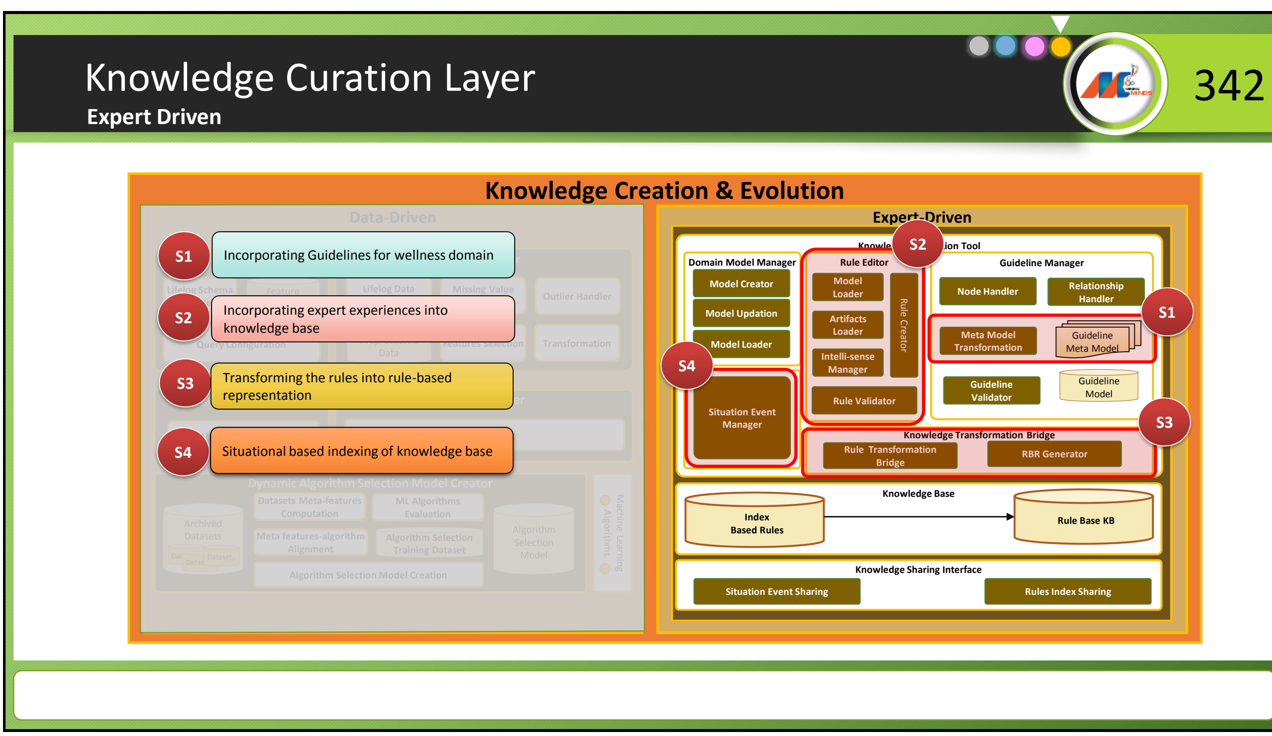
- Goal
  - Provide user-friendly environment to create recommendation's and alert's guidelines to transform experts knowledge into knowledge base
- Objectives
  - ❖Create easy-to-use Rule Editor to facilitate the domain experts to create knowledge rules using contextual selection of concepts from Intelli-sense window and/or domain model tree
  - ❖Providing user-friendly Guideline Editor to generate guidelines with the help of Intelli-sense and domain model tree selection as in Rule Editor
  - ❖Transformation of guidelines into knowledge rules and then to executable format to generate recommendations and alerts



Motivations



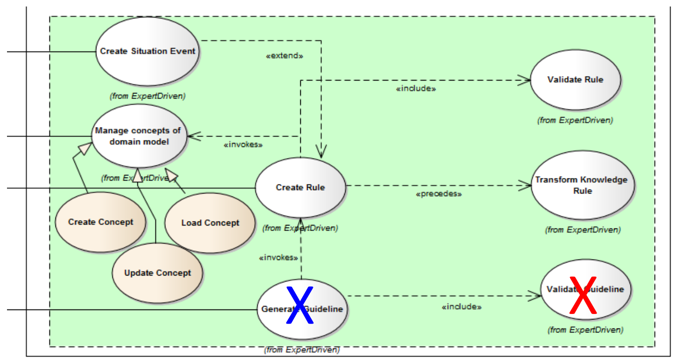

Challenges and Solutions
341
Motivations
Challenges
Solutions
1
Incorporating Guidelines for wellness domain
2
Incorporating expert experiences into knowledge base
3
Transforming the rules into rule-based representation
4
Situational based indexing of knowledge base
Creating unified guideline representation model.
Providing easy-to-use and customized environment to create rules
Providing unified model to multiple representation
Identification of salient features in situation for indexing rules
S1
• Meta Model Transformation
• Guideline Meta Model
S2
Rule Editor
S3
Knowledge Transformation Bridge
S4
Situation Event Management


Knowledge Curation Layer
Expert Driven
342
Knowledge Creation & Evolution
Data-Driven
Expert-Driven
S1
Incorporating Guidelines for wellness domain
S2
Incorporating expert experiences into knowledge base
S3
Transforming the rules into rule-based representation
S4
Situational based indexing of knowledge base
Domain Model Manager
Model Creator
Model Updation
Model Loader
Situation Event Manager
Rule Editor
Model Loader
Artifacts Loader
Intelli-sense Manager
Rule Creator
Rule Validator
Guideline Manager
Node Handler
Relationship Handler
Meta Model Transformation
Guideline Meta Model
Guideline Validator
Guideline Model
Knowledge Transformation Bridge
Rule Transformation Bridge
RBR Generator
Knowledge Base
Index Based Rules
Rule Base KB
Knowledge Sharing Interface
Situation Event Sharing
Rules Index Sharing

Expert Driven Knowledge Acquisition-Use cases

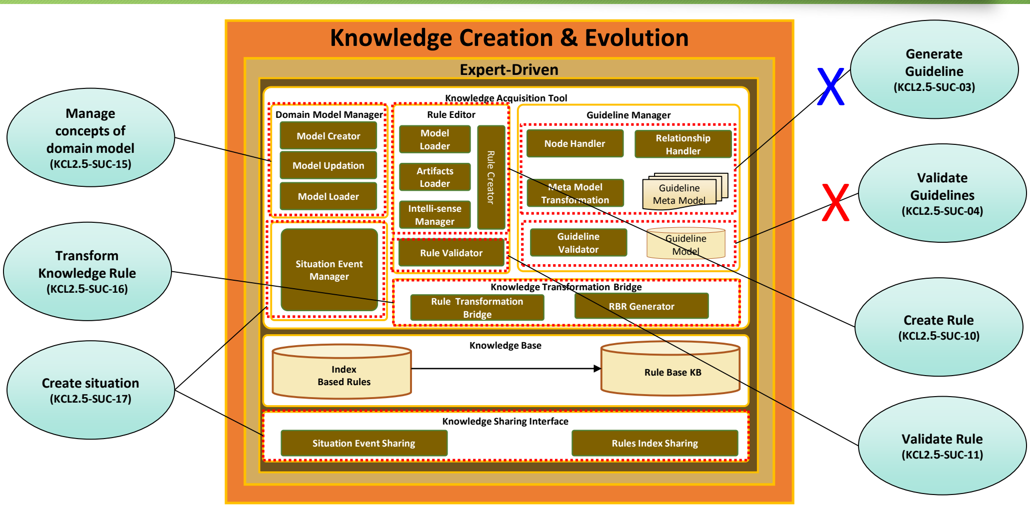
343

Use Case #ID	Description
KCL2.5-SUC-03	Generate Guideline
KCL2.5-SUC-04	Validate Guideline
KCL2.5-SUC-10	Create Rule
KCL2.5-SUC-11	Validate Rule
KCL2.5-SUC-15	Manage Concepts of Domain Model
KCL2.5-SUC-16	Transform Knowledge Rule
KCL2.5-SUC-17	Create Situation Event
KCL2.5-SUC-20	Create Concept
KCL2.5-SUC-21	Update Concept
KCL2.5-SUC-22	Load Concept



Expert Driven Knowledge Acquisition-Use cases

344



Expert Driven Knowledge Scenario for #2 path

345

6 Share Rule with corresponding index

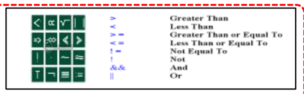
Rule Index:
Rule Condition → Rule 1

5 Rule and Index saved into KB

4 Rule is validated

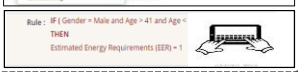
Rule 1:
If Gender = Male and Age Group = Adult (19-45) and current Activity = Sitting and Activity Duration = 1hour and Health Status = normal
Then Recommendation 1

3.c Expert selects desired artifacts

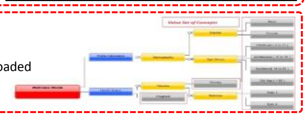


3.b Intelli-sense fetches the related concepts of value set

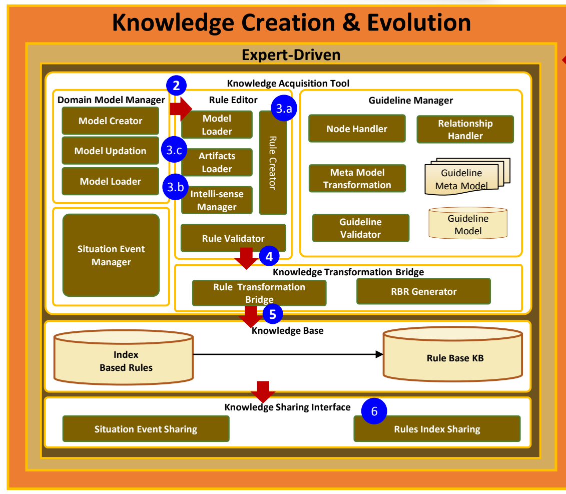
3.a Expert creating rules in the editor

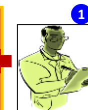


2 Existing Wellness Mode is Loaded



1 Expert opens rule editor



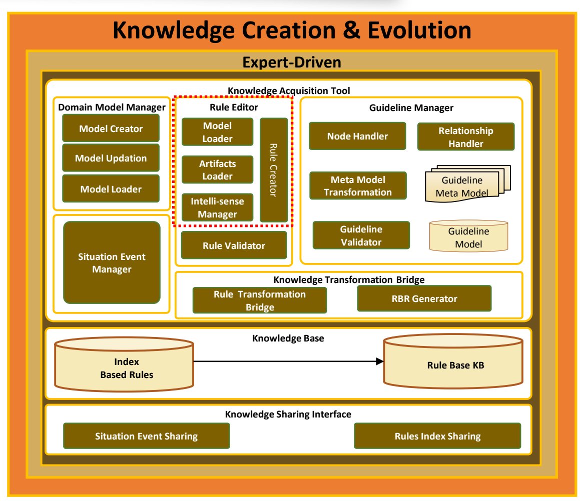


Methodology: Rule Editor

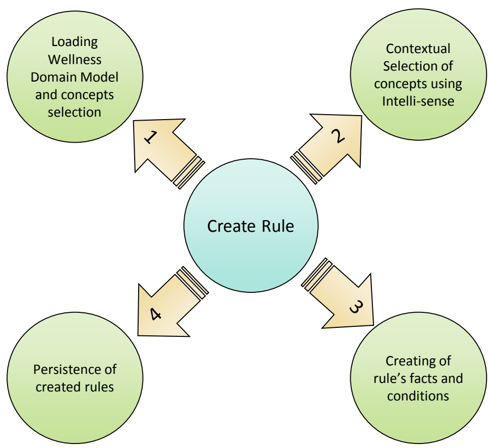
346

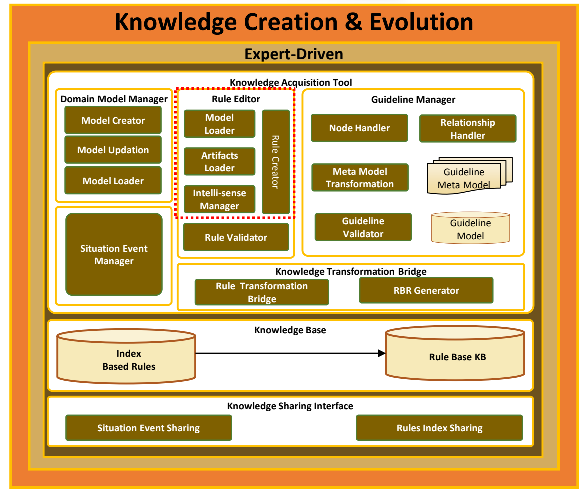
Use-case (KCL2.5-SUC-10): *Create Rule*

- **Objective**
  - ➢ Transform the expert's knowledge and guidelines into knowledge base rules
- **Methodology**
  - ➢ *Approach*: Model View Controller (MVC)
    - ➢ *Client side script: AngularJS*
    - ➢ *Server side script: Spring MVC*
  - ➢ ***Rule Editor*** *(operations)*
    1. **Input**: Concepts from wellness model/Intelli-sense, Expert's/Guideline knowledge
    2. **Processing**:
       a. Wellness domain model loads to the editor
       b. Domain experts create facts and conclusion of rules using Intelli-sense window and wellness model
       c. Domain experts select the required artifacts
       d. The created rule transform into plain rule format and store into knowledge base
    3. **Output**:
       a. Rule in plain rule format



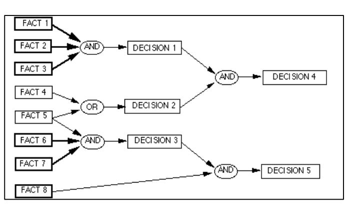
Use Case Functions: Rule Editor





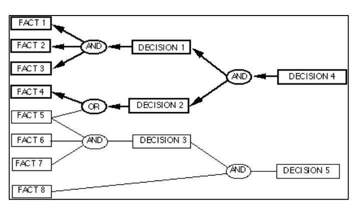
Rule Reasoning Strategies

Forward Chaining



Features
Works from facts to a conclusion (Data Driven)
Appropriate when all the facts are provided with the problem statement
Aims for finding conclusion
Appropriate for monitoring, planning, and interpretation applications
Breadth-first search

Backward Chaining

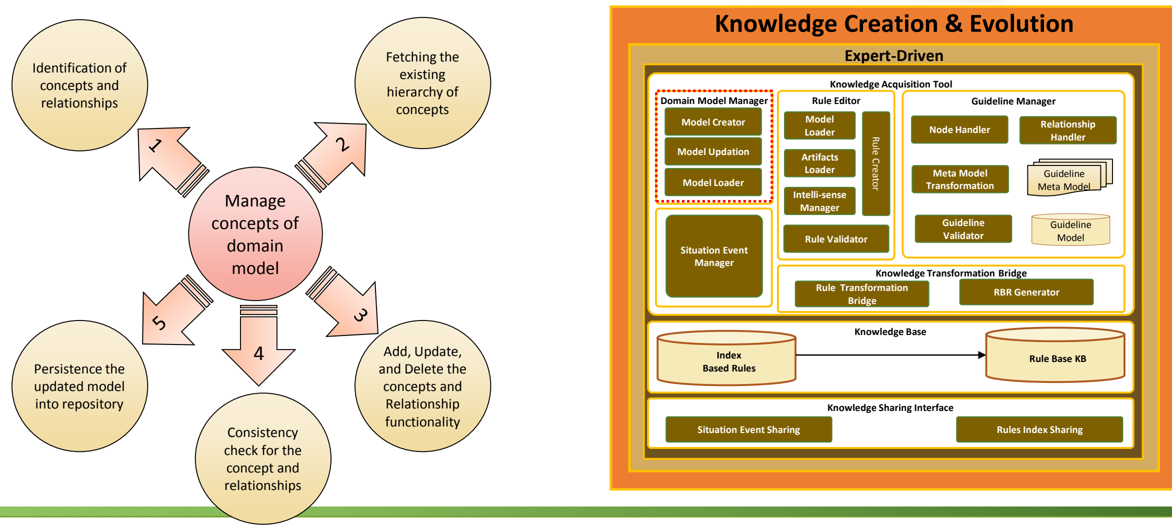


Features
Works from the conclusion to facts (Goal Driven)
Appropriate when the goal is given in the problem statement
Aims for finding necessary data
Appropriate for Diagnostic, prescription and debugging applications
Depth-first search

We will use Forward Chaining

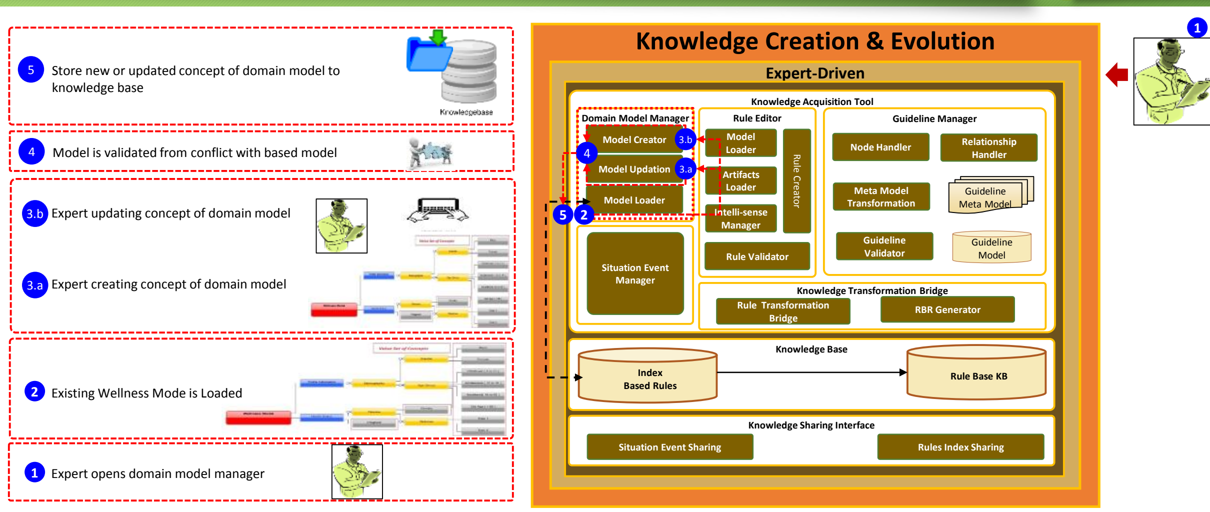
Use case functions: Domain Model Manager

349



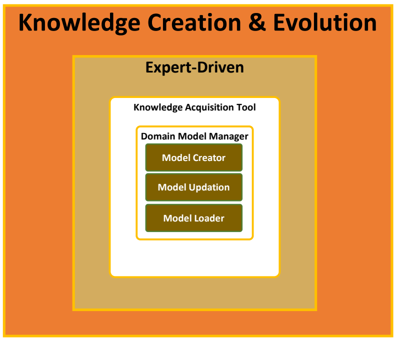
Domain Model Manager Scenario

350



Domain Model Manager – Internal Modules

351



Model Creator — IN: Expert Do(creation) / OUT: new Models and Relations

This module supply **Creation** of model by Expert View. The most of important thing is finding duplicated model and confliction with existing model.

Model Updation — IN: Expert Do(modification, Deletion) / OUT: updated Models and Relations

This module supply **Modification** and **Deletion** of model by Expert View.
Domain Expert can manage several concept model.

Model Loader — IN: System Do(load KB) / OUT: stored Models and Relations by formatted forms

This module supply **Loading** of concept model from Knowledge data base. It can be provided by **concept model class** or **json format markup language**.

Methodology: Domain Model Manager

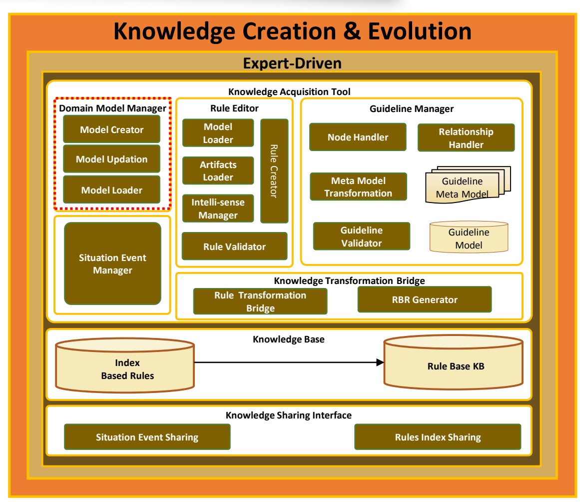
352

Use-case (KCL2.5-SUC-20): *Create concept*

Objective

- Provides creation flow of concepts of domain model by domain expert

- **Methodology**
  - *Approach*: Model View Controller (MVC)
    - *Client side script: AngularJS and Graphic tree generation*
    - *Server side script: Spring MVC + Restful Services*
  - ***Domain Model Manager** (operations)*
    1. **Input**: Concepts
    2. **Processing**:
       a. Domain expert identify the concepts to add
       b. Identify the relationship of new concept with other existing concepts if exists
       c. Add concepts and create relationships
       d. Store the updated model into repository
    3. **Output**:
       a. Updated wellness model



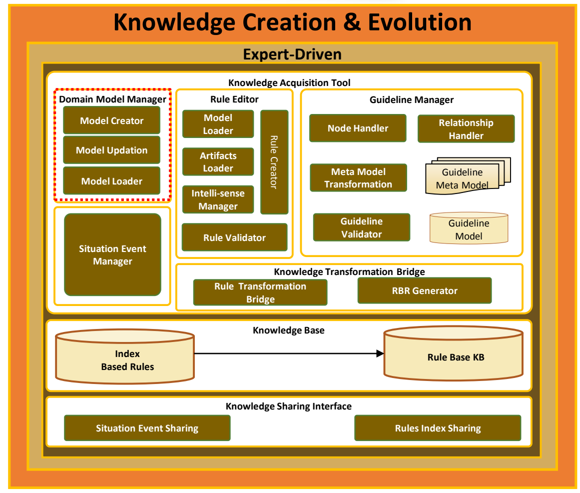
Methodology: Domain Model Manager

Use-case (KCL2.5-SUC-21): *Update concept*

Objective

- Provides updating flow of concepts of domain model about modification or deletion by domain expert

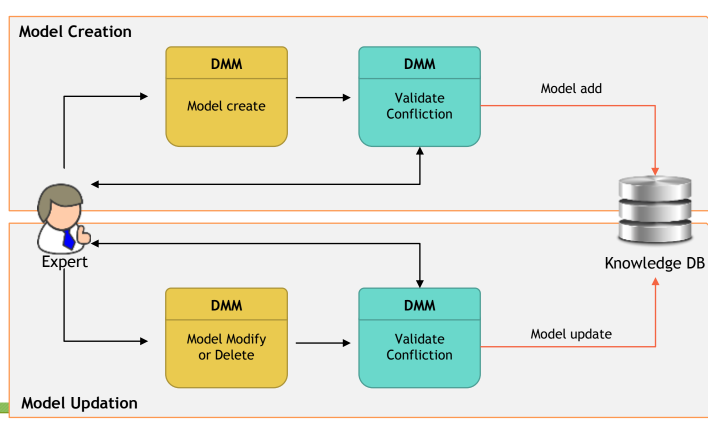
- **Methodology**
  - *Approach*: Model View Controller (MVC*)*
    - *Client side script: AngularJS and Graphic tree generation*
    - *Server side script: Spring MVC + Restful Services*
  - ***Domain Model Manager*** *(operations)*
    1. **Input**: Modification or deletion of concepts
    2. **Processing**:
       a. Domain expert updating concept to modify or delete
       b. Identify the relationship of updated concept with other existing concepts if exists
       c. Modify or delete concepts and update relationships
       d. Store the updated model into repository
    3. **Output**:
       a. Updated wellness model



Domain Model Manager – Data Work Flow

DMM Backend Framework | DMM Real View Framework

Domain Model Manager data flow is shown like this.



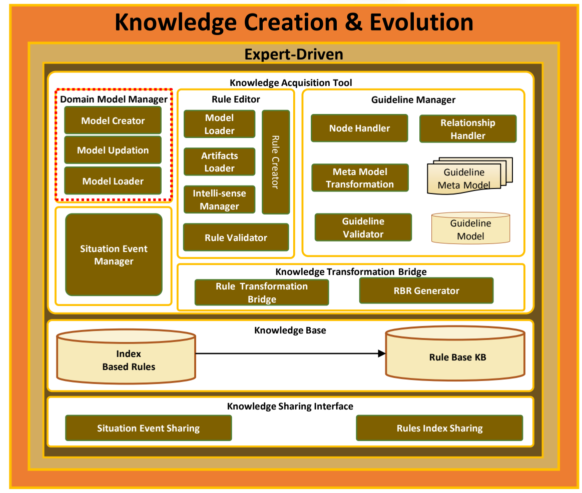
Methodology: Domain Model Manager

Use-case (KCL2.5-SUC-22): *Load concept*

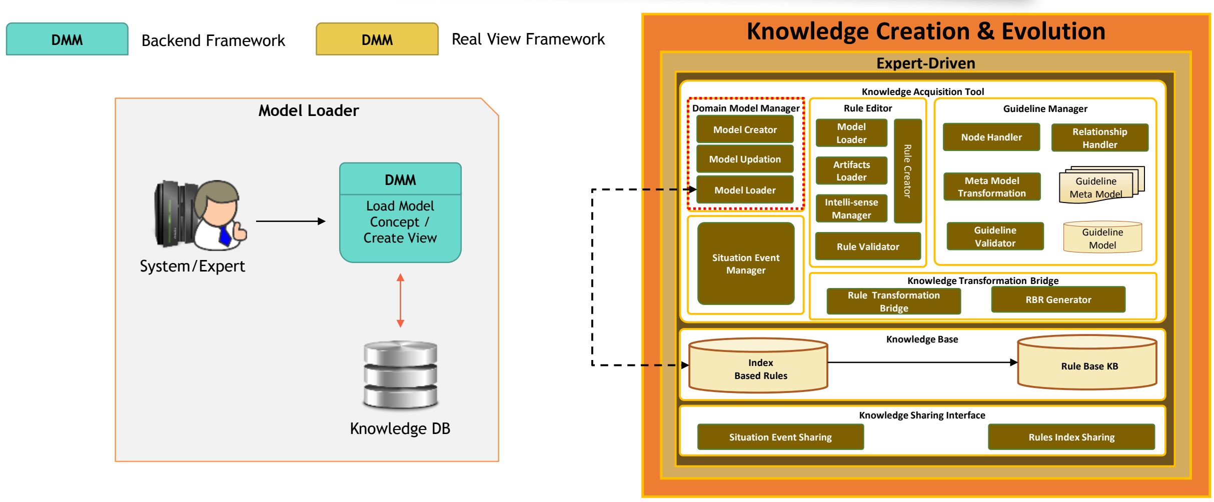
Objective

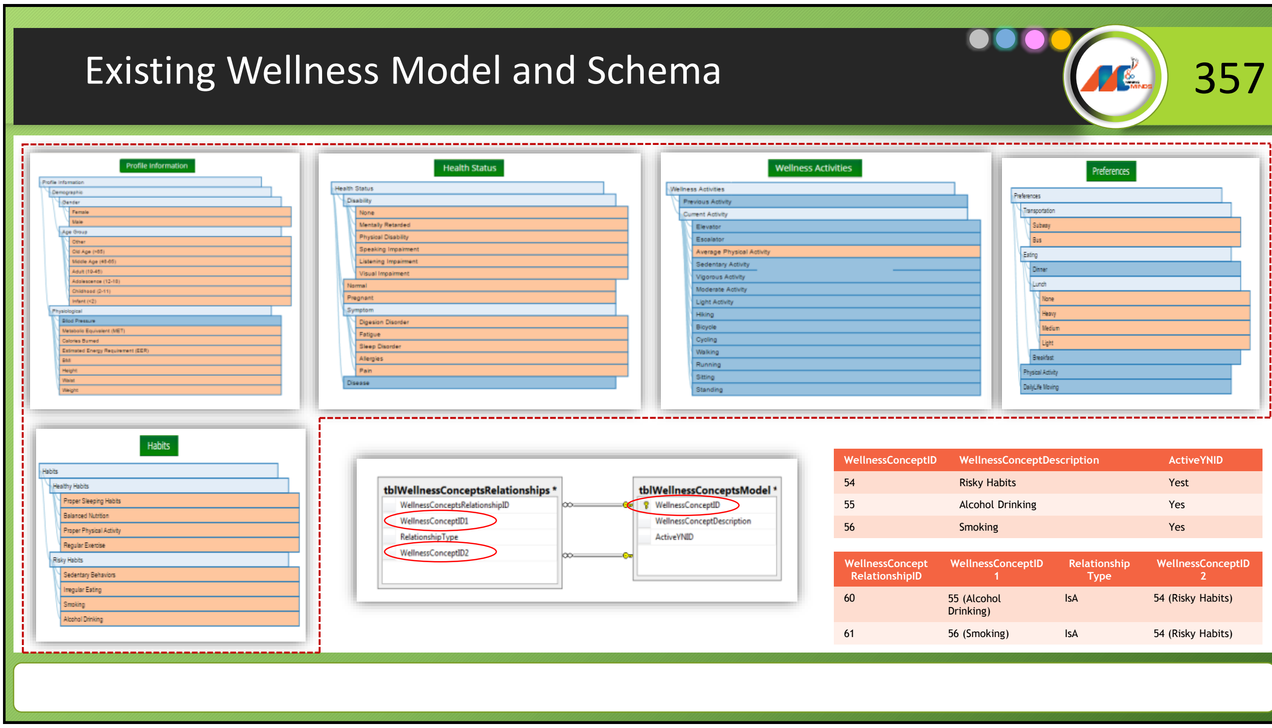
- ➢ Provides domain models from knowledge base to domain model manager.

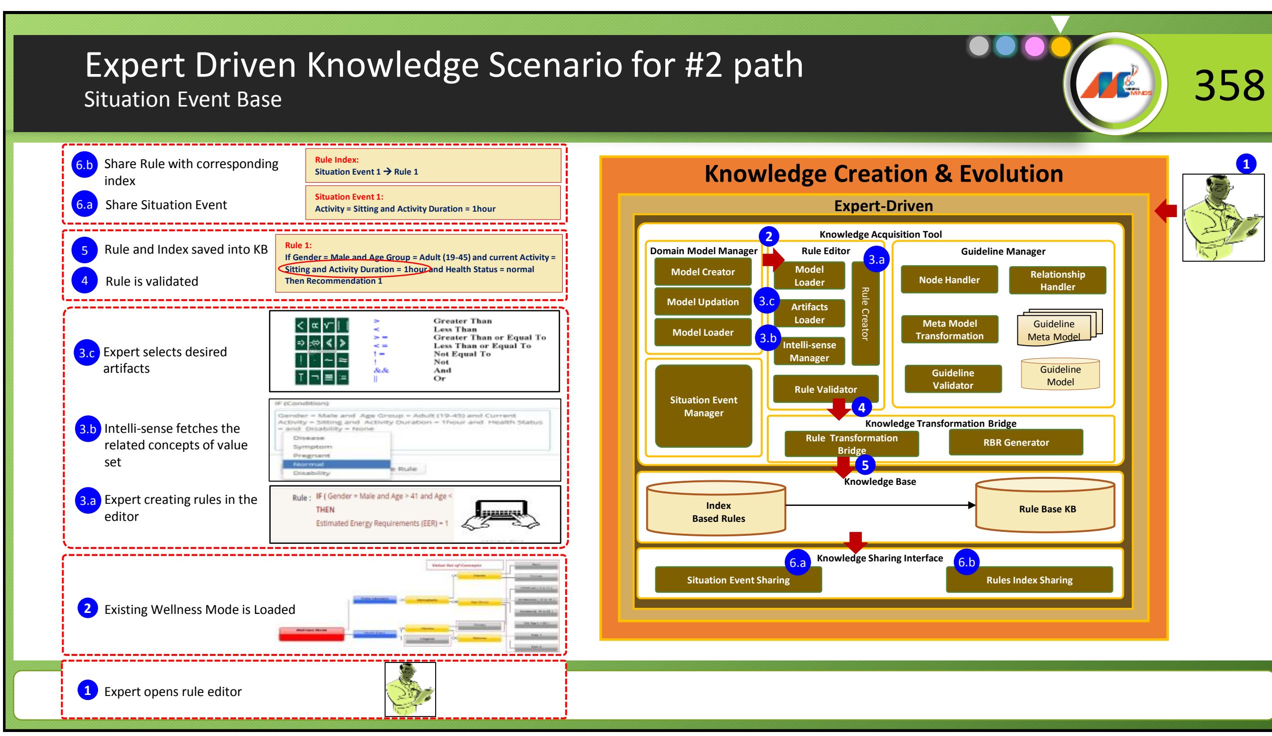
- **Methodology**
  - ➢ *Approach*: Model View Controller (MVC)
    - ➢ *Client side script: AngularJS*
    - ➢ *Server side script: Spring MVC + Hibernate + MS-SQL*
  - ➢ ***Domain Model Manager*** *(operations)*
    1. **Input**: Signal of domain model manager
    2. **Processing**:
       a. Domain expert uses domain model manager
       b. System loads domain models from knowledge base
    3. **Output**:
       a. Stored wellness model



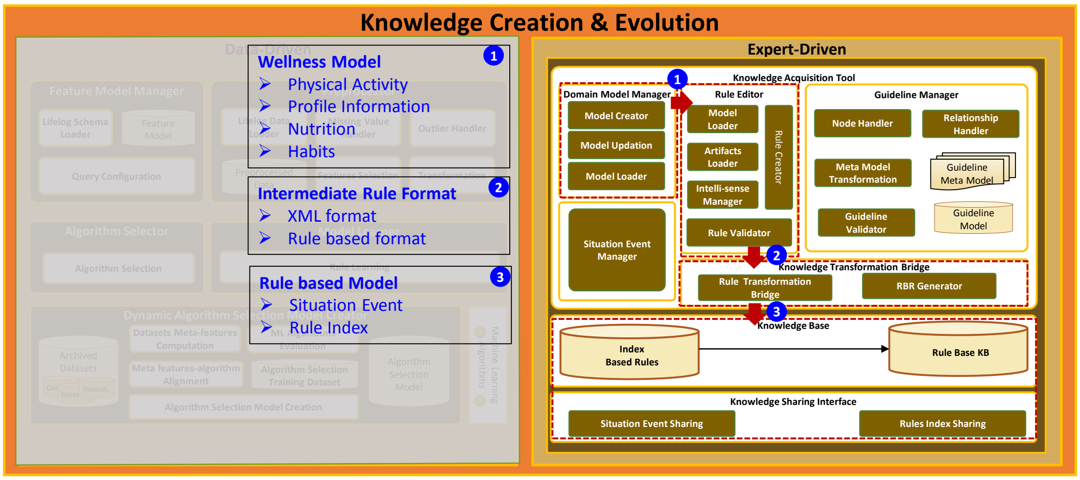
Domain Model Manager – Data Work Flow




Existing Wellness Model and Schema
357
Profile Information
Health Status
Wellness Activities
Preferences
Habits
tblWellnessConceptsRelationships *
WellnessConceptsRelationshipID
WellnessConceptID1
RelationshipType
WellnessConceptID2
tblWellnessConceptsModel *
WellnessConceptID
WellnessConceptDescription
ActiveYNID
WellnessConceptID | WellnessConceptDescription | ActiveYNID
54 | Risky Habits | Yest
55 | Alcohol Drinking | Yes
56 | Smoking | Yes
WellnessConcept RelationshipID | WellnessConceptID 1 | Relationship Type | WellnessConceptID 2
60 | 55 (Alcohol Drinking) | IsA | 54 (Risky Habits)
61 | 56 (Smoking) | IsA | 54 (Risky Habits)


Expert Driven Knowledge Scenario for #2 path
Situation Event Base
358
6.b Share Rule with corresponding index
6.a Share Situation Event
Rule Index:
Situation Event 1 → Rule 1
Situation Event 1:
Activity = Sitting and Activity Duration = 1hour
5 Rule and Index saved into KB
4 Rule is validated
Rule 1:
If Gender = Male and Age Group = Adult (19-45) and current Activity = Sitting and Activity Duration = 1hour and Health Status = normal
Then Recommendation 1
3.c Expert selects desired artifacts
Greater Than
Less Than
Greater Than or Equal To
Less Than or Equal To
Not Equal To
Not
And
Or
3.b Intelli-sense fetches the related concepts of value set
3.a Expert creating rules in the editor
2 Existing Wellness Mode is Loaded
1 Expert opens rule editor
Knowledge Creation & Evolution
Expert-Driven
Knowledge Acquisition Tool
Domain Model Manager
Model Creator
Model Updation
Model Loader
Rule Editor
Model Loader
Artifacts Loader
Intelli-sense Manager
Rule Creator
Rule Validator
Guideline Manager
Node Handler
Relationship Handler
Meta Model Transformation
Guideline Meta Model
Guideline Validator
Guideline Model
Situation Event Manager
Knowledge Transformation Bridge
Rule Transformation Bridge
RBR Generator
Knowledge Base
Index Based Rules
Rule Base KB
Knowledge Sharing Interface
Situation Event Sharing
Rules Index Sharing

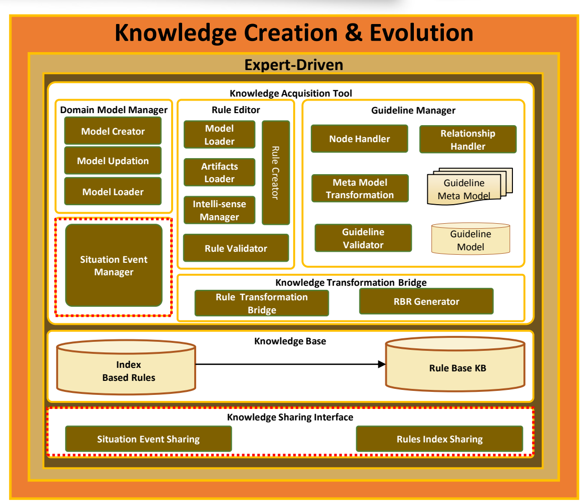
Expert Driven Rule Creation Workflow for #2 path



Methodology: Situation Event Manager

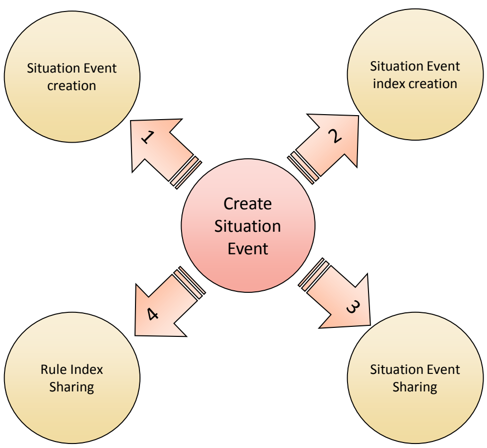
Use-case (KCL2.5-SUC-15): *Create Situation Event*

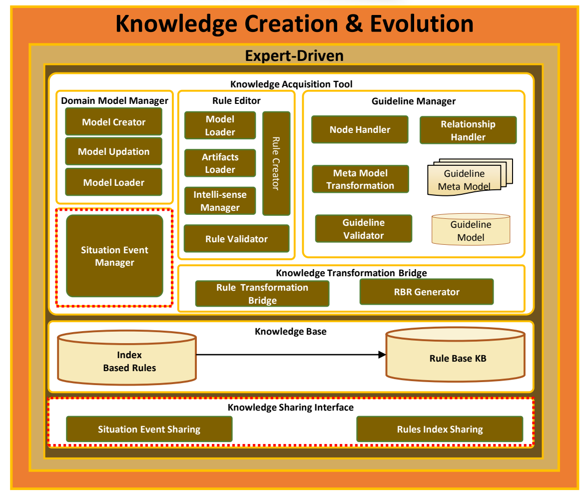
- **Objective**
  - To include set of associated recommendation rules and associate rules into index
- **Methodology**
  - *Approach*: Model View Controller (MVC)
    - *Server side script: Spring MVC*
  - ***Situation Event Manager*** *(operations)*
    1. **Input**: Plain rule in XML format
    2. **Processing**:
       a. Domain expert selects salient features (indicating as event) from conditions of the rule
       b. System creates situation event with selected salient features
       c. Assign index to the situation and associate to the rule
       d. Store the index with associated rule in knowledge base index
    3. **Output**:
       a. Indexed rules
       b. Situation Event: JSON



Use case functions: Situation Event Manager and Knowledge Sharing Interface

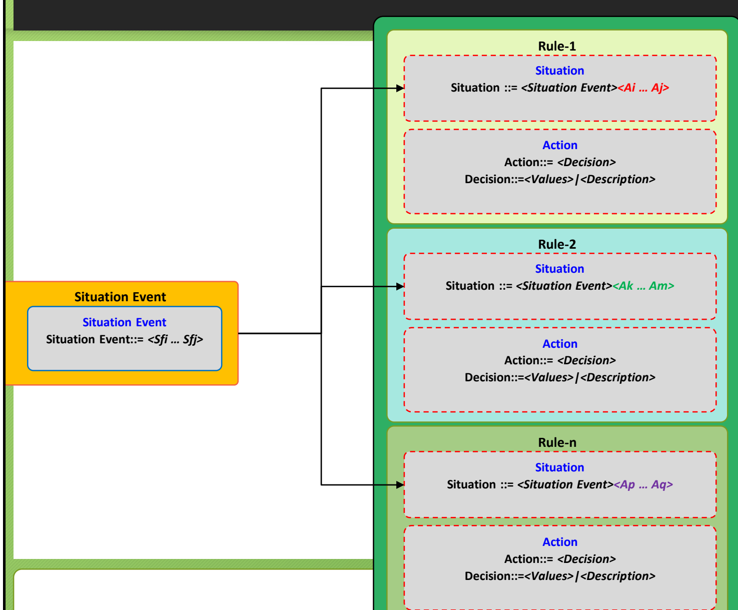
361



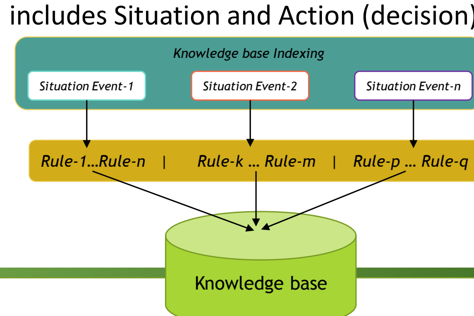


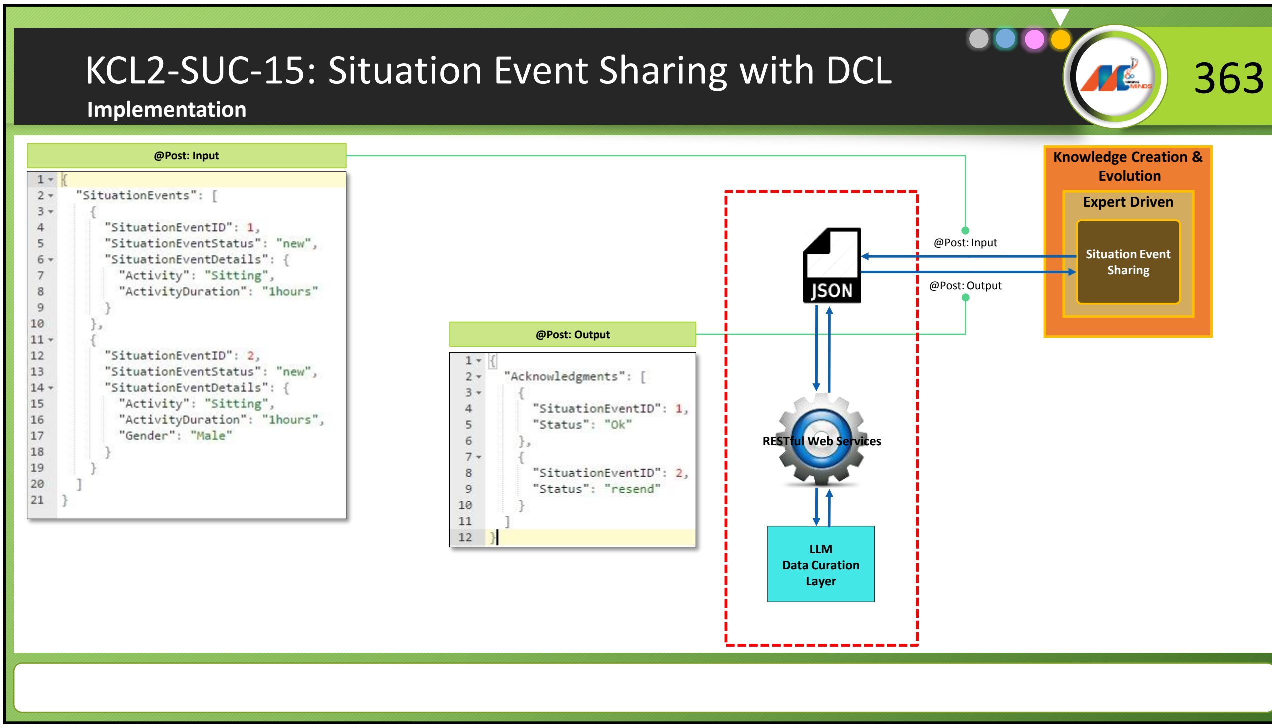
Situation, Situation Event and Rule

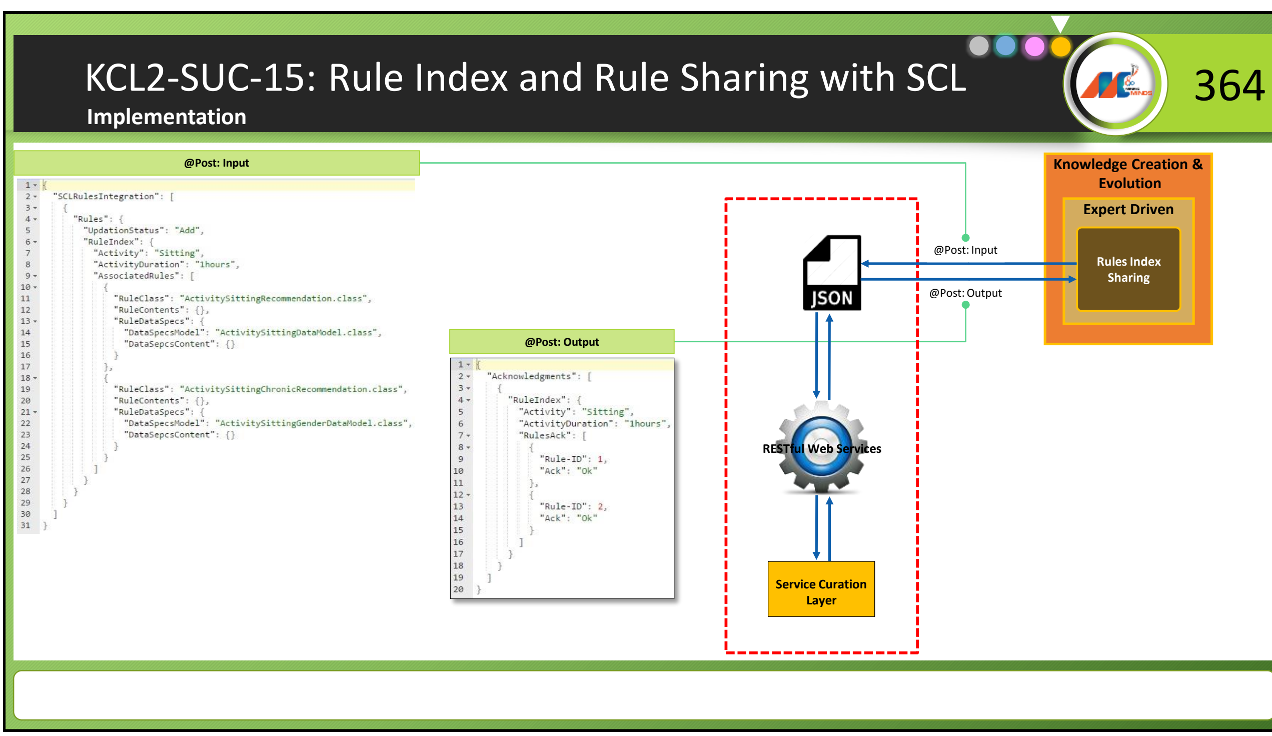
362



- *Situation* constitute part of rule which includes data elements(condition) required to process rule.
- *Situation Event* is part of *Situation* which includes *salient features* of data elements(condition).
- *Rule* is part of knowledge base which includes Situation and Action (decision)




KCL2-SUC-15: Situation Event Sharing with DCL
Implementation
363
@Post: Input
{
"SituationEvents": [
{
"SituationEventID": 1,
"SituationEventStatus": "new",
"SituationEventDetails": {
"Activity": "Sitting",
"ActivityDuration": "1hours"
}
},
{
"SituationEventID": 2,
"SituationEventStatus": "new",
"SituationEventDetails": {
"Activity": "Sitting",
"ActivityDuration": "1hours",
"Gender": "Male"
}
}
]
}
@Post: Output
{
"Acknowledgments": [
{
"SituationEventID": 1,
"Status": "Ok"
},
{
"SituationEventID": 2,
"Status": "resend"
}
]
}
JSON
RESTful Web Services
LLM
Data Curation
Layer
@Post: Input
@Post: Output
Knowledge Creation & Evolution
Expert Driven
Situation Event Sharing

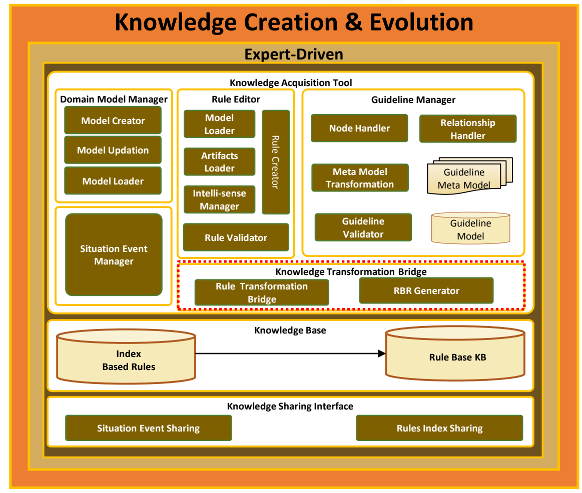

KCL2-SUC-15: Rule Index and Rule Sharing with SCL
Implementation
364
@Post: Input
{
"SCLRulesIntegration": [
{
"Rules": {
"UpdationStatus": "Add",
"RuleIndex": {
"Activity": "Sitting",
"ActivityDuration": "1hours",
"AssociatedRules": [
{
"RuleClass": "ActivitySittingRecommendation.class",
"RuleContents": {},
"RuleDataSpecs": {
"DataSpecsModel": "ActivitySittingDataModel.class",
"DataSepcsContent": {}
}
},
{
"RuleClass": "ActivitySittingChronicRecommendation.class",
"RuleContents": {},
"RuleDataSpecs": {
"DataSpecsModel": "ActivitySittingGenderDataModel.class",
"DataSepcsContent": {}
}
}
]
}
}
}
]
}
@Post: Output
{
"Acknowledgments": [
{
"RuleIndex": {
"Activity": "Sitting",
"ActivityDuration": "1hours",
"RulesAck": [
{
"Rule-ID": 1,
"Ack": "Ok"
},
{
"Rule-ID": 2,
"Ack": "Ok"
}
]
}
}
]
}
JSON
RESTful Web Services
Service Curation
Layer
@Post: Input
@Post: Output
Knowledge Creation & Evolution
Expert Driven
Rules Index Sharing

Methodology: Knowledge Transformation Bridge

365

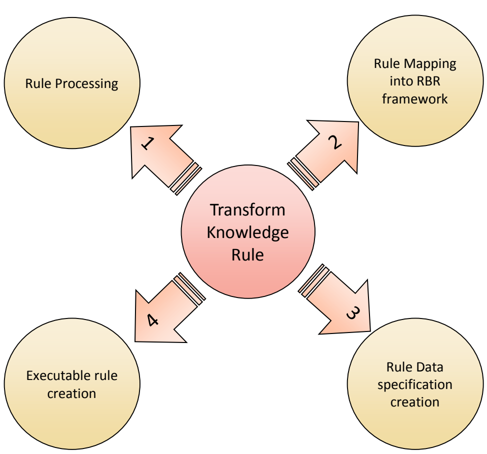
Use-case (KCL2.5-SUC-15): *Transform knowledge Rule*

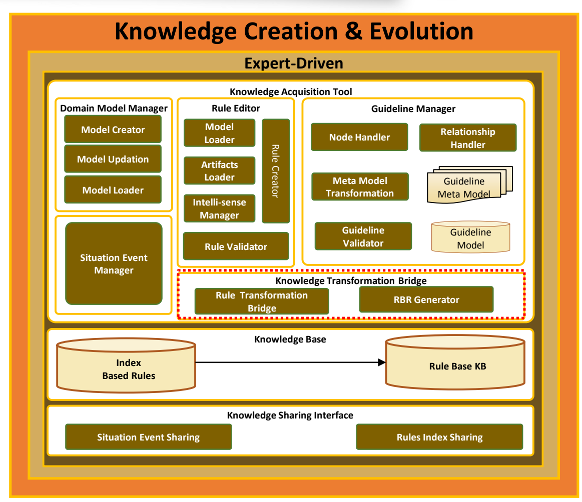
- **Objective**
  - ➢ Transform the created rules and guidelines into executable format to generate recommendations
- **Methodology**
  - ➢ *Approach*: Template based code generations
    - ➢ *Server side script: Core Java*
  - ➢ ***Knowledge Transformation Bridge*** *(operations)*
    1. **Input**: Plain rule in XML format
    2. **Processing**:
       a. The system identifies appropriate representation model
       b. Fetch the artifacts and controls of the selected representation model
       c. Transform the rule into selected representation model using its artifacts, controls and syntax
       d. Store the created/updated rule into repository
    3. **Output**:
       a. Executable rules into knowledge base

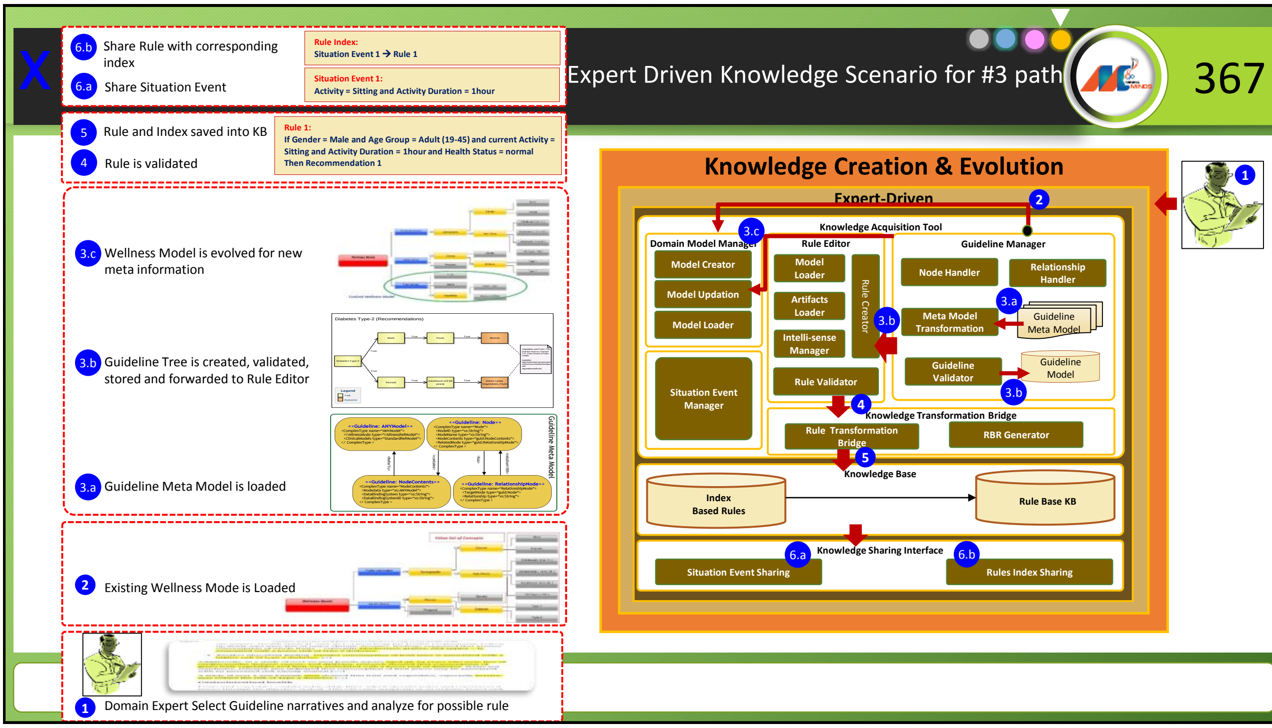


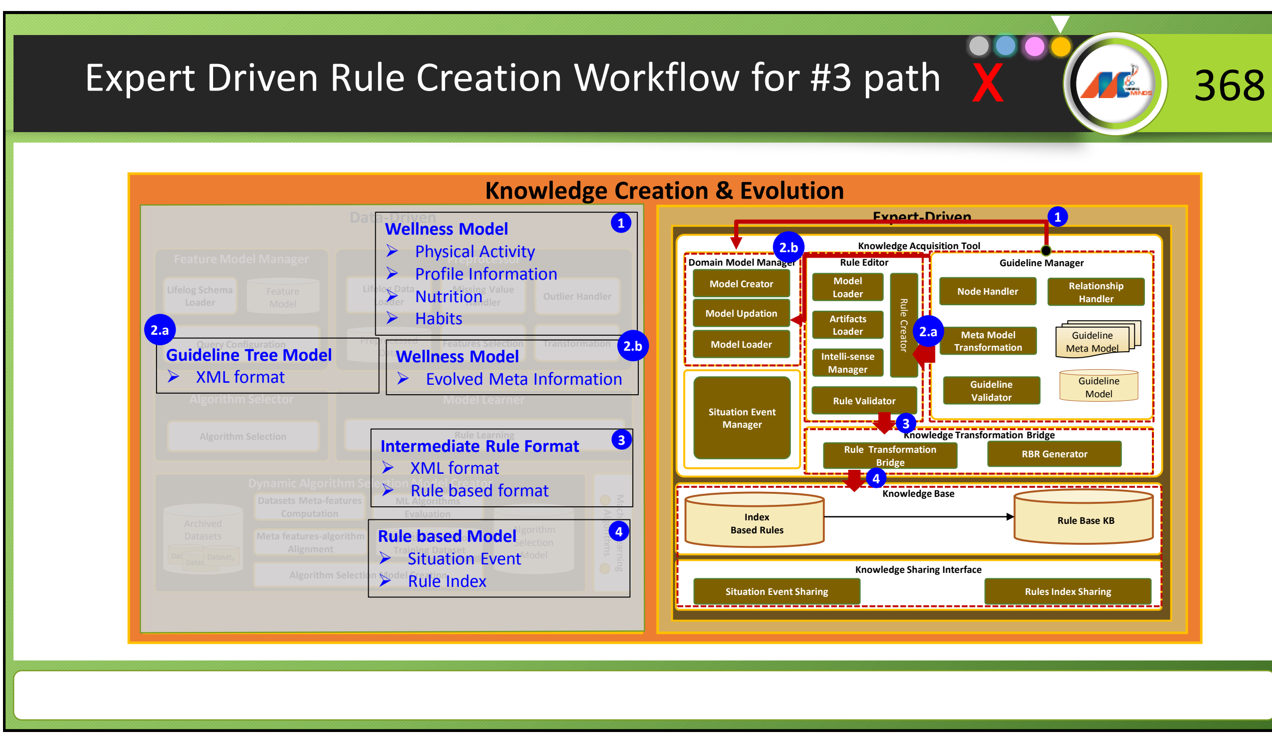
Use case functions: Knowledge Transformation Bridge

366






Expert Driven Knowledge Scenario for #3 path
367
X
6.b Share Rule with corresponding index
6.a Share Situation Event
Rule Index:
Situation Event 1 → Rule 1
Situation Event 1:
Activity = Sitting and Activity Duration = 1hour
5 Rule and Index saved into KB
4 Rule is validated
Rule 1:
If Gender = Male and Age Group = Adult (19-45) and current Activity = Sitting and Activity Duration = 1hour and Health Status = normal
Then Recommendation 1
3.c Wellness Model is evolved for new meta information
3.b Guideline Tree is created, validated, stored and forwarded to Rule Editor
Diabetes Type-2 (Recommendations)
3.a Guideline Meta Model is loaded
Guideline Meta Model
2 Existing Wellness Mode is Loaded
1 Domain Expert Select Guideline narratives and analyze for possible rule
Knowledge Creation & Evolution
Expert-Driven
Knowledge Acquisition Tool
Domain Model Manager
Model Creator
Model Updation
Model Loader
Situation Event Manager
Rule Editor
Model Loader
Artifacts Loader
Intelli-sense Manager
Rule Validator
Rule Creator
Guideline Manager
Node Handler
Relationship Handler
Meta Model Transformation
Guideline Meta Model
Guideline Validator
Guideline Model
Knowledge Transformation Bridge
Rule Transformation Bridge
RBR Generator
Knowledge Base
Index Based Rules
Rule Base KB
Knowledge Sharing Interface
Situation Event Sharing
Rules Index Sharing

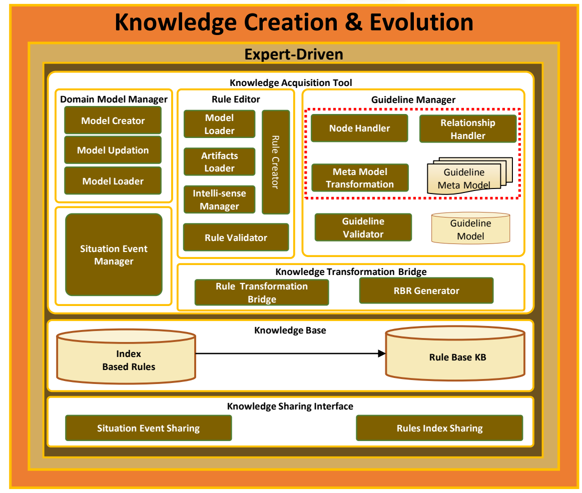

Expert Driven Rule Creation Workflow for #3 path
X
368
Knowledge Creation & Evolution
1 Wellness Model
Physical Activity
Profile Information
Nutrition
Habits
2.a Guideline Tree Model
XML format
2.b Wellness Model
Evolved Meta Information
3 Intermediate Rule Format
XML format
Rule based format
4 Rule based Model
Situation Event
Rule Index
Expert-Driven
Knowledge Acquisition Tool
Domain Model Manager
Model Creator
Model Updation
Model Loader
Situation Event Manager
Rule Editor
Model Loader
Artifacts Loader
Intelli-sense Manager
Rule Validator
Rule Creator
Guideline Manager
Node Handler
Relationship Handler
Meta Model Transformation
Guideline Meta Model
Guideline Validator
Guideline Model
Knowledge Transformation Bridge
Rule Transformation Bridge
RBR Generator
Knowledge Base
Index Based Rules
Rule Base KB
Knowledge Sharing Interface
Situation Event Sharing
Rules Index Sharing

Methodology: Guideline Manager

369

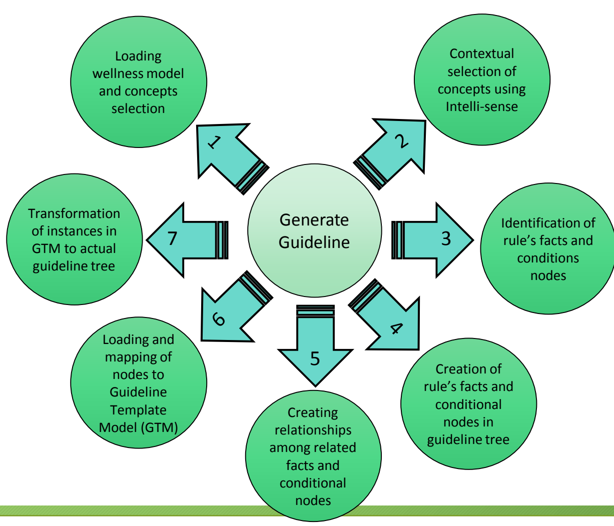
Use-case (KCL2.5-SUC-03): *Generate Guideline*

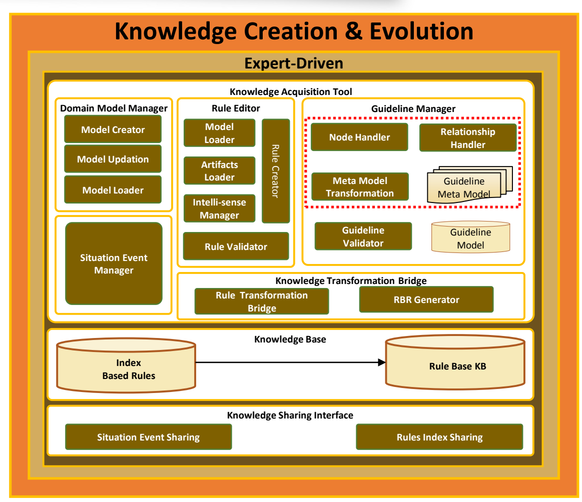
- **Objective**
  - ➢ Generate domain guidelines in tree format with easy manner
- **Methodology**
  - ➢ *Approach*: Model View Controller (MVC*)*
    - ➢ *Client side script: AngularJS*
    - ➢ *Server side script: Spring MVC*
  - ➢ ***Guideline Manager*** *(operations)*
    1. **Input**: Concepts from wellness model, Expert's/Guideline knowledge
    2. **Processing**:
       a. Domain expert generate tree nodes using Intelli-sense window and wellness model
       b. Make relationships among the nodes according to expert's knowledge
       c. Load guideline meta model and map the created nodes and relationships
       d. Transform the mapped nodes and relationships into tree in form of guideline meta model
    3. **Output**:
       a. Guideline Tree in graphical view
       b. Guideline Tree in XML (Guideline Meta Model)

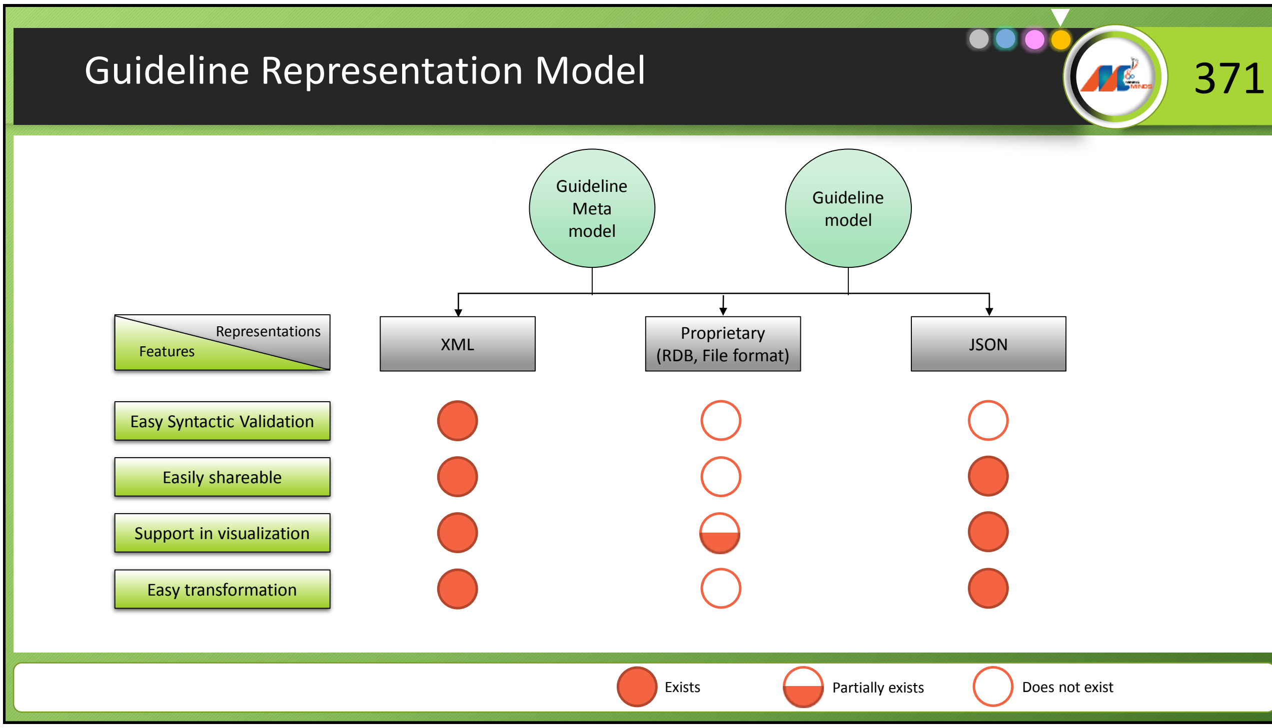


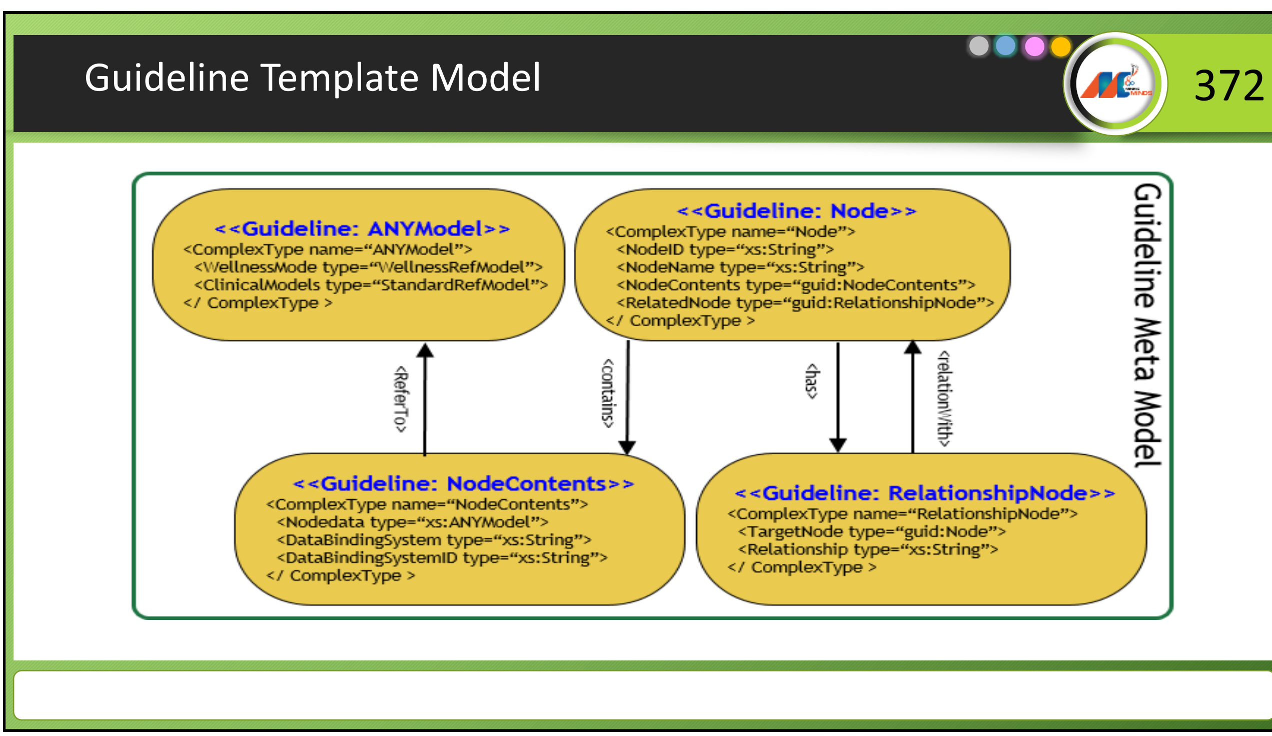
Use case functions: Guideline Manager

370



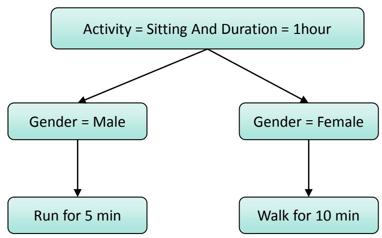


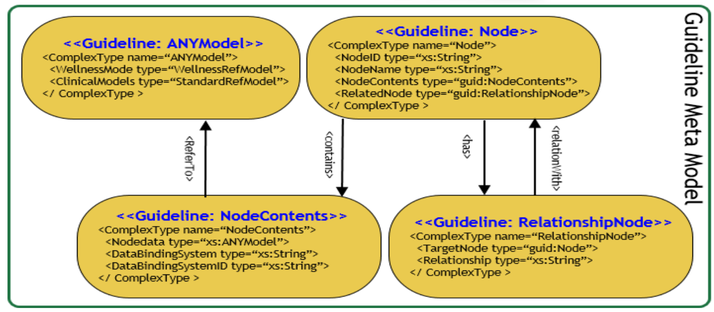

Guideline Representation Model
371
Guideline Meta model
Guideline model
Representations
Features
XML
Proprietary (RDB, File format)
JSON
Easy Syntactic Validation
Easily shareable
Support in visualization
Easy transformation
Exists
Partially exists
Does not exist


Guideline Template Model
372
<<Guideline: ANYModel>>
<ComplexType name="ANYModel">
<WellnessMode type="WellnessRefModel">
<ClinicalModels type="StandardRefModel">
</ ComplexType >
<<Guideline: Node>>
<ComplexType name="Node">
<NodeID type="xs:String">
<NodeName type="xs:String">
<NodeContents type="guid:NodeContents">
<RelatedNode type="guid:RelationshipNode">
</ ComplexType >
<ReferTo>
<contains>
<has>
<relationWith>
<<Guideline: NodeContents>>
<ComplexType name="NodeContents">
<Nodedata type="xs:ANYModel">
<DataBindingSystem type="xs:String">
<DataBindingSystemID type="xs:String">
</ ComplexType >
<<Guideline: RelationshipNode>>
<ComplexType name="RelationshipNode">
<TargetNode type="guid:Node">
<Relationship type="xs:String">
</ ComplexType >
Guideline Meta Model

Guideline Template Model

373





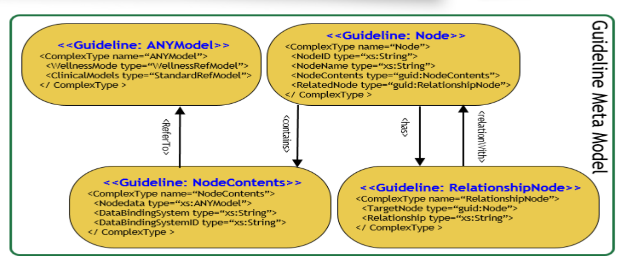
Conditional Nodes : Activity = Sitting And Duration = 1hour | Gender = Female | Gender = Male

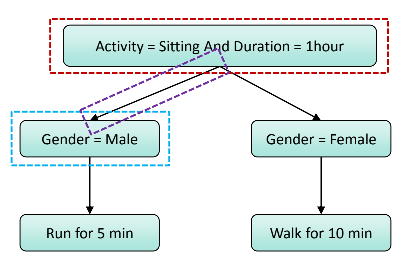
Decisional Nodes : Run for 5 min | Walk for 10 min

Guideline Template Model

374

```
<Guideline>
  <Node>
    <NodeID = Activity1 />
    <NodeName = rootNode />
    <NodeType = conditional />
    <NodeContents>
      <WellnessModel:Activity>
        <ActivityName> Sitting </ActivityName>
        <ActivityDuration> 1hour </ActivityDuration>
      </WellnessModel:Activity>
    </NodeContents>
    <RelatedNode>
      <TargetNode>Gender1</TargetNode>
      <Relationship>Child</Relationship>
    </RelatedNode>
  </Node>
  <Node>
    <NodeID = Gender1 />
    <NodeName = genderMale />
    <NodeType = conditional />
    <NodeContents>
      <WellnessModel:Gender>Male</WellnessModel:Gender>
</NodeContents>
    <RelatedNode>
      <TargetNode>Activity1</TargetNode>
      <Relationship>Parent</Relationship>
    </RelatedNode>
  </Node>
  ……..
</Guideline>
```





Contributions

375

- Provide user-friendly Rule Editor to domain experts
- Provide easy-to-use Guideline Editor to transform the guidelines from text to computer executable guidelines and rules
- Provide uniform Guideline Template Model (GTM)
- Provide wellness model manager to create wellness model by experts
- Index based rules (Situation enabled) generation and sharing for different services

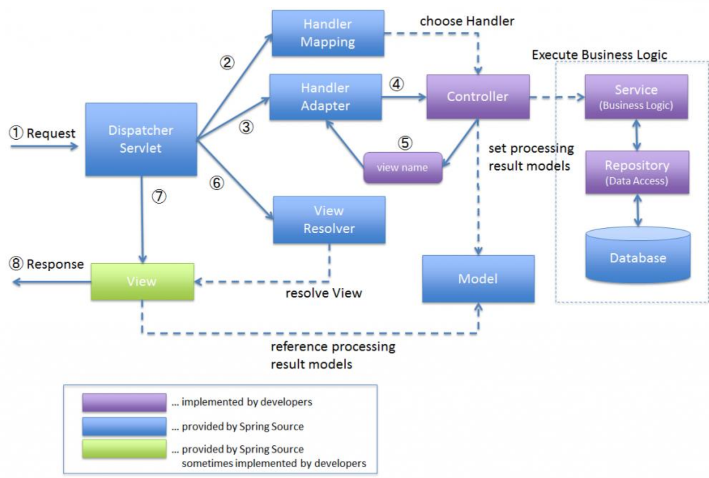
Selected Tools and Techniques

376

- Language
  - Java
- Frameworks:
  - Spring MVC
  - AngularJS
  - Hibernate
- Database:
  - MS SQL Server
- IDEs:
  - Eclipse

Request-response flow including services with Spring MVC

Spring MVC Request Lifecycle Representing Backend Services



경희대학교 KYUNG HEE UNIVERSITY UTAS

http://terasolunaorg.github.io/

Service Curation Layer (SCL 2.5)

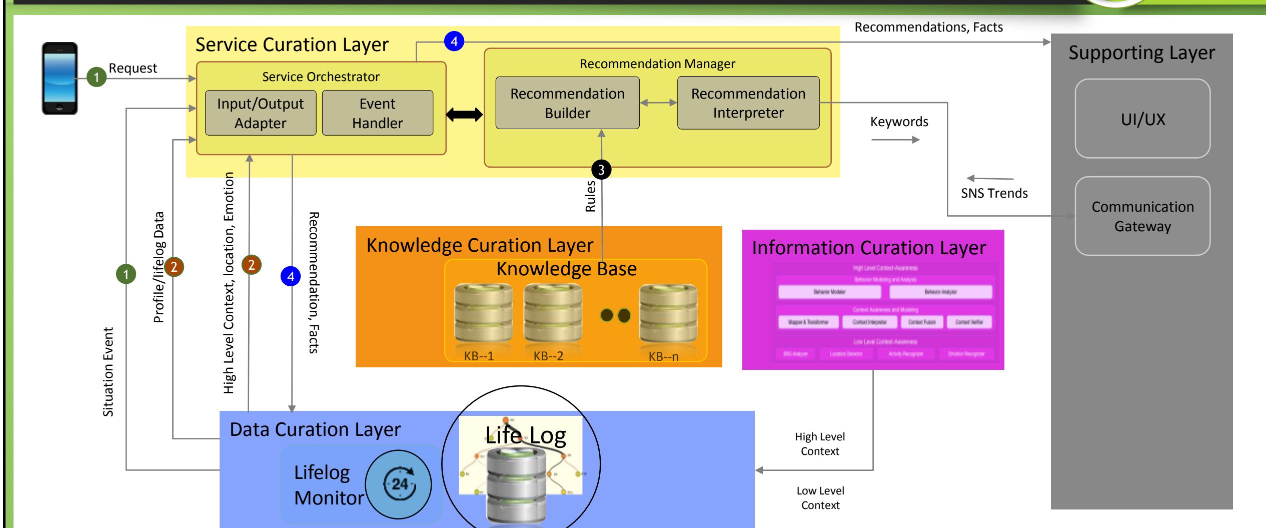
경희대학교 KYUNG HEE UNIVERSITY KITECH 한국생산기술연구원 POSTECH 포항공과대학교 UTAS 녹십자헬스케어 TAPACROSS U2 system User & Usability system

Goal and Objectives

- Goal
  - Providing precise and personalized wellbeing recommendations.

- Objectives
  - Enabling efficient communication within SCL components and other layers of the MM platform
  - Generating the accurate recommendations using users profile and lifelog information.
  - Interpreting the recommendations based on user context to ensure the deliver at right time, the right contents with right explanations.

SCL 2.5 Functional Diagram

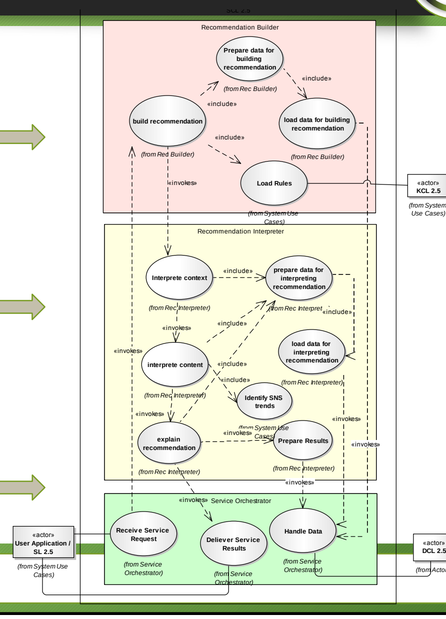


SCL Components and Req Specifications

Recommendation Builder

Recommendation Interpreter

Service Orchestrator



Rahman Ali

Imran Ali

Muhammad Afzal

Service Curation Layer - Architecture

Service Curation Layer

Service Orchestrator
- Input/output Adapter
- Event Handler

Recommendation Manager

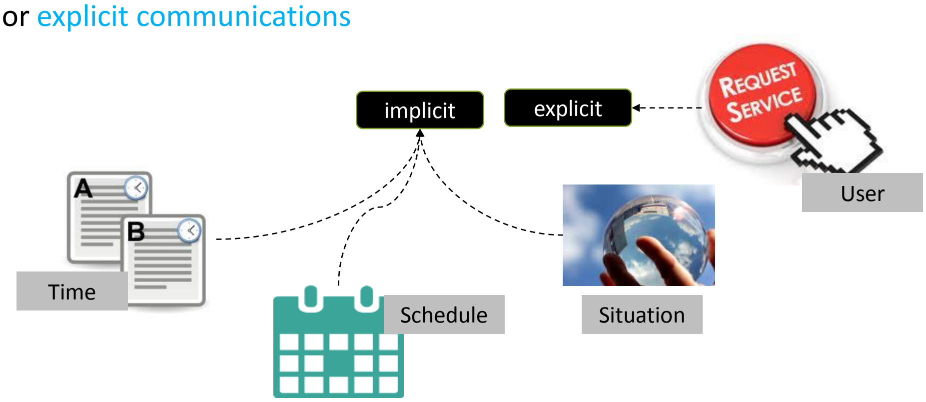
Recommendation Builder
- Lifelog Data loading Interface
  - Data Fetcher
  - Data Transformer
  - Utility Library
- Knowledge Loading Interface
  - Rules Loader
- Rule-based Reasoning
  - Pattern Matcher
  - Conflict Resolver
  - Result Generator

Recommendation Interpreter
- Context Interpreter
  - Context Selector
  - Context Interpreter
- Content Interpreter
  - Content Filterer
  - SNS Trend identifier
- Data Manager
  - Data Fetcher
  - Data Preparation
- Explanation Manager
  - Explanation Generator
  - Education Support

Service Orchestrator

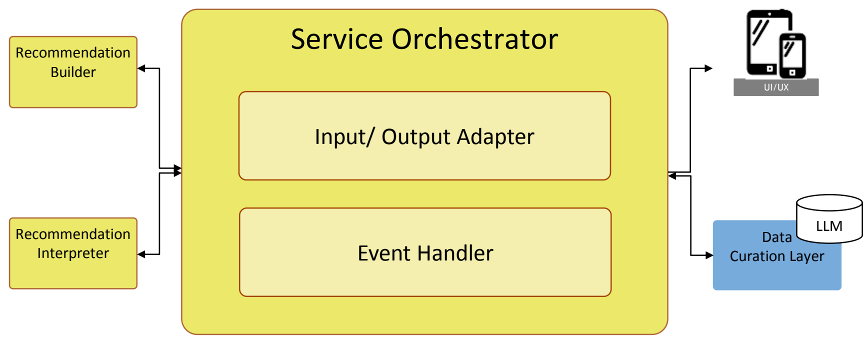
Introduction

- Orchestrator enable communication of SCL with other layers and user application
- It allows implicit or explicit communications



Service Orchestrator - Architecture

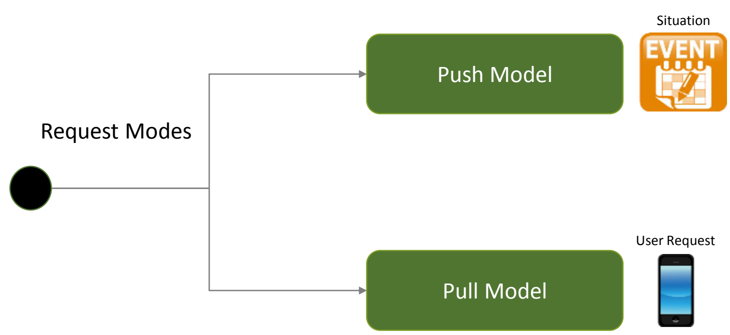
385

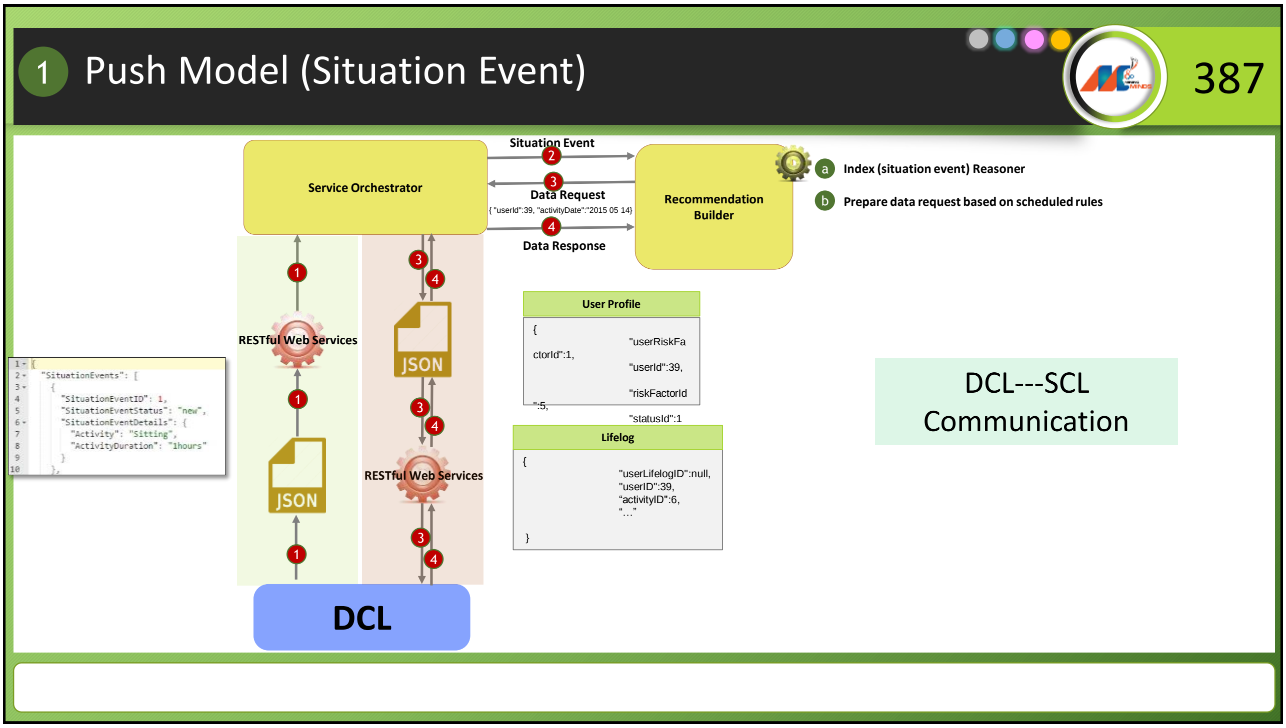


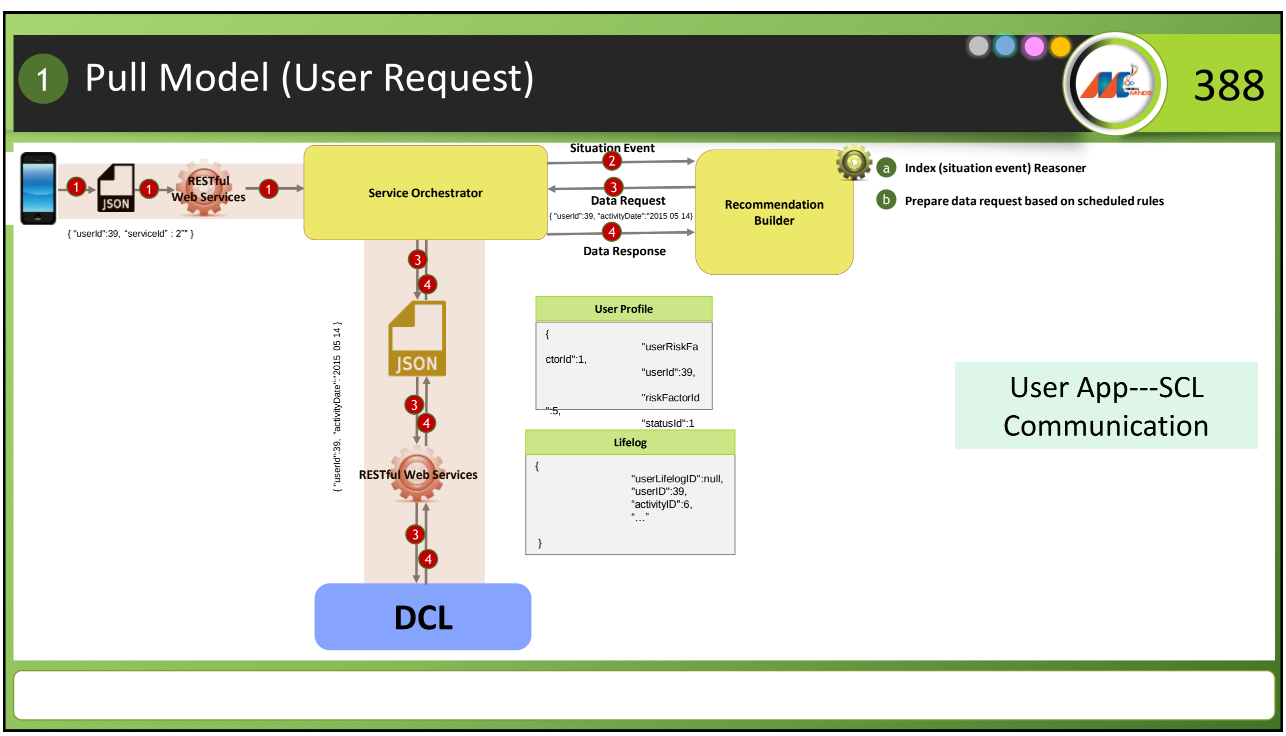
경희대학교
KYUNG HEE UNIVERSITY

Request for Recommendation

386




1
Push Model (Situation Event)
387
Situation Event
2
Service Orchestrator
3
Data Request
{ "userId":39, "activityDate":"2015 05 14}
4
Data Response
Recommendation Builder
a Index (situation event) Reasoner
b Prepare data request based on scheduled rules
RESTful Web Services
JSON
RESTful Web Services
User Profile
Lifelog
DCL---SCL Communication
DCL


1
Pull Model (User Request)
388
JSON
RESTful Web Services
{ "userId":39, "serviceId" : 2"" }
Service Orchestrator
Situation Event
Data Request
{ "userId":39, "activityDate":"2015 05 14}
Data Response
Recommendation Builder
a Index (situation event) Reasoner
b Prepare data request based on scheduled rules
{ "userId":39, "activityDate":"2015 05 14 }
JSON
RESTful Web Services
User Profile
Lifelog
User App---SCL Communication
DCL

Contributions

389

- Handling the situation based on dynamically generated events from DCL
- Handling user request explicitly made for the recommendation
- Handles the communication of data required for SCL services

경희대학교 KYUNG HEE UNIVERSITY

Recommendation Builder

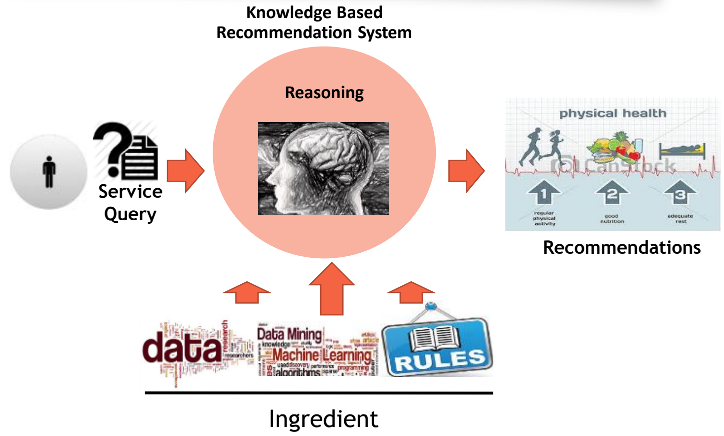
경희대학교 KYUNG HEE UNIVERSITY KITECH 한국생산기술연구원 POSTECH 포항공과대학교 UTAS 녹십자헬스케어 TAPACROSS U2 system User & Usability system

Introduction (1/3)

391

Reasoning (definition)

- An application whose computational function is to generate recommendations for user query using the available data and knowledge rules with the help of intelligent ML techniques [1].



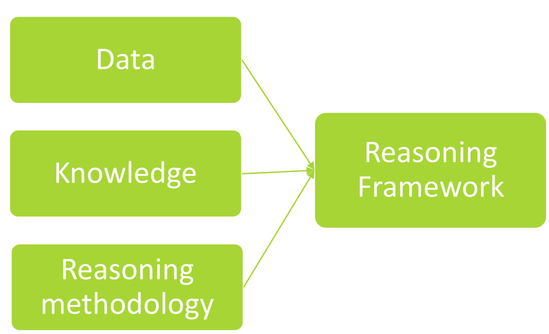
경희대학교 KYUNG HEE UNIVERSITY

[1] Moses, Yoram; Vardi, Moshe Y; Fagin, Ronald; Halpern, Joseph Y (2003). Reasoning About Knowledge. MIT Press. ISBN 978-0-262-56200-3.

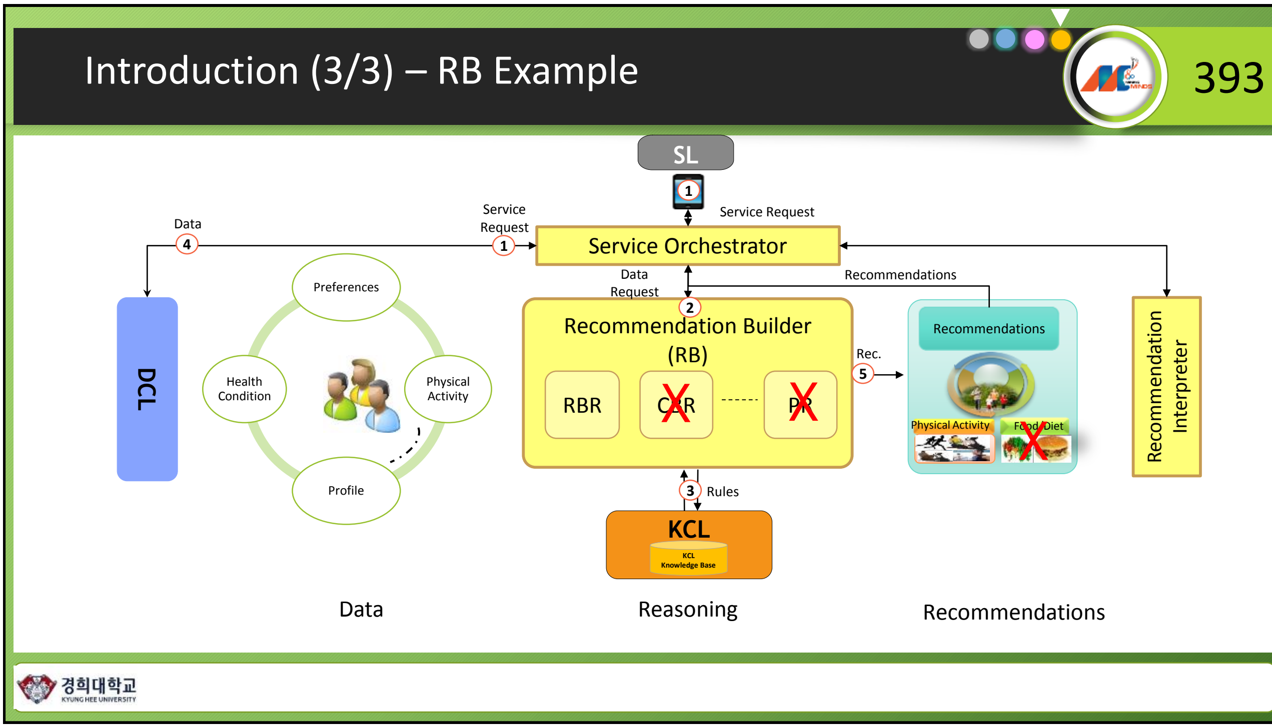
Introduction (2/3)

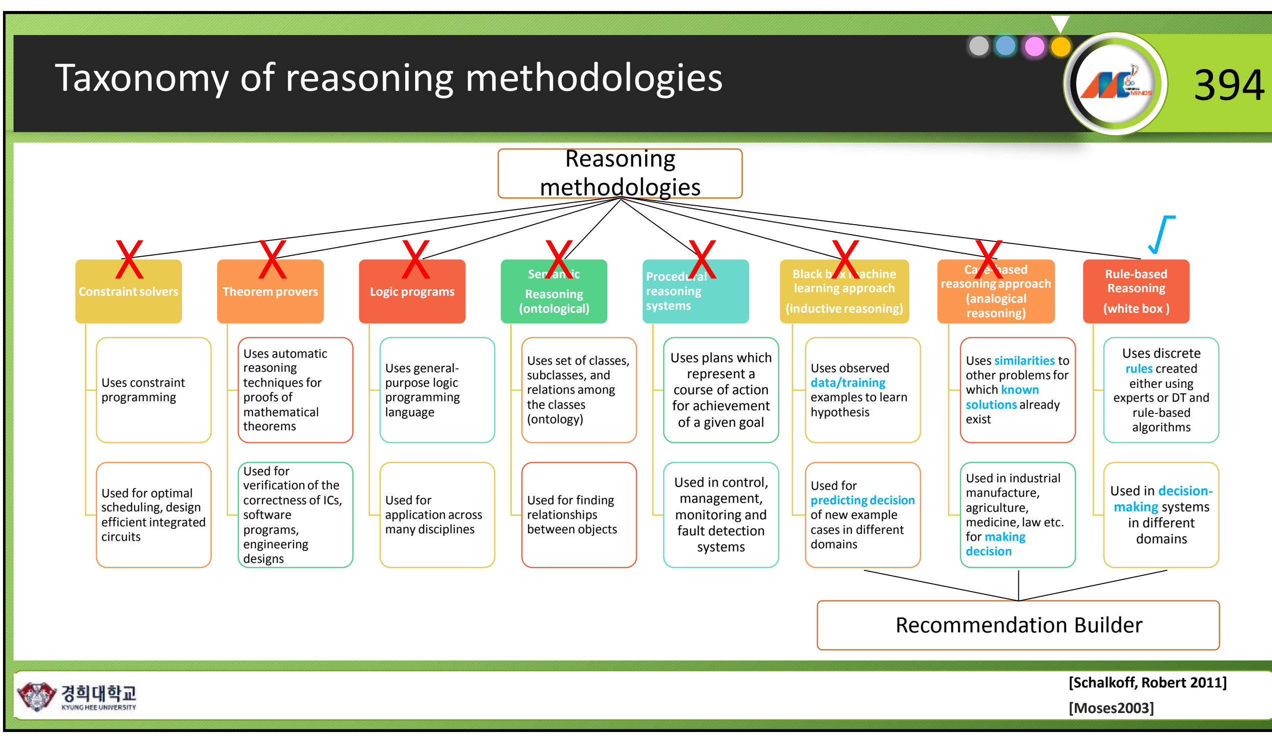
392

Ingredients of a Reasoning Framework



경희대학교 KYUNG HEE UNIVERSITY


Introduction (3/3) – RB Example
393
SL
1
Service Request
Service Request
1
Service Orchestrator
Data
4
DCL
Preferences
Health Condition
Physical Activity
Profile
Data Request
2
Recommendations
Recommendation Builder (RB)
RBR
CBR
PR
Rec.
5
Recommendations
Physical Activity
Food Diet
Recommendation Interpreter
3
Rules
KCL
KCL Knowledge Base
Data
Reasoning
Recommendations
경희대학교
KYUNG HEE UNIVERSITY


Taxonomy of reasoning methodologies
394
Reasoning methodologies
Constraint solvers
Theorem provers
Logic programs
Semantic Reasoning (ontological)
Procedural reasoning systems
Black box machine learning approach (inductive reasoning)
Case-based reasoning approach (analogical reasoning)
Rule-based Reasoning (white box)
√
Uses constraint programming
Uses automatic reasoning techniques for proofs of mathematical theorems
Uses general-purpose logic programming language
Uses set of classes, subclasses, and relations among the classes (ontology)
Uses plans which represent a course of action for achievement of a given goal
Uses observed data/training examples to learn hypothesis
Uses similarities to other problems for which known solutions already exist
Uses discrete rules created either using experts or DT and rule-based algorithms
Used for optimal scheduling, design efficient integrated circuits
Used for verification of the correctness of ICs, software programs, engineering designs
Used for application across many disciplines
Used for finding relationships between objects
Used in control, management, monitoring and fault detection systems
Used for predicting decision of new example cases in different domains
Used in industrial manufacture, agriculture, medicine, law etc. for making decision
Used in decision-making systems in different domains
Recommendation Builder
[Schalkoff, Robert 2011]
[Moses2003]
경희대학교
KYUNG HEE UNIVERSITY

Related Work

395

Reasoning Type	Reference	Domain	Service	Key Features	
				Methodology	Technique
Machine learning approach (Black box approach)	(Zazzi2010)-Project (2006-08) -Study	Treatment - HIV	Predicts the efficiency of possible Anti-HIV drug combinations	• Use several prediction engines • combines their results	Use Classification technique
	(Drăgulescu2007) - Tool as a proof of research	Diagnosis - hepatitis	Predicts hepatitis infection	• Logical, Statistical and ANN are used to first detect hepatitis type • Then treat type B and C	Logical, Statistical and ANN
	[Pandey2012] - Tool as a proof of research	Diagnosis -diabetes	Predicts diabetes types	• Uses Apriori algorithm • Finds type of diabetes based on the symptoms	Apriori algorithm
Case-based reasoning approach	[Abdus Salam Khan, 2003]	Wellness - food Recommendation	Automatically construct food menu based on individual preferences	• User personal profile and preferences are taken • Menus and their contents are retrieved using CBR • Domain expert to directly interact	CBR
	[Trewin2000] -prototype	Restaurants - recommendation	Recommends location-based restaurants with their offered services	• Asks for location • Asks for preferences, price, food etc. • Similarity-based algorithms are used	CBR
	[Evans-Romaine3003] -prototype	Educating medicine students for – recommending exercise regimes	Prescribing Exercise Regimens for Cardiac and Pulmonary Disease Patients with CBR	• Medicine students learn to prescribe exercise regimens for cardiac and pulmonary disease patients • CBR similarly is used to retrieved successful cases	CBR
Rule-based Reasoning	[Minutolo2010] -prototype	Cardiac patients -monitoring	Detection of of abnormal or emergency situations in cardiac patient using contextual information, i.e. data from a wearable (ECG) that records patient's posture and physical activities.	• Monitors physical activities • Monitors body postures • Detect abnormalities using RBR approach	RBR
	[Al-Dhuhli,2013]	Diet - recommendation	Generalized diet recommendations are generated	• Knowledge is extracted from domain experts • Uses e2go freeware rule-based shell. • Diet recommendations are generated	RBR
	(Husain2010) - Model [Lim2013]	Wellness therapy - Recommendation	Predicts suitable personalized wellness therapy for individual	• Uses RBR for the complementary and alternative medicine	RBR
	(Yuan2014) -Framework	Wellness Therapy - Recommendation	Predicts therapy to the elderly people	• Uses sensor fusion in a smart home environment for data	RBR
	(Yuan Bingchuan , John Herbert, 2014)	Healthcare: at-home - Monitoring of the elderly activities	Provide s personalized healthcare services for elderly by using reasoning framework for CARA system	• personalized, flexible, and extensible hybrid	Fuzzy-RBR

Limitations of Existing Work

396

- Lack of reliable recommendations due to lack of knowledge rules
- Existing knowledge-based recommendation systems are highly-coupled and dependent on close integration of data, knowledge and reasoning methodology
- Reasoning process is based on well-prepared user data

경희대학교
KYUNG HEE UNIVERSITY

Motivations

397

Recommendation Builder (RM)

Reliable Recommendations
→ In wellbeing applications, users are interested in reliable service recommendations

Diverse Services Generation
→ Diverse services generation increase utilization scope of the system

Accurate Recommendations
→ Accurate recommendation from a set of candidate recommendations increase user satisfaction

경희대학교
KYUNG HEE UNIVERSITY

Motivations – Challenges - Solutions

398

Motivations	Challenges		Solutions
Reliable Recommendations	→ Integration of knowledge rules	S1	→ Knowledge Loading Interface (Indexed-based Knowledge Integration)
Diverse Services Generation	→ Design and development of flexible rule-based reasoning framework	S2	→ S1 + Data Loading Interface + RBR (Flexible RBR Framework)
Accurate Recommendations	→ Finding appropriate rules during execution and generation of recommendations	S3	→ Rule-based Reasoner (RBR with Forward Chaining & Conflict Resolution)

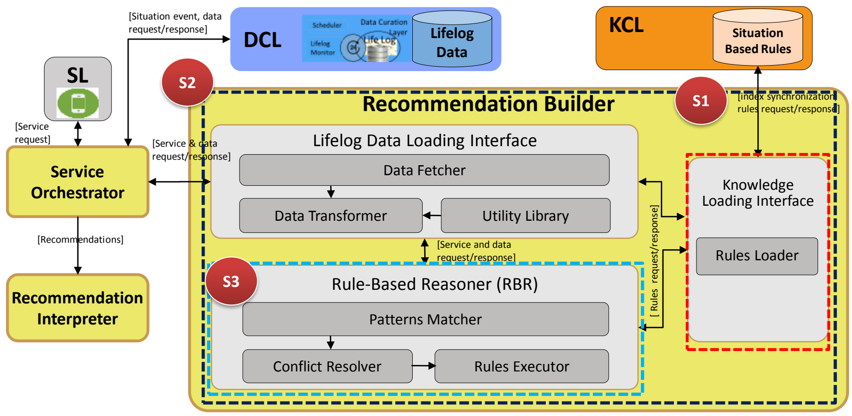
경희대학교
KYUNG HEE UNIVERSITY

Recommendation Builder - Architecture

S1:
Indexed-based Knowledge Integration

S2:
Flexible RBR Framework

S3:
RBR with Forward Chaining and Conflict Resolution



Recommendation Builder - Architecture

1 **Lifelog Data Loading Interface**

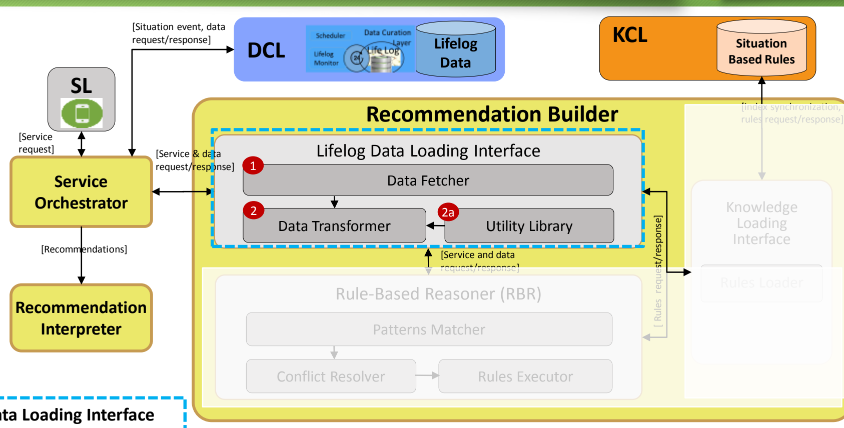
Data Fetcher

1. Receives requests from SL and DCL for recommendations
2. Parse request
3. Requests life-log data from DCL
4. Forwards recommendations to Recommendation Interpreter

2 **Lifelog Data Loading Interface**

Data Transformer

- Parse rules conditions part
- Conformance check of condition attributes and data
- Transformation of data to the condition formats



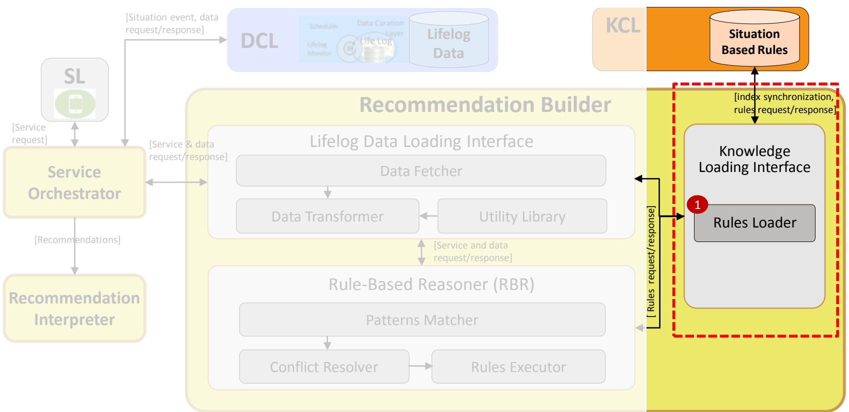
2a **Lifelog Data Loading Interface**

Utility Library

- Utilities functions are defined
- Computations for rules condition attributes are done

Recommendation Builder - Architecture

401



1 **Knowledge Loading Interface**

Rules Loader

- Make a request for knowledge
- Send request to KCK
- Receive rules from KCL
- Provides Rules to Reasoner

경희대학교 KYUNG HEE UNIVERSITY

Recommendation Builder - Architecture

402

1 **Rule-based Reasoner**

Pattern Matcher

1. Matches rules against refined data
2. Uses forward chaining mechanism
3. Retrieves set of matched rules

2 **Rule-based Reasoner**

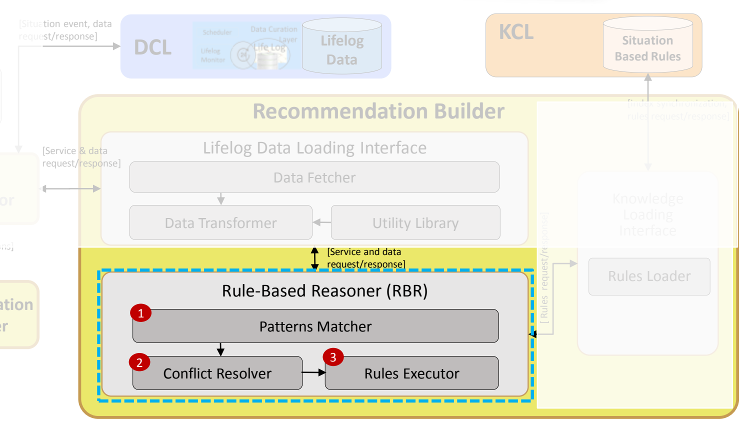
Conflict Resolver

- Filters the set of reterived rules
- Uses conflict resolution strategy
- Selects the final rules

3 **Rule-based Reasoner**

Rules Executer

- Binds data with condition of the rules
- Execute Rule
- Provides recommendation

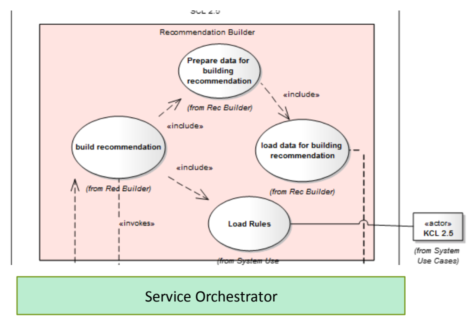


경희대학교 KYUNG HEE UNIVERSITY

Recommendation Builder – Use cases

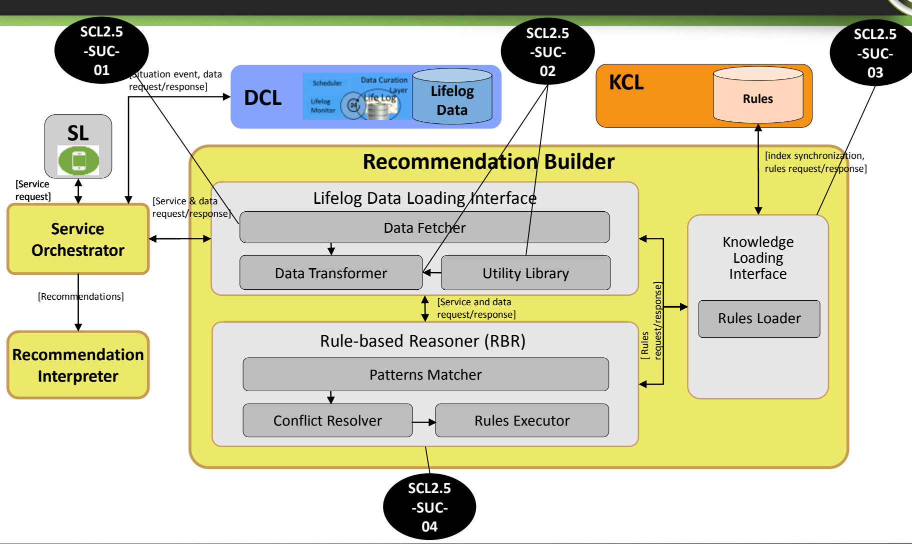
403

Use Case #ID	Description
SCL2.5-SUC-01	Load data for building recommendation
SCL2.5-SUC-02	Prepare data for building recommendation
SCL2.5-SUC-03	Load Rules
SCL2.5-SUC-04	Build Recommendation



Recommendation Builder - Architecture

404



Use-case description

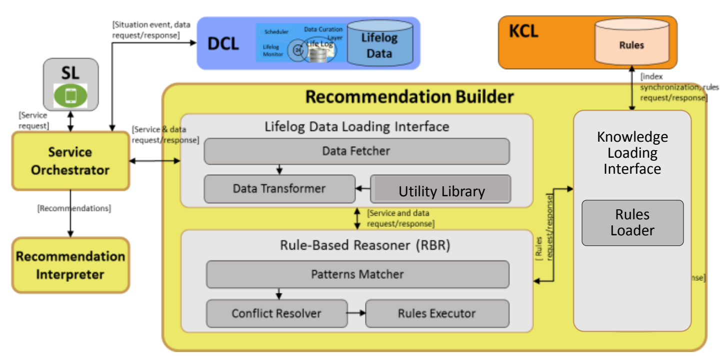
SCL2.5-SUC-01

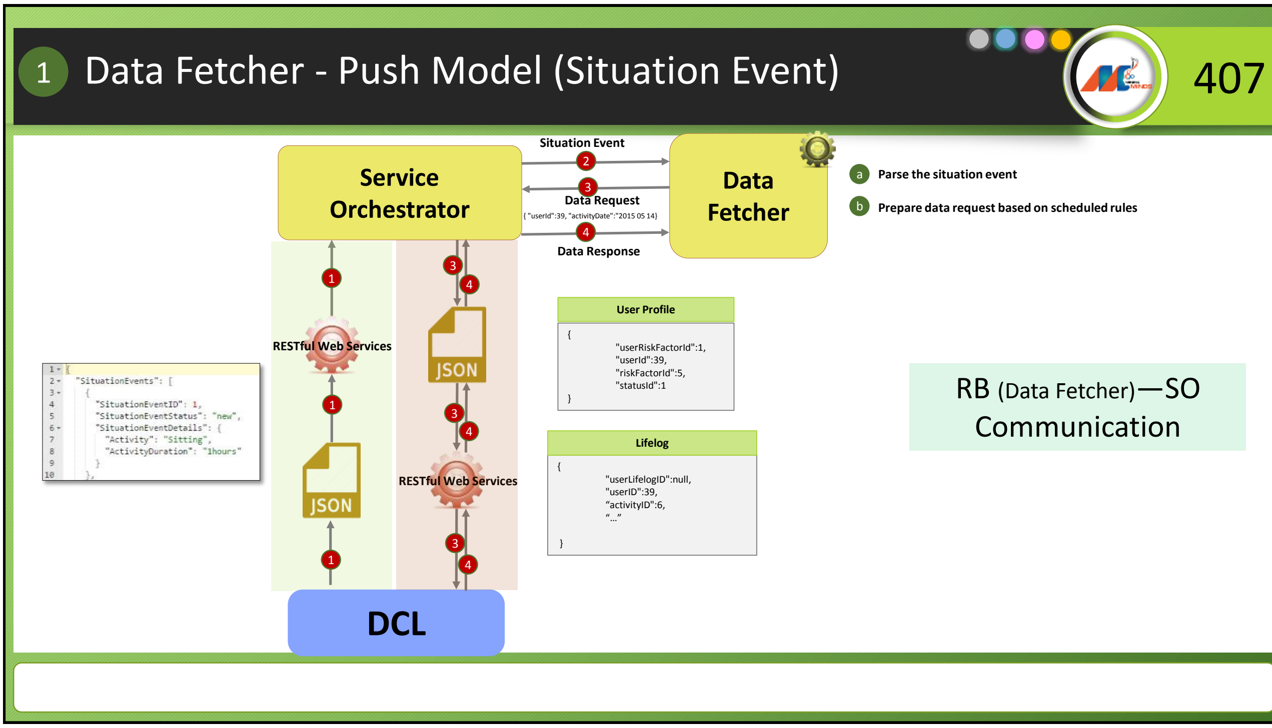
Data Fetcher

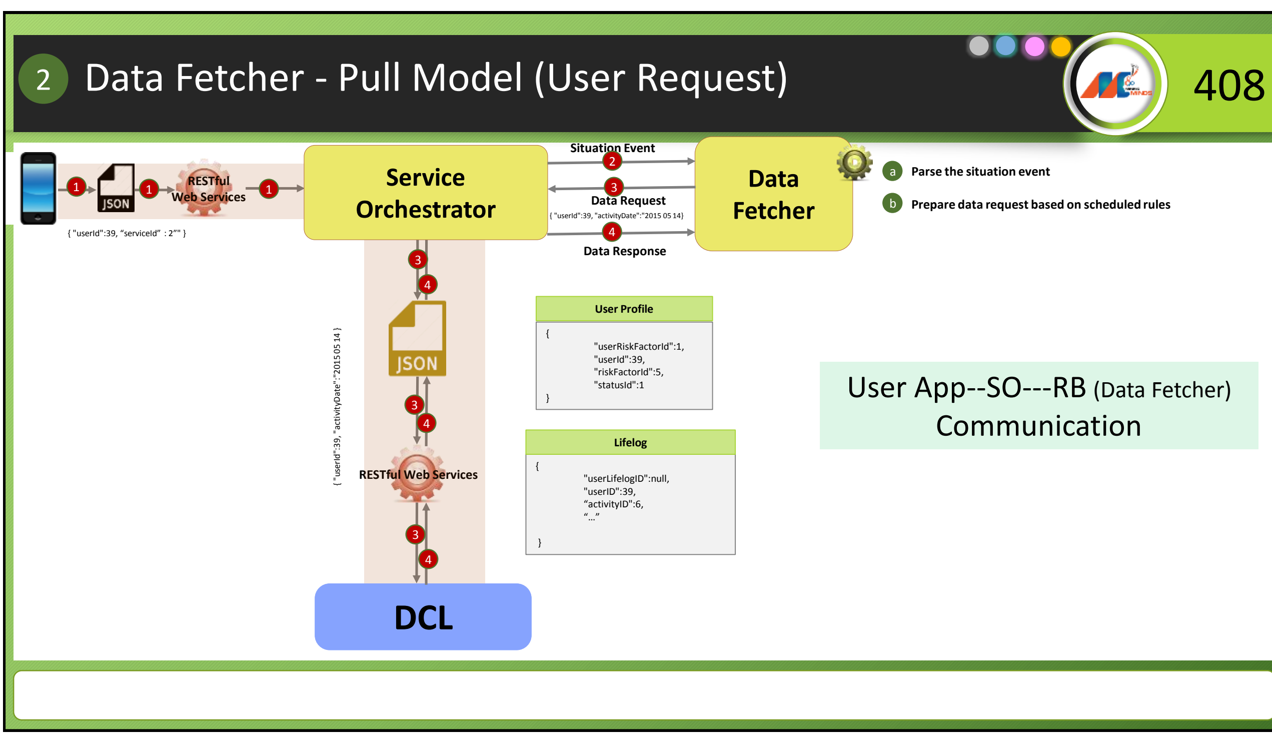
Methodology: Data Fetcher

Use-case (SCL2.5-SUC-01): *Load Data for Building Recommendation*

- **Objective**
  - Loads data from DCL through SO and forwards recommendations to SO
- **Methodology**
  - *Communication*: *Restful Service* between *SO* and *RB*
  - ***Data Fetcher*** *(operation)*
    1. **Input**: receives *SL* and *DCL* requests through *SO*
    2. **Processing**:
       a. parses requests based on *Service Id* and *Situation Event*
       b. requests data for condition attributes of the rules through *SO*
       c. receives data from *SO*
    3. **Output**:
       a. forwards loaded data to *Data Transformer*
       b. forwards recommendations to *SO*




1
Data Fetcher - Push Model (Situation Event)
407
Service Orchestrator
Data Fetcher
Situation Event
2
3
Data Request
{ "userId":39, "activityDate":"2015 05 14}
4
Data Response
a Parse the situation event
b Prepare data request based on scheduled rules
RESTful Web Services
JSON
User Profile
{
"userRiskFactorId":1,
"userId":39,
"riskFactorId":5,
"statusId":1
}
Lifelog
{
"userLifelogID":null,
"userID":39,
"activityID":6,
"..."
}
RB (Data Fetcher)—SO Communication
DCL


2
Data Fetcher - Pull Model (User Request)
408
JSON
RESTful Web Services
{ "userId":39, "serviceId" : 2"" }
Service Orchestrator
Data Fetcher
Situation Event
Data Request
{ "userId":39, "activityDate":"2015 05 14}
Data Response
a Parse the situation event
b Prepare data request based on scheduled rules
{ "userId":39, "activityDate":"2015 05 14 }
User Profile
{
"userRiskFactorId":1,
"userId":39,
"riskFactorId":5,
"statusId":1
}
Lifelog
{
"userLifelogID":null,
"userID":39,
"activityID":6,
"..."
}
User App--SO---RB (Data Fetcher) Communication
DCL

Use-case description

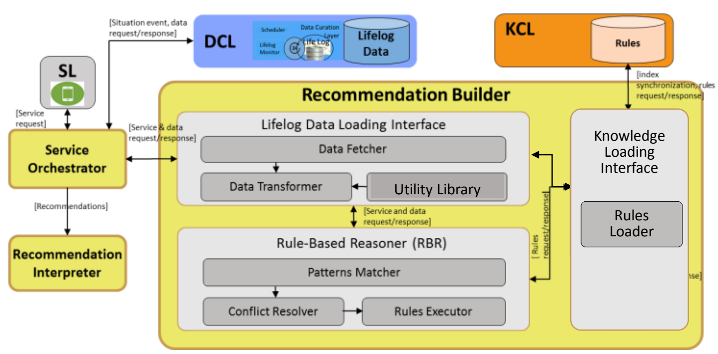
SCL2.5-SUC-03

Knowledge Loading Interface

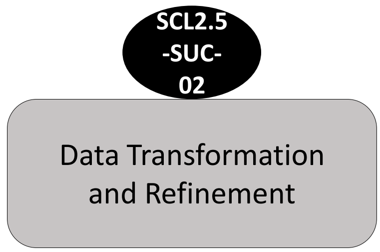
Methodology: Knowledge Loading Interface

Use-case (SCL2.5-SUC-03): *Load Rules*

- **Objective**
  - Enables Rules Integration
- **Methodology**
  - ***Communication***: *Restful Service* between *KCL* and *RB*
  - ***Rule Loader*** *(operation)*
    1. **Input**: *situation request*
    2. **Processing**:
       a. requests KCL for the rules under the specified situation event
       b. receives retrieved rules from KCL
       c. Provides rules to RBR
    3. **Output**:
       a. A subset of rules, which is forwarded to reasoner



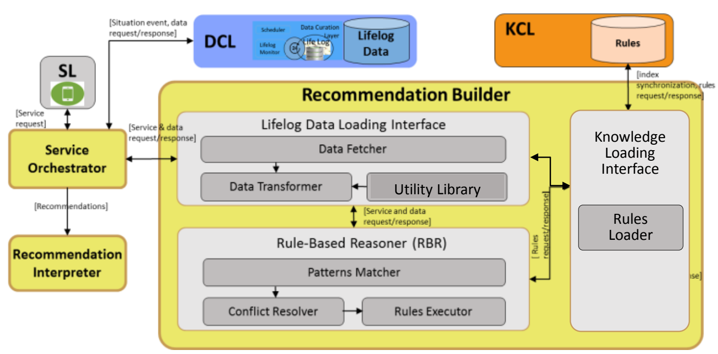
Use-case description



Methodology: Data Transformer & Refiner

Use-case (SCL2.5-SUC-02): *Prepare Data for Building Recommendation*

- **Objective**
  - Transforms and refines loaded data to be complaint with conditions of the rules
- **Methodology**
  - *Pre-request*: research on Understanding indexes, understanding concepts, formulas, and equations
  - *Transformer (operation)*
    1. **Input**: loaded lifelong data
    2. **Processing**:
       a. Parse rules condition part
       b. Conformance check of loaded data and condition attributes of the rules
       c. Transforms data to the formats of the attributes e.g., age calculation from date of birth etc.
       d. ***Utility Library*** is used for define and implement utility functions, such as computations on lifelog data and abstracting to required values using standard equations, formula (*e.g., METs* value)
    3. **Output**: Refined data complaint with rules



Data Transformation & Refinement - Example

Rule 1:

If Gender.value = Male, Age.value > 30, WeightStatus.value = Overweight, MET.value = 10.3 → **Recommendation** (Brisk walk for 30 minutes)

Data Transformer ← **Data Refiner** ← **Data Fetcher**

Transformation
- **Rule Parsing For Condition part**
  - **(**Gender, AgeGroup, WeightStatus, MET**)**
- **Conformance check**
  - (AgeGroup, WeightStatus, MET)
- **Refinement using utility function**
  - Dob → **Age** (computation)
  - Height, Weight→ **WeightStatus** (transformation)
    - Height, Weight → BMI
    - BMI → WeightStatus
  - Gender, Weight, Physical activities→ **MET** (computation)
    - Phy. Act → Accumulation → Calories → **METs**

Lifelog loaded Data
- Gender: Male
- DoB: 1985
- Height: 6ft
- Weight: 89Kg
- Physical activities
  - Walking: 20m
  - Running: 30m
  - .
  - .
  - Jogging: 10m

Use-case description

SCL2.5 -SUC- 04

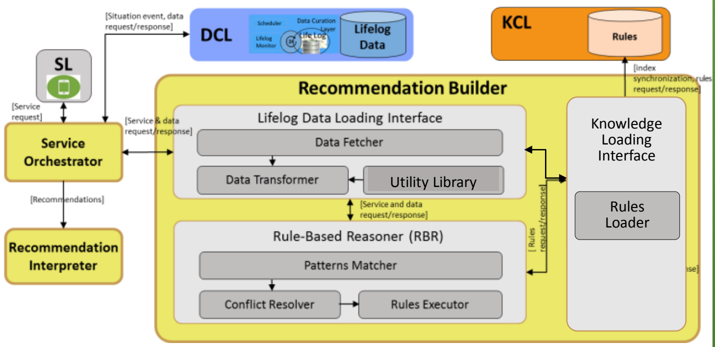
Rule-based Reasoner (RBR)

Methodology: Rule-based Reasoner

415

Use-case (SCL2.5-SUC-04):*Building Recommendations*

- **Objective**
  - ➢ To generate knowledge-based recommendations
- **Methodology**
  - ➢ *Rule-based Reasoning with Forward Chaining*
  - ➢ ***Pattern Matcher*** *(operation)*
    1. **Input**: Refined data + loaded rules
    2. **Processing**:
       a. matches data against condition part of the rules
       b. selects the matched rules
       c. creates the list of candidate rules
    3. **Output**: List of candidate rules
  - ➢ ***Conflict Resolver*** *(operation)*
    1. **Input**: *candidate rules*
    2. **Processing**:
       a. Filters appropriate rule using conflict resolution strategy
    3. **Output**: Final rules to execute
  - ➢ ***Rule Executer*** *(operation)*
    1. **Input**: *final rules*
    2. **Processing**:
       a. Binds data with conditions, generates rec.
    3. **Output**: Recommendations



경희대학교
KYUNG HEE UNIVERSITY

Rule-based Reasoning (Pattern Matcher) – Controlling Strategies

416

Controlling strategies

Forward chaining
(data driven)

- Works from *facts (conditions) to a conclusion*.
- Matches *data* against '*conditions*' of rules in the rule-base.

Backward chaining
(goal-driven)

- Works from the *conclusion to the facts (conditions)*.
- Matches a *goal* against *'conclusions*' of rules in the rule-base.

Mixed chaining
(forward-backward or backward-forward)

- Some rules are used for *chaining forwards*, and others for *chaining backwards*.
- System first chains in one direction, then switches to the other direction (use of meta-rules).

경희대학교
KYUNG HEE UNIVERSITY

Choice of appropriate control strategy

417

Forward chaining	Backward chaining
• Appropriate when all the facts are provided with the problem statement	• Appropriate when the goal is given in the problem statement
• Appropriate when there are many possible goals or there isn't any sensible way to guess what the goal is at the beginning of the reasoning.	• Appropriate when goal can sensibly be guessed at the beginning of the reasoning
• Appropriate for monitoring, planning, and interpretation applications	• Appropriate for Diagnostic, prescription and debugging applications
• Starts from data/request	• Starts from conclusion/decision
• Aims for finding conclusion(s)	• Aims for finding necessary data (reasons of decision)
• Bottom-up reasoning	• Top-down reasoning
• Breadth-first search	• Depth-first search
• Flow is from facts/conditions to conclusion	• Flow is from consequent to conditions

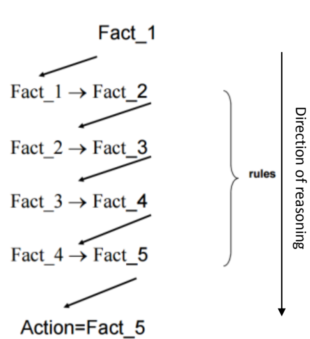
Our Choice is Forward Chaining

경희대학교 KYUNG HEE UNIVERSITY

http://www.ijetae.com/files/Volume2Issue10/IJETAE_1012_48.pdf

Forward chaining strategy

418



While (**goal** is not reached)

Foreach rule in the **Rulebase**

1. bind **Data** by matching to **Facts** (**conditions**)
2. if all **Facts** are matched, generate **Action (Conclusion**)
3. **Repeat** 1. and 2. for all matching/instantiation alternatives

Endfor

Continue

경희대학교 KYUNG HEE UNIVERSITY

Forward Chaining - Example

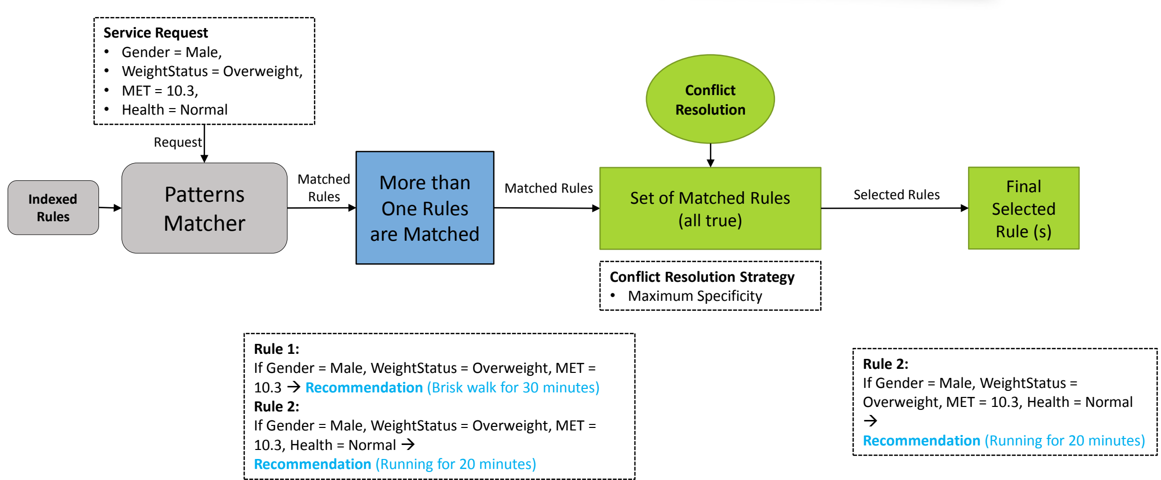
R#1: If gender=male, user_age=25, height=6.5 → **BMI** (<18, <19-25>, <25-30>, >30**)**

R#2: If BMI=X → **Weight Status** (UW, OW, N, OB)

R#3: If user_weight_Status= OW (overweight), health_Condition=Normal → **GoalForWeightLoss** = Burn 2000 Cal

R#4: If goal=2000 Cal, user_Interest=walking, phy_act_Routines=Active, diet_Routines=regular → **Brisk walk for 30 minutes**

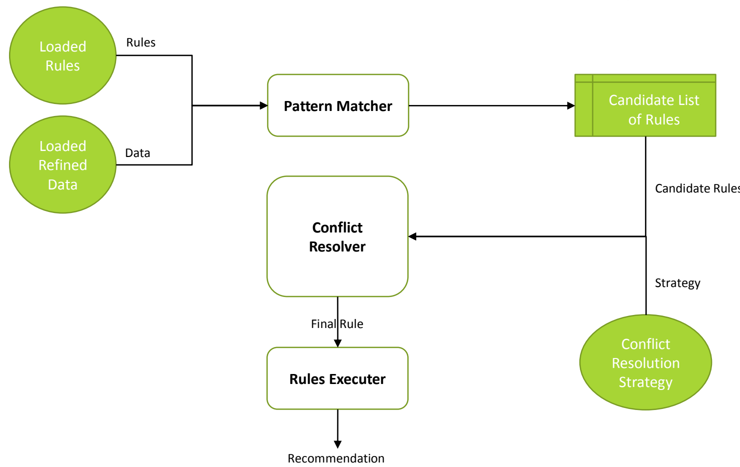
Challenge in forward chaining (pattern matching)



RBR with Forward Channing - Flowchart

421

- Reasoning with forward chaining and conflict resolution



Possible conflict resolution strategies

422

- Specificity or Maximum Specificity
  - based on number of condition attributes matched
  - choose the rule with the most/least matches for condition attributes
- Priority-based approach
  - arrange condition attributes in priority queue
  - use rule dealing with highest priority condition attributes
- Explicit /meta-rules for conflict resolution
  - rule based system within a rule based system
  - use meta-rules to resolve the conflict

- Fire All Selected Rules
  - Executes all the matched rules
- Context Limiting
  - partition rulebase into disjoint subsets
  - doing this we can have subsets and we may also have preconditions
- Execution Time
  - Chose the one with faster execution
- Physically ordering of rules
  - hard to add rules based on their physical order
- Random
  - Randomly pick one rule for execution

Rule-based Reasoning - Example

423

Rule 1:
If Gender = Male, WeightStatus = Overweight, MET = 10.3 → **Recommendation** (Brisk walk for 30 minutes)
Rule 2:
If Gender = Male, WeightStatus = Overweight, MET = 10.3, Health = Normal → **Recommendation** (Running for 20 minutes)
Rule 3:
If Gender = Male, WeightStatus = Overweight, MET = 10.3, Health = Disable, Age >30 → **Recommendation** (Jogging for 10 minutes + walk for 30 minutes)

Rule-based Reasoner ← **Data Transformer & Refiner**

Rule-based Reasoner

Pattern Matcher
- **Rule Condition Part Matching**
  - Matched rules: Rule 1, Rule 2
  - Not matched rule: Rule 3
  - Candidate Rules: Rule 1, Rule 2

Conflict Resolver
  - Conflicting List: Rule 1, Rule 2
- **Resolution**
  - Using *Maximum Specificity* resolution strategy
  - Rule 2 is selected for execution

Rules Executer
  - Fires Rule 2
  - Produce RHS of Rule 2 (Running for 20 minutes) as recommendations

Transformed and Refined Data
- Gender: Male
- DoB: 1985
- WeightStatus: Overweight
- Health: Normal
- MET: 10.3

경희대학교
KYUNG HEE UNIVERSITY

Tools and Technologies

424

- Language: Java
- Java Framework: Spring
- Web Server: Glassfish
- NetBean IDE/Eclipse
- Communications
  - Orchestrator-Recommendation builder (*Restful Services*)
  - Recommendation Builder - KCL (*Restful Services*)

경희대학교
KYUNG HEE UNIVERSITY

Contributions

425

- Indexed-based knowledge integration for reliable recommendations
- Transformation and refinement of user lifelog information for recommendations
- Design and development of flexible reasoning framework with forward chaining strategy
- Provisioning of non-conflicting accurate recommendations using specificity conflict resolution strategy

경희대학교
KYUNG HEE UNIVERSITY

Conclusions

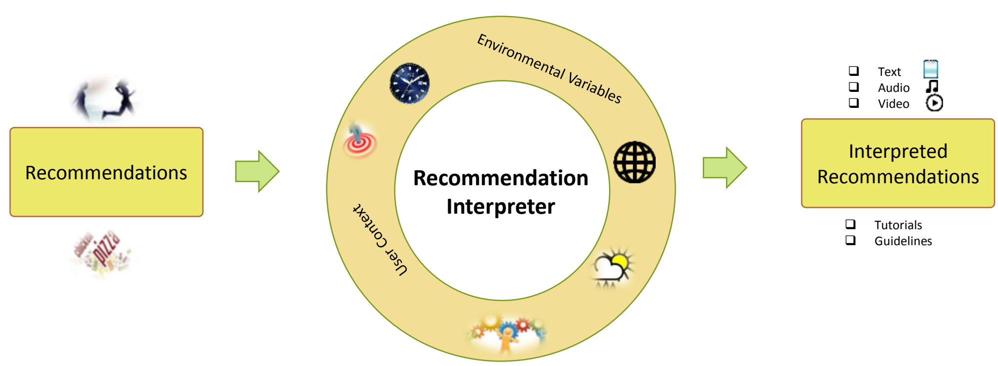
426

- Recommendation builder(RB) provides reliable recommendations using rule-based reasoning with forward chaining mechanism.
- A flexible reasoning framework with loosely coupled data and knowledge that supports diverse services
- Conflict resolution through the minimum/maximum attribute matching

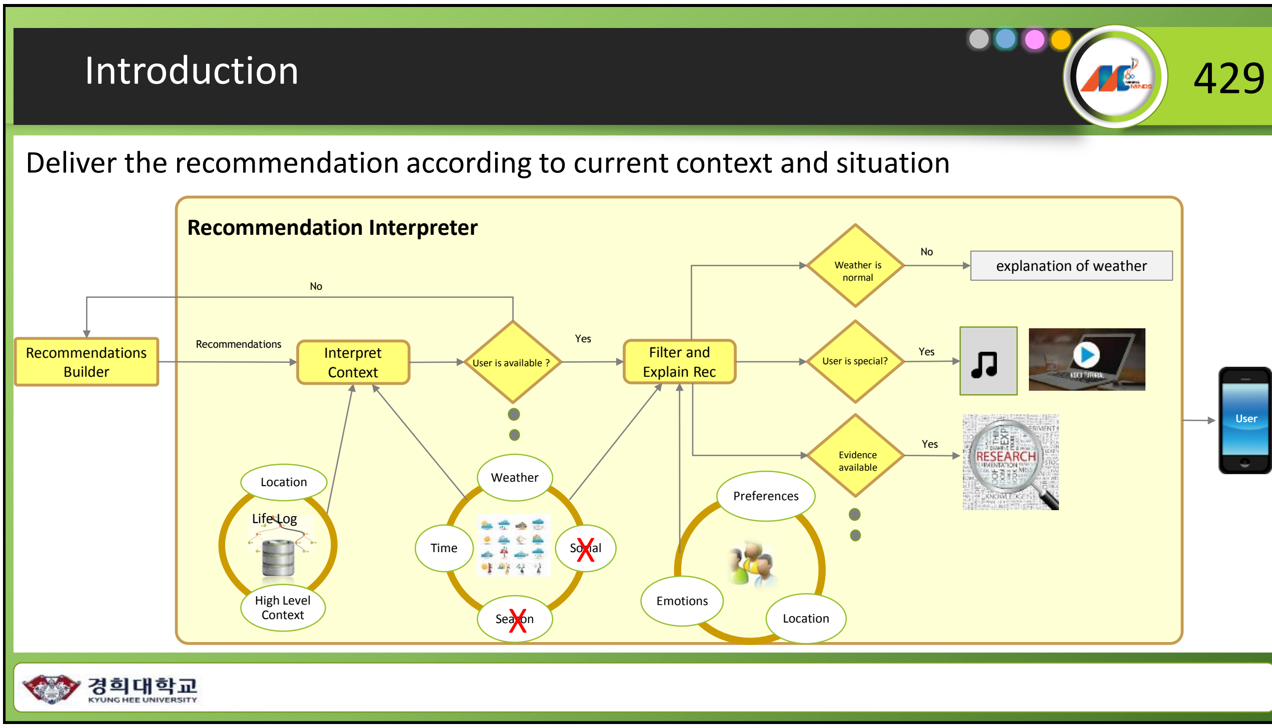
Recommendation Interpreter

Introduction: Interpretations

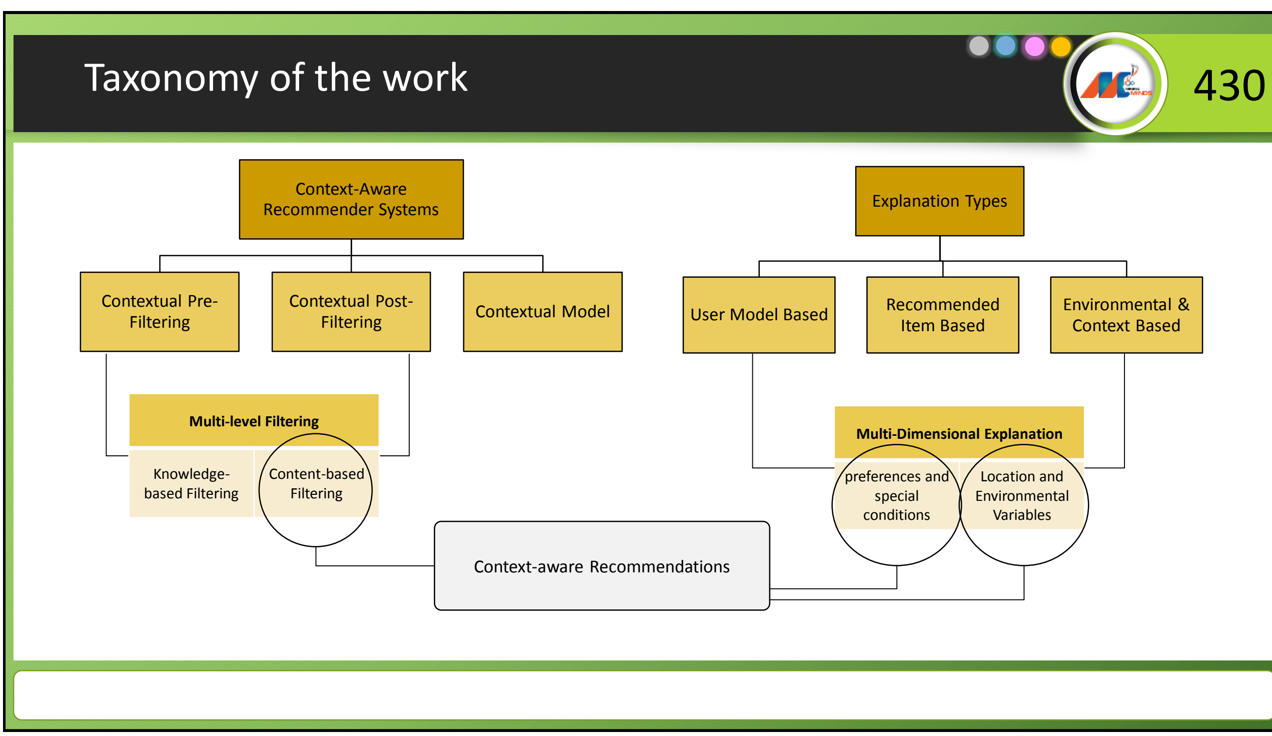
Interpret the recommendations according to the context of the user and environmental variables to deliver the appropriate recommendation at right time



경희대학교
KYUNG HEE UNIVERSITY



Introduction
429
Deliver the recommendation according to current context and situation
Recommendation Interpreter
Recommendations Builder
Recommendations
No
Interpret Context
User is available ?
Yes
Filter and Explain Rec
Weather is normal
No
explanation of weather
User is special?
Yes
Evidence available
Yes
RESEARCH
User
Location
Life Log
High Level Context
Weather
Time
Social
Season
Preferences
Emotions
Location
경희대학교
KYUNG HEE UNIVERSITY



Taxonomy of the work
430
Context-Aware Recommender Systems
Contextual Pre-Filtering
Contextual Post-Filtering
Contextual Model
Multi-level Filtering
Knowledge-based Filtering
Content-based Filtering
Explanation Types
User Model Based
Recommended Item Based
Environmental & Context Based
Multi-Dimensional Explanation
preferences and special conditions
Location and Environmental Variables
Context-aware Recommendations

Related Work

431

Argus - Pedometer and GPS Activity tracker

Features

- Daily Steps and active Calorie counter
- GPS mapping of Running, Cycling and Driving
- Creates Food diary of all your meals with food photo snapping
- Set daily goals for Steps, Sleep time and Hydration
- Observe trends over time with all variables to track wellness

Limitations

- Only counts and tracks not recommend
- No contextual filtration and explanation

[Azumio 2015]

Hybrid Reasoning Framework for Pervasive Healthcare

Features

- Context aware timely services for elderly people
- Generate recommendation based on case base reasoning and fuzzy rule base reasoning
- Context-aware sensor data fusion
- Anomaly detection to support daily activity living

Limitations

- Lack of incorporating special condition
- Lack of explanations

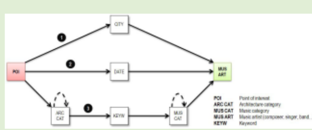
[Yuan 2014]

경희대학교 KYUNG HEE UNIVERSITY

Related Work

432

A Generic Semantic-based Framework for Cross-domain Recommendation



Features

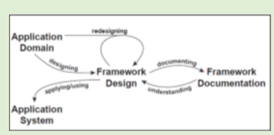
- Generate a generic-base description framework and provide cross-domain recommendation
- Generate open structure knowledge sources from multiple domains

Limitations

- Lack of contextual filtering
- No explanations

[Fernandez 2011]

CrosSing



Features

- Generic, flexible and dynamic structure for knowledge-based recommendation
- East integration of new domain

Limitations

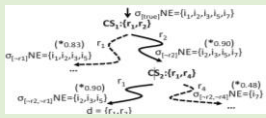
- Research based study…not real implementation
- Knowledge Acquisition process bottleneck

[AZAK 2010]

경희대학교 KYUNG HEE UNIVERSITY

Related Work

433

Plausible Repairs	Knowledgeable Explanation
	
Features	**Features**
• Generate personalized explanation on plausible repairs • Provide justification for recommended fault	• Predicate-based explanation • Finite state automata
Limitations	**Limitations**
• Applicable to item repair domain	• Generate explanations only when user interest matches
[Felfernig et al. 2009]	[Zanker and Ninaus 2010]

경희대학교
KYUNG HEE UNIVERSITY

Limitations of existing work

434

- Existing systems are largely contextual pre-filtering
- Deliver the recommendation to the user without considering the current context of the user.
- Not considering user emotion and environmental variables for explanations

Motivation

435

- To deliver recommendation at appropriate time based on user current context
- To filter out unnecessary recommendations based on user preferences
- To explain recommendation according to situation for better understandability

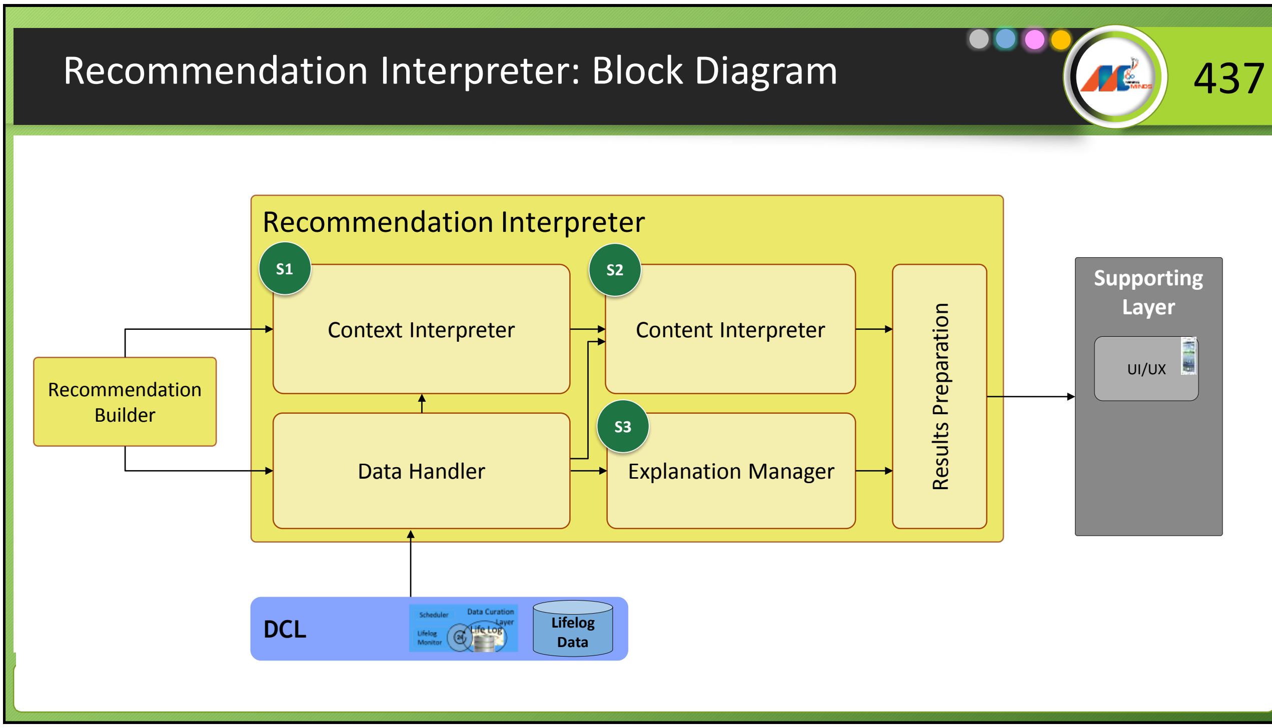
경희대학교 KYUNG HEE UNIVERSITY

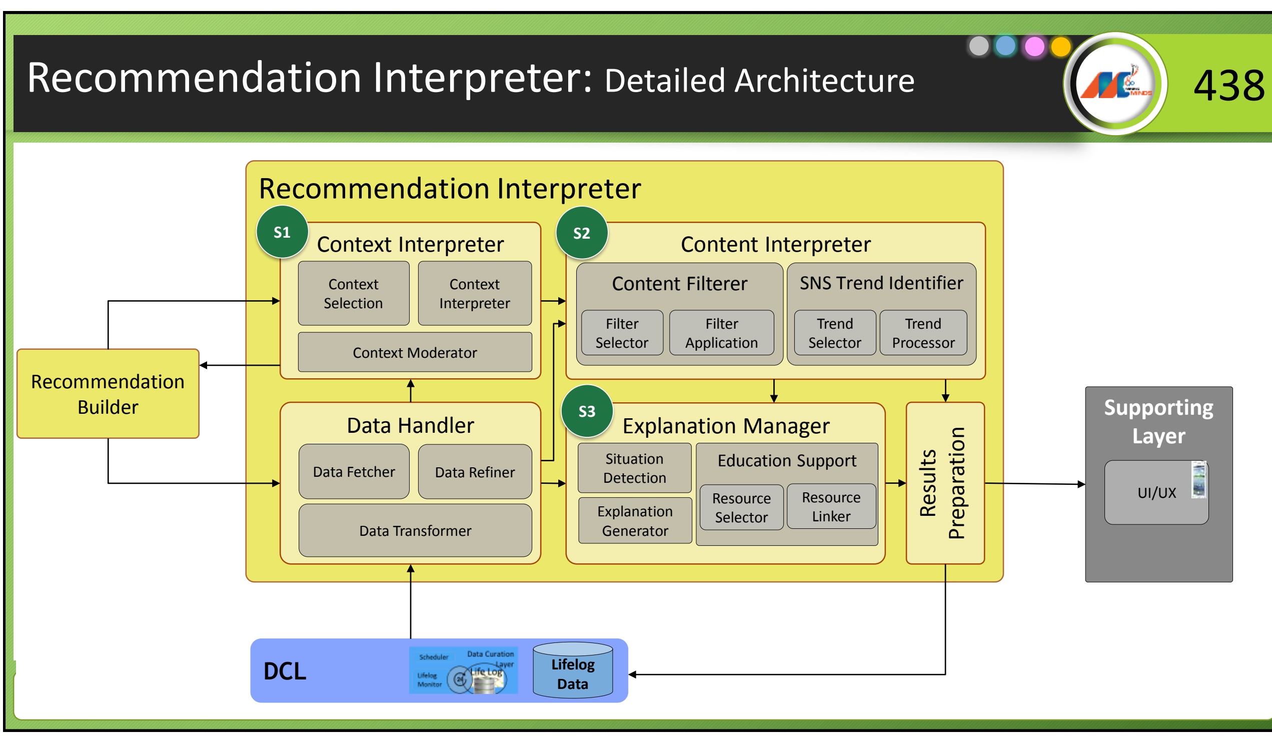
Challenges and Solutions

436

Motivations	Challenges		Solutions
Delivering recommendation based on user current context	→ Cross-context interpretations (When to deliver?)	S1	→ **Context aware recommendations**
Filtering out unnecessary recommendations based on user preferences	→ Exploiting user preferences (What, whom and how to deliver?)	S2	→ **User aware content filtration**
Explaining recommendation according to situation	→ Multidimensional explanations (Why this recommendation?)	S3	→ **Situation aware explanations**

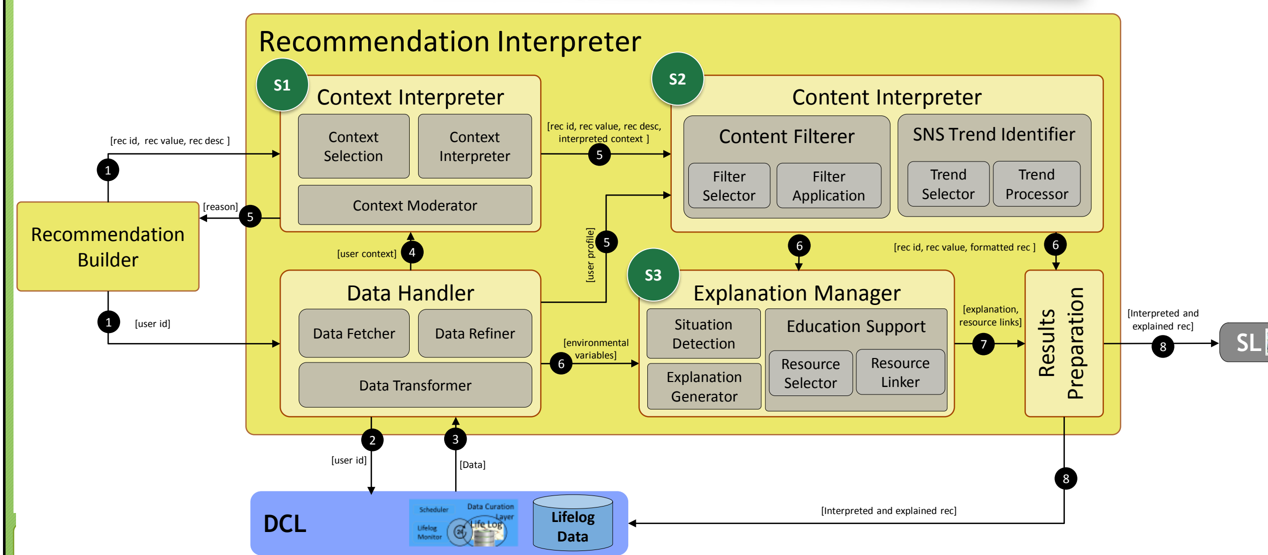
경희대학교 KYUNG HEE UNIVERSITY


Recommendation Interpreter: Block Diagram
437
Recommendation Interpreter
S1
Context Interpreter
S2
Content Interpreter
Results Preparation
Supporting Layer
UI/UX
Recommendation Builder
Data Handler
S3
Explanation Manager
DCL
Lifelog Data


Recommendation Interpreter: Detailed Architecture
438
Recommendation Interpreter
S1
Context Interpreter
Context Selection
Context Interpreter
Context Moderator
S2
Content Interpreter
Content Filterer
Filter Selector
Filter Application
SNS Trend Identifier
Trend Selector
Trend Processor
Recommendation Builder
Data Handler
Data Fetcher
Data Refiner
Data Transformer
S3
Explanation Manager
Situation Detection
Explanation Generator
Education Support
Resource Selector
Resource Linker
Results Preparation
Supporting Layer
UI/UX
DCL
Lifelog Data

Recommendation Interpreter: Data Flow

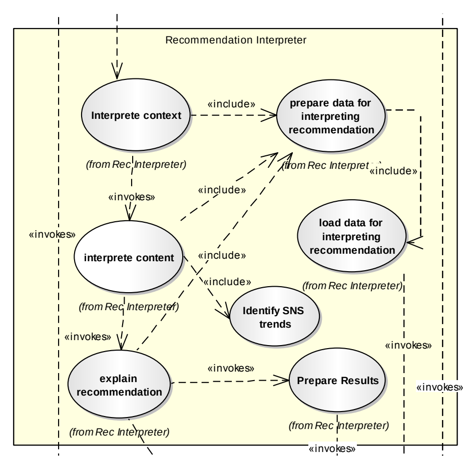
439



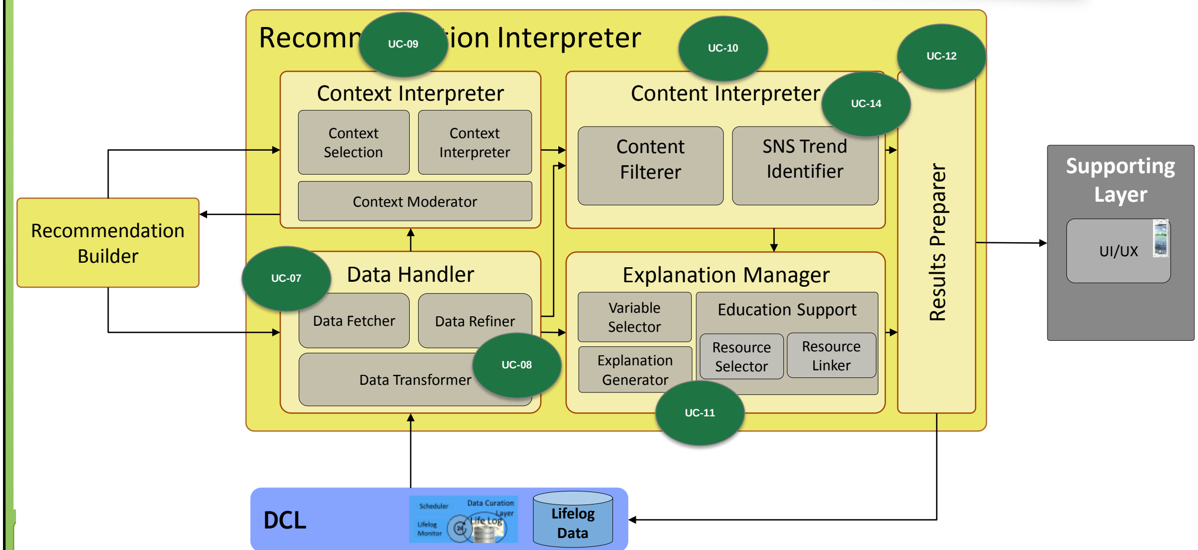
MM V 2.0 Implementation Scope

440

SCL2.5-SUC-05	Load data for interpreting recommendation
SCL2.5-SUC-06	Prepare data for interpreting recommendation
SCL2.5-SUC-07	Interpret Context
SCL2.5-SUC-8	Interpret Content
SCL2.5-SUC-9	Explain recommendations
SCL2.5-SUC-10	Prepare Results
SCL2.5-SUC-14	Identify SNS trends



Recommendation Interpreter: Refined Architecture



SCL2.5 -SUC- 09

Interpret Context

Interpret Context (Use Case 09): Methodology

443

Methodology Description

Use Case 09: *Interpret Context*

Objective: *To ensure the time is appropriate to deliver the recommendations.*

Input:

1. Recommendation 2. Context

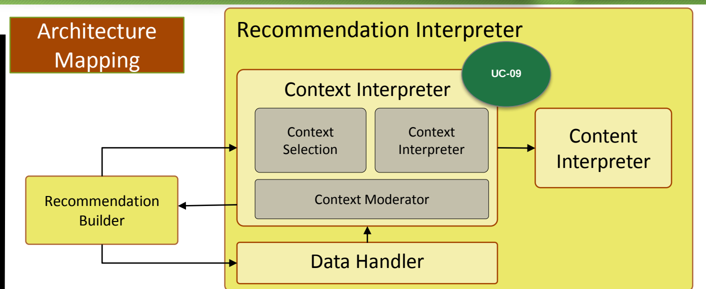
Process:

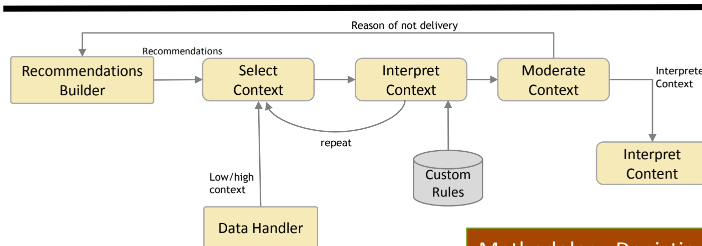
1. *Get the recommendation generated by RB*
2. *Selects context from fetched by Data Handler*
3. *Interpret the selected context*
   a. *Fetch the rules from custom rules repository*
   b. *Parse the rule on selected context*
4. *Moderate the context*
   a. *If time is appropriate → forward the rec and context to content interpreter*
   b. *Else inform the rec builder with reason*

Output:

Interpreted context

Architecture Mapping

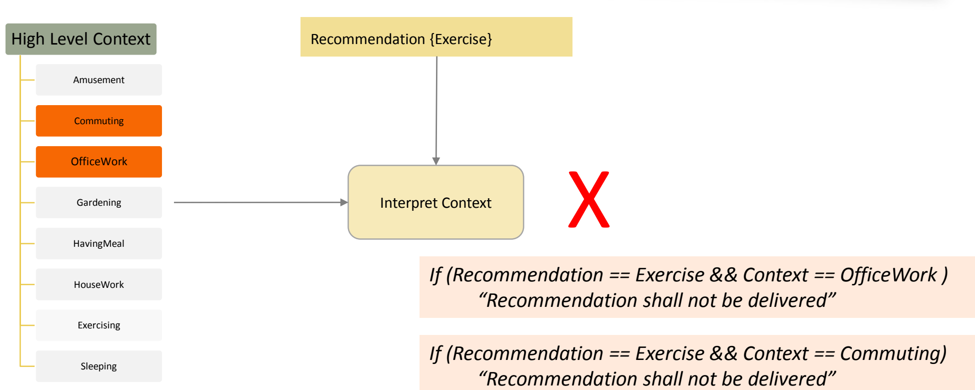


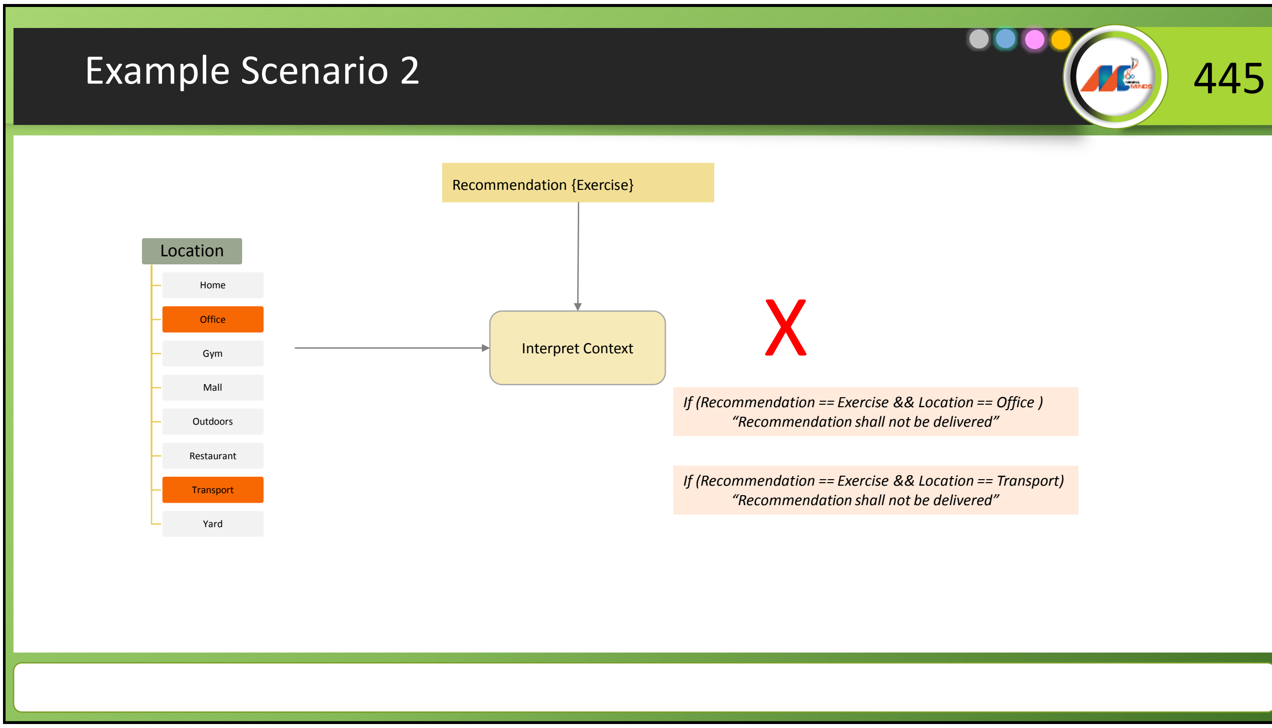


Methodology Depiction

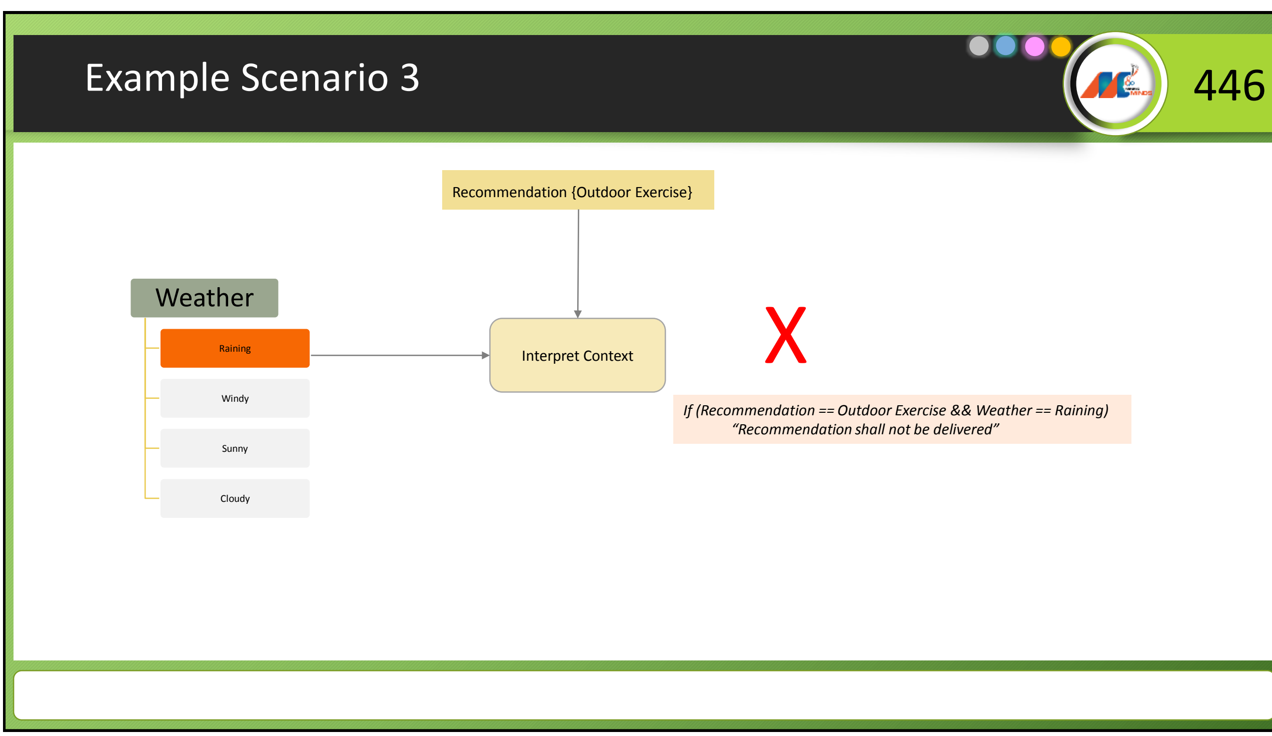
Example Scenario 1

444





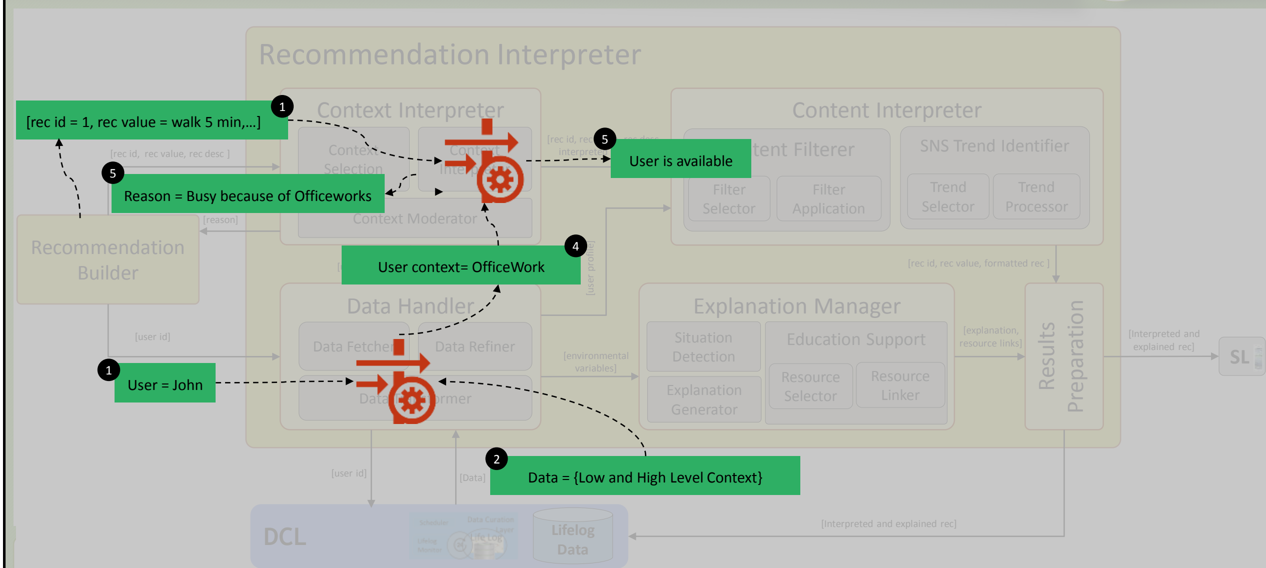
Example Scenario 2
445
Recommendation {Exercise}
Location
Home
Office
Gym
Mall
Outdoors
Restaurant
Transport
Yard
Interpret Context
X
If (Recommendation == Exercise && Location == Office)
"Recommendation shall not be delivered"
If (Recommendation == Exercise && Location == Transport)
"Recommendation shall not be delivered"



Example Scenario 3
446
Recommendation {Outdoor Exercise}
Weather
Raining
Windy
Sunny
Cloudy
Interpret Context
X
If (Recommendation == Outdoor Exercise && Weather == Raining)
"Recommendation shall not be delivered"

Interpret Context (Use Case 09): Example Scenario

447



448

SCL2.5
-SUC-
10

Interpret Content

경희대학교
KYUNG HEE UNIVERSITY

Interpret Content (Use Case 10): Methodology

449

Methodology Description

Use Case 10: *Interpret Content*

Objective: *To ensure the contents are filtered according to user preferences.*

Input:

1. Interpreted context(s) 2. User Preferences

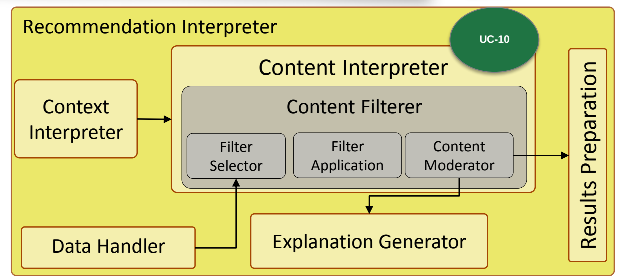
Process:

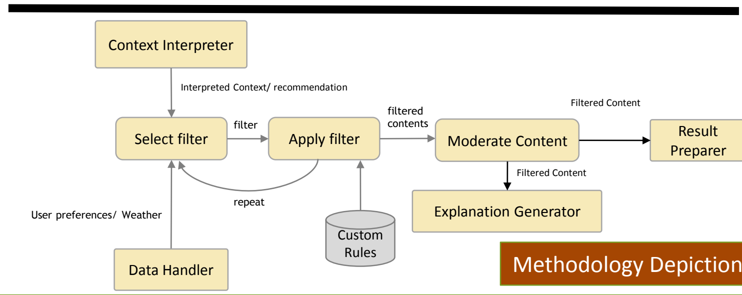
1. *Selects the context filter from context interpreter and (user preferences) filters from data handler*
2. *Content Filtration*
   a. *Get the data (recommendation and filter)*
   b. *Fetch the rules from custom rules repository*
   c. *Parse the rule on selected filter*
3. *Moderate the context*
   a. *If contents are understandable → forward the filtered contents to Result Preparer*
   b. *Else forward to the explanation manager*

Output:

Filtered content

Architecture Mapping

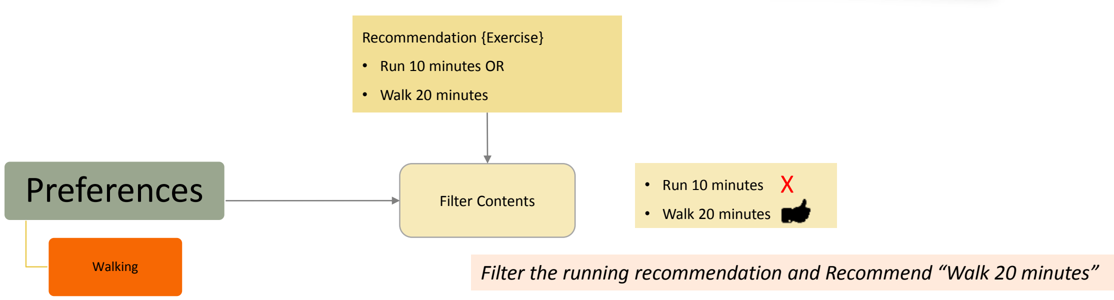




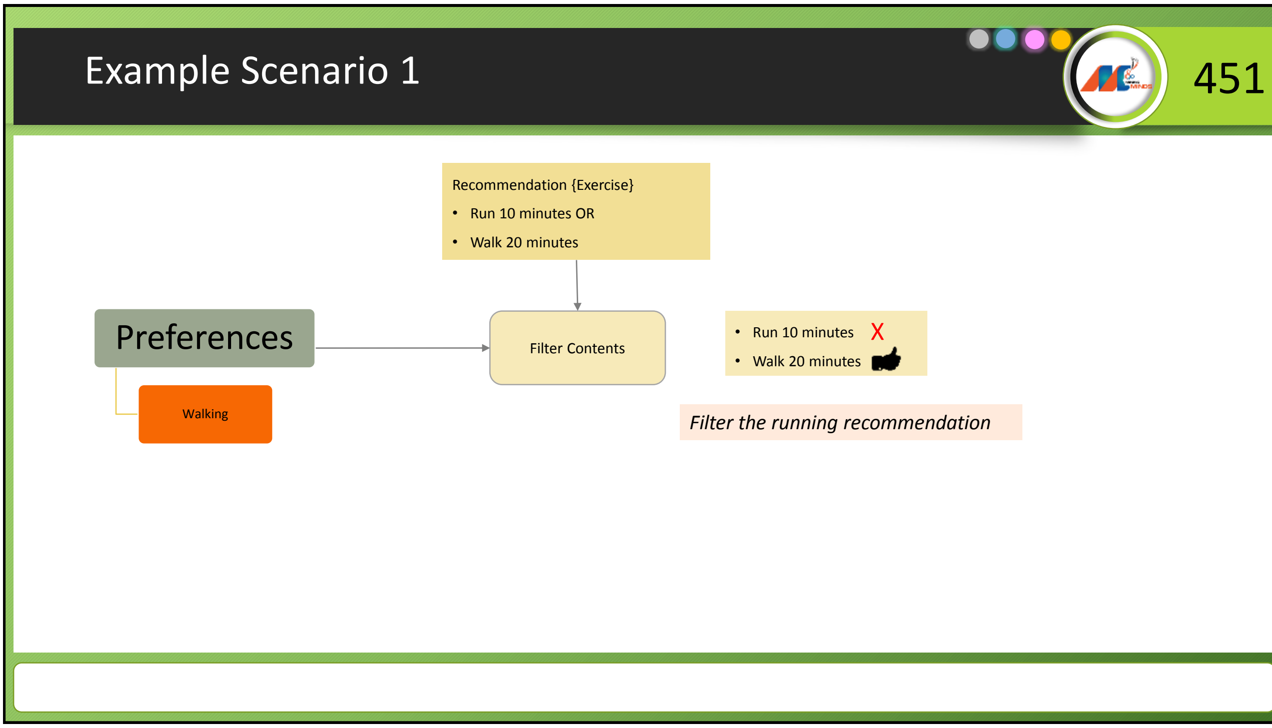
Methodology Depiction

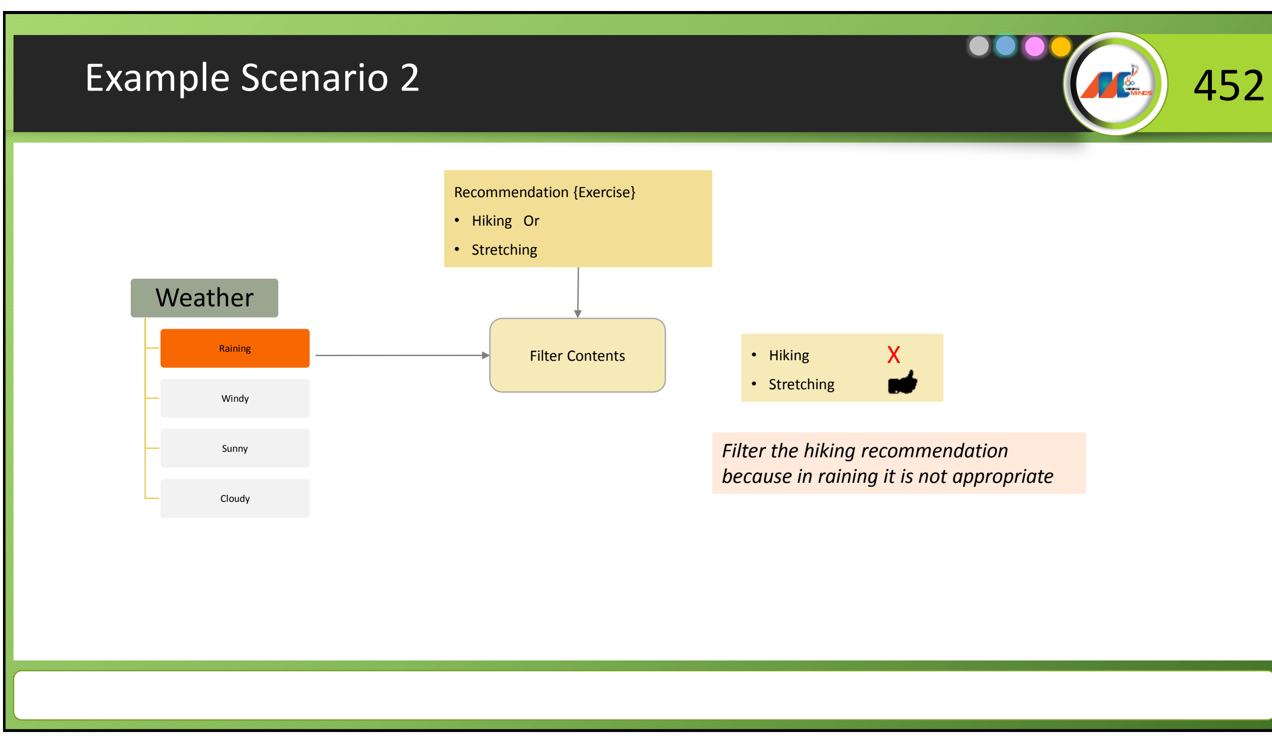
Example Scenario 1

450

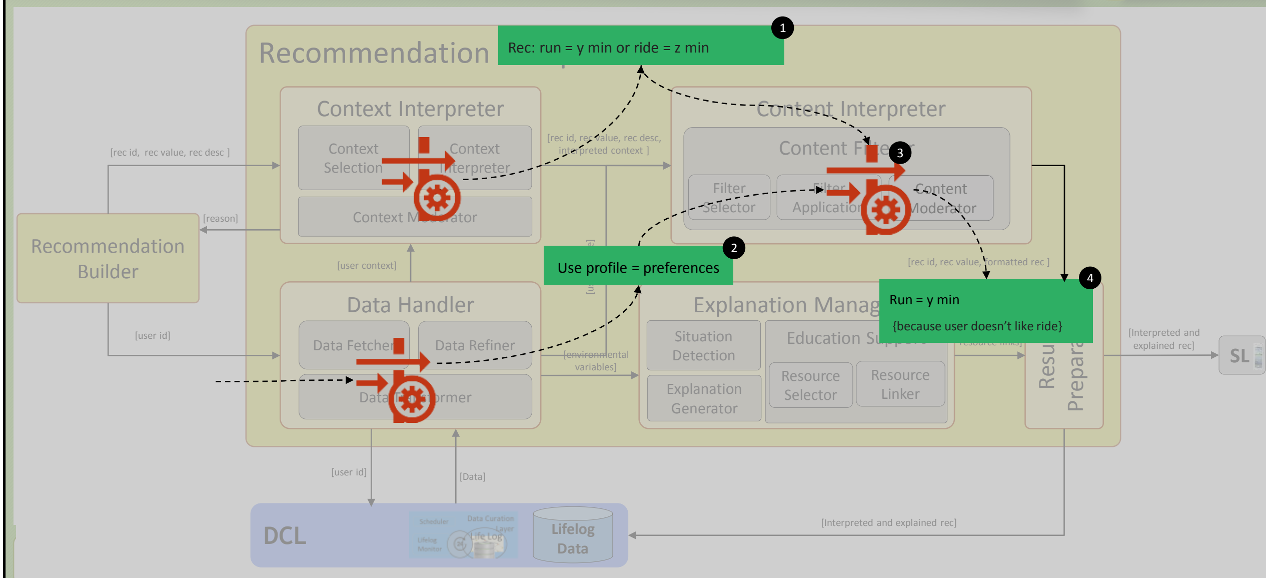


Filter the running recommendation and Recommend "Walk 20 minutes"


Example Scenario 1
451
Recommendation {Exercise}
• Run 10 minutes OR
• Walk 20 minutes
Preferences
Walking
Filter Contents
• Run 10 minutes X
• Walk 20 minutes
Filter the running recommendation


Example Scenario 2
452
Recommendation {Exercise}
• Hiking Or
• Stretching
Weather
Raining
Windy
Sunny
Cloudy
Filter Contents
• Hiking X
• Stretching
Filter the hiking recommendation
because in raining it is not appropriate

Interpret Content (Use Case 10): Example Scenario



SCL2.5 -SUC- 11

Explain Recommendation

Explain Recommendation (Use Case 11): Methodology

455

Methodology Description

Use Case 11: *Explain Recommendation*

Objective: *To enhance the understandability of recommendation.*

Input:

1. Filtered Recommendation 2. Variables (Weather, Emotion, Location)

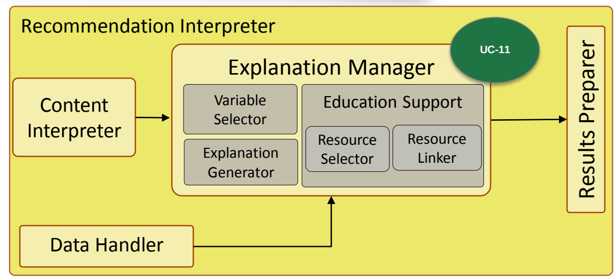
Process:

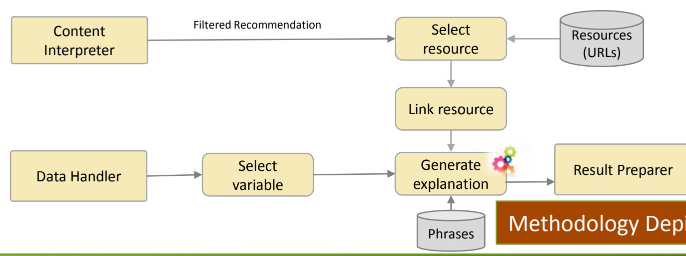
1. *Select the data of weather, location and emotion, from Data Handler*
2. *Explanation Generation*
   a. *Get the data (weather/location/emotion)*
   b. *Fetch the phrases from the phrase repository*
   c. *Get the URL for linked resource selected by resource selector.*
   d. *Generate explanation based on the matched phrase.*
3. *Forward the explanation to result preparer.*

Output:

Explained Recommendation

Architecture Mapping

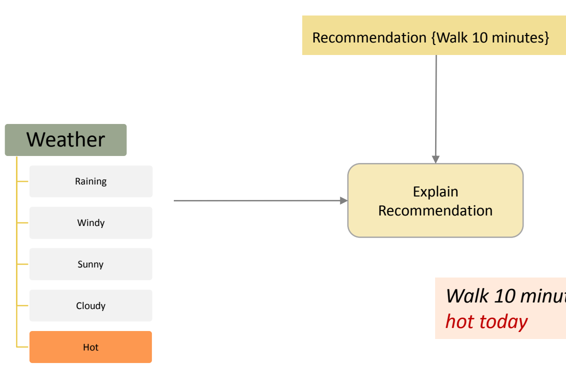




Methodology Depiction

Example Scenario 1

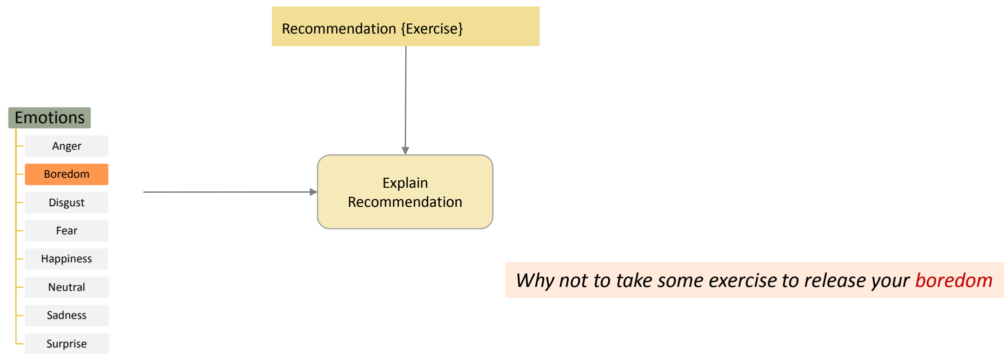
456



Walk 10 minutes: take water with you while walking because weather is too hot today

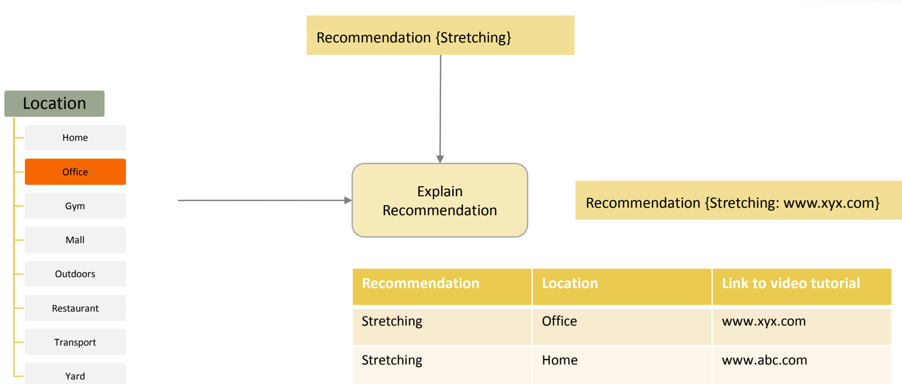
Example Scenario 2

457



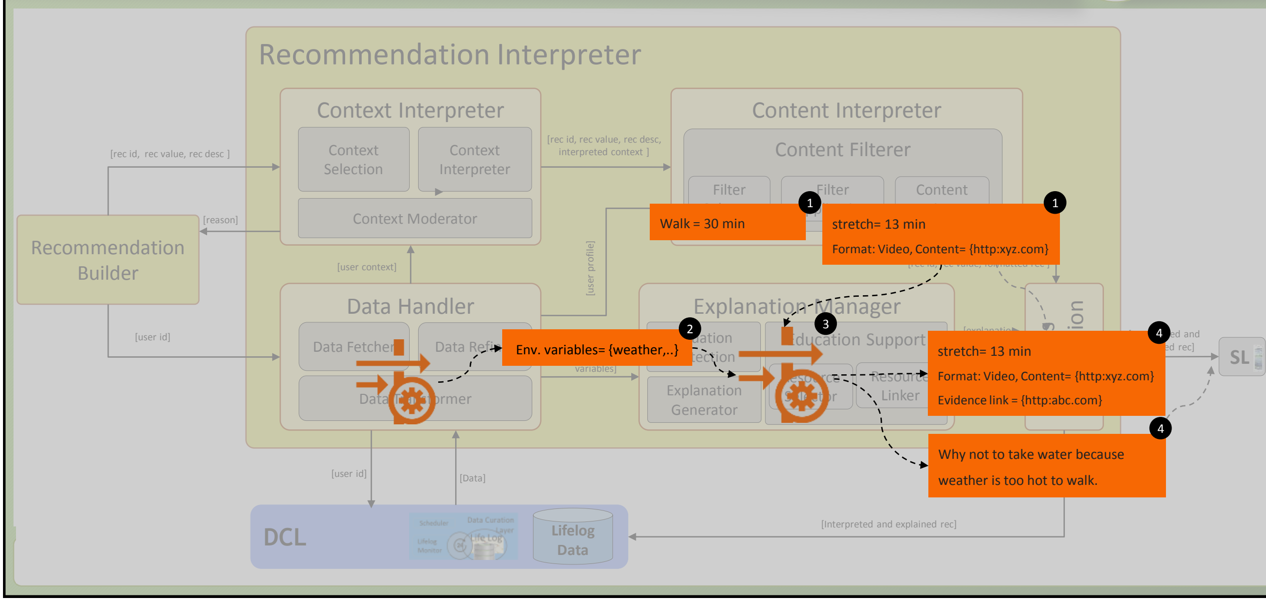
Example Scenario 3: Education Support

458



Recommendation	Location	Link to video tutorial
Stretching	Office	www.xyx.com
Stretching	Home	www.abc.com
--	--	--

Explain Recommendation (Use Case 11): Example Scenario



SCL2.5 -SUC- 12

Prepare and Deliver Results

경희대학교
KYUNG HEE UNIVERSITY

Prepare and Deliver Results (Use Case 12): Methodology

UC-12

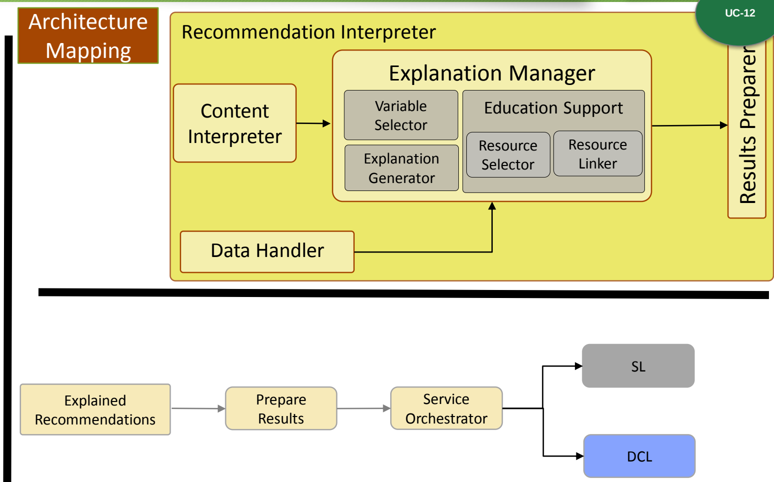
Methodology Description

Use Case 12: *Prepare and Deliver the results*
Objective: *To ensure the results are delivered correctly.*
Input:
1. *Interpreted and Explained Recommendation*

Process:
1. *Get the explained recommendation from Explanation Manager*
2. *Prepare the results*
3. *Deliver the results*
   a. *Persist the recommendation in intermediate database*
   b. *Forward the recommendation to UI/UX of supporting layer.*

Output:
Prepared Results

Architecture Mapping



Methodology Depiction

SCL2.5-SUC-14

Social Trend Identifier

경희대학교 KYUNG HEE UNIVERSITY

Social Trends Analyzer (Use Case 14): Methodology

Methodology Description

Use Case 14: *Social Trend Identifier*

Objective: *To provide popular and preference based nutrition recommendations*

Input:

1. Recommendation 2. Food Preferences

3. Nutrition Items 3. SNS Trends

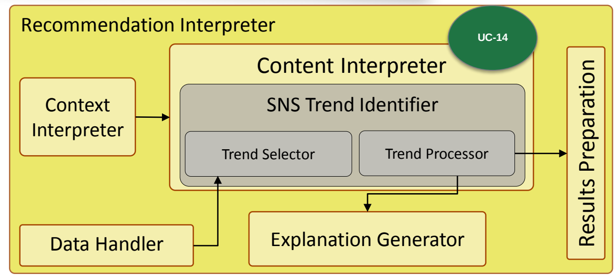
Process:

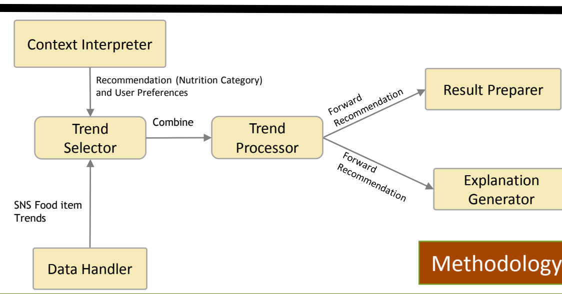
1. *Receive Recommendation and food Preferences*
2. *Receive Food items from DCL*
3. *Send Food items to SL*
4. *Receive Food items SNS Trends*
5. *Evaluate User's Preferences and SNS Trends*
6. *Pick most relevant Food items for recommendation*
7. *Forward the recommendation to result preparer*

Output:

Preference based Nutrition Recommendation

Architecture Mapping





Methodology Depiction

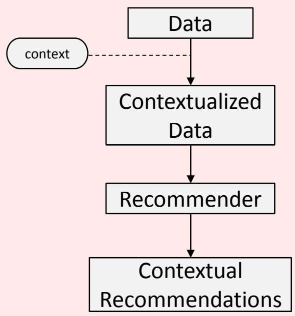
Strategies

Filtering Strategies

465

Pre-contextual Filtering

Use context prior to generate recommendations

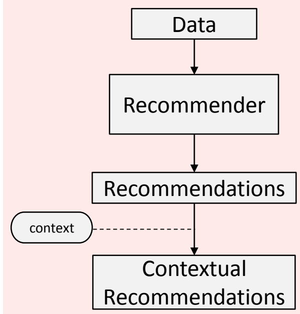


Preferable when recommendation are required to generate only contextualized data

Mostly it is used in rating recommender systems.

Post-contextual Filtering

Use context after the generation of recommendations



Preferable when a system recommends first and contextualize afterward.

In Mining Mind post-filtering strategy is more suitable because of two reasons:

1. Architecture is fit for this strategy
2. Taking advantage of non-contextualized data to generate recommendations

Explanation Strategies

466

White Box

- It discloses and exploit the underlying conceptual model of the recommendation engine
- It explains "how" the recommendation is generated
- Explicit

Black Box

- It does not disclose the functioning of the system to user
- It explains "why" the recommendation is generated
- Implicit
- It adds reasons to satisfy the user

- In wellness system, black box explanation are more appropriate and suitable where users are most likely interested in in reasons rather in internal details.

- In clinical systems, white box explanation are preferable to provide the internal details of the system.

Tools and Technologies

467

- Language: Java
- Java Framework: J2EE
- Web Server: Glassfish
- IDE: NetBeans/Eclipse
- Communications
  - Rec Builder-Rec Interpreter (*Method Call*)
  - Rec Interpreter – SL (Restful Service)

경희대학교 KYUNG HEE UNIVERSITY

Contributions

468

- Delivering recommendation at right time
  - Context Interpreter
- Delivering the right recommendation to the user by filtering the unnecessary contents
  - Content Interpreter
- Advocate the recommendation with reason for increasing the ownership of the towards the system and understandability
  - Explanation manager

Conclusion

- Recommendation Interpreter interprets the context and contents before delivering the recommendation to the end user
- It exploits the user context in order to particularize and contextualize the recommendation according to user situation.
- It provides additional information by providing educational facts and social trends.

Service Curation Layer (SCL 2.5)

Descriptive Analytics

경희대학교
KYUNG HEE UNIVERSITY

Background

472

V 1.0 Traditional Analytics

- Internally Sourced and relatively small structured data
- Teams of Analysts
- Internal Decision support

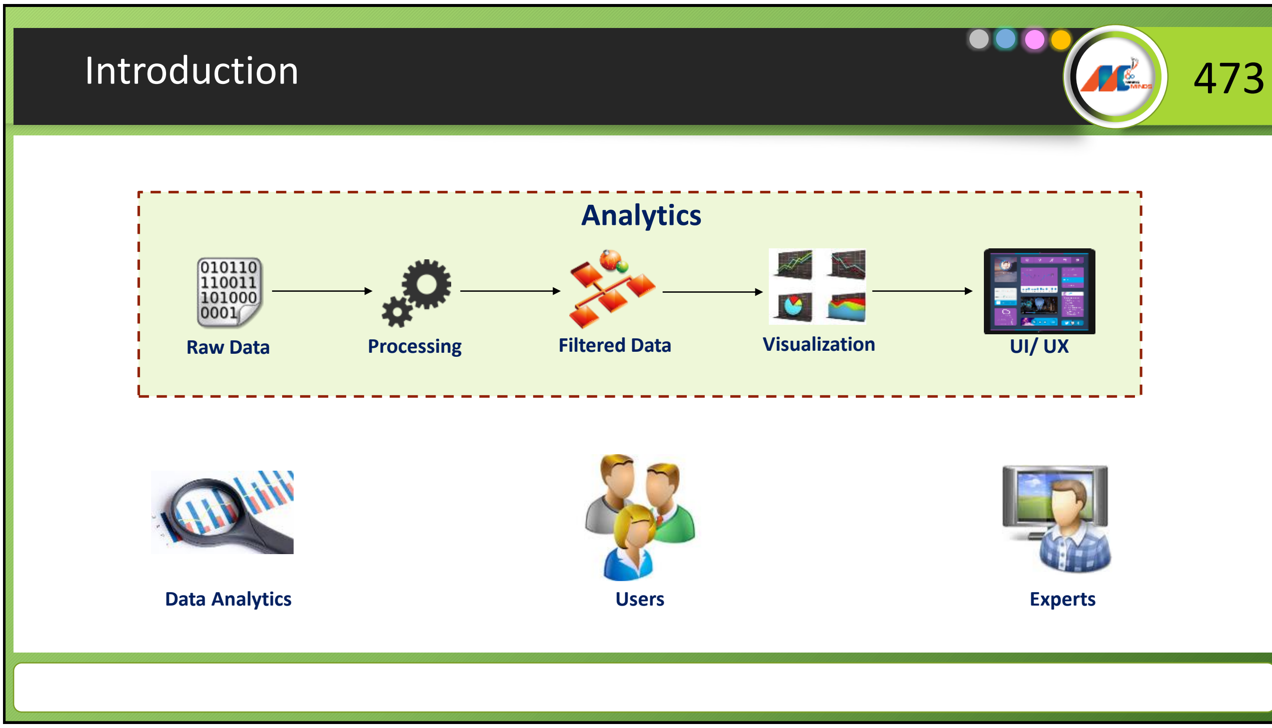
V 2.5 Descriptive Analytics

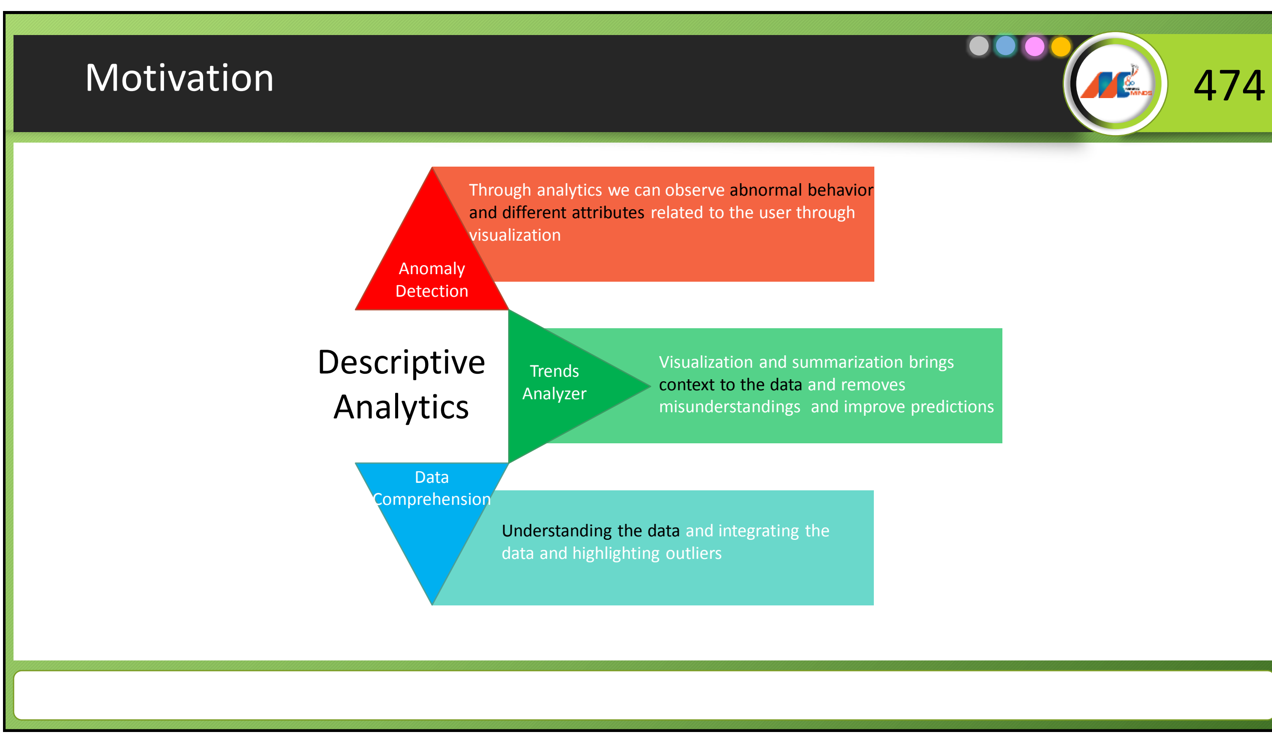
- Complex Large unstructured data sources
- New Analytical and computation capabilities
- Data based Product and Services

Provide trending information

Quantitative summary

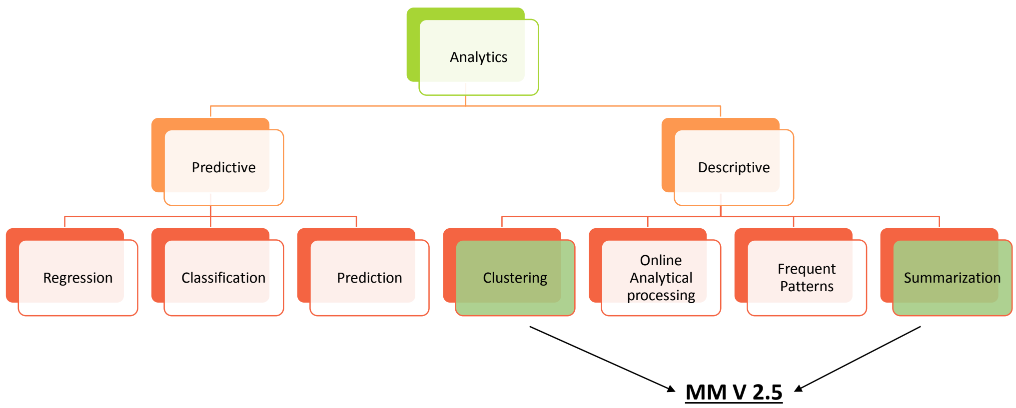
Data visualization


Introduction
473
Analytics
010110
110011
101000
0001
Raw Data
Processing
Filtered Data
Visualization
UI/ UX
Data Analytics
Users
Experts


Motivation
474
Anomaly Detection
Through analytics we can observe abnormal behavior and different attributes related to the user through visualization
Descriptive Analytics
Trends Analyzer
Visualization and summarization brings context to the data and removes misunderstandings and improve predictions
Data Comprehension
Understanding the data and integrating the data and highlighting outliers

Taxonomy

475



A Survey of Predictive Analytics in Data Mining with Big Data ATHABASCA UNIVERSITY July 2014

Related Work

476

	Systems	Limitations
Clustering	• The automated extraction of interrelated data objects from ERP systems is discussed but without using a graph model and for the single analytical goal of process mining	• No parameter type classification • No distribution of data ranges and rendering information
Clustering	• Gradoop (Graph analytics on Hadoop) analyze graph data for business intelligence and social network analysis.	• Only graph analytics and focus on trends based on images • It stores graph formats only
Summarization	• Radoop is a big data analytics solution for Hadoop which computes the jobs on the cluster using ensemble learning	• It is based on complex machine learning techniques • Less information as it is being developed into a product.

Rudolf, Michael, et al. "SynopSys: large graph analytics in the SAP HANA database through summarization." *First International Workshop on Graph Data Management Experiences and Systems*. ACM, 2013.
Junghanns, Martin, et al. "GRADOOP: Scalable Graph Data Management and Analytics with Hadoop." *arXiv preprint arXiv:1506.00548* (2015).
radoop.eu

Limitations of Existing Work

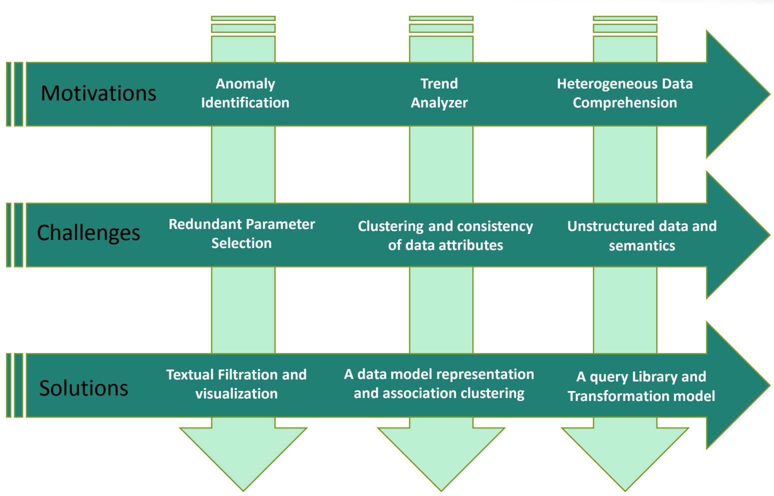
477

- No query library for mapping the queries, lack of which results in inefficient data retrieval in case of big data.
- Lack of Parameter filtration which makes it difficult to group and summarize the data.
- No model transformation for preserving semantics of data and integration of heterogeneous data

경희대학교 KYUNG HEE UNIVERSITY

MM V2.5 Challenges and Solutions

478



경희대학교 KYUNG HEE UNIVERSITY

Solutions and Methods

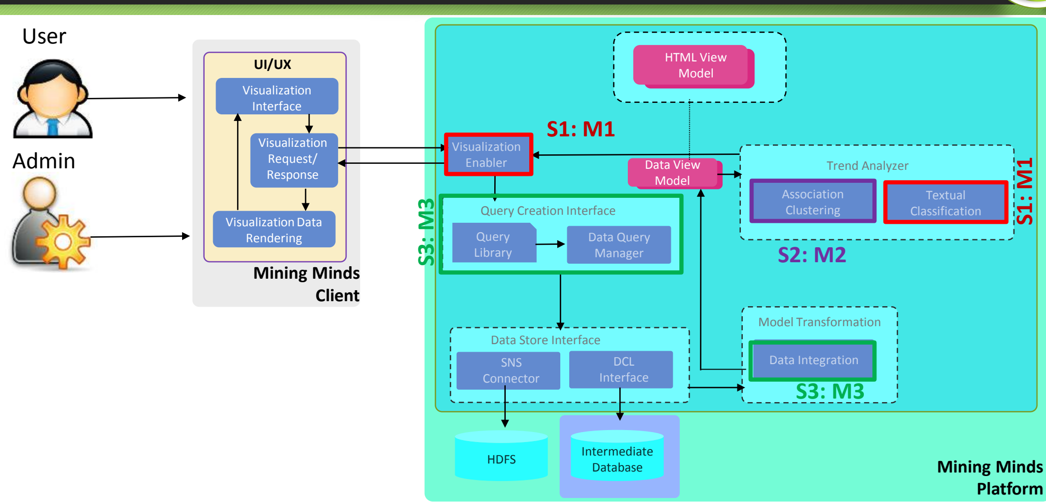
479

Solutions		Methods
S1	Textual Filtration and visualization	M1: Taking additional information about the data for filtering into categories and visualizing data based on the predefined templates.
S2	A data model representation and association clustering	M2: Designing a data model and associating data and grouping them according to a data model
S3	A query Library and transformation Guidelines	M3: A Query Manager and information about data formats for transformation

경희대학교
KYUNG HEE UNIVERSITY

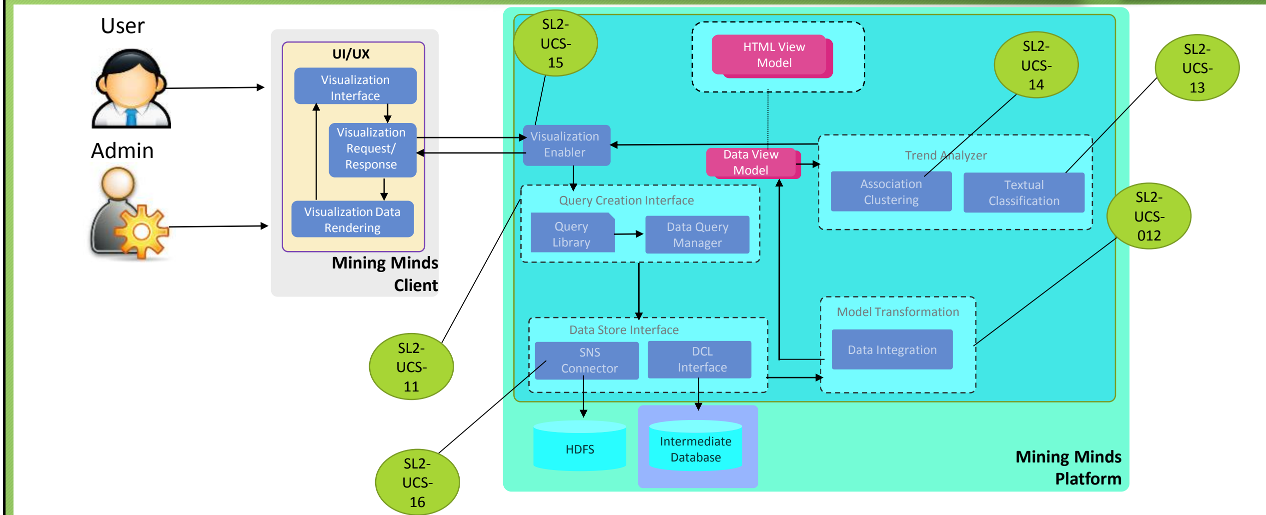
Descriptive Analytics in MMV2.5

480



Descriptive Analytics in MMV2.5

481

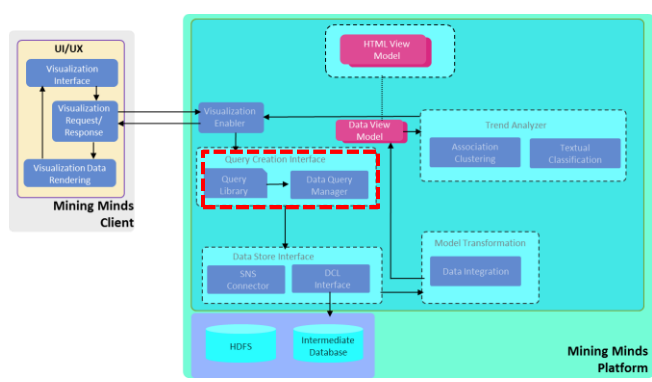


Methodology : Query Creation Interface

482

Use-case (SL2.5-UCS-11): *Map Request to Query*

- **Objective**
  - Map user request to query library to fetch data
- **Methodology**
  - *Communication*: *Request Response with UI*
  - ***Query Creation Interface*** *((Request/Operation)*
    1. **Input**: receives *request from the expert*
    2. **Processing**:
       - The parameters of the request is extracted and sent to query manager.
       - The query is broken in conditional, temporal and return attributes.
       - The query manager matches the parameters with the predefined queries in the query library.
    3. **Output:**
       a. The query is selected to be posed on the data store interface and data is extracted.

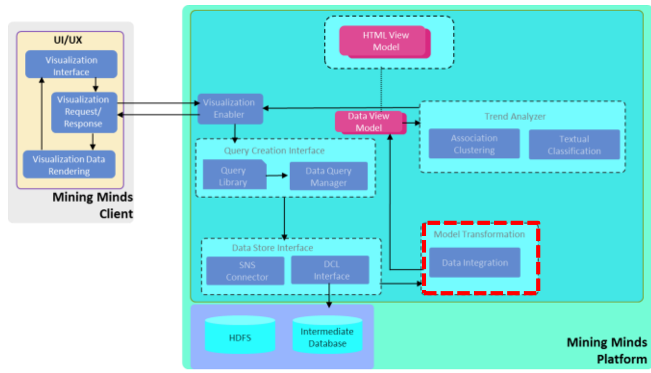


Methodology : Model Transformation

483

Use-case (SL2.5-UCS-12**):** *Transform Data*

- **Objective**
  - The mapping query is to be transformed to specific model structure for trend analysis.
- **Methodology**
  - *Communication*: *Data is passed to model transformation from data store interface*
  - ***Data Integration****(operation)*
    1. **Input**: receives *request from the expert*
    2. **Processing**:
       - The unstructured data from the big data repository is sent to the data integration component.
       - The data is transformed in a table/JSON.
       - The query manager matches the parameters with the predefined queries in the query library.
    3. **Output**:
       a. The data is transformed into the standard JSON format.

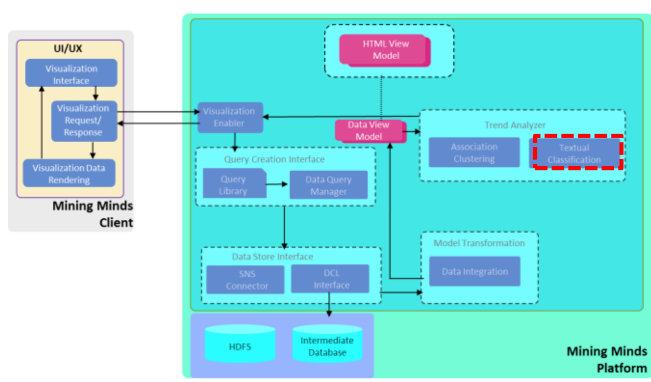


Methodology : Textual Classification

484

Use-case (SL2.5-UCS-13**):** *Classify Data*

- **Objective**
  - The data is to be classified so that further analytics can be performed on the expert query.
- **Methodology**
  - *Communication*: *Transformed data is further classified for analytics*
  - *Textual Classification(operation)*
    1. **Input**: receives *data from model transformation*
    2. **Processing**:
       - The textual data is extracted from the transformed data.
       - The textual data is associated with each other to be based on the query attribute and their corresponding facts.
    3. **Output**:
       a. The data is classified for the analytics.

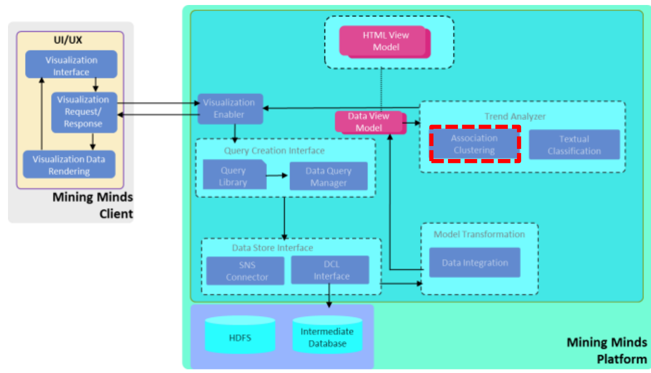


Methodology : Association Clustering

485

Use-case (SL2.5-UCS-14**):** *Analyze data*

- **Objective**
  - To prepare data for visualization and analytics
- **Methodology**
  - *Trend Analysis (operation)*
    1. **Input**: receives *data from model transformation*
    2. **Processing**:
       - The data classifier passes the data for association clustering.
       - The temporal and numerical data is analyzed for clustering.
       - The data is clustered into a group for graph plotting.
       - The textual data is associated with the numerical data to create analytics based on the textual attribute and their corresponding facts
    3. **Output**:
       a. The data is prepared for visualization and analytics.

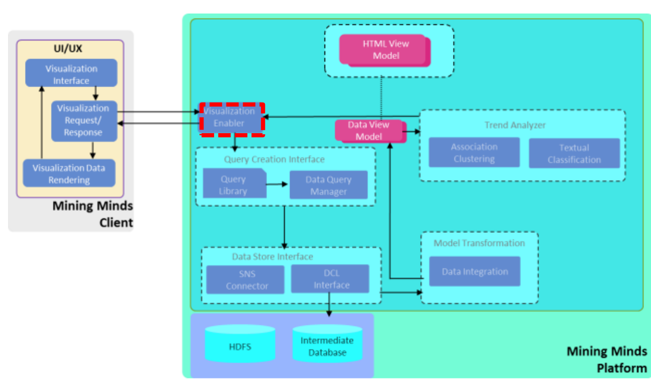


Methodology : Visualization Enabler

486

Use-case (SL2.5-UCS-15**):** *Display Analytics*

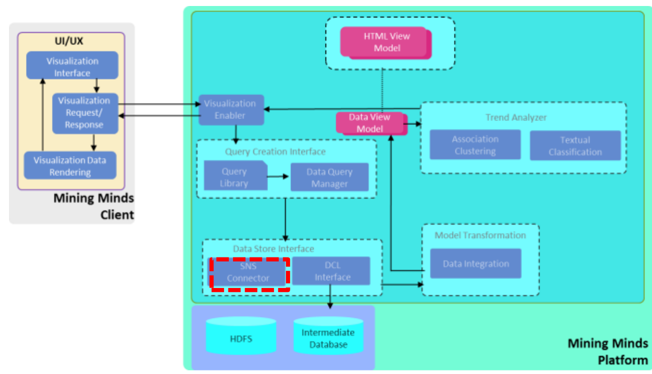
- **Objective**
  - The data is sent to visualization enabler distinguishable by their attributes and association.
- **Methodology**
  - *Display Analytics (operation)*
    1. **Input**: receives *data from model transformation*
    2. **Processing**:
       - The data is categorized according to the graph templates for visualization.
       - The scales are defined for the grouped data to be plotted on the coordinates.
       - The association text and the relevant facts about the data is also attached to the graph as analytics.
    3. **Output**:
       a. The analytics and relevant visualization is sent to the user interface



Methodology : Visualization Enabler

Use-case (SL2.5-UCS-16)**:** *Get Personalized SNS*

- **Objective**
  - ➢ The SNS trends are given to service curation layer (SCL) from tapacross.
- **Methodology**
  - ➢ *Get Personalized SNS(operation)*
    1. **Input**: SCL sends keywords to supporting layer
    2. **Processing**:
       - The descriptive analytics module receives keywords from SCL.
       - The keywords are appended in the webservice to get the trends from tapacross.
       - Tapacross sends the trends of the keywords in terms of frequency.
       - The trends and keywords are transformed into the JSON format agreed with SCL.
       - The json is forwarded to the SCL.
    3. **Output**:
       a. The SNS trends of keywords are sent back to SCL



Workflow of Descriptive Analytics

Graph Data

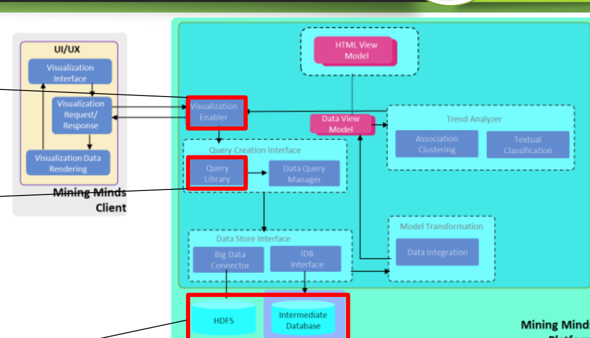
Id	X-axis	Y-axis	Multiple series	Graph
1	Yes	No	No	Pie
2	Yes	Yes	Yes	Line
3	Yes	Yes	Yes	Bar
4	Yes	Yes	No	Bubble

Query Library

Return Attribute	Conditional Attributes	Temporal Attribute
Food	Age	Days
Activity	Duration	DateTime
*.activity	uid	None
Activity	GPS/Location	Days

Data Store

uid	ActId	DateTime	Activity	Duration	accX	accY	accZ
2	2	20/08/1513: 00:03	Walking	5Min	1213.23 1123	1213.23 1123	1213.23 1123
29	2	20/08/1510: 00:03	Walking	45m24s	14334.1 23	14334.1 23	14334.1 23
30	4	20/08/1511: 00:03	Sitting	2hr	1232	1232	1232
23	2	21/08/1518: 00:03	Walking	10min	1213.23 1123	1213.23 1123	1213.23 1123
32	4	21/08/1512: 00:03	Sitting	2m24s	14334.1 23	14334.1 23	14334.1 23
30	3	21/08/1513: 00:03	running	4m	1232.84 1	1232.84 1	1232.84 1



Workflow of Descriptive Analytics

489

Graph Data

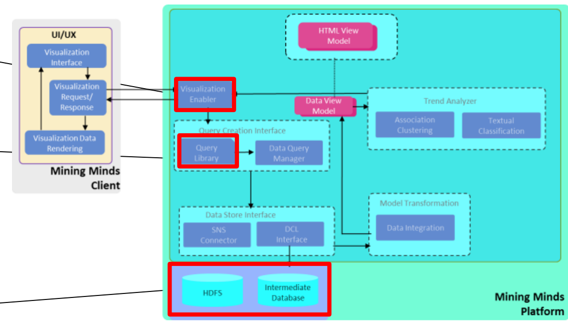
Id	X-axis	Y-axis	Multiple series	Graph
1	Yes	No	No	Pie
2	Yes	Yes	Yes	Line
3	Yes	Yes	Yes	Bar
4	Yes	Yes	No	Bubble

Query Library

Return Attribute	Conditional Attributes	Temporal Attribute
Food	Age	Days
Activity	Duration	DateTime
*.activity	uid	None
Activity	GPS/Location	Days

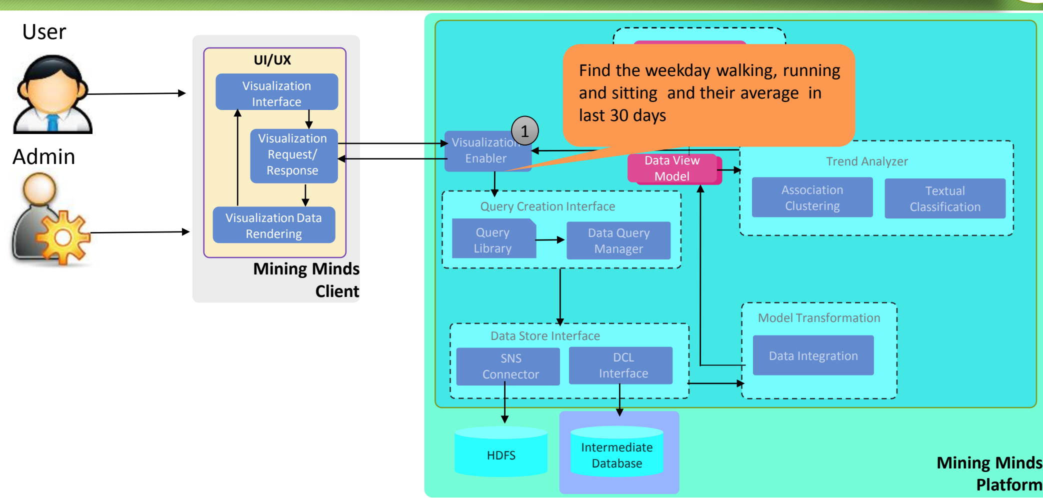
Data Store

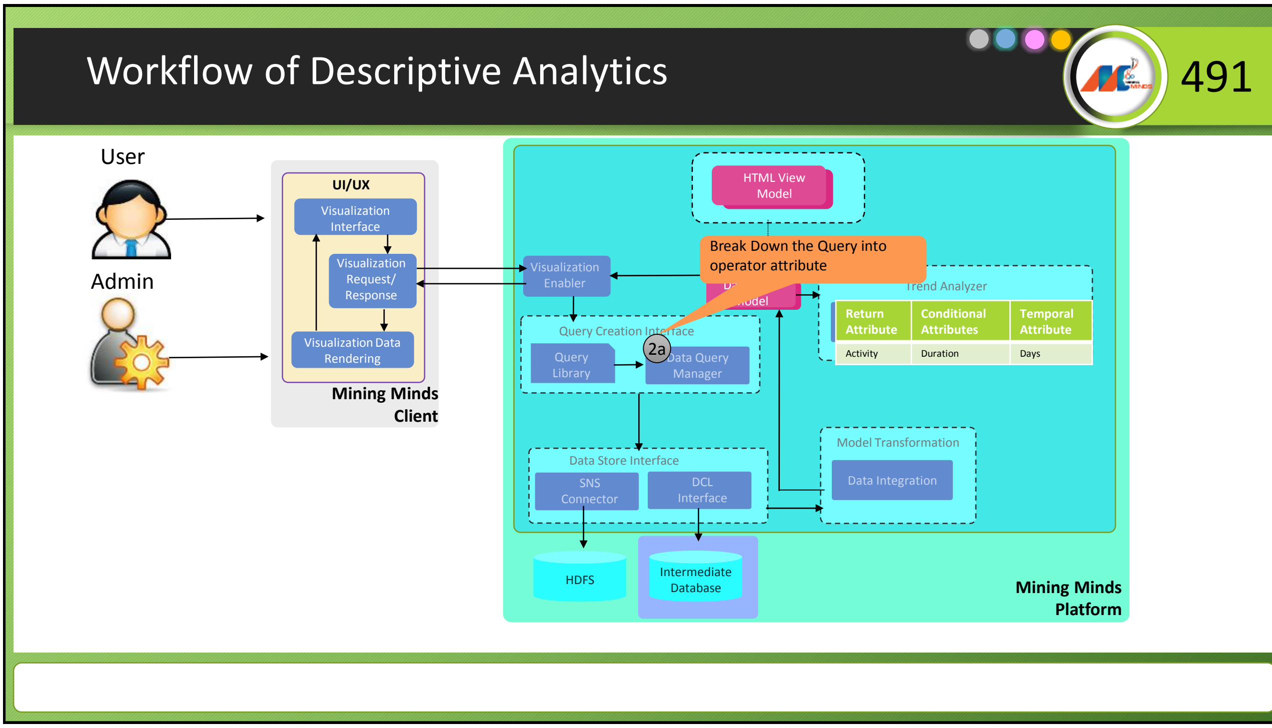
uid	ActId	DateTime	Activity	Duration	accX	accY	accZ
2	2	20/08/1513: 00:03	Walking	5Min	1213.23 1123	1213.23 1123	1213.23 1123
29	2	20/08/1510: 00:03	Walking	45m24s	14334.1 23	14334.1 23	14334.1 23
30	4	20/08/1511: 00:03	Sitting	2hr	1232	1232	1232
23	2	21/08/1518: 00:03	Walking	10min	1213.23 1123	1213.23 1123	1213.23 1123
32	4	21/08/1512: 00:03	Sitting	2m24s	14334.1 23	14334.1 23	14334.1 23
30	3	21/08/1513: 00:03	running	4m	1232.84 1	1232.84 1	1232.84 1

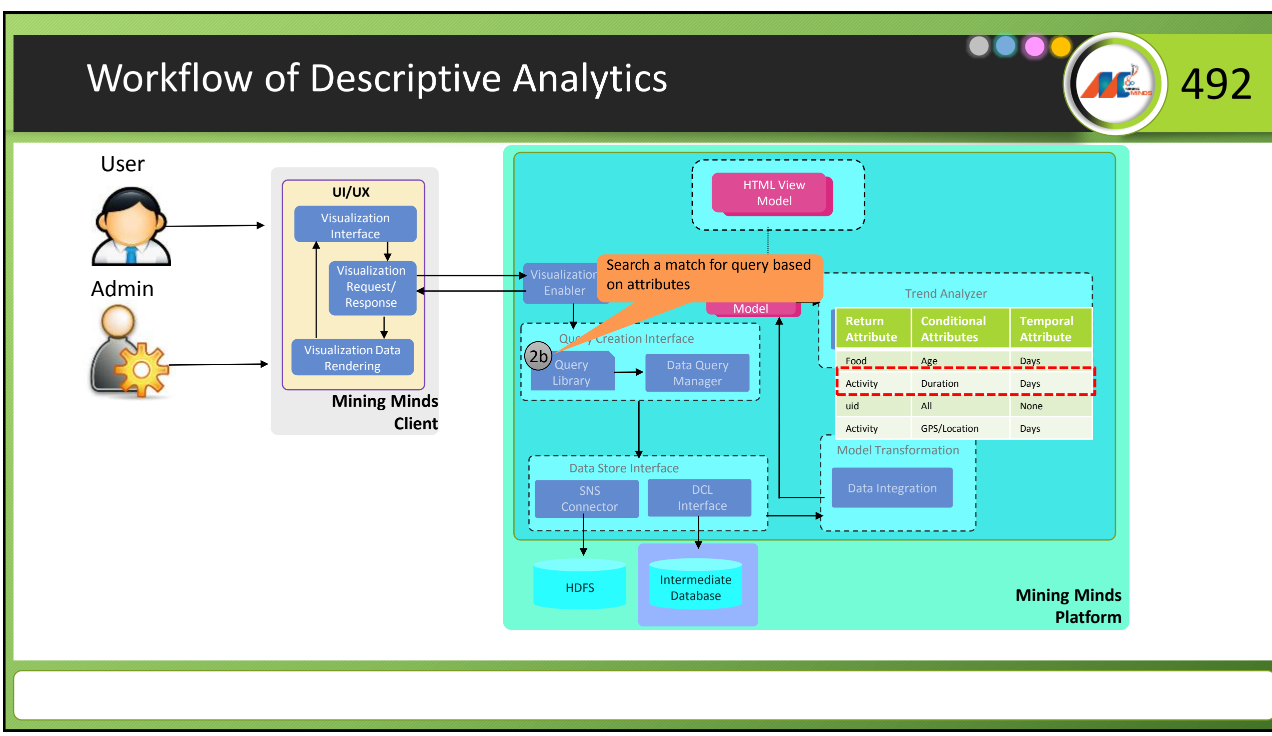


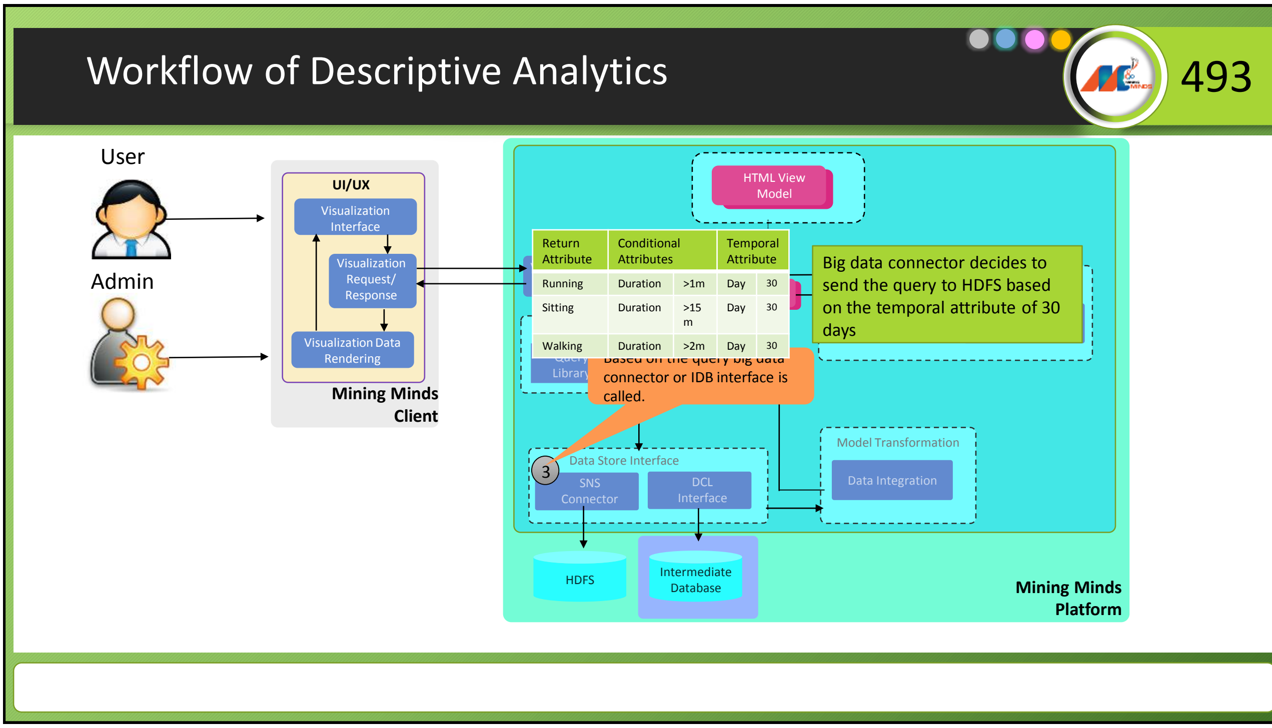
Workflow of Descriptive Analytics

490

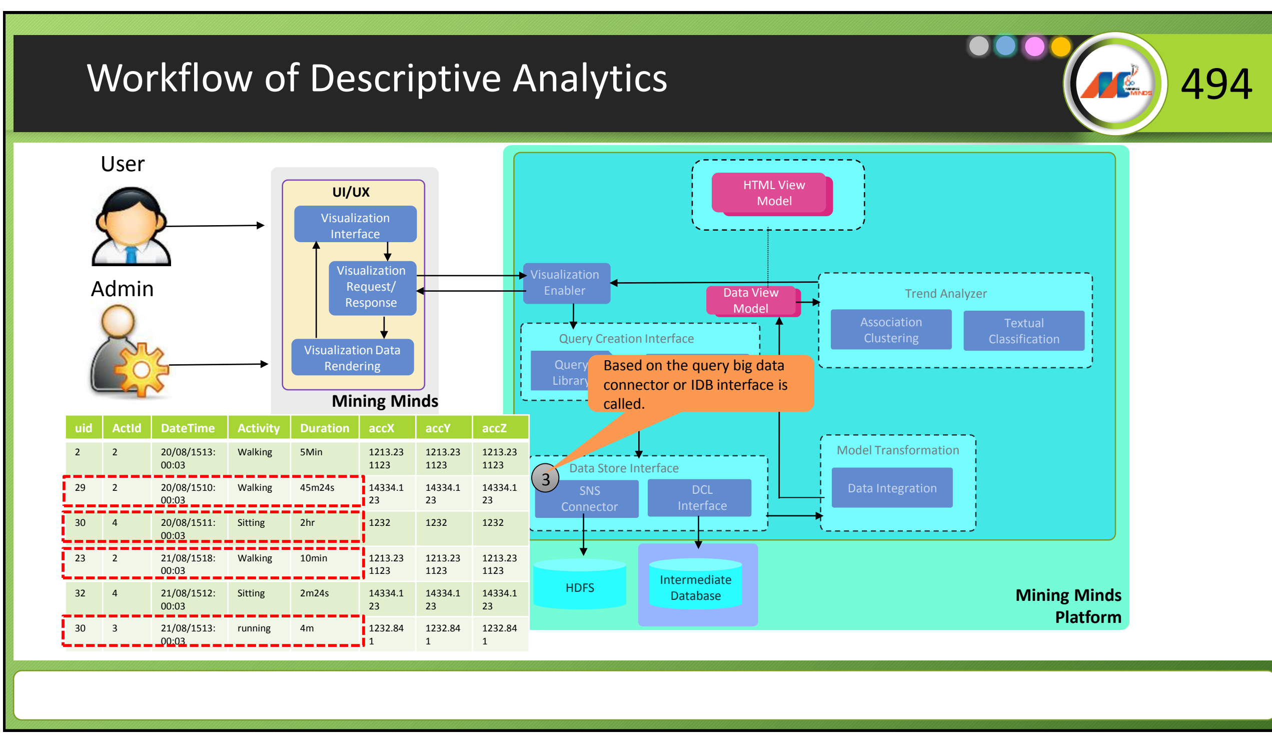



Workflow of Descriptive Analytics
491
User
Admin
UI/UX
Visualization Interface
Visualization Request/ Response
Visualization Data Rendering
Mining Minds Client
HTML View Model
Visualization Enabler
Break Down the Query into operator attribute
Trend Analyzer
Return Attribute
Conditional Attributes
Temporal Attribute
Activity
Duration
Days
Query Creation Interface
2a
Query Library
Data Query Manager
Model Transformation
Data Integration
Data Store Interface
SNS Connector
DCL Interface
HDFS
Intermediate Database
Mining Minds Platform

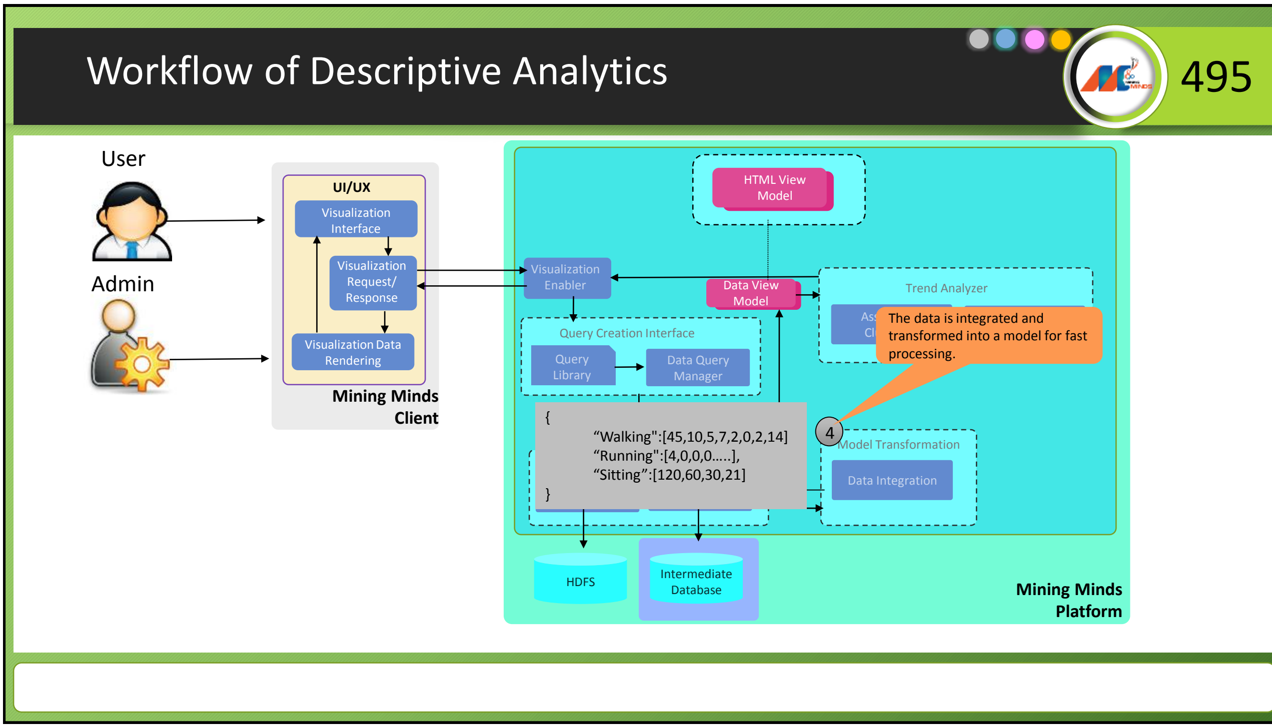

Workflow of Descriptive Analytics
492
User
Admin
UI/UX
Visualization Interface
Visualization Request/ Response
Visualization Data Rendering
Mining Minds Client
HTML View Model
Visualization Enabler
Search a match for query based on attributes
Model
Trend Analyzer
Return Attribute
Conditional Attributes
Temporal Attribute
Food
Age
Days
Activity
Duration
Days
uid
All
None
Activity
GPS/Location
Days
2b
Query Library
Data Query Manager
Model Transformation
Data Integration
Data Store Interface
SNS Connector
DCL Interface
HDFS
Intermediate Database
Mining Minds Platform

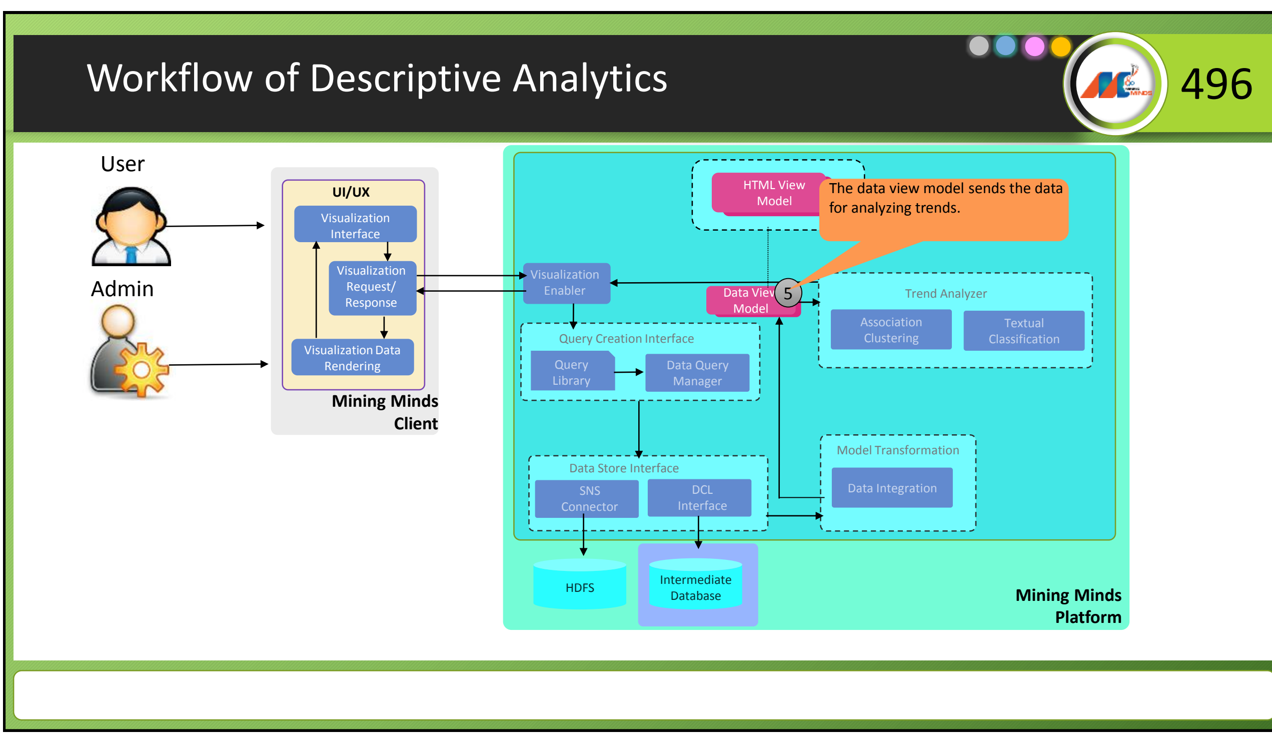


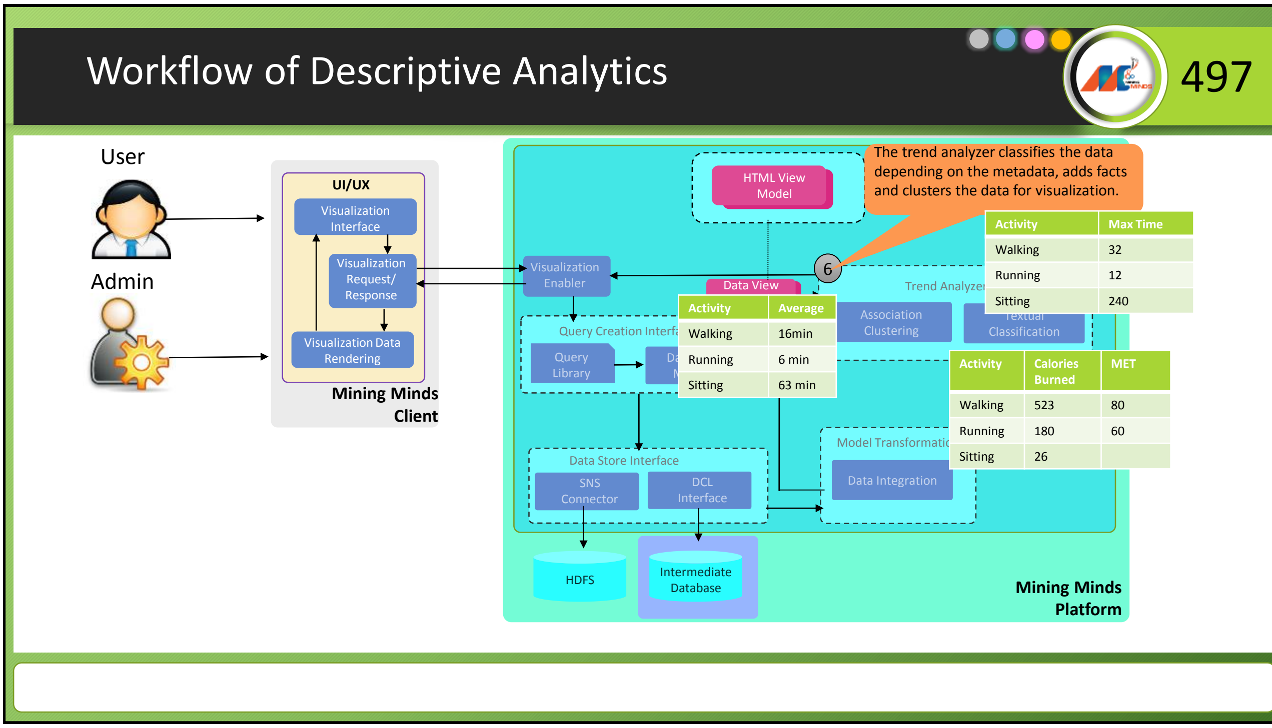
Return Attribute	Conditional Attributes		Temporal Attribute	
Running	Duration	>1m	Day	30
Sitting	Duration	>15 m	Day	30
Walking	Duration	>2m	Day	30

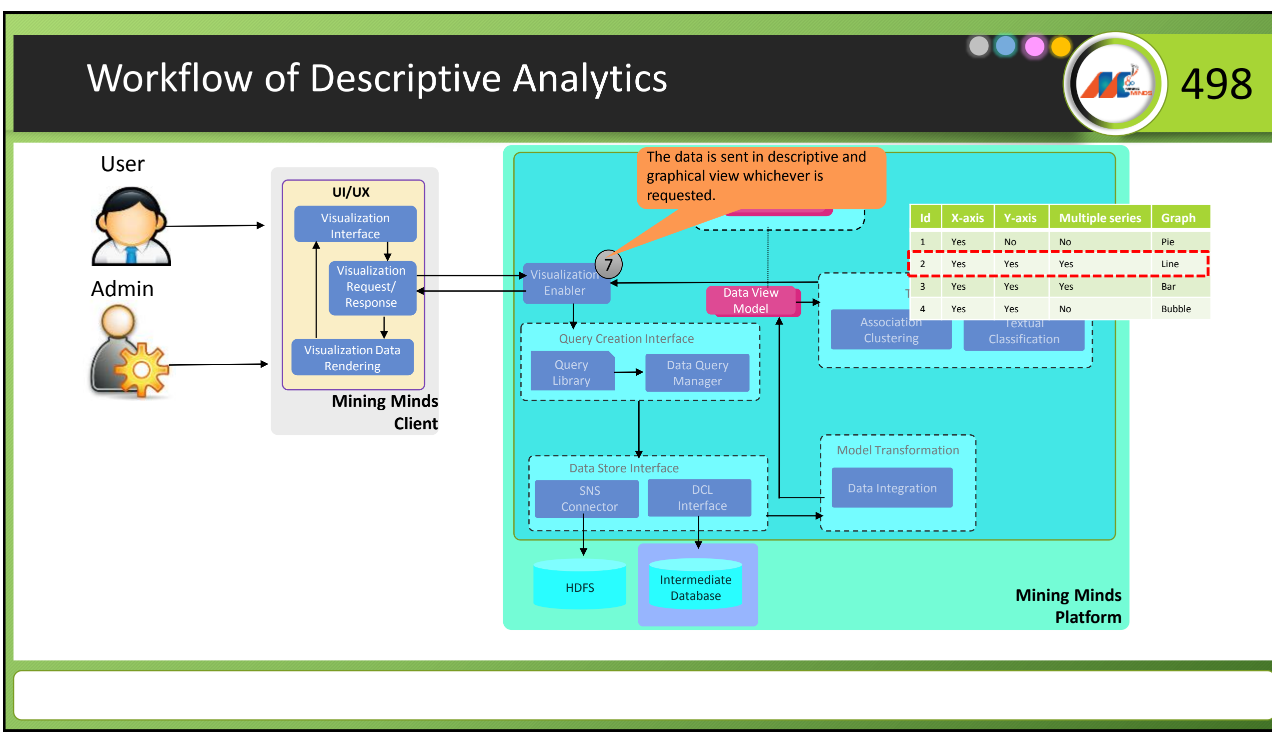


uid	ActId	DateTime	Activity	Duration	accX	accY	accZ
2	2	20/08/1513: 00:03	Walking	5Min	1213.23 1123	1213.23 1123	1213.23 1123
29	2	20/08/1510: 00:03	Walking	45m24s	14334.1 23	14334.1 23	14334.1 23
30	4	20/08/1511: 00:03	Sitting	2hr	1232	1232	1232
23	2	21/08/1518: 00:03	Walking	10min	1213.23 1123	1213.23 1123	1213.23 1123
32	4	21/08/1512: 00:03	Sitting	2m24s	14334.1 23	14334.1 23	14334.1 23
30	3	21/08/1513: 00:03	running	4m	1232.84 1	1232.84 1	1232.84 1


Workflow of Descriptive Analytics
495
User
Admin
UI/UX
Visualization Interface
Visualization Request/ Response
Visualization Data Rendering
Mining Minds Client
HTML View Model
Visualization Enabler
Data View Model
Trend Analyzer
The data is integrated and transformed into a model for fast processing.
Query Creation Interface
Query Library
Data Query Manager
{
"Walking":[45,10,5,7,2,0,2,14]
"Running":[4,0,0,0.....],
"Sitting":[120,60,30,21]
}
4
Model Transformation
Data Integration
HDFS
Intermediate Database
Mining Minds Platform

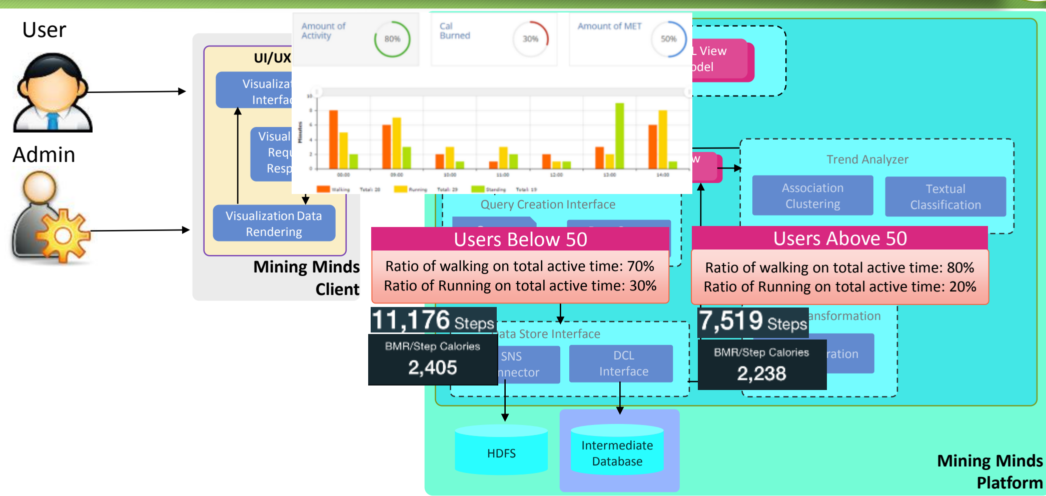

Workflow of Descriptive Analytics
496
User
Admin
UI/UX
Visualization Interface
Visualization Request/ Response
Visualization Data Rendering
Mining Minds Client
HTML View Model
The data view model sends the data for analyzing trends.
Visualization Enabler
Data View Model
5
Trend Analyzer
Association Clustering
Textual Classification
Query Creation Interface
Query Library
Data Query Manager
Data Store Interface
SNS Connector
DCL Interface
Model Transformation
Data Integration
HDFS
Intermediate Database
Mining Minds Platform


Workflow of Descriptive Analytics
497
User
Admin
UI/UX
Visualization Interface
Visualization Request/ Response
Visualization Data Rendering
Mining Minds Client
HTML View Model
The trend analyzer classifies the data depending on the metadata, adds facts and clusters the data for visualization.
6
Visualization Enabler
Data View
Trend Analyzer
Query Creation Interface
Query Library
Association Clustering
Textual Classification
Activity | Max Time
Walking | 32
Running | 12
Sitting | 240
Activity | Average
Walking | 16min
Running | 6 min
Sitting | 63 min
Activity | Calories Burned | MET
Walking | 523 | 80
Running | 180 | 60
Sitting | 26
Model Transformation
Data Integration
Data Store Interface
SNS Connector
DCL Interface
HDFS
Intermediate Database
Mining Minds Platform


Workflow of Descriptive Analytics
498
User
Admin
UI/UX
Visualization Interface
Visualization Request/ Response
Visualization Data Rendering
Mining Minds Client
The data is sent in descriptive and graphical view whichever is requested.
7
Visualization Enabler
Data View Model
Query Creation Interface
Query Library
Data Query Manager
Association Clustering
Textual Classification
Id | X-axis | Y-axis | Multiple series | Graph
1 | Yes | No | No | Pie
2 | Yes | Yes | Yes | Line
3 | Yes | Yes | Yes | Bar
4 | Yes | Yes | No | Bubble
Model Transformation
Data Integration
Data Store Interface
SNS Connector
DCL Interface
HDFS
Intermediate Database
Mining Minds Platform

Workflow of Descriptive Analytics

499



Tools /Technologies- Data Store Interface and Query Creation Interface

500

Options

- Apache Hive
- Pig Latin
- **MapReduce**
- MongoDB

Selected Tool

- Much Easier to Scale out.
- Compressed data could be queried
- Faster than most alternatives and mature

Tools /Technologies- Visualization, Model Transformation and Trend Analyzer

501

Options

- Java
- **PHP**

Hosting and Styling

- CSS
- Apache Webserver

Options

- **AmCharts Graph API**
- Anychart
- D3

- Fast, flexible, and free.
- Much faster to develop and PHP tends to have fewer issues when using shared hosting.

- Stable and been longer on marketing
- Theme Support

Contributions

502

- A query library for efficient data retrieval from big data repository.
- Model Transformation module to transform the data from HDFS and integrate it semantically
- Data Store interface for bringing unstructured (Big Data) and structured data (Intermediate).
- Trend analyzer to classify parameters and associate and cluster them for further insights
  - Focus on grouping and association techniques

MMv1.5 Implementation Challenges

503

Issues in MMv1.5	Solutions in MMv2
• Multiple Nested Queries to Data Curation Layer for visualization. • Performance bottleneck with monthly and weekly queries.	• SQL Server procedure modification in life log data. • A query library for efficient data retrieval for big data repository.

MMv2 Implementation Challenges

504

Issues in MMv1.5	Issues in MMv2	Solutions in MMv2.5
• Multiple Nested Queries to Data Curation Layer for visualization. • Performance bottleneck with monthly and weekly queries.	• The SNS results were usually inconclusive and not correctly mapped to user scenario • The overall execution times of analyzing the user timeline was very slow and error prone	• The relevant SNS results are returned to map to the user scenario through filtration • The execution time is reduced through the new introduced DCL web services and lighter visualization

Conclusion

- Big data analytics and visualization go hand in hand as we need an effective way to display the data intuitively for the users and the developers.
- The parameters vary from social attributes, temporal attributes and text based.
- We take the data and display visualization/analytics in to different users which vary from service users to domain experts.
- The data is associated and clustered to represent similar information and highlight outliers.

UI/UX Authoring Tool

Agenda

507

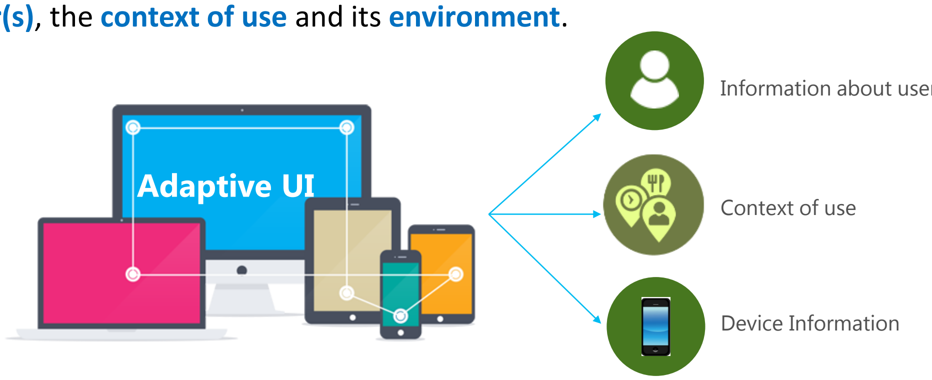
- Introduction
- Related Work
- Motivation
- Challenges and Solutions
- Proposed Architecture
- Workflow
- Tools and Methods
- Contributions
- Conclusion
- References

경희대학교 KYUNG HEE UNIVERSITY

Introduction [1/2]

508

An Adaptive User interface is a User Interface(UI) which **adapts**, its **layout** and **elements** to individual users based on information acquired about its **user(s)**, the **context of use** and its **environment**.

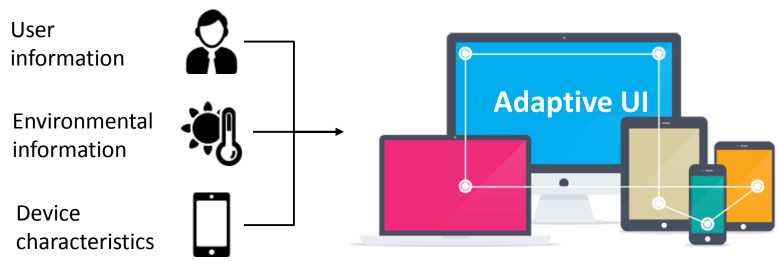


경희대학교 KYUNG HEE UNIVERSITY

Introduction [2/2]

509

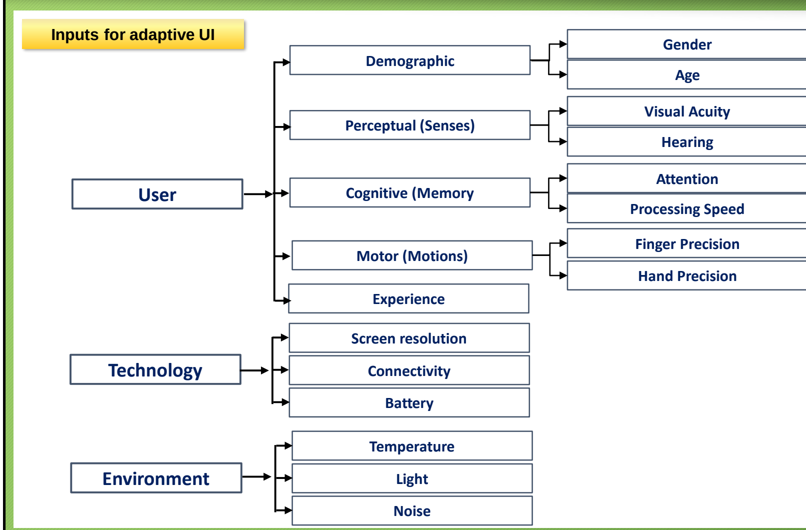
- In order to achieve a **satisfactory adaptation**, it is necessary to have several inputs such as

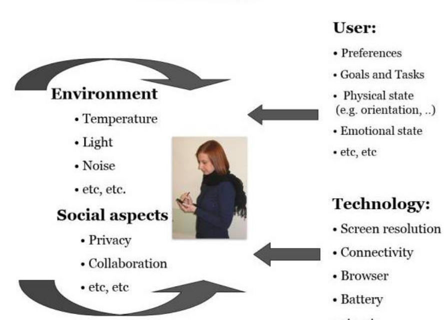


경희대학교 KYUNG HEE UNIVERSITY

Inputs for Adaptive User interface (AUI)

510





[https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/user-interface-design-adaptation]

경희대학교 KYUNG HEE UNIVERSITY

Taxonomy of UI adaptation

511

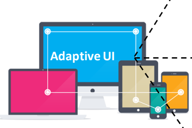
UI Adaptation

- **Adaptive UI** — The UI automatically adapted according to user preference and context of use
  - Granularity levels
    - Presentation ✔
    - Content ✔
    - Navigation
- **Adaptable UI** — The user configures manually the system UI according to his personal preferences

경희대학교 KYUNG HEE UNIVERSITY

Granularity levels- UI adaptation at different level

512



Navigation

- To guide a user to their specific goal within the system by altering the way the system is navigated based on certain factors of the user

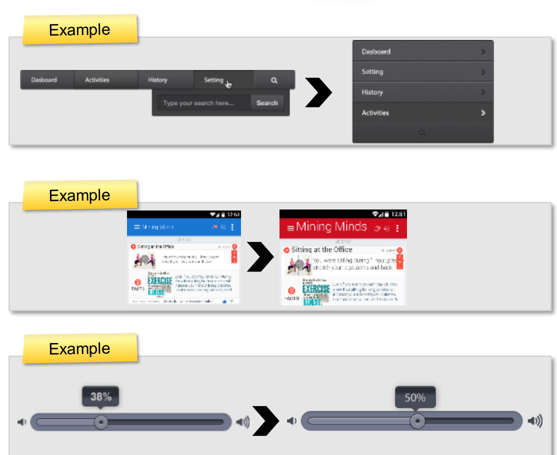
Presentation

ABC

- It concern the design of the interface, e.g. font size or color, layout of user interface elements

Content

- Adaptations that deal with the context of the situation occurred.



경희대학교 KYUNG HEE UNIVERSITY U2 system User & Usability system

Composition of adaptive systems

513

3 stages for adaptive UI

1. Data acquisition

Collection of data about the user

- Implicit Data Collection
- Explicit Data Collection

2. UX Metrics

UX

Creating or **updating** a user data using different techniques

- Performance Metrics
- Issue-Based Metrics
- Self-Reported Metrics
- Behavioral and Physiological Metrics

3. Adaptation Engine

Deciding how to adapt the system behavior based on data and adaptive rules

- Rule Based UI adaptive
- Semantic reasoner

- Environmental characteristics
- Task-oriented
- User characteristics
- Technical characteristics
- Questionnaires
- Rating system

Note: Initially, we start with minimal user data, residing in life-log, and create the rest of required data with the help of our own UX metrics collection techniques e.g. performance, self-reported , physiological metrics.

경희대학교 KYUNG HEE UNIVERSITY

Goal of Adaptation

514

- Easy, efficient and effective to use
- To make the complex system usable
- Present want a user want to see
- Speed up and simplify use
- A UI that fit to heterogeneous group of user
- A UI that considers increasing the user experience

경희대학교 KYUNG HEE UNIVERSITY

Overview of existing adaptive systems

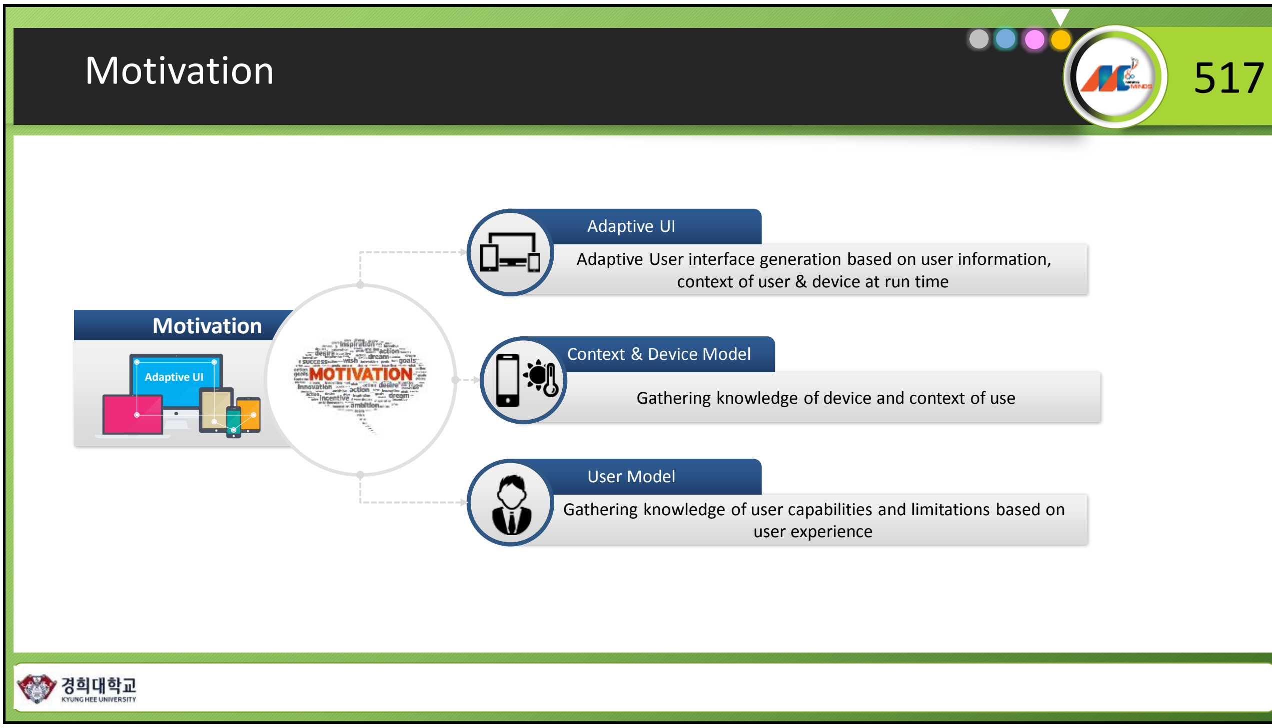
515

Existing Systems	Descriptions	Pros:	Cons:
Doppelgänger [20]	It was intended to produce a personalized, printed Newspaper for the user	• Sharing of user information among several applications. • Diverse types of sensors contribute to an extensive user model. • Unobtrusive user modelling.	• No systematic feedback mechanism
Flexcel [21]	It enhances Microsoft Excel by an adaptive User Interface	• Users have control over their own user profiles	• Some of the user dialogues for adaptability seem very complex
Lumière-Project[22]	It led to the later MS-Office assistant	• It combined the temporal reasoning and Bayesian user models in order to manage the uncertainty of recognizing user goals from a stream of user actions over time.	• It only focus on recognizing user goals in order to provide appropriate
Lifestyle Finder[25]	It gives the user suggestions for interesting websites	• User profiling and clustering is based on publically available demographic mass data	• Adaptation covers only the selection of content • User modelling covers aspects such as purchasing history, lifestyle characteristics and survey responses
Supple[26]	It is an application that adapts the display of objects considering window size and user preferences	• Run-time rendering of the user interface • Information about the user is collected by analyzing user tracking	• Adaptation focuses on layout and selection of appropriate controls and display elements • It does not address accessibility issues • It does not provide an authoring tool
MYUI [19]	It generates individualized user interfaces and performs adaptations to diverse user needs, devices and environmental conditions during run time.	• Toolkit: supports industrial developers and designers to easily create self-adaptive applications • Explicit and implicit data collection about user for user modeling • Run-time rendering of user interface.	• No feedback functionality • Manual setting for platform device category

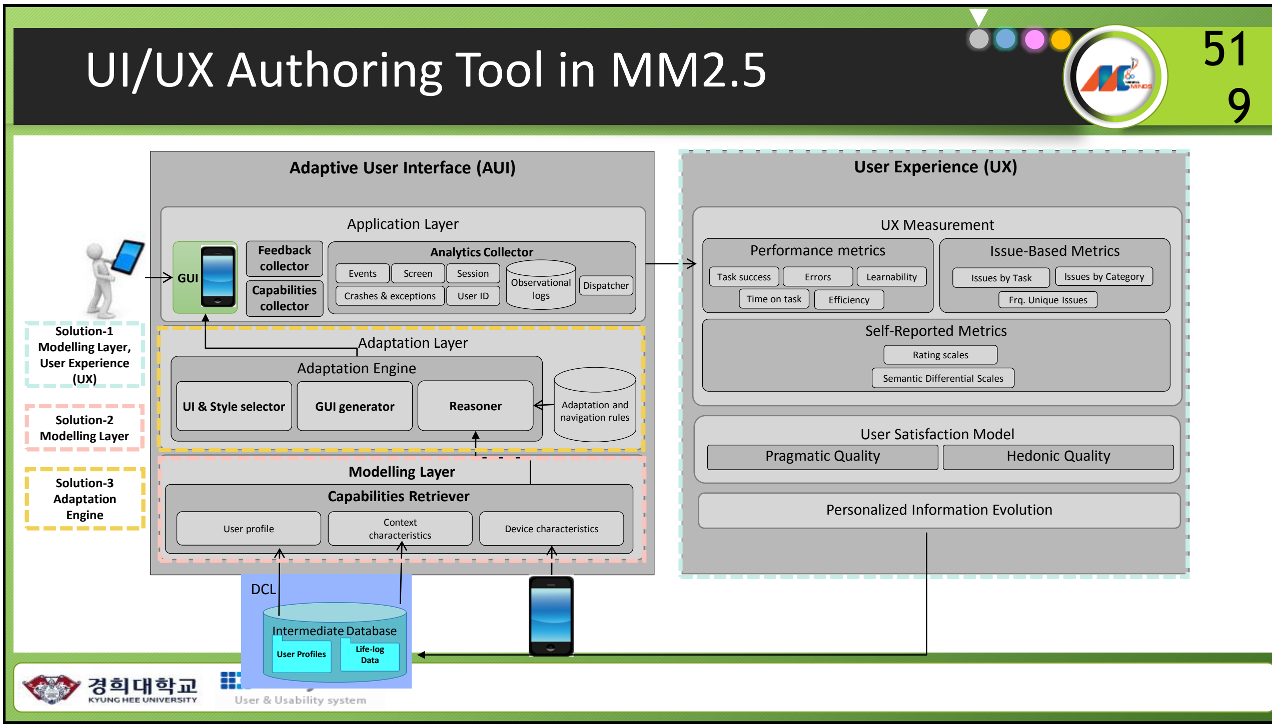
Limitations of existing Work

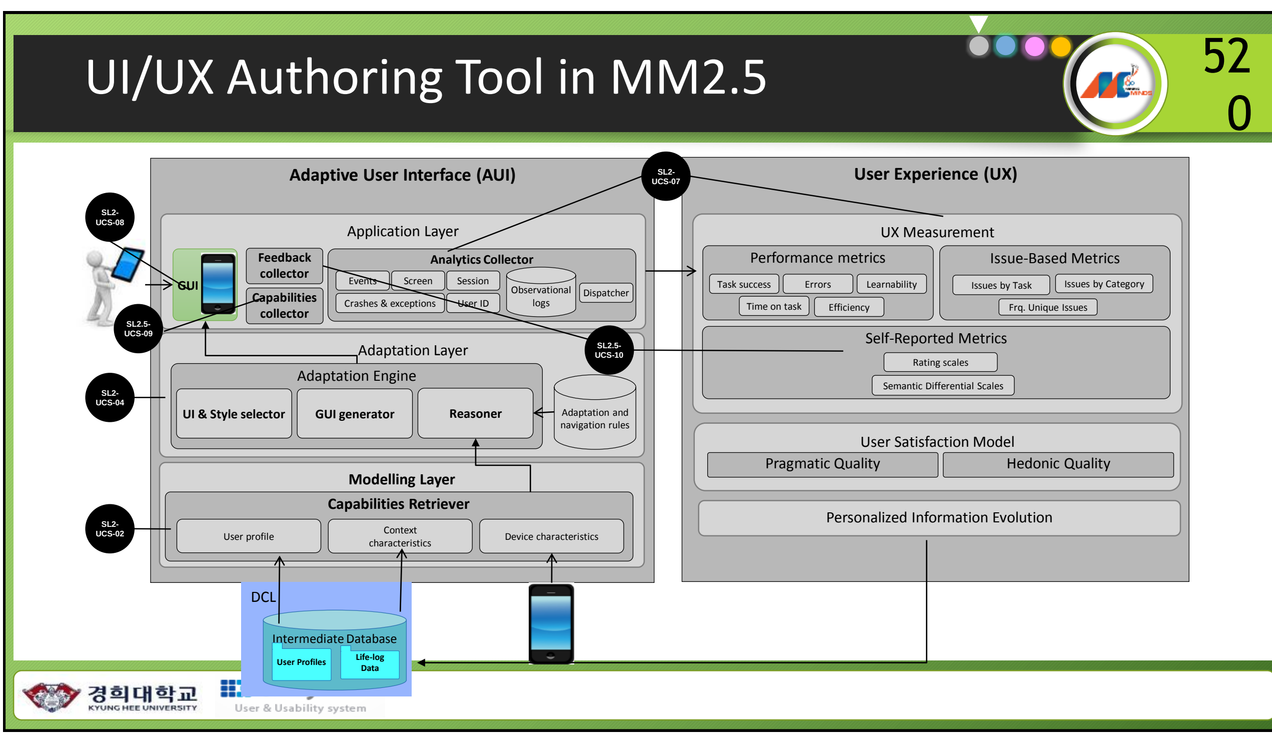
516

- The existing user models are not comprehensive
  - Lacks behavioral & physiological measurement for adaptive UI
  - Environmental variables
- Lack addressing accessibility issues
- No feedback functionality
- Lack of user experience for adaptive UI


Motivation
517
Motivation
Adaptive UI
MOTIVATION
Adaptive UI
Adaptive User interface generation based on user information, context of user & device at run time
Context & Device Model
Gathering knowledge of device and context of use
User Model
Gathering knowledge of user capabilities and limitations based on user experience
경희대학교
KYUNG HEE UNIVERSITY

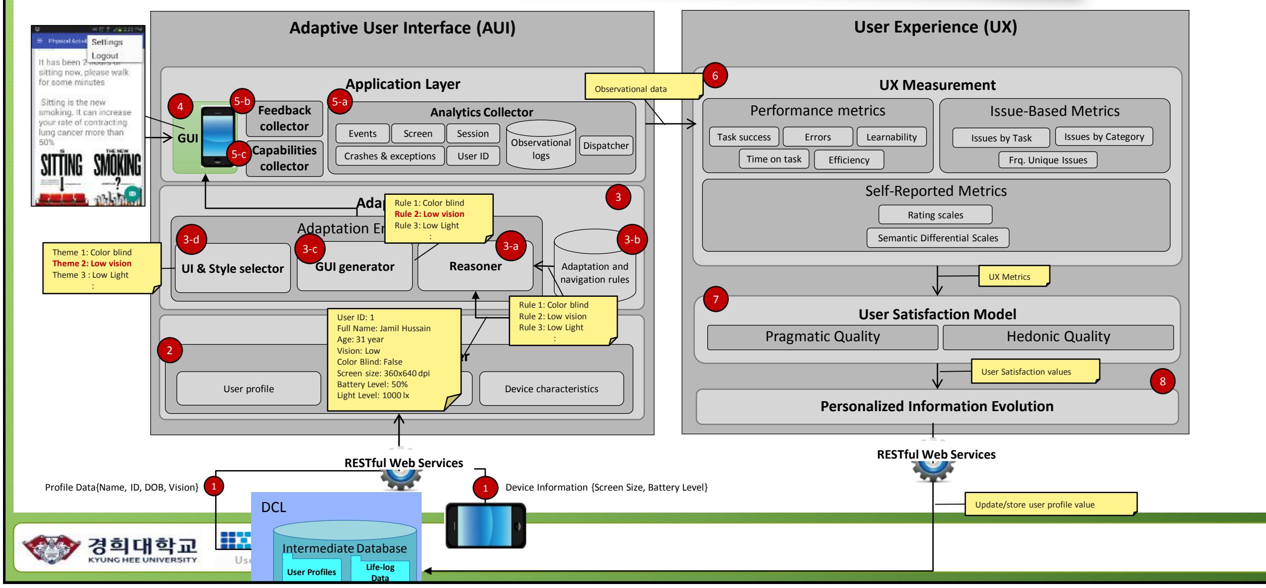

Challenges and Solutions
518
Motivation
Challenges
Solution
MOTIVATION
Adaptive User interface generation based on user information, context of user & device
Real time user interface rendering based on user , context and device information
S3
Adaptation Engine
Gathering knowledge of device and context of use
Gathering correct context and device information
S2
Modelling Layer
Gathering knowledge of user capabilities and limitations based on user experience
User experience quantification and comprehensive user model creation
S1
Modelling Layer, User Experience (UX)
경희대학교
KYUNG HEE UNIVERSITY


UI/UX Authoring Tool in MM2.5
51
9
Adaptive User Interface (AUI)
Application Layer
GUI
Feedback collector
Capabilities collector
Analytics Collector
Events
Screen
Session
Crashes & exceptions
User ID
Observational logs
Dispatcher
Adaptation Layer
Adaptation Engine
UI & Style selector
GUI generator
Reasoner
Adaptation and navigation rules
Modelling Layer
Capabilities Retriever
User profile
Context characteristics
Device characteristics
Solution-1 Modelling Layer, User Experience (UX)
Solution-2 Modelling Layer
Solution-3 Adaptation Engine
DCL
Intermediate Database
User Profiles
Life-log Data
User Experience (UX)
UX Measurement
Performance metrics
Task success
Errors
Learnability
Time on task
Efficiency
Issue-Based Metrics
Issues by Task
Issues by Category
Frq. Unique Issues
Self-Reported Metrics
Rating scales
Semantic Differential Scales
User Satisfaction Model
Pragmatic Quality
Hedonic Quality
Personalized Information Evolution
경희대학교
KYUNG HEE UNIVERSITY
User & Usability system


UI/UX Authoring Tool in MM2.5
52
0
Adaptive User Interface (AUI)
SL2-UCS-07
SL2-UCS-08
SL2.5-UCS-09
SL2-UCS-04
SL2-UCS-02
SL2.5-UCS-10
Application Layer
GUI
Feedback collector
Capabilities collector
Analytics Collector
Events
Screen
Session
Crashes & exceptions
User ID
Observational logs
Dispatcher
Adaptation Layer
Adaptation Engine
UI & Style selector
GUI generator
Reasoner
Adaptation and navigation rules
Modelling Layer
Capabilities Retriever
User profile
Context characteristics
Device characteristics
DCL
Intermediate Database
User Profiles
Life-log Data
User Experience (UX)
UX Measurement
Performance metrics
Task success
Errors
Learnability
Time on task
Efficiency
Issue-Based Metrics
Issues by Task
Issues by Category
Frq. Unique Issues
Self-Reported Metrics
Rating scales
Semantic Differential Scales
User Satisfaction Model
Pragmatic Quality
Hedonic Quality
Personalized Information Evolution
경희대학교
KYUNG HEE UNIVERSITY
User & Usability system

UI/UX Authoring Tool – Communication View

521



UI/UX Authoring Tool – Overall Concept

522

UX Measurement
- It deals with metrics that reveal something about the user experience- about the personal experience/ interaction of user with product or system like
  - ✓ effectiveness (being able to complete a task),
  - ✓ efficiency (the amount of effort required to complete the task),
  - ✓ satisfaction (the degree to which the user was happy with his or her experience while performing the task)

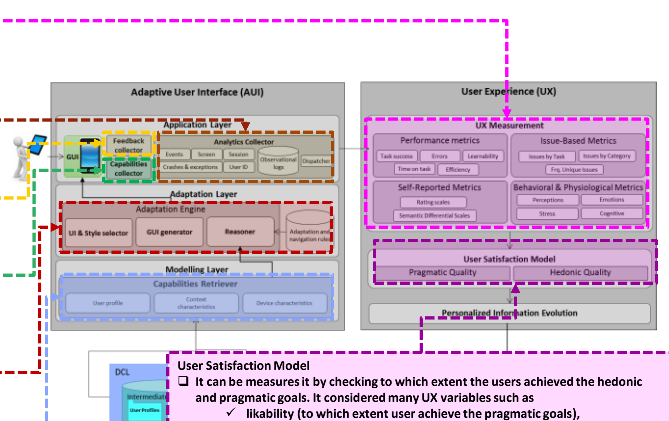
Analytics Collector
- User interaction behavioral data collection
  - ✓ To measure the Pragmatic qualities

Feedback Collector
- Responsible for display the prompt feedback form based on implicit collected data then the user feedback data can be send to self-reporting component to in order to find out (1) the hedonic and (2) pragmatic quality

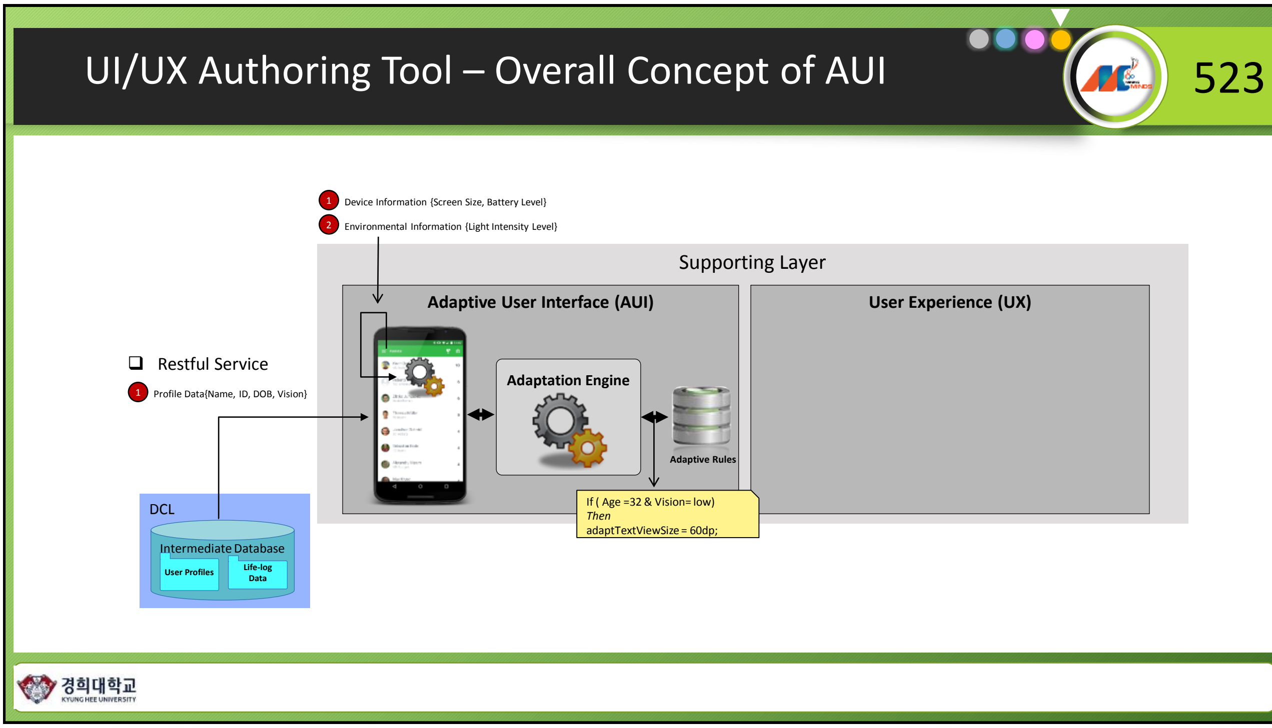
Capabilities collector
- Responsible to collect information about user perception, cognitive and motor skill using different game/ techniques

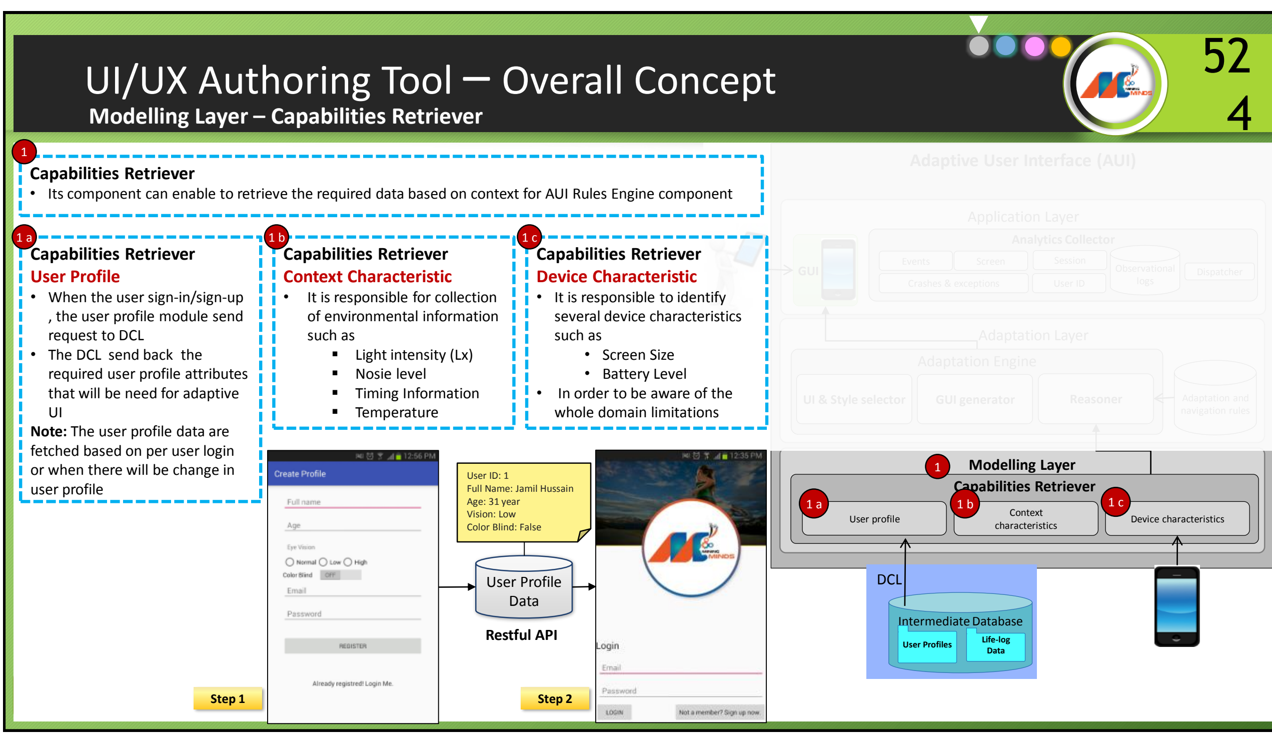
Adaptation Engine
- Reasonar
  - ✓ Input data
  - ✓ Rules
  - ✓ Fire rules based on collected data
- GUI Generator render UI based on fired rules

Capabilities Retriever
- It collect data for adaptation engine such as
  - ✓ User Profile data
  - ✓ Environmental data
  - ✓ Device information



User Satisfaction Model
- It can be measures it by checking to which extent the users achieved the hedonic and pragmatic goals. It considered many UX variables such as
  - ✓ likability (to which extent user achieve the pragmatic goals),
  - ✓ pleasure (to which extent user achieve the hedonic goals),
  - ✓ comfort (to which extent user feel comfort), and
  - ✓ trust (to which extent user satisfy with overall system).


UI/UX Authoring Tool – Overall Concept of AUI
523
1 Device Information {Screen Size, Battery Level}
2 Environmental Information {Light Intensity Level}
Supporting Layer
Adaptive User Interface (AUI)
User Experience (UX)
Restful Service
1 Profile Data{Name, ID, DOB, Vision}
Adaptation Engine
Adaptive Rules
If (Age =32 & Vision= low)
Then
adaptTextViewSize = 60dp;
DCL
Intermediate Database
User Profiles
Life-log Data
경희대학교
KYUNG HEE UNIVERSITY


UI/UX Authoring Tool – Overall Concept
Modelling Layer – Capabilities Retriever
524
1
Capabilities Retriever
• Its component can enable to retrieve the required data based on context for AUI Rules Engine component
1 a
Capabilities Retriever
User Profile
• When the user sign-in/sign-up , the user profile module send request to DCL
• The DCL send back the required user profile attributes that will be need for adaptive UI
Note: The user profile data are fetched based on per user login or when there will be change in user profile
1 b
Capabilities Retriever
Context Characteristic
• It is responsible for collection of environmental information such as
▪ Light intensity (Lx)
▪ Nosie level
▪ Timing Information
▪ Temperature
1 c
Capabilities Retriever
Device Characteristic
• It is responsible to identify several device characteristics such as
• Screen Size
• Battery Level
• In order to be aware of the whole domain limitations
Create Profile
Full name
Age
Eye Vision
Normal Low High
Color Blind
Email
Password
REGISTER
Already registred! Login Me.
Step 1
User ID: 1
Full Name: Jamil Hussain
Age: 31 year
Vision: Low
Color Blind: False
User Profile Data
Restful API
Step 2
Login
Email
Password
LOGIN
Not a member? Sign up now
Adaptive User Interface (AUI)
1 Modelling Layer
Capabilities Retriever
1 a User profile
1 b Context characteristics
1 c Device characteristics
DCL
Intermediate Database
User Profiles
Life-log Data

UI/UX Authoring Tool – Overall Concept

Modelling Layer – Capabilities Retriever

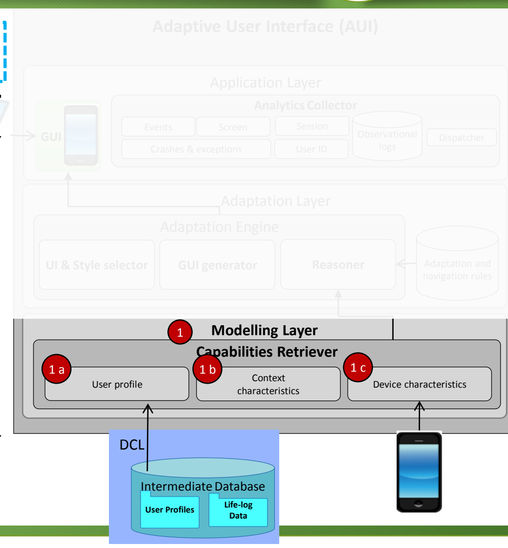
525

Capabilities Retriever

- Its component can enable to retrieve the required data based on context for AUI Rules Engine component

Algorithm 1 Capabilities Retriever

```
function retrieve capabilities(x, y , z)
    Input: Three kind of information where x is user profile information , y is environmental information,
    and z is device information
    Output: The required capabilities data are retrieve for adaptive rule engine
If User is not valid Then
  | return(x) ← error
else
  | set(x) ← (u_ID, name, age impairment, & disability, …. )
  | get(light_level) ← lux
  | get(noise_level) ← dBA
  | set(y) ← (light_level, noise_level )
  | get(screen_size) ← dp
  return (x, y, z)
end
```



UI/UX Authoring Tool – Overall Concept

Adaptation Layer – Adaptation Engine

526

Adaptation Engine

- It can perform reasoning based on pre-defined adaption and navigations rules. It recommend the user interface components and its styles to render the adaptive UI

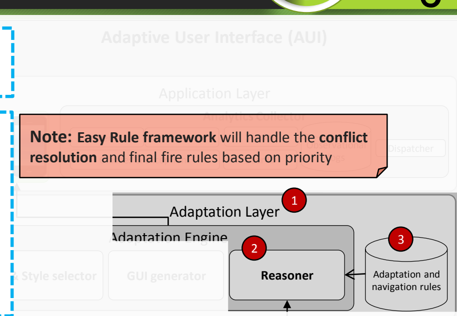
Adaptation Engine
Adaptation & navigation rules

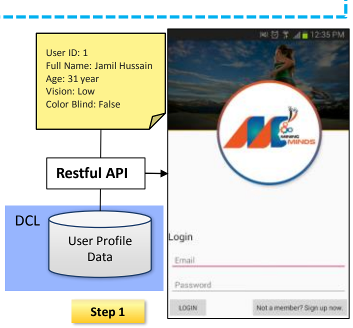
- The adaptation rules are classified according to the target user disabilities, environmental conditions and as well as accessibility rule to in order to increase positive user experience (UX)

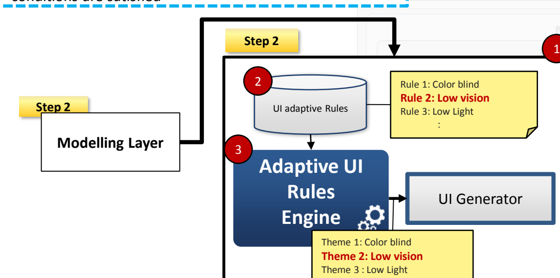
Adaptation Engine
Reasoner (Rule Base)

Easy Rules is a simple yet powerful Java rules engine framework

- **Name**: a unique rule name within a rules namespace
- **Description**: a brief description of the rule
- **Priority**: rule priority regarding to other rules
- **Conditions**: set of conditions that should be satisfied to apply the rule Actions: set of actions to perform when conditions are satisfied

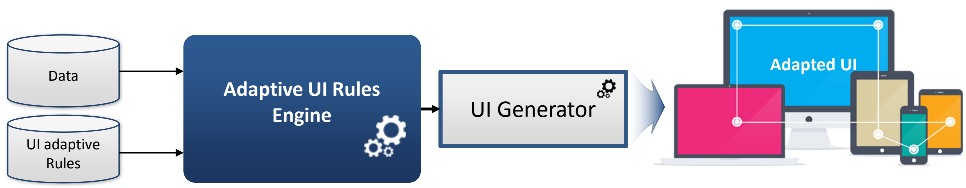






UI/UX Authoring Tool – Overall Concept

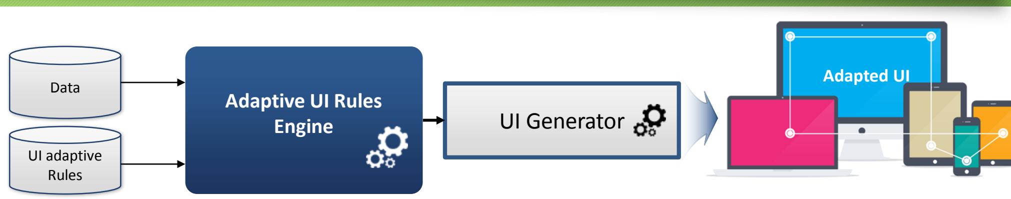
Adaptation Layer – Adaptation Engine

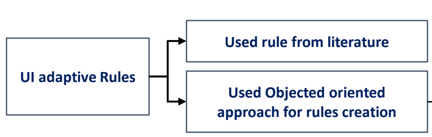


- ❑ Adaptive UI Rule Engine use the user, environmental, technology information and UI adaptive rules to fire the UI rules
- ❑ UI generator use the fire UI rule in order to generates the final adapted UI

UI/UX Authoring Tool – Overall Concept

Adaptation Layer – Adaptation Engine





Easy Rules : Easy Rules is a simple yet powerful Java rules engine framework

First, define your rule..

```
@Rule (name = "my awesome rule" )
public class MyRule {
  @Condition
  public boolean when() {
    return true;
  }
  @Action
  public void then() {
    System.out.println("Easy Rules rocks!");
  }
}
```

Then, fire it!

```
public class Test {
  public static void main(String[] args) {
    // create a rules engine
    RulesEngine rulesEngine =
              aNewRulesEngine().build();
    //register the rule
    rulesEngine.registerRule(new MyRule());
    //fire rules
    rulesEngine.fireRules();
  }
}
```

Note: In MMv2.5 we will created rule Authoring tool, so that UX expert will be able to create new rule, update and delete the existing rules.

http://www.easyrules.org/index.html

UI/UX Authoring Tool – Overall Concept

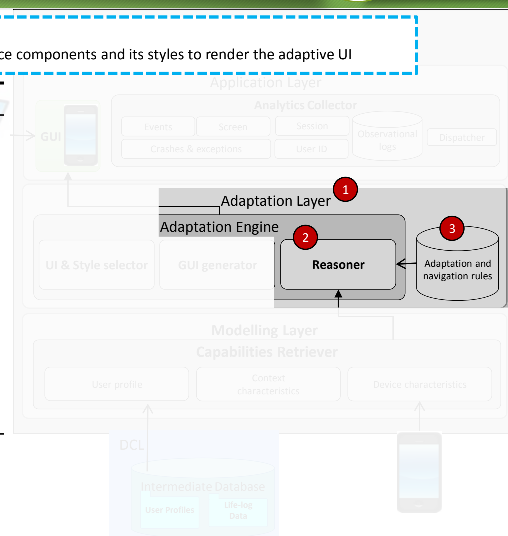
Adaptation Layer – Adaptation Engine

1 **Adaptation Engine**

- It can perform reasoning based on pre-defined adaption and navigations rules. It recommend the user interface components and its styles to render the adaptive UI

Algorithm 2 Adaptation Engine

```
Function reasoning(x, y )
      Input: Two kind of information where x is retrieved information, y is adaptation rule sets
      Output: Fire the final rule so that GUI generator can render the GUI based on that rule
If User is not valid Then
   | return(x) ← error
else
   | set(x) ← (user profile, environmental information, device information)
   | set(y) ← load all rules from adaption and navigation rules from the local DB
   | foreach input ∊ y do
   |    | registered all rules to rule engine
   | end
   | fire_rule ← fire rules based on matching
   | return (fire_rule)
end
```



UI/UX Authoring Tool – Overall Concept

Adaptation Layer – Adaptation Engine

1 **Adaptation Engine**

- It can perform reasoning based on pre-defined adaption and navigations rules. It recommend the user interface components and its styles to render the adaptive UI

2 **Adaptation Engine**

GUI Generator

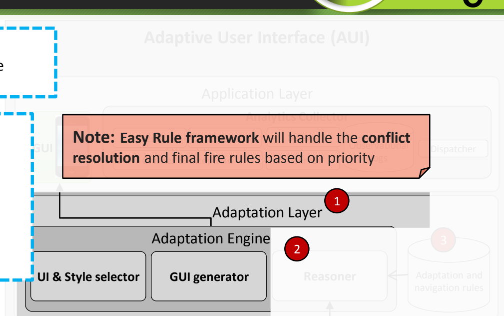
- GUI received the fire rules
- GUI generator

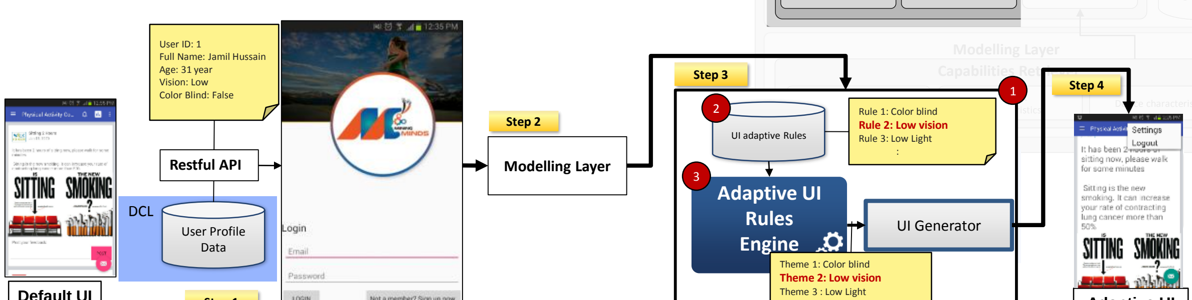
3 **Adaptation Engine**

UI & style selector

Easy Rules is a simple yet powerful Java rules engine framework

- **Name**: a unique rule name within a rules namespace
- **Description**: a brief description of the rule
- **Priority**: rule priority regarding to other rules





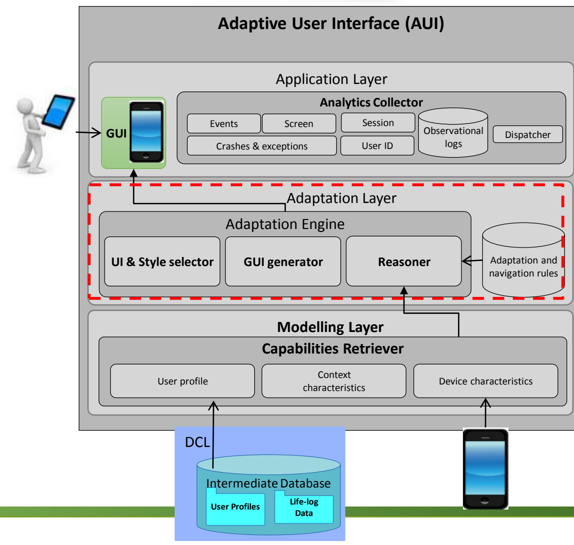
Methodology : Adaptation Layer

531

Methodology Description

SL2.5-SUC-01: *Monitor Configuration for target Variables*

- **Objective**
  - To lead the dynamic adaptation of the elements presented in the user interface
- **Methodology**
  - *Communication*: adaptation and modelling layer
  - **Semantic Modeller**
    1. **Input**: receives *request from the modelling layer*
    2. **Processing**:
       - User request for personalize user interface to UI/UX.
       - UI/UX send request to adaptation engine for generating the personalized user interface based on user, context and device information.
       - Adaptation engine perform reasoning based on pre-defined adaption and navigations rules.
       - Reasoner recommend the user interface components and its styles to generate the UI.
       - Then personalized generated UI is displayed to end user.
    3. **Output:**
       - Adaptive UI rendered/generated based on collected information



경희대학교 KYUNG HEE UNIVERSITY U2 system User & Usability system

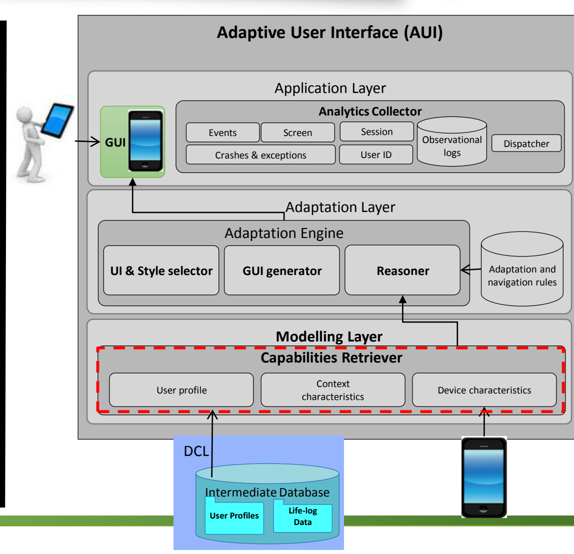
Methodology : Capabilities Retriever

532

Methodology Description

Use-case (SL2.5-UCS-02): *Retrieve capabilities for user interface adaptation*

- **Objective**
  - The Capabilities retriever is a software module that allows the system gathering non physiological capabilities of the user, collect different variables of the current environment situation, and identify several device characteristics in order to be aware of the whole domain limitations
- **Methodology**
  - *Communication: request to DCL 2 using restful API*
  - **Capabilities Retriever**
    1. **Input**: receives *request from the application layer*
    2. **Processing**:
       - SL generates request for user, device, and context information collection from DCL 2
       - DCL 2 provide the requested data based on User ID
    3. **Output:**
       - The required capabilities data are retrieve for mapping data on the AdaptiveUI Ontology
       - forwards loaded data to *Semantic Modeller*

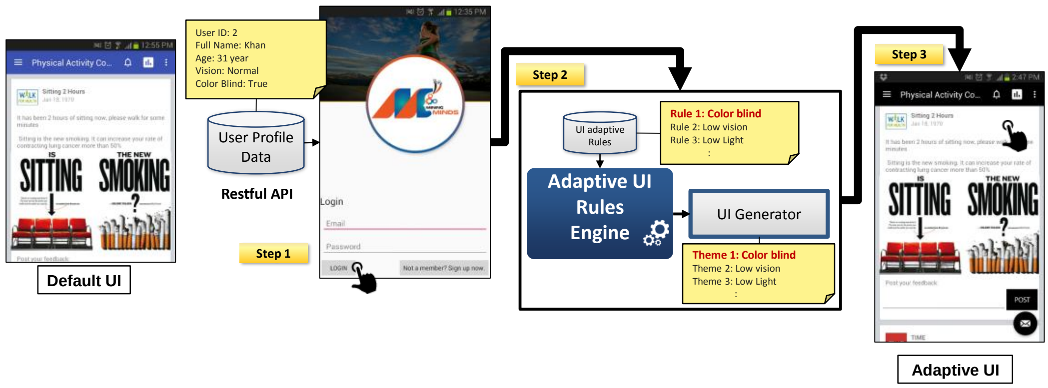


Variables fetch frequency

- ❑ The user Profile data will be fetched one time unless user profile is not changed
- ❑ The context and environmental information will be fetched after fixed interval of time (e.g. 1 min)

App Demo Scenario (2): Color Blind

533

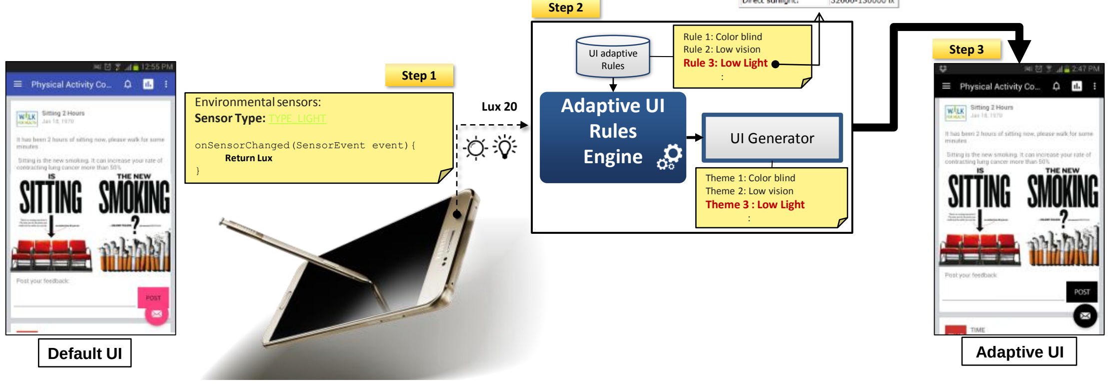


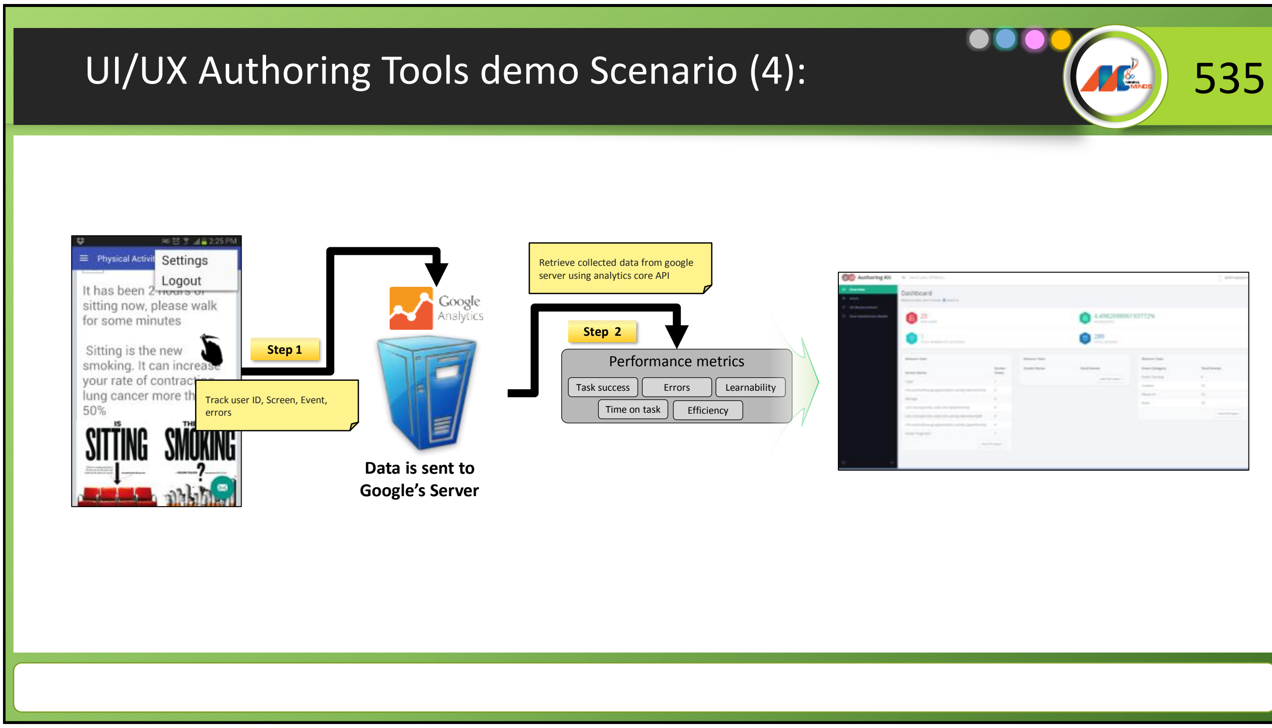
App Demo Scenario (3): Adaptation based on environmental variables

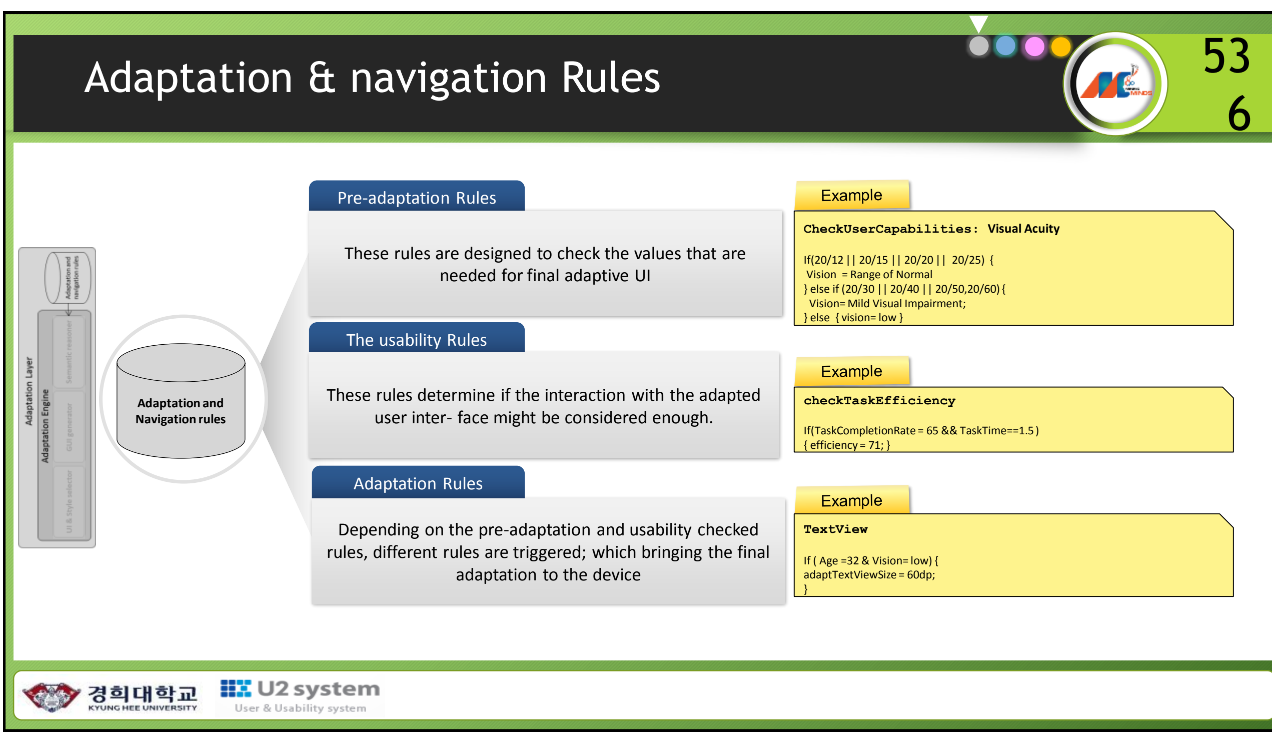
534

Example lux values

Clear night, no moon:	.002 lx
Clear night, full moon:	.27-1 lx
Family living room:	50 lx
Sunrise/sunset:	300-500 lx
Overcast day:	1000 lx
Daylight:	10000-25000 lx
Direct sunlight:	32000-130000 lx




UI/UX Authoring Tools demo Scenario (4):
535
Settings
Logout
It has been 2 hours of sitting now, please walk for some minutes
Sitting is the new smoking. It can increase your rate of contracting lung cancer more than 50%
SITTING
SMOKING
Step 1
Track user ID, Screen, Event, errors
Google Analytics
Data is sent to Google's Server
Retrieve collected data from google server using analytics core API
Step 2
Performance metrics
Task success
Errors
Learnability
Time on task
Efficiency
Dashboard


Adaptation & navigation Rules
53
6
Adaptation Layer
Adaptation Engine
Adaptation and Navigation rules
Pre-adaptation Rules
These rules are designed to check the values that are needed for final adaptive UI
The usability Rules
These rules determine if the interaction with the adapted user inter- face might be considered enough.
Adaptation Rules
Depending on the pre-adaptation and usability checked rules, different rules are triggered; which bringing the final adaptation to the device
Example
CheckUserCapabilities: Visual Acuity
If(20/12 || 20/15 || 20/20 || 20/25) {
Vision = Range of Normal
} else if (20/30 || 20/40 || 20/50,20/60) {
Vision= Mild Visual Impairment;
} else { vision= low }
Example
checkTaskEfficiency
If(TaskCompletionRate = 65 && TaskTime==1.5)
{ efficiency = 71; }
Example
TextView
If (Age =32 & Vision= low) {
adaptTextViewSize = 60dp;
}
경희대학교
KYUNG HEE UNIVERSITY
U2 system
User & Usability system

Adaptation Rule Format

537

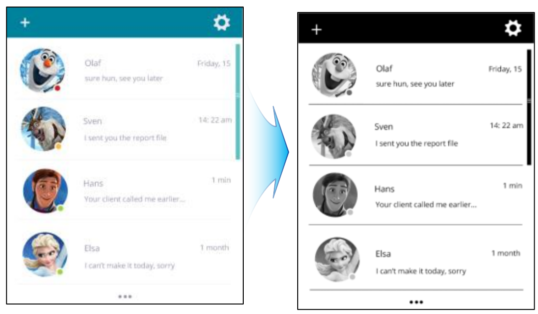
Rule Format: **Event / Condition / Action**

Example

Event: The user interface is activated

Condition : The user is color-blind

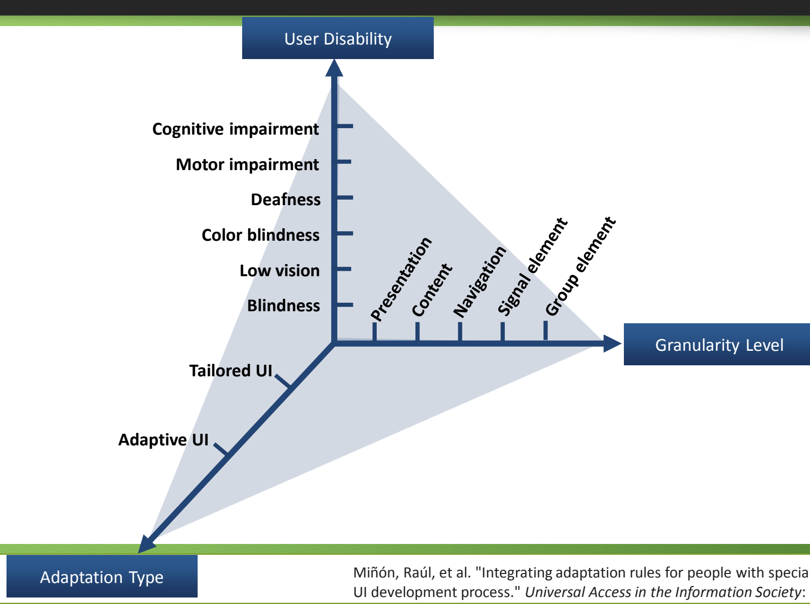
Action: change the foreground color to white and the background color to black in order to provide a high-contrast UI.



경희대학교 KYUNG HEE UNIVERSITY

Design space of the adaptation rules

538



경희대학교 KYUNG HEE UNIVERSITY

Miñón, Raúl, et al. "Integrating adaptation rules for people with special needs in model-based UI development process." *Universal Access in the Information Society*: 1-16.

Pre-adaptation Rules

539

Pre-adaptation Rules

This group of rules are designed to check the values that are needed for final adaptive UI

Pre-adaptation Rule	Classes	Descriptions	Details
checkBatteryLevelSufficient	Device, Battery	this rule evaluates if the current battery level is enough to perform any adaptation.	x ≤ 15 → not sufficient 15 < x ≤ 50 → sufficient 50 < x ≤ 100 → optimal
checklightlevel	Context, Light	This rule evaluate the current environment light condition using lux	0.0001 lux → Moonless, overcast night sky (starlight) 0.002 Lux → Moonless clear night sky with air-glow 0.27-1.0 → Full moon on a clear night
checkNosieLevel	Context, Nosie	This rule evaluate the background noise using Decibels (dB) scale	0 → absolute threshold of hearing 10 → Breathing 40 → Library 60 → office
....	....	....	....
CheckUserCapabilities	User	This rule evaluate the user capabilities such as sight , hearing, movement attention, problems	{20/12, 20/15, 20/20, 20/25} → Range of Normal Vision (sight) {20/30,20/40,20/50,20/60} → Mild Visual Impairment (sight)

경희대학교
KYUNG HEE UNIVERSITY

Usability Rules

540

The usability Rules

These rules determine if the interaction with the adapted user inter- face might be considered enough.

Usability Rule	Classes	Descriptions
checkTaskEffectiveness , checkTaskCompletion , checkErrorFrequency	Effectiveness	It measures the proportion of goals of the task achieved correctly. It measures the proportion of the task that is completed. It measures the frequency of errors.
…	…	…
checkTaskTime , checkTaskEfficiency, checkRelativeUserEfficiency	Efficiency	It measures the required time to complete the current task. It measures the efficiency of the user. It compares the efficiency of the user compared to an expert.

경희대학교
KYUNG HEE UNIVERSITY

Adaptation Rules

541

Adaptation Rules

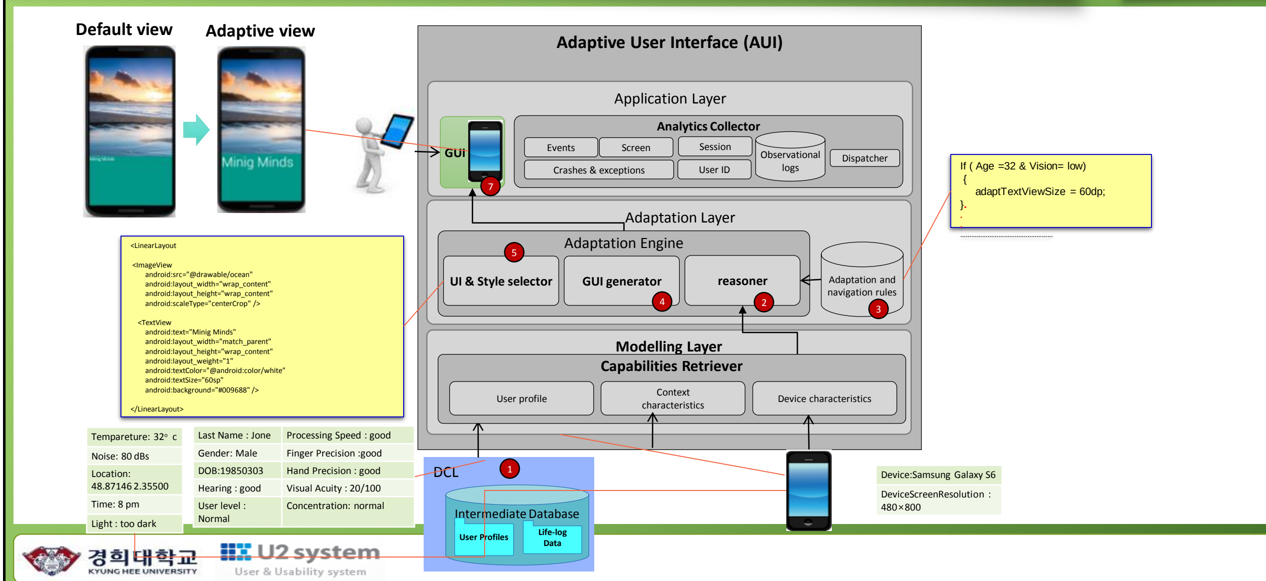
Depending on the pre-adaptation and usability checked rules, different rules are triggered; which bringing the final adaptation to the device

Adaptation Rule	Classes	Descriptions
adaptBrightness and adaptVolume	Brightness, Volume	These rules adapt the screen brightness and volume of device according to user profile values
adaptButtonSize , adaptButtonBackgroundColor, adaptButtonTextColor,	Button	These rules considered the buttons configuration according the user disabilities and its profile
…	…	…
adaptTextViewSize, adaptTextViewTextColor adaptTextViewBackgoundColor	TextView	These rules considered the TextViews configuration according the user disabilities and its profile

경희대학교 KYUNG HEE UNIVERSITY

The information flow within the AdaptUI

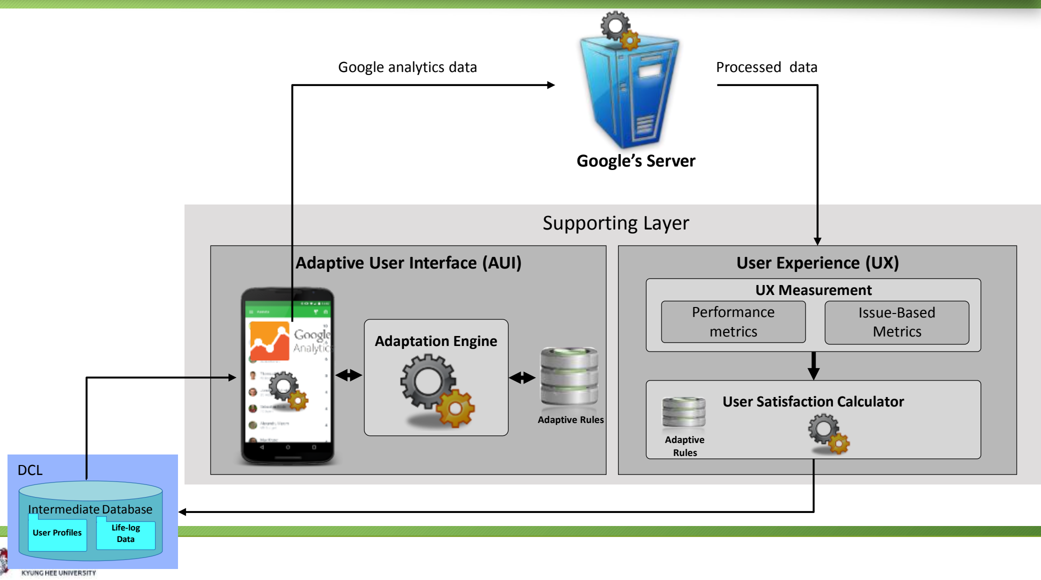
542



경희대학교 KYUNG HEE UNIVERSITY U2 system User & Usability system

Strategy II: UX Experience(UX)

543



KYUNG HEE UNIVERSITY

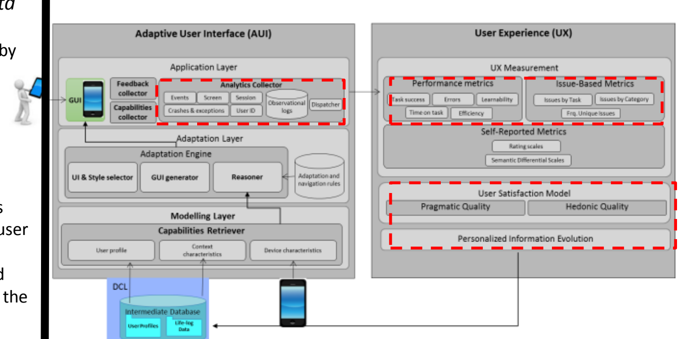
Methodology : Analytics collector and UX measurement

544

Methodology Description

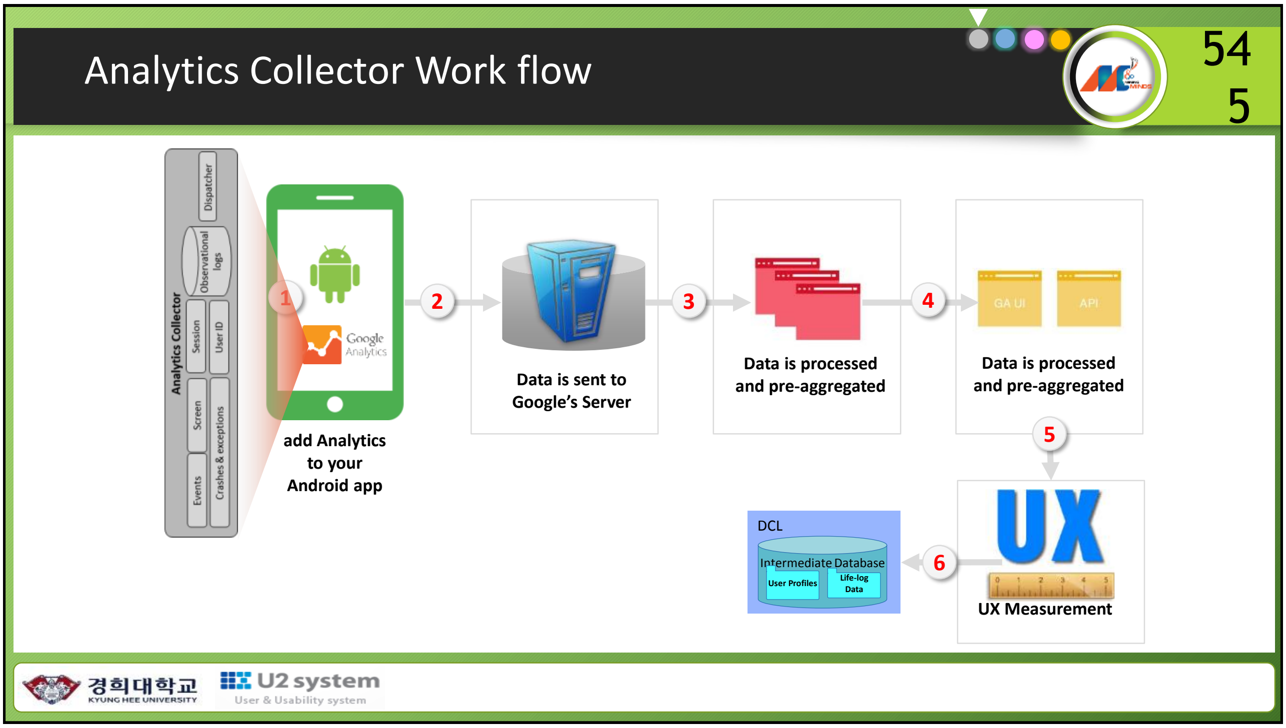
Use-case (SL2.5-UCS-05): *Collect and analyze observational data*

- **Objective**
  - To identify areas of improvement and maximize the user interaction by analyzing the user interaction with app.
- **Methodology**
  - *Communication*: end user interaction with MM app
  - **Analytics collector and UX measurement**
    1. **Input**: receives *request from end user*
    2. **Processing**:
       - Analytics collector collect the user interaction data such as user ID, event, session, screen, crashes & exceptions, and user timings
       - The collected data is stored locally before being dispatched
       - Data is dispatched for user experience measurement from the app for every 30 minutes
       - the pragmatic quality such as usability-(e.g. performance, issues) are calculated in order to find the user experience (UX)
    3. **Output:**
       a. UX quality variables are sent to DCL2 for storage/updating in user profile.

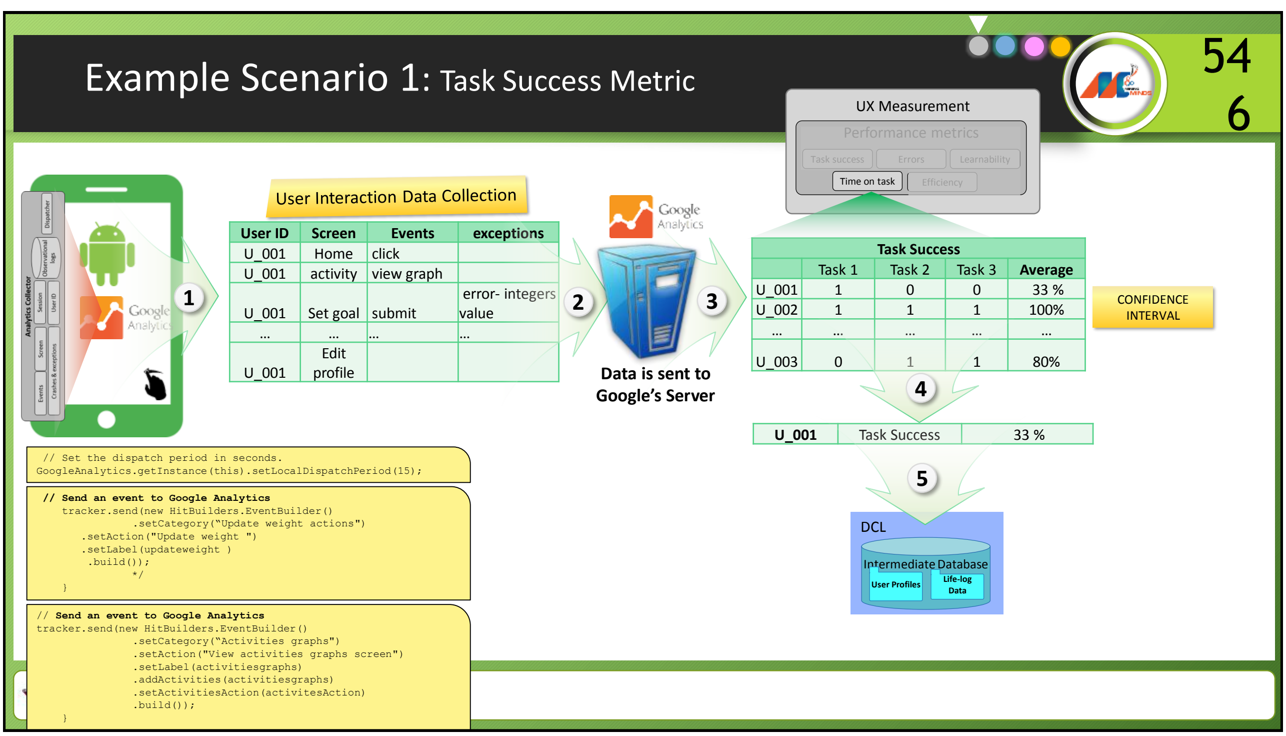


경희대학교 KYUNG HEE UNIVERSITY

U2 system User & Usability system



Analytics Collector Work flow
54
5
Analytics Collector
Dispatcher
Observational logs
Session
User ID
Screen
Crashes & exceptions
Events
1
Google Analytics
add Analytics to your Android app
2
Data is sent to Google's Server
3
Data is processed and pre-aggregated
4
GA UI
API
Data is processed and pre-aggregated
5
UX
UX Measurement
6
DCL
Intermediate Database
User Profiles
Life-log Data
경희대학교
KYUNG HEE UNIVERSITY
U2 system
User & Usability system



Example Scenario 1: Task Success Metric
54
6
UX Measurement
Performance metrics
Task success
Errors
Learnability
Time on task
Efficiency
1
User Interaction Data Collection

User ID	Screen	Events	exceptions
U_001	Home	click	
U_001	activity	view graph	
U_001	Set goal	submit	error- integers value
...	...	...	...
U_001	Edit profile		

Google Analytics
2
3
Data is sent to Google's Server

Task Success				
	Task 1	Task 2	Task 3	Average
U_001	1	0	0	33 %
U_002	1	1	1	100%
...	...	...	...	...
U_003	0	1	1	80%

CONFIDENCE INTERVAL
4

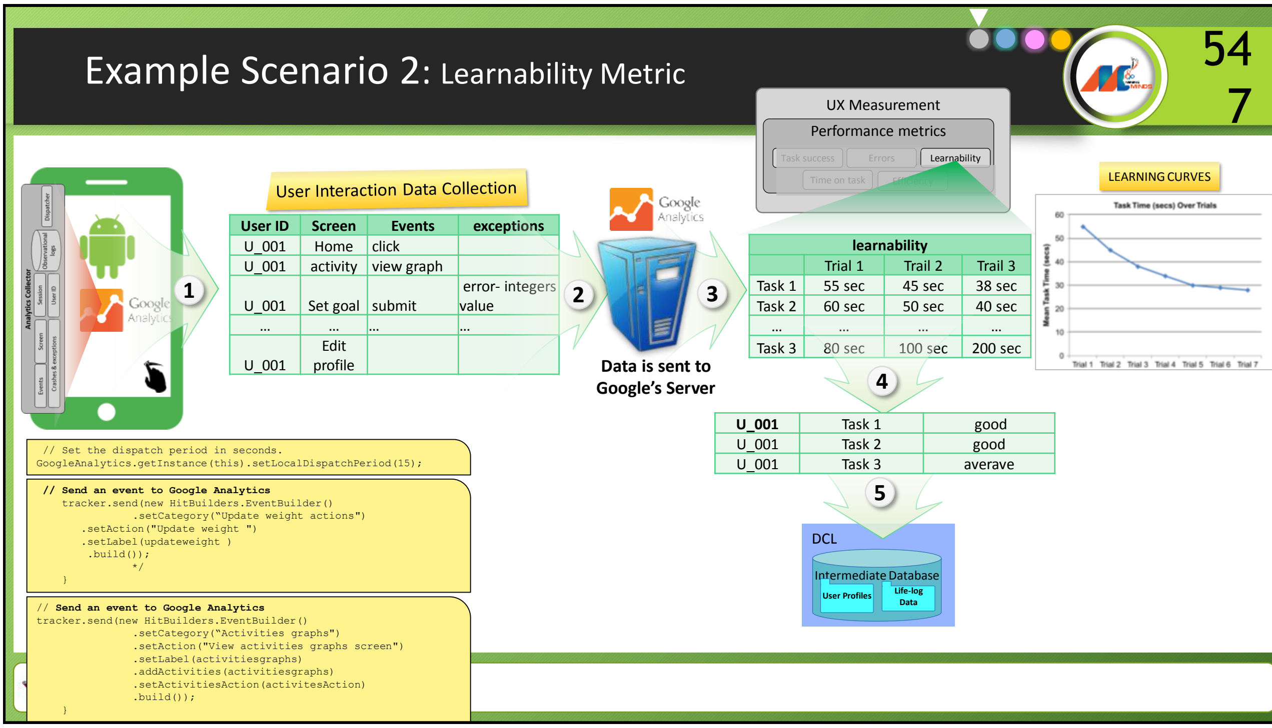
U_001	Task Success	33 %

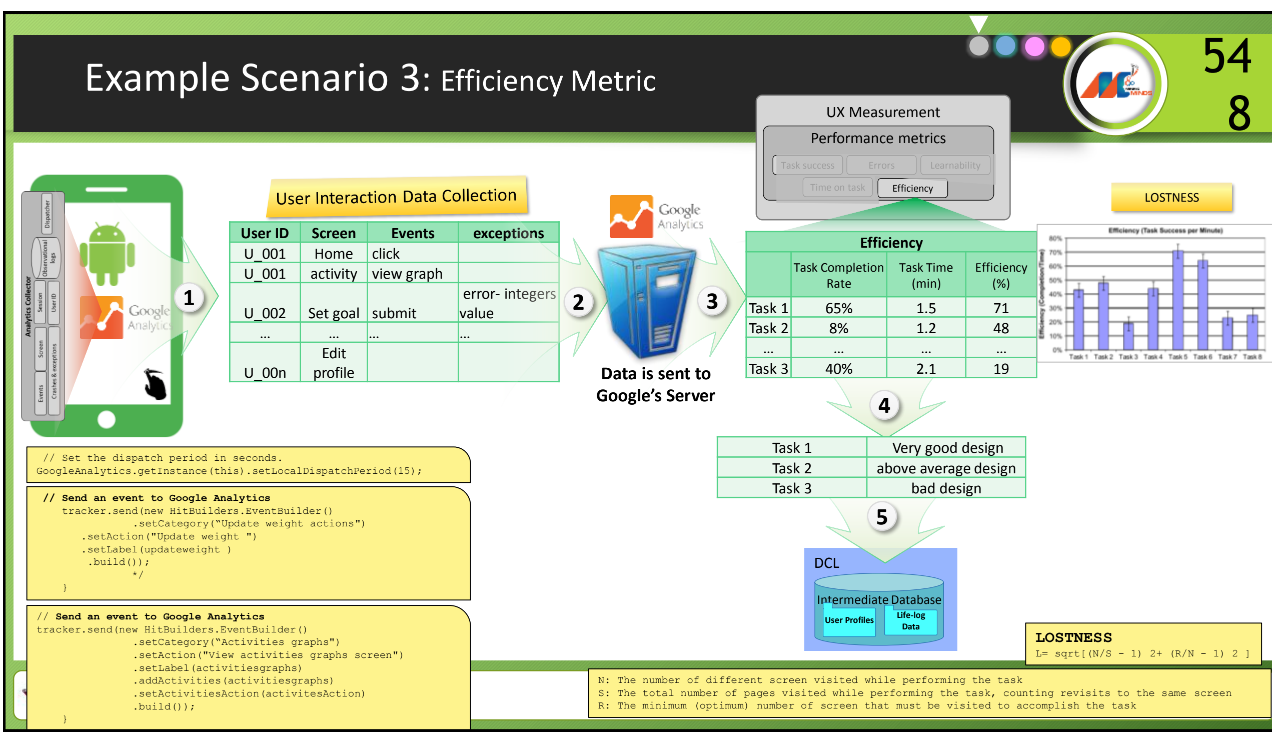
5
DCL
Intermediate Database
User Profiles
Life-log Data

```
 // Set the dispatch period in seconds.
GoogleAnalytics.getInstance(this).setLocalDispatchPeriod(15);
```

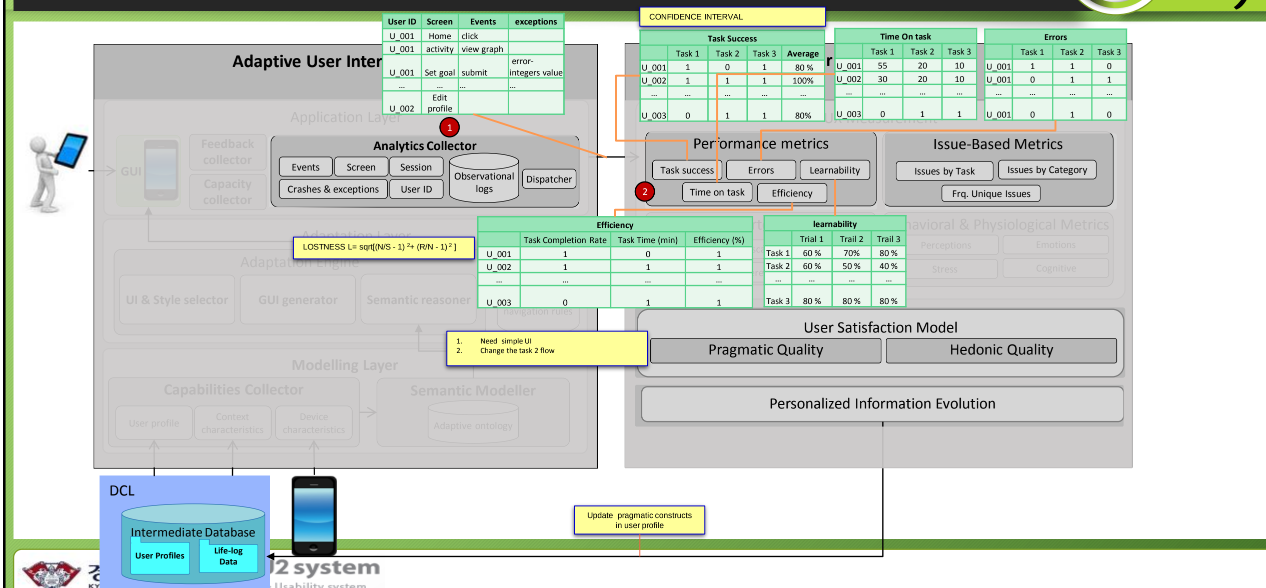
```
 // Send an event to Google Analytics
    tracker.send(new HitBuilders.EventBuilder()
                .setCategory("Update weight actions")
       .setAction("Update weight ")
       .setLabel(updateweight )
        .build());
                */
    }
```

```
// Send an event to Google Analytics
tracker.send(new HitBuilders.EventBuilder()
               .setCategory("Activities graphs")
               .setAction("View activities graphs screen")
               .setLabel(activitiesgraphs)
               .addActivities(activitiesgraphs)
               .setActivitiesAction(activitesAction)
               .build());
    }
```


Example Scenario 2: Learnability Metric
54
7
UX Measurement
Performance metrics
Task success
Errors
Learnability
Time on task
LEARNING CURVES
Task Time (secs) Over Trials
Mean Task Time (secs)
Trial 1 Trial 2 Trial 3 Trial 4 Trial 5 Trial 6 Trial 7
User Interaction Data Collection
User ID | Screen | Events | exceptions
U_001 | Home | click
U_001 | activity | view graph
U_001 | Set goal | submit | error- integers value
... | ... | ... | ...
U_001 | Edit profile
1
2
3
Data is sent to Google's Server
Google Analytics
learnability
Trial 1 | Trail 2 | Trail 3
Task 1 | 55 sec | 45 sec | 38 sec
Task 2 | 60 sec | 50 sec | 40 sec
... | ... | ... | ...
Task 3 | 80 sec | 100 sec | 200 sec
4
U_001 | Task 1 | good
U_001 | Task 2 | good
U_001 | Task 3 | averave
5
DCL
Intermediate Database
User Profiles
Life-log Data
Analytics Collector
Dispatcher
Session
User ID
Screen
Events
Crashes & exceptions
// Set the dispatch period in seconds.
GoogleAnalytics.getInstance(this).setLocalDispatchPeriod(15);
// Send an event to Google Analytics
tracker.send(new HitBuilders.EventBuilder()
.setCategory("Update weight actions")
.setAction("Update weight ")
.setLabel(updateweight)
.build());
*/
}
// Send an event to Google Analytics
tracker.send(new HitBuilders.EventBuilder()
.setCategory("Activities graphs")
.setAction("View activities graphs screen")
.setLabel(activitiesgraphs)
.addActivities(activitiesgraphs)
.setActivitiesAction(activitesAction)
.build());
}


Example Scenario 3: Efficiency Metric
54
8
UX Measurement
Performance metrics
Task success
Errors
Learnability
Time on task
Efficiency
LOSTNESS
Efficiency (Task Success per Minute)
Task 1 Task 2 Task 3 Task 4 Task 5 Task 6 Task 7 Task 8
User Interaction Data Collection
User ID | Screen | Events | exceptions
U_001 | Home | click
U_001 | activity | view graph
U_002 | Set goal | submit | error- integers value
... | ... | ... | ...
U_00n | Edit profile
1
2
3
Data is sent to Google's Server
Google Analytics
Efficiency
Task Completion Rate | Task Time (min) | Efficiency (%)
Task 1 | 65% | 1.5 | 71
Task 2 | 8% | 1.2 | 48
... | ... | ... | ...
Task 3 | 40% | 2.1 | 19
4
Task 1 | Very good design
Task 2 | above average design
Task 3 | bad design
5
DCL
Intermediate Database
User Profiles
Life-log Data
LOSTNESS
L= sqrt[(N/S - 1) 2+ (R/N - 1) 2]
N: The number of different screen visited while performing the task
S: The total number of pages visited while performing the task, counting revisits to the same screen
R: The minimum (optimum) number of screen that must be visited to accomplish the task
// Set the dispatch period in seconds.
GoogleAnalytics.getInstance(this).setLocalDispatchPeriod(15);
// Send an event to Google Analytics
tracker.send(new HitBuilders.EventBuilder()
.setCategory("Update weight actions")
.setAction("Update weight ")
.setLabel(updateweight)
.build());
*/
}
// Send an event to Google Analytics
tracker.send(new HitBuilders.EventBuilder()
.setCategory("Activities graphs")
.setAction("View activities graphs screen")
.setLabel(activitiesgraphs)
.addActivities(activitiesgraphs)
.setActivitiesAction(activitesAction)
.build());
}

Example Scenario 4: All UX Measurement Metrics

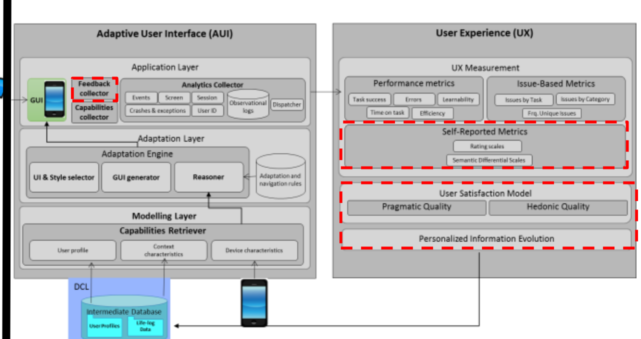


Methodology：Feedback Collector & Self-reported Metrics

Methodology Description

Use-case (SL2.5-UCS-10): *Collect user response data as self-reported metrics*

- **Objective**
  - To identify the user feeling/response based on user feedback in order to evaluate the user experience
- **Methodology**
  - *Communication*: end user fill the user feedback from by MM application
  - **Feedback data and UX measurement**
    1. **Input**: receives *request from end user*
    2. **Processing**:
       - The end user provide feedback using the questionnaire.
       - The feedback is sent to user experience in order to evaluate the user response.
       - user experience variables such as usability, pleasure, beauty are calculated based on filled questions
       - The UI/UX update the calculated variables values in user profile by sending request to DCL 2.5
    3. **Output:**
       a. UX quality variables are sent to DCL2.5 for storage/updating in user profile.



Tools and Technologies

551

- **Data Representation**
  - Object Oriented representation
- **Implementation Tools and Technologies**
  - Language Java
  - Language PHP
  - HTML 5
  - Angular.JS
  - JQuery
  - Web services (SOAP based/Restful)
  - Google analytics
  - Google tagmanage

Contributions

552

- Rules for adaptive UI
- User experience measurement toolkit development

경희대학교 KYUNG HEE UNIVERSITY

Conclusion

- UI/UX Authoring Tool is the adaptive face of Mining Minds Platform
- Personalized and contextual information presentation key aspect of adaptation
- User experience is the distinguishing factor of MM UI/UX Authoring tool
- User Experience (UX) can be improved with Adaptive UI
- Adaptive UI can improved accessibility
- Adaptive UI can improved users' performance and satisfaction
- Continuous evolution of UI with contextual information change
- UX Authoring Tool for user experience measurement

Supporting Layer (SL 2.5)

Privacy and Security

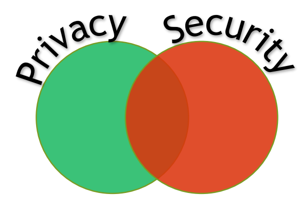
Introduction [1/2]

555

Privacy and Security are closely related terms having symbiotic relationship with each other.

Privacy applies to a consumers right to safeguard his or her information from any other party

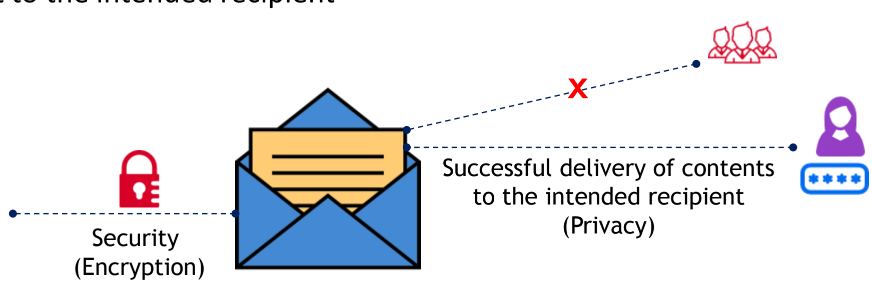
Security is the confidentiality, integrity, and availability of data as per the CIA triad model



Introduction [2/2]

556

Security is a concealed envelope and **Privacy** is the successful delivery of the message inside that content to the intended recipient



Overview of existing adaptive systems

557

Existing Systems	Descriptions	Pros:	Cons:
MyFitness Pall [4]	MyFitness pall is a weight lose applications that uses the consumption of calories and guide the user accordingly	• Easy to use and interactive • Comprehensive food catalogue over 5 million entries • Shares the privacy policy with user • Free to use	• The system lacks authentication
Endomondo [5]	Endomondo is a personal trainer in the pocket. It allows users to track their fitness and health statistics with a mobile application and website	• Stores data on cloud • Offers DLP in case of stolen device • Shares privacy policy with the user	• The system lacks authentication • Do not update profiles [3] • Lacks profile integrity check
iTriage[6]	iTriage is a FREE smartphone app with comprehensive medical content that guides you through the symptom-to-provider ™ pathway	• After checking symptoms of user disease related problem it recommends the nearest medical service (usage of geo location services)	• Authentication is missing • Geo location service is restricted with the discovery of nearest medical services alone
Runkeeper[7]	It record, monitors and visually display the running activities performed by a user	• Interactive and easy to use • Records running, walk, cycling, workout and track pace and weight • Secure storage and authentication is available	• Do not explicitly state about the user profile and its integrity check if modified accidently or maliciously
Argus[11]	It tailors workouts based on fitness goals. Helps the user to progress quickly without to worry about details.	• Tracks steps, calories, heart rate , meals and hours of sleep	• Profile update and its impact on system generated recommendation is not explicitly stated

경희대학교 KYUNG HEE UNIVERSITY

Limitations of existing Work

558

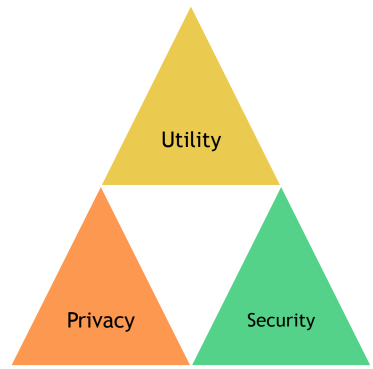
- Ability to detect duplicate sign on with respect to
  - User habitual workout area
  - Distance measure between two locations
- Profile integrity check
- Successful and integrity assured delivery of recommendations

경희대학교 KYUNG HEE UNIVERSITY

Motivation

559

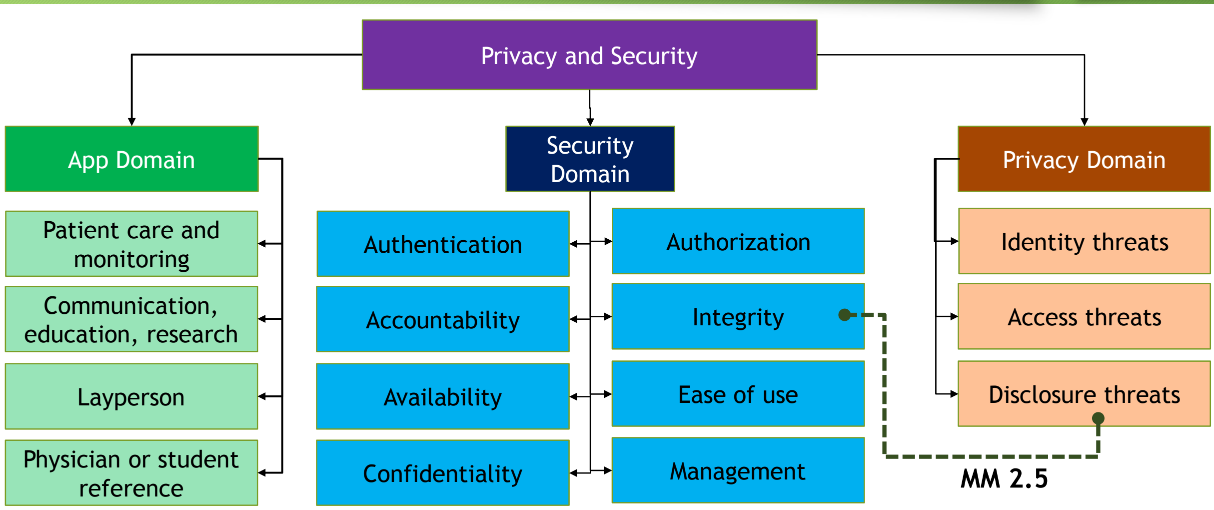
- Data security at rest and during its transmission to end user
- Privacy aware delivery of services (recommendations)
- Maximum utilization of data resources without disclosing the individual identity



경희대학교
KYUNG HEE UNIVERSITY

Taxonomy-Privacy and Security in mHealth Applications

560



경희대학교
KYUNG HEE UNIVERSITY

Miloslava Plachkinova et al. A taxonomy of mHealth Apps - Security and Privacy-2015

Challenges and Solutions

561

Challenge	Information Disclosure threat -Inference attack	Information Integrity threat-recommendations	Privacy and Security-Man in the Middle Attack
Solution	Achieving through query and response retention with successive queries	Monitoring of data resources affecting recommendations	Message digest and double acknowledgement strategy
Methodology	Preemptive discovery of information disclosure using query and response logs	Observer on profile update with user acknowledgement	Message digest using the public key of MM

경희대학교
KYUNG HEE UNIVERSITY

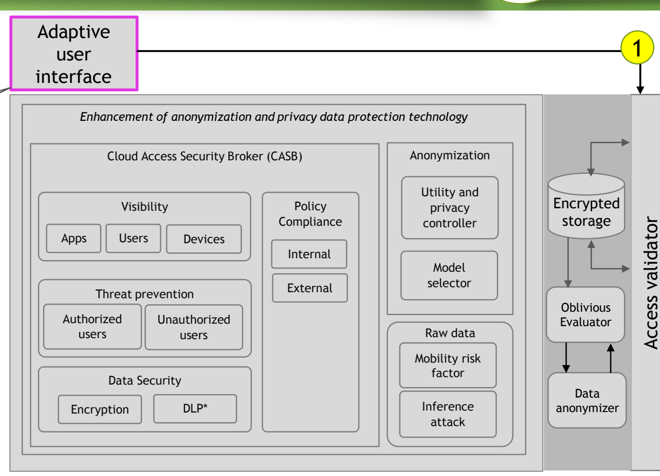
Disclosure Threat : Inference Attack (1/5)

562

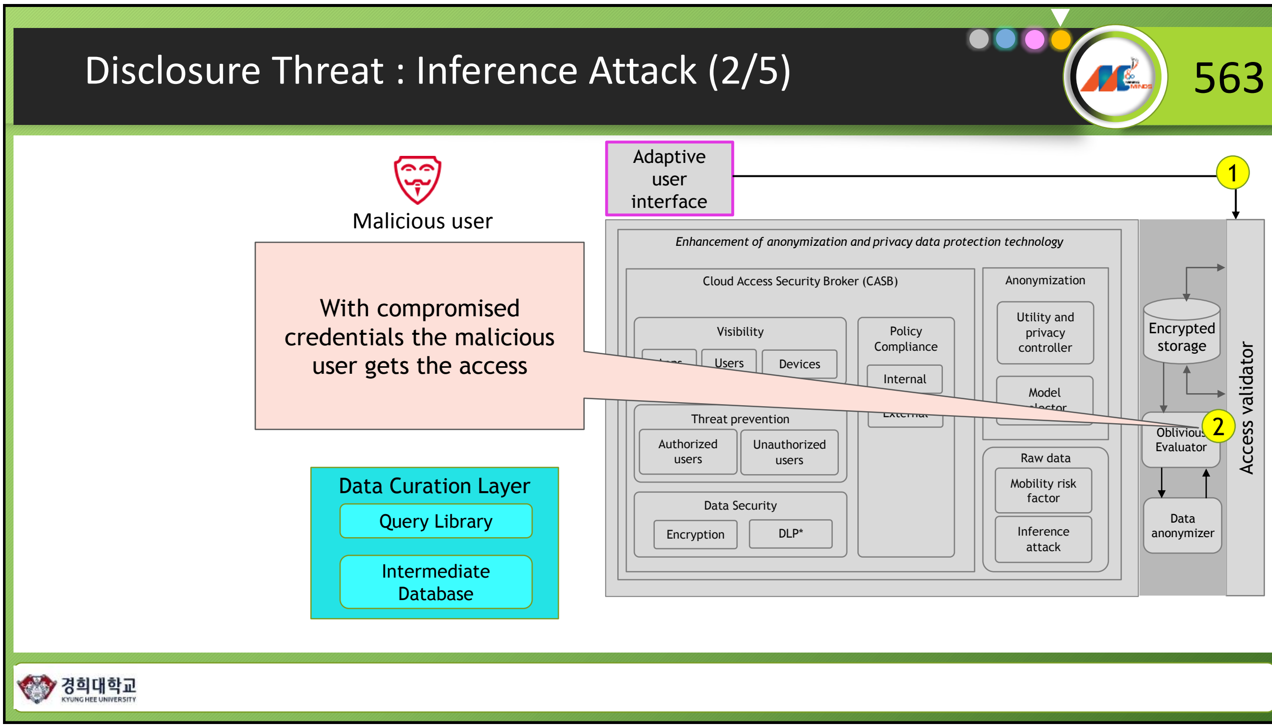
Malicious user

Using the adaptive user interface a malicious user interact with the system with compromised credentials

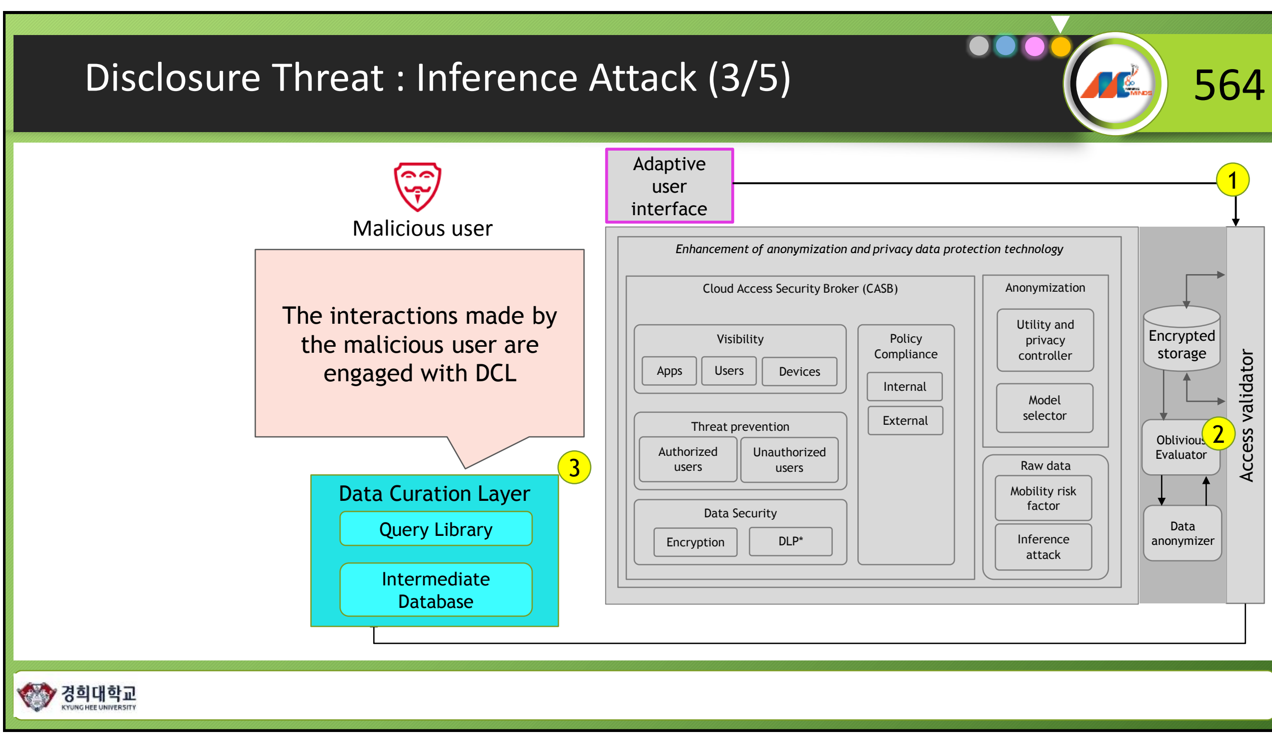
Data Curation Layer
Query Library
Intermediate Database



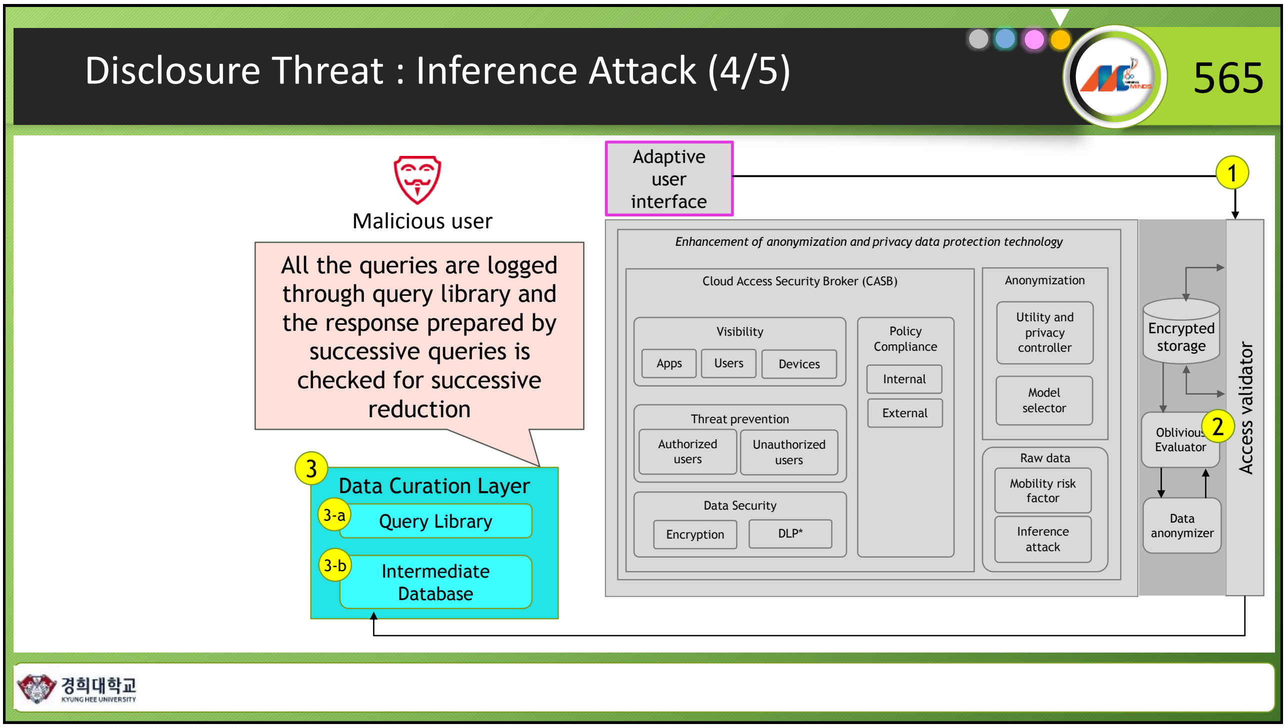
경희대학교
KYUNG HEE UNIVERSITY

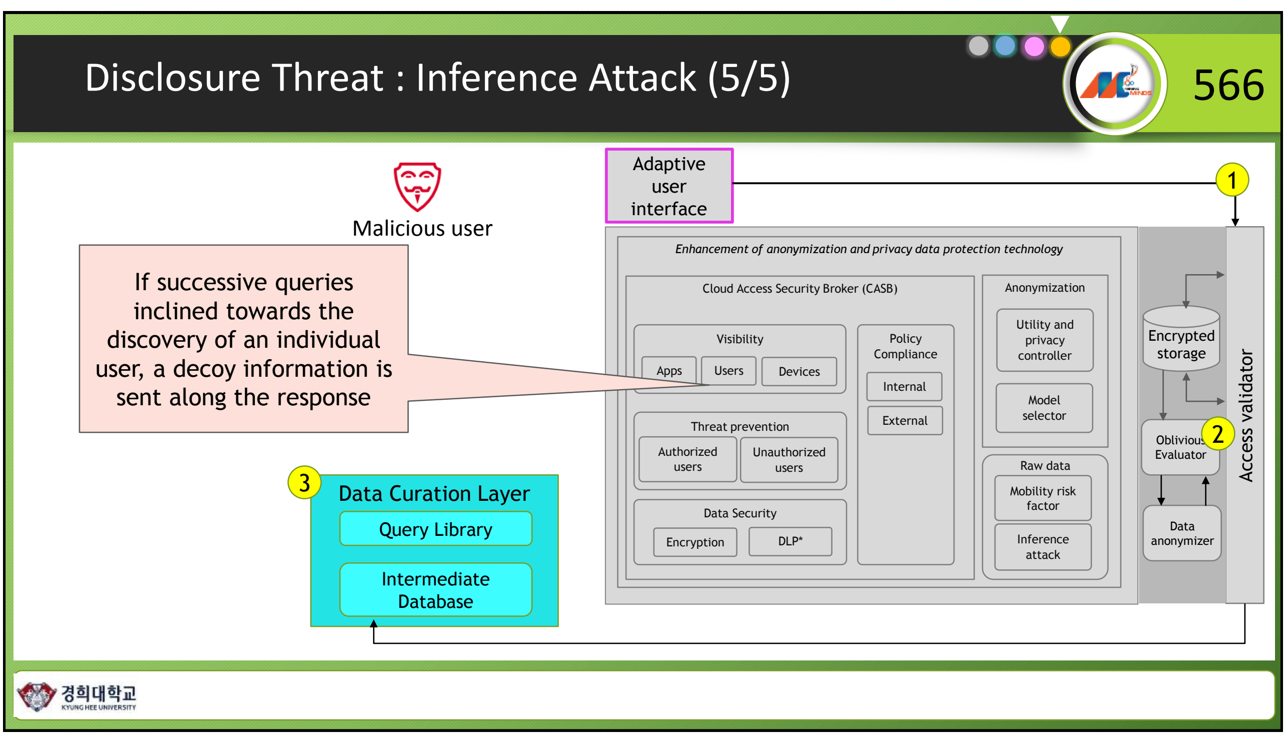


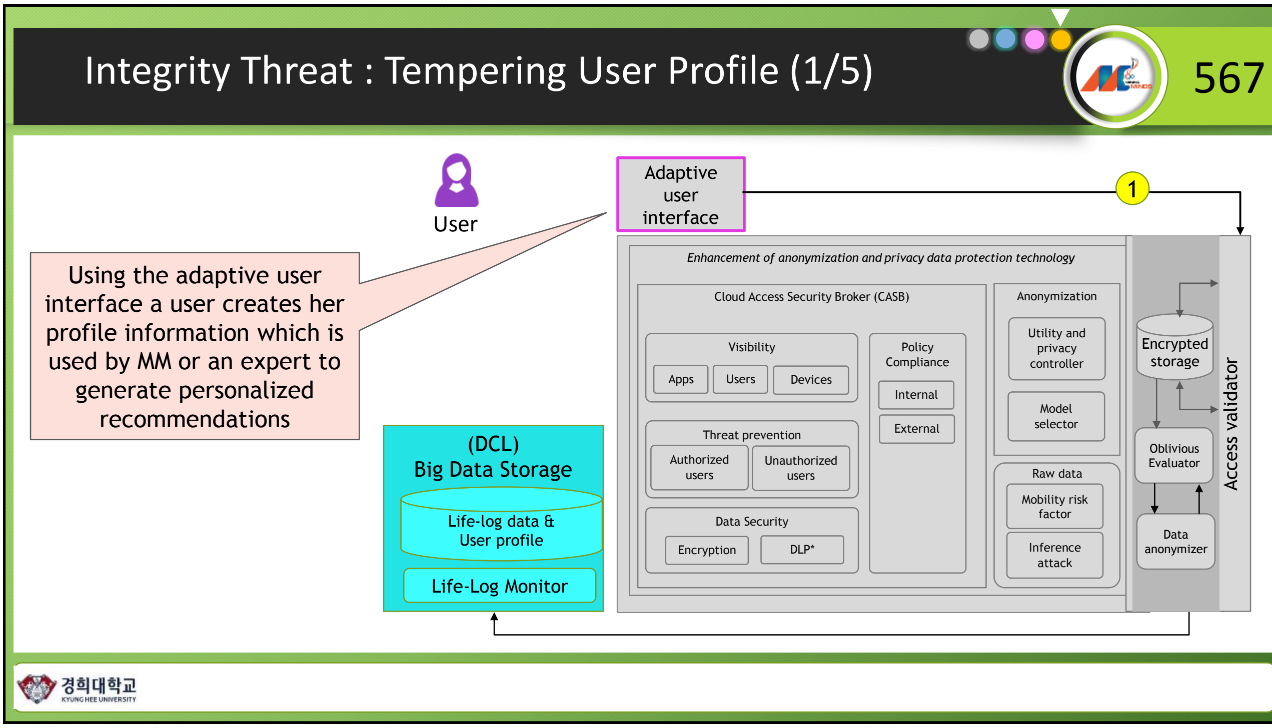
Disclosure Threat : Inference Attack (2/5)
563
Malicious user
With compromised credentials the malicious user gets the access
Data Curation Layer
Query Library
Intermediate Database
Adaptive user interface
1
Enhancement of anonymization and privacy data protection technology
Cloud Access Security Broker (CASB)
Visibility
Users
Devices
Policy Compliance
Internal
Threat prevention
Authorized users
Unauthorized users
Data Security
Encryption
DLP*
Anonymization
Utility and privacy controller
Model
Raw data
Mobility risk factor
Inference attack
Encrypted storage
Evaluator
2
Data anonymizer
Access validator
경희대학교
KYUNG HEE UNIVERSITY



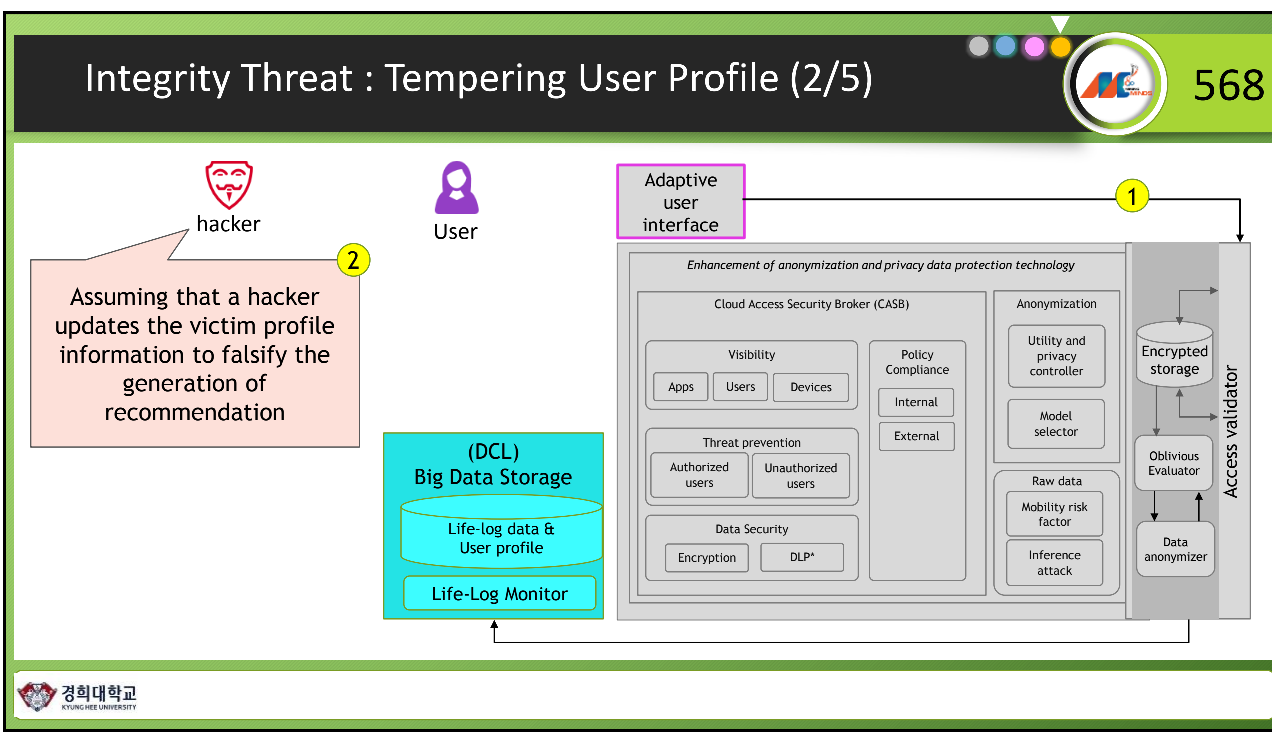
Disclosure Threat : Inference Attack (3/5)
564
Malicious user
The interactions made by the malicious user are engaged with DCL
3
Data Curation Layer
Query Library
Intermediate Database
Adaptive user interface
1
Enhancement of anonymization and privacy data protection technology
Cloud Access Security Broker (CASB)
Visibility
Apps
Users
Devices
Policy Compliance
Internal
External
Threat prevention
Authorized users
Unauthorized users
Data Security
Encryption
DLP*
Anonymization
Utility and privacy controller
Model selector
Raw data
Mobility risk factor
Inference attack
Encrypted storage
Oblivious Evaluator
2
Data anonymizer
Access validator
경희대학교
KYUNG HEE UNIVERSITY


Disclosure Threat : Inference Attack (4/5)
565
Adaptive user interface
Malicious user
All the queries are logged through query library and the response prepared by successive queries is checked for successive reduction
3
Data Curation Layer
3-a
Query Library
3-b
Intermediate Database
Enhancement of anonymization and privacy data protection technology
Cloud Access Security Broker (CASB)
Visibility
Apps
Users
Devices
Policy Compliance
Internal
External
Threat prevention
Authorized users
Unauthorized users
Data Security
Encryption
DLP*
Anonymization
Utility and privacy controller
Model selector
Raw data
Mobility risk factor
Inference attack
Encrypted storage
Oblivious Evaluator
Data anonymizer
Access validator
1
2
경희대학교
KYUNG HEE UNIVERSITY

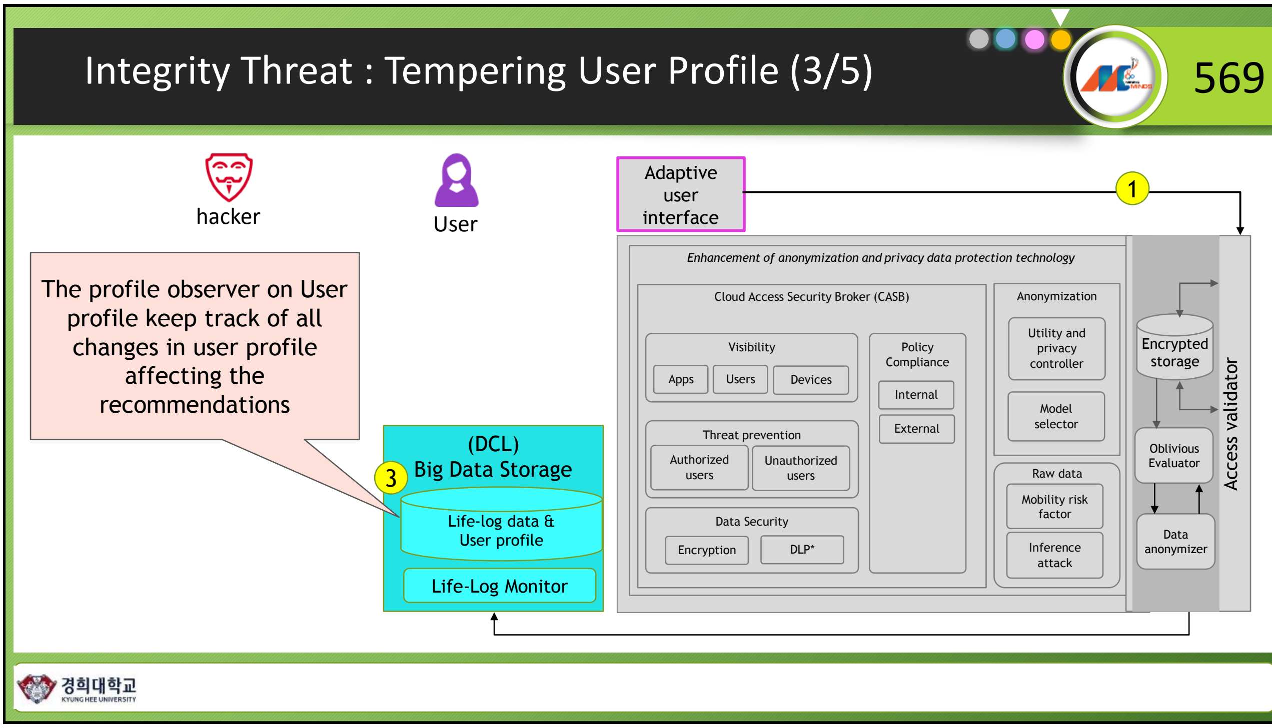

Disclosure Threat : Inference Attack (5/5)
566
Adaptive user interface
Malicious user
If successive queries inclined towards the discovery of an individual user, a decoy information is sent along the response
3
Data Curation Layer
Query Library
Intermediate Database
Enhancement of anonymization and privacy data protection technology
Cloud Access Security Broker (CASB)
Visibility
Apps
Users
Devices
Policy Compliance
Internal
External
Threat prevention
Authorized users
Unauthorized users
Data Security
Encryption
DLP*
Anonymization
Utility and privacy controller
Model selector
Raw data
Mobility risk factor
Inference attack
Encrypted storage
Oblivious Evaluator
Data anonymizer
Access validator
1
2
경희대학교
KYUNG HEE UNIVERSITY

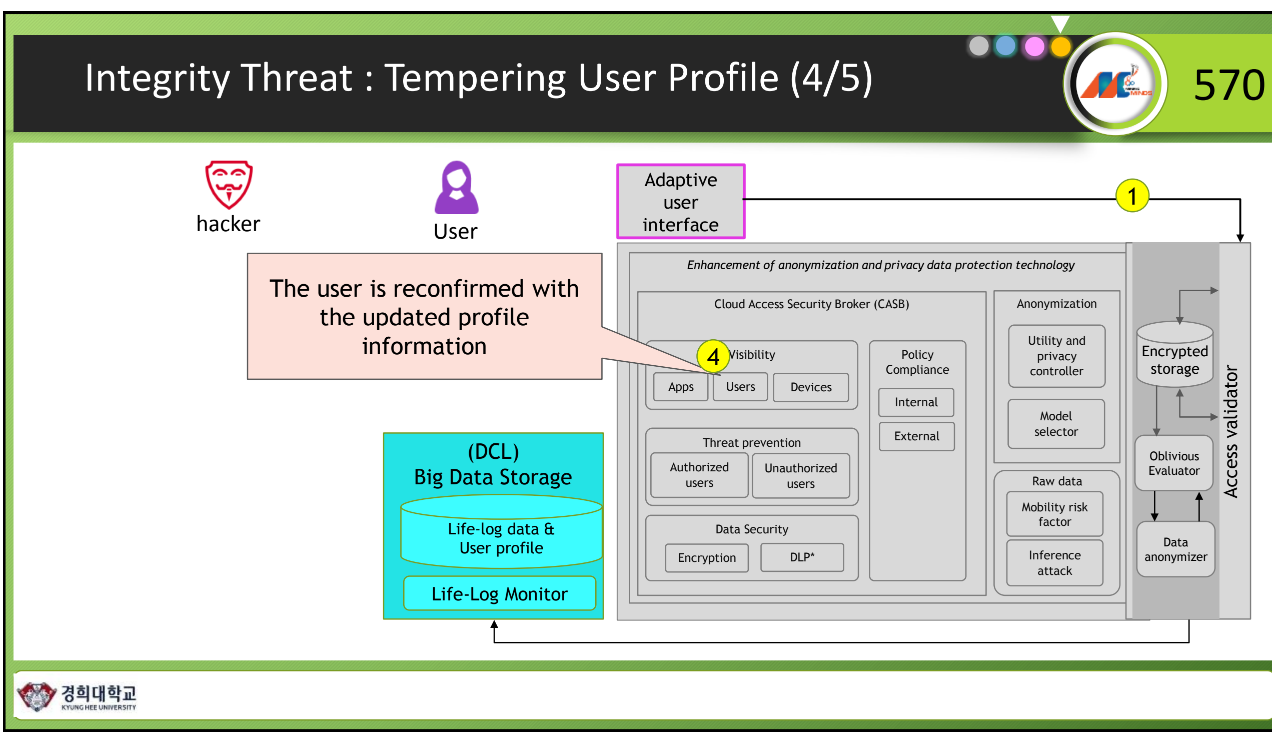


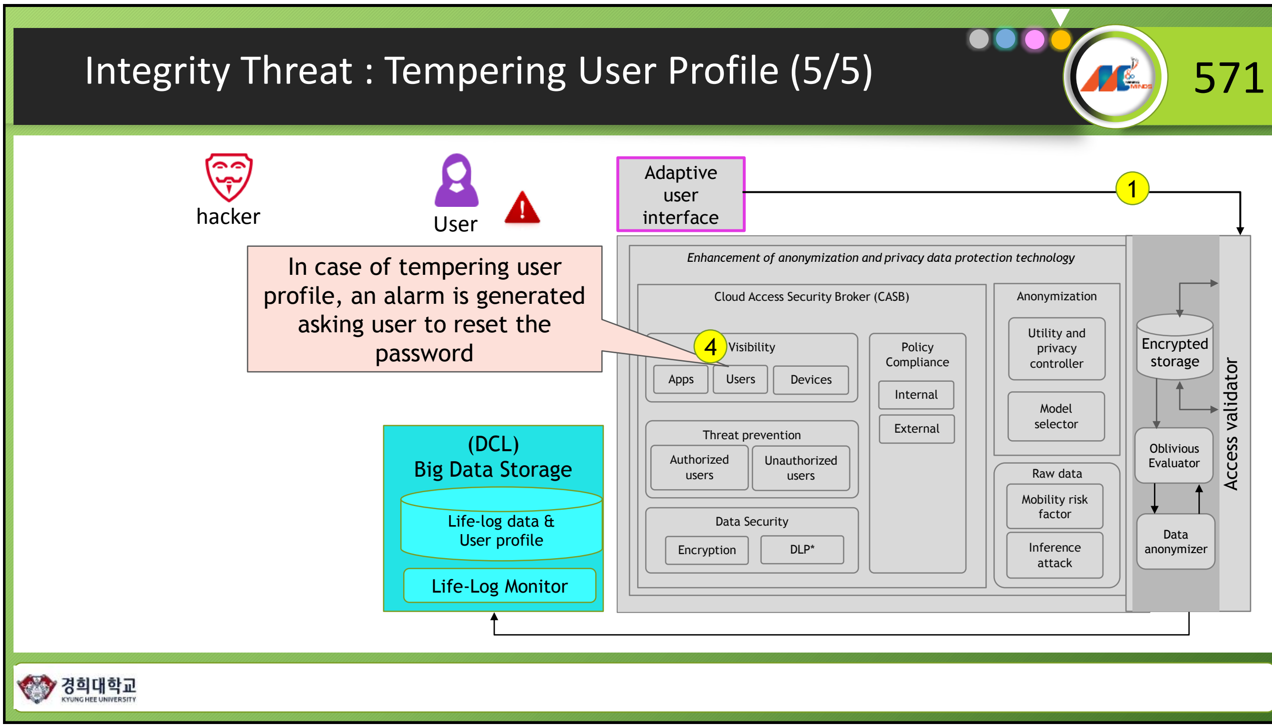
Integrity Threat : Tempering User Profile (1/5)
567
User
Adaptive user interface
1
Using the adaptive user interface a user creates her profile information which is used by MM or an expert to generate personalized recommendations
Enhancement of anonymization and privacy data protection technology
Cloud Access Security Broker (CASB)
Visibility
Apps
Users
Devices
Policy Compliance
Internal
External
Threat prevention
Authorized users
Unauthorized users
Data Security
Encryption
DLP*
Anonymization
Utility and privacy controller
Model selector
Raw data
Mobility risk factor
Inference attack
Encrypted storage
Oblivious Evaluator
Data anonymizer
Access validator
(DCL)
Big Data Storage
Life-log data & User profile
Life-Log Monitor
경희대학교
KYUNG HEE UNIVERSITY



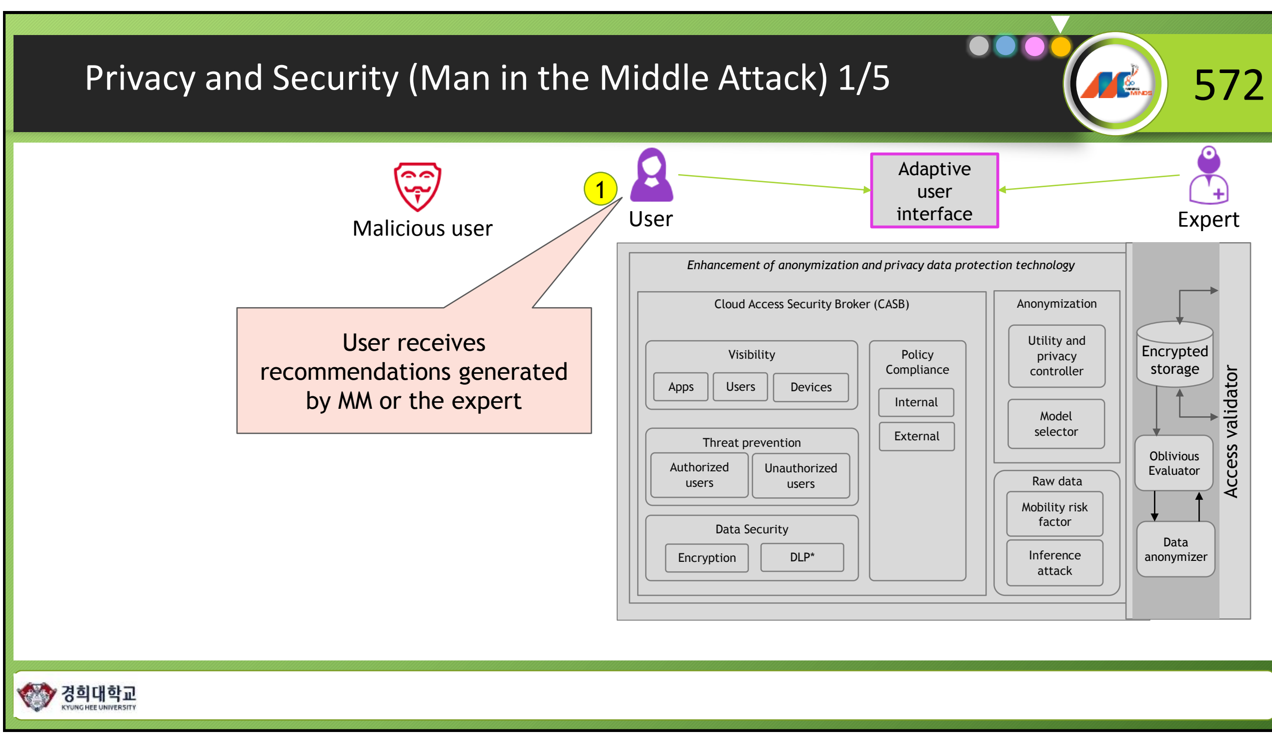
Integrity Threat : Tempering User Profile (2/5)
568
hacker
User
Adaptive user interface
1
2
Assuming that a hacker updates the victim profile information to falsify the generation of recommendation
Enhancement of anonymization and privacy data protection technology
Cloud Access Security Broker (CASB)
Visibility
Apps
Users
Devices
Policy Compliance
Internal
External
Threat prevention
Authorized users
Unauthorized users
Data Security
Encryption
DLP*
Anonymization
Utility and privacy controller
Model selector
Raw data
Mobility risk factor
Inference attack
Encrypted storage
Oblivious Evaluator
Data anonymizer
Access validator
(DCL)
Big Data Storage
Life-log data & User profile
Life-Log Monitor
경희대학교
KYUNG HEE UNIVERSITY


Integrity Threat : Tempering User Profile (3/5)
569
hacker
User
The profile observer on User profile keep track of all changes in user profile affecting the recommendations
Adaptive user interface
1
Enhancement of anonymization and privacy data protection technology
Cloud Access Security Broker (CASB)
Visibility
Apps
Users
Devices
Policy Compliance
Internal
External
Threat prevention
Authorized users
Unauthorized users
Data Security
Encryption
DLP*
Anonymization
Utility and privacy controller
Model selector
Raw data
Mobility risk factor
Inference attack
Encrypted storage
Oblivious Evaluator
Data anonymizer
Access validator
(DCL)
Big Data Storage
3
Life-log data & User profile
Life-Log Monitor
경희대학교
KYUNG HEE UNIVERSITY

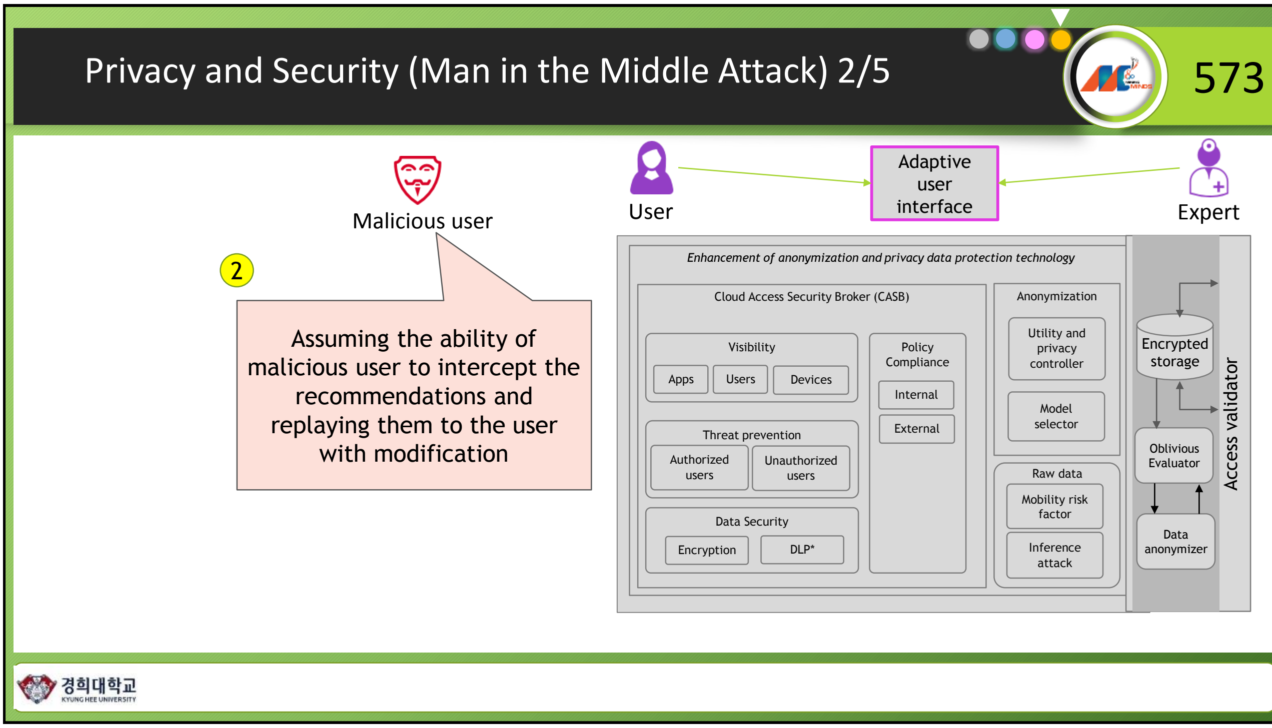

Integrity Threat : Tempering User Profile (4/5)
570
hacker
User
The user is reconfirmed with the updated profile information
Adaptive user interface
1
Enhancement of anonymization and privacy data protection technology
Cloud Access Security Broker (CASB)
4
Visibility
Apps
Users
Devices
Policy Compliance
Internal
External
Threat prevention
Authorized users
Unauthorized users
Data Security
Encryption
DLP*
Anonymization
Utility and privacy controller
Model selector
Raw data
Mobility risk factor
Inference attack
Encrypted storage
Oblivious Evaluator
Data anonymizer
Access validator
(DCL)
Big Data Storage
Life-log data & User profile
Life-Log Monitor
경희대학교
KYUNG HEE UNIVERSITY



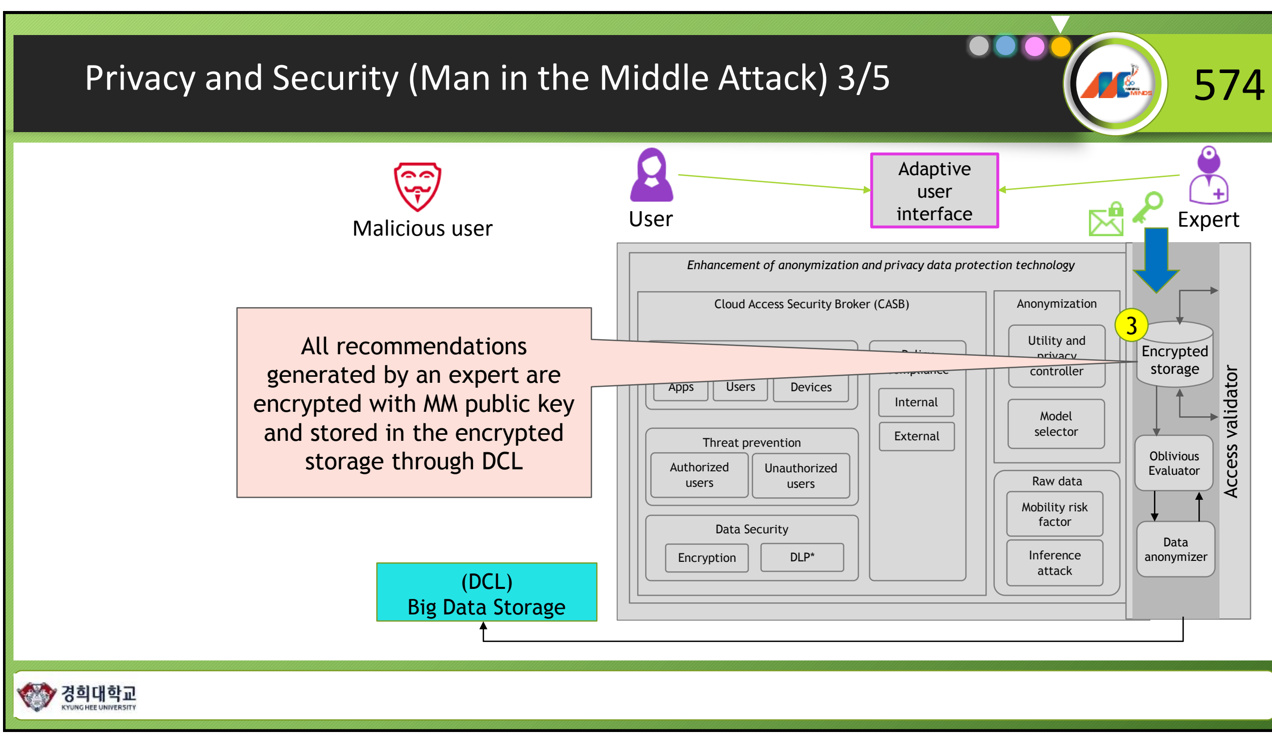
Integrity Threat : Tempering User Profile (5/5)
571
hacker
User
Adaptive user interface
1
In case of tempering user profile, an alarm is generated asking user to reset the password
Enhancement of anonymization and privacy data protection technology
Cloud Access Security Broker (CASB)
4
Visibility
Apps
Users
Devices
Policy Compliance
Internal
External
Threat prevention
Authorized users
Unauthorized users
Data Security
Encryption
DLP*
Anonymization
Utility and privacy controller
Model selector
Raw data
Mobility risk factor
Inference attack
Encrypted storage
Oblivious Evaluator
Data anonymizer
Access validator
(DCL)
Big Data Storage
Life-log data & User profile
Life-Log Monitor
경희대학교
KYUNG HEE UNIVERSITY



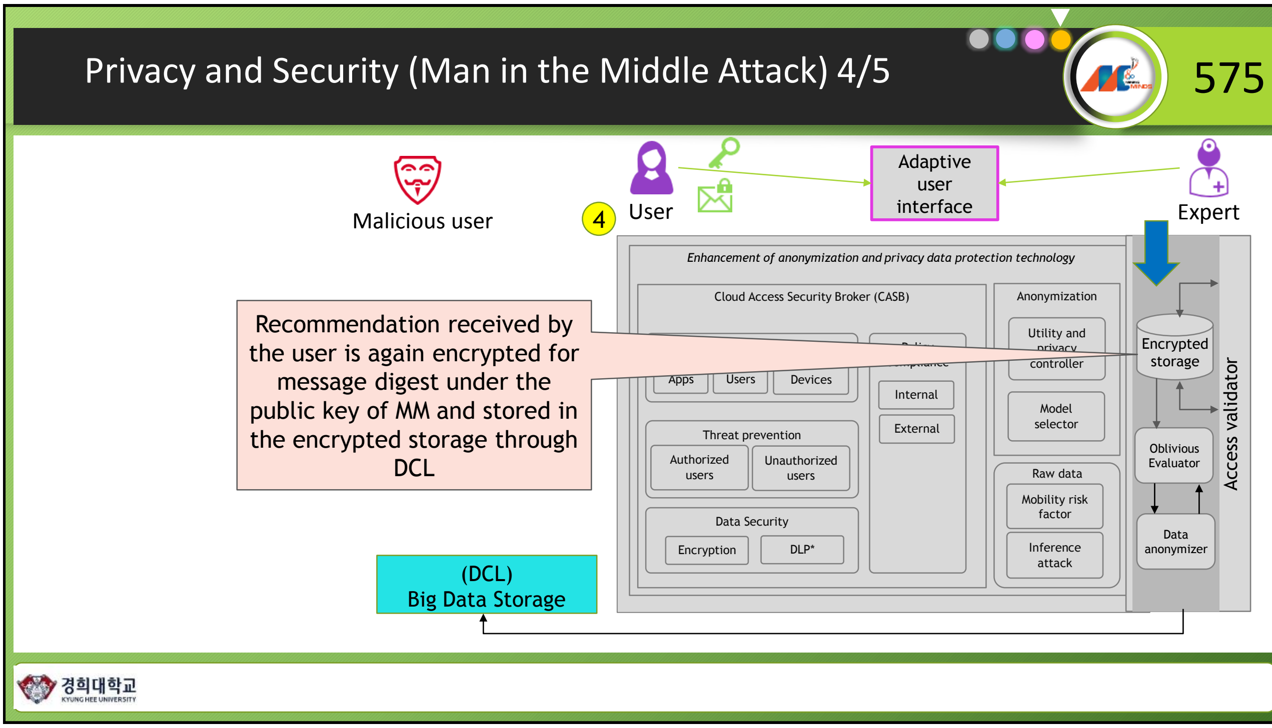
Privacy and Security (Man in the Middle Attack) 1/5
572
Malicious user
1
User
Adaptive user interface
Expert
User receives recommendations generated by MM or the expert
Enhancement of anonymization and privacy data protection technology
Cloud Access Security Broker (CASB)
Visibility
Apps
Users
Devices
Policy Compliance
Internal
External
Threat prevention
Authorized users
Unauthorized users
Data Security
Encryption
DLP*
Anonymization
Utility and privacy controller
Model selector
Raw data
Mobility risk factor
Inference attack
Encrypted storage
Oblivious Evaluator
Data anonymizer
Access validator
경희대학교
KYUNG HEE UNIVERSITY



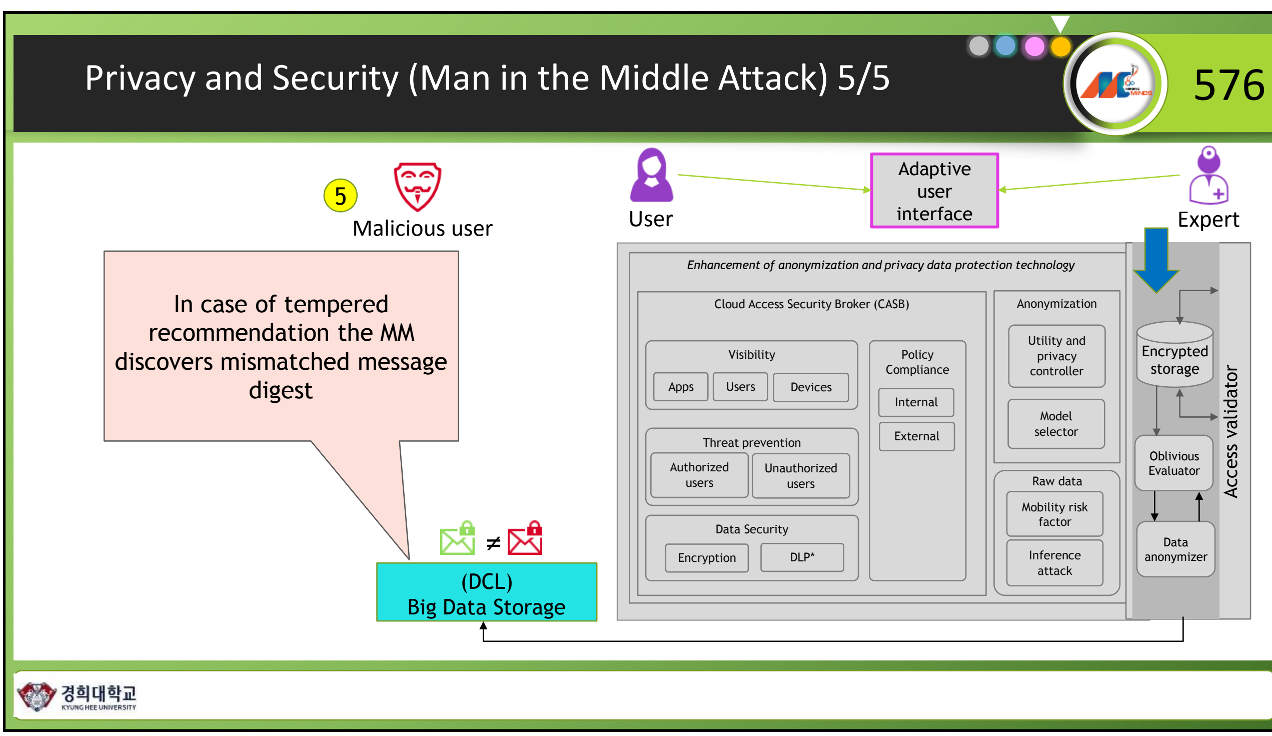
Privacy and Security (Man in the Middle Attack) 2/5
573
Malicious user
2
Assuming the ability of malicious user to intercept the recommendations and replaying them to the user with modification
User
Adaptive user interface
Expert
Enhancement of anonymization and privacy data protection technology
Cloud Access Security Broker (CASB)
Visibility
Apps
Users
Devices
Policy Compliance
Internal
External
Threat prevention
Authorized users
Unauthorized users
Data Security
Encryption
DLP*
Anonymization
Utility and privacy controller
Model selector
Raw data
Mobility risk factor
Inference attack
Encrypted storage
Oblivious Evaluator
Data anonymizer
Access validator
경희대학교
KYUNG HEE UNIVERSITY



Privacy and Security (Man in the Middle Attack) 3/5
574
Malicious user
User
Adaptive user interface
Expert
All recommendations generated by an expert are encrypted with MM public key and stored in the encrypted storage through DCL
3
Enhancement of anonymization and privacy data protection technology
Cloud Access Security Broker (CASB)
Apps
Users
Devices
Internal
External
Threat prevention
Authorized users
Unauthorized users
Data Security
Encryption
DLP*
Anonymization
controller
Model selector
Raw data
Mobility risk factor
Inference attack
Encrypted storage
Oblivious Evaluator
Data anonymizer
Access validator
(DCL)
Big Data Storage
경희대학교
KYUNG HEE UNIVERSITY



Privacy and Security (Man in the Middle Attack) 4/5
575
Malicious user
4
User
Adaptive user interface
Expert
Enhancement of anonymization and privacy data protection technology
Cloud Access Security Broker (CASB)
Anonymization
Recommendation received by the user is again encrypted for message digest under the public key of MM and stored in the encrypted storage through DCL
Apps
Users
Devices
Internal
External
Utility and privacy controller
Encrypted storage
Model selector
Threat prevention
Authorized users
Unauthorized users
Oblivious Evaluator
Access validator
Raw data
Mobility risk factor
Data Security
Encryption
DLP*
Inference attack
Data anonymizer
(DCL)
Big Data Storage
경희대학교
KYUNG HEE UNIVERSITY



Privacy and Security (Man in the Middle Attack) 5/5
576
5
Malicious user
User
Adaptive user interface
Expert
In case of tempered recommendation the MM discovers mismatched message digest
Enhancement of anonymization and privacy data protection technology
Cloud Access Security Broker (CASB)
Anonymization
Visibility
Apps
Users
Devices
Policy Compliance
Internal
External
Utility and privacy controller
Encrypted storage
Model selector
Threat prevention
Authorized users
Unauthorized users
Oblivious Evaluator
Access validator
Raw data
Mobility risk factor
Data Security
Encryption
DLP*
Inference attack
Data anonymizer
≠
(DCL)
Big Data Storage
경희대학교
KYUNG HEE UNIVERSITY

Tools and Technologies

577

- Java
- MD5 Hashing
- Public key cryptography (A-symmetric encryption)

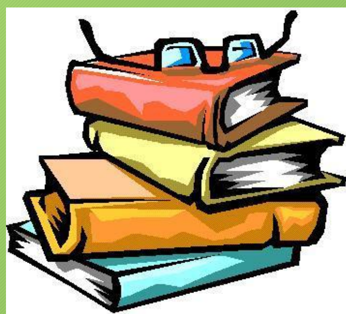
경희대학교
KYUNG HEE UNIVERSITY

Conclusion

578

- Ensuring data privacy and security have different considerations however are closely related
- The users data is sensitive in nature and has certain requirements for its storage and processing
- Besides data protection at rest, certain challenges of tempering, man in the middle attack and privacy exposure during data transmission are highlighted and resolved
- Ensuring data privacy and security in MM, achieves user confidence while using the offered services.

경희대학교
KYUNG HEE UNIVERSITY



REFERENCES

References- Lifelog Mapping and Representation

- **[1][Henricksen2002]** Henricksen, K., Indulska, J., Rakotonirainy, A. Modeling Context Information in Pervasive Computing Systems. In Pervasive Computing; Springer: Berlin, Germany, 2002; pp. 167–180.
- **[2][Henricksen2002]** Henricksen, K., Indulska, J. A software engineering framework for context-aware pervasive computing. In Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications, Orlando, FL, USA, 14–17 March 2004; pp. 77–86
- **[3][Gray2001]** Gray, P., Salber, D. Modelling and Using Sensed Context Information in the Design of Interactive Applications. In Engineering for Human-Computer Interaction; Springer: Berlin, Germany, 2001; pp. 317–335
- **[4][Chen2004]** Chen, H., Perich, F., Finin, T., Joshi, A. SOUPA: Standard ontology for ubiquitous and pervasive applications. In Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, Boston, MA, USA, 22 August 2004; pp. 258–267
- **[5][Petersen2012]** Petersen, A.K, Mikalsen, M. Context: Representation and Reasoning. Available Online: http://www.idi.ntnu.no/~anderpe/publications/RIA-05-AKP-MM-paper.pdf (accessed on 27 May 2014).
- **[6][Gu2004]** Gu, T., Pung, H.K. A middleware for building context-aware mobile services. In Proceedings of the 2004 IEEE 59th Vehicular Technology Conference, Milan, Italy, 17–19 May 2004; pp. 2656–2660
- **[7][Website]** OWL Web Ontology Language Overview. Available online: http://www.w3.org/TR/owl-features/ (accessed on 27 May 2014)
- **[8][Yamabe2005]** Yamabe,T., Takagi, A., Nakajima, T. Citron: A context information acquisition framework for personal devices. In Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, Hong Kong, China, 17–19 August 2005; pp. 489–495
- **[9][Bettini2010]** Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., Riboni, D. A survey of context modeling and reasoning techniques. Pervasive Mob. Comput. 2010, 6, 161–180

References- Lifelog Mapping and Representation

- **[10][Dey2001]** Dey, A.K., Abowd, G.D., Salber, D. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Hum. Comput. Interact. 2001, 16, 97–166.
- **[11][Castelli2007]** Castelli, G., Rosi, A., Mamei, M., Zambonelli, F. A simple model and infrastructure for context-aware browsing of the world. In Proceedings of the Fifth Annual IEEE International Conference on Pervasive Computing and Communications, White Plains, NY, USA, 19–23 March 2007; pp. 229–238
- **[12][Frank2011]** Frank, A.U. Tiers of ontology and consistency constraints in geographical information systems. Int. J. Geogr. Inf. Sci. 2001, 15, 667–678
- **[13][Henricksen2004]** Henricksen, K., Livingstone, S., Indulska, J. Towards a hybrid approach to context modeling, reasoning and interoperation. In Proceedings of the First International Workshop on Advanced Context Modelling, Reasoning and Management, Nottingham, UK, 7 September 2004
- **[14][Bardram2005]** Bardram, J.E. The Java Context Awareness Framework (JCAF)—A Service Infrastructure and Programming Framework for Context-Aware Applications. In Pervasive Computing; Springer: Berlin, Germany, 2005; pp. 98–115
- **[15][Mikalsen2006]** Mikalsen, M., Floch, J., Paspallis, N., Papadopoulos, G.A., Ruiz, P.A. Putting context in context: The role and design of context management in a mobility and adaptation enabling middleware. In Proceedings of the 7th International Conference on Mobile Data Management, (MDM 2006), Nara, Japan, 10–12 May 2006; pp. 76–83
- **[16][Park2007]** Park, Myung-Jae, et al. "An efficient and scalable management of ontology." Advances in Databases: Concepts, Systems and Applications. Springer Berlin Heidelberg, 2007. 975-980.
- **[17][Park2011]** Park, Jinsoo, Sunjoo Oh, and Joongho Ahn. "Ontology selection ranking model for knowledge reuse." Expert Systems with Applications 38.5 (2011): 5133-5144.

References-Streaming and Communication

- [1] Zaharia, Matei, et al. "Spark: cluster computing with working sets." Proceedings of the 2nd USENIX conference on Hot topics in cloud computing. 2010
- [2] Shahrivari, Saeed. "Beyond Batch Processing: Towards Real-Time and Streaming Big Data." arXiv preprint arXiv:1403.3375 (2014).
- [3] Lam, Wang, et al. "Muppet: MapReduce-style processing of fast data." Proceedings of the VLDB Endowment 5.12 (2012): 1814-1825
- [4] Pallickara, Shrideep, Jaliya Ekanayake, and Geoffrey Fox. "Granules: A lightweight, streaming runtime for cloud computing with support, for map-reduce." Cluster Computing and Workshops, 2009. CLUSTER'09. IEEE International Conference on. IEEE, 2009
- [5] Zinn, Daniel, et al. "Parallelizing XML data-streaming workflows via MapReduce." Journal of Computer and System Sciences 76.6 (2010): 447-463.
- [6] Ding, Mengwei, et al. "More convenient more overhead: the performance evaluation of Hadoop streaming." Proceedings of the 2011 ACM Symposium on Research in Applied Computation. ACM, 2011
- [7] Melnik, Sergey, et al. "Dremel: interactive analysis of web-scale datasets."*Proceedings of the VLDB Endowment* 3.1-2 (2010): 330-339.
- [8] Kornacker, Marcel, and Justin Erickson. "Cloudera Impala: real-time queries in Apache Hadoop, for real." 2012-10 [2013-02]. http://blog. cloudera. com/blog/2012/10/cloudera-impalareal-time-queries-in-apache-hadoop-for-real(2012).
- [9] https://storm.incubator.apache.org/
- [10] http://sfb876.tu-dortmund.de/sPublicPublicationFiles/bockermann_2014b.pdf p21-24
- [11] https://amplab.cs.berkeley.edu/
- [12] Hausenblas, Michael, and Jacques Nadeau. "Apache drill: interactive Ad-Hoc analysis at scale." Big Data 1.2 (2013): 100-104.
- [13] https://spark.apache.org/
- [14] https://github.com/apache/spark
- [15] http://csrc.nist.gov/groups/SMA/forum/documents/june2012presentations/fcsm_june2012_cooper_mell.pdf

References- Lifelog Monitoring and Prediction

583

- **[Yang 2015]** Yang, Pei-Ching, et al. "Life Record: A Smartphone-Based Daily Activity Monitoring System." Advances in Swarm and Computational Intelligence. Springer International Publishing, 2015. 380-385.
- **[Zini 2015]** Zini, Floriano, Martin Reinstadler, and Francesco Ricci. "Life-logs Aggregation for Quality of Life Monitoring." Proceedings of the 5th International Conference on Digital Health 2015. ACM, 2015
- **[Kwon 2014]** Kwon, Yongjin, et al. "Lifelog agent for human activity pattern analysis on Health Avatar platform." Healthcare informatics research 20.1 (2014): 69-75.
- http://www.wikihow.com/Use-Quantified-Self-Strategies-to-Improve-Your-Life
- http://www.limitsofebm.org/the-flaw-in-ebm/why-quantify-everything/
- **[Rabbi 2015]** Rabbi, Mashfiqui, et al. "Automated Personalized Feedback for Physical Activity and Dietary Behavior Change With Mobile Phones: A Randomized Controlled Trial on Adults." JMIR mHealth and uHealth 3.2 (2015): e42.
- **[Raghupathi2014]** Raghupathi, Wullianallur, and Viju Raghupathi. "Big data analytics in healthcare: promise and potential." Health Information Science and Systems 2.1 (2014): 3.
- **[Zazzi2010]** M. Zazzi, R. Kaiser, A. Sonnerborg, et al., "Prediction of response to antiretroviral therapy by human experts and by the EuResist data-driven expert system (the EVE study). " HIV Medicine, 2010.
- **[Husain2010]** Husain, W. and P. Lim Thean, "The development of personalized wellness therapy recommender system using hybrid case-based reasoning. Computer Technology and Development (ICCTD) ", 2010 2nd International Conference on.

경희대학교
KYUNG HEE UNIVERSITY

References- Lifelog Monitoring and Prediction

584

- **[Drăgulescu2007]** Drăgulescu, Doina, and Adriana Albu. "Medical Predictions System." Acta Polytechnica Hungarica 4, no. 3 (2007): 89-101.
- **[Pandey2012]** Dhiraj Pandey, Santosh kumar, "PREDICTION SYSTEM TO SUPPORT MEDICAL INFORMATION SYSTEM USING DATA MINING APPROACH", International Journal of Engineering Research and Applications (IJERA), Vol. 2, Issue 3, May-Jun 2012, pp.1988-1996.
- **[Yuan2014]** Yuan, Bingchuan, and John Herbert. "Context-aware hybrid reasoning framework for pervasive healthcare. Personal and Ubiquitous Computing ": April 2014, Volume 18, Issue 4, pp 865-881.
- **[Thomas2014]** Thomas, Diana M., et al. "Dynamic model predicting overweight, obesity, and extreme obesity prevalence trends." Obesity 22.2 (2014): 590-597.
- **[Schalkoff, Robert 2011]** Intelligent Systems: Principles, Paradigms and Pragmatics: Principles, Paradigms and Pragmatics. Jones & Bartlett Learning. ISBN 978-0-7637-8017-3**.**
- **[Moses2003]** Yoram; Vardi, Moshe Y; Fagin, Ronald; Halpern, Joseph Y (2003). "Reasoning About Knowledge. " MIT Press. ISBN 978-0-262-56200-3.
- **[Shelley2014]** Shelley L. Schmidt, MD; Nabihah Tayob, MS; Meilan K. Han, MD; et al. "Predicting Pulmonary Fibrosis Disease Course From Past Trends in Pulmonary Function." Chest. 2014;145(3):579-585.

경희대학교
KYUNG HEE UNIVERSITY

References- Inertial Activity Recognition

585

[1] Chun zhu, Weihua Sheng, "Human Daily Activity Recognition in Robust-assisted Living Using Multi-sensor Fusion", 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, May 12-17, 2009

[2] Jun-Ki Min, Sung-Bae Cho, "Activity Recognition Based on Wearable Sensors Using Selection/Fusion Hybrid Ensemble, 2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Anchorage, AK, Oct 9-12, 2011

[3] Lei Gao, Alan K. Bourke, John Nelson, "A System for Activity Recognition Using Multi-Sensor Fusion", 33rd Annual International Conference of the IEEE EMBS, Boston, Massachusetts USA, Aug 30 – Sep 3, 2011

[4] Ming Zeng, Xiao Wang, Le T. Nguyen, Pang Wu, Ole J. Mengshoel, Joy Zhang, "Adaptive Activity Recognition with Dynamic Heterogeneous Sensor Fusion", 6th International Conference on Mobile Computing, Applications and Services (MobiCASE), Austin, IX, United States, Nov 6-7, 2014

[5] Muhammad Shoaib, Stephan Bosch, Ozlem Durmaz Incel, Hans Scholten, Paul J.M.Havigna, "Fusion of Smartphone Motion Sensors for Physical Activity Recognition", Vol. 14, Issue 6, p.10146-10176, 2014

Reference- Video based Activity Recognition

586

- P. Dollar, V. Rabaud, G. Cottrell, S. Belongie, Behavior recognition via sparse spatio-temporal features, in: Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on, 2005, pp. 65–72.
- X. Wu, D. Xu, L. Duan, J. Luo, Y. Jia, Action recognition using multilevel features and latent structural svm, Circuits and Systems for Video Technology, IEEE Transactions on 23 (8) (2013) 1422–1431.
- Y. Chen, Z. Li, X. Guo, Y. Zhao, A. Cai, A spatio-temporal interest point detector based on vorticity for action recognition, in: Multimedia and Expo Workshops (ICMEW), 2013 IEEE International Conference on, 2013, pp. 1–6.
- H. Wang, C. Yuan, W. Hu, H. Ling, W. Yang, C. Sun, Action recognition using nonnegative action component representation and sparse basis selection, Image Processing, IEEE Transactions on 23 (2) (2014) 570–581.
- L. Wang, L. Cheng, T. H. Thi, J. Zhang, Human action recognition from boosted pose estimation, in: Digital Image Computing: Techniques and Applications (DICTA), 2010 International Conference on, 2010, pp. 308–313.
- E.-J. Weng, L.-C. Fu, On-line human action recognition by combining joint tracking and key pose recognition, in: Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, 2012, pp. 4112–4117.
- Y. Zheng, Y.-J. Zhang, X. Li, B.-D. Liu, Action recognition in still images using a combination of human pose and context information, in: Image Processing (ICIP), 2012 19th IEEE International Conference on, 2012, pp. 785–788.
- L. Liu, L. Shao, X. Zhen, X. Li, Learning discriminative key poses for action recognition, Cybernetics, IEEE Transactions on 43 (6) (2013) 1860–1870.

Reference- Video based Activity Recognition

587

- R. Alazrai, Y. Mowafi, C. G. Lee, Anatomical-plane-based representation for humanhuman interactions analysis, Pattern Recognition 48 (8) (2015) 2346 – 2363.
- Y. Kong, W. Liang, Z. Dong, Y. Jia, Recognising human interaction from videos by a discriminative model, Computer Vision, IET 8 (4) (2014) 277–286.
- Y. Kong, Y. Jia, Y. Fu, Interactive phrases: Semantic descriptions for human interaction recognition, Pattern Analysis and Machine Intelligence, IEEE Transactions on 36 (9) (2014) 1775–1788.
- M. Ryoo, J. Aggarwal, Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities, in: Computer Vision, 2009 IEEE 12th International Conference on, 2009, pp. 1593–1600.
- L. Meng, L. Qing, P. Yang, J. Miao, X. Chen, D. Metaxas, Activity recognition based on semantic spatial relation, in: Pattern Recognition (ICPR), 2012 21st International Conference on, 2012, pp. 609–612.
- U. Gaur, Y. Zhu, B. Song, A. Roy-Chowdhury, A string of feature graphs model for recognition of complex activities in natural videos, in: Computer Vision (ICCV), 2011 IEEE International Conference on, 2011, pp. 2595–2602.
- Y. Wang, G. Mori, Human action recognition by semilatent topic mo dels, Pattern Analysis and Machine Intelligence, IEEE Transactions on 31 (10) (2009) 1762–1774.
- D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
- S. Yang, C. Yuan, W. Hu, X. Ding, A hierarchical model based on latent dirichlet allocation for action recognition, in: Pattern Recognition (ICPR), 2014 22nd International Conference on, 2014, pp. 2613–2618.

References- Activity based Emotion Recognition

588

[1] Burkhardt, F., Paeschke, A., Rolfes, M., Sendlmeier, W., & Weiss, B. (2005). A Database of German Emotional Speech (pp. 1517–1520). Proc. Interspeech.
[2] Burger, S., MacLaren, V., & Yu, H. (2002). The ISL Meeting Corpus: The Impact of Meeting Type on Speech Style. Proc. Eighth Int'l Conf. Spoken Language Processing.
[3] Roisman, G. I., Tsai, J. L., & Chiang, K. S. (2004). The Emotional Integration of Childhood Experience: Physiological, Facial Expressive, and Self-Reported Emotional Response during the Adult Attachment Interview. Developmental Psychology, 40(5), 776–789. doi:10.1037/0012-1649.40.5.776
[4] Douglas-Cowie, E., Campbell, N., Cowie, R., & Roach, P. (2003). Emotional Speech: Towards a New Generation of Database. Speech Communication, 40(1/2), 33–60. doi:10.1016/S0167- 6393(02)00070-5
[5] Busso, C., Deng, Z., Yildirim, S., Bulut, M., Lee, C. M., Kazemzadeh, A., et al. (2004). Analysis of Emotion Recognition Using Facial Expressions. Speech and Multimodal Information. Proc. ACM Int'l Conf. Multimodal Interfaces (pp. 205-211).
[6] Haq, S., & Jackson, P. J. B. (2009). SpeakerDependent Audio-Visual Emotion Recognition. Proc. Auditory-Visual Speech Processing.
[7] Borchert, M., & Düsterhöft, A. (2005). Emotions in Speech – Experiments with Prosody and Quality Features in Speech for Use in Categorical and Dimensional Emotion Recognition Environments. Proc. IEEE Int'l Conf. on Natural Language Processing and Knowledge
[8] Luengo, I., & Navas, E. (2005). Automatic Emotion Recognition using Prosodic Parameters (pp. 493–496). Proc. Interspeech.Engineering (pp. 147–151).
[9] Chi-Chun Lee, Emily Mower, Carlos Busso, Sungbok Lee, Shrikanth Narayanan, Emotion recognition using a hierarchical binary decision tree approach, Speech Communication, Volume 53, Issues 9–10, November–December 2011, Pages 1162-1171, ISSN 0167-6393, http://dx.doi.org/10.1016/j.specom.2011.06.004.

경희대학교
KYUNG HEE UNIVERSITY

References- Activity based Emotion Recognition

589

[10] Schuller, B.; Vlasenko, B.; Eyben, F.; Rigoll, G.; Wendemuth, A., "Acoustic emotion recognition: A benchmark comparison of performances," Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on , vol., no., pp.552,557, Nov. 13 2009-Dec. 17 2009 doi: 10.1109/ASRU.2009.5372886

[11] Kunxia Wang; Ning An; Bing Nan Li; Yanyong Zhang; Lian Li, "Speech Emotion Recognition Using Fourier Parameters," Affective Computing, IEEE Transactions on , vol.6, no.1, pp.69,75, Jan.-March 1 2015
doi: 10.1109/TAFFC.2015.2392101

[12] Schuller, B., Muller, R., Hornler, B., Hothker, A., Konosu, H., & Rigoll, G. (2007). Audiovisual Recognition of Spontaneous Interest within Conversations Proc. ACM Int'l Conf. Multimodal Interfaces (pp. 30-37).

[13] Soujanya Poria, Erik Cambria, Amir Hussain, Guang-Bin Huang, Towards an intelligent framework for multimodal affective data analysis, Neural Networks, Volume 63, March 2015, Pages 104-116, ISSN 0893-6080, http://dx.doi.org/10.1016/j.neunet.2014.10.005.

[14] T. Vogt, E. André and N. Bee, "EmoVoice - A framework for online recognition of emotions from voice," in Proceedings of Workshop on Perception and Interactive Technologies for Speech-Based Systems, 2008.

[15] EMOSpeech http://emospeech.net/

[16] Moodies Emotions Analysis https://itunes.apple.com/us/app/moodies-emotions-analytics/id793945393?mt=8

[17] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

[18] Florian Eyben, Felix Weninger, Florian Gross, Björn Schuller: "Recent Developments in openSMILE, the Munich Open-Source Multimedia Feature Extractor", In Proc. ACM Multimedia (MM), Barcelona, Spain, ACM, ISBN 978-1-4503-2404-5, pp. 835-838, October 2013

경희대학교
KYUNG HEE UNIVERSITY

References- High Level Context Awareness

590

- **[1][G. Chen]** X. Ding, K. Huang, X. Ye, and C. Zhang, "Changing Health Behaviors through Social and Physical Context Awareness," pp. 663–667, 2015.
- **[2][D. Moen]** N. Mckelvey, and K. Curran, "Context Awareness in Mobile Devices," pp. 247–252, 2015.
- **[3][C. Gurrin]** A. F. Smeaton, and A. R. Doherty, "LifeLogging: Personal Big Data," Found. Trends® Inf. Retr., vol. 8, no. 1, pp. 1–125, 2014.
- **[4][Bellavista]** Paolo, et al. "A survey of context data distribution for mobile ubiquitous systems." ACM Computing Surveys (CSUR) 44.4 (2012): 24.
- **[5][H. Lee]** J. Kwon, "Ontology model-based situation and socially-aware health care service in a smart home environment," Int. J. Smart Home, vol. 7, no. 5, pp. 239–250, 2013.
- **[6][C. Perera]** A. Zaslavsky, P. Christen, M. Compton, and D. Georgakopoulos, "Context-aware sensor search, selection and ranking model for internet of things middleware," Proc. - IEEE Int. Conf. Mob. Data Manag., vol. 1, pp. 314–322, 2013.
- **[7][A. M. Khattak]** N. Akbar, M. Aazam, T. Ali, A. M. Khan, S. Jeon, M. Hwang, and S. Lee, "Context Representation and Fusion: Advancements and Opportunities," pp. 9628–9668, 2014.
- **[8][Strang]** Thomas, and Claudia Linnhoff-Popien. "A context modeling survey." Workshop Proceedings. 2004.
- **[9][Dey]** Anind K. "Understanding and using context." Personal and ubiquitous computing 5.1 (2001): 4-7.
- **[10][Abowd]** Gregory D., et al. "Towards a better understanding of context and context-awareness." Handheld and ubiquitous computing. Springer Berlin Heidelberg, 1999.
- **[11][Schilit]** Bill, Norman Adams, and Roy Want. "Context-aware computing applications." Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on. IEEE, 1994.
- **[12][Wang]** X. H., et al. "Ontology Based Context Modeling and Reasoning using OWL, in 'PERCOMW'04: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops'." IEEE Computer Society, Washington, DC, USA (2004): 18.

경희대학교
KYUNG HEE UNIVERSITY

References- Knowledge Curation Layer

591

- [1] Sánchez, L. E., Diaz-Pace, J. A., Zunino, A., Moisan, S., and Rigault, J. P., "An Approach for Managing Quality Attributes at Runtime using Feature Models," In *Software Components, Architectures and Reuse (SBCARS), 2014 Eighth Brazilian Symposium on*, pp. 11-20. IEEE, September, 2014.
- [2] Vranić, V., "Reconciling feature modeling: A feature modeling Metamodel," In *Object-Oriented and Internet-Based Technologies,* pp. 122-137, Springer Berlin Heidelberg, 2004.
- [3] Kwiatkowska, E.J.; Fargion, G.S., "Application of machine-learning techniques toward the creation of a consistent and calibrated global chlorophyll concentration baseline dataset using remotely sensed ocean color data," Geoscience and Remote Sensing, IEEE Transactions on , vol.41, no.12, pp.2844-2860, Dec. 2003.
- [4] Dimitriadis, S.; Goumopoulos, C., "Applying Machine Learning to Extract New Knowledge in Precision Agriculture Applications," Informatics. PCI '08. pp.100-104, 2008
- [5] Smith, K. A.; Woo, F.; Ciesielski, V.; Ibrahim, R., "Matching data mining algorithm suitability to data characteristics using a self-organising map," 2002.
- [6] Ali, S. and Smith, K., "On learning algorithm selection for classification", Appl. Soft Comput, 6(2), 119–138, 2006.
- [7] Song, Qinbao, Guangtao Wang, and Chao Wang. "Automatic recommendation of classification algorithms based on data set characteristics" Pattern recognition 45, no. 7 (2012): 2672-2689.
- [8] Wang, Guangtao, Qinbao Song, Xueying Zhang, and Kaiyuan Zhang. "A generic multilabel learning-based classification algorithm recommendation method." ACM Transactions on Knowledge Discovery from Data (TKDD), 9(1), 2014.
- [9] Bachman, R. E.; Hoffman, R. D.; Johnson, V. M.; McDavid, D. W.; Mazina, D. I., "Search engine facility with automated knowledge retrieval, generation and maintenance." U.S. Patent No. 7,216,121. 8 May 2007.
- [10] Georg, Gersende, Brigitte Séroussi, and Jacques Bouaud. "Extending the GEM model to support knowledge extraction from textual guidelines." *International journal of medical informatics* 74.2 (2005): 79-87.

경희대학교 KYUNG HEE UNIVERSITY UTAS

References [Recommendation Builder]

592

- **[Yuan2014]**, Bingchuan, and John Herbert. "Context-aware hybrid reasoning framework for pervasive healthcare." *Personal and ubiquitous computing* 18, no. 4 (2014): 865-881.
- [**Zazzi2010**] M. Zazzi, R. Kaiser, A. Sonnerborg, et al., Prediction of response to antiretroviral therapy by human experts and by the EuResist data-driven expert system (the EVE study). HIV Medicine, 2010.
- [**Husain2010**] Husain, W. and P. Lim Thean, The development of personalized wellness therapy recommender system using hybrid case-based reasoning. Computer Technology and Development (ICCTD), 2010 2nd International Conference on.
- [**Drăgulescu2007**] Drăgulescu, Doina, and Adriana Albu. "Medical Predictions System." Acta Polytechnica Hungarica 4, no. 3 (2007): 89-101.
- [**Pandey2012**] Dhiraj Pandey, Santosh kumar, "PREDICTION SYSTEM TO SUPPORT MEDICAL INFORMATION SYSTEM USING DATA MINING APPROACH", International Journal of Engineering Research and Applications (IJERA), Vol. 2, Issue 3, May-Jun 2012, pp.1988-1996.
- [**Yuan2014**] Yuan, Bingchuan, and John Herbert. "Context-aware hybrid reasoning framework for pervasive healthcare. Personal and Ubiquitous Computing: April 2014, Volume 18, Issue 4, pp 865-881.
- **[CASNET]** http://web.cs.wpi.edu/~dcb/courses/CS538/documents/2002/CASNET-slides.pdf
- **[MYCIN]** http://web.onetel.net.uk/~hibou/ai-course/mycin.txt
- **[Schalkoff, Robert 2011]** *Intelligent Systems: Principles, Paradigms and Pragmatics: Principles, Paradigms and Pragmatics*. Jones & Bartlett Learning. ISBN 978-0-7637-8017-3.
- **[Moses2003]**, Yoram; Vardi, Moshe Y; Fagin, Ronald; Halpern, Joseph Y (2003). *Reasoning About Knowledge*. MIT Press. ISBN 978-0-262-56200-3.
- **[Khan2003],** Khan, Abdus Salam, and Achim Hoffmann. "Building a case-based diet recommendation system without a knowledge engineer." *Artificial Intelligence in Medicine* 27, no. 2 (2003): 155-179.
- **[Trewin200]** Trewin, Shari. "Knowledge-based recommender systems." *Encyclopedia of Library and Information Science: Volume 69-Supplement 32* (2000): 180.
- **[Evans-Romaine3003]** Evans-Romaine, Kathleen, and Cindy Marling. "Prescribing exercise regimens for cardiac and pulmonary disease patients with CBR." In *Workshop on CBR in the health sciences at 5th international conference on case-based reasoning (ICCBR-03)*, pp. 45-62. 2003.
- **[Lim2013],** Lim, Thean Pheng, Wahidah Husain, and Nasriah Zakaria. "Recommender System for Personalised Wellness Therapy."
- [**Minutolo2010**] Minutolo, Aniello, Giovanna Sannino, Massimo Esposito, and Giuseppe De Pietro. "A rule-based mHealth system for cardiac monitoring." In *Biomedical Engineering and Sciences (IECBES), 2010 IEEE EMBS Conference on*, pp. 144-149. IEEE, 2010.

경희대학교 KYUNG HEE UNIVERSITY

References [Recommendation Interpreter]

593

- [Azumio 2015] Argus, https://itunes.apple.com/us/app/argus-pedometer-gps-activity/id624329444?mt=8, Accessed on July 12, 2015
- [Yuan 2014] Yuan, Bingchuan, and John Herbert. "Context-aware hybrid reasoning framework for pervasive healthcare." *Personal and ubiquitous computing* 18.4 (2014): 865-881.
- [Personalization2015] Personalization, http://www.quora.com/What-is-the-definition-of-personalization, visited on Feb 23 2015
- [Burke2002] Burke R., Hybrid Recommender Systems: Survey and Experiments, User Modeling and User-Adapted Interaction, v.12 n.4, p.331-370, 2002
- [Chung2007] Chung, R., Sundaram, D. and Srinivasan, A. ,Integrated Personal Recommender Systems. ICEC‟07, 2007
- [Elias2013] Elias, Alexander P. "Personalized cross-domain recommender system." U.S. Patent Application 13/781,297.
- [Jannach2010] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, Recommender Systems - An Introduction. Cambridge University Press (CUP), 2010.
- [Tintarev2007] N. Tintarev, "Explanations of recommendations," in Proceedings of the 2007 ACM Conference on Recommender Systems (RecSys'07). Minneapolis, MN, USA: ACM, 2007, pp. 203–206.
- Tintarev2007] N. Tintarev and J. Masthoff, "Effective explanations of recommendations: user-centered design," in Proceedings of the 2007 ACM Conference on Recommender Systems (RecSys'07). Minneapolis, MN, USA: ACM, 2007, pp. 153–156.
- [Fogg1999] B. J. Fogg, "Persuasive technologies," Communications of the ACM, vol. 42, no. 5, pp. 26–29, 1999.
- [Grabner2003] S. Grabner-Kr¨auter and E. A. Kaluscha, "Empirical research in on-line trust: a review and critical assessment," International Journal of Human-Computer Studies, vol. 58, no. 6, pp. 783–812, 2003.
- [Fernandez2011] Fernández-Tobías, Ignacio, et al. "A generic semantic-based framework for cross-domain recommendation." Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems. ACM, 2011.
- [AZAK2010] AZAK, MUSTAFA. CrosSing: A framework to develop knowledge-based recommenders in cross domains. Diss. MIDDLE EAST TECHNICAL UNIVERSITY, 2010.

References- Descriptive Analytics

594

1. [Glatz2014] Glatz, Eduard, et al. "Visualizing big network traffic data using frequent pattern mining and hypergraphs." *Computing* 96.1 (2014): 27-38.
2. [Gehlenborg2010] Gehlenborg, Nils, et al. "Visualization of omics data for systems biology." Nature methods 7 (2010): S56-S68.
3. [Bennett2009] Bennett, Jeremy S., "Massive model visualization: An investigation into spatial partitioning" (2009). Graduate Theses and Dissertations. Paper 10546.
4. [Dancy2008] Elizabeth Dancy, Uliyana Markova "Autonomic computing in Canadian academia"
5. [Dietrich2007] Dietrich, Andreas, Enrico Gobbetti, and Sung-Eui Yoon. "Massive-model rendering techniques: a tutorial." IEEE Computer Graphics and Applications27.6 (2007): 20-34.
6. [Shi2004] Shi, Rui, et al. "An adaptive image content representation and segmentation approach to automatic image annotation." *Image and Video Retrieval*. Springer Berlin Heidelberg, 2004. 545-554.
7. [McAdams2000] McAdams, Daniel A., and Kristin L. Wood. "Tuning parameter tolerance design: foundations, methods, and measures." Research in Engineering Design 12.3 (2000): 152-162.
8. Chang, Ya-Ting, and Shih-Wei Sun. "A realtime interactive visualization system for knowledge transfer from social media in a big data." *Information, Communications and Signal Processing (ICICS) 2013 9th International Conference on*. IEEE, 2013.
9. Kandogan, E., "Just-in-time annotation of clusters, outliers, and trends in point-based data visualizations," *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on* , vol., no., pp.73,82, 14-19 Oct. 2012

References-UI/IX Authoring Tool

595

1. J. Orwant. Doppelg¨ anger–a user modeling system. PhD thesis, Massachusetts Institute of Technology, 1991.
2. G. Brajnik and C. Tasso. A shell for developing non-monotonic user modeling systems. International Journal of Human-Computer Studies, 40(1):31–62, 1994.
3. A. Paiva and J. Self. Tagus—a user and learner modeling workbench. User Mod- eling and User-Adapted Interaction, 4(3):197–226, 1994.
4. J. Kay. The um toolkit for cooperative user modelling. User Modeling and User- Adapted Interaction, 4(3):149–196, 1994
5. W. Pohl. Logic-based representation and reasoning for user modeling shell systems. User Modeling and User-Adapted Interaction, 9(3):217–282, 1999.
6. G. Fischer. User modeling in human–computer interaction. User modeling and user-adapted interaction, 11(1):65–86, 2001. ISSN 0924-1868.
7. A. Kobsa. Generic user modeling systems. User modeling and user-adapted interaction, 11(1):49–63, 2001.
8. P. Gregor, A. F. Newell, and M. Zajicek. Designing for dynamic diversity: inter- faces for older people. In Proceedings of the fifth international ACM conference on Assistive technologies, page 151–156, 2002.
9. S. Gauch, J. Chaffee, and A. Pretschner. Ontology-based personalized search and browsing. Web Intelligence and Agent Systems, 1(3):219–234, 2003. URL
10. L. Razmerita, A. Angehrn, and A. Maedche. Ontology-based user mod- eling for knowledge management systems. In User Modeling 2003, page 213–217. Springer, 2003.
11. M. Hatala and R.Wakkary. Ontology-based user modeling in an augmented audio reality system for museums. User Modeling and User-Adapted Interaction, 15 (3-4):339–380, 2005. URL
12. F. Pereira. A triple user characterization model for video adaptation and quality of experience evaluatio. In Multimedia Signal Processing, 2005 IEEE 7th Workshop on, page 1–4, 2005. URL
13. D. Heckmann, T. Schwartz, B.s Brandherm, M. Schmitz, and M. vonWilamowitz- Moellendorff. Gumo–the general user model ontology. In User Modeling 2005, pages 428–432. Springer, 2005.

경희대학교 KYUNG HEE UNIVERSITY

References-UI/UX Authoring Tool

596

14. U. Persad, P. Langdon, and J. Clarkson. Characterising user capabilities to support inclusive design evaluation. Universal Access in the Information Society, 6(2): 119–135, 2007
15. P. Gregor, A. F. Newell, and M. Zajicek. Designing for dynamic diversity: inter- faces for older people. In Proceedings of the fifth international ACM conference on Assistive technologies, page 151–156, 2002
16. R. Casas, R. Blasco Mar´ ın, A. Robinet, A. Delgado, A. Yarza, J. Mcginn, R. Picking, and V. Grout. User modelling in ambient intelligence for elderly and dis- abled people. Computers Helping People with Special Needs, page 114–122, 2008. URL
17. C. Evers, R. Kniewel, K. Geihs, and L. Schmidt. Achieving user participation for adaptive applications. In Ubiquitous Computing and Ambient Intelligence, Lec- ture Notes in Computer Science, pages 200–207. Springer Berlin Heidelberg, Jan- uary 2012. ISBN
18. K. L. Skillen, L. Chen, C. D. Nugent, M. P. Donnelly, W. Burns, and I. Solheim. Ontological user profile modeling for context-aware application personalization. In Ubiquitous Computing and Ambient Intelligence 261–268. Springer, 2012. URL
19. http://myui.eu/
20. http://www.ibm.com/developerworks/library/wa-uihealth/
21. Bongartz, Sara, et al. "Adaptive User Interfaces for Smart Environments with the Support of Model-Based Languages." Ambient Intelligence. Springer Berlin Heidelberg, 2012. 33-48.
22. Jason, Bronwin, Andre Calitz, and Jean Greyling. "The evaluation of an adaptive user interface model." Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists. ACM, 2010.
23. http://www.w3.org/standards/techs/ccpp#w3c_all
24. Uaprofindex. http://www.w3.org/wiki/UAProfIndex
25. Wurfl - the mobile device database. http://wurfl.sourceforge.net/

경희대학교 KYUNG HEE UNIVERSITY

References-Privacy and Security

1. Privacy and Security in Mobile Health (mHealth) Researchhttp://www.ncbi.nlm.nih.gov/pmc/articles/PMC4432854/
2. Taxonomy of mHealth applications for Privacy and security Miloslava Plachkinova et al. A taxonomy of mHealth Apps – Security and Privacy-2015
3. Adhikari, Richards and Scott- Security and Privacy Issues Related to the Use of Mobile Health Apps, 25th Australasian Conference on Information Systems
4. Albert Lee and Mike Lee My Fitness pall , https://www.myfitnesspal.com/
5. Endomondo, a personal trainer in your pocket- https://www.endomondo.com/
6. iTriage – Fidns the closest medical service based on the symptoms and health condition - http://www.aetna.com/docfind/cms/assets/pdf/united/itriage.pdf
7. Everyone Every run , Run keeper, https://runkeeper.com/
8. Man in the middle attack http://www.veracode.com/security/man-middle-attack
9. Man in the middle attack by OWASP https://www.owasp.org/index.php/Man-in-the-middle_attack
10. Rivest, Ronald MD5 message digest algorithm - https://tools.ietf.org/html/rfc1321?ref=driverlayer.com
11. Quantify your day-to-day http://www.azumio.com/s/argus/index.html

경희대학교
KYUNG HEE UNIVERSITY

Section 4-D

Micro-Demos

Mining Minds Platform Ver. 2.5

STENCILS for Layers Micro-Demos

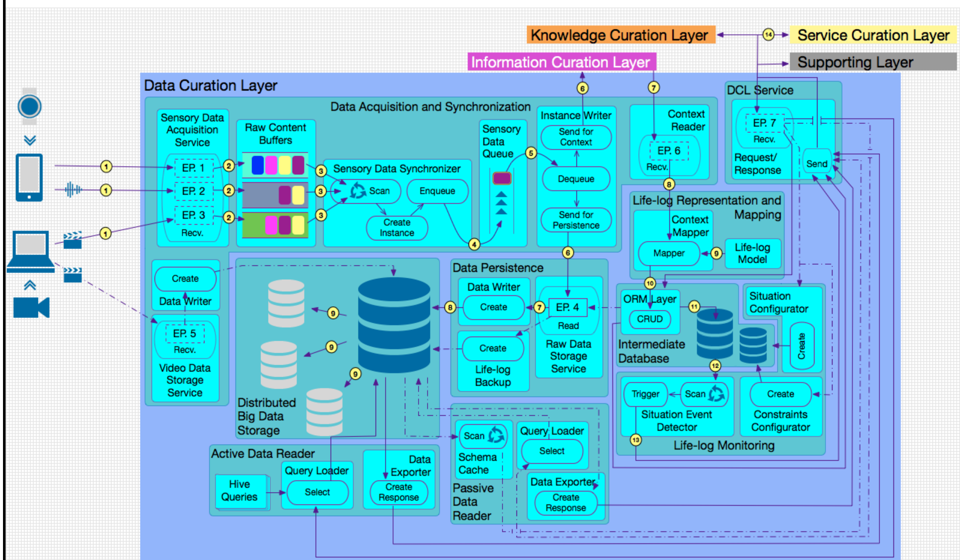
Professor Sungyoung Lee

February 3, 2016

Ubiquitous Computing Laboratory
Kyung Hee University, Korea

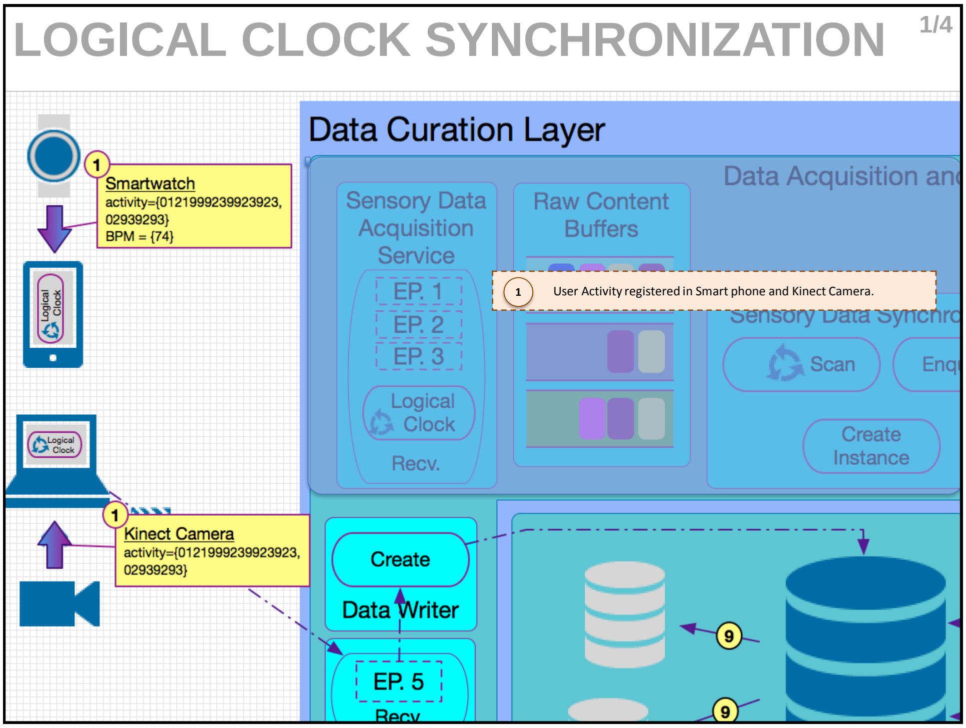
STENCILS for DCL Micro-Demos

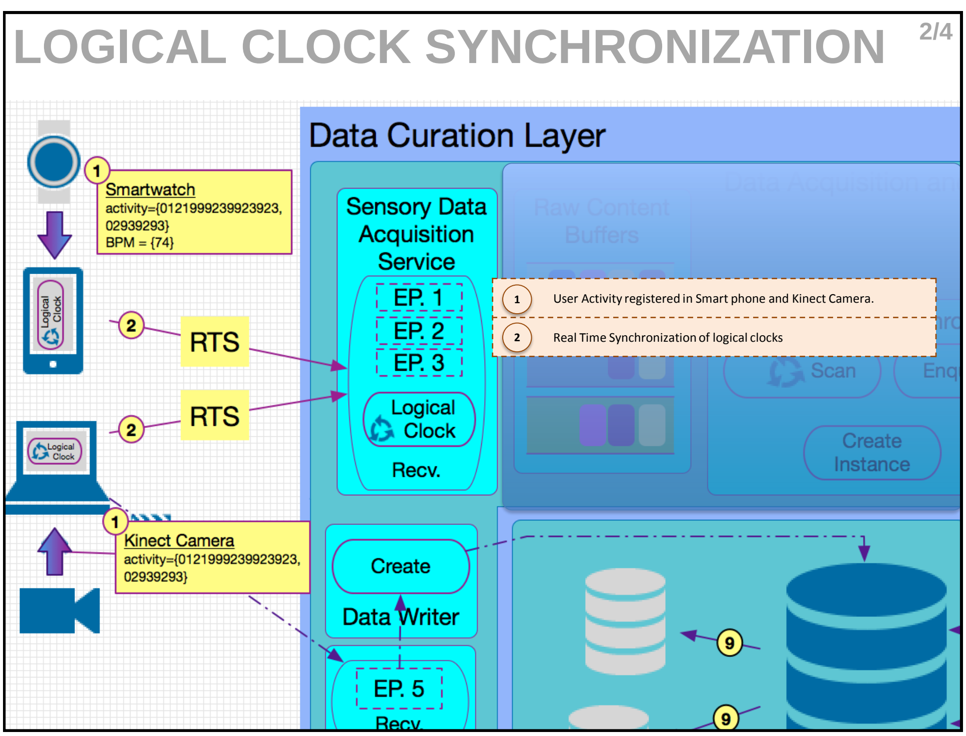
DCL Detailed Architecture

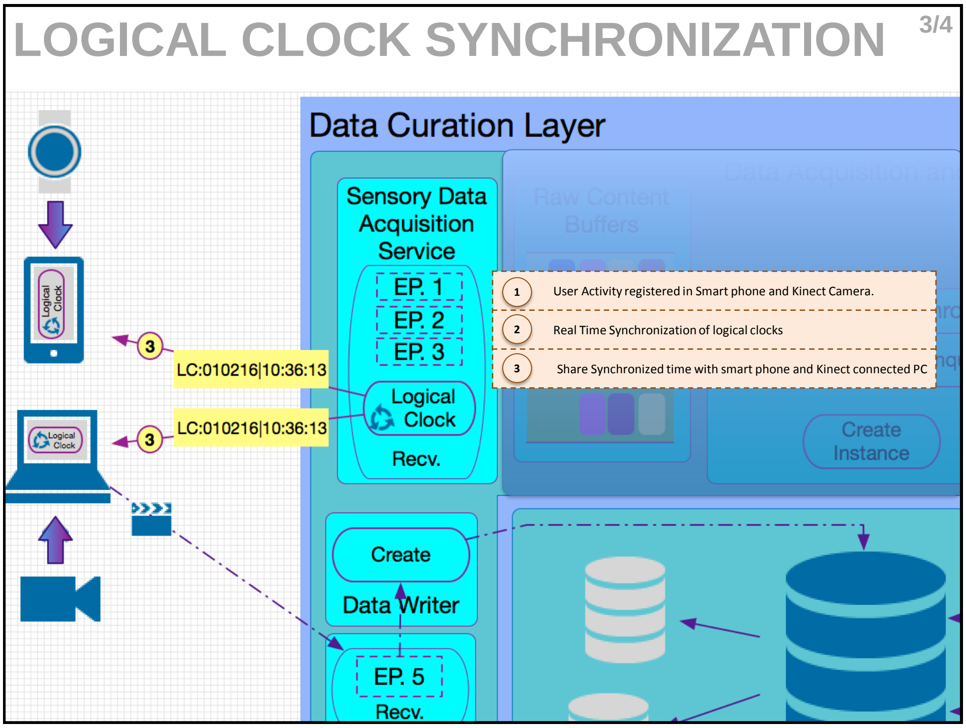


Scenario 1: Real-time

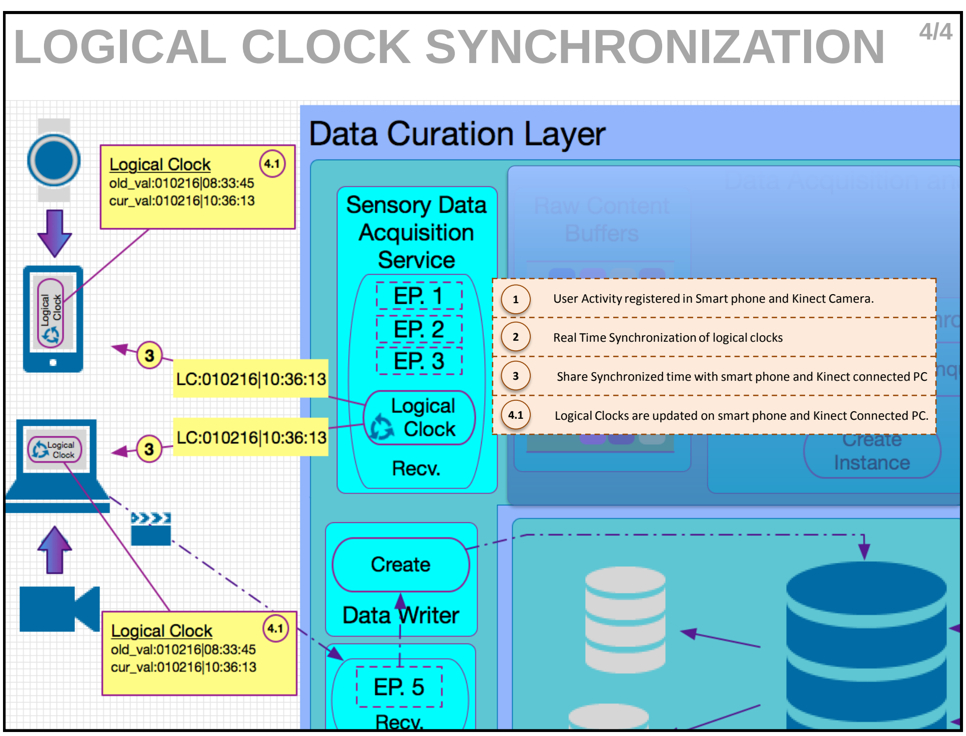
Desc. : Raw-sensory data from multimodal data sources is **acquisitioned**, and **persisted**. identified context on the sensory data is **mapped** to user life-log and **monitored** for situation detection.


LOGICAL CLOCK SYNCHRONIZATION
1/4
Data Curation Layer
Data Acquisition and
Sensory Data Acquisition Service
EP. 1
EP. 2
EP. 3
Logical Clock
Recv.
Raw Content Buffers
1 User Activity registered in Smart phone and Kinect Camera.
Scan
Create Instance
1 Smartwatch activity={01219992399923923, 02939293} BPM = {74}
1 Kinect Camera activity={01219992399923923, 02939293}
Create
Data Writer
EP. 5
Recv.
9
9

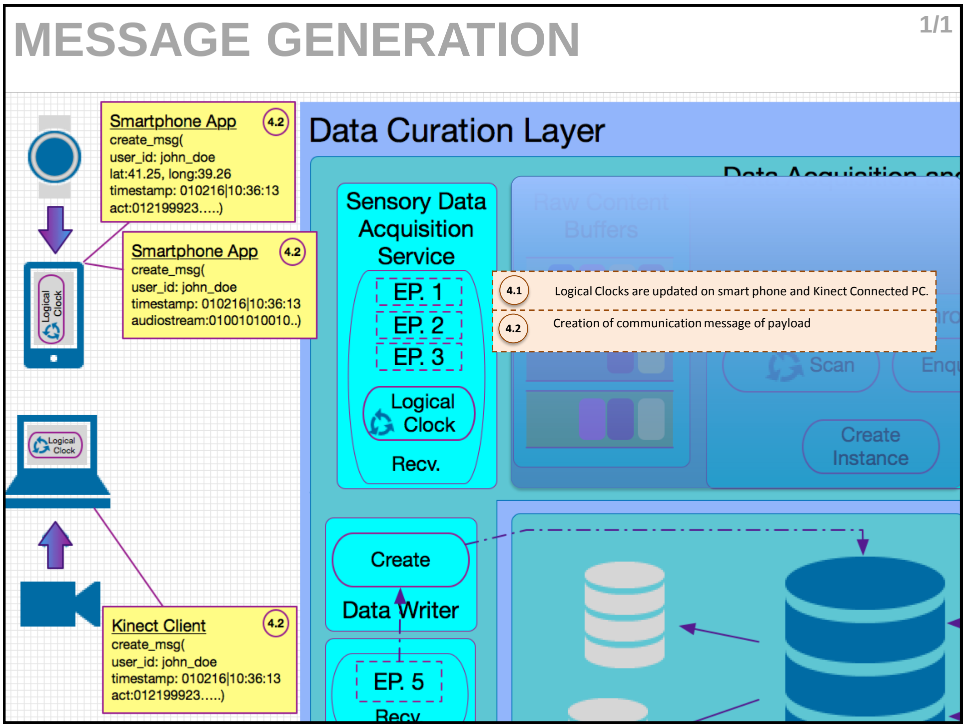

LOGICAL CLOCK SYNCHRONIZATION
2/4
Data Curation Layer
Sensory Data Acquisition Service
EP. 1
EP. 2
EP. 3
Logical Clock
Recv.
Raw Content Buffers
1 User Activity registered in Smart phone and Kinect Camera.
2 Real Time Synchronization of logical clocks
Scan
Create Instance
1 Smartwatch activity={01219992399923923, 02939293} BPM = {74}
2 RTS
2 RTS
1 Kinect Camera activity={01219992399923923, 02939293}
Create
Data Writer
EP. 5
Recv.
9
9

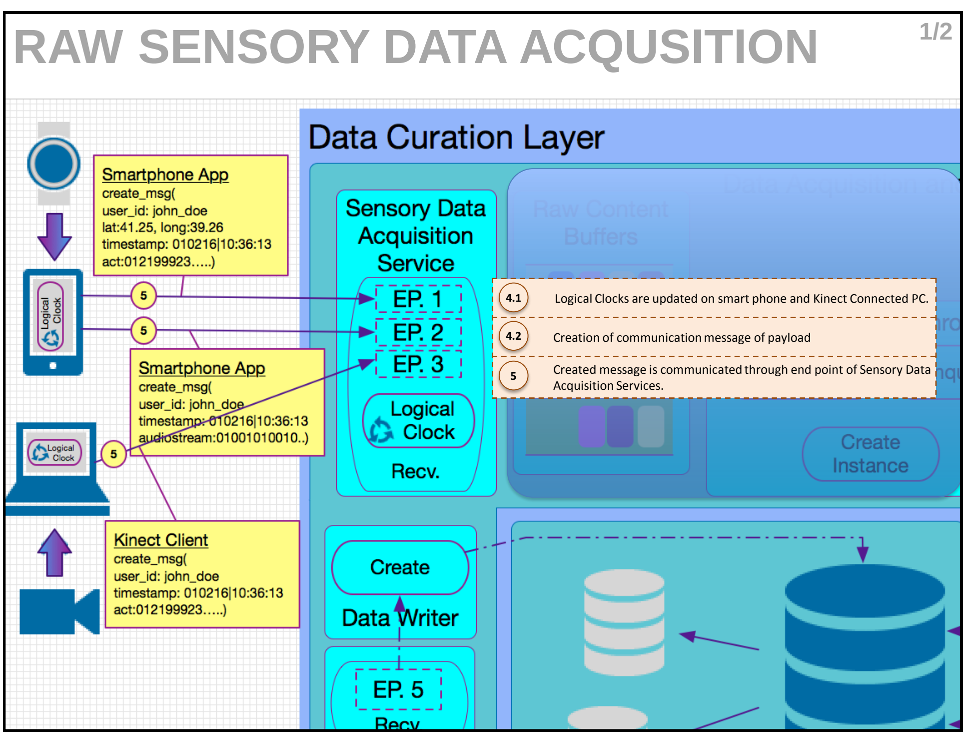


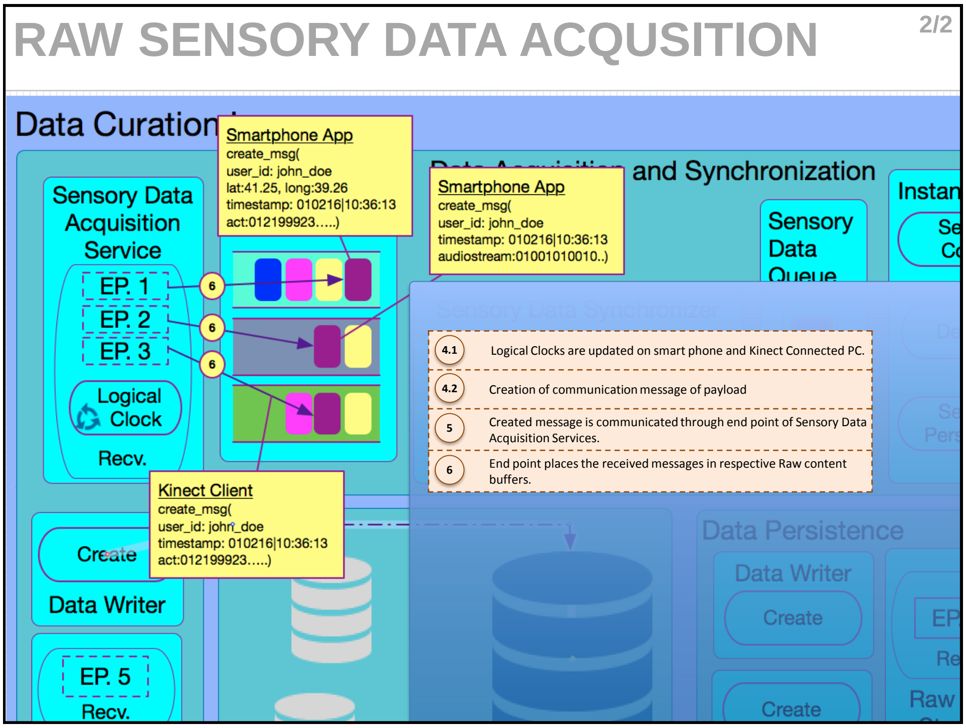
LOGICAL CLOCK SYNCHRONIZATION
3/4
Data Curation Layer
Sensory Data Acquisition Service
EP. 1
EP. 2
EP. 3
Logical Clock
Recv.
Raw Content Buffers
Create Instance
Logical Clock
LC:010216|10:36:13
LC:010216|10:36:13
3
3
Create
Data Writer
EP. 5
Recv.
1 User Activity registered in Smart phone and Kinect Camera.
2 Real Time Synchronization of logical clocks
3 Share Synchronized time with smart phone and Kinect connected PC

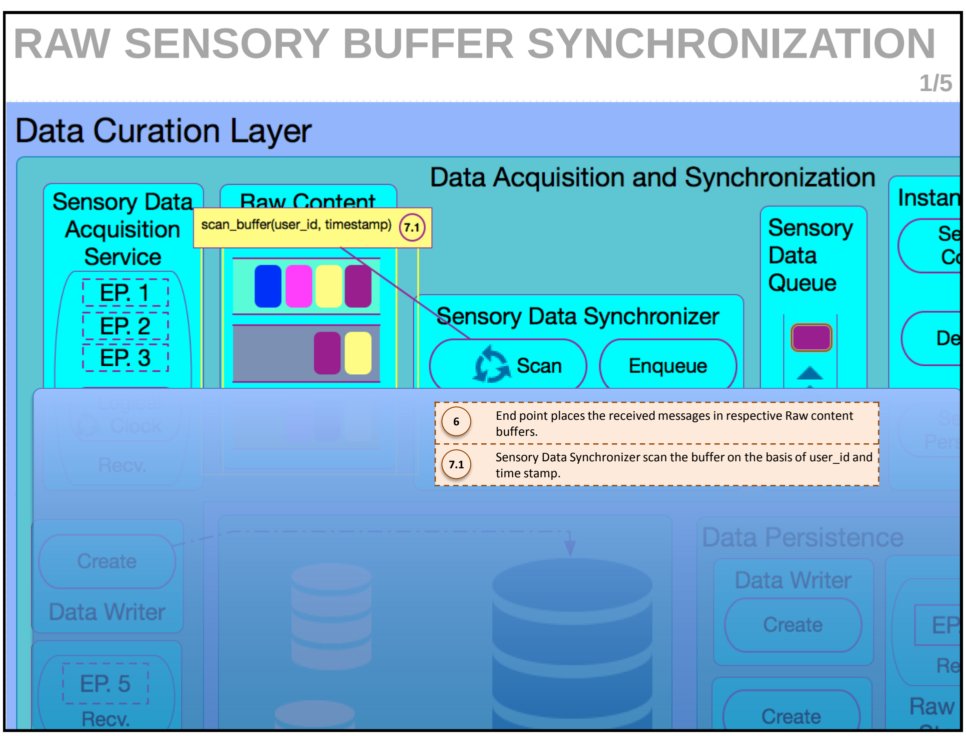


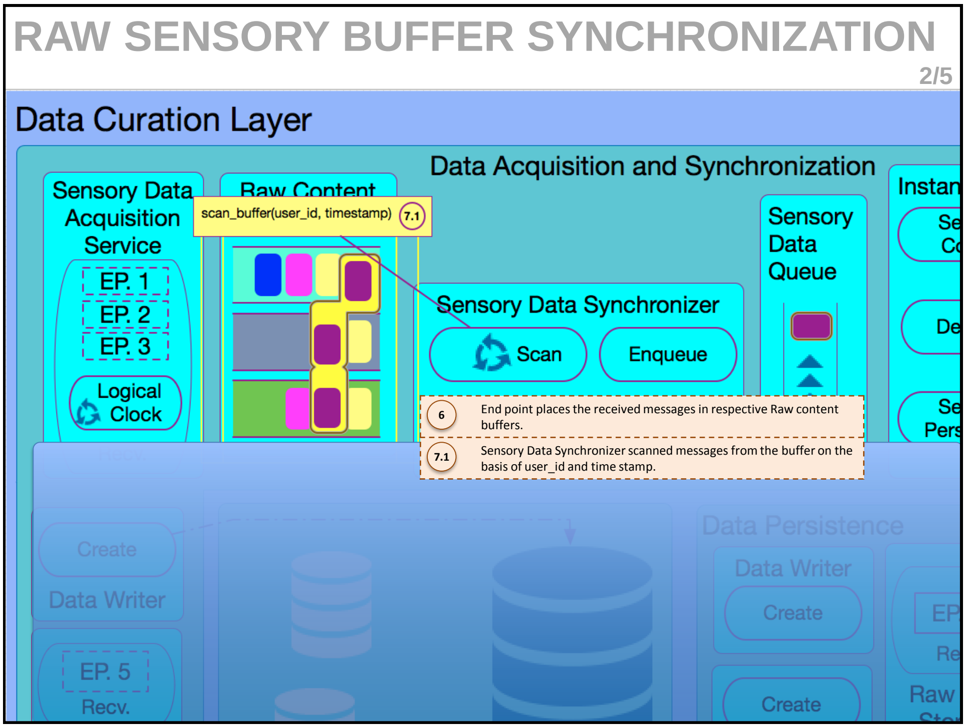
LOGICAL CLOCK SYNCHRONIZATION
4/4
Data Curation Layer
Logical Clock
4.1
old_val:010216|08:33:45
cur_val:010216|10:36:13
Sensory Data Acquisition Service
EP. 1
EP. 2
EP. 3
Logical Clock
Recv.
Raw Content Buffers
Create Instance
LC:010216|10:36:13
LC:010216|10:36:13
3
3
Create
Data Writer
EP. 5
Recv.
Logical Clock
4.1
old_val:010216|08:33:45
cur_val:010216|10:36:13
1 User Activity registered in Smart phone and Kinect Camera.
2 Real Time Synchronization of logical clocks
3 Share Synchronized time with smart phone and Kinect connected PC
4.1 Logical Clocks are updated on smart phone and Kinect Connected PC.


MESSAGE GENERATION
1/1
Data Curation Layer
Smartphone App
create_msg(
user_id: john_doe
lat:41.25, long:39.26
timestamp: 010216|10:36:13
act:012199923.....)
4.2
Smartphone App
create_msg(
user_id: john_doe
timestamp: 010216|10:36:13
audiostream:01001010010..)
4.2
Kinect Client
create_msg(
user_id: john_doe
timestamp: 010216|10:36:13
act:012199923.....)
4.2
Logical Clock
Sensory Data Acquisition Service
EP. 1
EP. 2
EP. 3
Logical Clock
Recv.
4.1 Logical Clocks are updated on smart phone and Kinect Connected PC.
4.2 Creation of communication message of payload
Raw Content Buffers
Scan
Create Instance
Create
Data Writer
EP. 5
Recv.

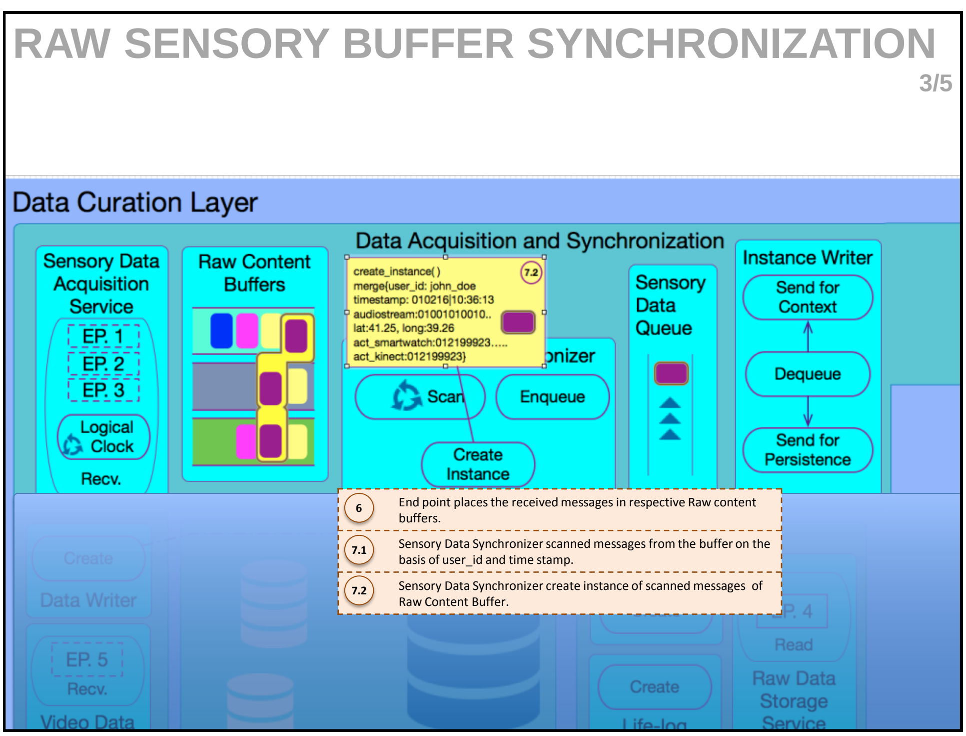

RAW SENSORY DATA ACQUSITION
1/2
Data Curation Layer
Smartphone App
create_msg(
user_id: john_doe
lat:41.25, long:39.26
timestamp: 010216|10:36:13
act:012199923.....)
Smartphone App
create_msg(
user_id: john_doe
timestamp: 010216|10:36:13
audiostream:01001010010..)
Kinect Client
create_msg(
user_id: john_doe
timestamp: 010216|10:36:13
act:012199923.....)
5
5
5
Logical Clock
Sensory Data Acquisition Service
EP. 1
EP. 2
EP. 3
Logical Clock
Recv.
4.1 Logical Clocks are updated on smart phone and Kinect Connected PC.
4.2 Creation of communication message of payload
5 Created message is communicated through end point of Sensory Data Acquisition Services.
Raw Content Buffers
Create Instance
Create
Data Writer
EP. 5
Recv.


RAW SENSORY DATA ACQUSITION
2/2
Data Curation
Smartphone App
create_msg(
user_id: john_doe
lat:41.25, long:39.26
timestamp: 010216|10:36:13
act:012199923…..)
Data Acquisition and Synchronization
Smartphone App
create_msg(
user_id: john_doe
timestamp: 010216|10:36:13
audiostream:01001010010..)
Sensory Data Acquisition Service
EP. 1
EP. 2
EP. 3
Logical Clock
Recv.
Sensory Data Queue
Instan
4.1 Logical Clocks are updated on smart phone and Kinect Connected PC.
4.2 Creation of communication message of payload
5 Created message is communicated through end point of Sensory Data Acquisition Services.
6 End point places the received messages in respective Raw content buffers.
Kinect Client
create_msg(
user_id: john_doe
timestamp: 010216|10:36:13
act:012199923…..)
Create
Data Writer
EP. 5
Recv.
Data Persistence
Data Writer
Create
Create
Raw

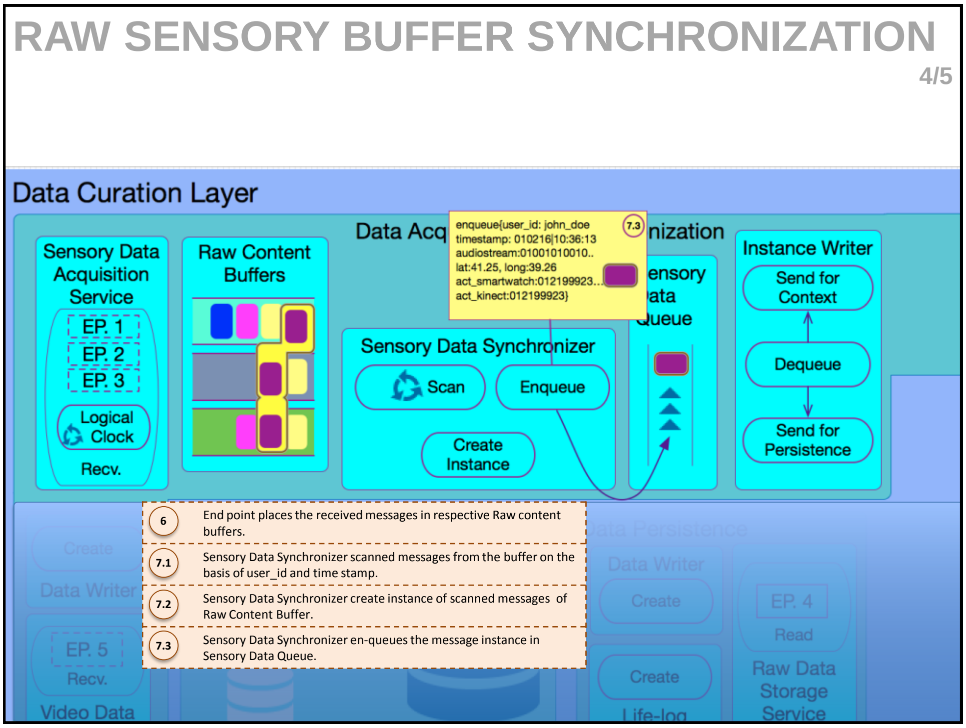

RAW SENSORY BUFFER SYNCHRONIZATION
1/5
Data Curation Layer
Data Acquisition and Synchronization
Sensory Data Acquisition Service
Raw Content
scan_buffer(user_id, timestamp) 7.1
EP. 1
EP. 2
EP. 3
Sensory Data Synchronizer
Scan
Enqueue
Sensory Data Queue
Instan
6 End point places the received messages in respective Raw content buffers.
7.1 Sensory Data Synchronizer scan the buffer on the basis of user_id and time stamp.
Create
Data Writer
EP. 5
Recv.
Data Persistence
Data Writer
Create
Create
Raw

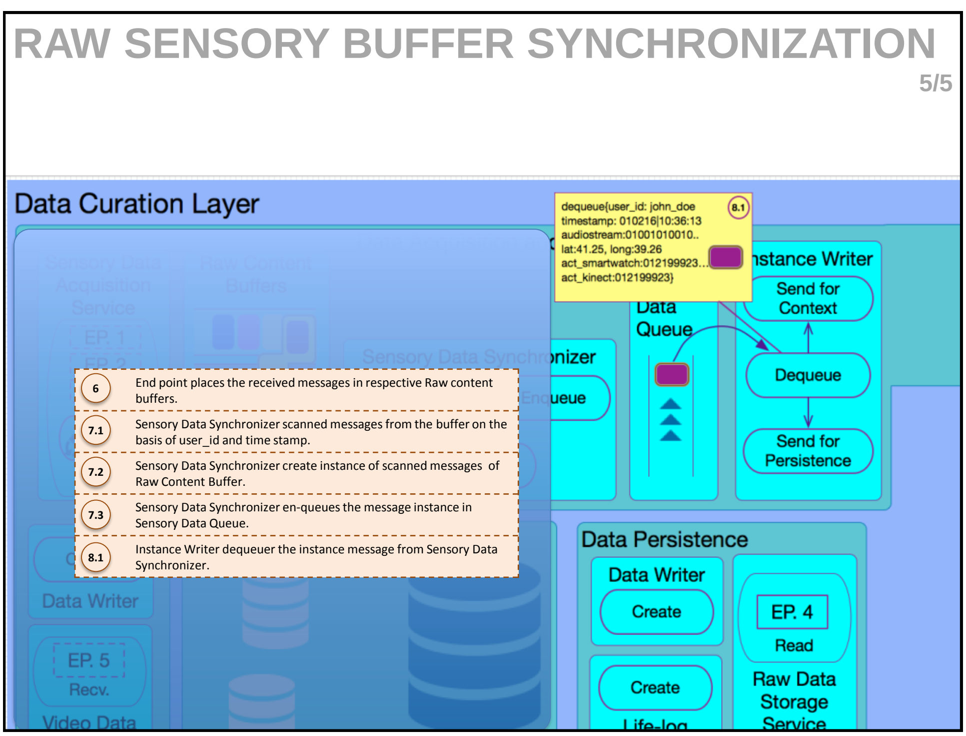


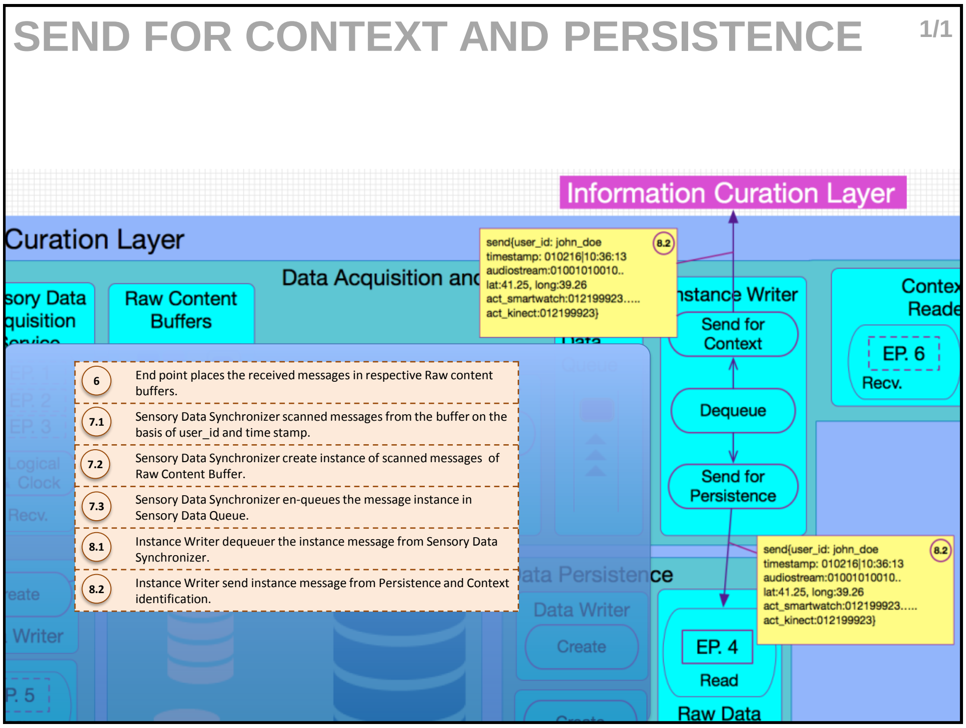
RAW SENSORY BUFFER SYNCHRONIZATION
2/5
Data Curation Layer
Data Acquisition and Synchronization
Sensory Data Acquisition Service
EP. 1
EP. 2
EP. 3
Logical Clock
Raw Content
scan_buffer(user_id, timestamp)
7.1
Sensory Data Queue
Sensory Data Synchronizer
Scan
Enqueue
6
End point places the received messages in respective Raw content buffers.
7.1
Sensory Data Synchronizer scanned messages from the buffer on the basis of user_id and time stamp.
Data Persistence
Data Writer
Create
Create
Data Writer
EP. 5
Recv.
Create
Raw

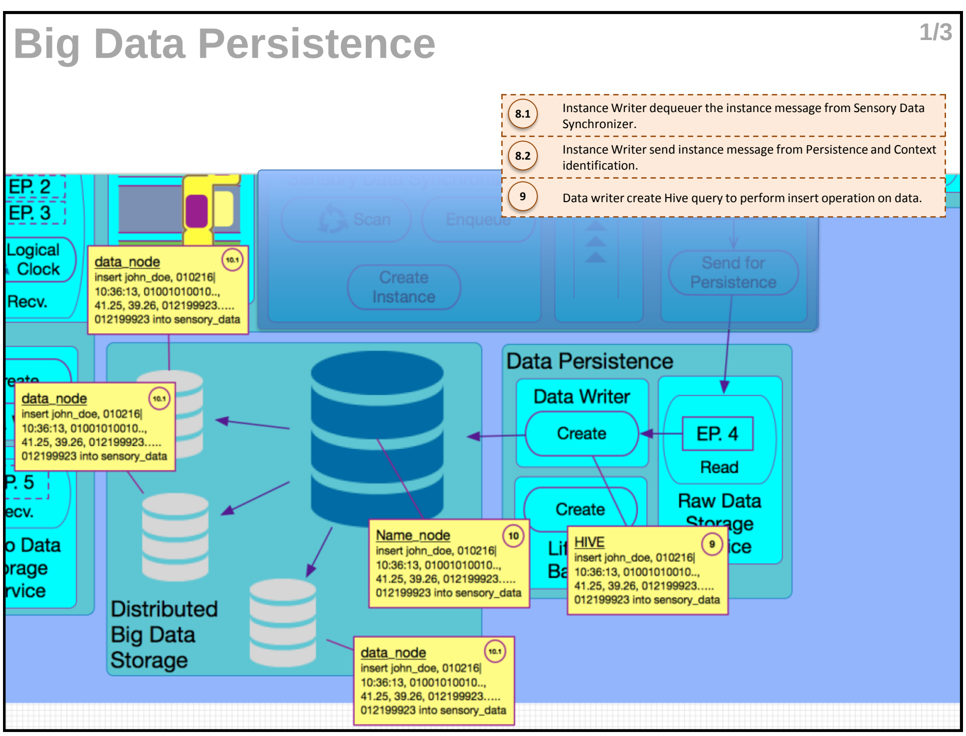


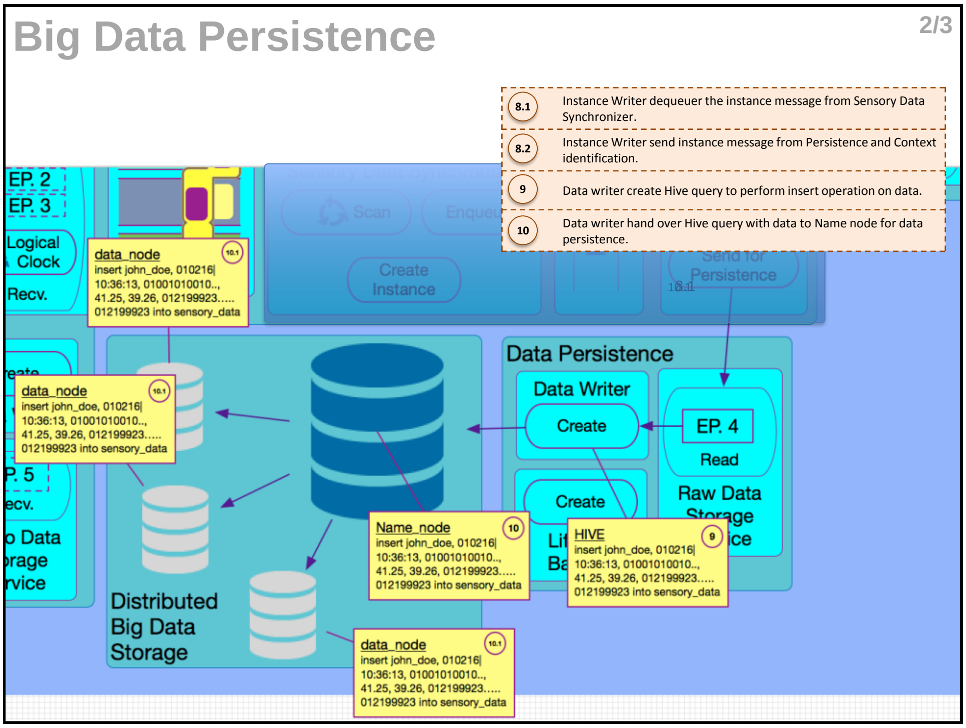
RAW SENSORY BUFFER SYNCHRONIZATION
3/5
Data Curation Layer
Data Acquisition and Synchronization
Sensory Data Acquisition Service
EP. 1
EP. 2
EP. 3
Logical Clock
Recv.
Raw Content Buffers
create_instance( )
merge{user_id: john_doe
timestamp: 010216|10:36:13
audiostream:01001010010..
lat:41.25, long:39.26
act_smartwatch:012199923.....
act_kinect:012199923}
7.2
Scan
Enqueue
Create Instance
Sensory Data Queue
Instance Writer
Send for Context
Dequeue
Send for Persistence
6
End point places the received messages in respective Raw content buffers.
7.1
Sensory Data Synchronizer scanned messages from the buffer on the basis of user_id and time stamp.
7.2
Sensory Data Synchronizer create instance of scanned messages of Raw Content Buffer.
Create
Data Writer
EP. 5
Recv.
Video Data
Create
Read
Raw Data Storage Service

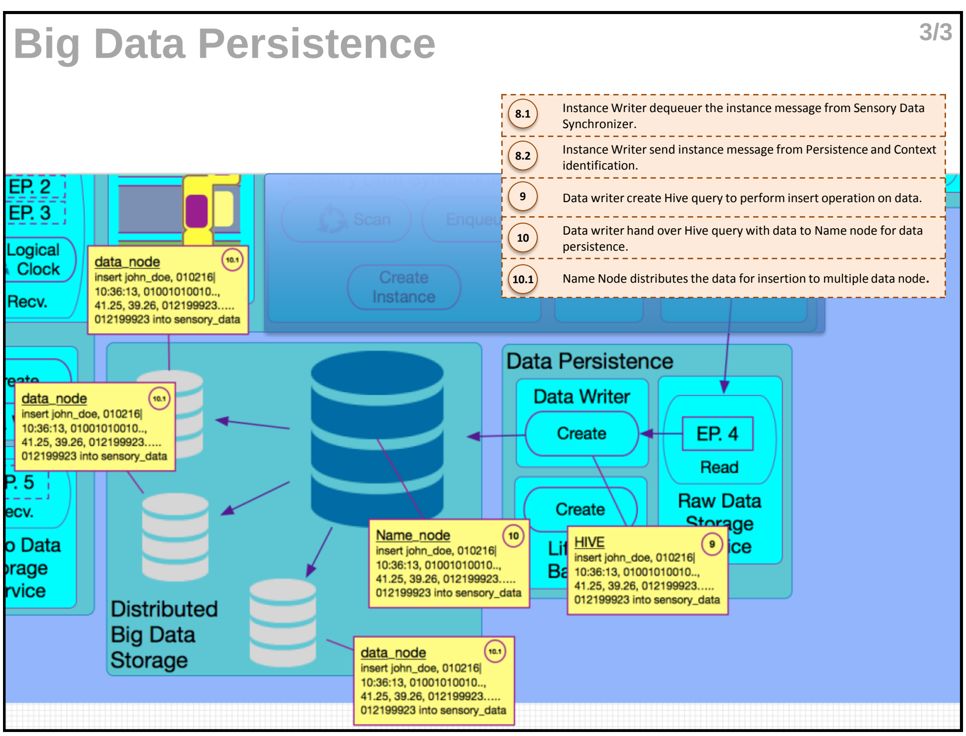

RAW SENSORY BUFFER SYNCHRONIZATION
4/5
Data Curation Layer
Data Acq
nization
enqueue{user_id: john_doe
timestamp: 010216|10:36:13
audiostream:01001010010..
lat:41.25, long:39.26
act_smartwatch:012199923…
act_kinect:012199923}
7.3
Sensory Data Acquisition Service
EP. 1
EP. 2
EP. 3
Logical Clock
Recv.
Raw Content Buffers
Sensory Data Synchronizer
Scan
Enqueue
Create Instance
ensory
ata
Queue
Instance Writer
Send for Context
Dequeue
Send for Persistence
6
End point places the received messages in respective Raw content buffers.
7.1
Sensory Data Synchronizer scanned messages from the buffer on the basis of user_id and time stamp.
7.2
Sensory Data Synchronizer create instance of scanned messages of Raw Content Buffer.
7.3
Sensory Data Synchronizer en-queues the message instance in Sensory Data Queue.

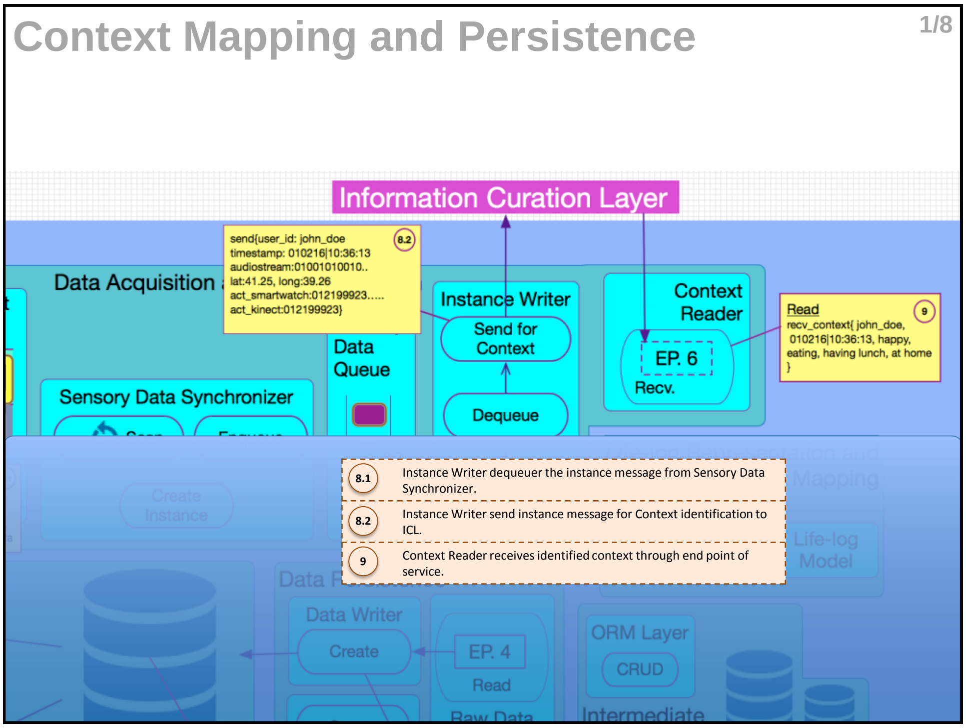

RAW SENSORY BUFFER SYNCHRONIZATION
5/5
Data Curation Layer
dequeue{user_id: john_doe
timestamp: 010216|10:36:13
audiostream:01001010010..
lat:41.25, long:39.26
act_smartwatch:012199923…
act_kinect:012199923}
8.1
nstance Writer
Send for Context
Dequeue
Send for Persistence
Data
Queue
6
End point places the received messages in respective Raw content buffers.
7.1
Sensory Data Synchronizer scanned messages from the buffer on the basis of user_id and time stamp.
7.2
Sensory Data Synchronizer create instance of scanned messages of Raw Content Buffer.
7.3
Sensory Data Synchronizer en-queues the message instance in Sensory Data Queue.
8.1
Instance Writer dequeuer the instance message from Sensory Data Synchronizer.
Data Persistence
Data Writer
Create
EP. 4
Read
Raw Data Storage Service
Create
Life-log
Data Writer
EP. 5
Recv.
Video Data

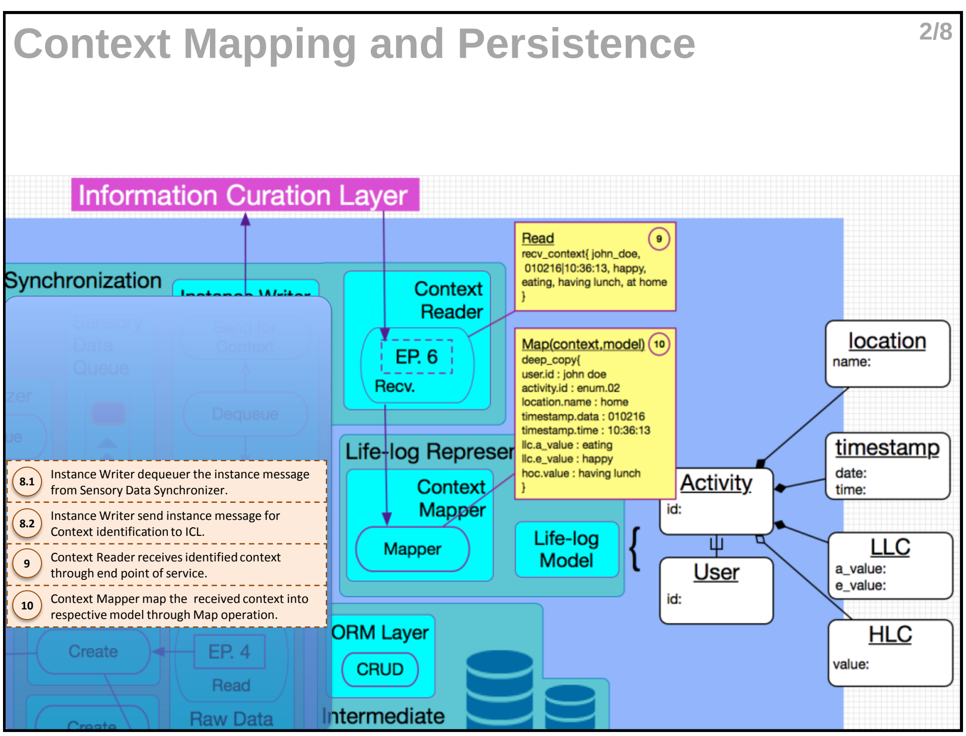

SEND FOR CONTEXT AND PERSISTENCE
1/1
Information Curation Layer
Curation Layer
Data Acquisition and
Raw Content Buffers
send{user_id: john_doe
timestamp: 010216|10:36:13
audiostream:01001010010..
lat:41.25, long:39.26
act_smartwatch:012199923.....
act_kinect:012199923}
8.2
Instance Writer
Send for Context
Dequeue
Send for Persistence
EP. 6
Recv.
6 End point places the received messages in respective Raw content buffers.
7.1 Sensory Data Synchronizer scanned messages from the buffer on the basis of user_id and time stamp.
7.2 Sensory Data Synchronizer create instance of scanned messages of Raw Content Buffer.
7.3 Sensory Data Synchronizer en-queues the message instance in Sensory Data Queue.
8.1 Instance Writer dequeuer the instance message from Sensory Data Synchronizer.
8.2 Instance Writer send instance message from Persistence and Context identification.
Data Persistence
Data Writer
Create
EP. 4
Read
Raw Data
send{user_id: john_doe
timestamp: 010216|10:36:13
audiostream:01001010010..
lat:41.25, long:39.26
act_smartwatch:012199923.....
act_kinect:012199923}
8.2

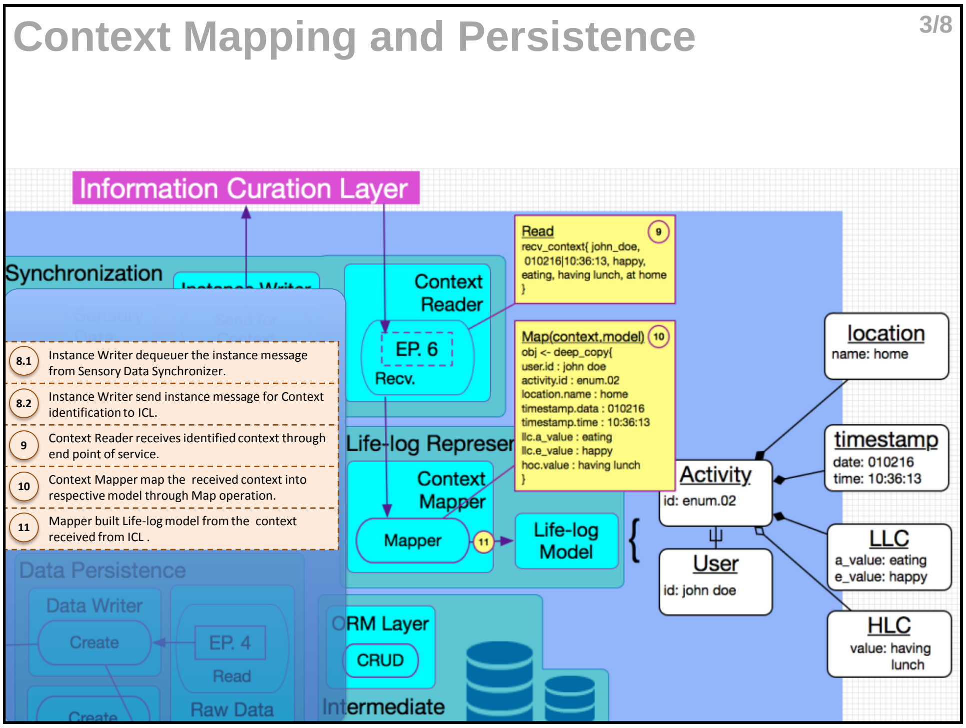

Big Data Persistence
1/3
8.1 Instance Writer dequeuer the instance message from Sensory Data Synchronizer.
8.2 Instance Writer send instance message from Persistence and Context identification.
9 Data writer create Hive query to perform insert operation on data.
EP. 2
EP. 3
Logical Clock
Recv.
data_node
insert john_doe, 010216|
10:36:13, 01001010010..,
41.25, 39.26, 012199923.....
012199923 into sensory_data
Create Instance
Send for Persistence
Data Persistence
Data Writer
Create
EP. 4
Read
Raw Data Storage
Create
Name_node
10
insert john_doe, 010216|
10:36:13, 01001010010..,
41.25, 39.26, 012199923.....
012199923 into sensory_data
HIVE
9
insert john_doe, 010216|
10:36:13, 01001010010..,
41.25, 39.26, 012199923.....
012199923 into sensory_data
Distributed Big Data Storage
data_node
insert john_doe, 010216|
10:36:13, 01001010010..,
41.25, 39.26, 012199923.....
012199923 into sensory_data


Big Data Persistence
2/3
8.1
Instance Writer dequeuer the instance message from Sensory Data Synchronizer.
8.2
Instance Writer send instance message from Persistence and Context identification.
9
Data writer create Hive query to perform insert operation on data.
10
Data writer hand over Hive query with data to Name node for data persistence.
EP. 2
EP. 3
Logical Clock
Recv.
Scan
Create Instance
Send for Persistence
data_node
insert john_doe, 010216|
10:36:13, 01001010010..,
41.25, 39.26, 012199923.....
012199923 into sensory_data
Data Persistence
Data Writer
Create
EP. 4
Read
Raw Data Storage
Name_node
HIVE
Distributed Big Data Storage

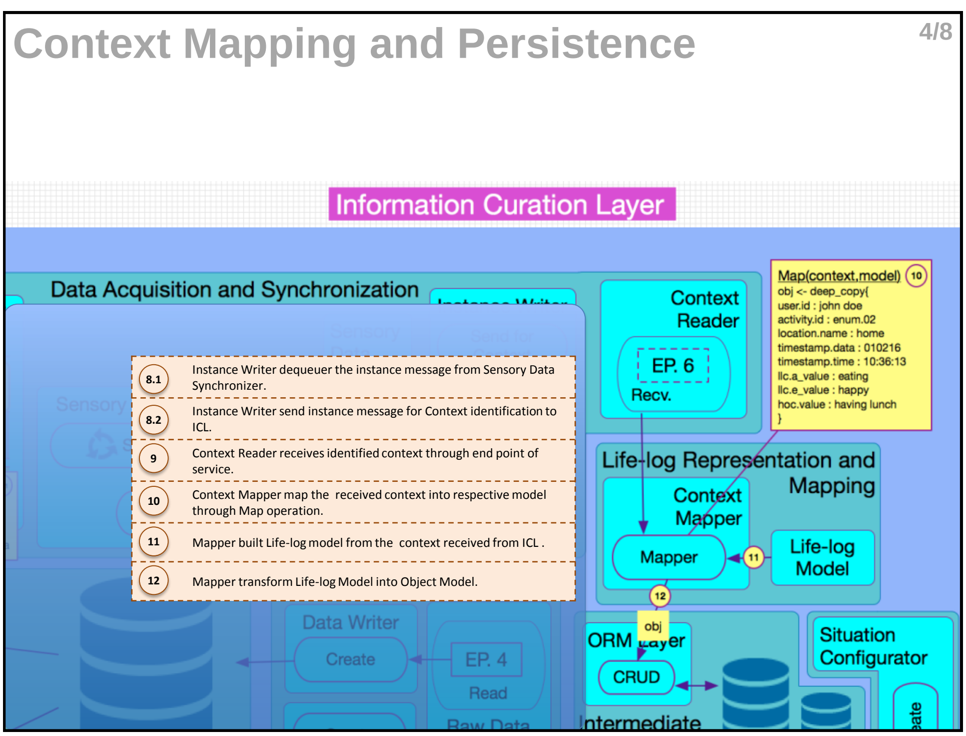

Big Data Persistence
3/3
8.1
Instance Writer dequeuer the instance message from Sensory Data Synchronizer.
8.2
Instance Writer send instance message from Persistence and Context identification.
9
Data writer create Hive query to perform insert operation on data.
10
Data writer hand over Hive query with data to Name node for data persistence.
10.1
Name Node distributes the data for insertion to multiple data node.
EP. 2
EP. 3
Logical Clock
Recv.
Scan
Create Instance
data_node
insert john_doe, 010216|
10:36:13, 01001010010..,
41.25, 39.26, 012199923.....
012199923 into sensory_data
Data Persistence
Data Writer
Create
EP. 4
Read
Raw Data Storage
Name_node
HIVE
Distributed Big Data Storage


Context Mapping and Persistence
1/8
Information Curation Layer
Data Acquisition
send{user_id: john_doe
timestamp: 010216|10:36:13
audiostream:01001010010..
lat:41.25, long:39.26
act_smartwatch:012199923.....
act_kinect:012199923}
8.2
Instance Writer
Send for Context
Dequeue
Data Queue
Sensory Data Synchronizer
Context Reader
EP. 6
Recv.
Read
recv_context{ john_doe,
010216|10:36:13, happy,
eating, having lunch, at home
}
9
8.1
Instance Writer dequeuer the instance message from Sensory Data Synchronizer.
8.2
Instance Writer send instance message for Context identification to ICL.
9
Context Reader receives identified context through end point of service.

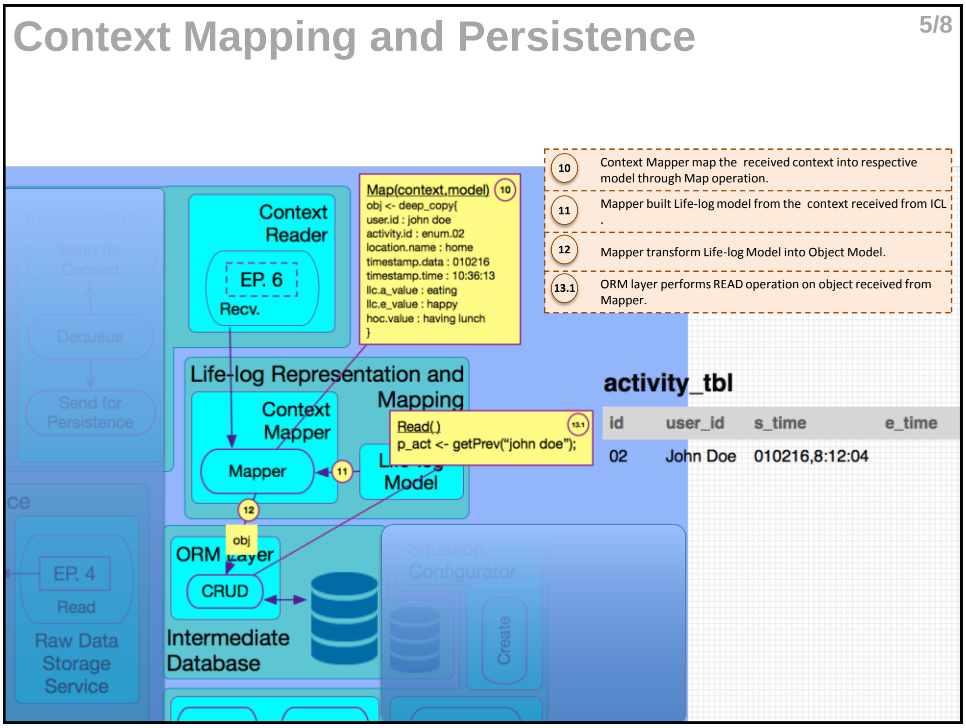

Context Mapping and Persistence
2/8
Information Curation Layer
Synchronization
Context Reader
EP. 6
Recv.
Read
recv_context{ john_doe,
010216|10:36:13, happy,
eating, having lunch, at home
}
9
Map(context,model)
10
deep_copy{
user.id : john doe
activity.id : enum.02
location.name : home
timestamp.data : 010216
timestamp.time : 10:36:13
llc.a_value : eating
llc.e_value : happy
hoc.value : having lunch
}
Life-log Represen
Context Mapper
Mapper
Life-log Model
ORM Layer
CRUD
Intermediate
Create
EP. 4
Read
Raw Data
Activity
id:
User
id:
location
name:
timestamp
date:
time:
LLC
a_value:
e_value:
HLC
value:
8.1
Instance Writer dequeuer the instance message from Sensory Data Synchronizer.
8.2
Instance Writer send instance message for Context identification to ICL.
9
Context Reader receives identified context through end point of service.
10
Context Mapper map the received context into respective model through Map operation.

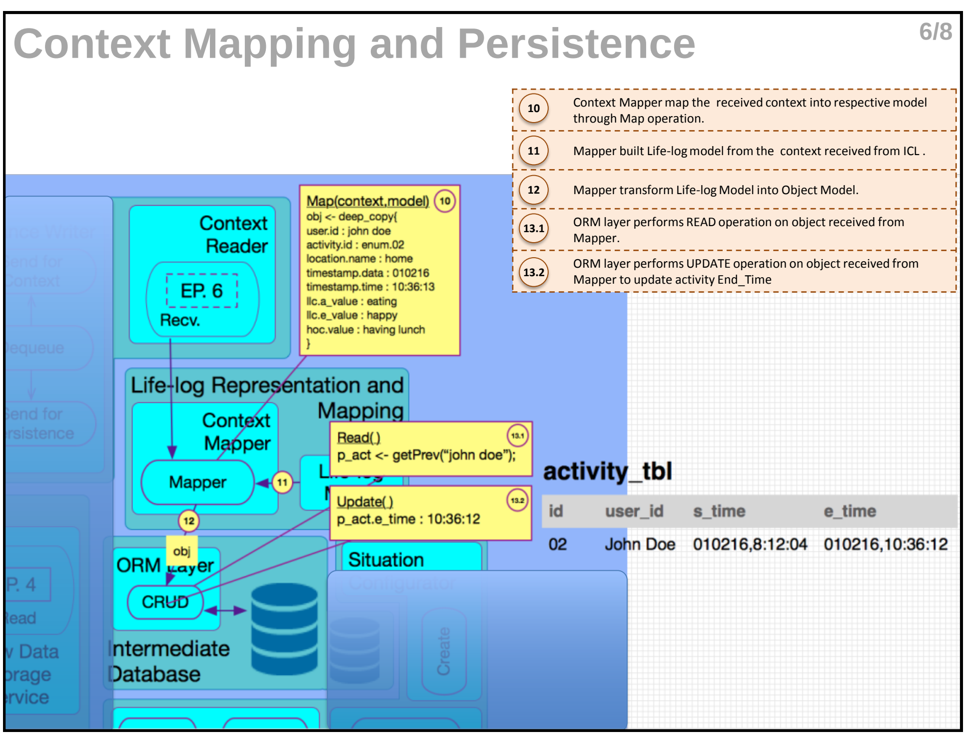


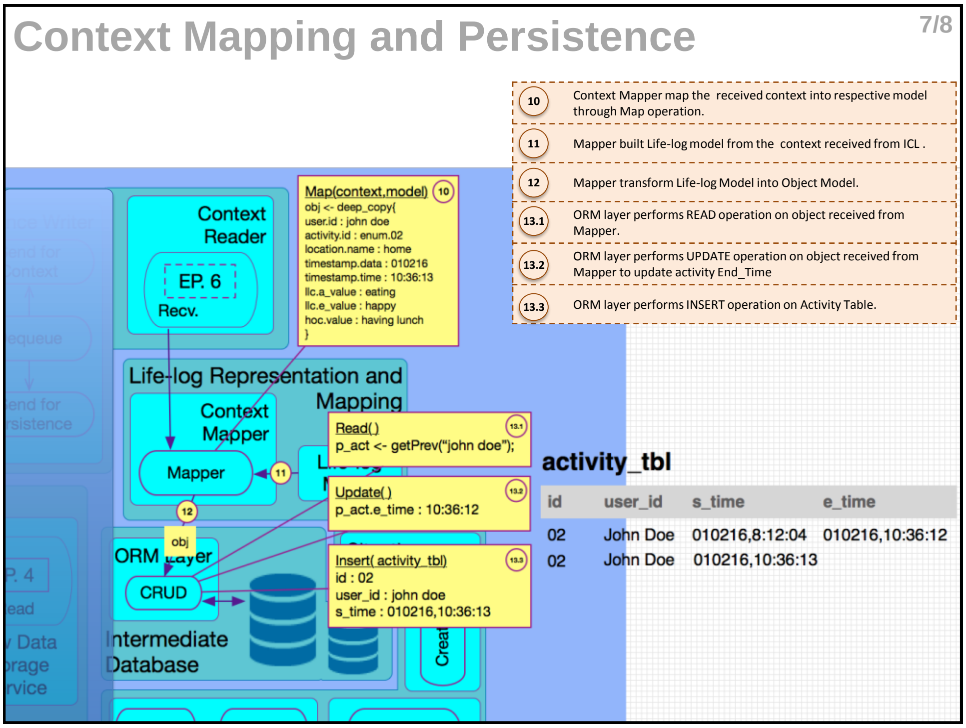
Context Mapping and Persistence
3/8
Information Curation Layer
Synchronization
Context Reader
EP. 6
Recv.
Read
recv_context{ john_doe,
010216|10:36:13, happy,
eating, having lunch, at home
}
9
Map(context,model)
10
obj <- deep_copy{
user.id : john doe
activity.id : enum.02
location.name : home
timestamp.data : 010216
timestamp.time : 10:36:13
llc.a_value : eating
llc.e_value : happy
hoc.value : having lunch
}
8.1
Instance Writer dequeuer the instance message from Sensory Data Synchronizer.
8.2
Instance Writer send instance message for Context identification to ICL.
9
Context Reader receives identified context through end point of service.
10
Context Mapper map the received context into respective model through Map operation.
11
Mapper built Life-log model from the context received from ICL .
Life-log Represer
Context Mapper
Mapper
11
Life-log Model
location
name: home
timestamp
date: 010216
time: 10:36:13
Activity
id: enum.02
User
id: john doe
LLC
a_value: eating
e_value: happy
HLC
value: having lunch
Data Persistence
Data Writer
Create
EP. 4
Read
Raw Data
ORM Layer
CRUD
Intermediate

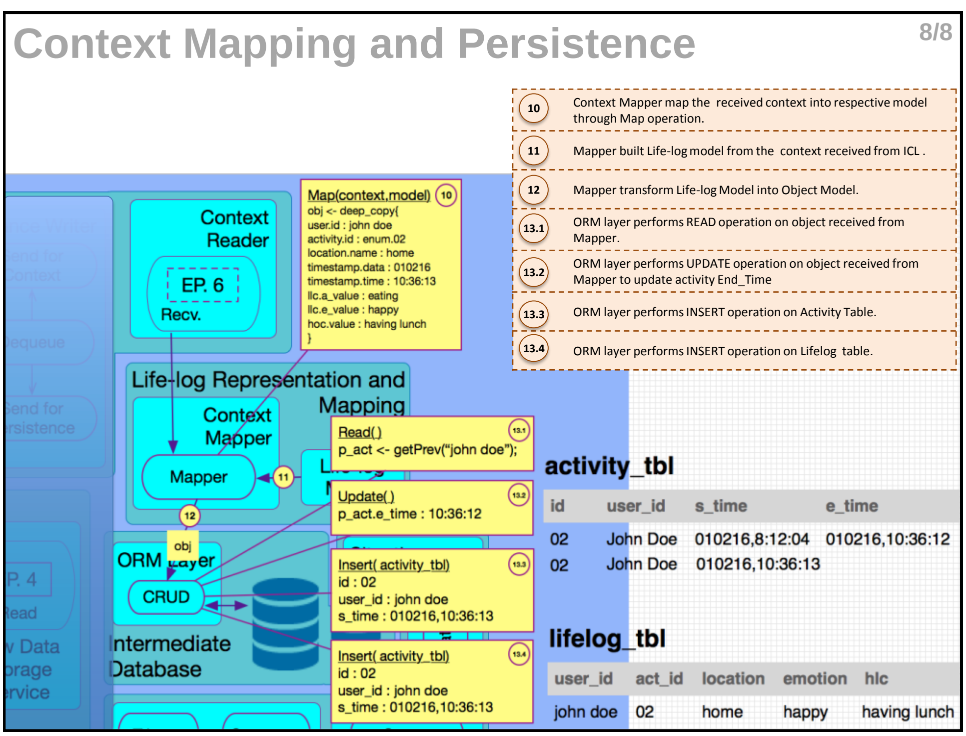


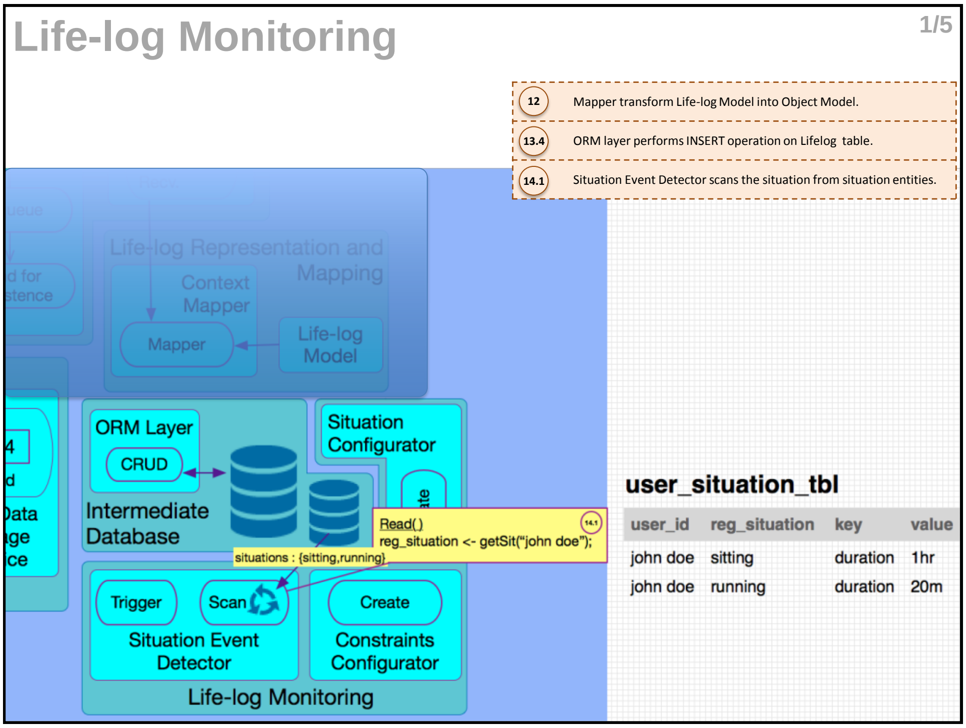
Context Mapping and Persistence
4/8
Information Curation Layer
Data Acquisition and Synchronization
Context Reader
EP. 6
Recv.
Map(context,model)
10
obj <- deep_copy{
user.id : john doe
activity.id : enum.02
location.name : home
timestamp.data : 010216
timestamp.time : 10:36:13
llc.a_value : eating
llc.e_value : happy
hoc.value : having lunch
}
8.1
Instance Writer dequeuer the instance message from Sensory Data Synchronizer.
8.2
Instance Writer send instance message for Context identification to ICL.
9
Context Reader receives identified context through end point of service.
10
Context Mapper map the received context into respective model through Map operation.
11
Mapper built Life-log model from the context received from ICL .
12
Mapper transform Life-log Model into Object Model.
Life-log Representation and Mapping
Context Mapper
Mapper
11
Life-log Model
12
obj
ORM Layer
CRUD
Intermediate
Situation Configurator
Data Writer
Create
EP. 4
Read
Raw Data

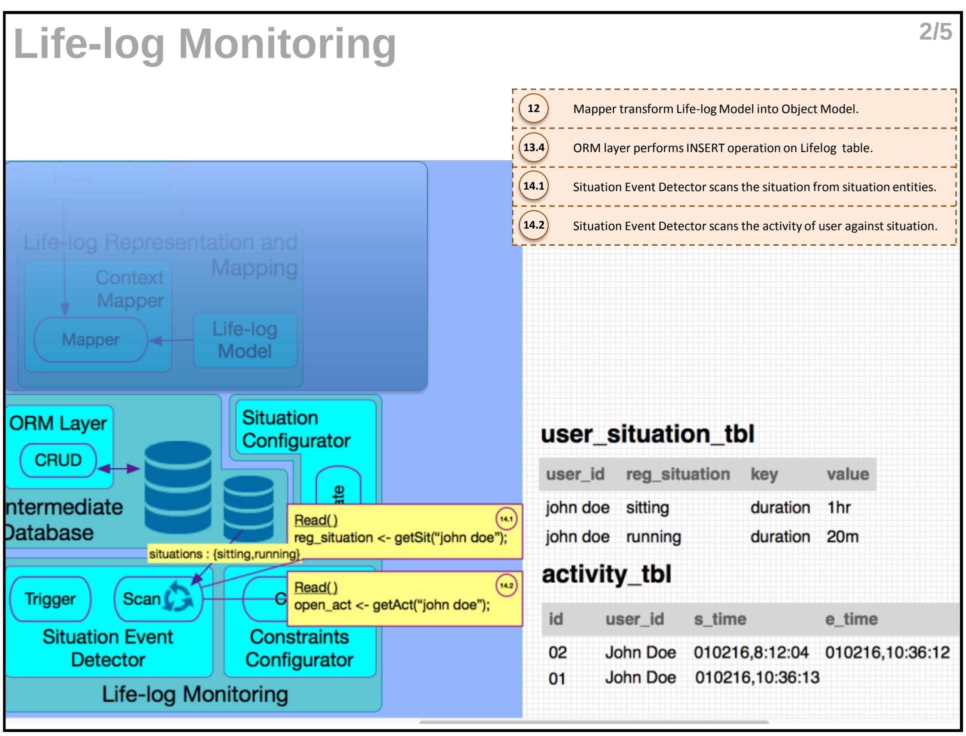

Context Mapping and Persistence
5/8
10 Context Mapper map the received context into respective model through Map operation.
11 Mapper built Life-log model from the context received from ICL .
12 Mapper transform Life-log Model into Object Model.
13.1 ORM layer performs READ operation on object received from Mapper.
Context Reader
EP. 6
Recv.
Map(context,model) 10
obj <- deep_copy(
user.id : john doe
activity.id : enum.02
location.name : home
timestamp.data : 010216
timestamp.time : 10:36:13
llc.a_value : eating
llc.e_value : happy
hoc.value : having lunch
}
Life-log Representation and Mapping
Context Mapper
Mapper
Model
Read()
p_act <- getPrev("john doe");
ORM Layer
obj
CRUD
Intermediate Database
EP. 4
Read
Raw Data Storage Service
Send for Context
Dequeue
Send for Persistence
Situation Configurator
Create
activity_tbl
id user_id s_time e_time
02 John Doe 010216,8:12:04


Context Mapping and Persistence
6/8
10 Context Mapper map the received context into respective model through Map operation.
11 Mapper built Life-log model from the context received from ICL .
12 Mapper transform Life-log Model into Object Model.
13.1 ORM layer performs READ operation on object received from Mapper.
13.2 ORM layer performs UPDATE operation on object received from Mapper to update activity End_Time
Context Reader
EP. 6
Recv.
Map(context,model) 10
obj <- deep_copy(
user.id : john doe
activity.id : enum.02
location.name : home
timestamp.data : 010216
timestamp.time : 10:36:13
llc.a_value : eating
llc.e_value : happy
hoc.value : having lunch
}
Life-log Representation and Mapping
Context Mapper
Mapper
Read()
p_act <- getPrev("john doe");
Update()
p_act.e_time : 10:36:12
ORM Layer
obj
CRUD
Situation
Intermediate Database
Create
activity_tbl
id user_id s_time e_time
02 John Doe 010216,8:12:04 010216,10:36:12

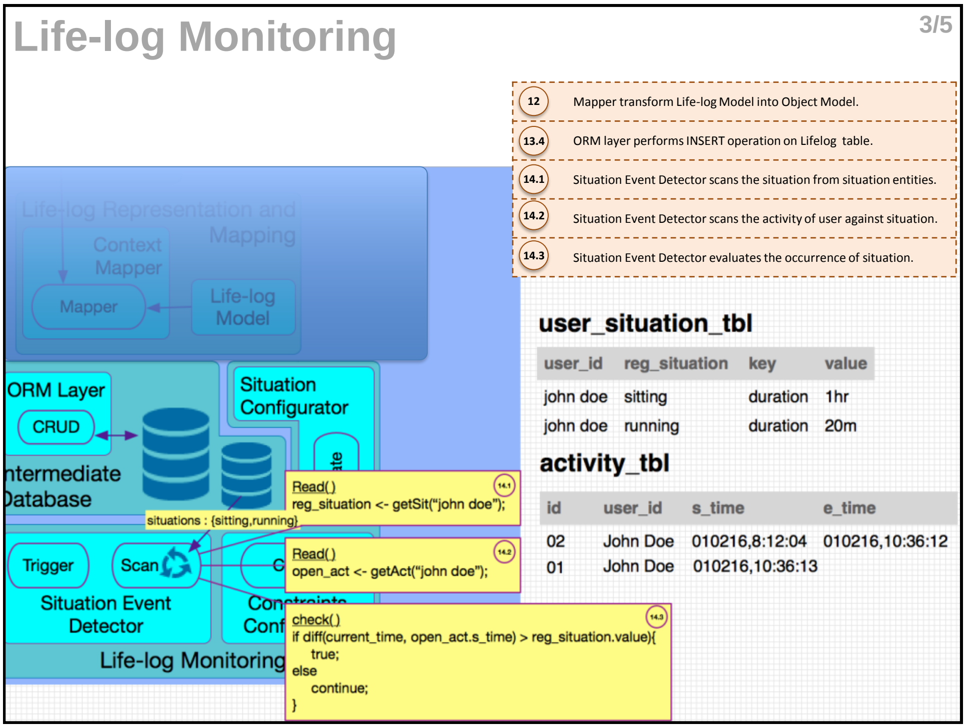

Context Mapping and Persistence
7/8
10 Context Mapper map the received context into respective model through Map operation.
11 Mapper built Life-log model from the context received from ICL .
12 Mapper transform Life-log Model into Object Model.
13.1 ORM layer performs READ operation on object received from Mapper.
13.2 ORM layer performs UPDATE operation on object received from Mapper to update activity End_Time
13.3 ORM layer performs INSERT operation on Activity Table.
Map(context,model) 10
obj <- deep_copy{
user.id : john doe
activity.id : enum.02
location.name : home
timestamp.data : 010216
timestamp.time : 10:36:13
llc.a_value : eating
llc.e_value : happy
hoc.value : having lunch
}
Context Reader
EP. 6
Recv.
Life-log Representation and Mapping
Context Mapper
Mapper
11
12
obj
ORM Layer
CRUD
Intermediate Database
Read( )
p_act <- getPrev("john doe");
Update( )
p_act.e_time : 10:36:12
Insert(activity_tbl)
id : 02
user_id : john doe
s_time : 010216,10:36:13
activity_tbl
id | user_id | s_time | e_time
02 | John Doe | 010216,8:12:04 | 010216,10:36:12
02 | John Doe | 010216,10:36:13

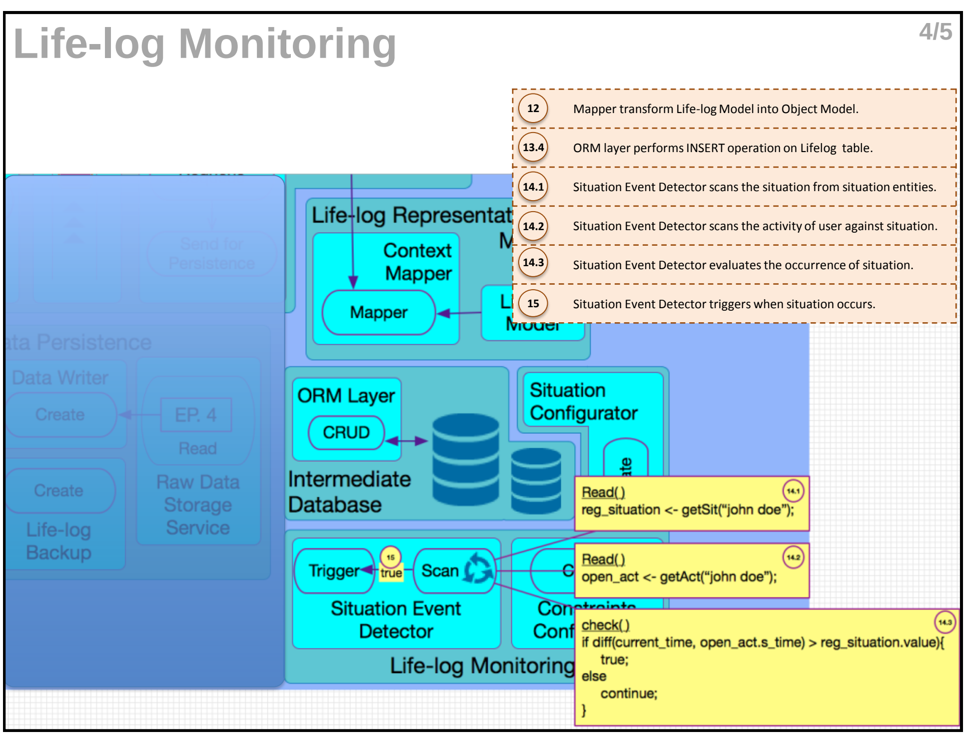

Context Mapping and Persistence
8/8
10 Context Mapper map the received context into respective model through Map operation.
11 Mapper built Life-log model from the context received from ICL .
12 Mapper transform Life-log Model into Object Model.
13.1 ORM layer performs READ operation on object received from Mapper.
13.2 ORM layer performs UPDATE operation on object received from Mapper to update activity End_Time
13.3 ORM layer performs INSERT operation on Activity Table.
13.4 ORM layer performs INSERT operation on Lifelog table.
Map(context,model) 10
obj <- deep_copy{
user.id : john doe
activity.id : enum.02
location.name : home
timestamp.data : 010216
timestamp.time : 10:36:13
llc.a_value : eating
llc.e_value : happy
hoc.value : having lunch
}
Context Reader
EP. 6
Recv.
Life-log Representation and Mapping
Context Mapper
Mapper
11
12
obj
ORM Layer
CRUD
Intermediate Database
Read( )
p_act <- getPrev("john doe");
Update( )
p_act.e_time : 10:36:12
Insert(activity_tbl)
id : 02
user_id : john doe
s_time : 010216,10:36:13
Insert(activity_tbl)
id : 02
user_id : john doe
s_time : 010216,10:36:13
activity_tbl
id | user_id | s_time | e_time
02 | John Doe | 010216,8:12:04 | 010216,10:36:12
02 | John Doe | 010216,10:36:13
lifelog_tbl
user_id | act_id | location | emotion | hlc
john doe | 02 | home | happy | having lunch


Life-log Monitoring
1/5
12 Mapper transform Life-log Model into Object Model.
13.4 ORM layer performs INSERT operation on Lifelog table.
14.1 Situation Event Detector scans the situation from situation entities.
Life-log Representation and Mapping
Context Mapper
Mapper
Life-log Model
ORM Layer
CRUD
Intermediate Database
Situation Configurator
Read()
reg_situation <- getSit("john doe");
situations : {sitting,running}
Trigger
Scan
Situation Event Detector
Create
Constraints Configurator
Life-log Monitoring
user_situation_tbl
user_id	reg_situation	key	value
john doe	sitting	duration	1hr
john doe	running	duration	20m


Life-log Monitoring
2/5
12 Mapper transform Life-log Model into Object Model.
13.4 ORM layer performs INSERT operation on Lifelog table.
14.1 Situation Event Detector scans the situation from situation entities.
14.2 Situation Event Detector scans the activity of user against situation.
Life-log Representation and Mapping
Context Mapper
Mapper
Life-log Model
ORM Layer
CRUD
ntermediate Database
Situation Configurator
Read()
reg_situation <- getSit("john doe");
situations : {sitting,running}
Read()
open_act <- getAct("john doe");
Trigger
Scan
Situation Event Detector
Constraints Configurator
Life-log Monitoring
user_situation_tbl
user_id	reg_situation	key	value
john doe	sitting	duration	1hr
john doe	running	duration	20m

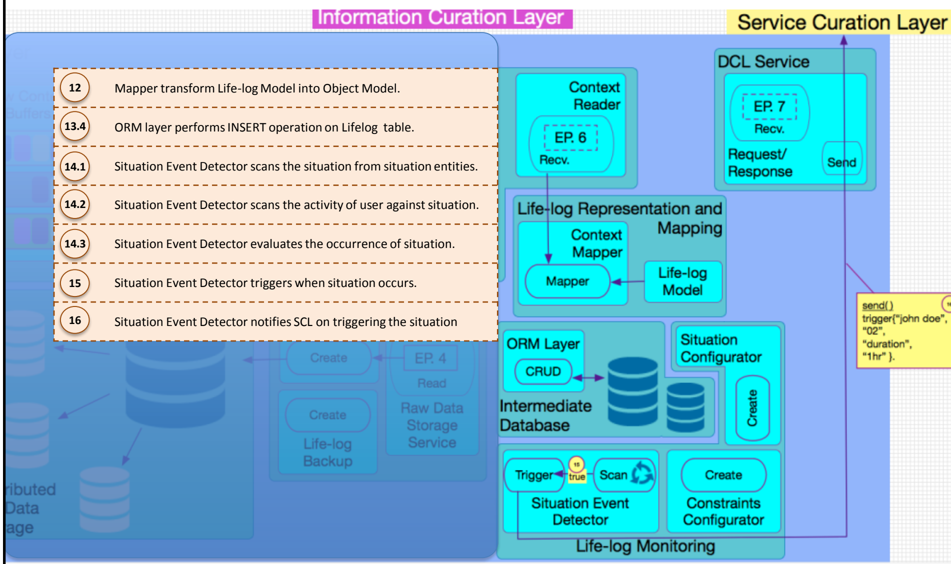
activity_tbl
id	user_id	s_time	e_time
02	John Doe	010216,8:12:04	010216,10:36:12
01	John Doe	010216,10:36:13	


Life-log Monitoring
3/5
12 Mapper transform Life-log Model into Object Model.
13.4 ORM layer performs INSERT operation on Lifelog table.
14.1 Situation Event Detector scans the situation from situation entities.
14.2 Situation Event Detector scans the activity of user against situation.
14.3 Situation Event Detector evaluates the occurrence of situation.
Life-log Representation and Mapping
Context Mapper
Mapper
Life-log Model
ORM Layer
CRUD
ntermediate Database
Situation Configurator
situations : {sitting,running}
Trigger
Scan
Situation Event Detector
Life-log Monitoring
Read()
reg_situation <- getSit("john doe");
Read()
open_act <- getAct("john doe");
check()
if diff(current_time, open_act.s_time) > reg_situation.value){
true;
else
continue;
}
user_situation_tbl
user_id reg_situation key value
john doe sitting duration 1hr
john doe running duration 20m
activity_tbl
id user_id s_time e_time
02 John Doe 010216,8:12:04 010216,10:36:12
01 John Doe 010216,10:36:13


Life-log Monitoring
4/5
12 Mapper transform Life-log Model into Object Model.
13.4 ORM layer performs INSERT operation on Lifelog table.
14.1 Situation Event Detector scans the situation from situation entities.
14.2 Situation Event Detector scans the activity of user against situation.
14.3 Situation Event Detector evaluates the occurrence of situation.
15 Situation Event Detector triggers when situation occurs.
Send for Persistence
ata Persistence
Data Writer
Create
EP. 4
Read
Create
Raw Data Storage Service
Life-log Backup
Life-log Representat
Context Mapper
Mapper
ORM Layer
CRUD
Intermediate Database
Situation Configurator
Trigger
true
Scan
Situation Event Detector
Life-log Monitoring
Read()
reg_situation <- getSit("john doe");
Read()
open_act <- getAct("john doe");
check()
if diff(current_time, open_act.s_time) > reg_situation.value){
true;
else
continue;
}

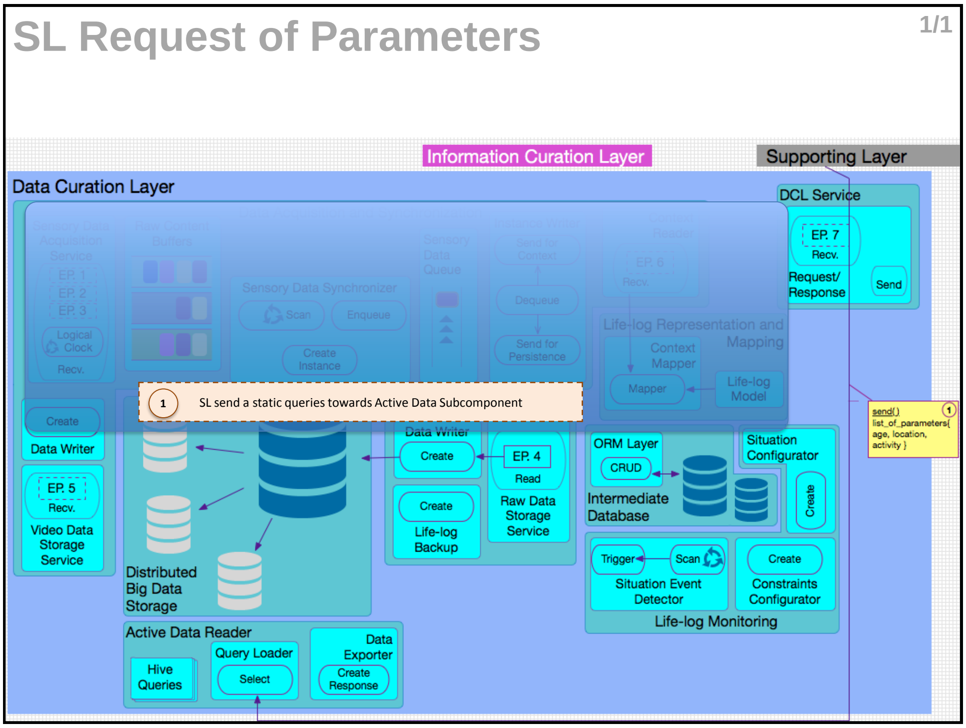
Life-log Monitoring

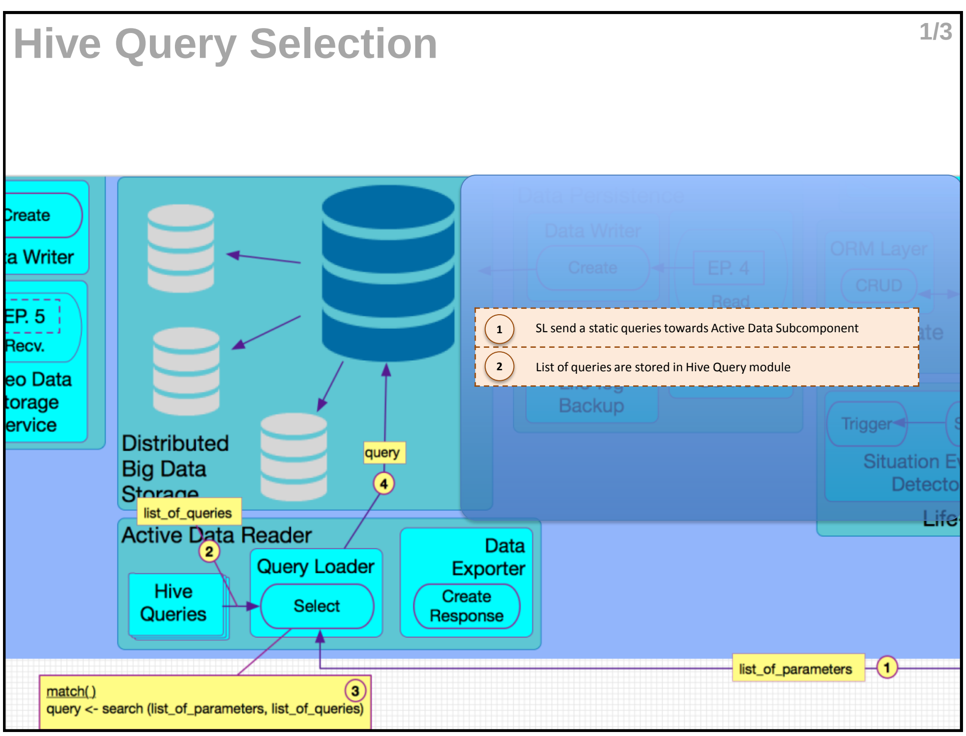
5/5

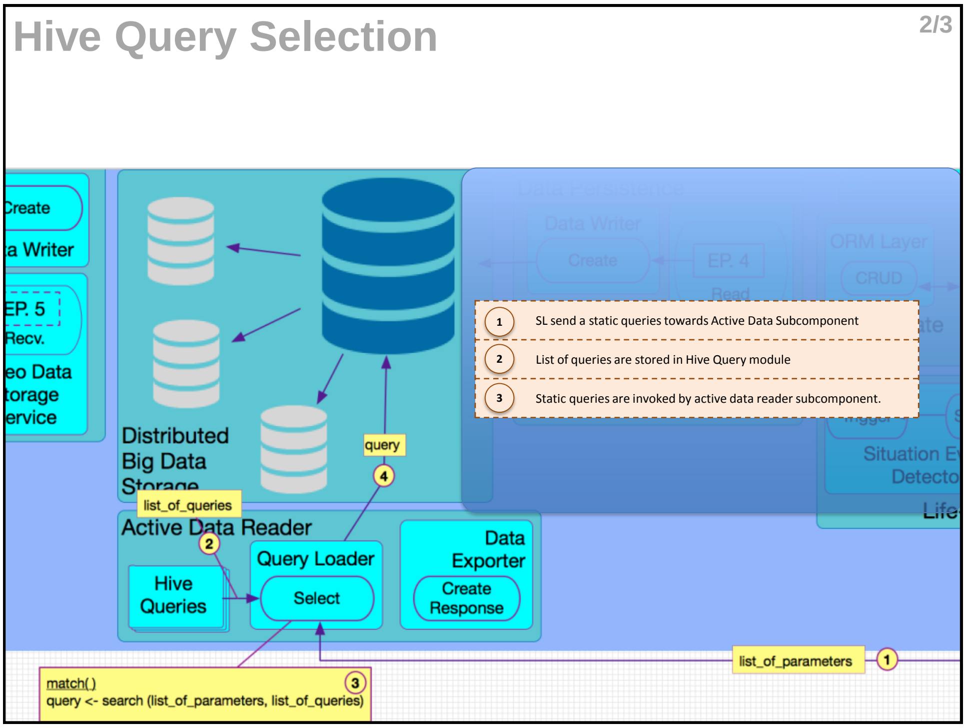


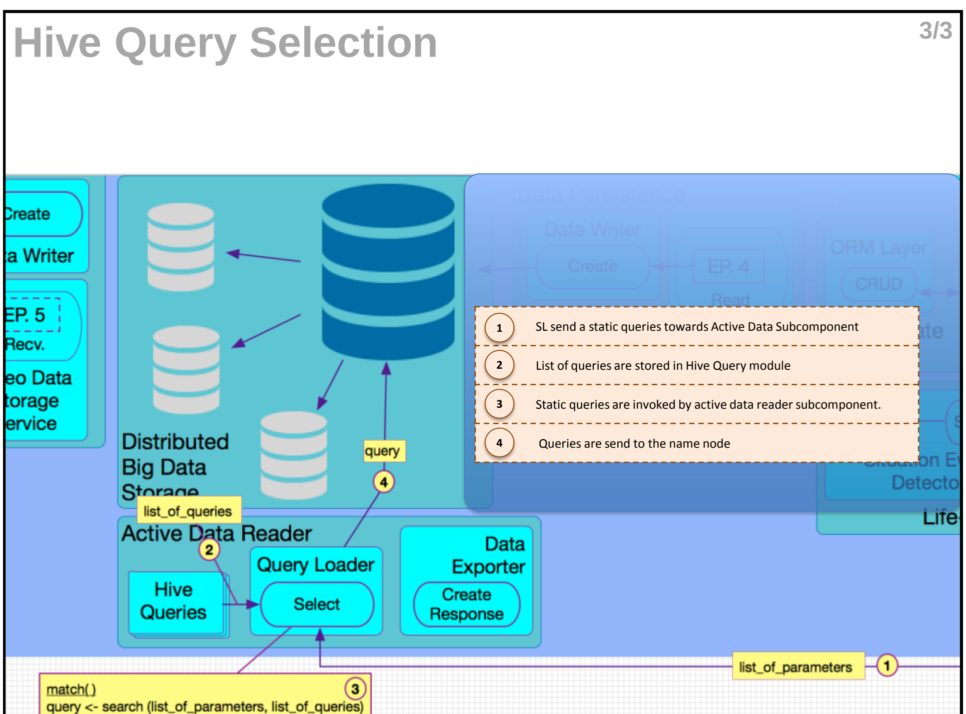
Scenario 2: Real-time

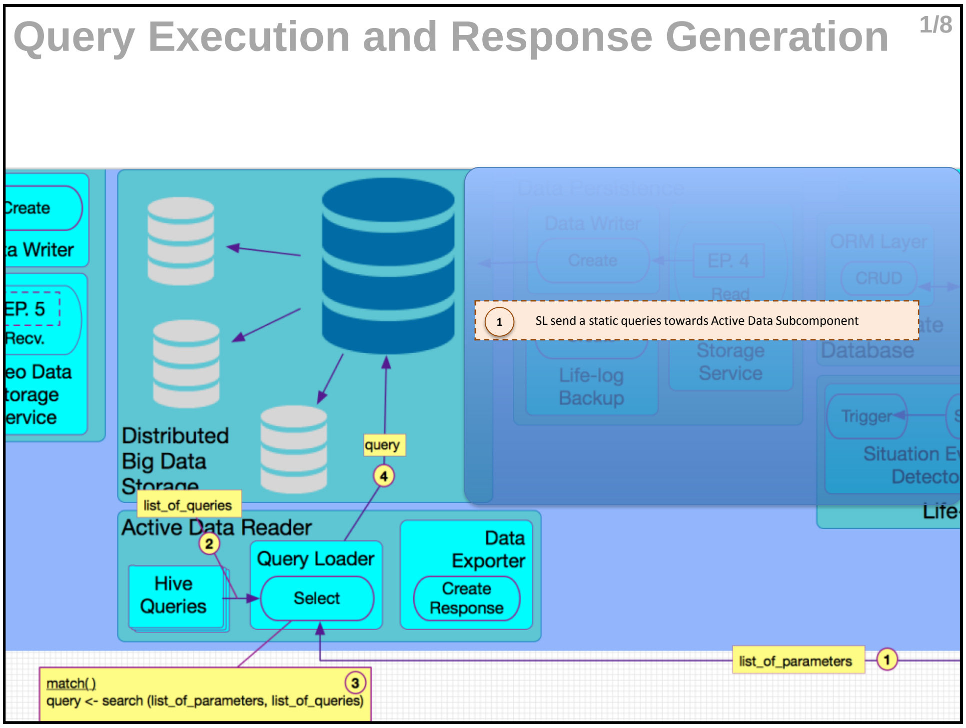
Desc. : Raw-sensory data from big data storage is requested for **visualization** and **analytics**.

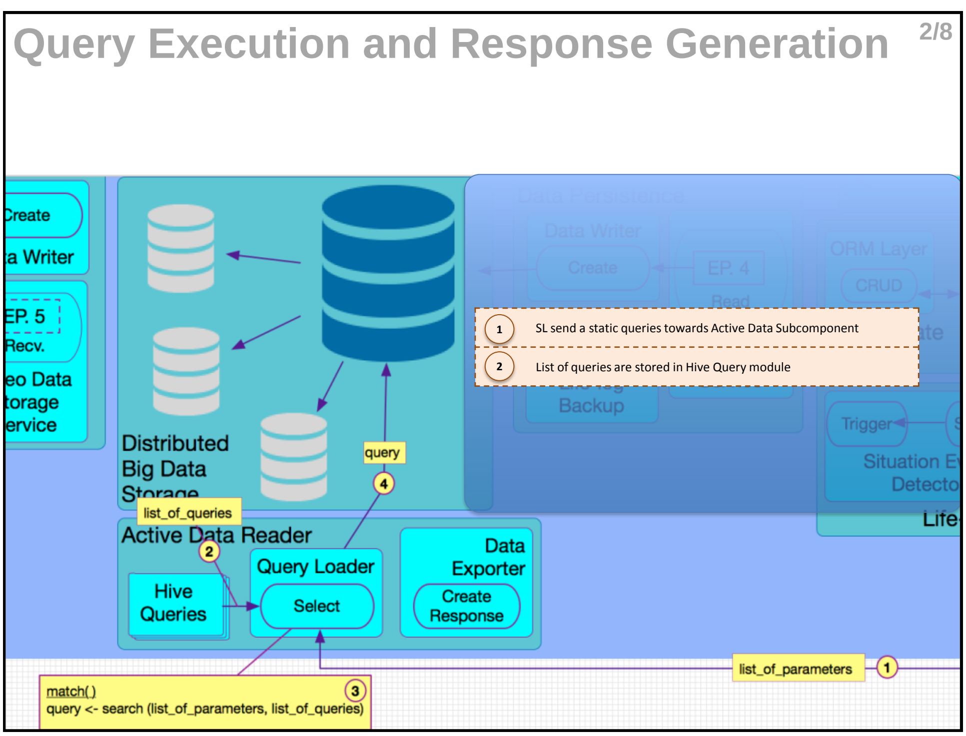

SL Request of Parameters
1/1
Information Curation Layer
Supporting Layer
Data Curation Layer
DCL Service
EP. 7
Recv.
Request/
Response
Send
1
SL send a static queries towards Active Data Subcomponent
send()
list_of_parameters{ age, location, activity }
Create
Data Writer
EP. 5
Recv.
Video Data Storage Service
Data Writer
Create
EP. 4
Read
Create
Life-log Backup
Raw Data Storage Service
ORM Layer
CRUD
Intermediate Database
Situation Configurator
Create
Trigger
Scan
Situation Event Detector
Create
Constraints Configurator
Life-log Monitoring
Distributed Big Data Storage
Active Data Reader
Hive Queries
Query Loader
Select
Data Exporter
Create Response


Hive Query Selection
1/3
Create
a Writer
EP. 5
Recv.
eo Data
torage
ervice
Distributed Big Data Storage
1
SL send a static queries towards Active Data Subcomponent
2
List of queries are stored in Hive Query module
query
4
list_of_queries
Active Data Reader
2
Hive Queries
Query Loader
Select
Data Exporter
Create Response
list_of_parameters
1
match()
3
query <- search (list_of_parameters, list_of_queries)


Hive Query Selection
2/3
Create
a Writer
EP. 5
Recv.
eo Data
torage
ervice
Distributed
Big Data
Storage
query
4
list_of_queries
Active Data Reader
2
Query Loader
Hive
Queries
Select
Data
Exporter
Create
Response
1 SL send a static queries towards Active Data Subcomponent
2 List of queries are stored in Hive Query module
3 Static queries are invoked by active data reader subcomponent.
Situation E
Detecto
Life
list_of_parameters
1
match( )
3
query <- search (list_of_parameters, list_of_queries)

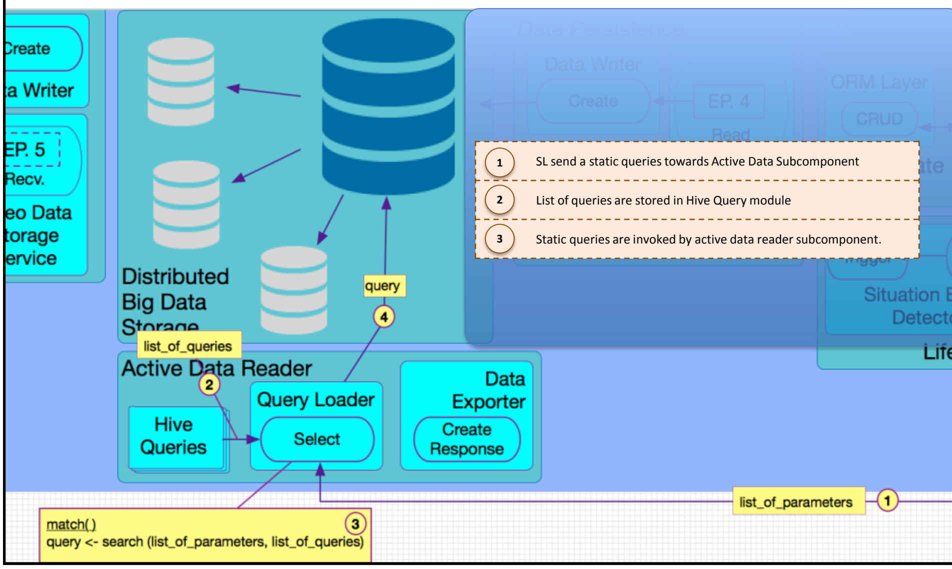

Hive Query Selection
3/3
Create
a Writer
EP. 5
Recv.
eo Data
torage
ervice
Distributed
Big Data
Storage
query
4
list_of_queries
Active Data Reader
2
Query Loader
Hive
Queries
Select
Data
Exporter
Create
Response
1 SL send a static queries towards Active Data Subcomponent
2 List of queries are stored in Hive Query module
3 Static queries are invoked by active data reader subcomponent.
4 Queries are send to the name node
Detecto
Life
list_of_parameters
1
match( )
3
query <- search (list_of_parameters, list_of_queries)


Query Execution and Response Generation
1/8
Create
a Writer
EP. 5
Recv.
eo Data
torage
ervice
Distributed
Big Data
Storage
query
4
list_of_queries
Active Data Reader
2
Query Loader
Select
Hive
Queries
Data
Exporter
Create
Response
1
SL send a static queries towards Active Data Subcomponent
Data Persistence
Data Writer
Create
EP. 4
Read
Storage
Service
Life-log
Backup
ORM Layer
CRUD
Database
Trigger
Situation E
Detecto
Life
list_of_parameters
1
match()
3
query <- search (list_of_parameters, list_of_queries)


Query Execution and Response Generation
2/8
Create
a Writer
EP. 5
Recv.
eo Data
torage
ervice
Distributed
Big Data
Storage
query
4
list_of_queries
Active Data Reader
2
Query Loader
Select
Hive
Queries
Data
Exporter
Create
Response
1
SL send a static queries towards Active Data Subcomponent
2
List of queries are stored in Hive Query module
Data Persistence
Data Writer
Create
EP. 4
Read
Backup
ORM Layer
CRUD
Trigger
Situation E
Detecto
Life
list_of_parameters
1
match()
3
query <- search (list_of_parameters, list_of_queries)

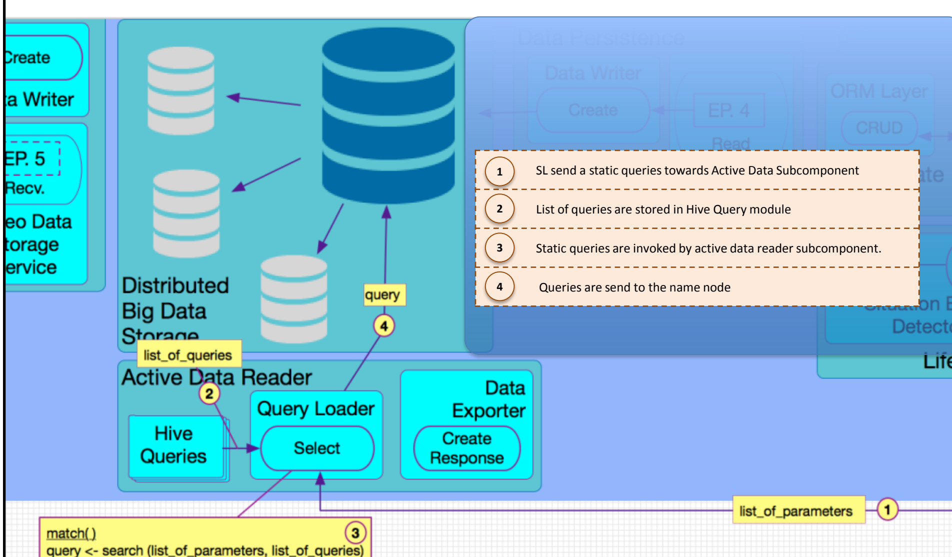
Query Execution and Response Generation

3/8



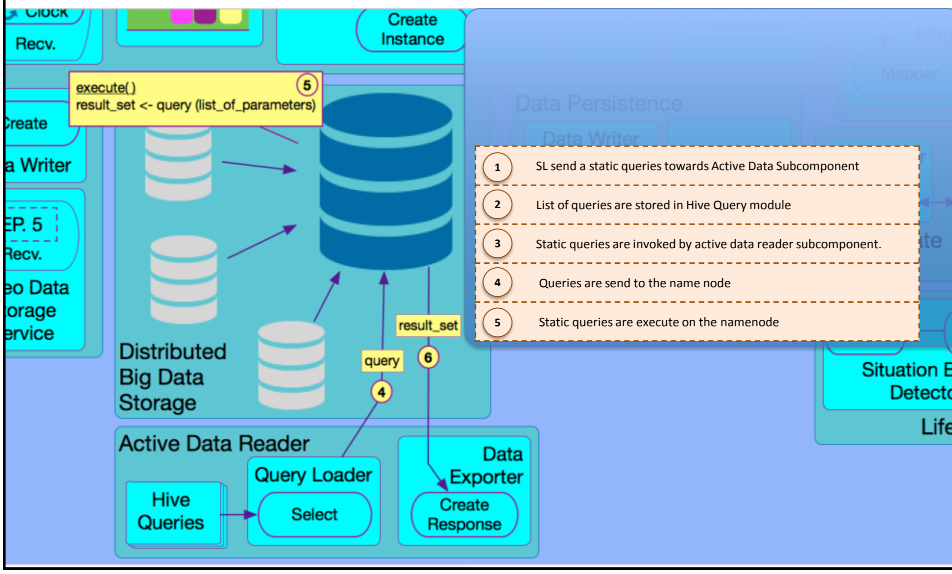
Query Execution and Response Generation

4/8



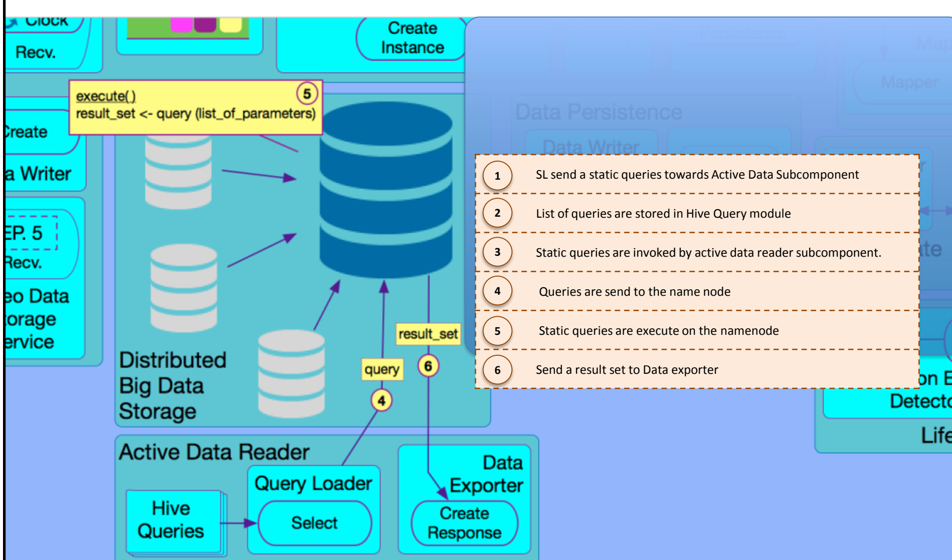
Query Execution and Response Generation

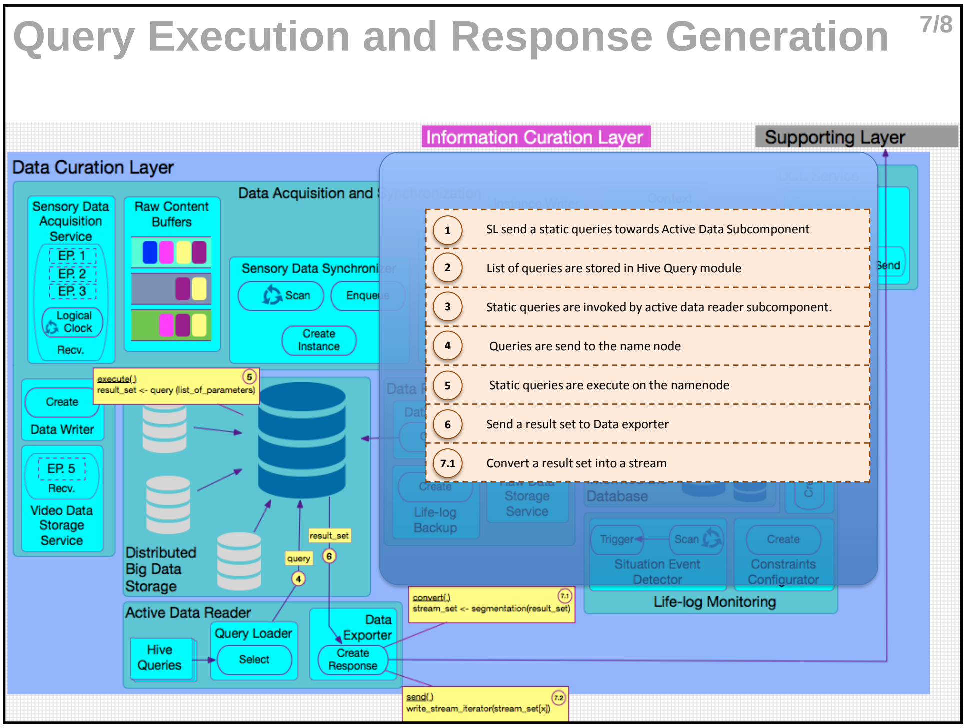
5/8

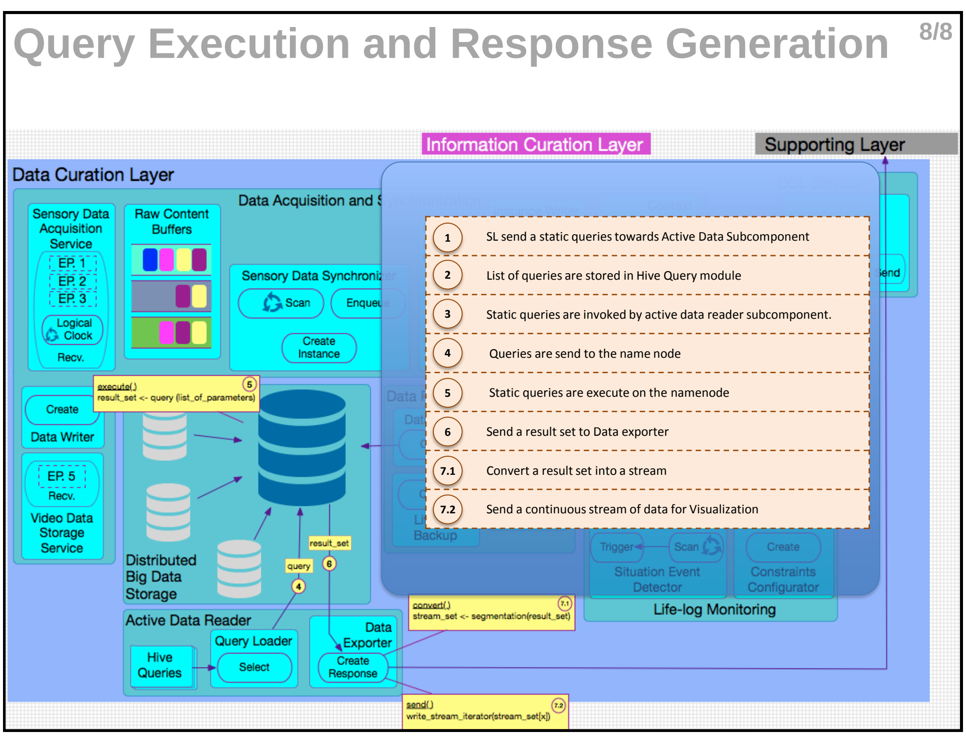


Query Execution and Response Generation

6/8




Query Execution and Response Generation
7/8
Information Curation Layer
Supporting Layer
Data Curation Layer
Data Acquisition and
Sensory Data Acquisition Service
EP. 1
EP. 2
EP. 3
Logical Clock
Recv.
Raw Content Buffers
Sensory Data Synchroni
Scan
Create Instance
Create
Data Writer
EP. 5
Recv.
Video Data Storage Service
execute()
result_set <- query (list_of_parameters)
5
Distributed Big Data Storage
result_set
query
6
4
Active Data Reader
Hive Queries
Query Loader
Select
Data Exporter
Create Response
convert()
stream_set <- segmentation(result_set)
7.1
send()
write_stream_iterator(stream_set[x])
7.2
Life-log Backup
Storage Service
Database
Trigger
Scan
Situation Event Detector
Create
Constraints Configurator
Life-log Monitoring
1 SL send a static queries towards Active Data Subcomponent
2 List of queries are stored in Hive Query module
3 Static queries are invoked by active data reader subcomponent.
4 Queries are send to the name node
5 Static queries are execute on the namenode
6 Send a result set to Data exporter
7.1 Convert a result set into a stream

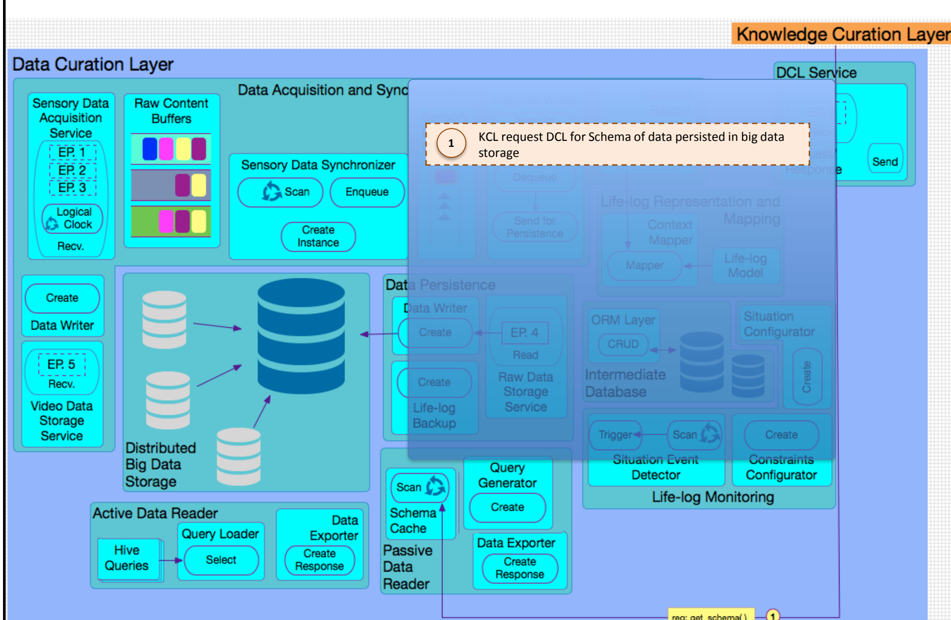

Query Execution and Response Generation
8/8
Information Curation Layer
Supporting Layer
Data Curation Layer
Data Acquisition and
Sensory Data Acquisition Service
EP. 1
EP. 2
EP. 3
Logical Clock
Recv.
Raw Content Buffers
Sensory Data Synchroni
Scan
Create Instance
Create
Data Writer
EP. 5
Recv.
Video Data Storage Service
execute()
result_set <- query (list_of_parameters)
5
Distributed Big Data Storage
result_set
query
6
4
Active Data Reader
Hive Queries
Query Loader
Select
Data Exporter
Create Response
convert()
stream_set <- segmentation(result_set)
7.1
send()
write_stream_iterator(stream_set[x])
7.2
Backup
Trigger
Scan
Situation Event Detector
Create
Constraints Configurator
Life-log Monitoring
1 SL send a static queries towards Active Data Subcomponent
2 List of queries are stored in Hive Query module
3 Static queries are invoked by active data reader subcomponent.
4 Queries are send to the name node
5 Static queries are execute on the namenode
6 Send a result set to Data exporter
7.1 Convert a result set into a stream
7.2 Send a continuous stream of data for Visualization

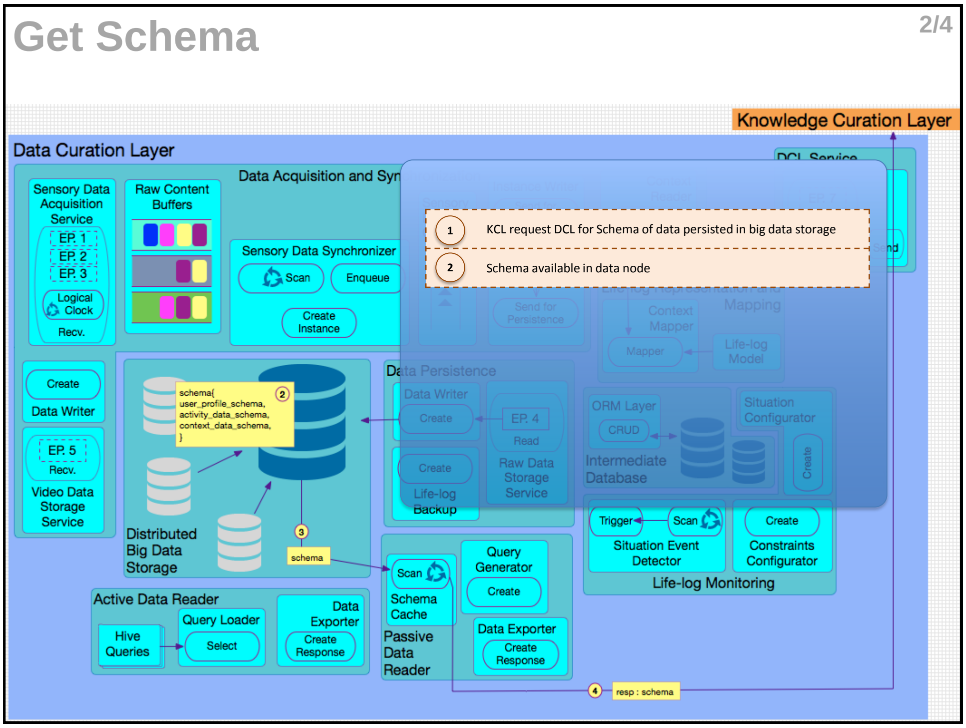
Scenario 3: Off-line (Passive)

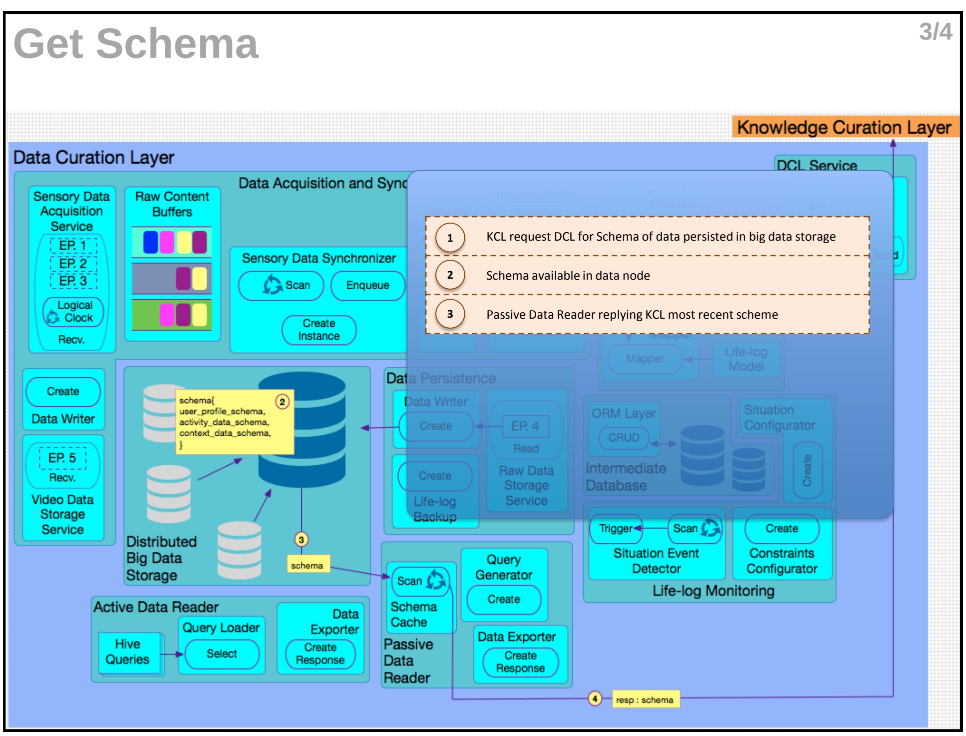
Desc. : Raw-sensory data from big data storage is requested for **model training** by KCL.

Get Schema

1/4

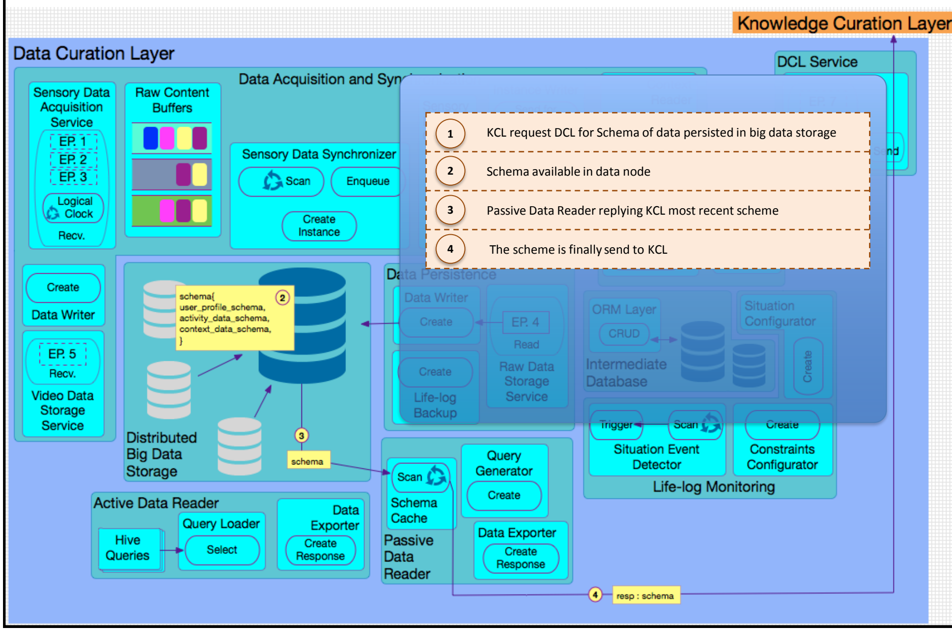



Get Schema
2/4
Knowledge Curation Layer
Data Curation Layer
Data Acquisition and Syn
DCL Service
1 KCL request DCL for Schema of data persisted in big data storage
2 Schema available in data node
Sensory Data Acquisition Service
EP. 1
EP. 2
EP. 3
Logical Clock
Recv.
Raw Content Buffers
Sensory Data Synchronizer
Scan
Enqueue
Create Instance
Create
Data Writer
EP. 5
Recv.
Video Data Storage Service
schema{
user_profile_schema,
activity_data_schema,
context_data_schema,
}
Distributed Big Data Storage
schema
Data Persistence
Data Writer
Create
EP. 4
Read
Raw Data Storage Service
Life-log Backup
ORM Layer
CRUD
Intermediate Database
Situation Configurator
Context Mapper
Mapper
Mapping
Life-log Model
Trigger
Scan
Situation Event Detector
Create
Constraints Configurator
Life-log Monitoring
Scan
Schema Cache
Query Generator
Create
Data Exporter
Create Response
Passive Data Reader
Active Data Reader
Hive Queries
Query Loader
Select
Data Exporter
Create Response
resp : schema


Get Schema
3/4
Knowledge Curation Layer
Data Curation Layer
Data Acquisition and Sync
DCL Service
1 KCL request DCL for Schema of data persisted in big data storage
2 Schema available in data node
3 Passive Data Reader replying KCL most recent scheme
Sensory Data Acquisition Service
EP. 1
EP. 2
EP. 3
Logical Clock
Recv.
Raw Content Buffers
Sensory Data Synchronizer
Scan
Enqueue
Create Instance
Create
Data Writer
EP. 5
Recv.
Video Data Storage Service
schema{
user_profile_schema,
activity_data_schema,
context_data_schema,
}
Distributed Big Data Storage
schema
Data Persistence
Data Writer
Create
EP. 4
Read
Raw Data Storage Service
Life-log Backup
ORM Layer
CRUD
Intermediate Database
Situation Configurator
Mapper
Life-log Model
Trigger
Scan
Situation Event Detector
Create
Constraints Configurator
Life-log Monitoring
Scan
Schema Cache
Query Generator
Create
Data Exporter
Create Response
Passive Data Reader
Active Data Reader
Hive Queries
Query Loader
Select
Data Exporter
Create Response
resp : schema

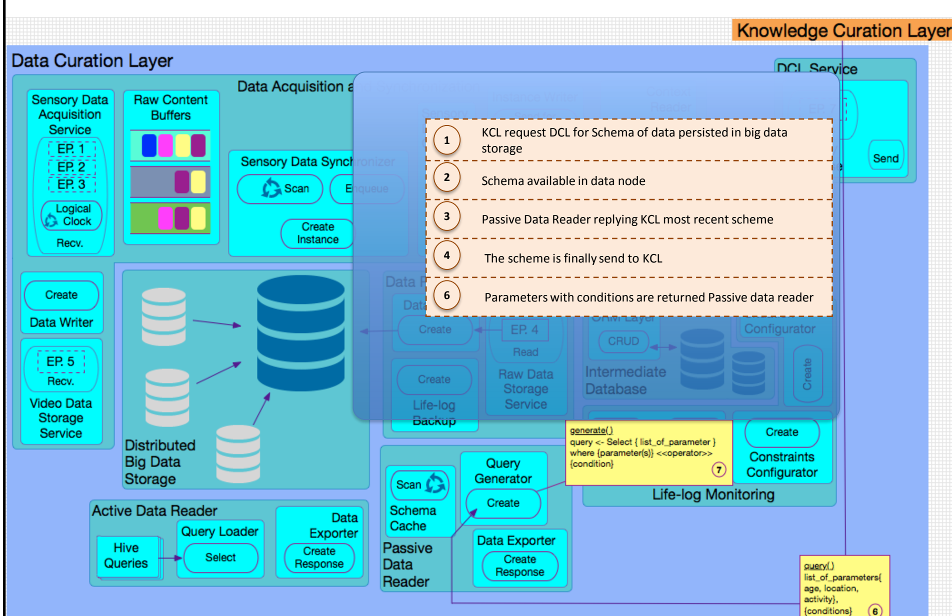
Get Schema

4/4



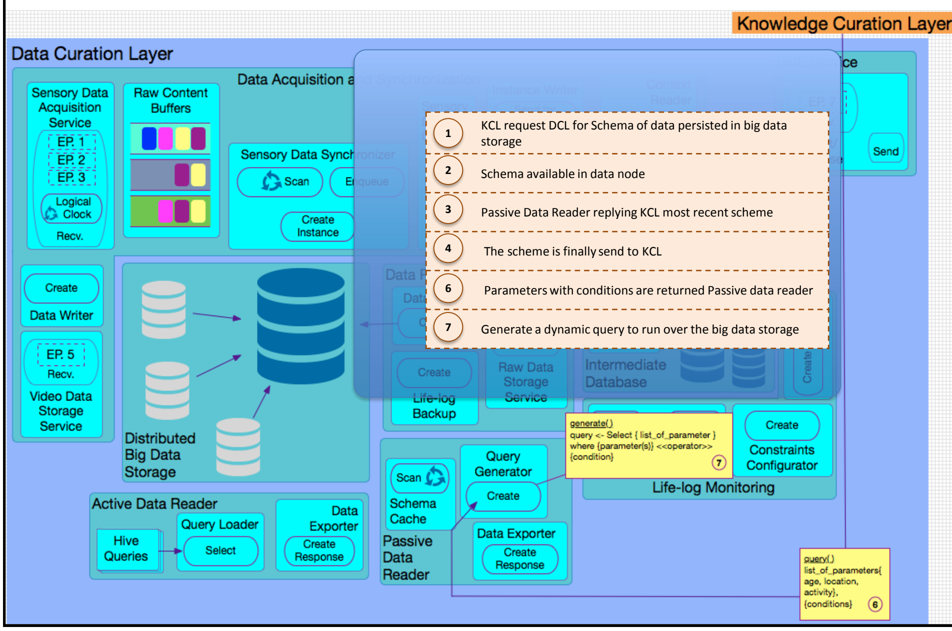
Generate and Execute Query

1/2



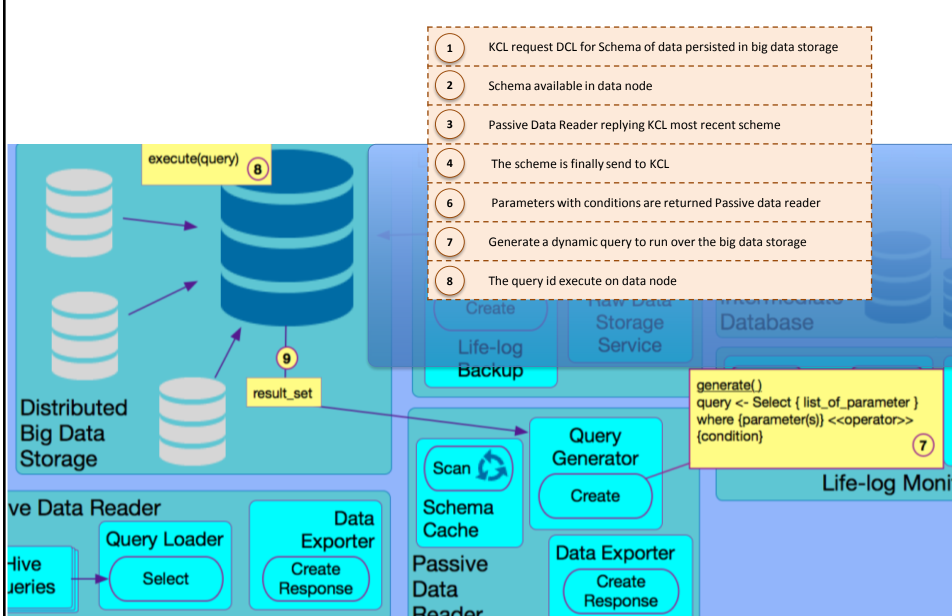
Generate and Execute Query

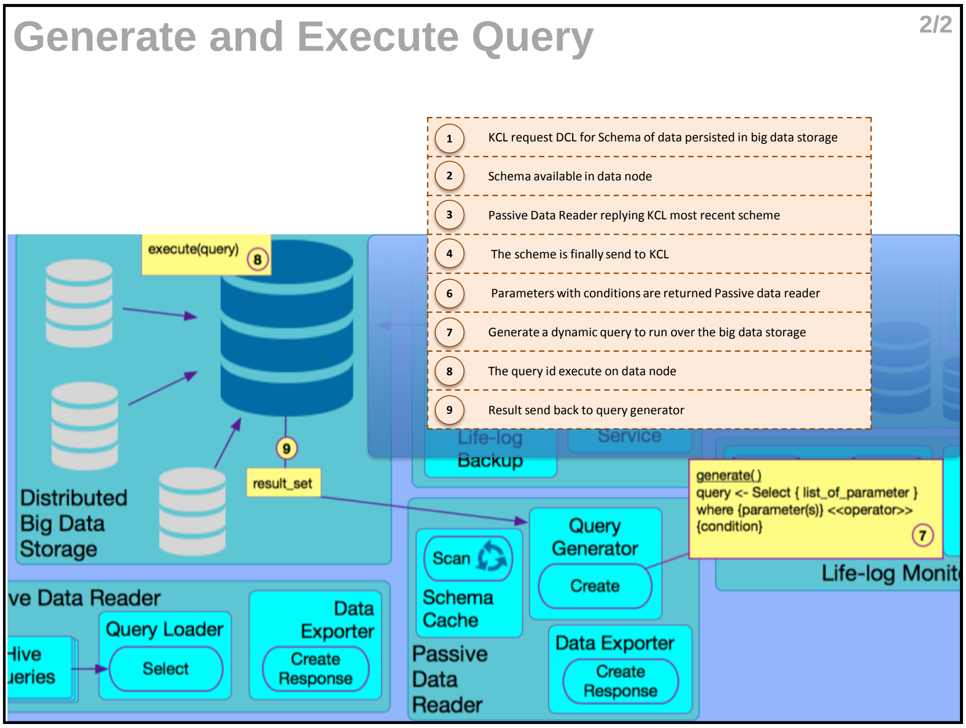
1/2

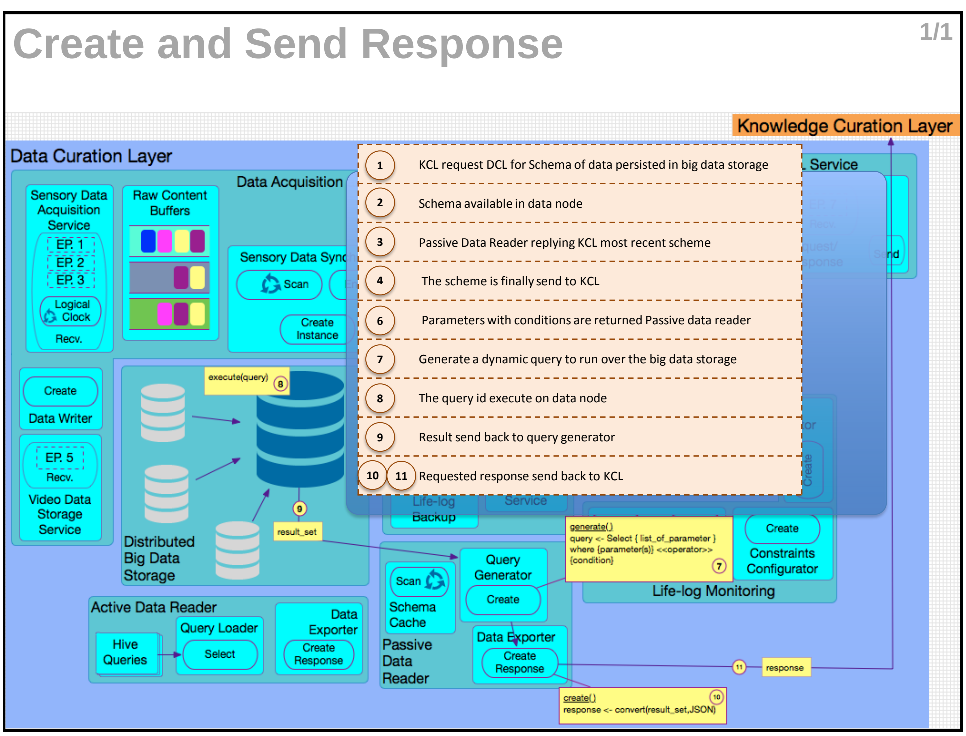


Generate and Execute Query

2/2




Generate and Execute Query
2/2
1 KCL request DCL for Schema of data persisted in big data storage
2 Schema available in data node
3 Passive Data Reader replying KCL most recent scheme
4 The scheme is finally send to KCL
6 Parameters with conditions are returned Passive data reader
7 Generate a dynamic query to run over the big data storage
8 The query id execute on data node
9 Result send back to query generator
execute(query) 8
9
result_set
Distributed Big Data Storage
generate()
query <- Select { list_of_parameter } where {parameter(s)} <<operator>> {condition} 7
Life-log Backup
Service
Query Generator
Create
Scan
Schema Cache
Life-log Monit
ve Data Reader
Query Loader
Select
Data Exporter
Create Response
Hive ueries
Passive Data Reader
Data Exporter
Create Response

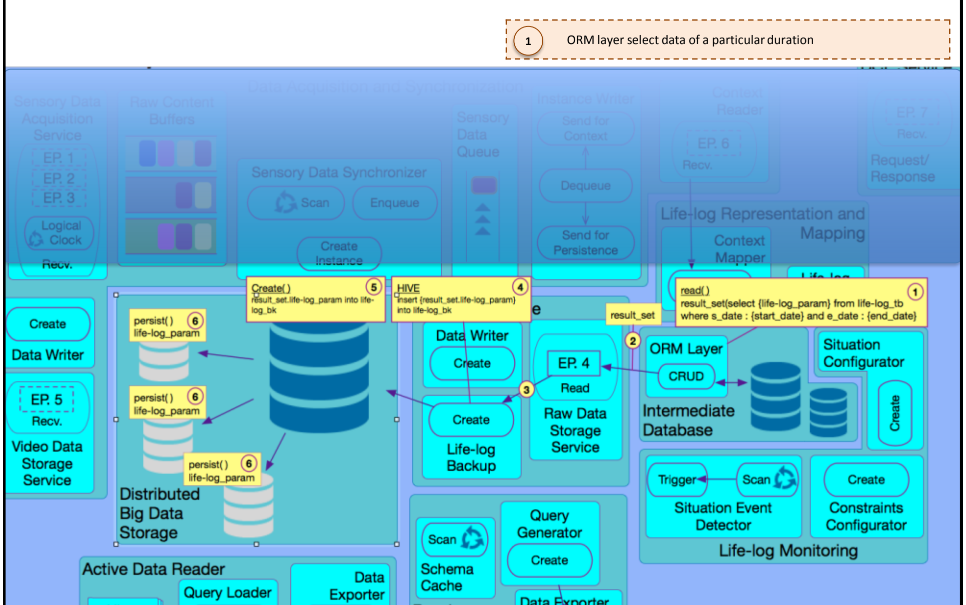

Create and Send Response
1/1
Knowledge Curation Layer
Data Curation Layer
1 KCL request DCL for Schema of data persisted in big data storage
2 Schema available in data node
3 Passive Data Reader replying KCL most recent scheme
4 The scheme is finally send to KCL
6 Parameters with conditions are returned Passive data reader
7 Generate a dynamic query to run over the big data storage
8 The query id execute on data node
9 Result send back to query generator
10 11 Requested response send back to KCL
Data Acquisition
Sensory Data Acquisition Service
EP. 1
EP. 2
EP. 3
Logical Clock
Recv.
Raw Content Buffers
Sensory Data Sync
Scan
Create Instance
Create
Data Writer
EP. 5
Recv.
Video Data Storage Service
execute(query) 8
9
result_set
Distributed Big Data Storage
Life-log Backup
Service
Query Generator
Create
Scan
Schema Cache
Passive Data Reader
Data Exporter
Create Response
generate()
query <- Select { list_of_parameter } where {parameter(s)} <<operator>> {condition} 7
Create
Constraints Configurator
Life-log Monitoring
Active Data Reader
Hive Queries
Query Loader
Select
Data Exporter
Create Response
11 response
create() 10
response <- convert(result_set,JSON)
Service

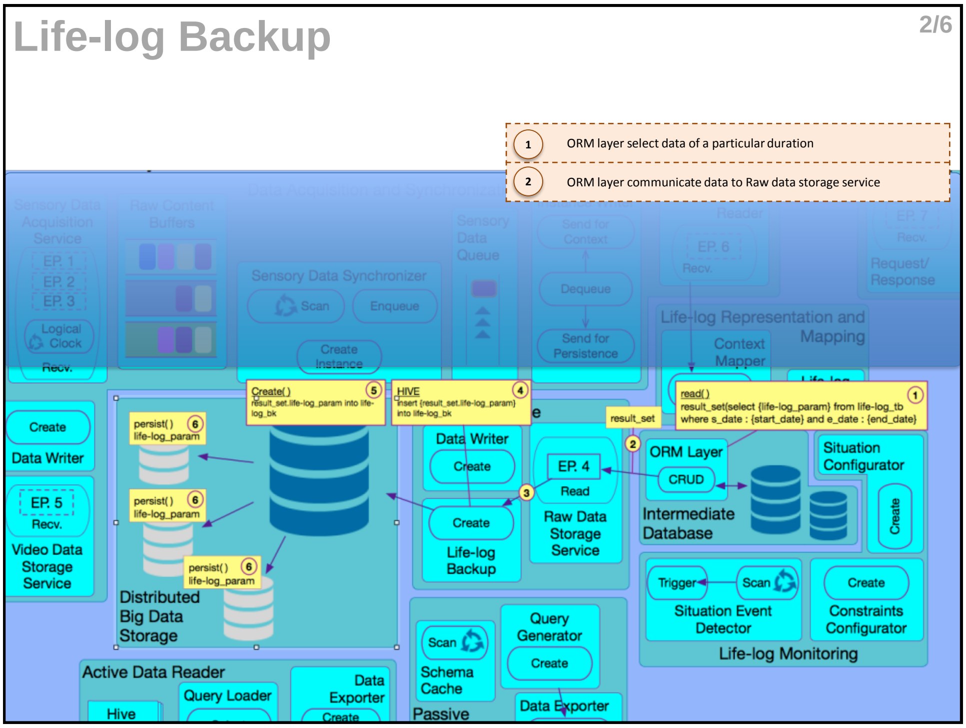
Scenario 4: Periodic

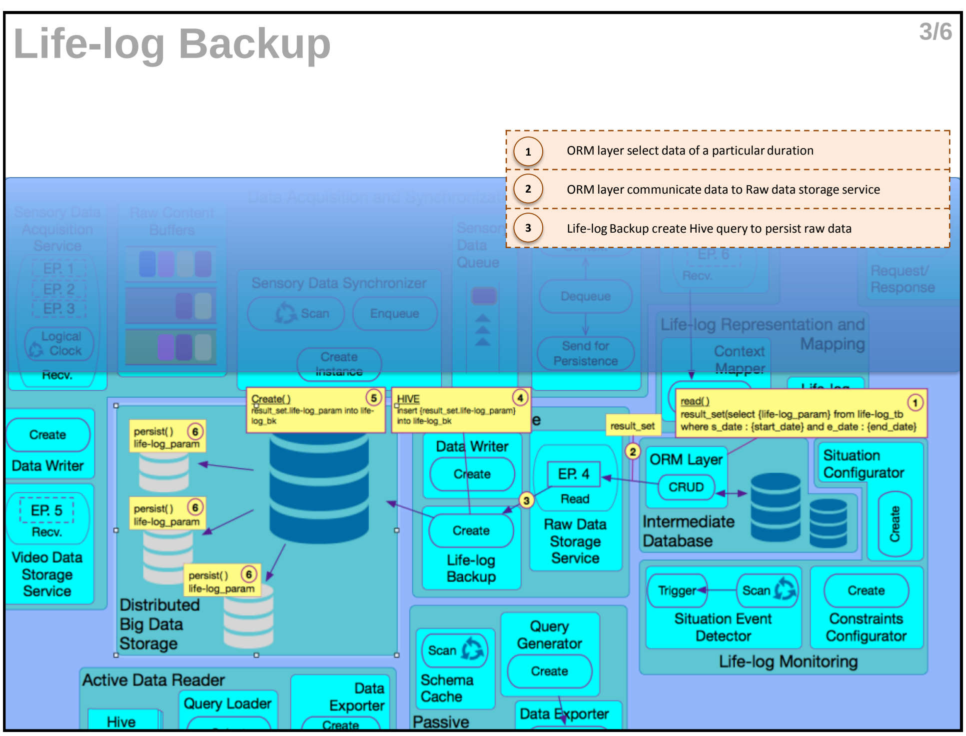
Desc. : **Life-log** data is **backed up** over big data storage
Video data contents from **Kinect source** are **backed Up** in big data storage

Life-log Backup

1/6



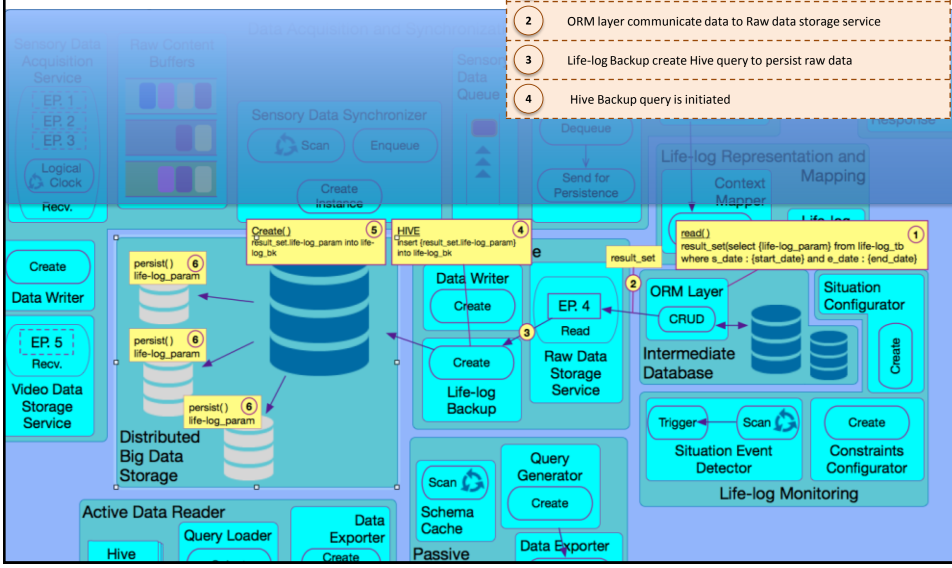

Life-log Backup
2/6
1 ORM layer select data of a particular duration
2 ORM layer communicate data to Raw data storage service
read( ) result_set(select {life-log_param} from life-log_tb where s_date : {start_date} and e_date : {end_date}
result_set
HIVE insert {result_set.life-log_param} into life-log_bk
Create( ) result_set.life-log_param into life-log_bk
persist( ) life-log_param
Data Writer
Create
EP. 5
Recv.
Video Data Storage Service
Distributed Big Data Storage
Life-log Backup
EP. 4
Read
Raw Data Storage Service
ORM Layer
CRUD
Intermediate Database
Situation Configurator
Trigger
Scan
Situation Event Detector
Constraints Configurator
Life-log Monitoring
Schema Cache
Query Generator
Data Exporter
Active Data Reader
Query Loader
Hive
Passive
Life-log Representation and Mapping
Context Mapper


Life-log Backup
3/6
1 ORM layer select data of a particular duration
2 ORM layer communicate data to Raw data storage service
3 Life-log Backup create Hive query to persist raw data
read( ) result_set(select {life-log_param} from life-log_tb where s_date : {start_date} and e_date : {end_date}
result_set
HIVE insert {result_set.life-log_param} into life-log_bk
Create( ) result_set.life-log_param into life-log_bk
persist( ) life-log_param
Data Writer
Create
EP. 5
Recv.
Video Data Storage Service
Distributed Big Data Storage
Life-log Backup
EP. 4
Read
Raw Data Storage Service
ORM Layer
CRUD
Intermediate Database
Situation Configurator
Trigger
Scan
Situation Event Detector
Constraints Configurator
Life-log Monitoring
Schema Cache
Query Generator
Data Exporter
Active Data Reader
Query Loader
Hive
Passive
Life-log Representation and Mapping
Context Mapper

Life-log Backup

4/6

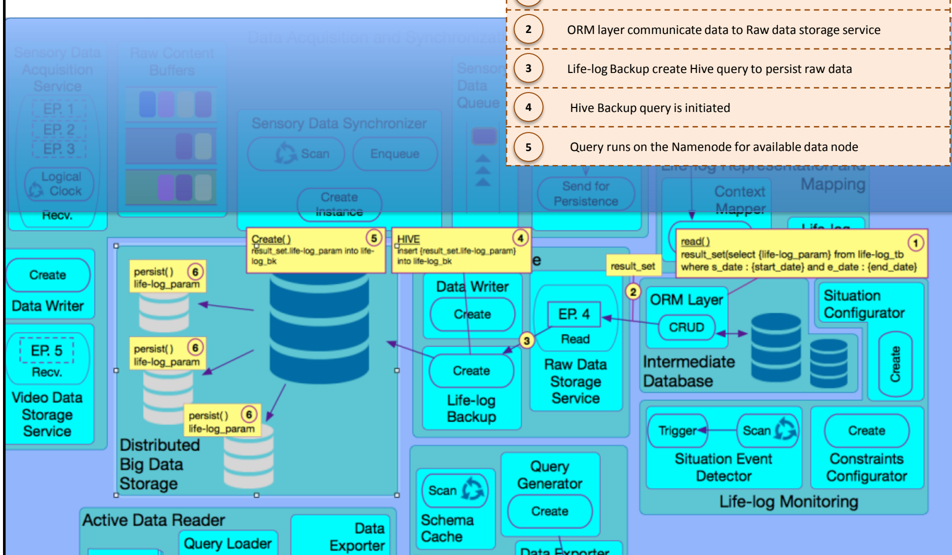
1. ORM layer select data of a particular duration
2. ORM layer communicate data to Raw data storage service
3. Life-log Backup create Hive query to persist raw data
4. Hive Backup query is initiated

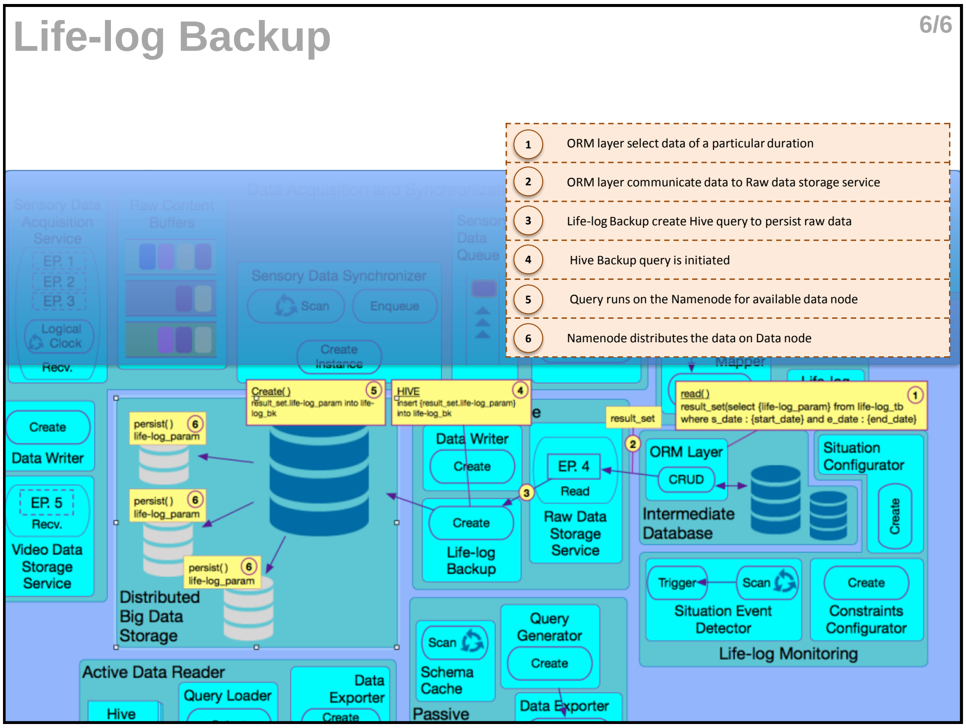


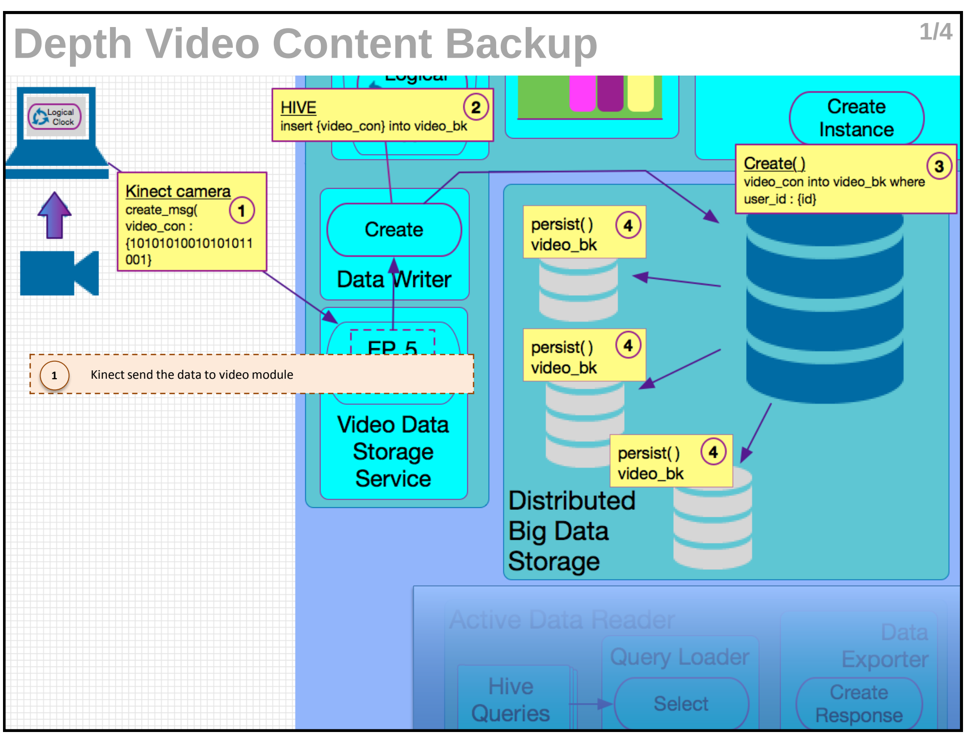
Life-log Backup

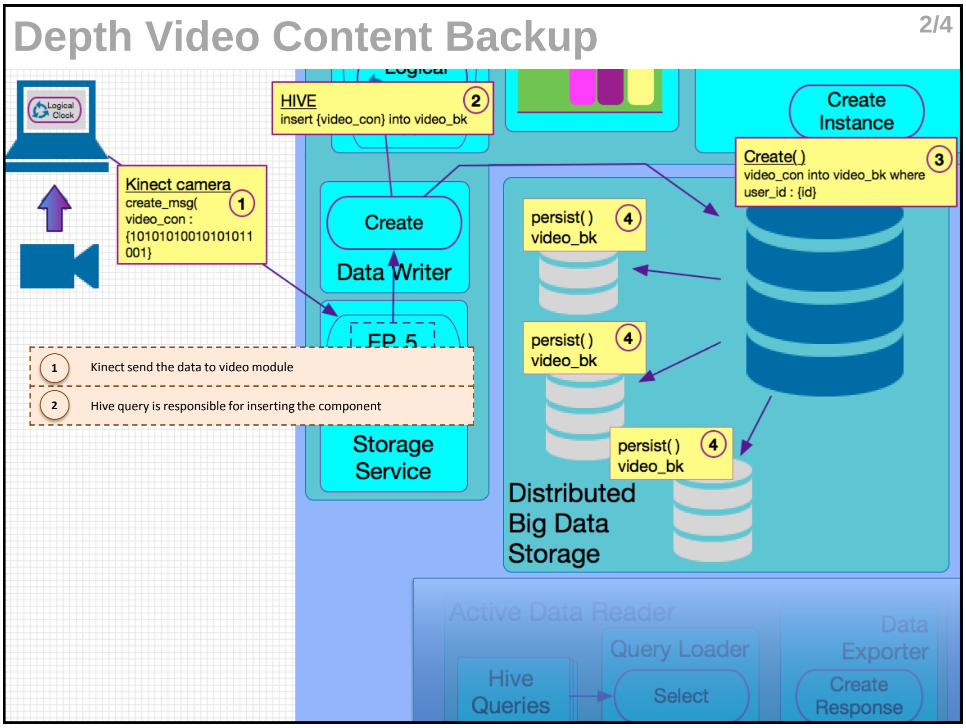
5/6

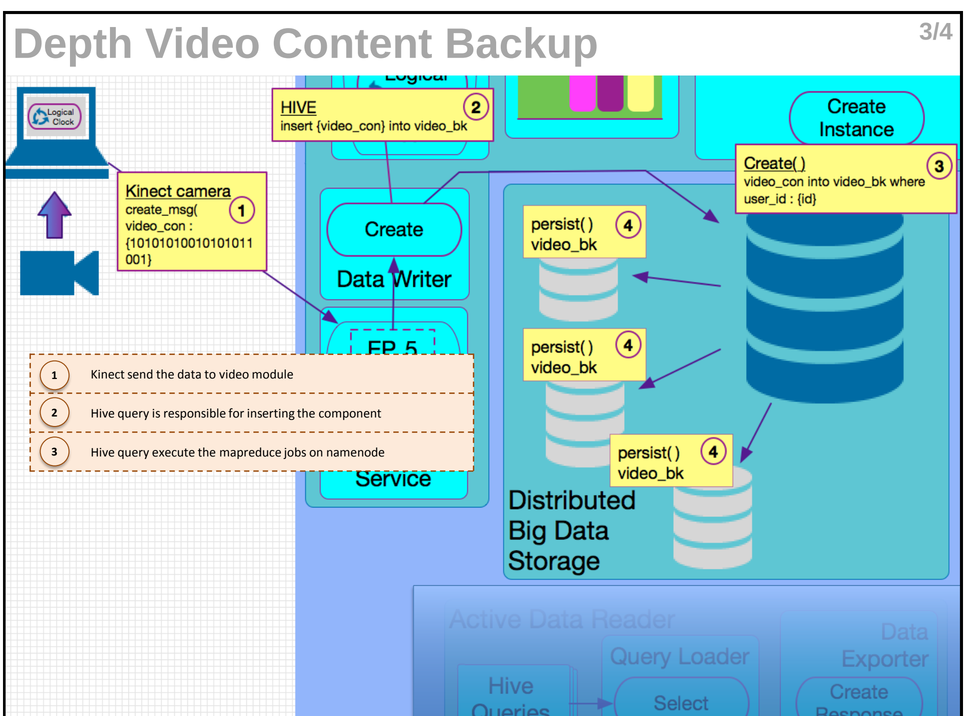
1. ORM layer select data of a particular duration
2. ORM layer communicate data to Raw data storage service
3. Life-log Backup create Hive query to persist raw data
4. Hive Backup query is initiated
5. Query runs on the Namenode for available data node

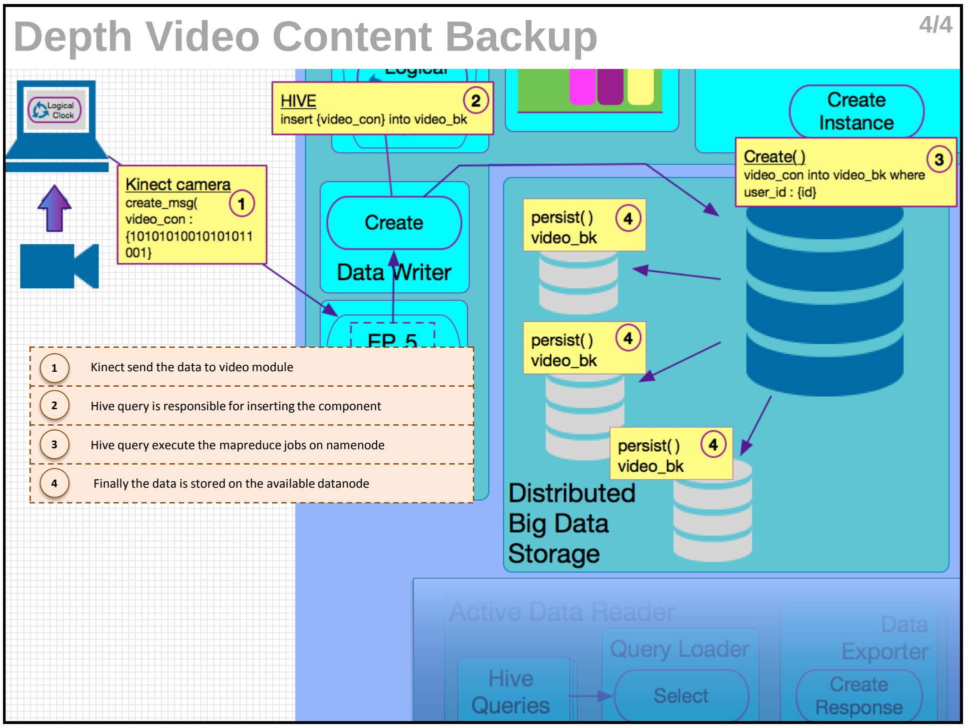



Life-log Backup
6/6
1 ORM layer select data of a particular duration
2 ORM layer communicate data to Raw data storage service
3 Life-log Backup create Hive query to persist raw data
4 Hive Backup query is initiated
5 Query runs on the Namenode for available data node
6 Namenode distributes the data on Data node
Create( ) result_set.life-log_param into life-log_bk
HIVE insert {result_set.life-log_param} into life-log_bk
read( ) result_set(select {life-log_param} from life-log_tb where s_date : {start_date} and e_date : {end_date}
result_set
persist( ) life-log_param
Create
Data Writer
EP. 5
Recv.
Video Data Storage Service
Distributed Big Data Storage
Create
Life-log Backup
EP. 4
Read
Raw Data Storage Service
ORM Layer
CRUD
Intermediate Database
Situation Configurator
Trigger
Scan
Situation Event Detector
Constraints Configurator
Life-log Monitoring
Scan
Schema Cache
Query Generator
Passive
Data Exporter
Active Data Reader
Query Loader
Hive


Depth Video Content Backup
1/4
Logical Clock
Kinect camera
create_msg(video_con : {10101010010101011 001}
HIVE
insert {video_con} into video_bk
Create Instance
Create( )
video_con into video_bk where user_id : {id}
Create
Data Writer
EP. 5
1 Kinect send the data to video module
Video Data Storage Service
persist( ) video_bk
Distributed Big Data Storage
Active Data Reader
Query Loader
Hive Queries
Select
Data Exporter
Create Response

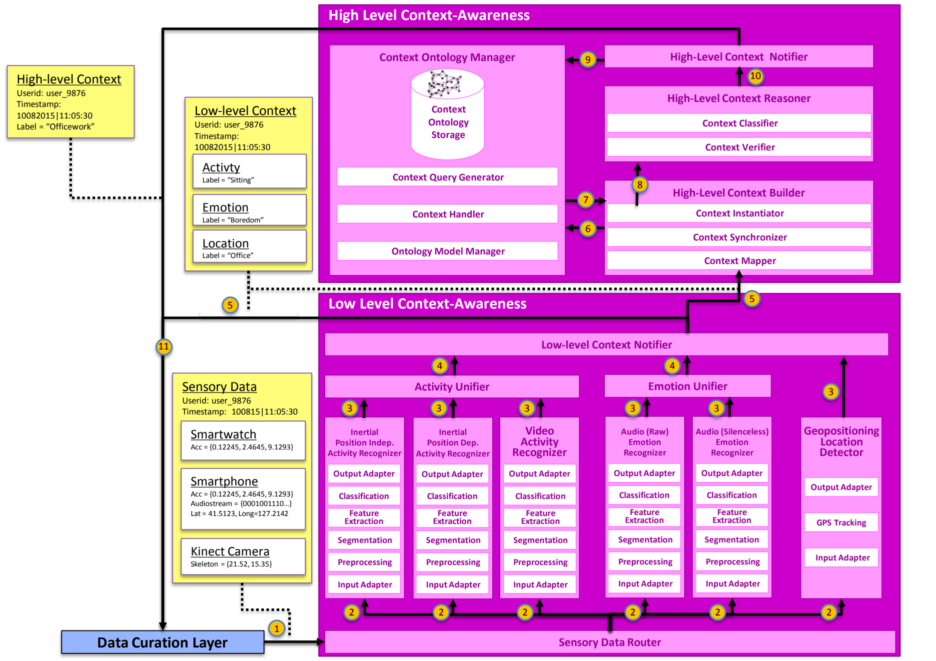

Depth Video Content Backup
2/4
Logical Clock
HIVE
insert {video_con} into video_bk
2
Create Instance
Create( )
video_con into video_bk where user_id : {id}
3
Kinect camera
create_msg(video_con : {10101010010101011 001}
1
Create
Data Writer
FP 5
persist( ) video_bk
4
1 Kinect send the data to video module
2 Hive query is responsible for inserting the component
Storage Service
Distributed Big Data Storage
Active Data Reader
Query Loader
Data Exporter
Hive Queries
Select
Create Response


Depth Video Content Backup
3/4
Logical Clock
HIVE
insert {video_con} into video_bk
2
Create Instance
Create( )
video_con into video_bk where user_id : {id}
3
Kinect camera
create_msg(video_con : {10101010010101011 001}
1
Create
Data Writer
FP 5
persist( ) video_bk
4
1 Kinect send the data to video module
2 Hive query is responsible for inserting the component
3 Hive query execute the mapreduce jobs on namenode
Service
Distributed Big Data Storage
Active Data Reader
Query Loader
Data Exporter
Hive Queries
Select
Create Response



STENCILS for ICL Micro-Demos

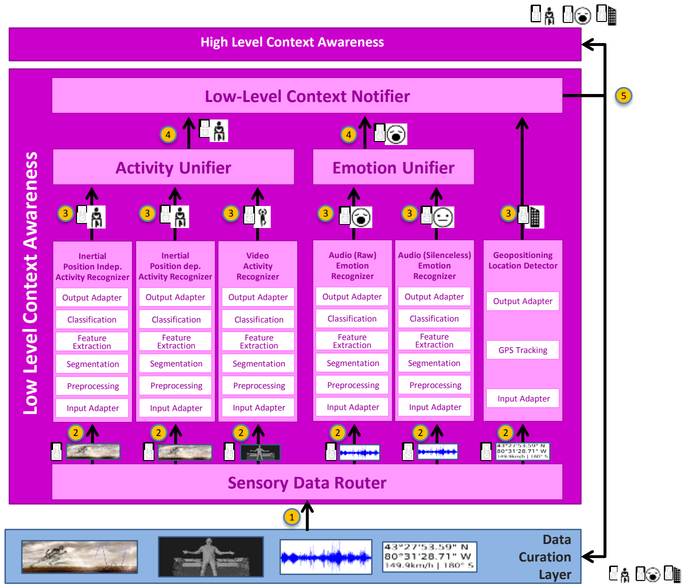
ICL Detailed Architecture



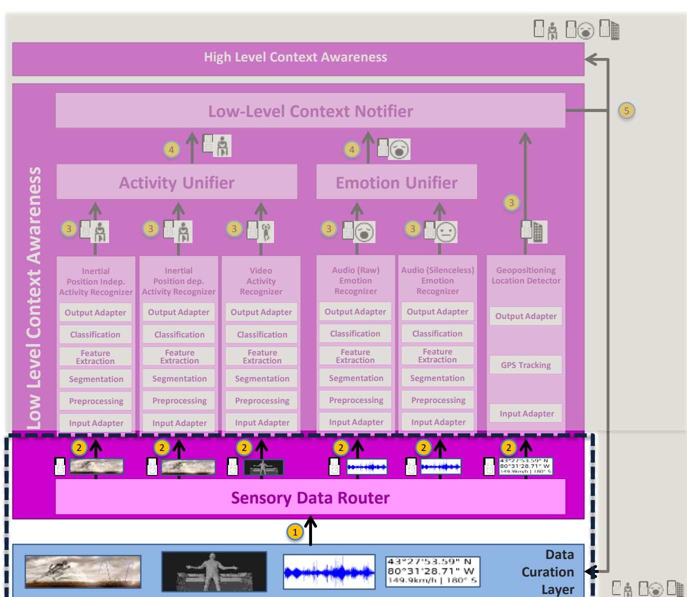
Scenario 1: Low level Context Recognition

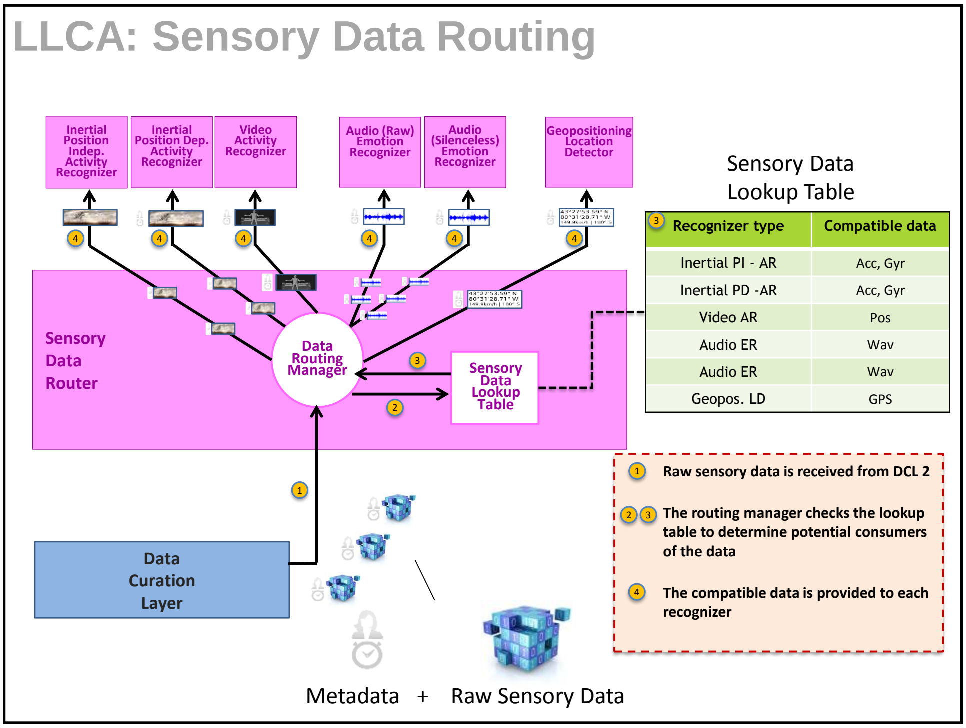
Desc. : Low-level Context is **recognized** based on Raw-sensory data from multimodal data sources which is received by DCL.

LLCA: Overall

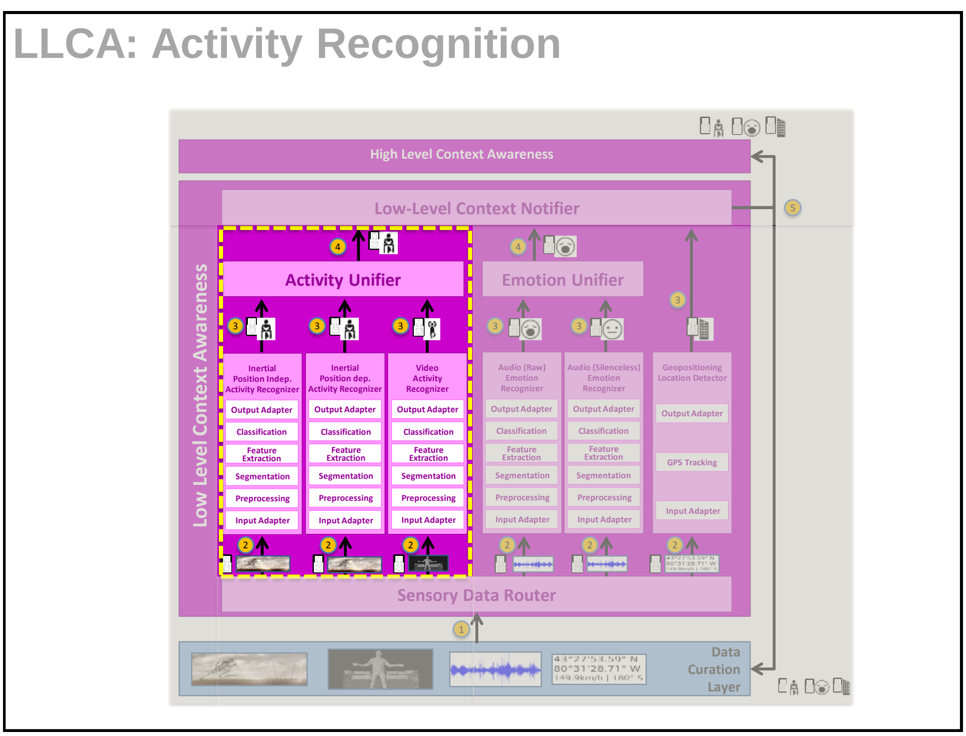


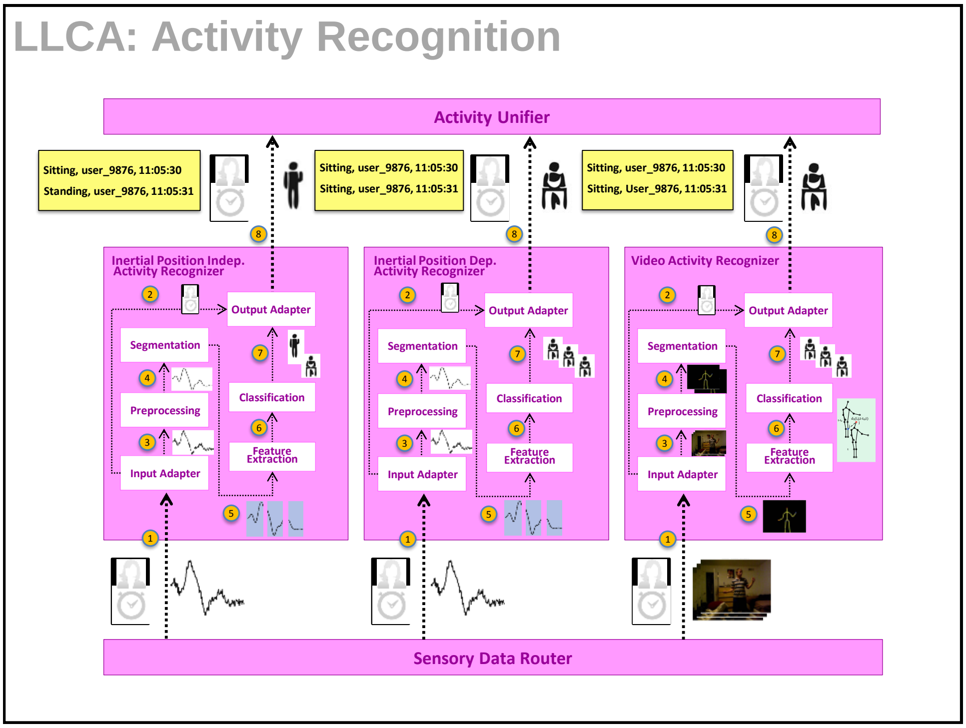
LLCA: Sensory Data Routing

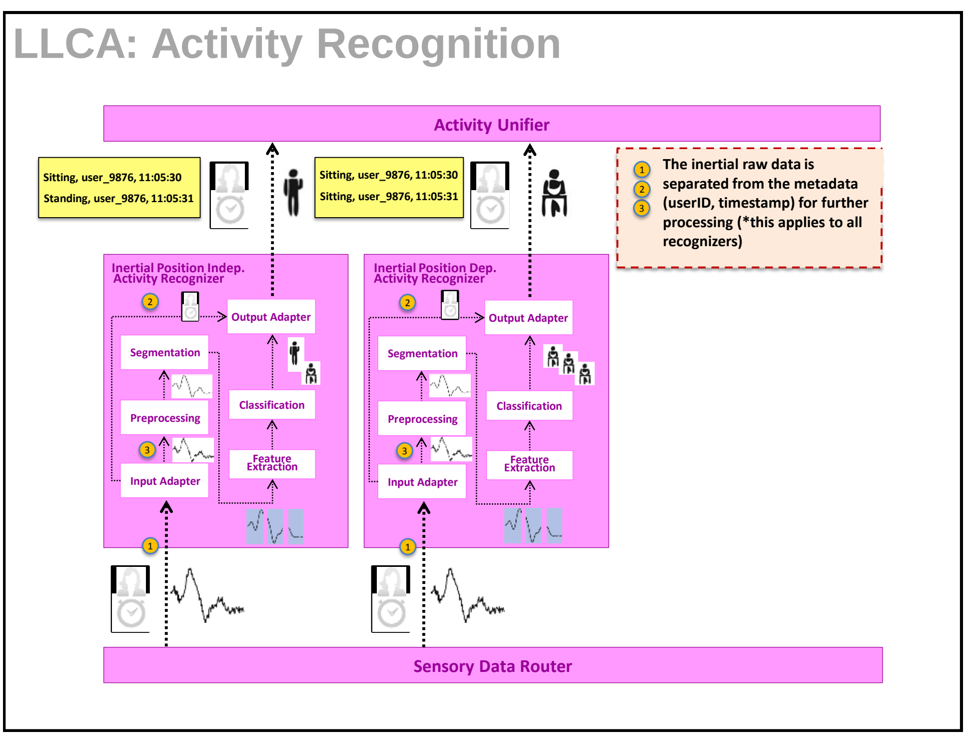


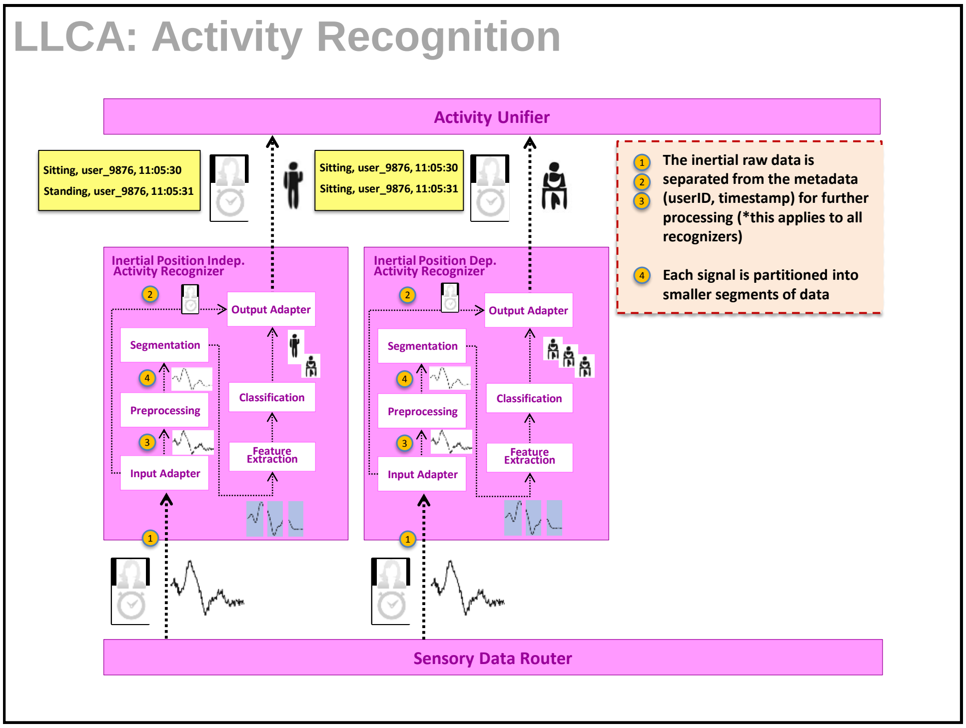


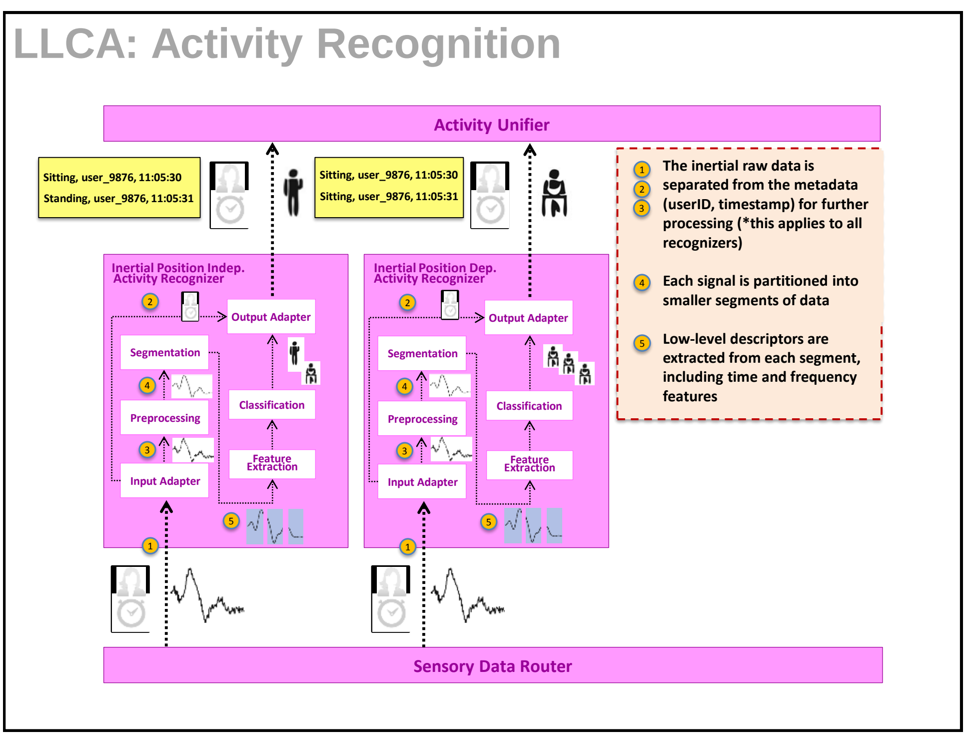
3 Recognizer type	Compatible data
Inertial PI - AR	Acc, Gyr
Inertial PD -AR	Acc, Gyr
Video AR	Pos
Audio ER	Wav
Audio ER	Wav
Geopos. LD	GPS



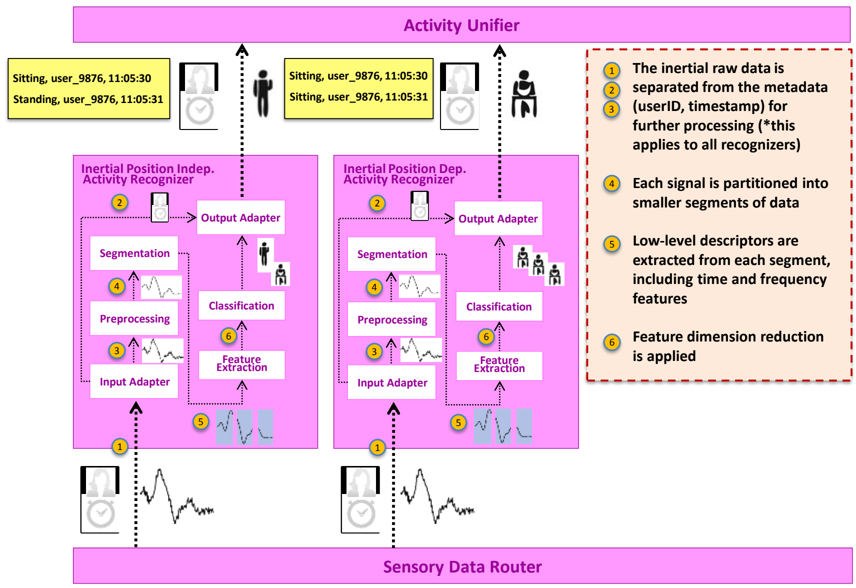

LLCA: Activity Recognition
Activity Unifier
Sitting, user_9876, 11:05:30
Standing, user_9876, 11:05:31
Sitting, user_9876, 11:05:30
Sitting, user_9876, 11:05:31
Sitting, user_9876, 11:05:30
Sitting, User_9876, 11:05:31
Inertial Position Indep. Activity Recognizer
Output Adapter
Segmentation
Classification
Preprocessing
Feature Extraction
Input Adapter
Inertial Position Dep. Activity Recognizer
Output Adapter
Segmentation
Classification
Preprocessing
Feature Extraction
Input Adapter
Video Activity Recognizer
Output Adapter
Segmentation
Classification
Preprocessing
Feature Extraction
Input Adapter
Sensory Data Router


LLCA: Activity Recognition
Activity Unifier
Sitting, user_9876, 11:05:30
Standing, user_9876, 11:05:31
Sitting, user_9876, 11:05:30
Sitting, user_9876, 11:05:31
The inertial raw data is separated from the metadata (userID, timestamp) for further processing (*this applies to all recognizers)
Inertial Position Indep. Activity Recognizer
Output Adapter
Segmentation
Classification
Preprocessing
Feature Extraction
Input Adapter
Inertial Position Dep. Activity Recognizer
Output Adapter
Segmentation
Classification
Preprocessing
Feature Extraction
Input Adapter
Sensory Data Router

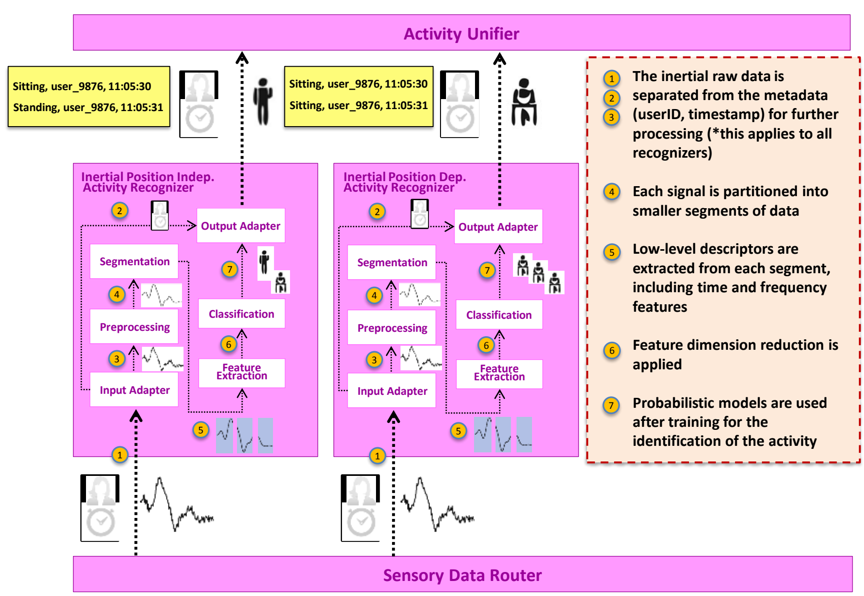

LLCA: Activity Recognition
Activity Unifier
Sitting, user_9876, 11:05:30
Standing, user_9876, 11:05:31
Sitting, user_9876, 11:05:30
Sitting, user_9876, 11:05:31
Inertial Position Indep. Activity Recognizer
Inertial Position Dep. Activity Recognizer
Output Adapter
Segmentation
Classification
Preprocessing
Feature Extraction
Input Adapter
Sensory Data Router
1 2 3 The inertial raw data is separated from the metadata (userID, timestamp) for further processing (*this applies to all recognizers)
4 Each signal is partitioned into smaller segments of data

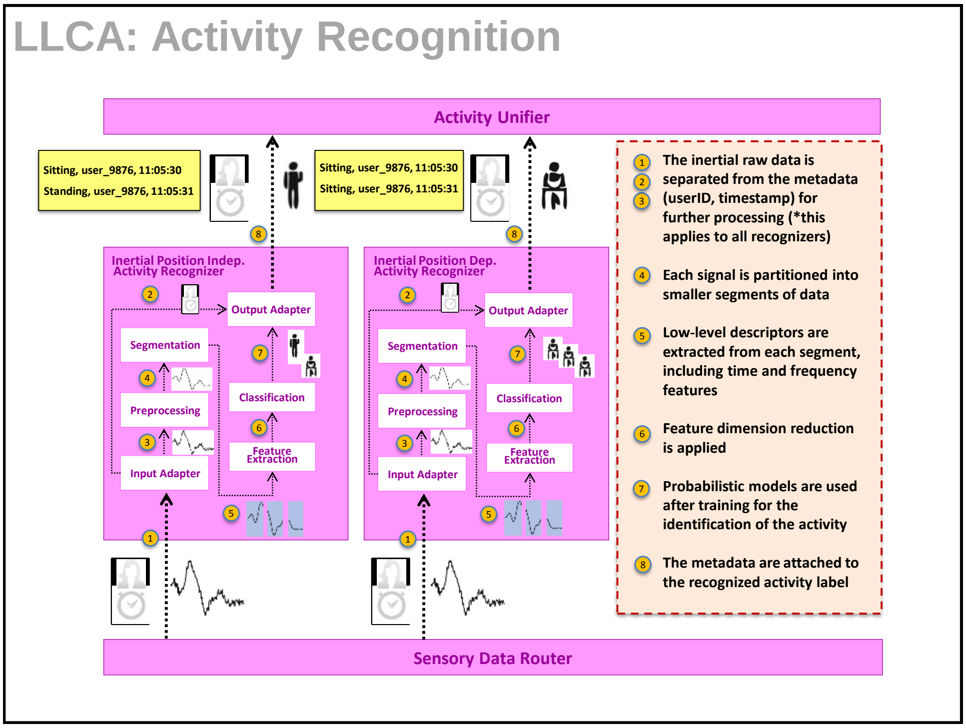

LLCA: Activity Recognition
Activity Unifier
Sitting, user_9876, 11:05:30
Standing, user_9876, 11:05:31
Sitting, user_9876, 11:05:30
Sitting, user_9876, 11:05:31
Inertial Position Indep. Activity Recognizer
Inertial Position Dep. Activity Recognizer
Output Adapter
Segmentation
Classification
Preprocessing
Feature Extraction
Input Adapter
Sensory Data Router
1 2 3 The inertial raw data is separated from the metadata (userID, timestamp) for further processing (*this applies to all recognizers)
4 Each signal is partitioned into smaller segments of data
5 Low-level descriptors are extracted from each segment, including time and frequency features

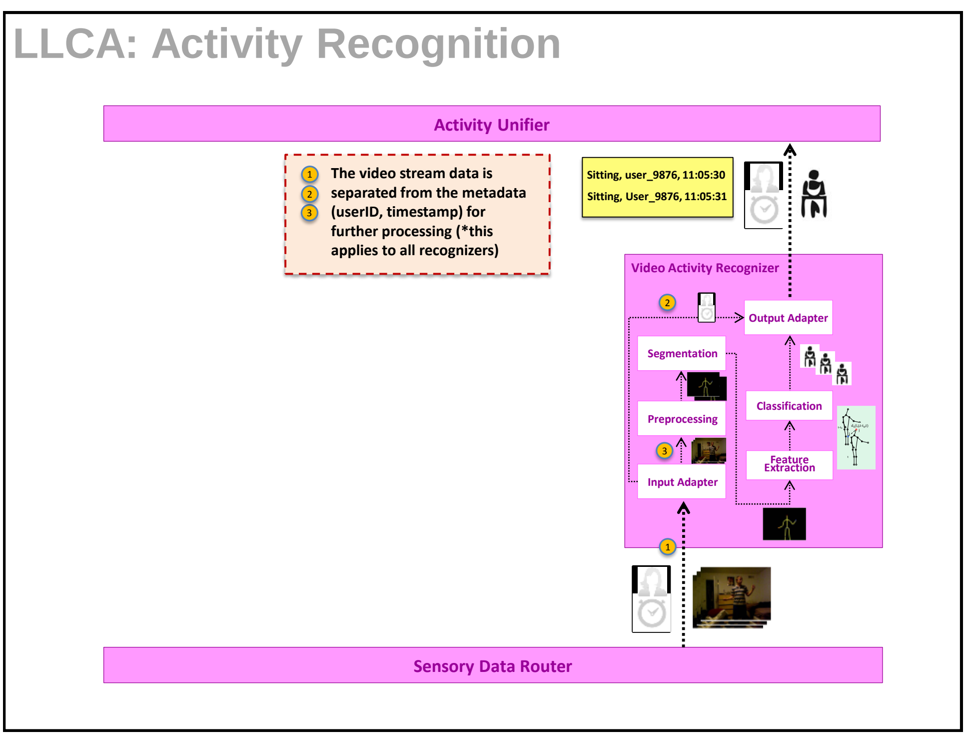
LLCA: Activity Recognition



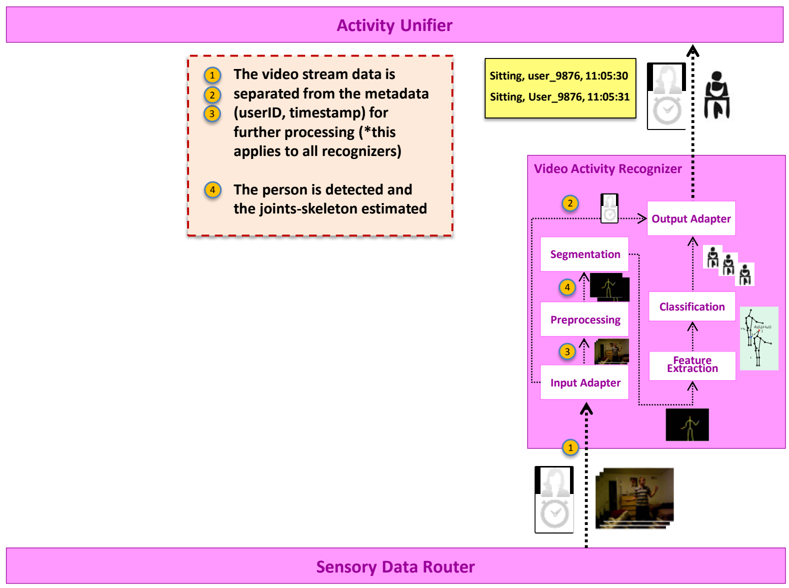
LLCA: Activity Recognition



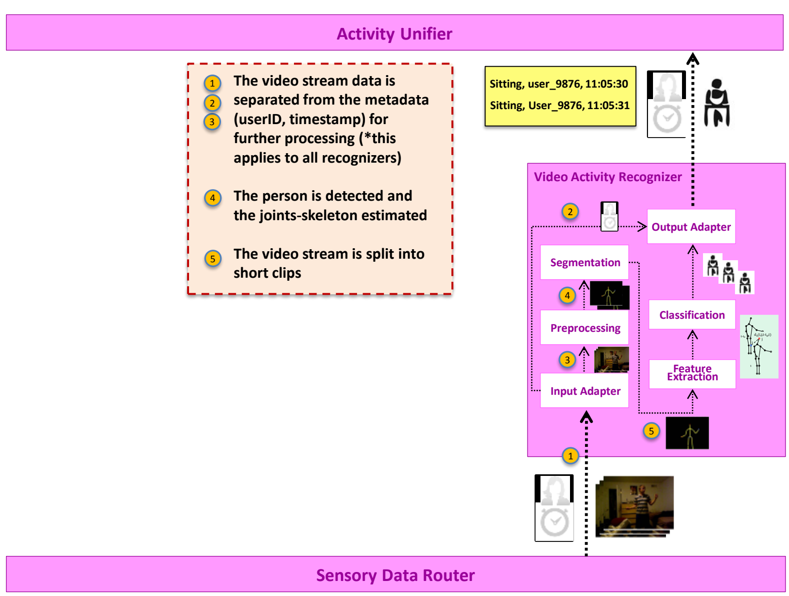

LLCA: Activity Recognition
Activity Unifier
Sitting, user_9876, 11:05:30
Standing, user_9876, 11:05:31
Sitting, user_9876, 11:05:30
Sitting, user_9876, 11:05:31
Inertial Position Indep. Activity Recognizer
Inertial Position Dep. Activity Recognizer
Output Adapter
Segmentation
Classification
Preprocessing
Feature Extraction
Input Adapter
Output Adapter
Segmentation
Classification
Preprocessing
Feature Extraction
Input Adapter
1 2 3 The inertial raw data is separated from the metadata (userID, timestamp) for further processing (*this applies to all recognizers)
4 Each signal is partitioned into smaller segments of data
5 Low-level descriptors are extracted from each segment, including time and frequency features
6 Feature dimension reduction is applied
7 Probabilistic models are used after training for the identification of the activity
8 The metadata are attached to the recognized activity label
Sensory Data Router


LLCA: Activity Recognition
Activity Unifier
1 2 3 The video stream data is separated from the metadata (userID, timestamp) for further processing (*this applies to all recognizers)
Sitting, user_9876, 11:05:30
Sitting, User_9876, 11:05:31
Video Activity Recognizer
Output Adapter
Segmentation
Classification
Preprocessing
Feature Extraction
Input Adapter
Sensory Data Router

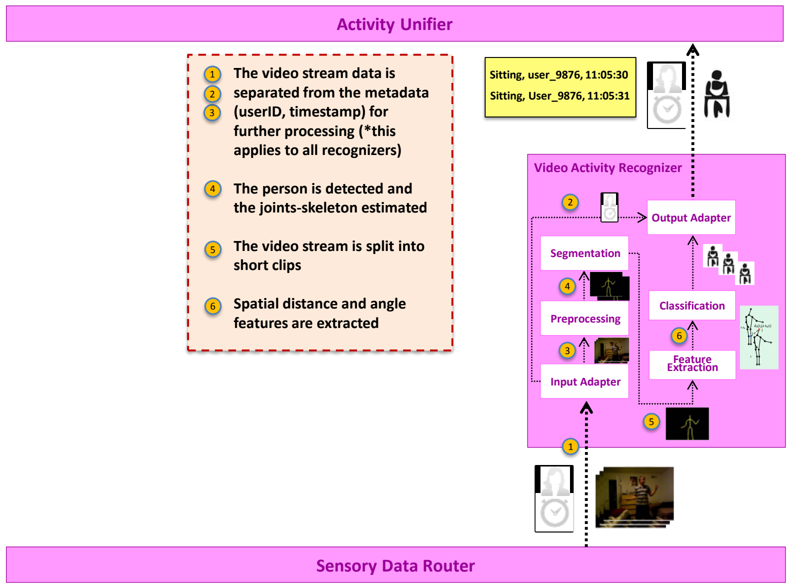
LLCA: Activity Recognition



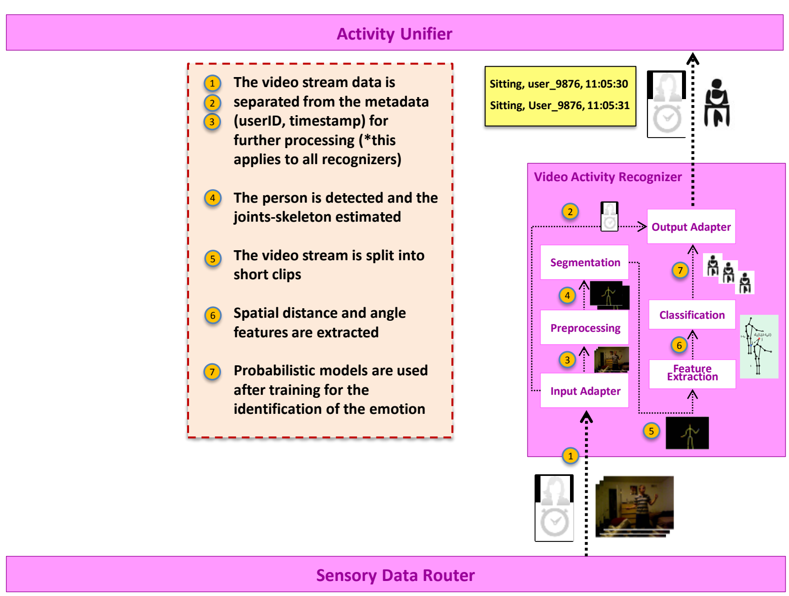
LLCA: Activity Recognition

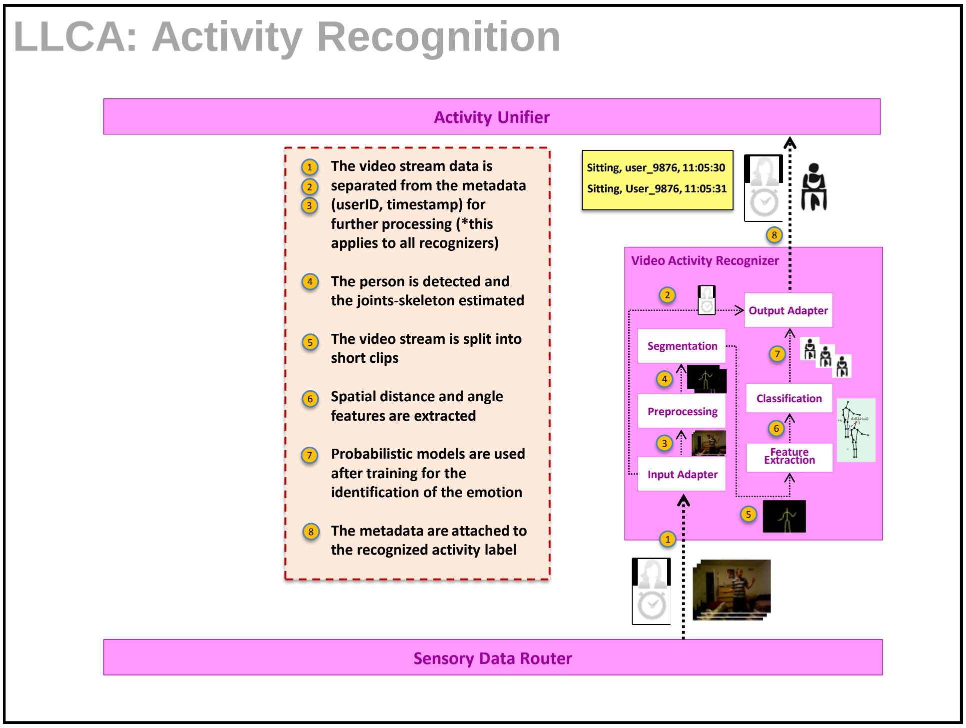


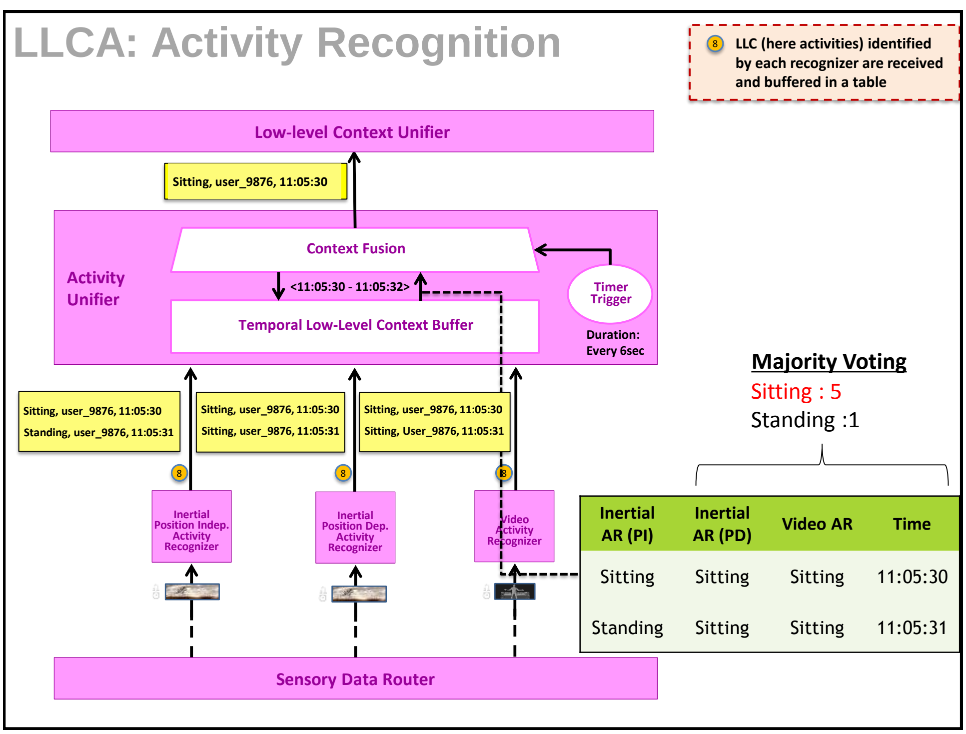
LLCA: Activity Recognition

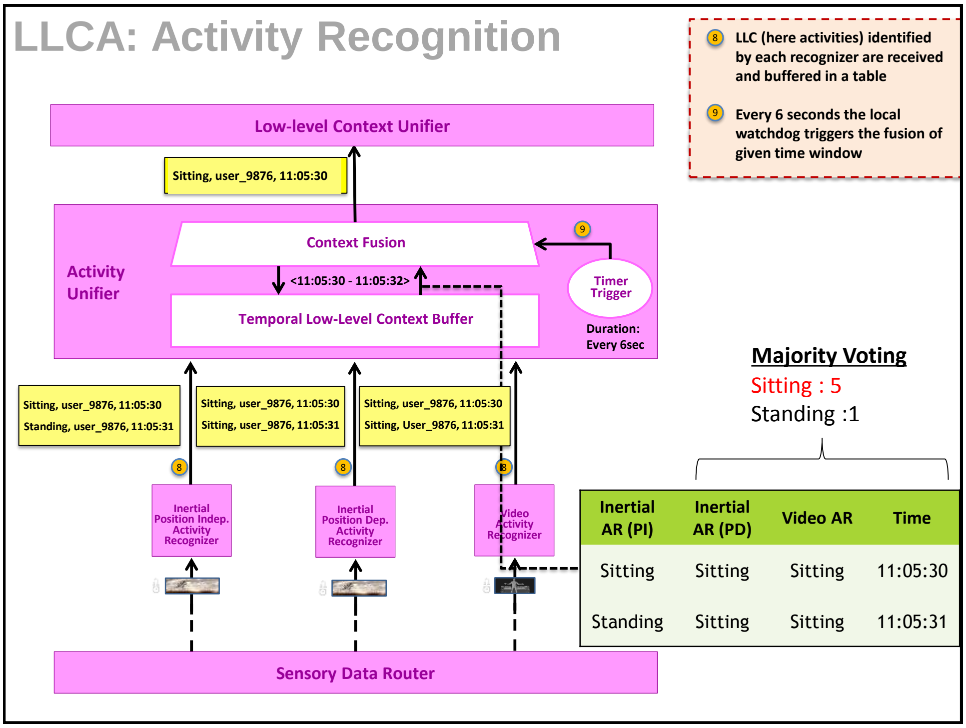


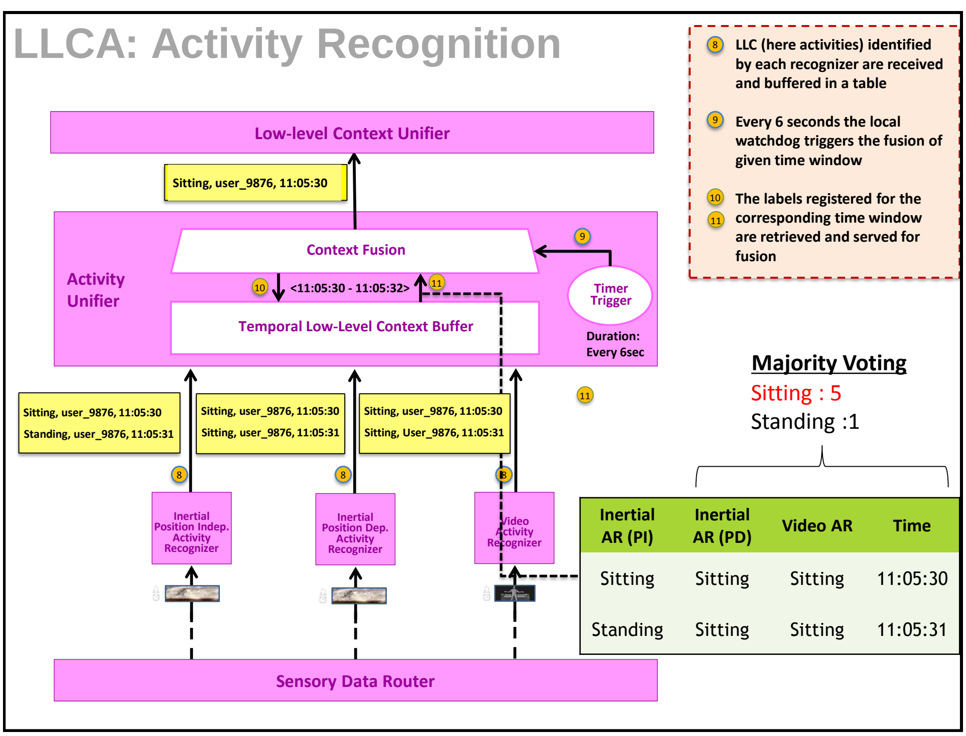
LLCA: Activity Recognition



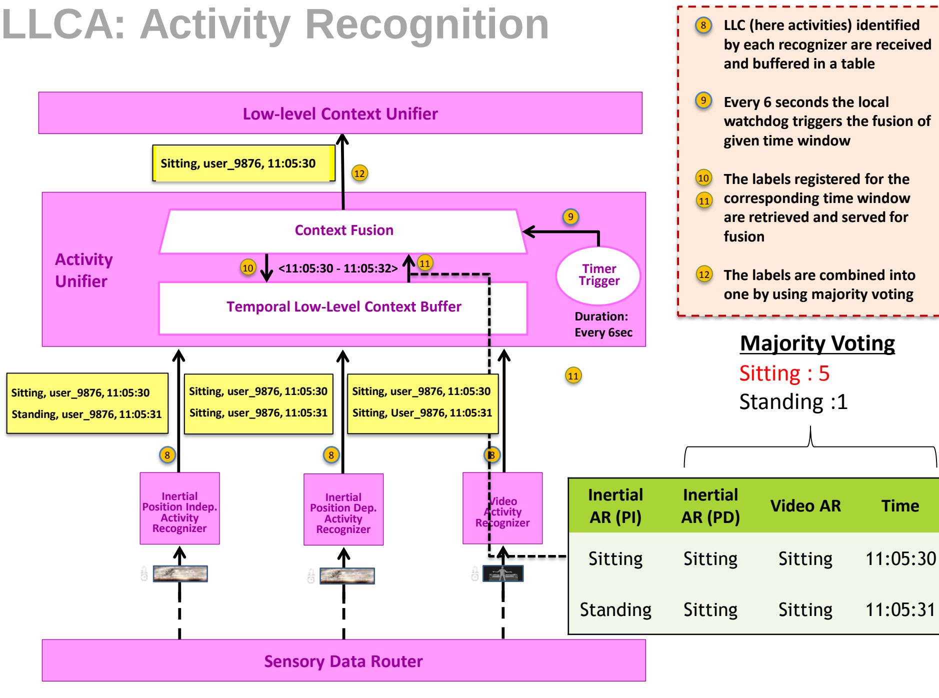

LLCA: Activity Recognition
Activity Unifier
1 2 3 The video stream data is separated from the metadata (userID, timestamp) for further processing (*this applies to all recognizers)
4 The person is detected and the joints-skeleton estimated
5 The video stream is split into short clips
6 Spatial distance and angle features are extracted
7 Probabilistic models are used after training for the identification of the emotion
8 The metadata are attached to the recognized activity label
Sitting, user_9876, 11:05:30
Sitting, User_9876, 11:05:31
Video Activity Recognizer
Output Adapter
Segmentation
Classification
Preprocessing
Feature Extraction
Input Adapter
Sensory Data Router


LLCA: Activity Recognition
8 LLC (here activities) identified by each recognizer are received and buffered in a table
Low-level Context Unifier
Sitting, user_9876, 11:05:30
Activity Unifier
Context Fusion
<11:05:30 - 11:05:32>
Timer Trigger
Temporal Low-Level Context Buffer
Duration: Every 6sec
Sitting, user_9876, 11:05:30
Standing, user_9876, 11:05:31
Sitting, user_9876, 11:05:30
Sitting, user_9876, 11:05:31
Sitting, user_9876, 11:05:30
Sitting, User_9876, 11:05:31
Inertial Position Indep. Activity Recognizer
Inertial Position Dep. Activity Recognizer
Video Activity Recognizer
Majority Voting
Sitting : 5
Standing :1
Inertial AR (PI) | Inertial AR (PD) | Video AR | Time
Sitting | Sitting | Sitting | 11:05:30
Standing | Sitting | Sitting | 11:05:31
Sensory Data Router


LLCA: Activity Recognition
8 LLC (here activities) identified by each recognizer are received and buffered in a table
9 Every 6 seconds the local watchdog triggers the fusion of given time window
Low-level Context Unifier
Sitting, user_9876, 11:05:30
Activity Unifier
Context Fusion
<11:05:30 - 11:05:32>
Temporal Low-Level Context Buffer
Timer Trigger
Duration: Every 6sec
Sitting, user_9876, 11:05:30
Standing, user_9876, 11:05:31
Sitting, user_9876, 11:05:30
Sitting, user_9876, 11:05:31
Sitting, user_9876, 11:05:30
Sitting, User_9876, 11:05:31
Inertial Position Indep. Activity Recognizer
Inertial Position Dep. Activity Recognizer
Video Activity Recognizer
Sensory Data Router
Majority Voting
Sitting : 5
Standing :1
Inertial AR (PI) | Inertial AR (PD) | Video AR | Time
Sitting | Sitting | Sitting | 11:05:30
Standing | Sitting | Sitting | 11:05:31

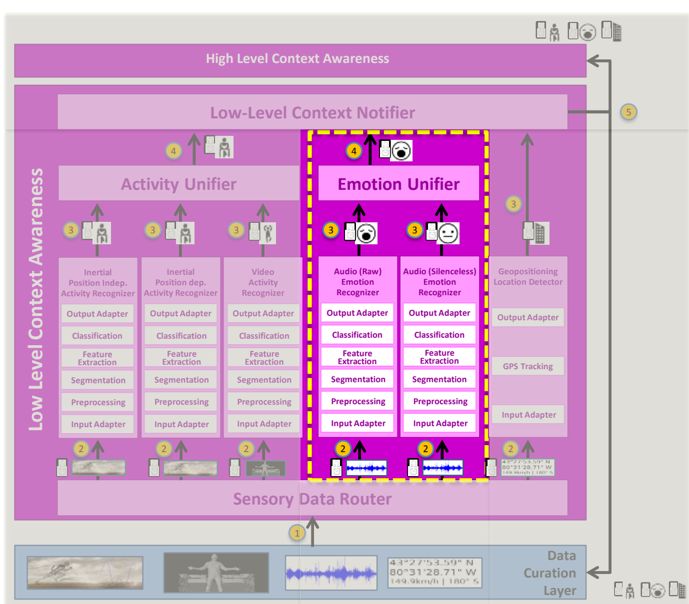

LLCA: Activity Recognition
8 LLC (here activities) identified by each recognizer are received and buffered in a table
9 Every 6 seconds the local watchdog triggers the fusion of given time window
10 11 The labels registered for the corresponding time window are retrieved and served for fusion
Low-level Context Unifier
Sitting, user_9876, 11:05:30
Activity Unifier
Context Fusion
<11:05:30 - 11:05:32>
Temporal Low-Level Context Buffer
Timer Trigger
Duration: Every 6sec
Sitting, user_9876, 11:05:30
Standing, user_9876, 11:05:31
Sitting, user_9876, 11:05:30
Sitting, user_9876, 11:05:31
Sitting, user_9876, 11:05:30
Sitting, User_9876, 11:05:31
Inertial Position Indep. Activity Recognizer
Inertial Position Dep. Activity Recognizer
Video Activity Recognizer
Sensory Data Router
Majority Voting
Sitting : 5
Standing :1
Inertial AR (PI) | Inertial AR (PD) | Video AR | Time
Sitting | Sitting | Sitting | 11:05:30
Standing | Sitting | Sitting | 11:05:31

LLCA: Activity Recognition

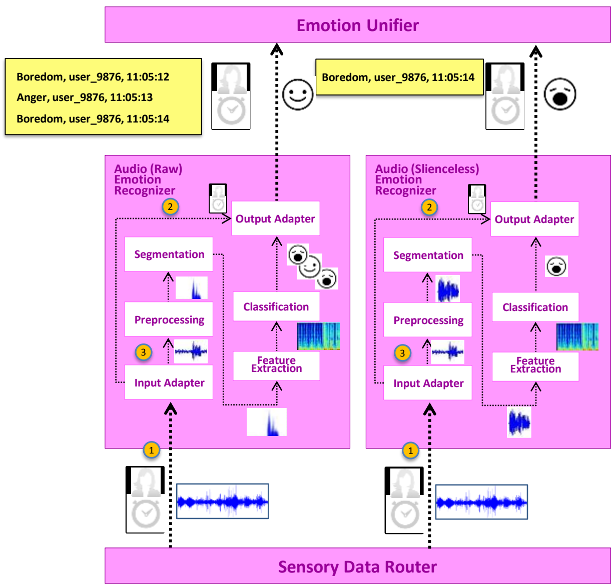


Inertial AR (PI)	Inertial AR (PD)	Video AR	Time
Sitting	Sitting	Sitting	11:05:30
Standing	Sitting	Sitting	11:05:31

LLCA: Emotion Recognition

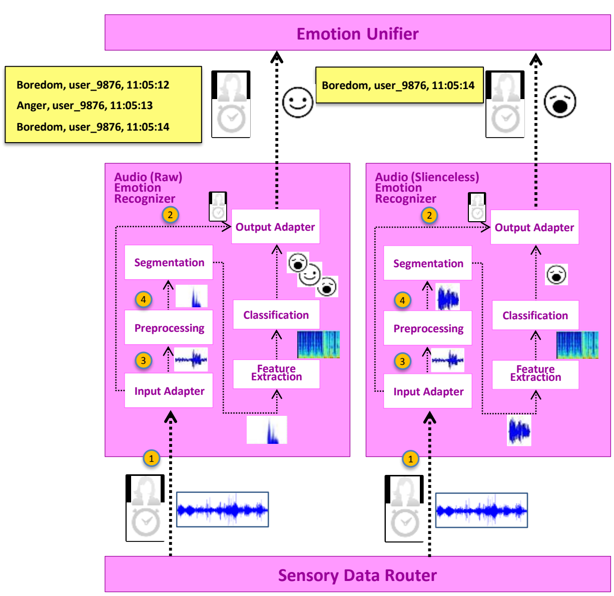


LLCA: Emotion Recognition



1 2 3 The raw audio data is separated from the metadata (userID, timestamp) for further processing (*this applies to all recognizers)

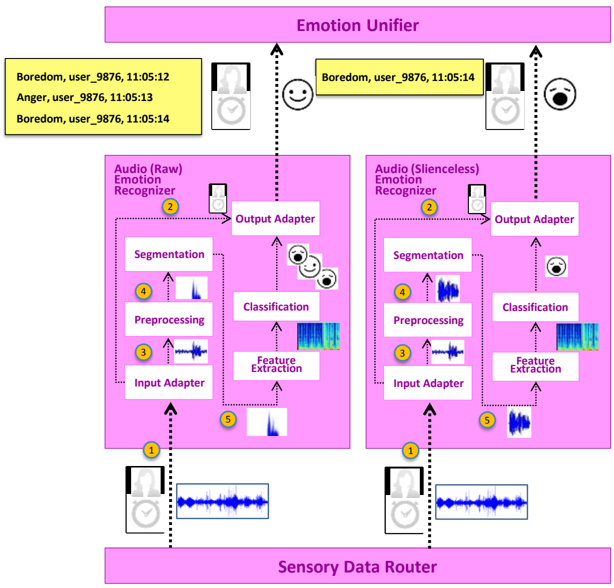
LLCA: Emotion Recognition



1 2 3 The raw audio data is separated from the metadata (userID, timestamp) for further processing (*this applies to all recognizers)

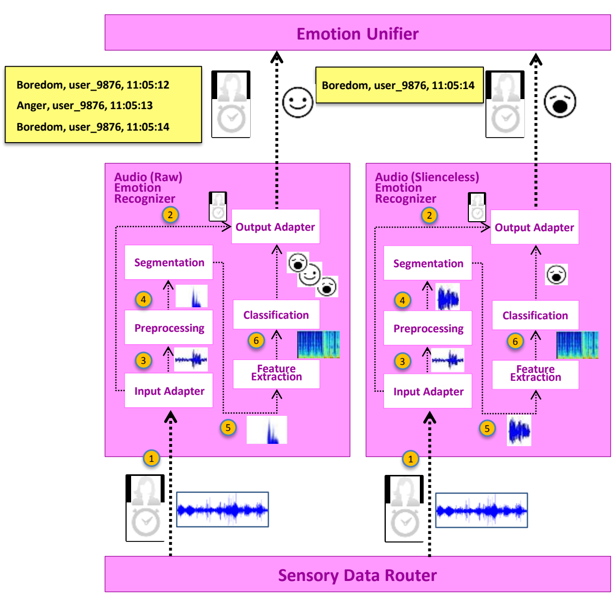
4 The raw audio signal is adapted through reducing the background noise and removing silences from the speech

LLCA: Emotion Recognition



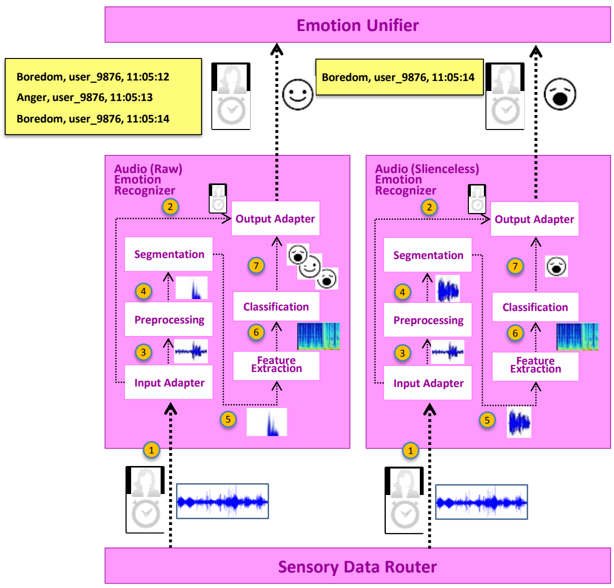
1. 2. 3. The raw audio data is separated from the metadata (userID, timestamp) for further processing (*this applies to all recognizers)
4. The raw audio signal is adapted through reducing the background noise and removing silences from the speech
5. Each speech segment is partitioned into smaller segments of data

LLCA: Emotion Recognition



1. 2. 3. The raw audio data is separated from the metadata (userID, timestamp) for further processing (*this applies to all recognizers)
4. The raw audio signal is adapted through reducing the background noise and removing silences from the speech
5. Each speech segment is partitioned into smaller segments of data
6. Low-level descriptors are extracted from each segment, including cepstral coefficients and statistical values

LLCA: Emotion Recognition



1 2 3 The raw audio data is separated from the metadata (userID, timestamp) for further processing (*this applies to all recognizers)

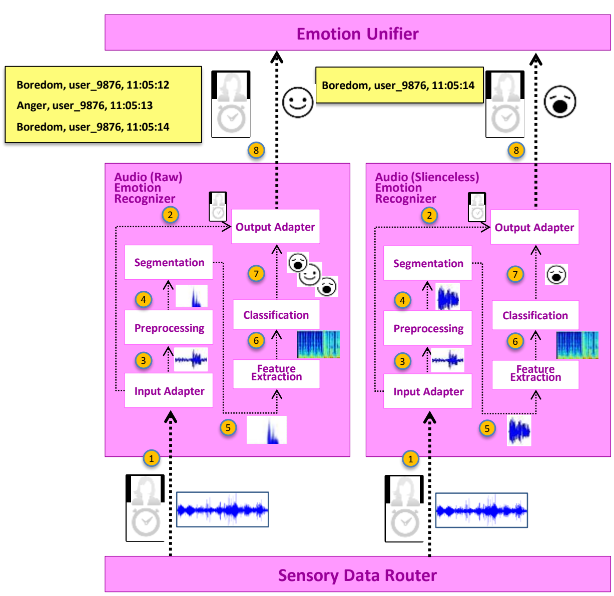
4 The raw audio signal is adapted through reducing the background noise and removing silences from the speech

5 Each speech segment is partitioned into smaller segments of data

6 Low-level descriptors are extracted from each segment, including cepstral coefficients and statistical values

7 Probabilistic models are used after training for the identification of the emotion

LLCA: Emotion Recognition



1 2 3 The raw audio data is separated from the metadata (userID, timestamp) for further processing (*this applies to all recognizers)

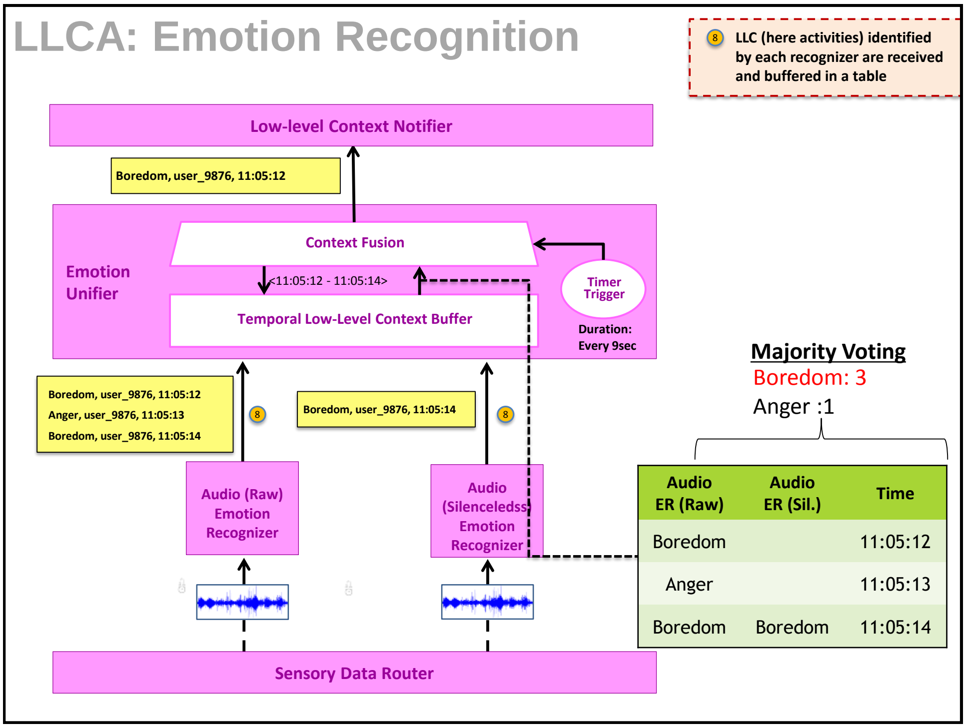
4 The raw audio signal is adapted through reducing the background noise and removing silences from the speech

5 Each speech segment is partitioned into smaller segments of data

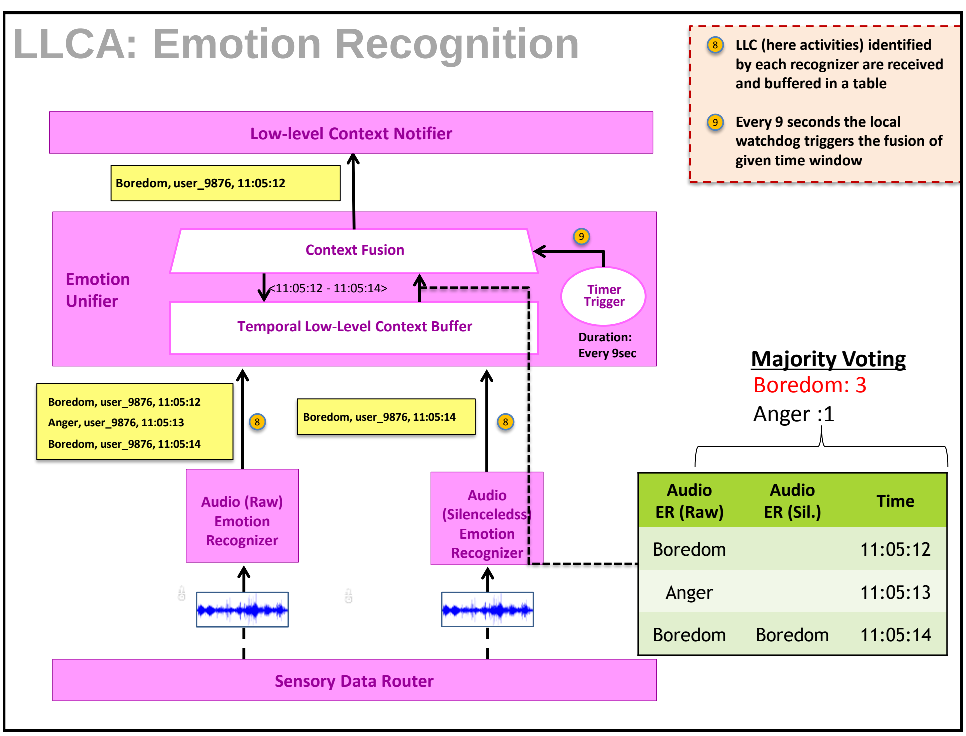
6 Low-level descriptors are extracted from each segment, including cepstral coefficients and statistical values

7 Probabilistic models are used after training for the identification of the emotion

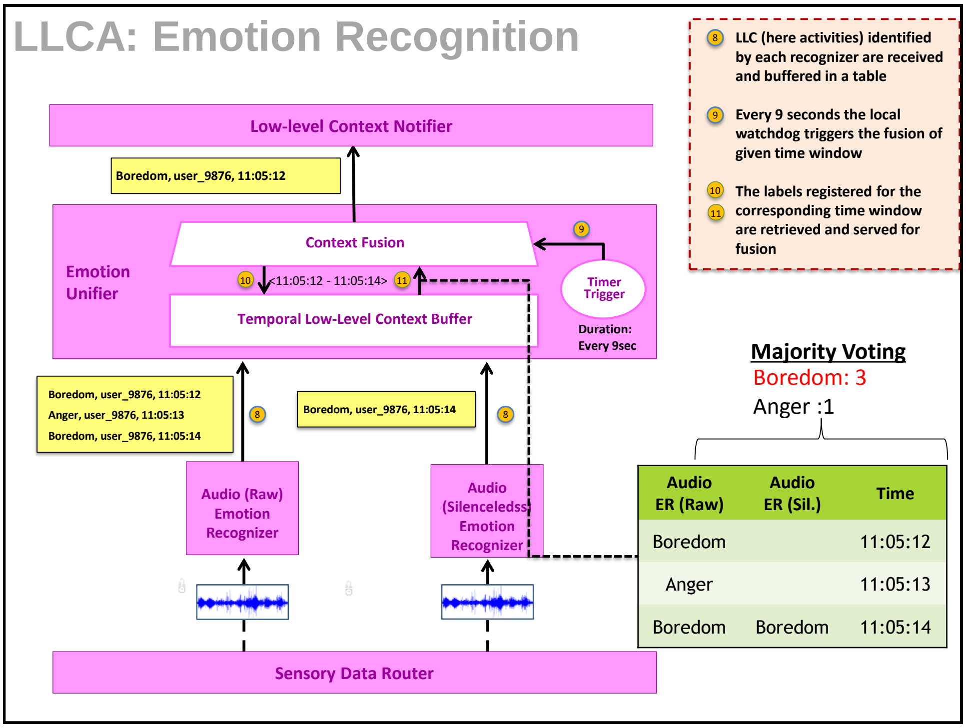
8 The metadata are attached to the recognized emotion label

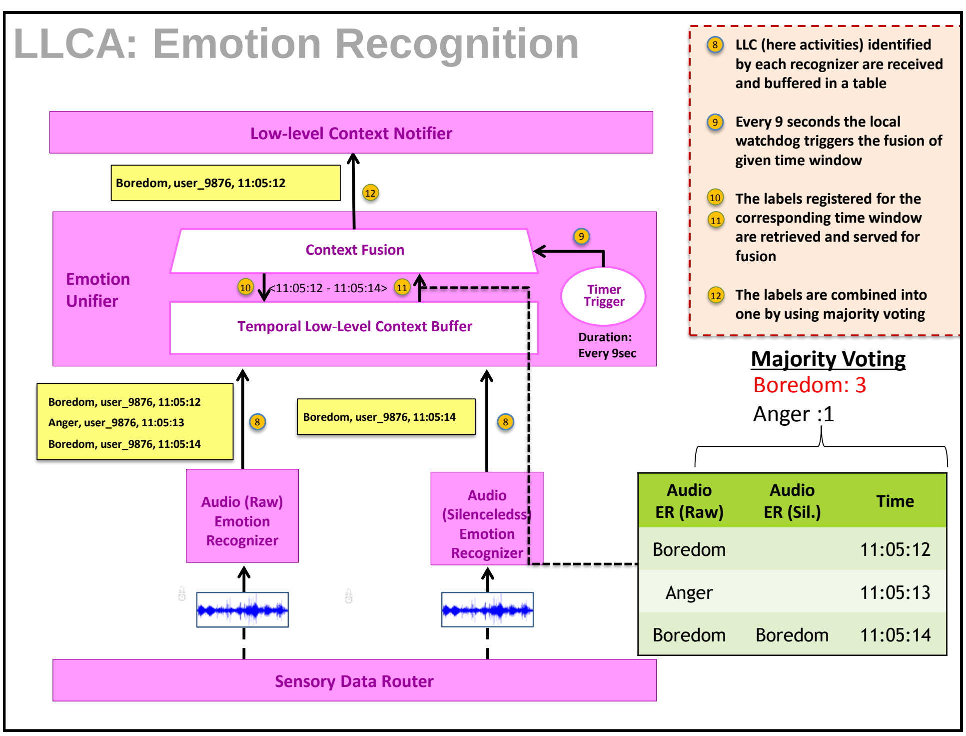


LLCA: Emotion Recognition
8 LLC (here activities) identified by each recognizer are received and buffered in a table
Low-level Context Notifier
Boredom, user_9876, 11:05:12
Context Fusion
Emotion Unifier
<11:05:12 - 11:05:14>
Timer Trigger
Temporal Low-Level Context Buffer
Duration: Every 9sec
Majority Voting
Boredom: 3
Anger :1
Boredom, user_9876, 11:05:12
Anger, user_9876, 11:05:13
Boredom, user_9876, 11:05:14
8
Boredom, user_9876, 11:05:14
8
Audio (Raw) Emotion Recognizer
Audio (Silenceledss) Emotion Recognizer
Audio ER (Raw)
Audio ER (Sil.)
Time
Boredom
11:05:12
Anger
11:05:13
Boredom
Boredom
11:05:14
Sensory Data Router

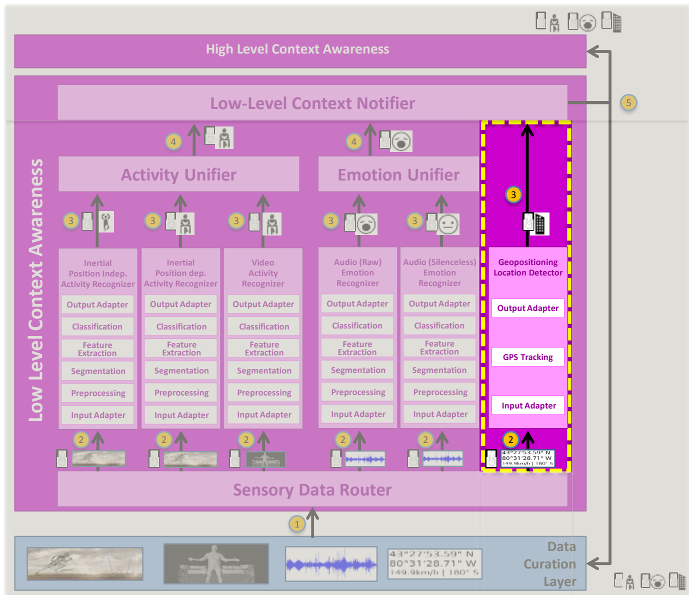


LLCA: Emotion Recognition
8 LLC (here activities) identified by each recognizer are received and buffered in a table
9 Every 9 seconds the local watchdog triggers the fusion of given time window
Low-level Context Notifier
Boredom, user_9876, 11:05:12
Context Fusion
9
Emotion Unifier
<11:05:12 - 11:05:14>
Timer Trigger
Temporal Low-Level Context Buffer
Duration: Every 9sec
Majority Voting
Boredom: 3
Anger :1
Boredom, user_9876, 11:05:12
Anger, user_9876, 11:05:13
Boredom, user_9876, 11:05:14
8
Boredom, user_9876, 11:05:14
8
Audio (Raw) Emotion Recognizer
Audio (Silenceledss) Emotion Recognizer
Audio ER (Raw)
Audio ER (Sil.)
Time
Boredom
11:05:12
Anger
11:05:13
Boredom
Boredom
11:05:14
Sensory Data Router

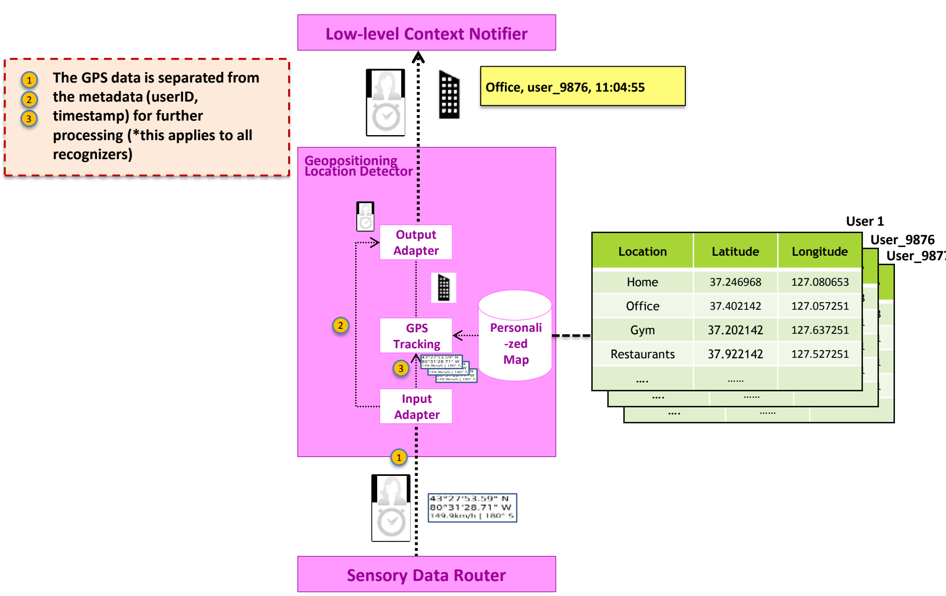

LLCA: Emotion Recognition
8 LLC (here activities) identified by each recognizer are received and buffered in a table
9 Every 9 seconds the local watchdog triggers the fusion of given time window
10 11 The labels registered for the corresponding time window are retrieved and served for fusion
Low-level Context Notifier
Boredom, user_9876, 11:05:12
Context Fusion
9
Emotion Unifier
10 <11:05:12 - 11:05:14> 11
Timer Trigger
Temporal Low-Level Context Buffer
Duration: Every 9sec
Majority Voting
Boredom: 3
Anger :1
Boredom, user_9876, 11:05:12
Anger, user_9876, 11:05:13
Boredom, user_9876, 11:05:14
8
Boredom, user_9876, 11:05:14
8
Audio (Raw) Emotion Recognizer
Audio (Silenceledss) Emotion Recognizer
Audio ER (Raw) | Audio ER (Sil.) | Time
Boredom | | 11:05:12
Anger | | 11:05:13
Boredom | Boredom | 11:05:14
Sensory Data Router

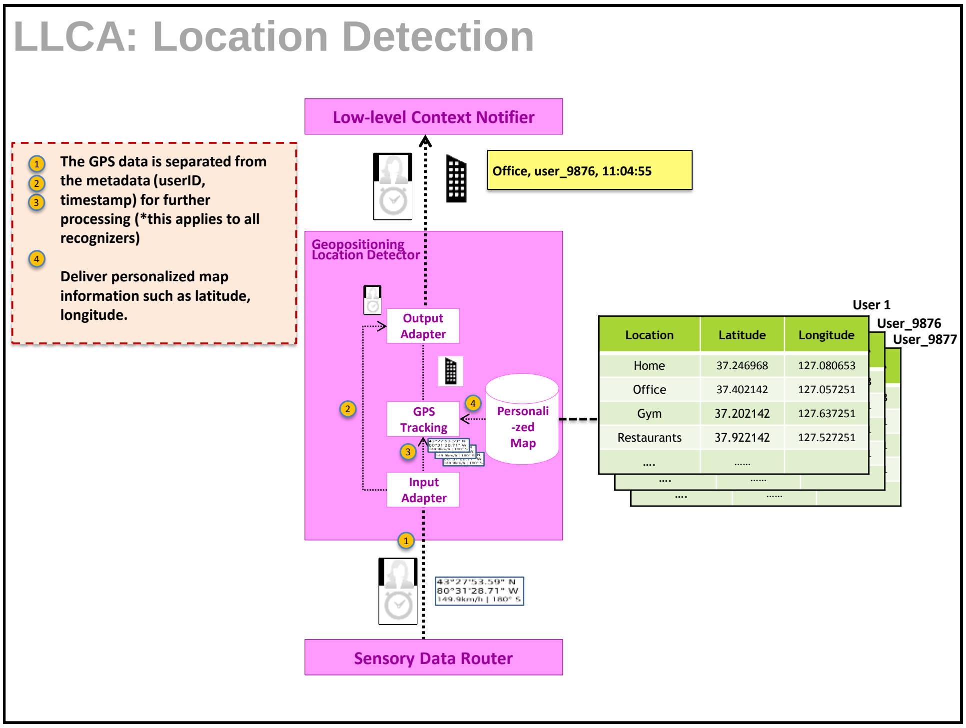

LLCA: Emotion Recognition
8 LLC (here activities) identified by each recognizer are received and buffered in a table
9 Every 9 seconds the local watchdog triggers the fusion of given time window
10 11 The labels registered for the corresponding time window are retrieved and served for fusion
12 The labels are combined into one by using majority voting
Low-level Context Notifier
Boredom, user_9876, 11:05:12
12
Context Fusion
9
Emotion Unifier
10 <11:05:12 - 11:05:14> 11
Timer Trigger
Temporal Low-Level Context Buffer
Duration: Every 9sec
Majority Voting
Boredom: 3
Anger :1
Boredom, user_9876, 11:05:12
Anger, user_9876, 11:05:13
Boredom, user_9876, 11:05:14
8
Boredom, user_9876, 11:05:14
8
Audio (Raw) Emotion Recognizer
Audio (Silenceledss) Emotion Recognizer
Audio ER (Raw) | Audio ER (Sil.) | Time
Boredom | | 11:05:12
Anger | | 11:05:13
Boredom | Boredom | 11:05:14
Sensory Data Router

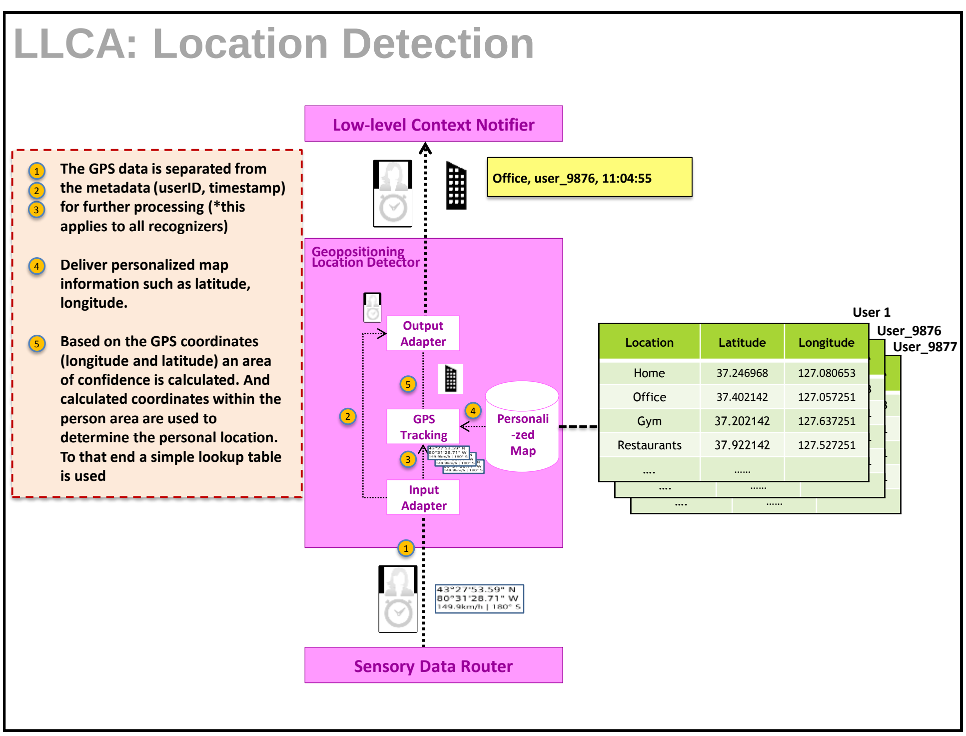
LLCA: Location Detection



LLCA: Location Detection




LLCA: Location Detection
Low-level Context Notifier
Office, user_9876, 11:04:55
1 2 3 The GPS data is separated from the metadata (userID, timestamp) for further processing (*this applies to all recognizers)
4 Deliver personalized map information such as latitude, longitude.
Geopositioning Location Detector
Output Adapter
GPS Tracking
Personali-zed Map
Input Adapter
43°27'53.59" N
80°31'28.71" W
149.9km/h | 180° S
Sensory Data Router
User 1
User_9876
User_9877
Location | Latitude | Longitude
Home | 37.246968 | 127.080653
Office | 37.402142 | 127.057251
Gym | 37.202142 | 127.637251
Restaurants | 37.922142 | 127.527251
.... | ......


LLCA: Location Detection
Low-level Context Notifier
Office, user_9876, 11:04:55
1 2 3 The GPS data is separated from the metadata (userID, timestamp) for further processing (*this applies to all recognizers)
4 Deliver personalized map information such as latitude, longitude.
5 Based on the GPS coordinates (longitude and latitude) an area of confidence is calculated. And calculated coordinates within the person area are used to determine the personal location. To that end a simple lookup table is used
Geopositioning Location Detector
Output Adapter
GPS Tracking
Personali-zed Map
Input Adapter
43°27'53.59" N
80°31'28.71" W
149.9km/h | 180° S
Sensory Data Router
User 1
User_9876
User_9877
Location | Latitude | Longitude
Home | 37.246968 | 127.080653
Office | 37.402142 | 127.057251
Gym | 37.202142 | 127.637251
Restaurants | 37.922142 | 127.527251
.... | ......

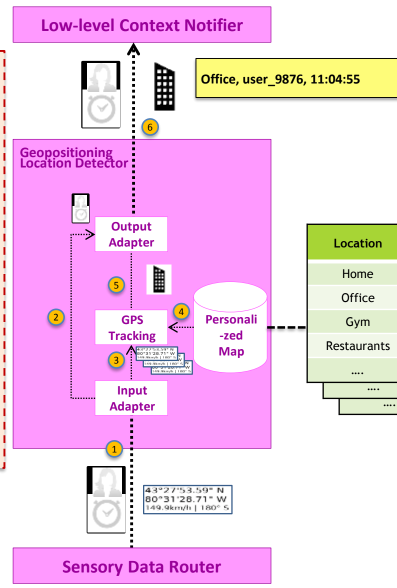
LLCA: Location Detection

1. The GPS data is separated from
2. the metadata (userID, timestamp)
3. for further processing (*this applies to all recognizers)

4. Deliver personalized map information such as latitude, longitude.

5. Based on the GPS coordinates (longitude and latitude) an area of confidence is calculated. And calculated coordinates within the person area are used to determine the personal location. To that end a simple lookup table is used

6. The metadata are attached to the recognized emotion label

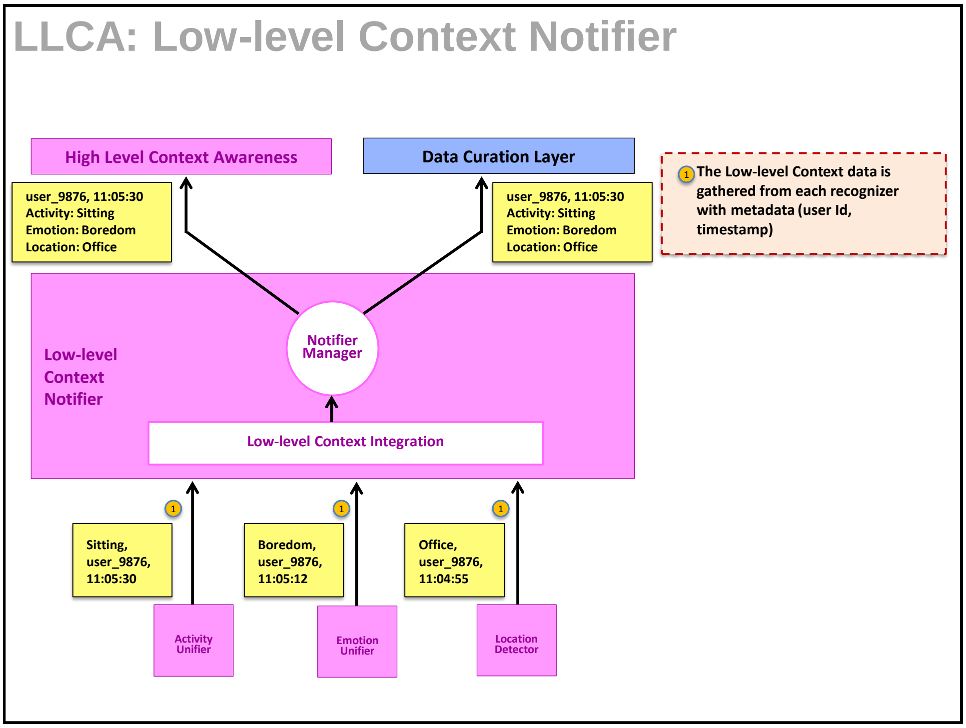


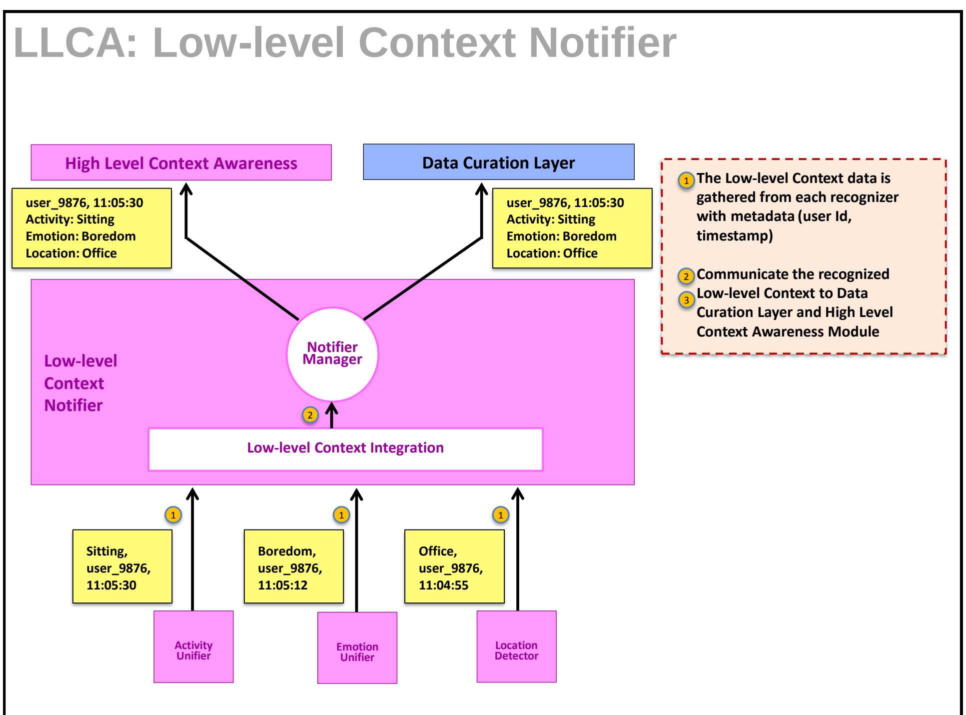
User 1
User_9876
User_9877

Location	Latitude	Longitude
Home	37.246968	127.080653
Office	37.402142	127.057251
Gym	37.202142	127.637251
Restaurants	37.922142	127.527251
....	......	

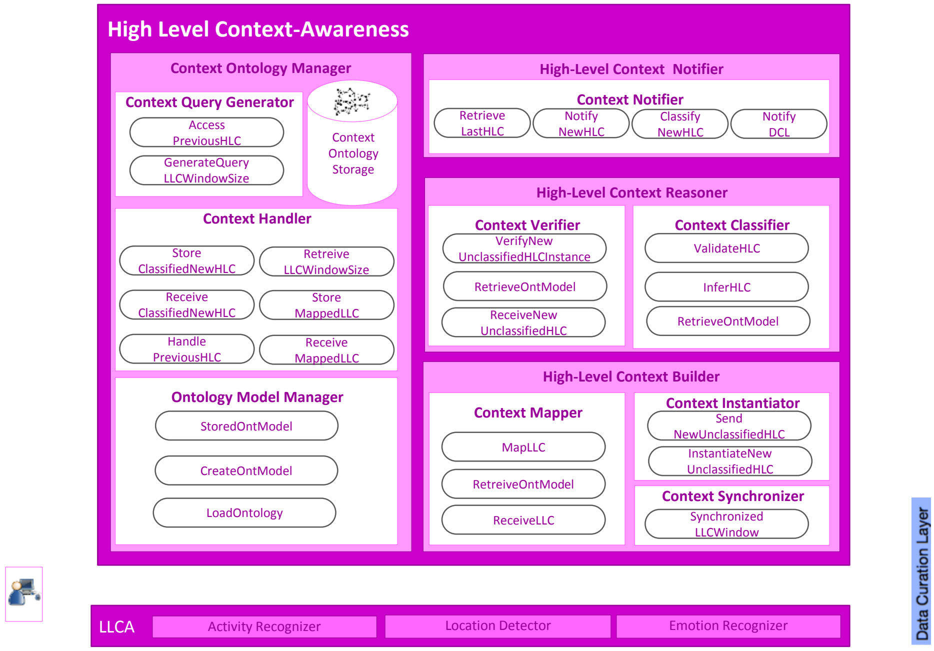
Scenario 2: LLC Notification to DCL & HLCA

Desc. : Low-level Context is **notified** to High Level Context Awareness and **persisted** at Data Curation Layer.


LLCA: Low-level Context Notifier
High Level Context Awareness
Data Curation Layer
user_9876, 11:05:30
Activity: Sitting
Emotion: Boredom
Location: Office
user_9876, 11:05:30
Activity: Sitting
Emotion: Boredom
Location: Office
1 The Low-level Context data is gathered from each recognizer with metadata (user Id, timestamp)
Low-level Context Notifier
Notifier Manager
Low-level Context Integration
1
1
1
Sitting, user_9876, 11:05:30
Boredom, user_9876, 11:05:12
Office, user_9876, 11:04:55
Activity Unifier
Emotion Unifier
Location Detector


LLCA: Low-level Context Notifier
High Level Context Awareness
Data Curation Layer
user_9876, 11:05:30
Activity: Sitting
Emotion: Boredom
Location: Office
user_9876, 11:05:30
Activity: Sitting
Emotion: Boredom
Location: Office
1 The Low-level Context data is gathered from each recognizer with metadata (user Id, timestamp)
2 3 Communicate the recognized Low-level Context to Data Curation Layer and High Level Context Awareness Module
Low-level Context Notifier
Notifier Manager
2
Low-level Context Integration
1
1
1
Sitting, user_9876, 11:05:30
Boredom, user_9876, 11:05:12
Office, user_9876, 11:04:55
Activity Unifier
Emotion Unifier
Location Detector

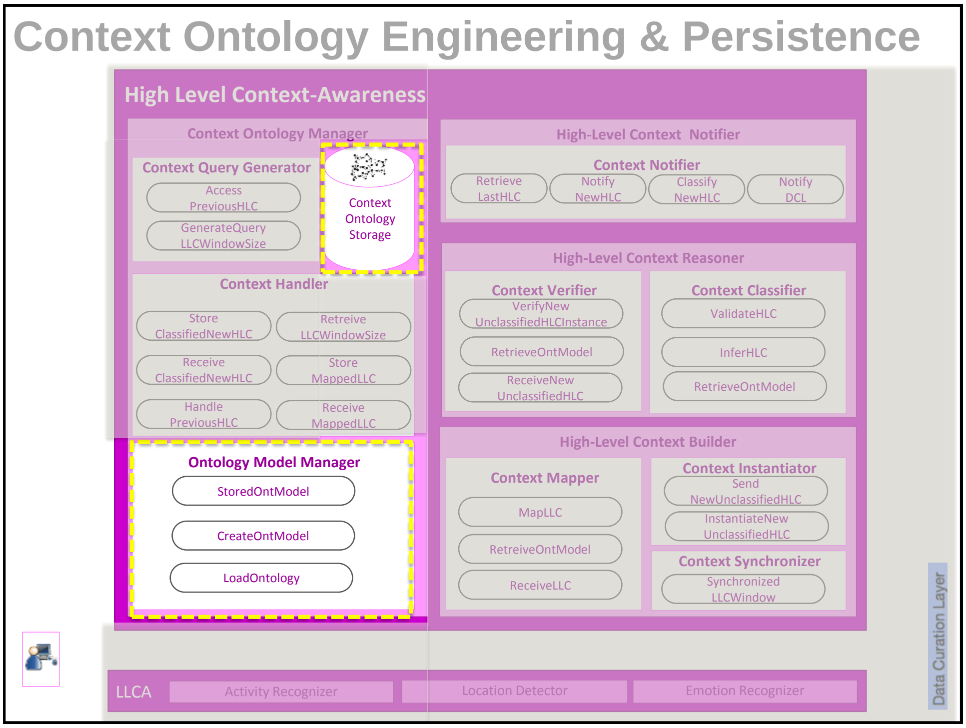
Overall HLCA

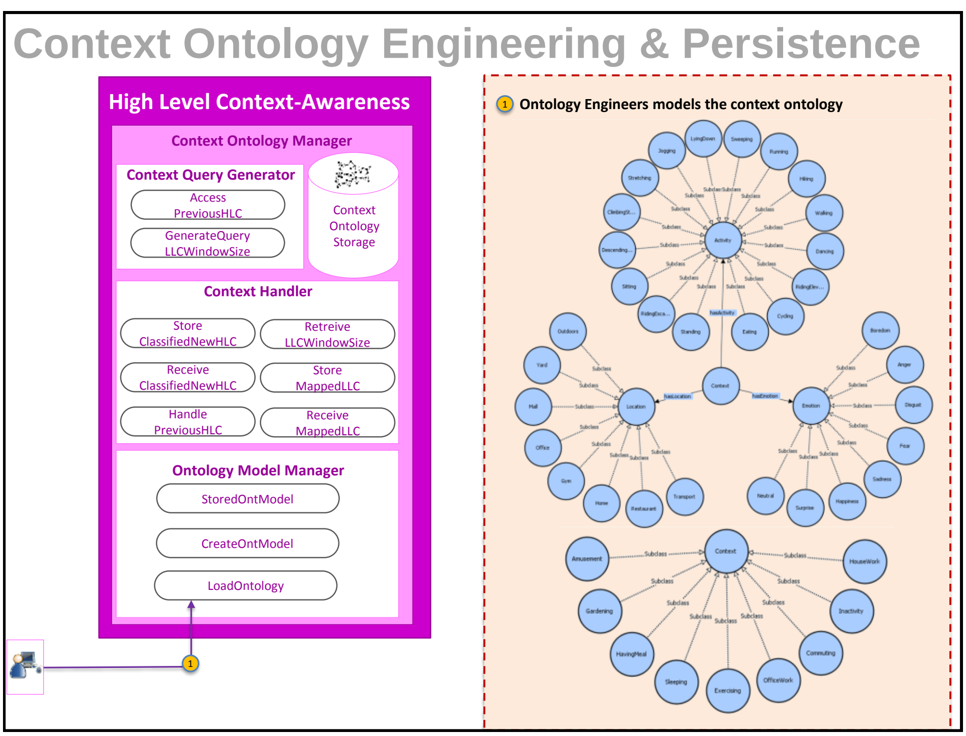


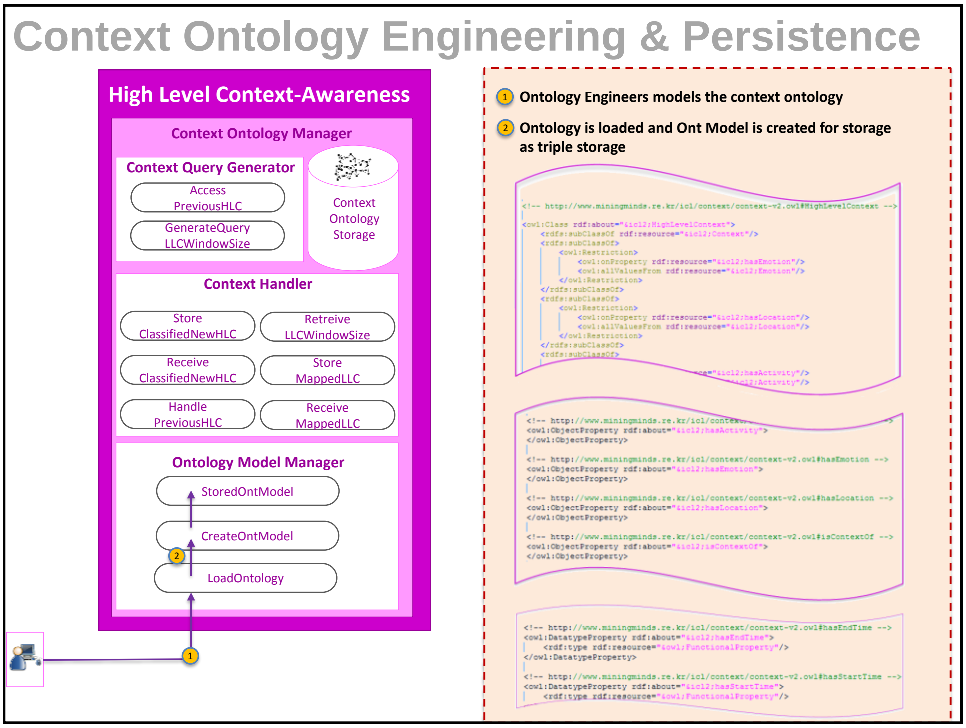
Scenario 3: Unclassified HLC from LLC

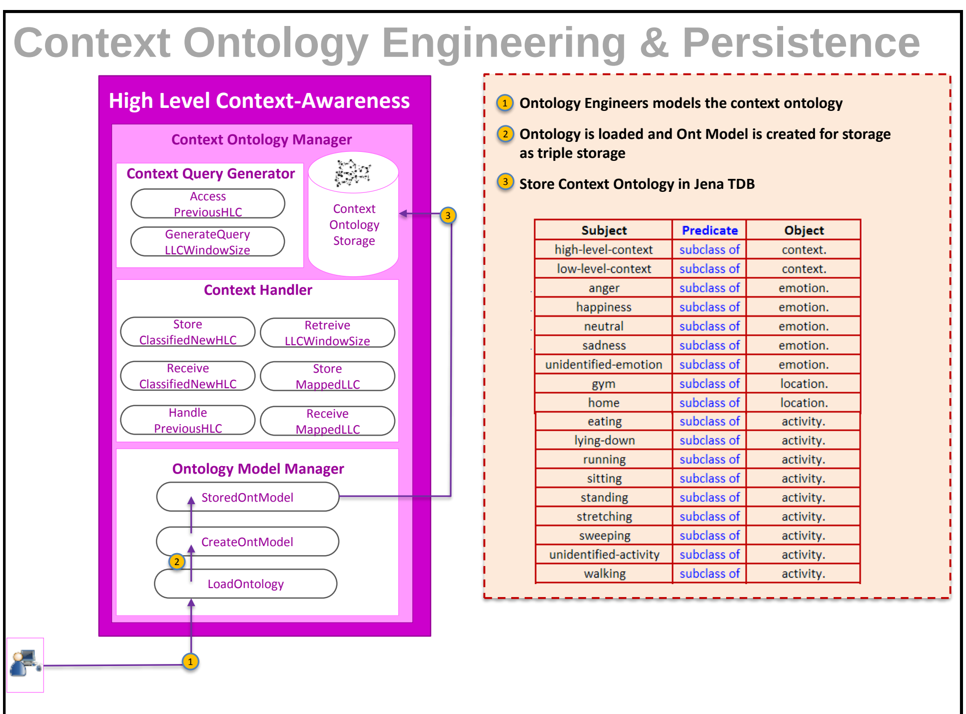
Desc. : Low level activities and High level activities are **modeled** in ontology and **persisted** in triple storage. Low level activities are **mapped**, then **synchronized** in a window size for HLC reasoning instantiation.

112

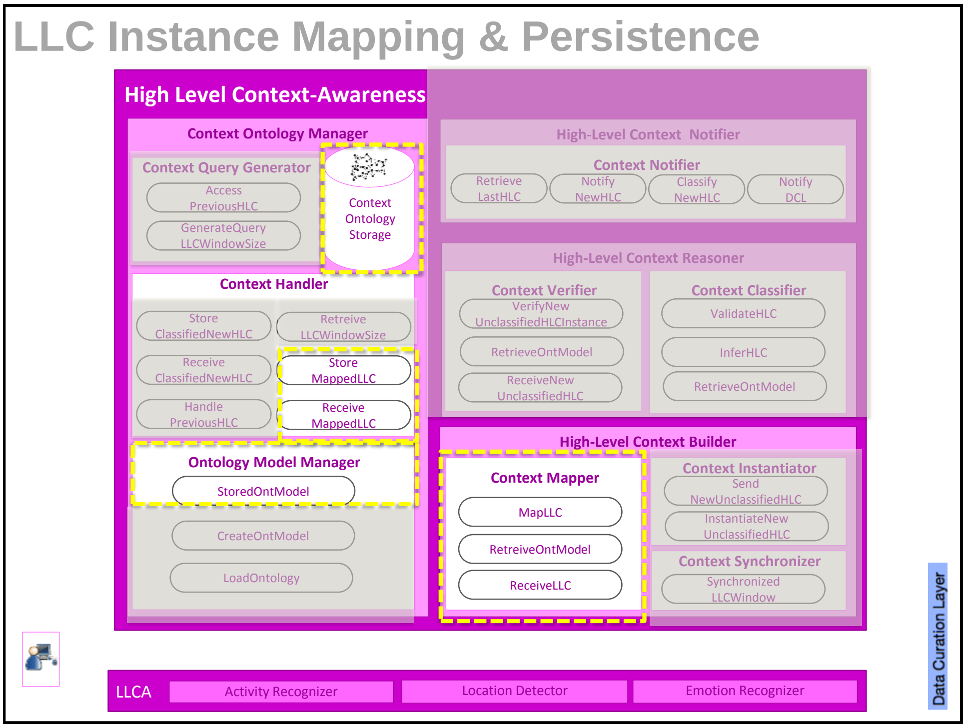

Context Ontology Engineering & Persistence
High Level Context-Awareness
Context Ontology Manager
Context Query Generator
Access PreviousHLC
GenerateQuery LLCWindowSize
Context Ontology Storage
Context Handler
Store ClassifiedNewHLC
Retreive LLCWindowSize
Receive ClassifiedNewHLC
Store MappedLLC
Handle PreviousHLC
Receive MappedLLC
Ontology Model Manager
StoredOntModel
CreateOntModel
LoadOntology
High-Level Context Notifier
Context Notifier
Retrieve LastHLC
Notify NewHLC
Classify NewHLC
Notify DCL
High-Level Context Reasoner
Context Verifier
VerifyNew UnclassifiedHLCInstance
RetrieveOntModel
ReceiveNew UnclassifiedHLC
Context Classifier
ValidateHLC
InferHLC
RetrieveOntModel
High-Level Context Builder
Context Mapper
MapLLC
RetreiveOntModel
ReceiveLLC
Context Instantiator
Send NewUnclassifiedHLC
InstantiateNew UnclassifiedHLC
Context Synchronizer
Synchronized LLCWindow
Data Curation Layer
LLCA
Activity Recognizer
Location Detector
Emotion Recognizer


Context Ontology Engineering & Persistence
High Level Context-Awareness
Context Ontology Manager
Context Query Generator
Access PreviousHLC
GenerateQuery LLCWindowSize
Context Ontology Storage
Context Handler
Store ClassifiedNewHLC
Retreive LLCWindowSize
Receive ClassifiedNewHLC
Store MappedLLC
Handle PreviousHLC
Receive MappedLLC
Ontology Model Manager
StoredOntModel
CreateOntModel
LoadOntology
1
1 Ontology Engineers models the context ontology

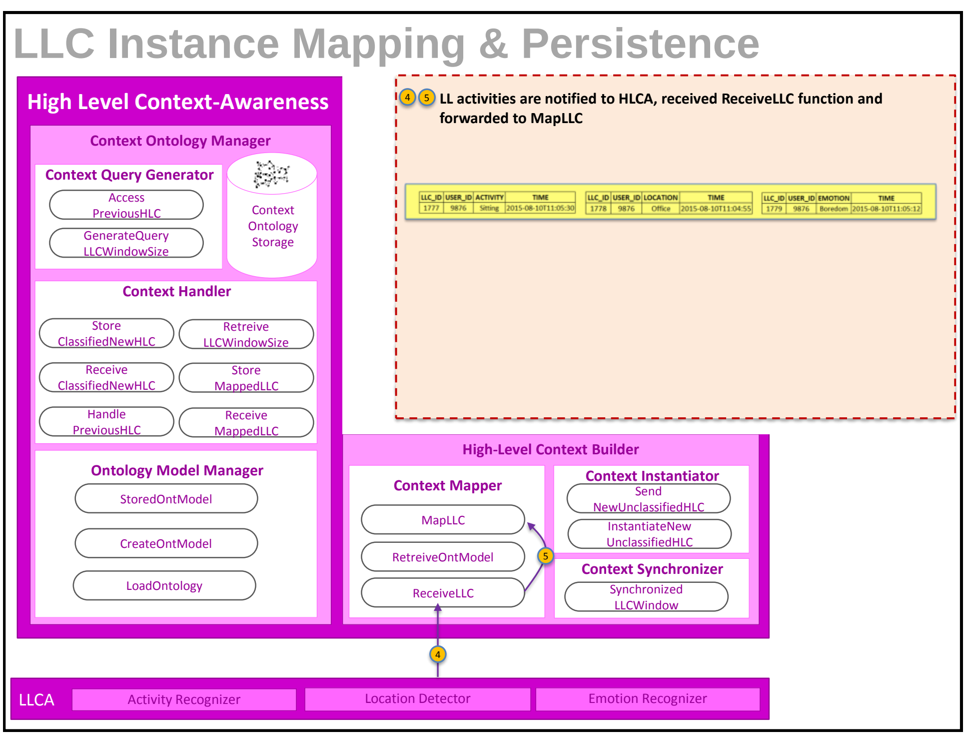




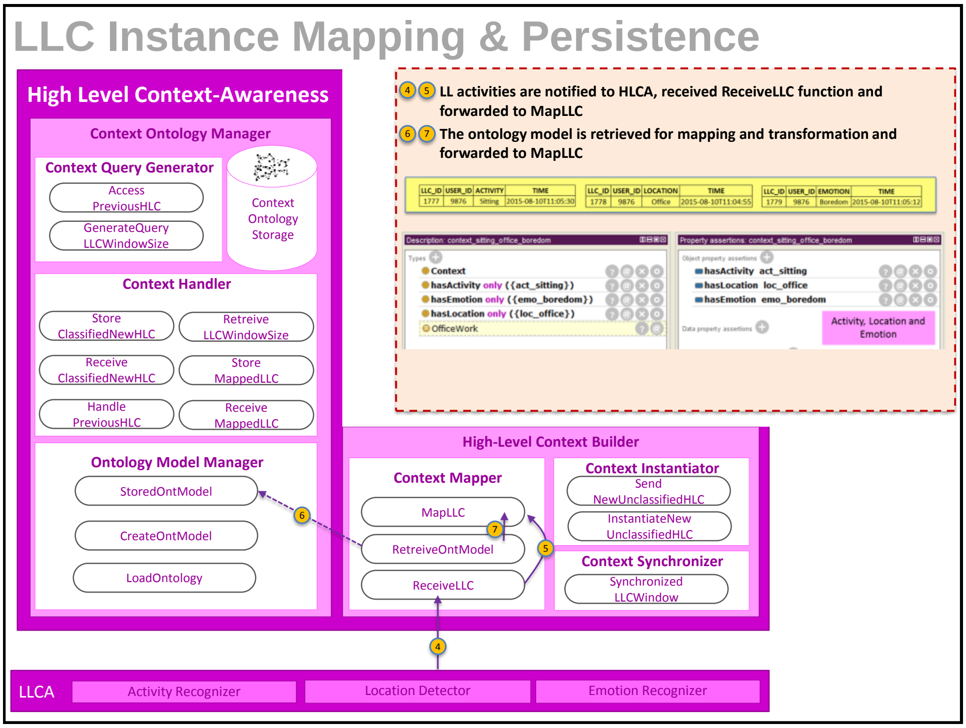
Subject	Predicate	Object
high-level-context	subclass of	context.
low-level-context	subclass of	context.
anger	subclass of	emotion.
happiness	subclass of	emotion.
neutral	subclass of	emotion.
sadness	subclass of	emotion.
unidentified-emotion	subclass of	emotion.
gym	subclass of	location.
home	subclass of	location.
eating	subclass of	activity.
lying-down	subclass of	activity.
running	subclass of	activity.
sitting	subclass of	activity.
standing	subclass of	activity.
stretching	subclass of	activity.
sweeping	subclass of	activity.
unidentified-activity	subclass of	activity.
walking	subclass of	activity.

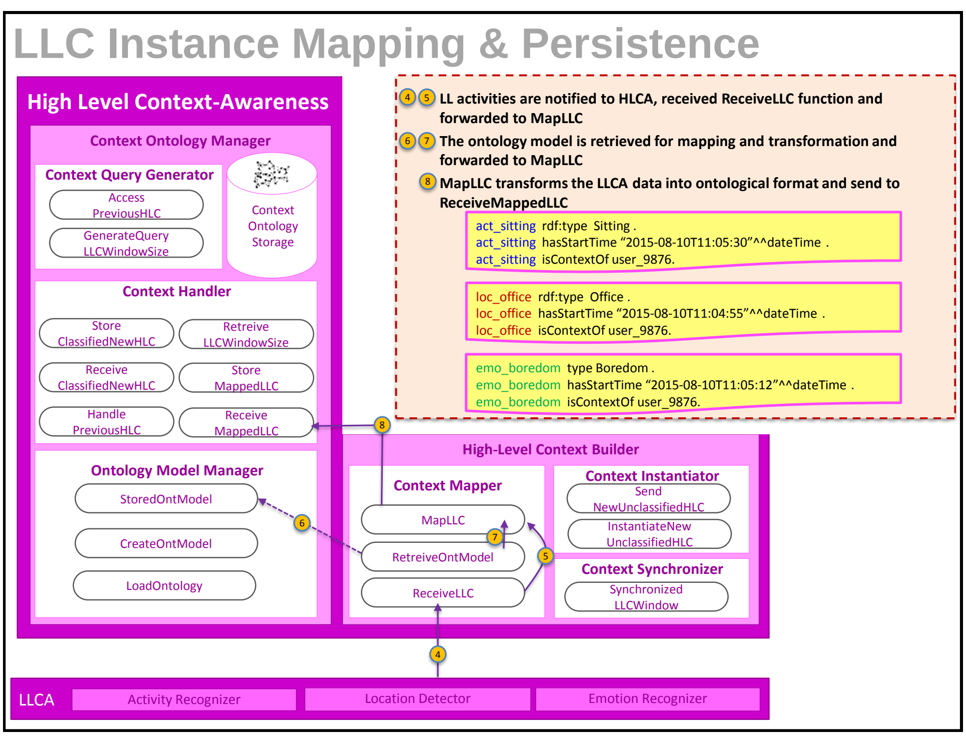


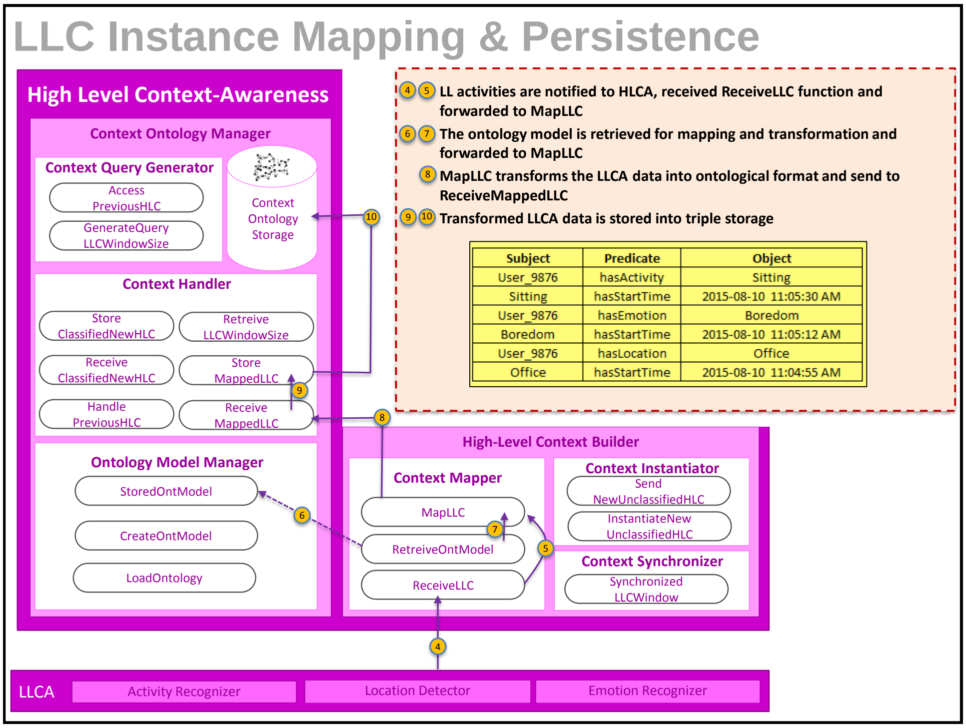
LLC Instance Mapping & Persistence
High Level Context-Awareness
Context Ontology Manager
Context Query Generator
Access PreviousHLC
GenerateQuery LLCWindowSize
Context Ontology Storage
Context Handler
Store ClassifiedNewHLC
Retreive LLCWindowSize
Receive ClassifiedNewHLC
Store MappedLLC
Handle PreviousHLC
Receive MappedLLC
Ontology Model Manager
StoredOntModel
CreateOntModel
LoadOntology
High-Level Context Notifier
Context Notifier
Retrieve LastHLC
Notify NewHLC
Classify NewHLC
Notify DCL
High-Level Context Reasoner
Context Verifier
VerifyNew UnclassifiedHLCInstance
RetrieveOntModel
ReceiveNew UnclassifiedHLC
Context Classifier
ValidateHLC
InferHLC
RetrieveOntModel
High-Level Context Builder
Context Mapper
MapLLC
RetreiveOntModel
ReceiveLLC
Context Instantiator
Send NewUnclassifiedHLC
InstantiateNew UnclassifiedHLC
Context Synchronizer
Synchronized LLCWindow
LLCA
Activity Recognizer
Location Detector
Emotion Recognizer
Data Curation Layer

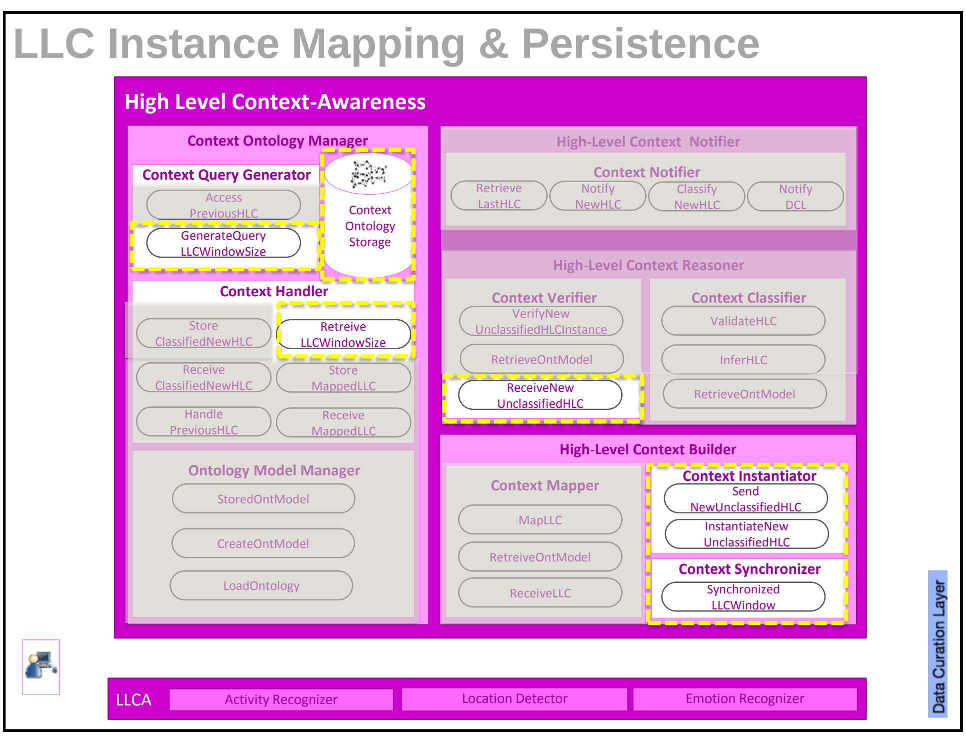


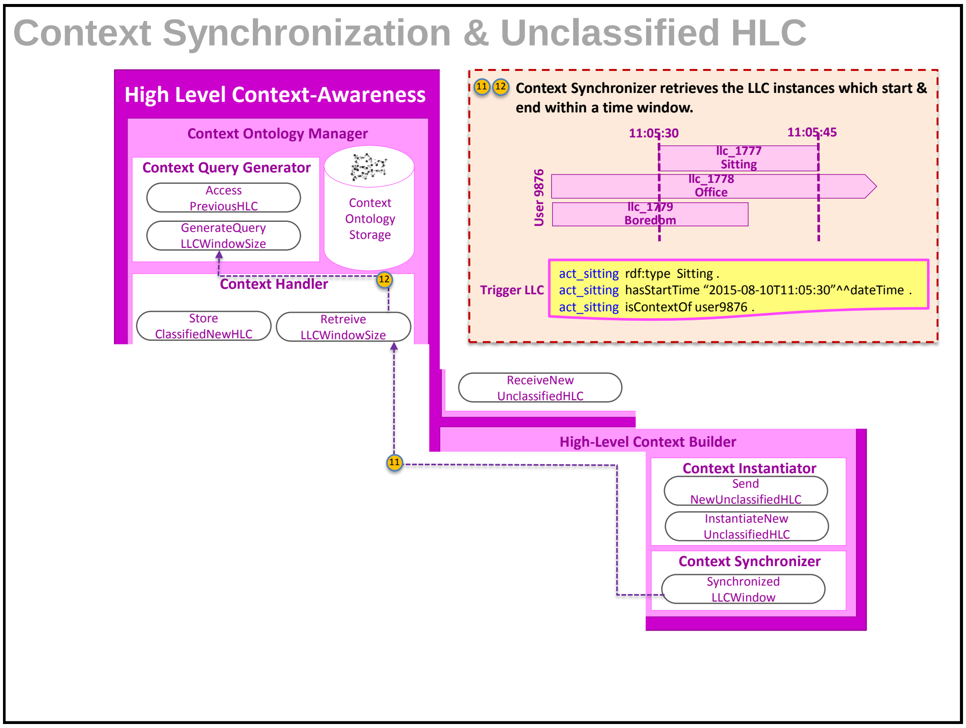
LLC Instance Mapping & Persistence
High Level Context-Awareness
Context Ontology Manager
Context Query Generator
Access PreviousHLC
GenerateQuery LLCWindowSize
Context Ontology Storage
Context Handler
Store ClassifiedNewHLC
Retreive LLCWindowSize
Receive ClassifiedNewHLC
Store MappedLLC
Handle PreviousHLC
Receive MappedLLC
Ontology Model Manager
StoredOntModel
CreateOntModel
LoadOntology
4 5 LL activities are notified to HLCA, received ReceiveLLC function and forwarded to MapLLC
LLC_ID USER_ID ACTIVITY TIME
1777 9876 Sitting 2015-08-10T11:05:30
LLC_ID USER_ID LOCATION TIME
1778 9876 Office 2015-08-10T11:04:55
LLC_ID USER_ID EMOTION TIME
1779 9876 Boredom 2015-08-10T11:05:12
High-Level Context Builder
Context Mapper
MapLLC
RetreiveOntModel
ReceiveLLC
Context Instantiator
Send NewUnclassifiedHLC
InstantiateNew UnclassifiedHLC
Context Synchronizer
Synchronized LLCWindow
5
4
LLCA
Activity Recognizer
Location Detector
Emotion Recognizer

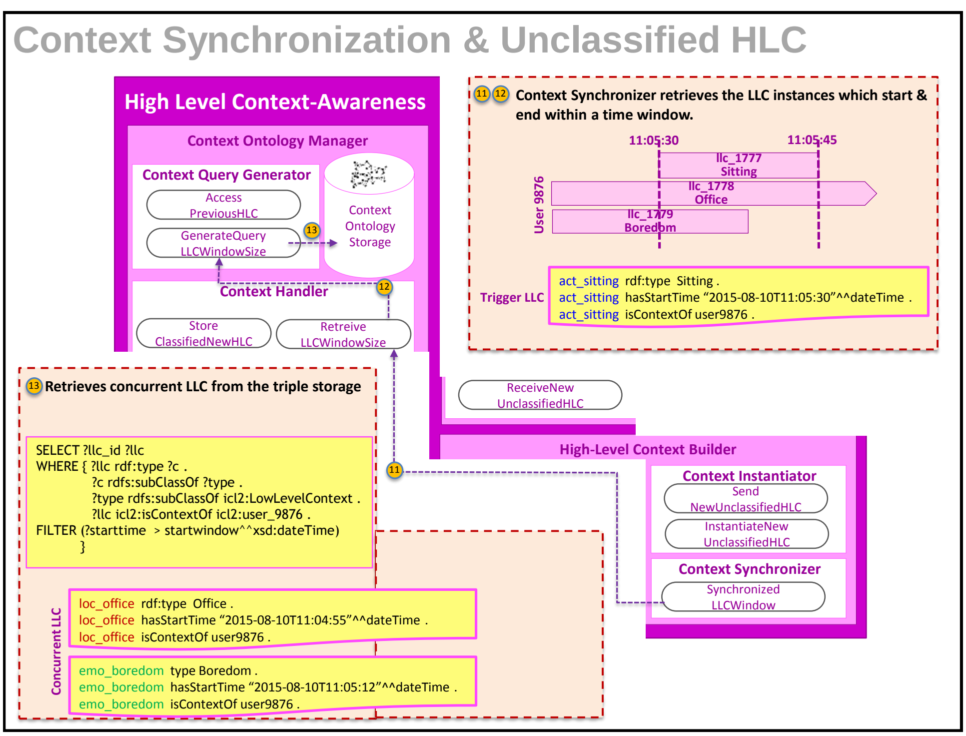

LLC Instance Mapping & Persistence
High Level Context-Awareness
Context Ontology Manager
Context Query Generator
Access PreviousHLC
GenerateQuery LLCWindowSize
Context Ontology Storage
Context Handler
Store ClassifiedNewHLC
Retreive LLCWindowSize
Receive ClassifiedNewHLC
Store MappedLLC
Handle PreviousHLC
Receive MappedLLC
Ontology Model Manager
StoredOntModel
CreateOntModel
LoadOntology
4 5 LL activities are notified to HLCA, received ReceiveLLC function and forwarded to MapLLC
6 7 The ontology model is retrieved for mapping and transformation and forwarded to MapLLC
LLC_ID USER_ID ACTIVITY TIME
1777 9876 Sitting 2015-08-10T11:05:30
LLC_ID USER_ID LOCATION TIME
1778 9876 Office 2015-08-10T11:04:55
LLC_ID USER_ID EMOTION TIME
1779 9876 Boredom 2015-08-10T11:05:12
Description: context_sitting_office_boredom
Context
hasActivity only ({act_sitting})
hasEmotion only ({emo_boredom})
hasLocation only ({loc_office})
OfficeWork
Property assertions: context_sitting_office_boredom
hasActivity act_sitting
hasLocation loc_office
hasEmotion emo_boredom
Activity, Location and Emotion
High-Level Context Builder
Context Mapper
MapLLC
RetreiveOntModel
ReceiveLLC
Context Instantiator
Send NewUnclassifiedHLC
InstantiateNew UnclassifiedHLC
Context Synchronizer
Synchronized LLCWindow
LLCA
Activity Recognizer
Location Detector
Emotion Recognizer

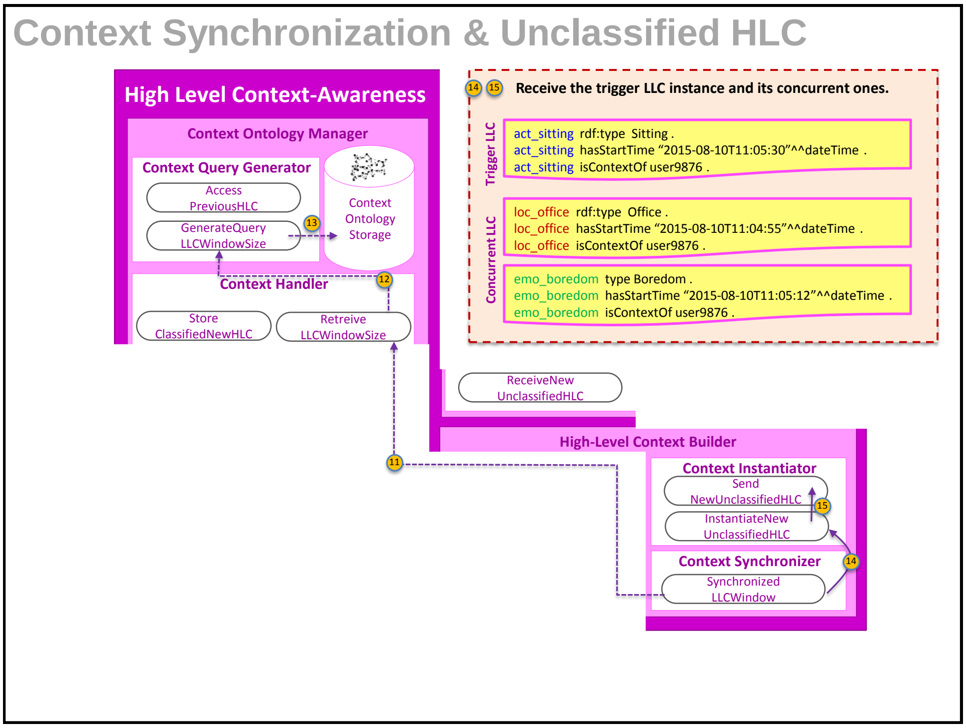

LLC Instance Mapping & Persistence
High Level Context-Awareness
Context Ontology Manager
Context Query Generator
Access PreviousHLC
GenerateQuery LLCWindowSize
Context Ontology Storage
Context Handler
Store ClassifiedNewHLC
Retreive LLCWindowSize
Receive ClassifiedNewHLC
Store MappedLLC
Handle PreviousHLC
Receive MappedLLC
Ontology Model Manager
StoredOntModel
CreateOntModel
LoadOntology
4 5 LL activities are notified to HLCA, received ReceiveLLC function and forwarded to MapLLC
6 7 The ontology model is retrieved for mapping and transformation and forwarded to MapLLC
8 MapLLC transforms the LLCA data into ontological format and send to ReceiveMappedLLC
act_sitting rdf:type Sitting .
act_sitting hasStartTime "2015-08-10T11:05:30"^^dateTime .
act_sitting isContextOf user_9876.
loc_office rdf:type Office .
loc_office hasStartTime "2015-08-10T11:04:55"^^dateTime .
loc_office isContextOf user_9876.
emo_boredom type Boredom .
emo_boredom hasStartTime "2015-08-10T11:05:12"^^dateTime .
emo_boredom isContextOf user_9876.
High-Level Context Builder
Context Mapper
MapLLC
RetreiveOntModel
ReceiveLLC
Context Instantiator
Send NewUnclassifiedHLC
InstantiateNew UnclassifiedHLC
Context Synchronizer
Synchronized LLCWindow
LLCA
Activity Recognizer
Location Detector
Emotion Recognizer

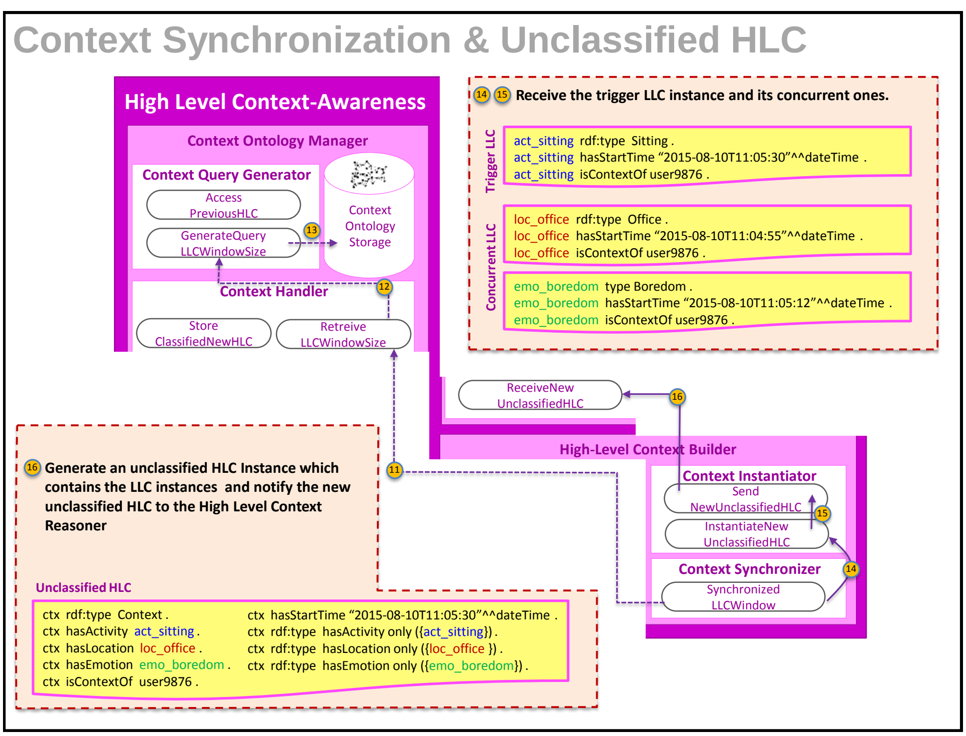

LLC Instance Mapping & Persistence
High Level Context-Awareness
Context Ontology Manager
Context Query Generator
Access PreviousHLC
GenerateQuery LLCWindowSize
Context Ontology Storage
Context Handler
Store ClassifiedNewHLC
Retreive LLCWindowSize
Receive ClassifiedNewHLC
Store MappedLLC
Handle PreviousHLC
Receive MappedLLC
Ontology Model Manager
StoredOntModel
CreateOntModel
LoadOntology
High-Level Context Builder
Context Mapper
MapLLC
RetreiveOntModel
ReceiveLLC
Context Instantiator
Send NewUnclassifiedHLC
InstantiateNew UnclassifiedHLC
Context Synchronizer
Synchronized LLCWindow
4 5 LL activities are notified to HLCA, received ReceiveLLC function and forwarded to MapLLC
6 7 The ontology model is retrieved for mapping and transformation and forwarded to MapLLC
8 MapLLC transforms the LLCA data into ontological format and send to ReceiveMappedLLC
9 10 Transformed LLCA data is stored into triple storage
Subject | Predicate | Object
User_9876 | hasActivity | Sitting
Sitting | hasStartTime | 2015-08-10 11:05:30 AM
User_9876 | hasEmotion | Boredom
Boredom | hasStartTime | 2015-08-10 11:05:12 AM
User_9876 | hasLocation | Office
Office | hasStartTime | 2015-08-10 11:04:55 AM
LLCA
Activity Recognizer
Location Detector
Emotion Recognizer


LLC Instance Mapping & Persistence
High Level Context-Awareness
Context Ontology Manager
Context Query Generator
Access PreviousHLC
GenerateQuery LLCWindowSize
Context Ontology Storage
Context Handler
Store ClassifiedNewHLC
Retreive LLCWindowSize
Receive ClassifiedNewHLC
Store MappedLLC
Handle PreviousHLC
Receive MappedLLC
Ontology Model Manager
StoredOntModel
CreateOntModel
LoadOntology
High-Level Context Notifier
Context Notifier
Retrieve LastHLC
Notify NewHLC
Classify NewHLC
Notify DCL
High-Level Context Reasoner
Context Verifier
VerifyNew UnclassifiedHLCInstance
RetreiveOntModel
ReceiveNew UnclassifiedHLC
Context Classifier
ValidateHLC
InferHLC
RetreiveOntModel
High-Level Context Builder
Context Mapper
MapLLC
RetreiveOntModel
ReceiveLLC
Context Instantiator
Send NewUnclassifiedHLC
InstantiateNew UnclassifiedHLC
Context Synchronizer
Synchronized LLCWindow
LLCA
Activity Recognizer
Location Detector
Emotion Recognizer
Data Curation Layer


Context Synchronization & Unclassified HLC
High Level Context-Awareness
Context Ontology Manager
Context Query Generator
Access PreviousHLC
GenerateQuery LLCWindowSize
Context Ontology Storage
Context Handler
Store ClassifiedNewHLC
Retreive LLCWindowSize
11 12 Context Synchronizer retrieves the LLC instances which start & end within a time window.
11:05:30
11:05:45
User 9876
llc_1777 Sitting
llc_1778 Office
llc_1779 Boredom
Trigger LLC
act_sitting rdf:type Sitting .
act_sitting hasStartTime "2015-08-10T11:05:30"^^dateTime .
act_sitting isContextOf user9876 .
ReceiveNew UnclassifiedHLC
High-Level Context Builder
Context Instantiator
Send NewUnclassifiedHLC
InstantiateNew UnclassifiedHLC
Context Synchronizer
Synchronized LLCWindow


Context Synchronization & Unclassified HLC
High Level Context-Awareness
Context Ontology Manager
Context Query Generator
Access PreviousHLC
GenerateQuery LLCWindowSize
Context Ontology Storage
Context Handler
Store ClassifiedNewHLC
Retreive LLCWindowSize
11 12 Context Synchronizer retrieves the LLC instances which start & end within a time window.
11:05:30
11:05:45
User 9876
llc_1777 Sitting
llc_1778 Office
llc_1779 Boredom
Trigger LLC
act_sitting rdf:type Sitting .
act_sitting hasStartTime "2015-08-10T11:05:30"^^dateTime .
act_sitting isContextOf user9876 .
13 Retrieves concurrent LLC from the triple storage
SELECT ?llc_id ?llc
WHERE { ?llc rdf:type ?c .
?c rdfs:subClassOf ?type .
?type rdfs:subClassOf icl2:LowLevelContext .
?llc icl2:isContextOf icl2:user_9876 .
FILTER (?starttime > startwindow^^xsd:dateTime)
}
Concurrent LLC
loc_office rdf:type Office .
loc_office hasStartTime "2015-08-10T11:04:55"^^dateTime .
loc_office isContextOf user9876 .
emo_boredom type Boredom .
emo_boredom hasStartTime "2015-08-10T11:05:12"^^dateTime .
emo_boredom isContextOf user9876 .
ReceiveNew UnclassifiedHLC
High-Level Context Builder
Context Instantiator
Send NewUnclassifiedHLC
InstantiateNew UnclassifiedHLC
Context Synchronizer
Synchronized LLCWindow


Context Synchronization & Unclassified HLC
High Level Context-Awareness
Context Ontology Manager
Context Query Generator
Access PreviousHLC
GenerateQuery LLCWindowSize
Context Ontology Storage
Context Handler
Store ClassifiedNewHLC
Retreive LLCWindowSize
14 15 Receive the trigger LLC instance and its concurrent ones.
Trigger LLC
act_sitting rdf:type Sitting .
act_sitting hasStartTime "2015-08-10T11:05:30"^^dateTime .
act_sitting isContextOf user9876 .
Concurrent LLC
loc_office rdf:type Office .
loc_office hasStartTime "2015-08-10T11:04:55"^^dateTime .
loc_office isContextOf user9876 .
emo_boredom type Boredom .
emo_boredom hasStartTime "2015-08-10T11:05:12"^^dateTime .
emo_boredom isContextOf user9876 .
ReceiveNew UnclassifiedHLC
High-Level Context Builder
Context Instantiator
Send NewUnclassifiedHLC
InstantiateNew UnclassifiedHLC
Context Synchronizer
Synchronized LLCWindow
11
12
13
14
15

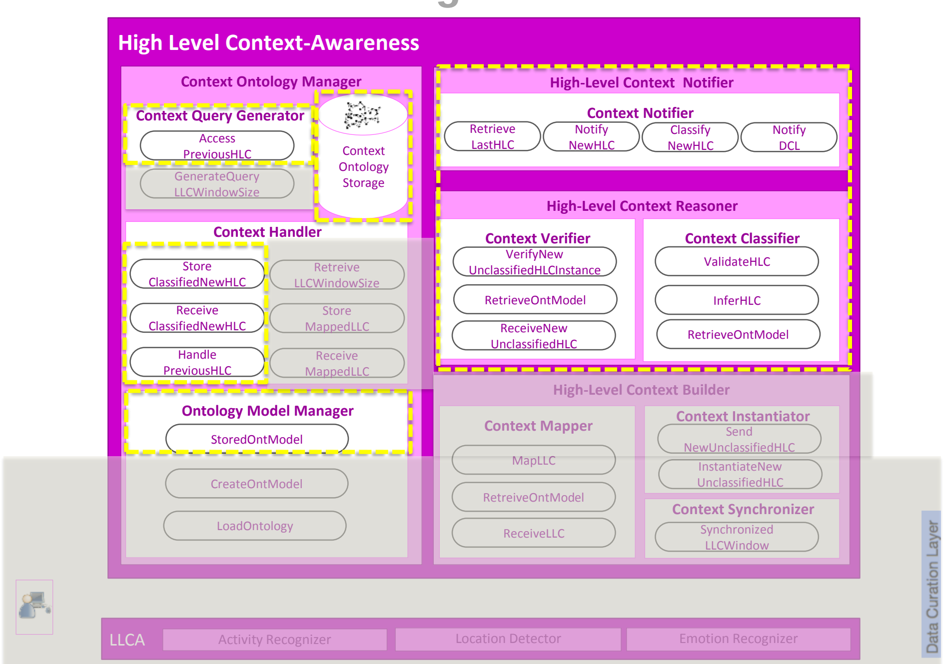

Context Synchronization & Unclassified HLC
High Level Context-Awareness
Context Ontology Manager
Context Query Generator
Access PreviousHLC
GenerateQuery LLCWindowSize
Context Ontology Storage
Context Handler
Store ClassifiedNewHLC
Retreive LLCWindowSize
14 15 Receive the trigger LLC instance and its concurrent ones.
Trigger LLC
act_sitting rdf:type Sitting .
act_sitting hasStartTime "2015-08-10T11:05:30"^^dateTime .
act_sitting isContextOf user9876 .
Concurrent LLC
loc_office rdf:type Office .
loc_office hasStartTime "2015-08-10T11:04:55"^^dateTime .
loc_office isContextOf user9876 .
emo_boredom type Boredom .
emo_boredom hasStartTime "2015-08-10T11:05:12"^^dateTime .
emo_boredom isContextOf user9876 .
ReceiveNew UnclassifiedHLC
16
High-Level Context Builder
Context Instantiator
Send NewUnclassifiedHLC
InstantiateNew UnclassifiedHLC
Context Synchronizer
Synchronized LLCWindow
16 Generate an unclassified HLC Instance which contains the LLC instances and notify the new unclassified HLC to the High Level Context Reasoner
Unclassified HLC
ctx rdf:type Context .
ctx hasActivity act_sitting .
ctx hasLocation loc_office .
ctx hasEmotion emo_boredom .
ctx isContextOf user9876 .
ctx hasStartTime "2015-08-10T11:05:30"^^dateTime .
ctx rdf:type hasActivity only ({act_sitting}) .
ctx rdf:type hasLocation only ({loc_office }) .
ctx rdf:type hasEmotion only ({emo_boredom}) .
11
12
13
14
15

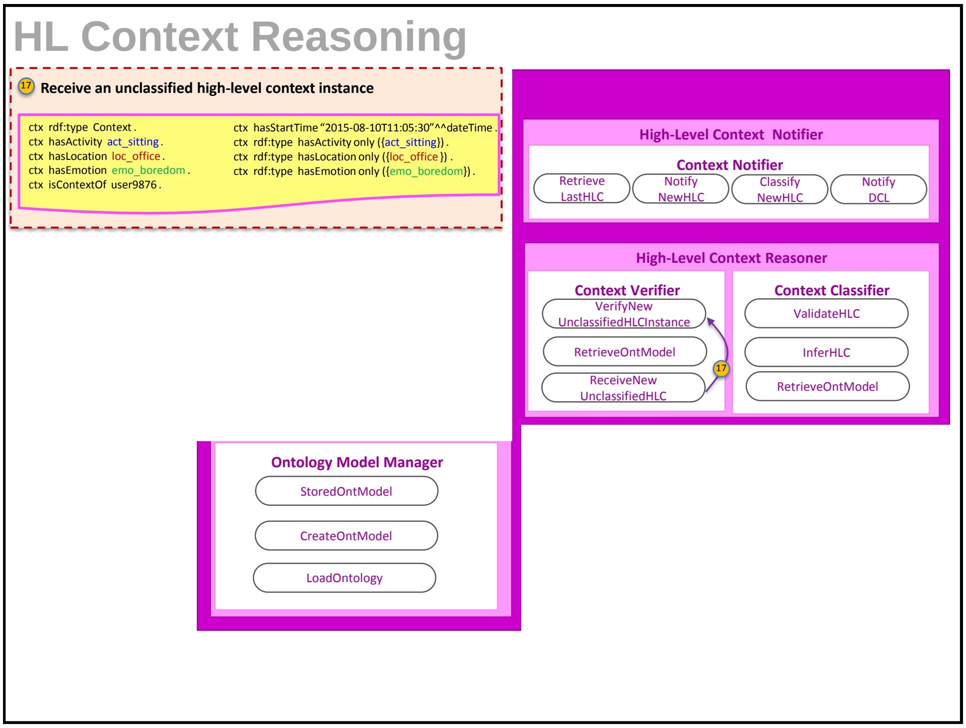
Scenario 4: Classified HLC to DCL

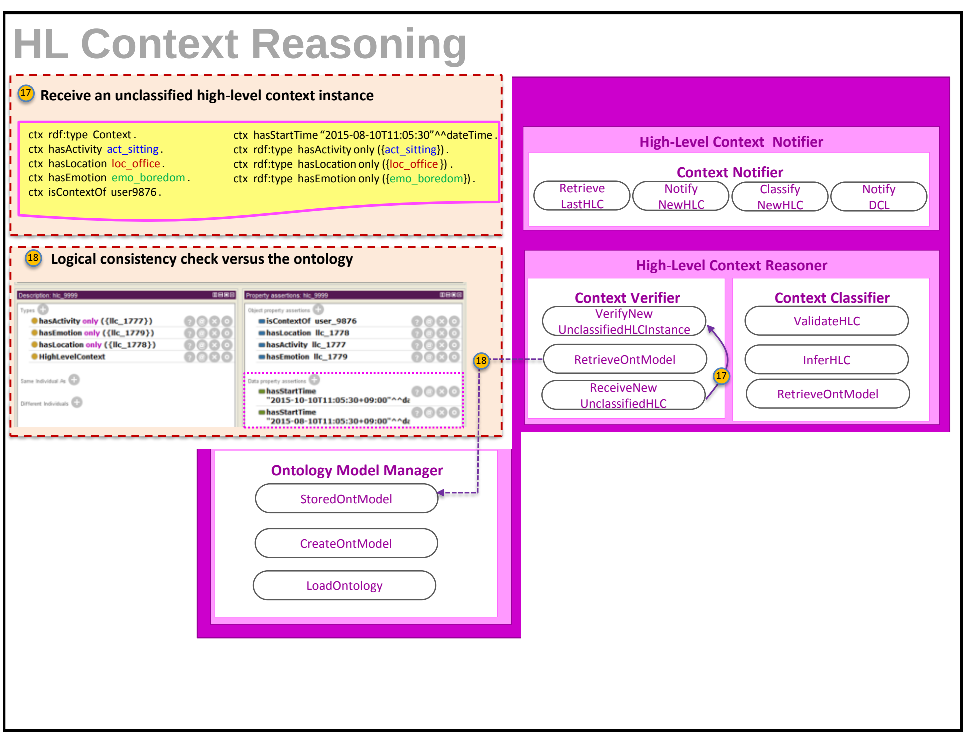
Desc. : LLC instances based **reasoning** for HLC identification and **persistence** in DCL and context ontology storage.

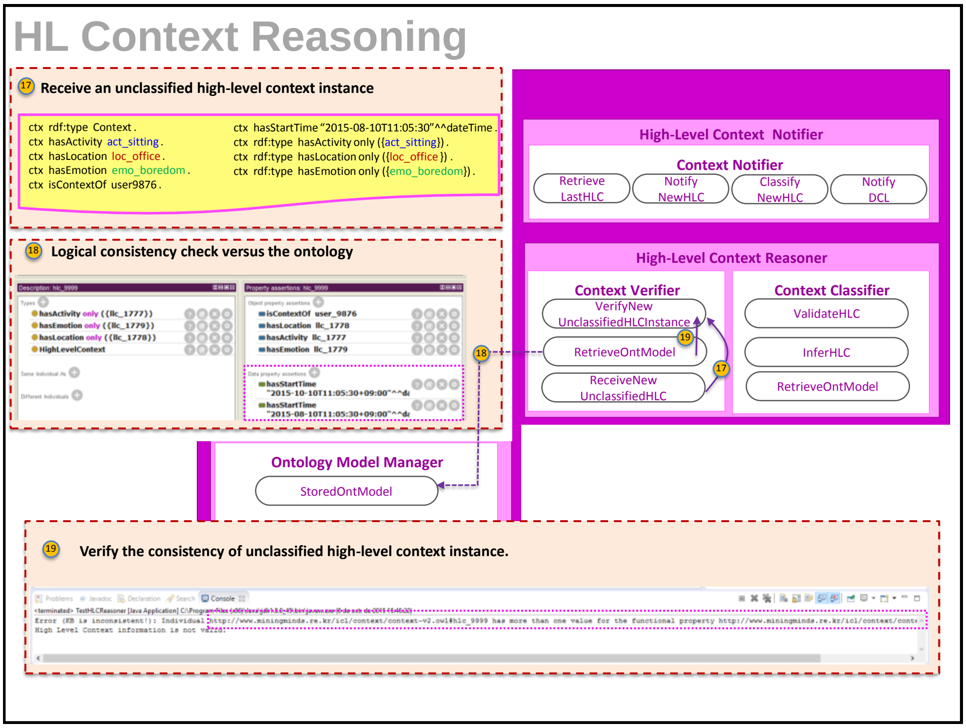
127

HL Context Reasoning

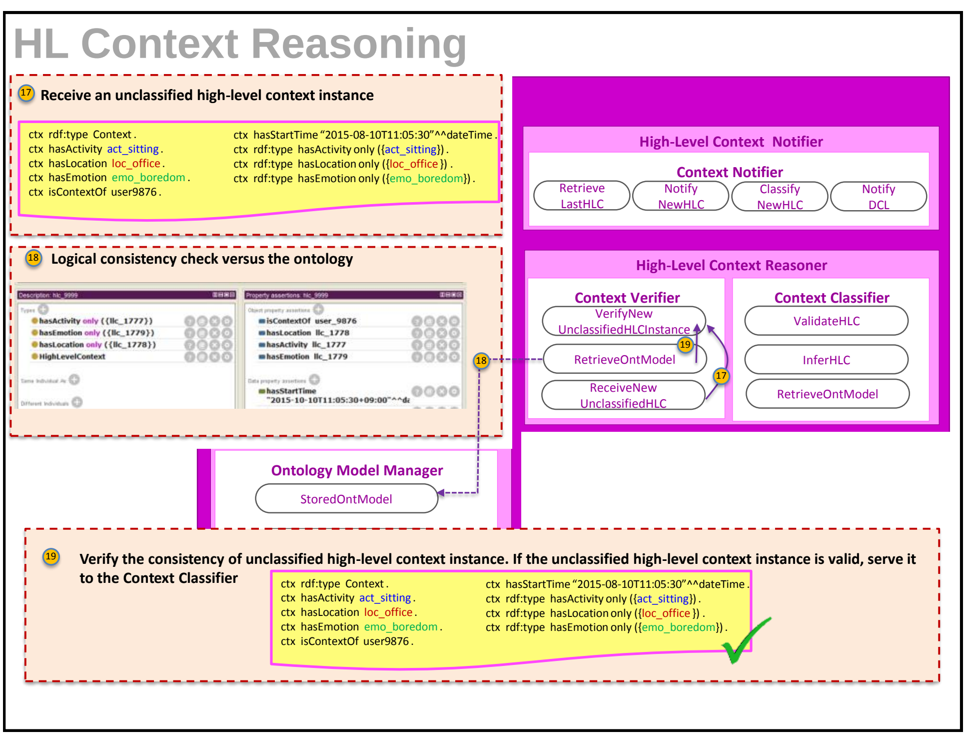



HL Context Reasoning
17 Receive an unclassified high-level context instance
ctx rdf:type Context .
ctx hasActivity act_sitting .
ctx hasLocation loc_office .
ctx hasEmotion emo_boredom .
ctx isContextOf user9876 .
ctx hasStartTime "2015-08-10T11:05:30"^^dateTime .
ctx rdf:type hasActivity only ({act_sitting}) .
ctx rdf:type hasLocation only ({loc_office}) .
ctx rdf:type hasEmotion only ({emo_boredom}) .
High-Level Context Notifier
Context Notifier
Retrieve LastHLC
Notify NewHLC
Classify NewHLC
Notify DCL
High-Level Context Reasoner
Context Verifier
VerifyNew UnclassifiedHLCInstance
RetrieveOntModel
ReceiveNew UnclassifiedHLC
17
Context Classifier
ValidateHLC
InferHLC
RetrieveOntModel
Ontology Model Manager
StoredOntModel
CreateOntModel
LoadOntology

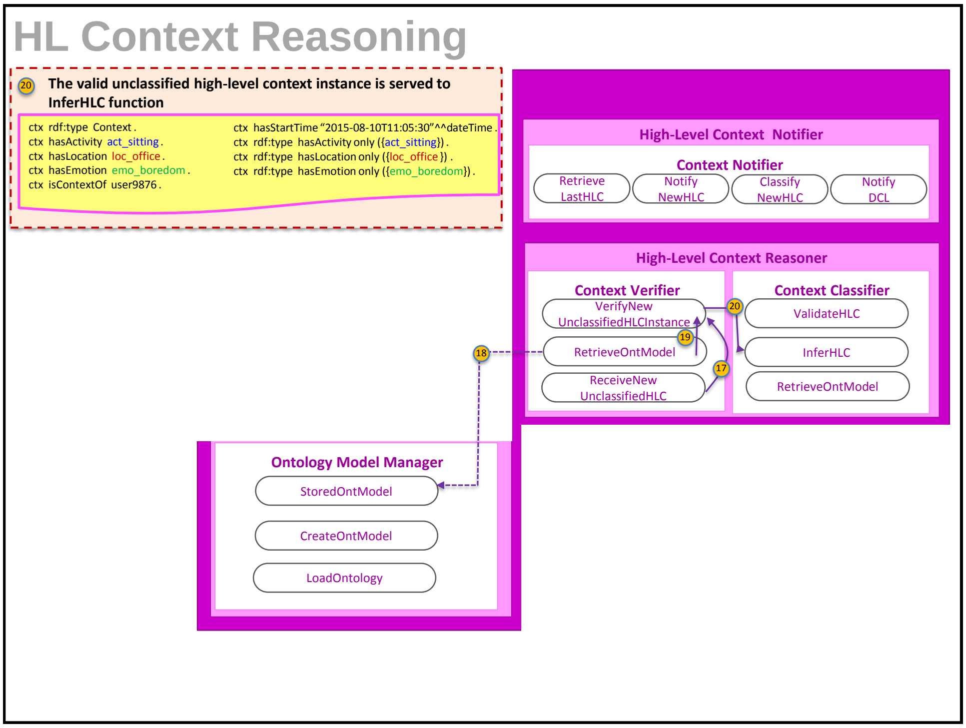

HL Context Reasoning
17 Receive an unclassified high-level context instance
ctx rdf:type Context .
ctx hasActivity act_sitting .
ctx hasLocation loc_office .
ctx hasEmotion emo_boredom .
ctx isContextOf user9876 .
ctx hasStartTime "2015-08-10T11:05:30"^^dateTime .
ctx rdf:type hasActivity only ({act_sitting}) .
ctx rdf:type hasLocation only ({loc_office}) .
ctx rdf:type hasEmotion only ({emo_boredom}) .
18 Logical consistency check versus the ontology
hasActivity only ({llc_1777})
hasEmotion only ({llc_1779})
hasLocation only ({llc_1778})
HighLevelContext
isContextOf user_9876
hasLocation llc_1778
hasActivity llc_1777
hasEmotion llc_1779
hasStartTime "2015-10-10T11:05:30+09:00"
hasStartTime "2015-08-10T11:05:30+09:00"
18
High-Level Context Notifier
Context Notifier
Retrieve LastHLC
Notify NewHLC
Classify NewHLC
Notify DCL
High-Level Context Reasoner
Context Verifier
VerifyNew UnclassifiedHLCInstance
RetrieveOntModel
ReceiveNew UnclassifiedHLC
17
Context Classifier
ValidateHLC
InferHLC
RetrieveOntModel
Ontology Model Manager
StoredOntModel
CreateOntModel
LoadOntology

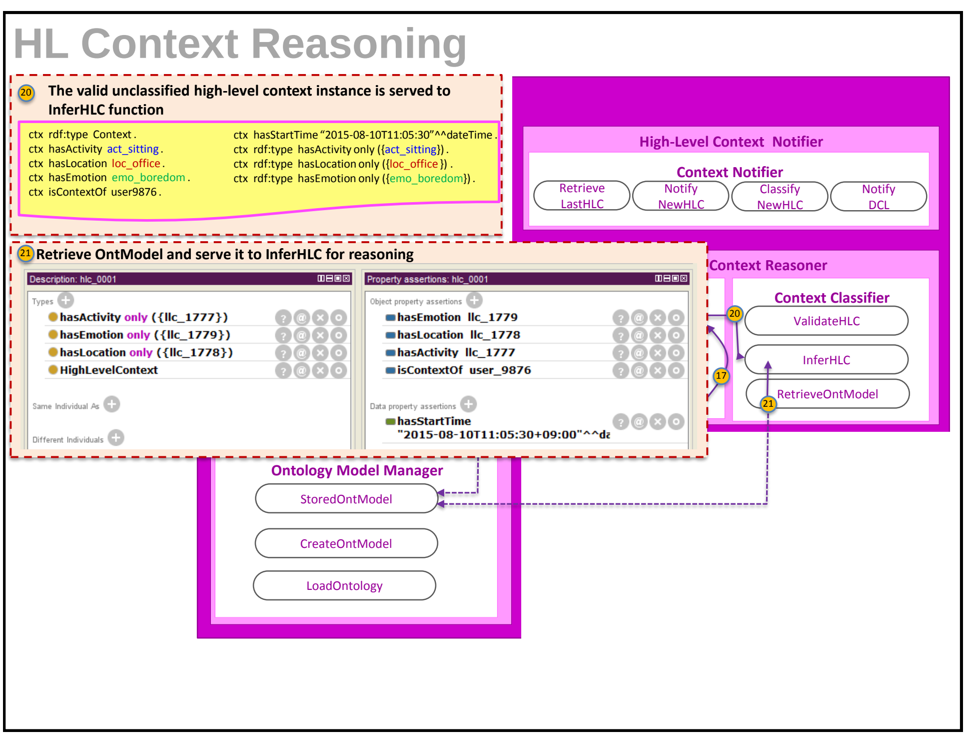


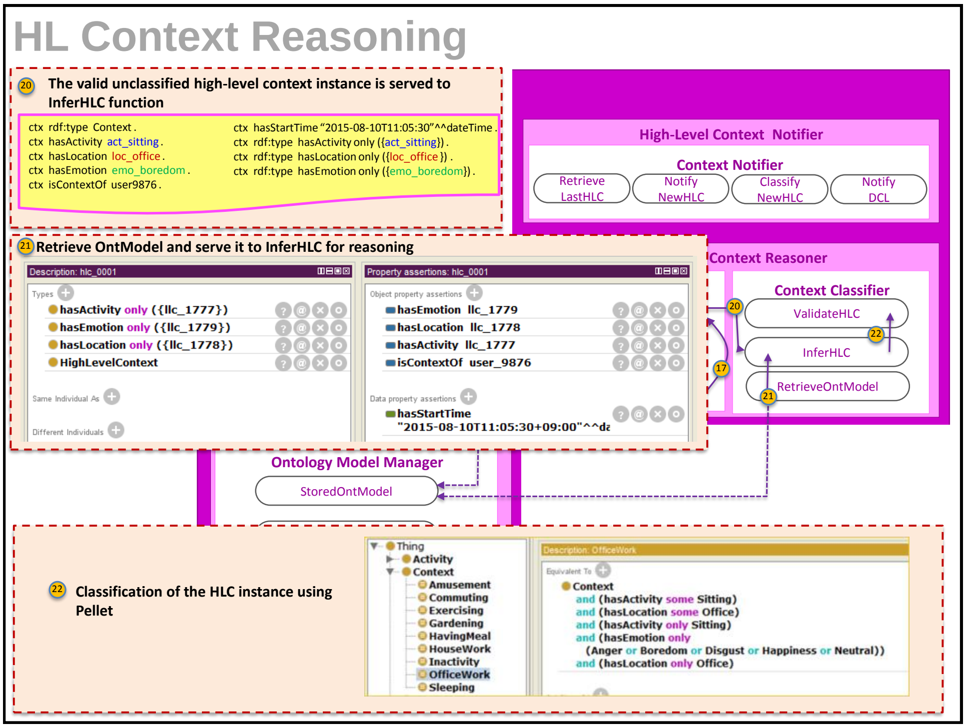
HL Context Reasoning
17 Receive an unclassified high-level context instance
ctx rdf:type Context .
ctx hasActivity act_sitting .
ctx hasLocation loc_office .
ctx hasEmotion emo_boredom .
ctx isContextOf user9876 .
ctx hasStartTime "2015-08-10T11:05:30"^^dateTime .
ctx rdf:type hasActivity only ({act_sitting}) .
ctx rdf:type hasLocation only ({loc_office}) .
ctx rdf:type hasEmotion only ({emo_boredom}) .
18 Logical consistency check versus the ontology
hasActivity only ({llc_1777})
hasEmotion only ({llc_1779})
hasLocation only ({llc_1778})
HighLevelContext
isContextOf user_9876
hasLocation llc_1778
hasActivity llc_1777
hasEmotion llc_1779
hasStartTime "2015-10-10T11:05:30+09:00"^^d
hasStartTime "2015-08-10T11:05:30+09:00"^^d
High-Level Context Notifier
Context Notifier
Retrieve LastHLC
Notify NewHLC
Classify NewHLC
Notify DCL
High-Level Context Reasoner
Context Verifier
VerifyNew UnclassifiedHLCInstance
RetrieveOntModel
ReceiveNew UnclassifiedHLC
Context Classifier
ValidateHLC
InferHLC
RetrieveOntModel
Ontology Model Manager
StoredOntModel
19 Verify the consistency of unclassified high-level context instance.
High Level Context information is not valid.

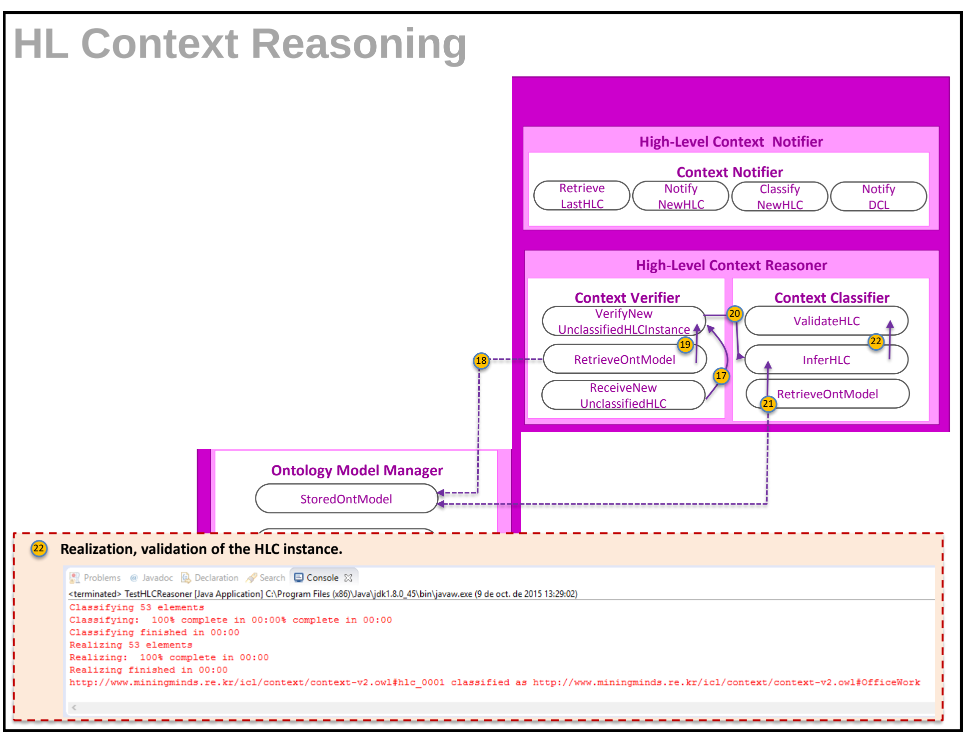


HL Context Reasoning
17 Receive an unclassified high-level context instance
ctx rdf:type Context .
ctx hasActivity act_sitting .
ctx hasLocation loc_office .
ctx hasEmotion emo_boredom .
ctx isContextOf user9876 .
ctx hasStartTime "2015-08-10T11:05:30"^^dateTime .
ctx rdf:type hasActivity only ({act_sitting}) .
ctx rdf:type hasLocation only ({loc_office}) .
ctx rdf:type hasEmotion only ({emo_boredom}) .
18 Logical consistency check versus the ontology
hasActivity only ({llc_1777})
hasEmotion only ({llc_1779})
hasLocation only ({llc_1778})
HighLevelContext
isContextOf user_9876
hasLocation llc_1778
hasActivity llc_1777
hasEmotion llc_1779
hasStartTime "2015-10-10T11:05:30+09:00"^^d
High-Level Context Notifier
Context Notifier
Retrieve LastHLC
Notify NewHLC
Classify NewHLC
Notify DCL
High-Level Context Reasoner
Context Verifier
VerifyNew UnclassifiedHLCInstance
RetrieveOntModel
ReceiveNew UnclassifiedHLC
Context Classifier
ValidateHLC
InferHLC
RetrieveOntModel
Ontology Model Manager
StoredOntModel
19 Verify the consistency of unclassified high-level context instance. If the unclassified high-level context instance is valid, serve it to the Context Classifier
ctx rdf:type Context .
ctx hasActivity act_sitting .
ctx hasLocation loc_office .
ctx hasEmotion emo_boredom .
ctx isContextOf user9876 .
ctx hasStartTime "2015-08-10T11:05:30"^^dateTime .
ctx rdf:type hasActivity only ({act_sitting}) .
ctx rdf:type hasLocation only ({loc_office}) .
ctx rdf:type hasEmotion only ({emo_boredom}) .

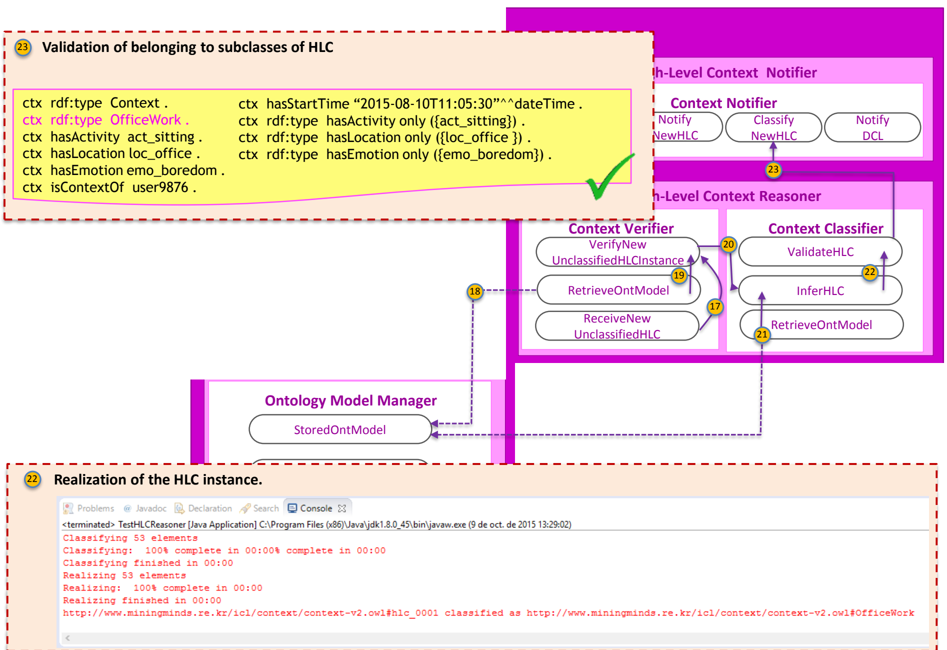

HL Context Reasoning
20 The valid unclassified high-level context instance is served to InferHLC function
ctx rdf:type Context .
ctx hasActivity act_sitting .
ctx hasLocation loc_office .
ctx hasEmotion emo_boredom .
ctx isContextOf user9876 .
ctx hasStartTime "2015-08-10T11:05:30"^^dateTime .
ctx rdf:type hasActivity only ({act_sitting}) .
ctx rdf:type hasLocation only ({loc_office}) .
ctx rdf:type hasEmotion only ({emo_boredom}) .
High-Level Context Notifier
Context Notifier
Retrieve LastHLC
Notify NewHLC
Classify NewHLC
Notify DCL
High-Level Context Reasoner
Context Verifier
VerifyNew UnclassifiedHLCInstance
RetrieveOntModel
ReceiveNew UnclassifiedHLC
Context Classifier
ValidateHLC
InferHLC
RetrieveOntModel
17
18
19
20
Ontology Model Manager
StoredOntModel
CreateOntModel
LoadOntology


HL Context Reasoning
20 The valid unclassified high-level context instance is served to InferHLC function
ctx rdf:type Context .
ctx hasActivity act_sitting .
ctx hasLocation loc_office .
ctx hasEmotion emo_boredom .
ctx isContextOf user9876 .
ctx hasStartTime "2015-08-10T11:05:30"^^dateTime .
ctx rdf:type hasActivity only ({act_sitting}) .
ctx rdf:type hasLocation only ({loc_office}) .
ctx rdf:type hasEmotion only ({emo_boredom}) .
21 Retrieve OntModel and serve it to InferHLC for reasoning
Description: hlc_0001
Types
hasActivity only ({llc_1777})
hasEmotion only ({llc_1779})
hasLocation only ({llc_1778})
HighLevelContext
Same Individual As
Different Individuals
Property assertions: hlc_0001
Object property assertions
hasEmotion llc_1779
hasLocation llc_1778
hasActivity llc_1777
isContextOf user_9876
Data property assertions
hasStartTime "2015-08-10T11:05:30+09:00"^^dá
High-Level Context Notifier
Context Notifier
Retrieve LastHLC
Notify NewHLC
Classify NewHLC
Notify DCL
Context Reasoner
Context Classifier
ValidateHLC
InferHLC
RetrieveOntModel
17
20
21
Ontology Model Manager
StoredOntModel
CreateOntModel
LoadOntology

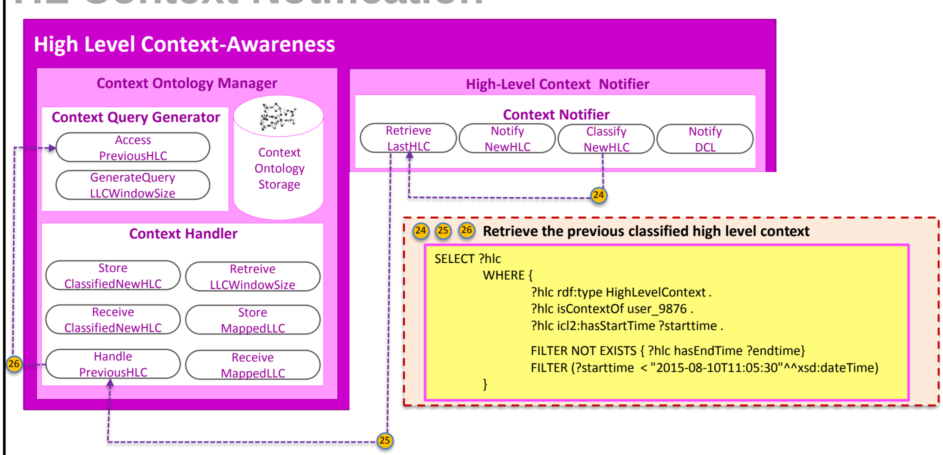

HL Context Reasoning
20 The valid unclassified high-level context instance is served to InferHLC function
ctx rdf:type Context .
ctx hasActivity act_sitting .
ctx hasLocation loc_office .
ctx hasEmotion emo_boredom .
ctx isContextOf user9876 .
ctx hasStartTime "2015-08-10T11:05:30"^^dateTime .
ctx rdf:type hasActivity only ({act_sitting}) .
ctx rdf:type hasLocation only ({loc_office}) .
ctx rdf:type hasEmotion only ({emo_boredom}) .
High-Level Context Notifier
Context Notifier
Retrieve LastHLC
Notify NewHLC
Classify NewHLC
Notify DCL
21 Retrieve OntModel and serve it to InferHLC for reasoning
Description: hlc_0001
Types
hasActivity only ({llc_1777})
hasEmotion only ({llc_1779})
hasLocation only ({llc_1778})
HighLevelContext
Same Individual As
Different Individuals
Property assertions: hlc_0001
Object property assertions
hasEmotion llc_1779
hasLocation llc_1778
hasActivity llc_1777
isContextOf user_9876
Data property assertions
hasStartTime "2015-08-10T11:05:30+09:00"^^dz
Context Reasoner
Context Classifier
ValidateHLC
InferHLC
RetrieveOntModel
Ontology Model Manager
StoredOntModel
22 Classification of the HLC instance using Pellet
Thing
Activity
Context
Amusement
Commuting
Exercising
Gardening
HavingMeal
HouseWork
Inactivity
OfficeWork
Sleeping
Description: OfficeWork
Equivalent To
Context
and (hasActivity some Sitting)
and (hasLocation some Office)
and (hasActivity only Sitting)
and (hasEmotion only
(Anger or Boredom or Disgust or Happiness or Neutral))
and (hasLocation only Office)

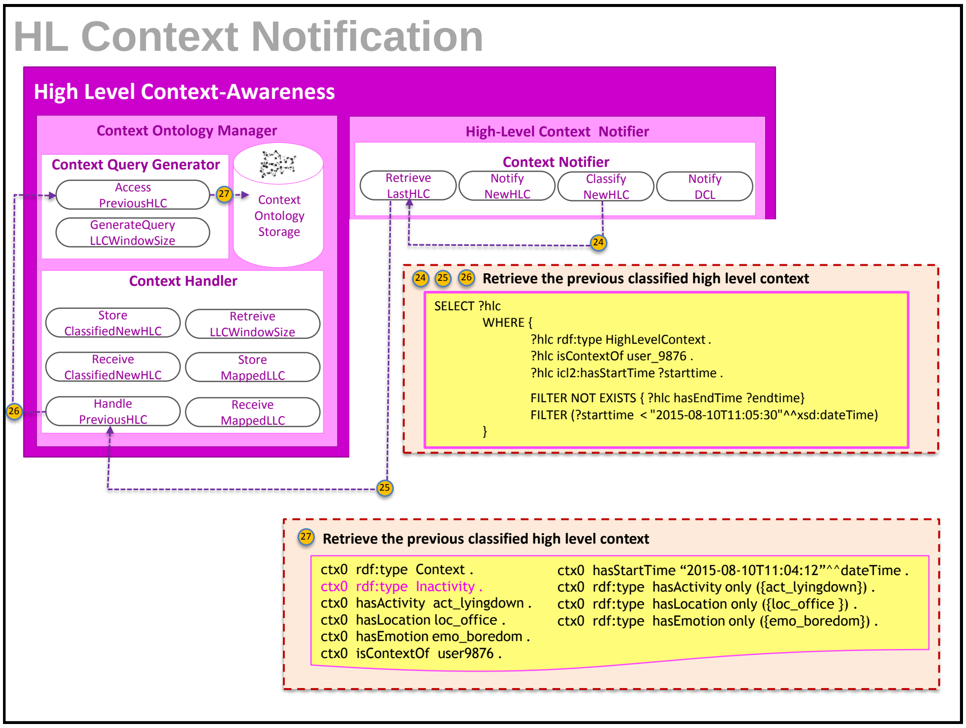

HL Context Reasoning
High-Level Context Notifier
Context Notifier
Retrieve LastHLC
Notify NewHLC
Classify NewHLC
Notify DCL
High-Level Context Reasoner
Context Verifier
VerifyNew UnclassifiedHLCInstance
RetrieveOntModel
ReceiveNew UnclassifiedHLC
Context Classifier
ValidateHLC
InferHLC
RetrieveOntModel
Ontology Model Manager
StoredOntModel
22 Realization, validation of the HLC instance.
Problems Javadoc Declaration Search Console
<terminated> TestHLCReasoner [Java Application] C:\Program Files (x86)\Java\jdk1.8.0_45\bin\javaw.exe (9 de oct. de 2015 13:29:02)
Classifying 53 elements
Classifying: 100% complete in 00:00% complete in 00:00
Classifying finished in 00:00
Realizing 53 elements
Realizing: 100% complete in 00:00
Realizing finished in 00:00
http://www.miningminds.re.kr/icl/context/context-v2.owl#hlc_0001 classified as http://www.miningminds.re.kr/icl/context/context-v2.owl#OfficeWork

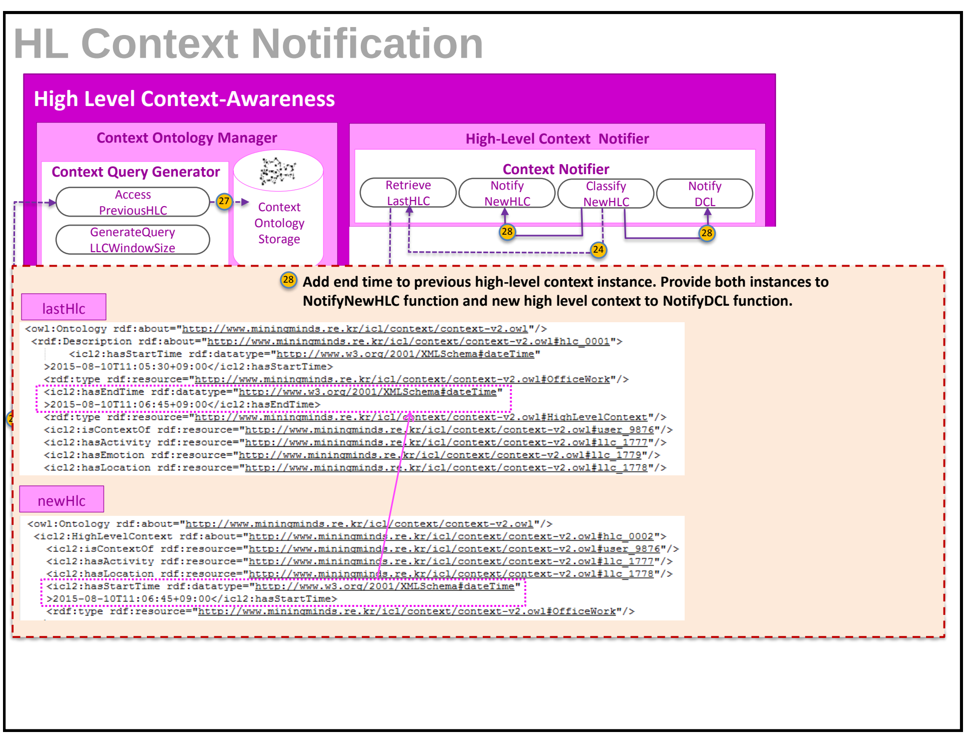
HL Context Reasoning

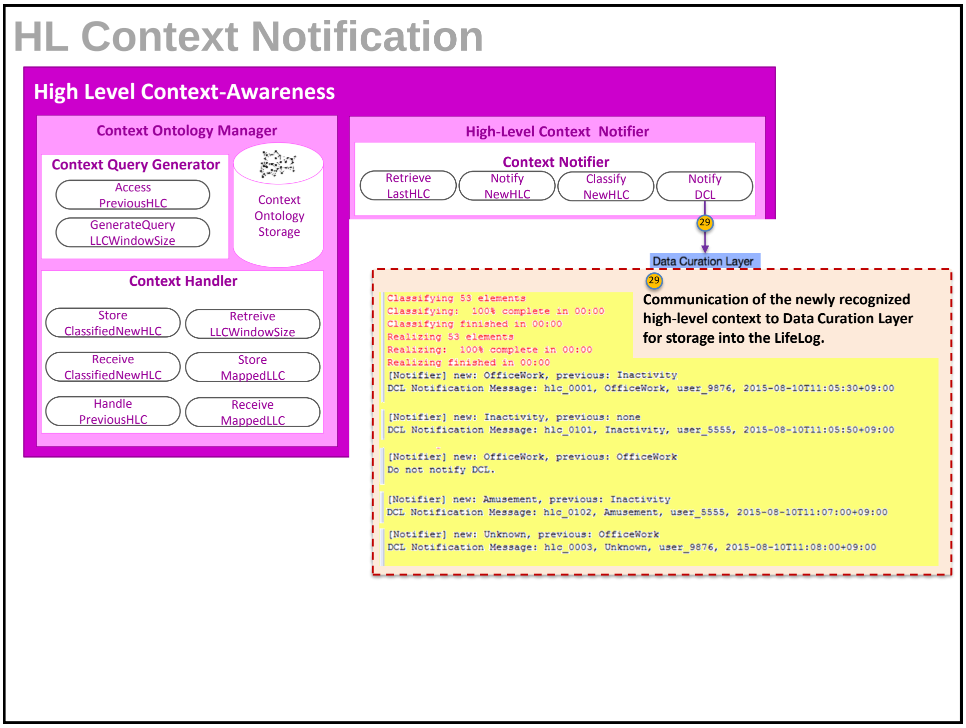


HL Context Notification

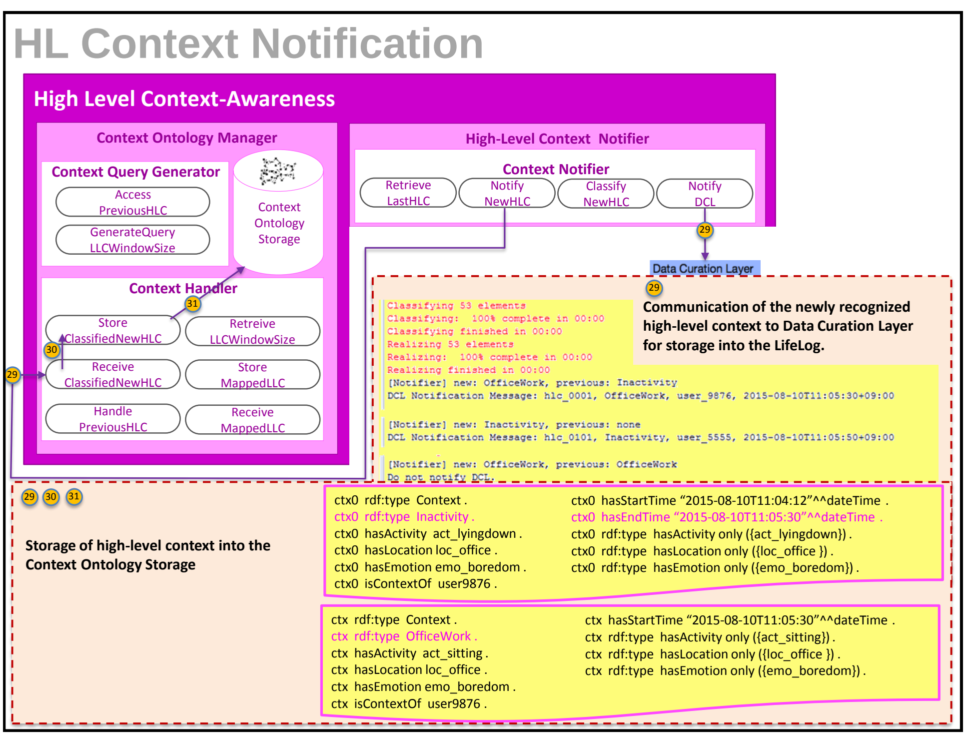



HL Context Notification
High Level Context-Awareness
Context Ontology Manager
Context Query Generator
Access PreviousHLC
GenerateQuery LLCWindowSize
Context Ontology Storage
Context Handler
Store ClassifiedNewHLC
Retreive LLCWindowSize
Receive ClassifiedNewHLC
Store MappedLLC
Handle PreviousHLC
Receive MappedLLC
High-Level Context Notifier
Context Notifier
Retrieve LastHLC
Notify NewHLC
Classify NewHLC
Notify DCL
27
26
24
25
24 25 26 Retrieve the previous classified high level context
SELECT ?hlc
WHERE {
?hlc rdf:type HighLevelContext .
?hlc isContextOf user_9876 .
?hlc icl2:hasStartTime ?starttime .
FILTER NOT EXISTS { ?hlc hasEndTime ?endtime}
FILTER (?starttime < "2015-08-10T11:05:30"^^xsd:dateTime)
}
27 Retrieve the previous classified high level context
ctx0 rdf:type Context .
ctx0 rdf:type Inactivity .
ctx0 hasActivity act_lyingdown .
ctx0 hasLocation loc_office .
ctx0 hasEmotion emo_boredom .
ctx0 isContextOf user9876 .
ctx0 hasStartTime "2015-08-10T11:04:12"^^dateTime .
ctx0 rdf:type hasActivity only ({act_lyingdown}) .
ctx0 rdf:type hasLocation only ({loc_office }) .
ctx0 rdf:type hasEmotion only ({emo_boredom}) .


HL Context Notification
High Level Context-Awareness
Context Ontology Manager
Context Query Generator
Access PreviousHLC
GenerateQuery LLCWindowSize
Context Ontology Storage
High-Level Context Notifier
Context Notifier
Retrieve LastHLC
Notify NewHLC
Classify NewHLC
Notify DCL
27
28
24
28
28 Add end time to previous high-level context instance. Provide both instances to NotifyNewHLC function and new high level context to NotifyDCL function.
lastHlc
<owl:Ontology rdf:about="http://www.miningminds.re.kr/icl/context/context-v2.owl"/>
<rdf:Description rdf:about="http://www.miningminds.re.kr/icl/context/context-v2.owl#hlc_0001">
<icl2:hasStartTime rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime"
>2015-08-10T11:05:30+09:00</icl2:hasStartTime>
<rdf:type rdf:resource="http://www.miningminds.re.kr/icl/context/context-v2.owl#OfficeWork"/>
<icl2:hasEndTime rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime"
>2015-08-10T11:06:45+09:00</icl2:hasEndTime>
<rdf:type rdf:resource="http://www.miningminds.re.kr/icl/context/context-v2.owl#HighLevelContext"/>
<icl2:isContextOf rdf:resource="http://www.miningminds.re.kr/icl/context/context-v2.owl#user_9876"/>
<icl2:hasActivity rdf:resource="http://www.miningminds.re.kr/icl/context/context-v2.owl#llc_1777"/>
<icl2:hasEmotion rdf:resource="http://www.miningminds.re.kr/icl/context/context-v2.owl#llc_1779"/>
<icl2:hasLocation rdf:resource="http://www.miningminds.re.kr/icl/context/context-v2.owl#llc_1778"/>
newHlc
<owl:Ontology rdf:about="http://www.miningminds.re.kr/icl/context/context-v2.owl"/>
<icl2:HighLevelContext rdf:about="http://www.miningminds.re.kr/icl/context/context-v2.owl#hlc_0002">
<icl2:isContextOf rdf:resource="http://www.miningminds.re.kr/icl/context/context-v2.owl#user_9876"/>
<icl2:hasActivity rdf:resource="http://www.miningminds.re.kr/icl/context/context-v2.owl#llc_1777"/>
<icl2:hasLocation rdf:resource="http://www.miningminds.re.kr/icl/context/context-v2.owl#llc_1778"/>
<icl2:hasStartTime rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime"
>2015-08-10T11:06:45+09:00</icl2:hasStartTime>
<rdf:type rdf:resource="http://www.miningminds.re.kr/icl/context/context-v2.owl#OfficeWork"/>



HL Context Notification
High Level Context-Awareness
Context Ontology Manager
Context Query Generator
Access PreviousHLC
GenerateQuery LLCWindowSize
Context Ontology Storage
Context Handler
Store ClassifiedNewHLC
Retreive LLCWindowSize
Receive ClassifiedNewHLC
Store MappedLLC
Handle PreviousHLC
Receive MappedLLC
High-Level Context Notifier
Context Notifier
Retrieve LastHLC
Notify NewHLC
Classify NewHLC
Notify DCL
29
Data Curation Layer
29
Communication of the newly recognized high-level context to Data Curation Layer for storage into the LifeLog.
Classifying 53 elements
Classifying: 100% complete in 00:00
Classifying finished in 00:00
Realizing 53 elements
Realizing: 100% complete in 00:00
Realizing finished in 00:00
[Notifier] new: OfficeWork, previous: Inactivity
DCL Notification Message: hlc_0001, OfficeWork, user_9876, 2015-08-10T11:05:30+09:00
[Notifier] new: Inactivity, previous: none
DCL Notification Message: hlc_0101, Inactivity, user_5555, 2015-08-10T11:05:50+09:00
[Notifier] new: OfficeWork, previous: OfficeWork
Do not notify DCL.
[Notifier] new: Amusement, previous: Inactivity
DCL Notification Message: hlc_0102, Amusement, user_5555, 2015-08-10T11:07:00+09:00
[Notifier] new: Unknown, previous: OfficeWork
DCL Notification Message: hlc_0003, Unknown, user_9876, 2015-08-10T11:08:00+09:00

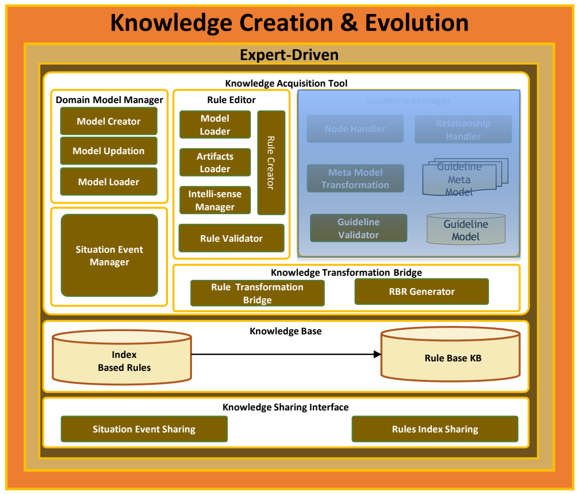


HL Context Notification
High Level Context-Awareness
Context Ontology Manager
Context Query Generator
Access PreviousHLC
GenerateQuery LLCWindowSize
Context Ontology Storage
Context Handler
31
Store ClassifiedNewHLC
Retreive LLCWindowSize
30
29
Receive ClassifiedNewHLC
Store MappedLLC
Handle PreviousHLC
Receive MappedLLC
High-Level Context Notifier
Context Notifier
Retrieve LastHLC
Notify NewHLC
Classify NewHLC
Notify DCL
29
Data Curation Layer
29
Communication of the newly recognized high-level context to Data Curation Layer for storage into the LifeLog.
Classifying 53 elements
Classifying: 100% complete in 00:00
Classifying finished in 00:00
Realizing 53 elements
Realizing: 100% complete in 00:00
Realizing finished in 00:00
[Notifier] new: OfficeWork, previous: Inactivity
DCL Notification Message: hlc_0001, OfficeWork, user_9876, 2015-08-10T11:05:30+09:00
[Notifier] new: Inactivity, previous: none
DCL Notification Message: hlc_0101, Inactivity, user_5555, 2015-08-10T11:05:50+09:00
[Notifier] new: OfficeWork, previous: OfficeWork
Do not notify DCL.
29 30 31
Storage of high-level context into the Context Ontology Storage
ctx0 rdf:type Context .
ctx0 rdf:type Inactivity .
ctx0 hasActivity act_lyingdown .
ctx0 hasLocation loc_office .
ctx0 hasEmotion emo_boredom .
ctx0 isContextOf user9876 .
ctx0 hasStartTime "2015-08-10T11:04:12"^^dateTime .
ctx0 hasEndTime "2015-08-10T11:05:30"^^dateTime .
ctx0 rdf:type hasActivity only ({act_lyingdown}) .
ctx0 rdf:type hasLocation only ({loc_office }) .
ctx0 rdf:type hasEmotion only ({emo_boredom}) .
ctx rdf:type Context .
ctx rdf:type OfficeWork .
ctx hasActivity act_sitting .
ctx hasLocation loc_office .
ctx hasEmotion emo_boredom .
ctx isContextOf user9876 .
ctx hasStartTime "2015-08-10T11:05:30"^^dateTime .
ctx rdf:type hasActivity only ({act_sitting}) .
ctx rdf:type hasLocation only ({loc_office }) .
ctx rdf:type hasEmotion only ({emo_boredom}) .

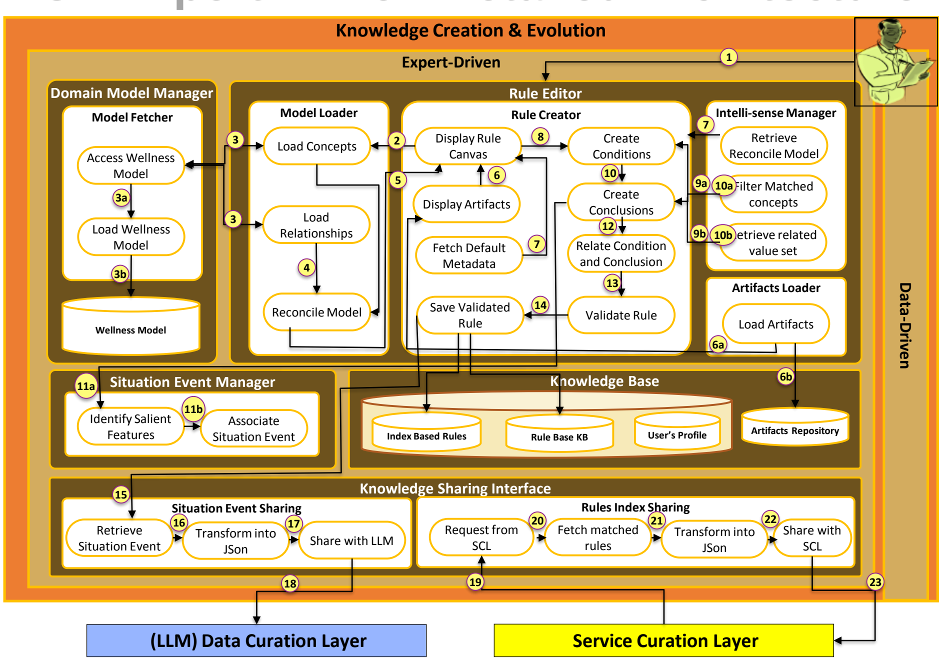
STENCILS for KCL Micro-Demos

STENCILS for Expert-Driven Micro-Demo

KCL Expert Driven Detailed Architecture



KCL Expert Driven Detailed Architecture



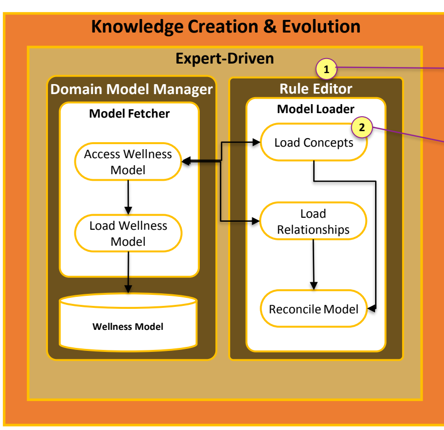
Scenario 1: Rule Creation

Desc. : I-KAT **rule creation** scenario for physical activity through **wellness** model and **Intelli-sense** support.

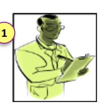
Major Steps:

1. Wellness Model Loading
2. Rule Creation
3. Intelli-sense based concept filtration

1. Wellness Model Loading



1



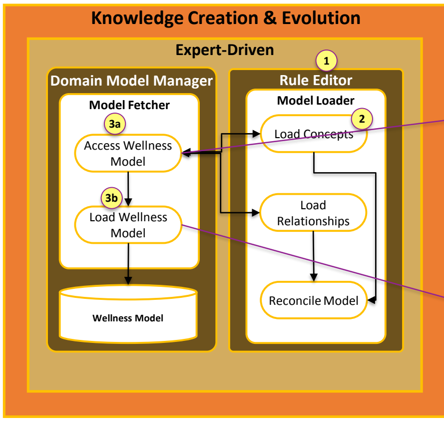
2

ID	Concepts
1	MM – Wellness Model
2	Profile Information
3	Physiological
4	Blood Pressure
5	Systolic
6	Diastolic

1 Domain expert starts creating rule using Rule Editor

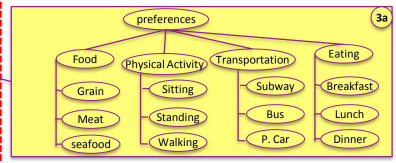
2 Load Concepts fetches all the concepts from wellness model

1. Wellness Model Loading



3b

```
<beans:bean id="dataSource1"
class="org.apache.commons.dbcp.BasicDataSource" destroy-
method="close">
    <beans:property name="driverClassName"
value="com.microsoft.sqlserver.jdbc.SQLServerDriver" />
     <beans:property name="url"
value="jdbc:sqlserver://163.180.116.194:1433;instance=SQLEXPRESS;dat
abaseName="DBName" />
    <beans:property name="username" value="sa" />
    <beans:property name="password" value="abcdefg" />
  </beans:bean>
```



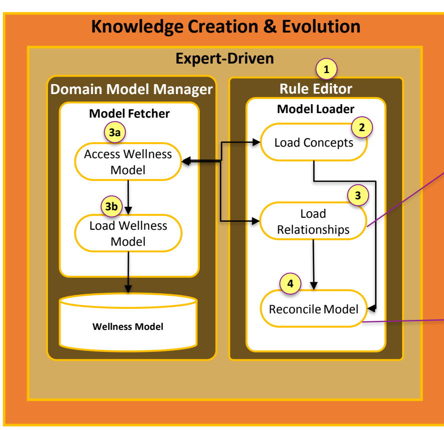
1 Domain expert starts creating rule using Rule Editor

2 Load Concepts fetches all the concepts from wellness model

3a Access Wellness Model function get the connection to the wellness model in database

3b Load Wellness Model loads the complete wellness model to the editor

1. Wellness Model Loading



3

C1ID	RelationID	C2ID
41	116680003	38
42	116680003	38
43	116680003	38
44	116680003	38
45	116680003	38

4

C1	R	C2
Pain	Is-a	Symptom
Allergies	Is-a	Symptom
Sleep Disorder	Is-a	Symptom
Fatigue	Is-a	Symptom
Digestion Disorder	Is-a	Symptom

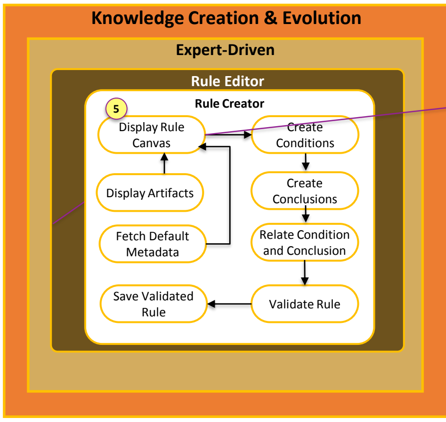
3a Access Wellness Model function get the connection to the wellness model in database

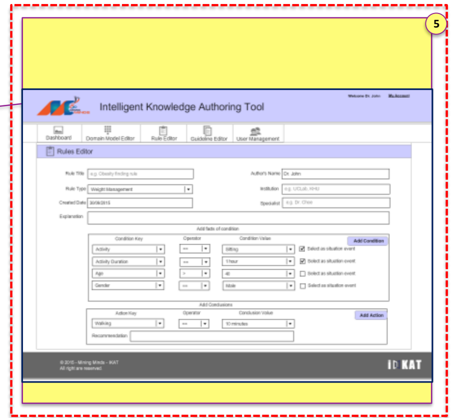
3b Load Wellness Model loads the complete wellness model to the editor

3 Load Relationships loads all the relationships among the concepts

4 Reconcile Model merge the concepts and relationships and transforms id's to labels of concepts.

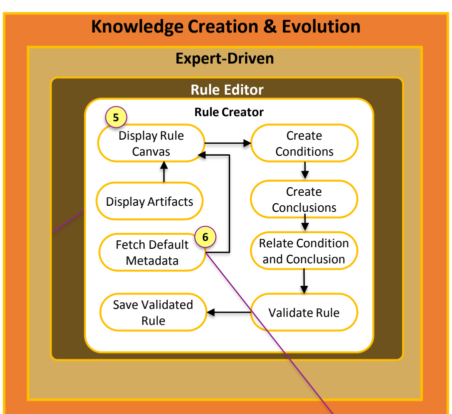
2. Rule Creation





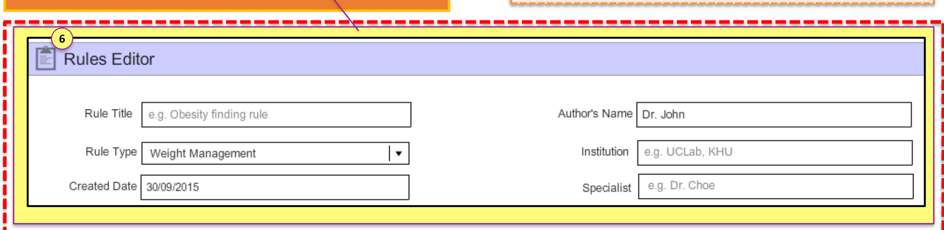
5 The graphical user interface (GUI) of the Rule Editor is rendered by Display Rule Canvas

2. Rule Creation

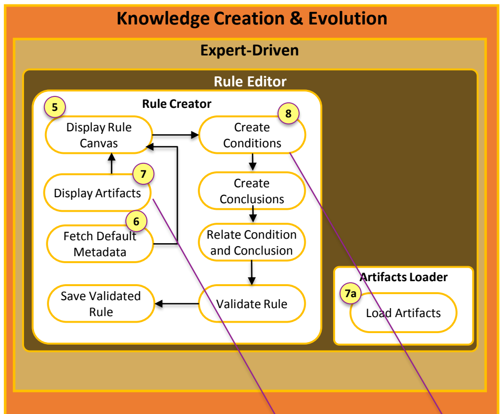


5 The graphical user interface (GUI) of the Rule Editor is rendered by Display Rule Canvas

6 The default information (Metadata)about the rules and some guidelines are fetched to Rule Editor by Fetch Default Metadata.



2. Rule Creation

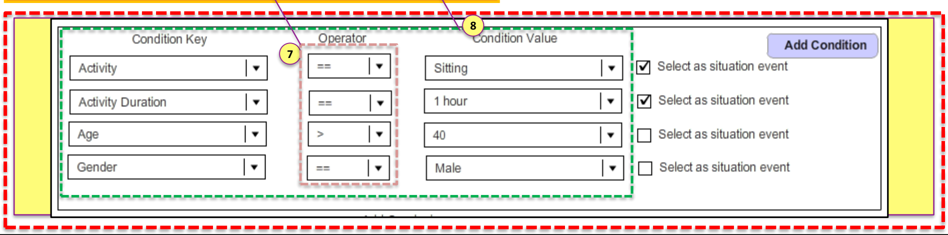


5 The graphical user interface (GUI) of the Rule Editor is rendered by Display Rule Canvas

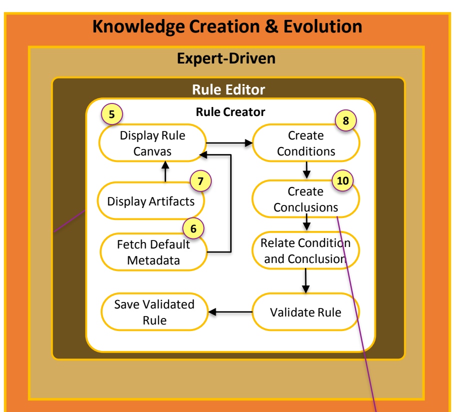
6 The default information (Metadata)about the rules and some guidelines are fetched to Rule Editor by Fetch Default Metadata.

7 Required operators are loaded to Rule Editor by Display Artifacts with the help of Load Artifacts

8 Create Conditions facilitates the expert to create desired facts that includes in rule creation with the help of Condition Keys, Operators, and Condition Values



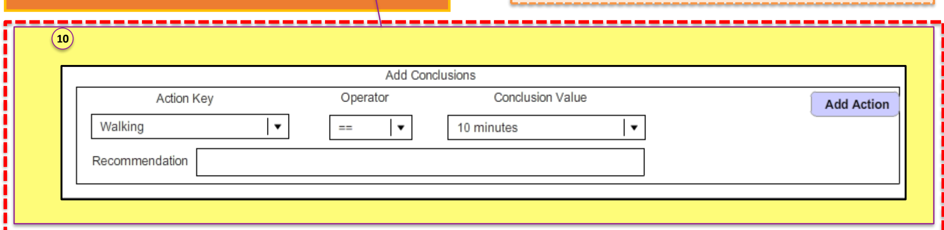
2. Rule Creation



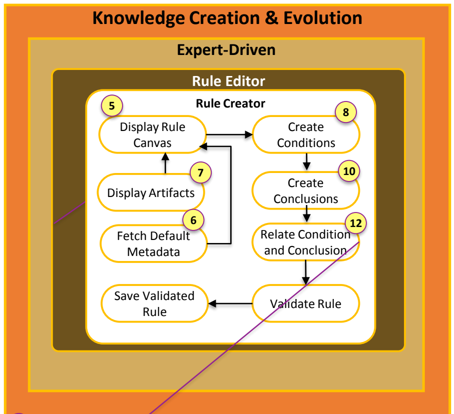
7 Required operators are loaded to Rule Editor by Display Artifacts with the help of Load Artifacts

8 Create Conditions facilitates the expert to create desired facts that includes in rule creation with the help of Condition Keys, Operators, and Condition Values

10 The Create Conclusions facilitates the domain expert to create multiple Conclusions with some Recommendations text for the explanation purpose



2. Rule Creation



8 Create Conditions facilitates the expert to create desired facts that includes in rule creation with the help of Condition Keys, Operators, and Condition Values

10 The Create Conclusions facilitates the domain expert to create multiple Conclusions with some Recommendations text for the explanation purpose

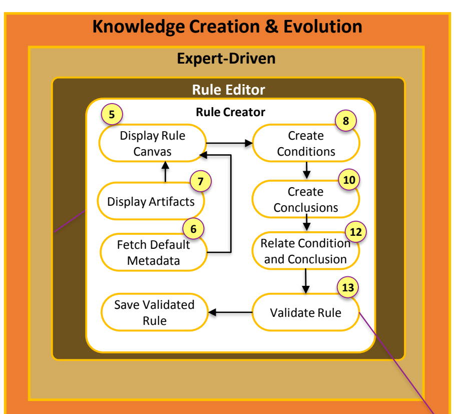
12 The Condition Key is associate with corresponding rule and that rule id is associate with corresponding conclusions.

12

RuleConID	RuleID	ConditionID
20550	20092	1
20551	20093	1
20552	20092	3
20553	20093	3
20554	20092	5

Rule ID	Key	Operator	Value
20092	Current Activity	=	Walking
20092	Activity Duration	=	10 min

2. Rule Creation



10 The Create Conclusions facilitates the domain expert to create multiple Conclusions with some Recommendations text for the explanation purpose

12 The Condition Key is associate with corresponding rule and that rule id is associate with corresponding conclusions.

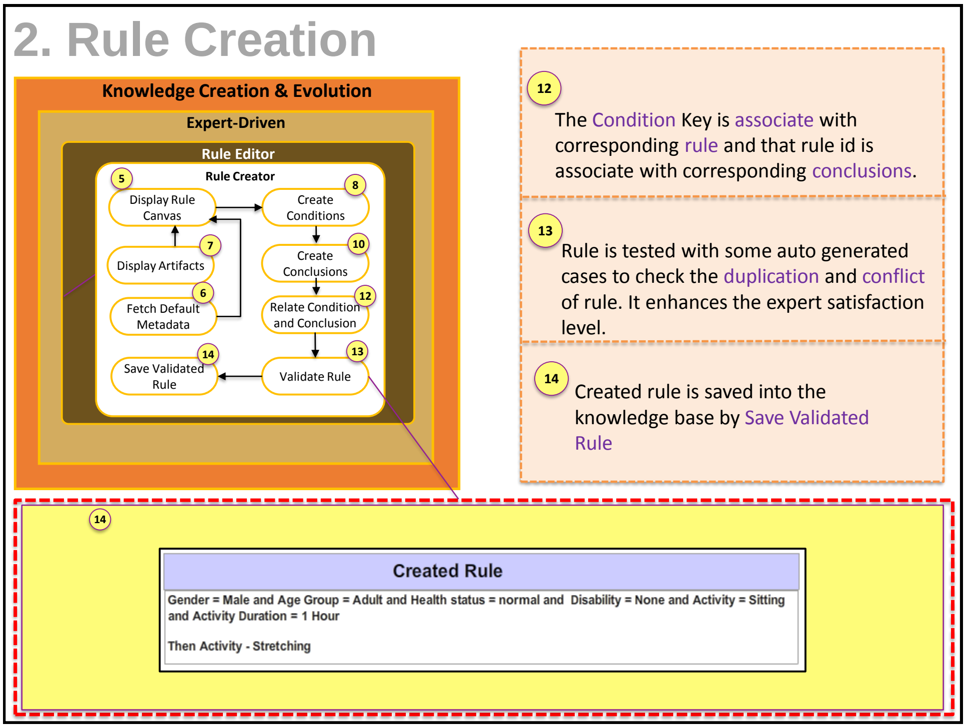
13 Rule is tested with some auto generated cases to check the duplication and conflict of rule. It enhances the expert satisfaction level.

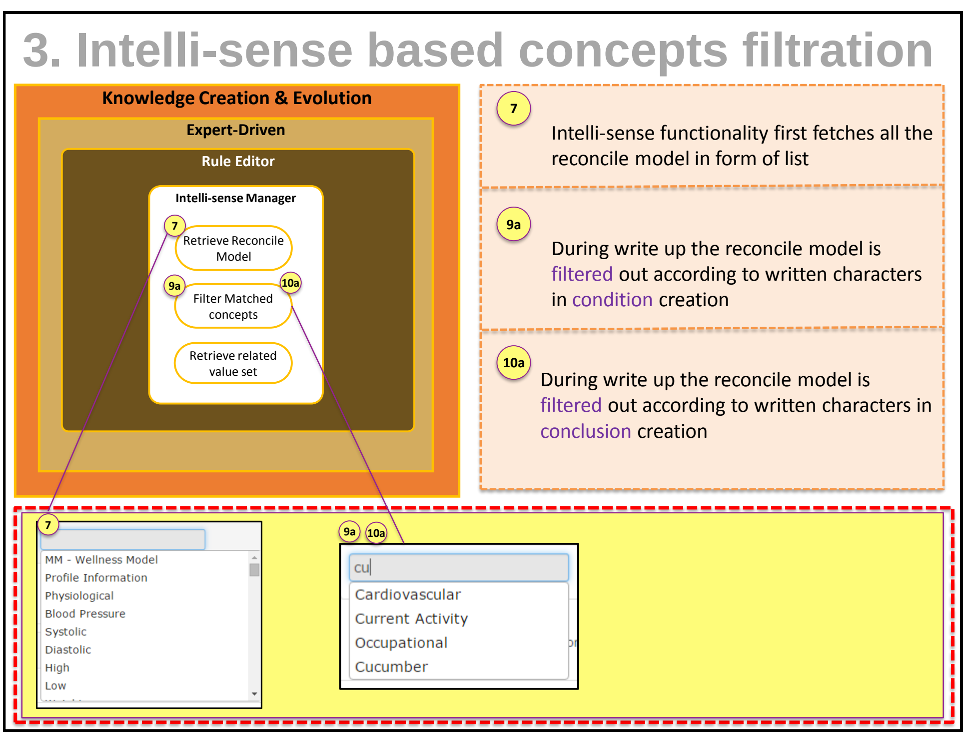
13 Auto-generated Test Cases

Gender	Age	Health Status	Disability	Activity	Activity Duration	Select
Male	42	normal	None	Sitting	1 Hour	☑
Male	45	normal	None	Sitting	1 Hour	☐
Male	40	normal	None	Sitting	1 Hour	☐
Male	42	normal	None	Sitting	1 Hour	☐

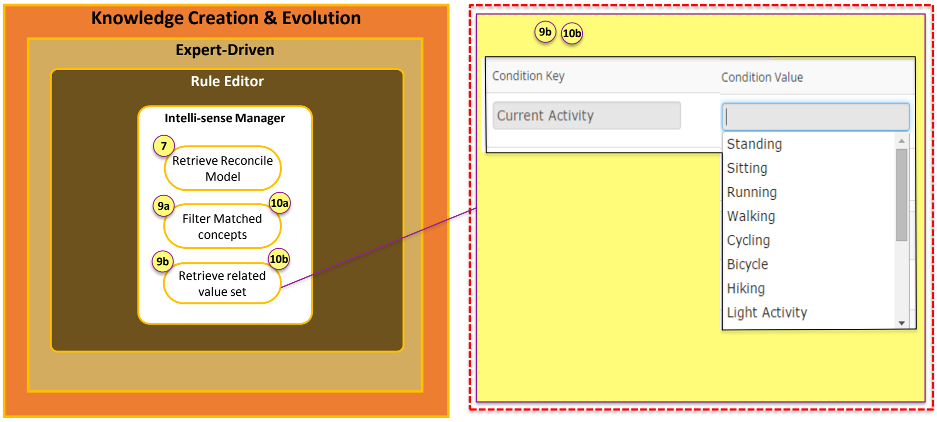
Add new test case | Execute selected case

The Created Rule is successfully executed for the selected case. (No duplication and No Conflict)


2. Rule Creation
Knowledge Creation & Evolution
Expert-Driven
Rule Editor
Rule Creator
5
Display Rule Canvas
8
Create Conditions
7
Display Artifacts
10
Create Conclusions
6
Fetch Default Metadata
12
Relate Condition and Conclusion
14
Save Validated Rule
13
Validate Rule
12
The Condition Key is associate with corresponding rule and that rule id is associate with corresponding conclusions.
13
Rule is tested with some auto generated cases to check the duplication and conflict of rule. It enhances the expert satisfaction level.
14
Created rule is saved into the knowledge base by Save Validated Rule
14
Created Rule
Gender = Male and Age Group = Adult and Health status = normal and Disability = None and Activity = Sitting and Activity Duration = 1 Hour
Then Activity - Stretching


3. Intelli-sense based concepts filtration
Knowledge Creation & Evolution
Expert-Driven
Rule Editor
Intelli-sense Manager
7
Retrieve Reconcile Model
9a
Filter Matched concepts
10a
Retrieve related value set
7
Intelli-sense functionality first fetches all the reconcile model in form of list
9a
During write up the reconcile model is filtered out according to written characters in condition creation
10a
During write up the reconcile model is filtered out according to written characters in conclusion creation
7
MM - Wellness Model
Profile Information
Physiological
Blood Pressure
Systolic
Diastolic
High
Low
9a
10a
cu
Cardiovascular
Current Activity
Occupational
Cucumber

3. Intelli-sense based concepts filtration



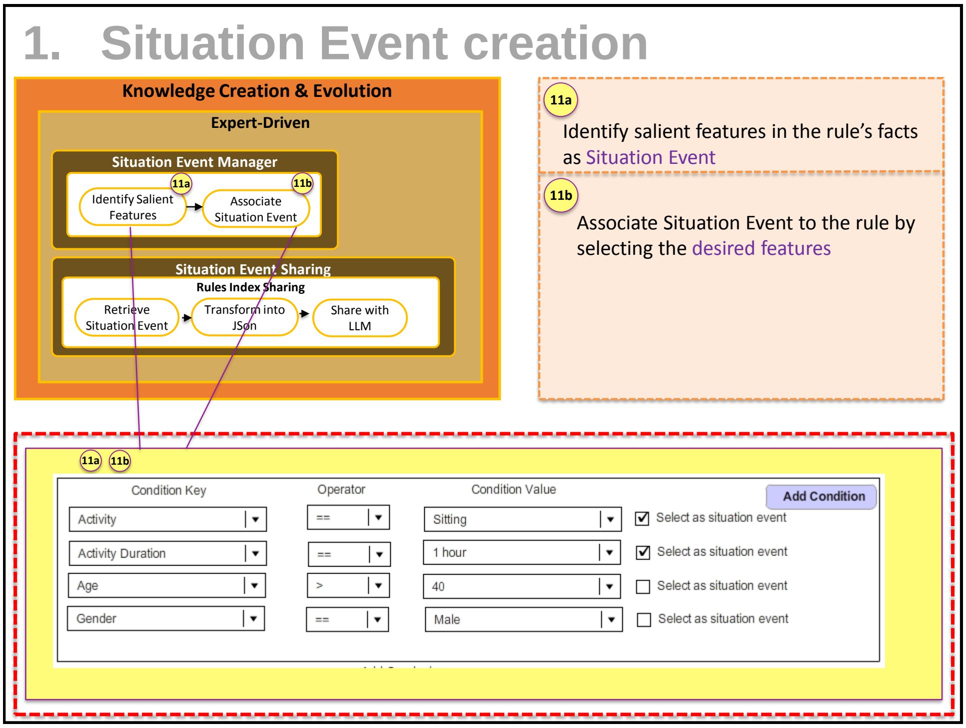
- 9a During write up the reconcile model is filtered out according to written characters in condition
- 10a During write up the reconcile model is filtered out according to written characters in conclusion
- 9b Retrieve all possible values set of selected concept in condition
- 10b Retrieve all possible values set of selected concept in conclusion

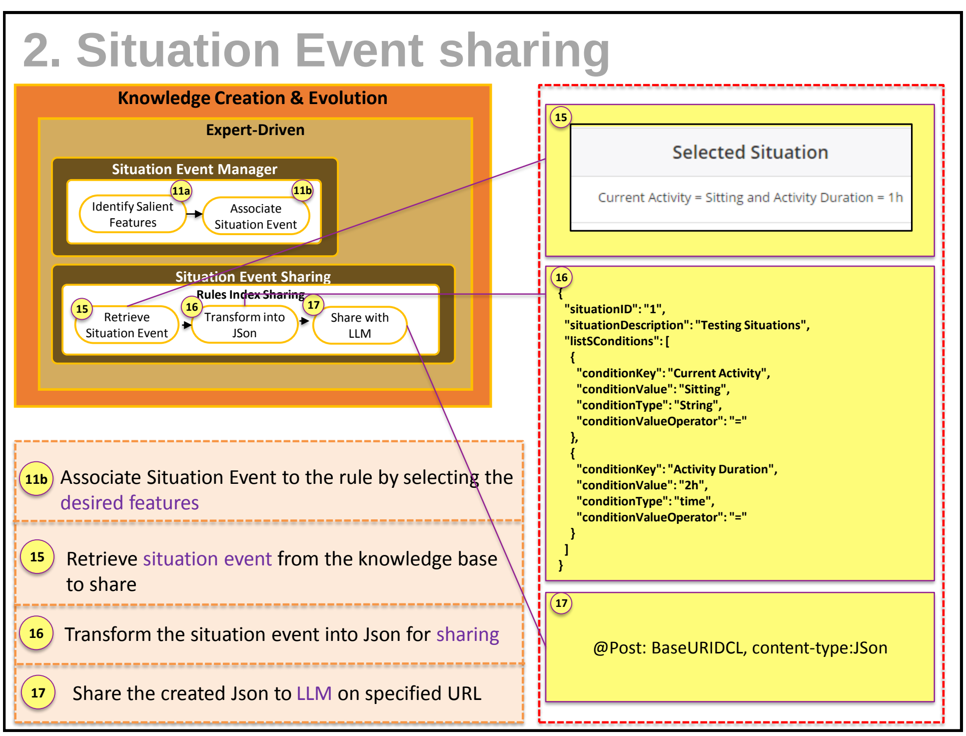
Scenario 2: Situation Event creation and sharing

Desc. : I-KAT **situation event** association with rule and **sharing** with LLM (DCL).

Major Steps:

1. Situation Event Creation
2. Situation Event Sharing


1. Situation Event creation
Knowledge Creation & Evolution
Expert-Driven
Situation Event Manager
11a
11b
Identify Salient Features
Associate Situation Event
Situation Event Sharing
Rules Index Sharing
Retrieve Situation Event
Transform into JSon
Share with LLM
11a
Identify salient features in the rule's facts as Situation Event
11b
Associate Situation Event to the rule by selecting the desired features
11a 11b
Condition Key
Operator
Condition Value
Add Condition
Activity
==
Sitting
Select as situation event
Activity Duration
==
1 hour
Select as situation event
Age
>
40
Select as situation event
Gender
==
Male
Select as situation event


2. Situation Event sharing
Knowledge Creation & Evolution
Expert-Driven
Situation Event Manager
11a
11b
Identify Salient Features
Associate Situation Event
Situation Event Sharing
Rules Index Sharing
15
Retrieve Situation Event
16
Transform into JSon
17
Share with LLM
11b
Associate Situation Event to the rule by selecting the desired features
15
Retrieve situation event from the knowledge base to share
16
Transform the situation event into Json for sharing
17
Share the created Json to LLM on specified URL
15
Selected Situation
Current Activity = Sitting and Activity Duration = 1h
16
{
"situationID": "1",
"situationDescription": "Testing Situations",
"listSConditions": [
{
"conditionKey": "Current Activity",
"conditionValue": "Sitting",
"conditionType": "String",
"conditionValueOperator": "="
},
{
"conditionKey": "Activity Duration",
"conditionValue": "2h",
"conditionType": "time",
"conditionValueOperator": "="
}
]
}
17
@Post: BaseURIDCL, content-type:JSon

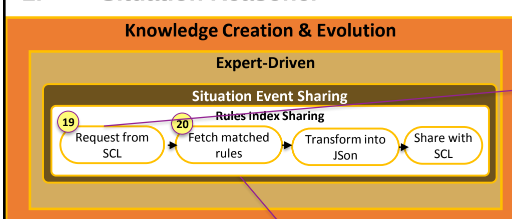
Scenario 3: Rule sharing for recommendation

Desc. : I-KAT **rule sharing** with SCL for **recommendation** generation based on received **situation event**

Major Steps:

1. Situation Event received from SCL
2. Retrieved matched rules (Situation Reasoning)
3. Share matched rules with SCL

1. Situation Event received from SCL
2. Situation Reasoner



19 Request from Service Curation Layer to get all matched rules for situation occurring

20 The situation reasoner fetches all matched rules from knowledge base

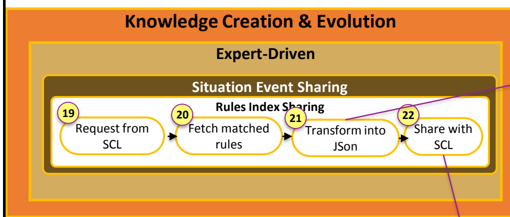
19


```
{
 "situationID": "1",
 "situationDescription": "Testing Situations",
 "listSConditions": [
  {
   "conditionKey": "Current Activity",
   "conditionValue": "Sitting",
   "conditionType": "String",
   "conditionValueOperator": "="
  },
  {
   "conditionKey": "Activity Duration",
   "conditionValue": "2h",
   "conditionType": "time",
   "conditionValueOperator": "="
  }
 ]
}
```


20


```
*************************************************
INFO : - Rule-ID: 20103
INFO : - Rule-ID: 20102
INFO : - Rules loaded successfully:[2016/02/04 17:41:46]: 2
INFO : - Time of Execution: 24 millisecond
INFO : -
*************************************************
```

3. Shares matched rules



21

```
[
 {
  "ruleConclusion": "Take a five-minute walk.",
  "conclusionList": [
   {
    "conclusionKey": "Current Activity",
    "conclusionValue": "Walking",
    "conclusionOperator": "=",
    "ruleID": 20103,
    "conclusionID": 10149
   }
  ],
  "conditionList": [
   {
    "conditionKey": "Current Activity",
    "conditionValue": "Sitting",
    "conditionType": "String",
    "conditionValueOperator": "=",
    "isItSituation": false,
    "conditionID": 1
   },
   {
    "conditionKey": "Activity Duration",
    "conditionValue": "2h",
    "conditionType": null,
    "conditionValueOperator": "=",
    "isItSituation": false,
    "conditionID": 20163
   }
  ],
  "ruleID": 20103
 }
]
```


22 @Post: BaseURISCL, content-type:JSon

19 Request from Service Curation Layer to get all matched rules for situation occurring

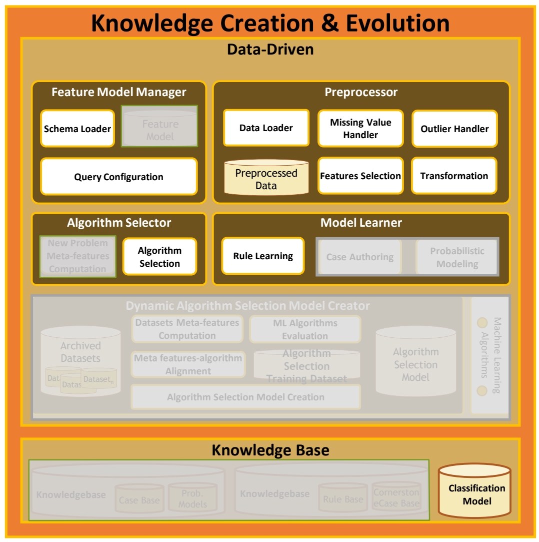
20 The situation reasoner fetches all matched rules from knowledge base

21 Transform the matched rules into Json for sharing

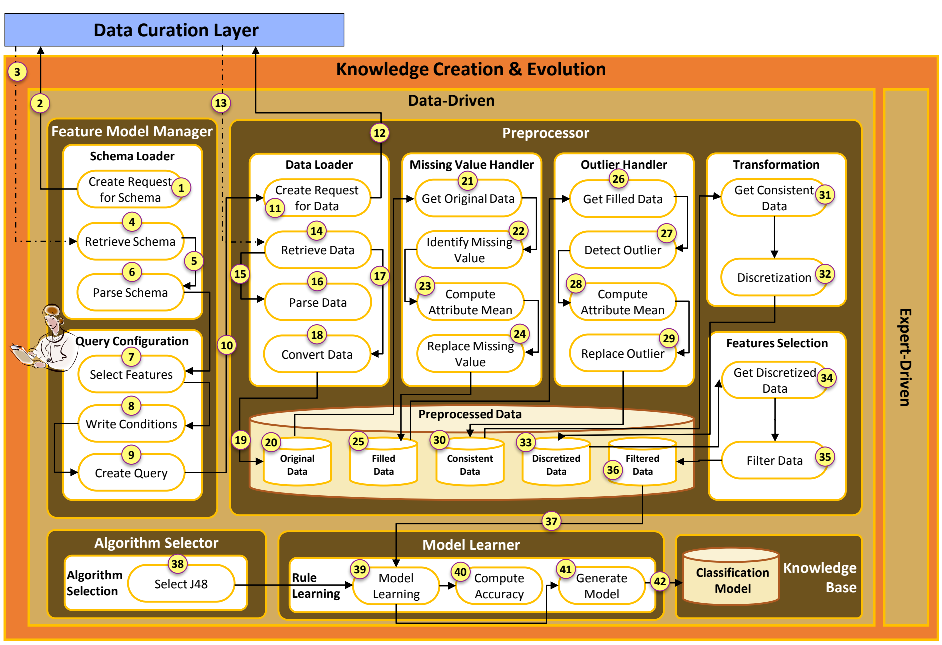
22 Share the created Json to SCL on specified URL

STENCILS for Data-Driven Micro-Demo

KCL Data Driven Detailed Architecture



KCL Data Driven Detailed Architecture



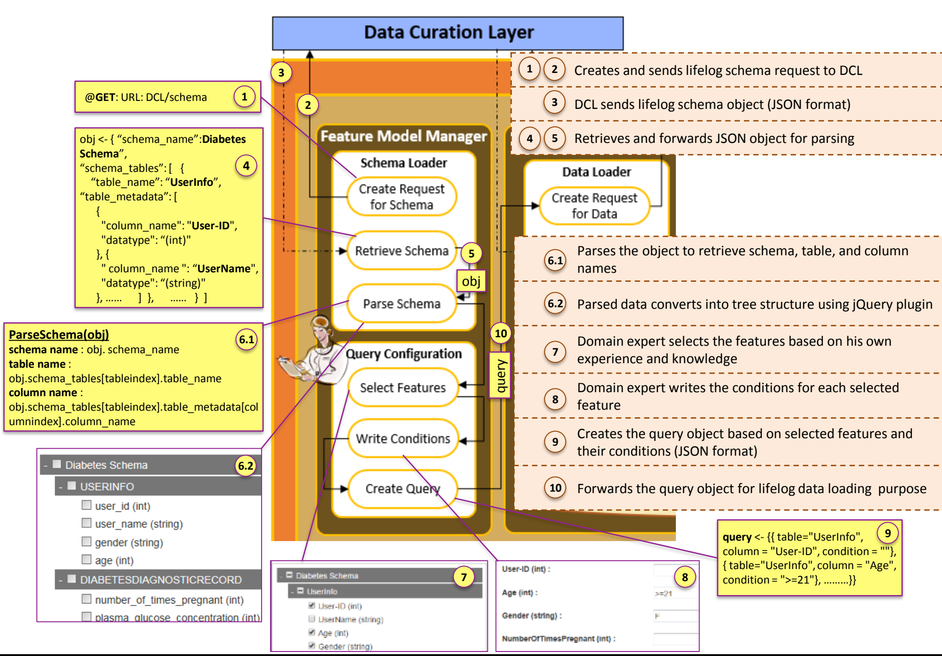
Scenario: Classification Model Creation

Desc. : Lifelog **schema** and **Data** is **acquired** from DCL. **Preprocessed** the data to generate the **classification model** using **J48** algorithm.

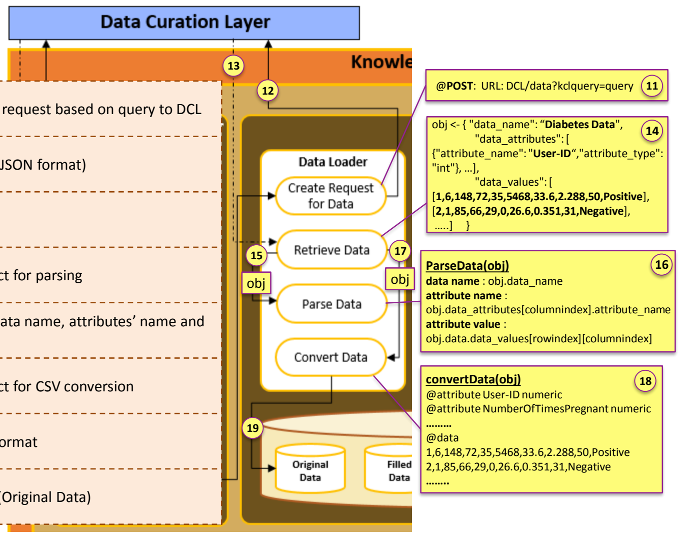
Major Steps:

1. Data Selection and Retrieval Process
2. Data Preprocessing
3. Model Learning Process

1 - Data Selection and Retrieval Process (1/2)

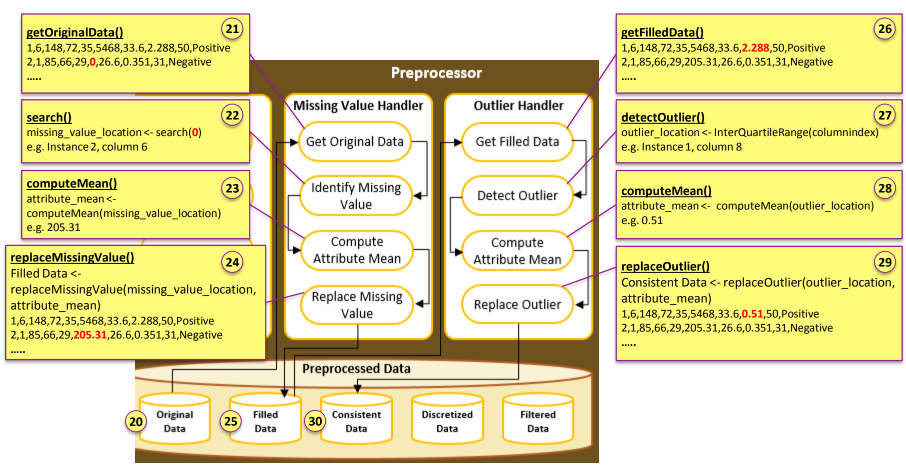


1 - Data Selection and Retrieval Process (2/2)



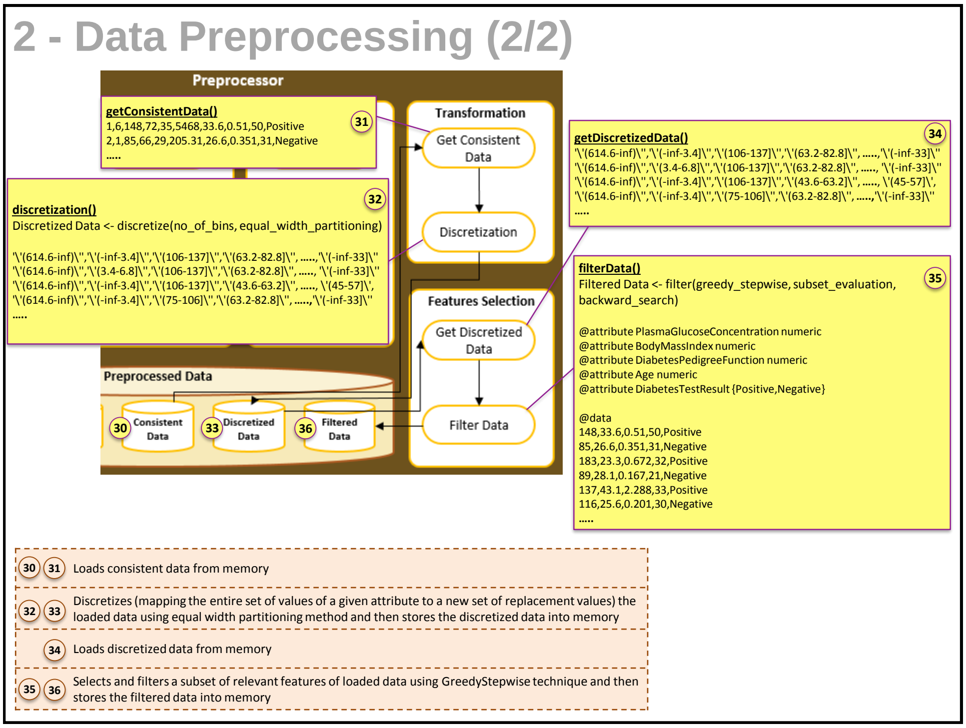
- (11) (12) Creates and sends lifelog data request based on query to DCL
- (13) DCL sends lifelog data object (JSON format)
- (14) Retrieves JSON object
- (15) Forwards retrieved JSON object for parsing
- (16) Parses the object to retrieve data name, attributes' name and value
- (17) Forwards retrieved JSON object for CSV conversion
- (18) Converts the object into CSV format
- (19) Persists the data into CSV file (Original Data)

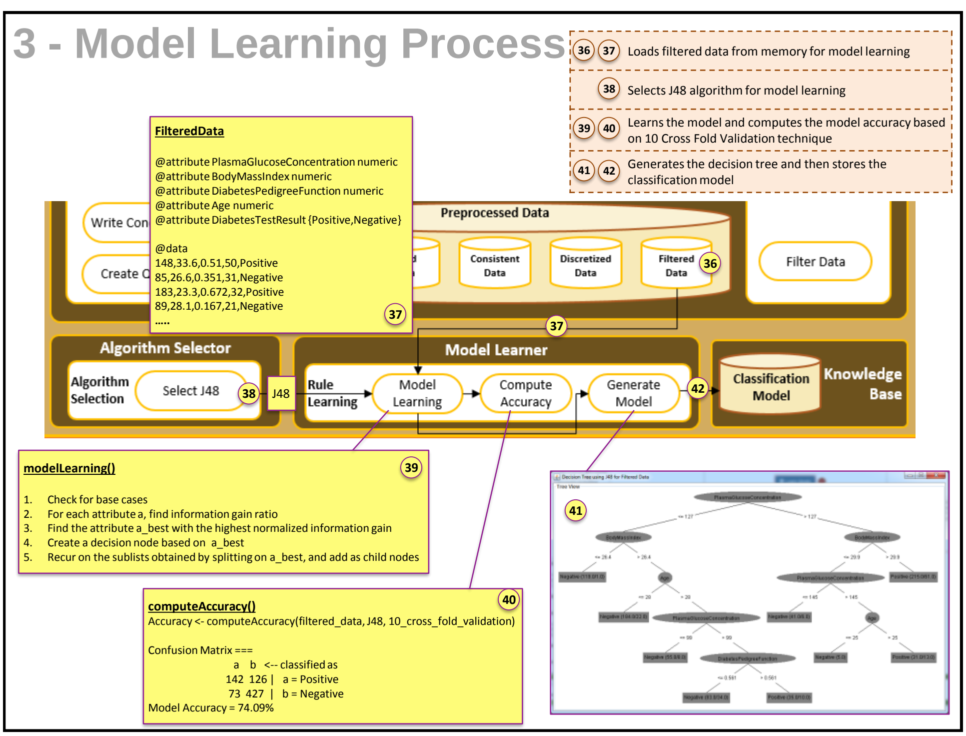
2 - Data Preprocessing (1/2)



- (20) (21) Loads original data from CSV file
- (22) Identifies and records the locations of missing values (i.e. 0) from original data
- (23) Computes the mean of all attributes having missing values
- (24) (25) Replaces each missing value with its corresponding attribute mean and then stores the filled data into memory

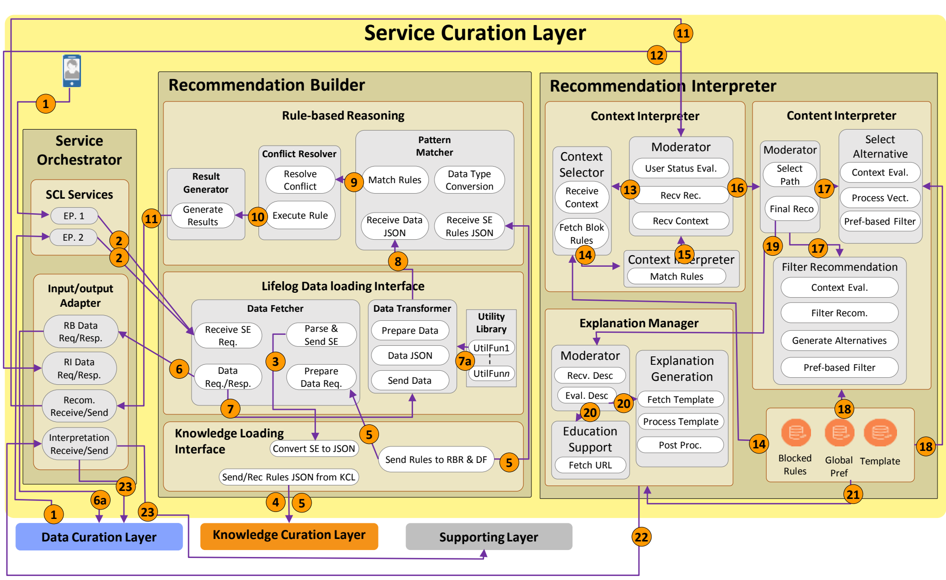
- (26) Loads filled data from memory
- (27) Detects the outliers using InterQuartileRange technique and records the locations of outliers (an observation which deviates so much from the other observations) from filled data
- (28) Computes the mean of all attributes having outliers
- (29) (30) Replaces each outlier with its corresponding attribute mean and then stores the consistent data into memory


2 - Data Preprocessing (2/2)
Preprocessor
getConsistentData()
1,6,148,72,35,5468,33.6,0.51,50,Positive
2,1,85,66,29,205.31,26.6,0.351,31,Negative
.....
31
Transformation
Get Consistent Data
discretization()
Discretized Data <- discretize(no_of_bins, equal_width_partitioning)
'\'(614.6-inf)\'','\'(-inf-3.4]\'','\'(106-137]\'','\'(63.2-82.8]\'', ...,'\'(-inf-33]\''
'\'(614.6-inf)\'','\'(3.4-6.8]\'','\'(106-137]\'','\'(63.2-82.8]\'', ..., '\'(-inf-33]\''
'\'(614.6-inf)\'','\'(-inf-3.4]\'','\'(106-137]\'','\'(43.6-63.2]\'', ..., '\'(45-57]\',
'\'(614.6-inf)\'','\'(-inf-3.4]\'','\'(75-106]\'','\'(63.2-82.8]\'', ...,'\'(-inf-33]\''
.....
32
Discretization
getDiscretizedData()
'\'(614.6-inf)\'','\'(-inf-3.4]\'','\'(106-137]\'','\'(63.2-82.8]\'', ...,'\'(-inf-33]\''
'\'(614.6-inf)\'','\'(3.4-6.8]\'','\'(106-137]\'','\'(63.2-82.8]\'', ..., '\'(-inf-33]\''
'\'(614.6-inf)\'','\'(-inf-3.4]\'','\'(106-137]\'','\'(43.6-63.2]\'', ..., '\'(45-57]\',
'\'(614.6-inf)\'','\'(-inf-3.4]\'','\'(75-106]\'','\'(63.2-82.8]\'', ...,'\'(-inf-33]\''
.....
34
Features Selection
Get Discretized Data
filterData()
Filtered Data <- filter(greedy_stepwise, subset_evaluation, backward_search)
@attribute PlasmaGlucoseConcentration numeric
@attribute BodyMassIndex numeric
@attribute DiabetesPedigreeFunction numeric
@attribute Age numeric
@attribute DiabetesTestResult {Positive,Negative}
@data
148,33.6,0.51,50,Positive
85,26.6,0.351,31,Negative
183,23.3,0.672,32,Positive
89,28.1,0.167,21,Negative
137,43.1,2.288,33,Positive
116,25.6,0.201,30,Negative
.....
35
Preprocessed Data
30 Consistent Data
33 Discretized Data
36 Filtered Data
Filter Data
30 31 Loads consistent data from memory
32 33 Discretizes (mapping the entire set of values of a given attribute to a new set of replacement values) the loaded data using equal width partitioning method and then stores the discretized data into memory
34 Loads discretized data from memory
35 36 Selects and filters a subset of relevant features of loaded data using GreedyStepwise technique and then stores the filtered data into memory


3 - Model Learning Process
36 37 Loads filtered data from memory for model learning
38 Selects J48 algorithm for model learning
39 40 Learns the model and computes the model accuracy based on 10 Cross Fold Validation technique
41 42 Generates the decision tree and then stores the classification model
FilteredData
@attribute PlasmaGlucoseConcentration numeric
@attribute BodyMassIndex numeric
@attribute DiabetesPedigreeFunction numeric
@attribute Age numeric
@attribute DiabetesTestResult {Positive,Negative}
@data
148,33.6,0.51,50,Positive
85,26.6,0.351,31,Negative
183,23.3,0.672,32,Positive
89,28.1,0.167,21,Negative
.....
37
Write Con
Create C
Preprocessed Data
Consistent Data
Discretized Data
Filtered Data
36
Filter Data
37
Algorithm Selector
Algorithm Selection
Select J48
38
J48
Model Learner
Rule Learning
Model Learning
Compute Accuracy
Generate Model
42
Classification Model
Knowledge Base
modelLearning()
39
1. Check for base cases
2. For each attribute a, find information gain ratio
3. Find the attribute a_best with the highest normalized information gain
4. Create a decision node based on a_best
5. Recur on the sublists obtained by splitting on a_best, and add as child nodes
computeAccuracy()
40
Accuracy <- computeAccuracy(filtered_data, J48, 10_cross_fold_validation)
Confusion Matrix ===
a b <-- classified as
142 126 | a = Positive
73 427 | b = Negative
Model Accuracy = 74.09%
41

STENCILS for SCL MMV2.0 Micro-Demo

SCL Detailed Architecture



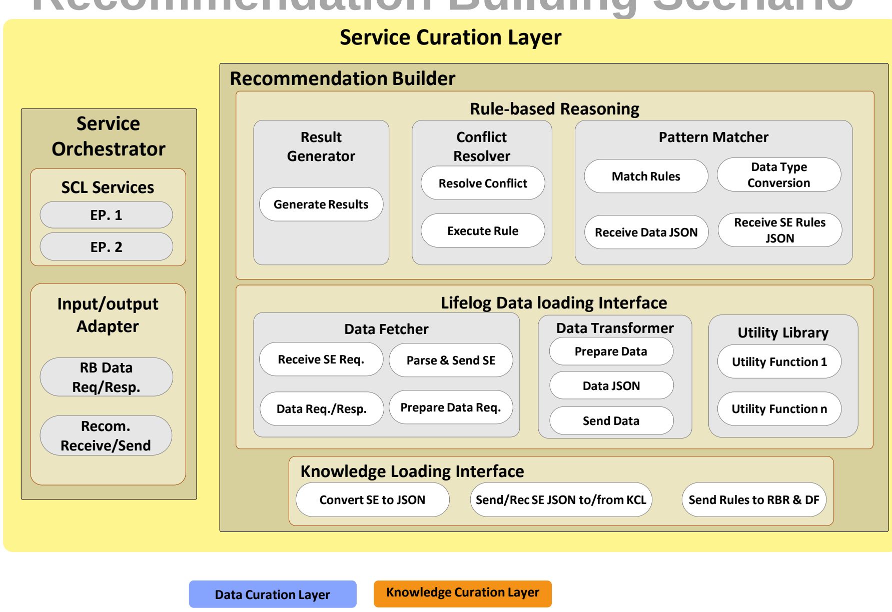
Service Curation Layer - Scenarios

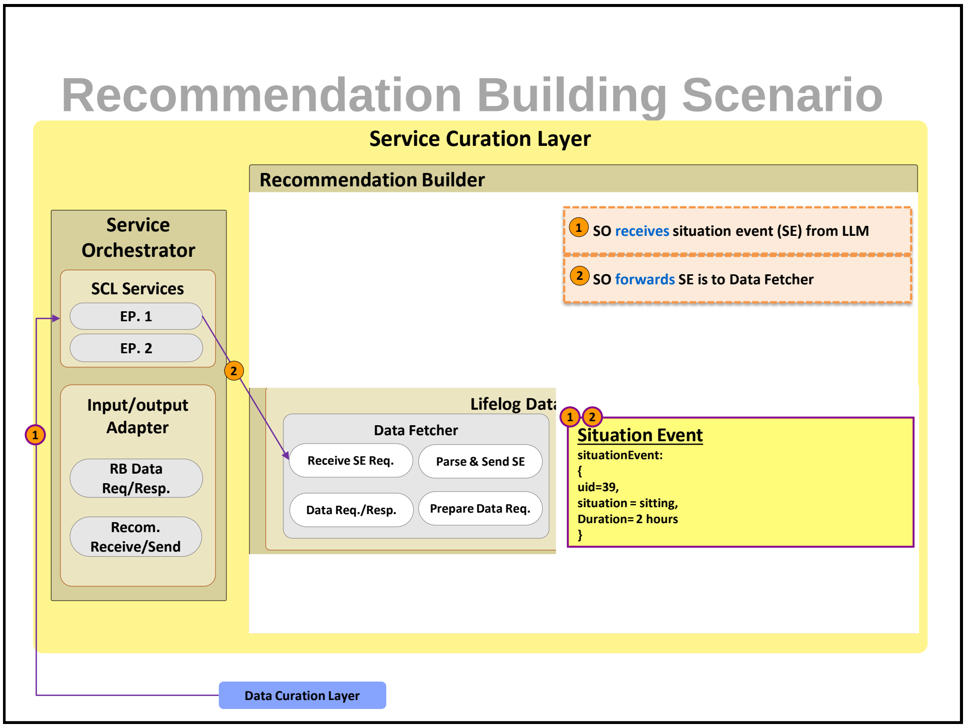
Scenario 1: Recommendation Building

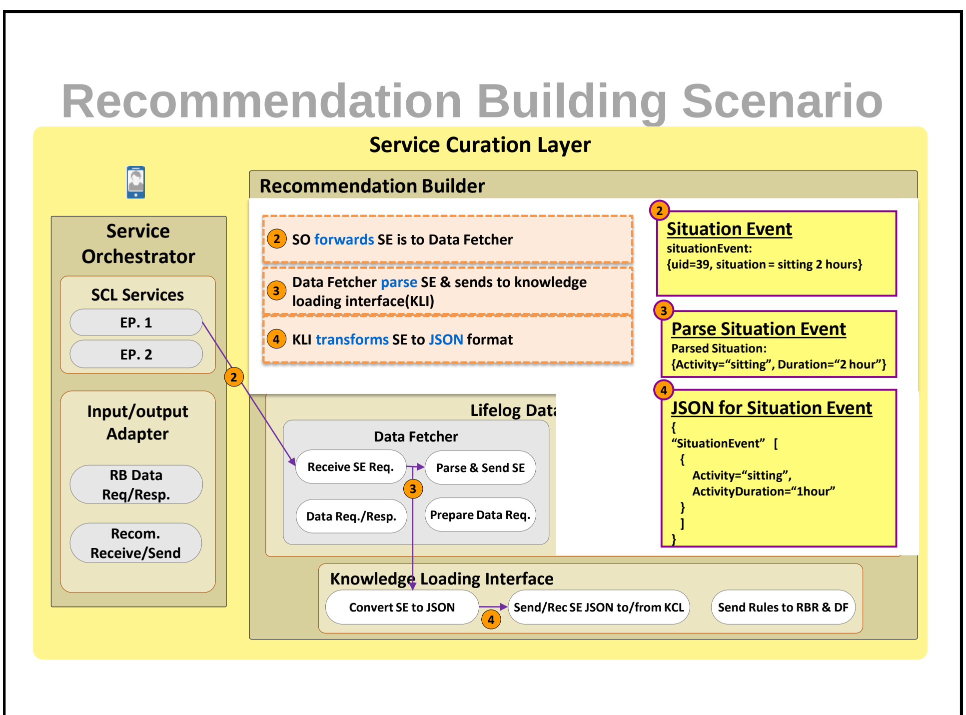
Desc : Generation of Recommendations for the Situations, triggered by LLM.

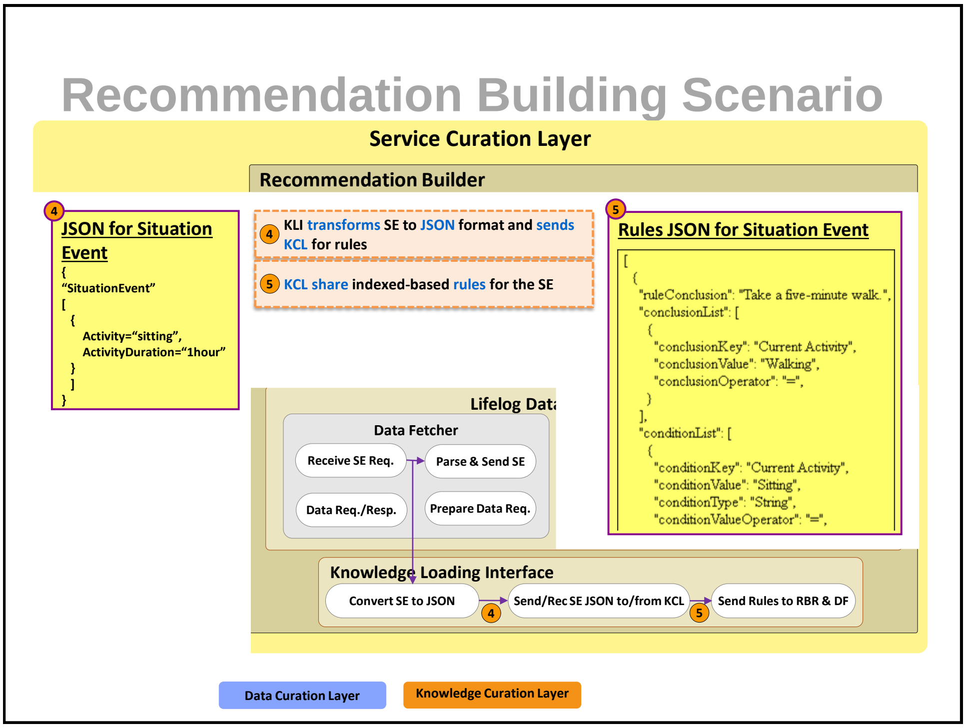
1. Situation Event (SE) received from LLM to Service Orchestrator (SO).
2. SO forwards the situation to Recommendation Builder(RB).
3. RB builds the recommendations Based on rules retrieved from KCL and life-log/user profile data from DCL.
4. RB sends the recommendation back to SO.
5. SO forwards the recommendation to Recommendation Interpreter.

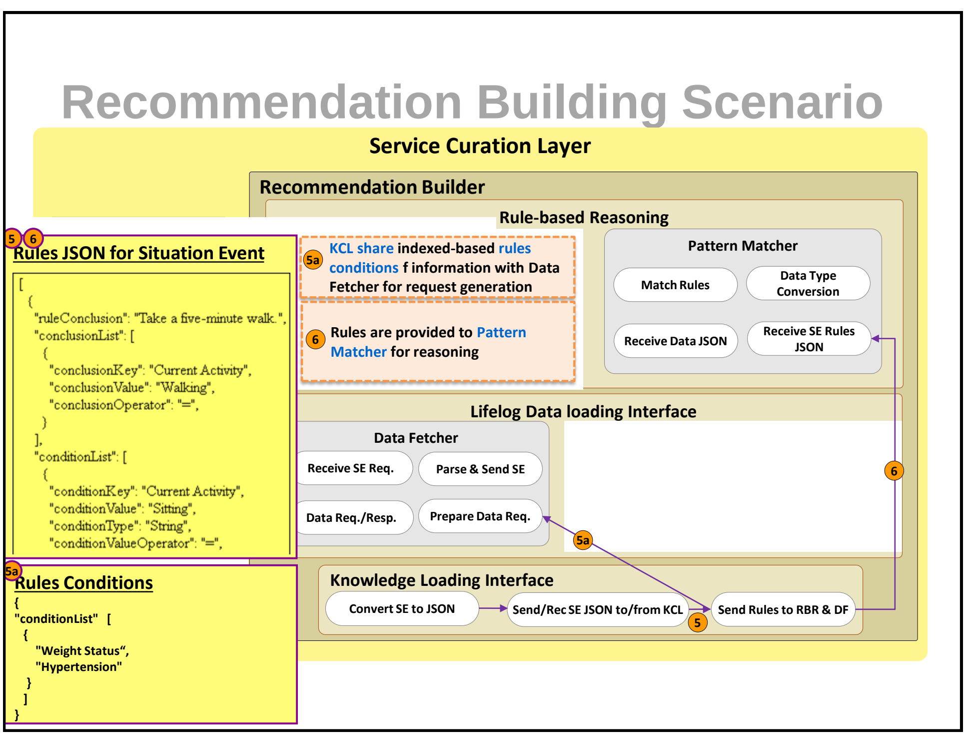
Recommendation Building Scenario

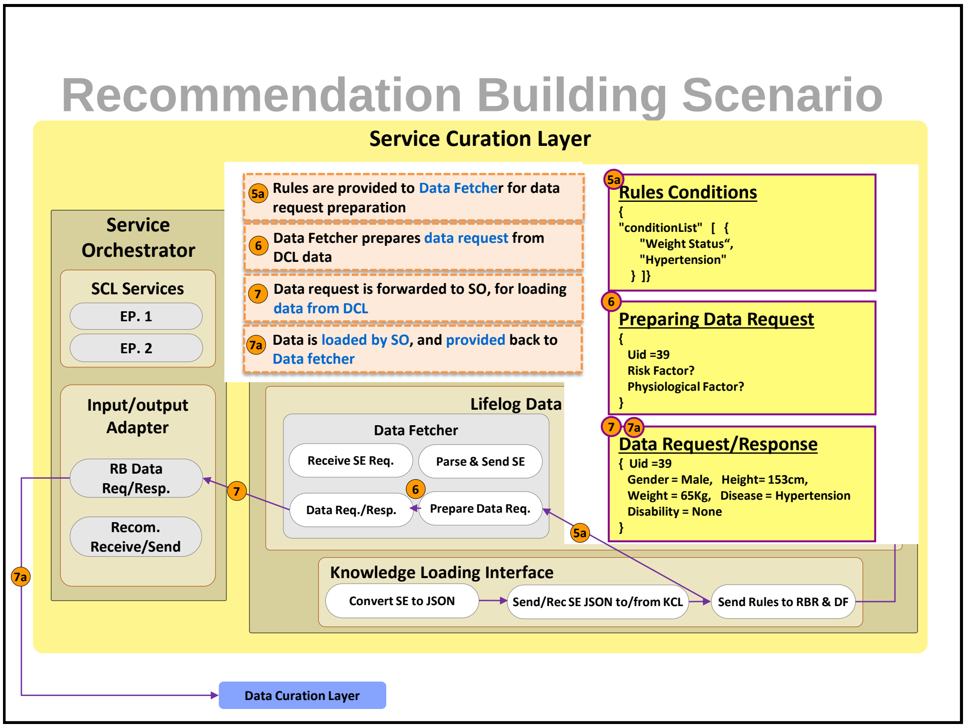


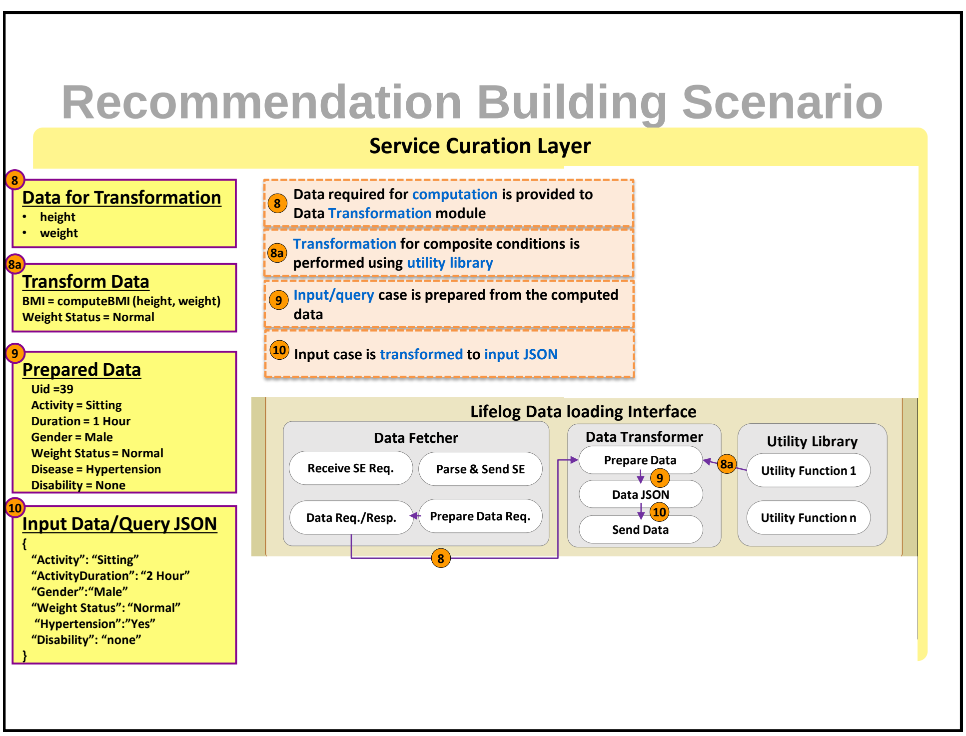

Recommendation Building Scenario
Service Curation Layer
Recommendation Builder
Service Orchestrator
SCL Services
EP. 1
EP. 2
Input/output Adapter
RB Data Req/Resp.
Recom. Receive/Send
1 SO receives situation event (SE) from LLM
2 SO forwards SE is to Data Fetcher
Lifelog Data
Data Fetcher
Receive SE Req.
Parse & Send SE
Data Req./Resp.
Prepare Data Req.
Situation Event
situationEvent:
{
uid=39,
situation = sitting,
Duration= 2 hours
}
Data Curation Layer

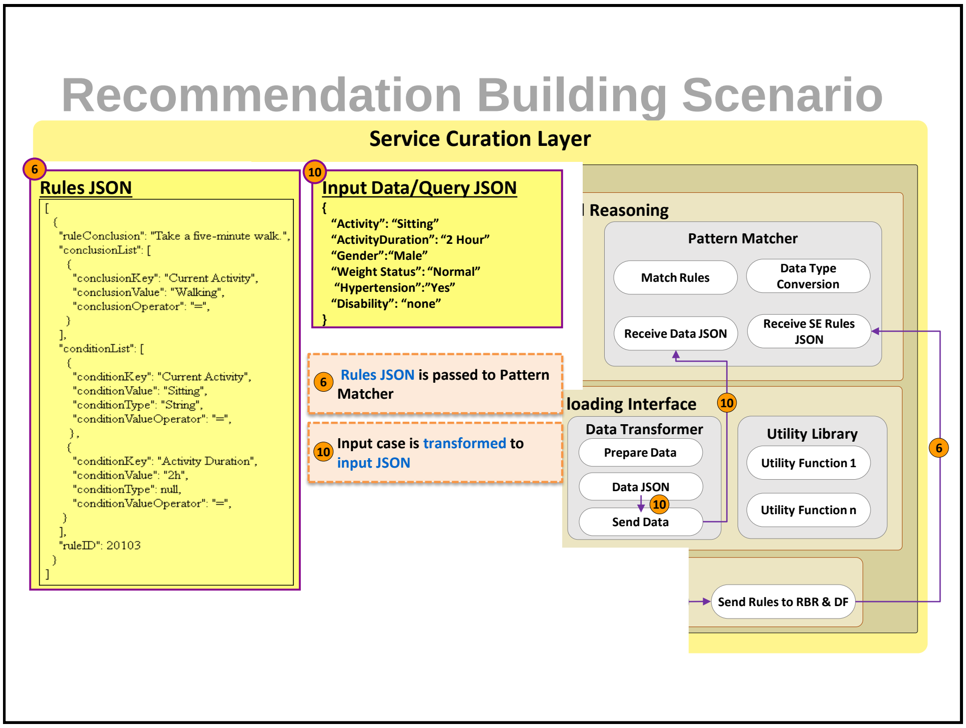

Recommendation Building Scenario
Service Curation Layer
Recommendation Builder
Service Orchestrator
SCL Services
EP. 1
EP. 2
Input/output Adapter
RB Data Req/Resp.
Recom. Receive/Send
2 SO forwards SE is to Data Fetcher
3 Data Fetcher parse SE & sends to knowledge loading interface(KLI)
4 KLI transforms SE to JSON format
Situation Event
situationEvent:
{uid=39, situation = sitting 2 hours}
Parse Situation Event
Parsed Situation:
{Activity="sitting", Duration="2 hour"}
JSON for Situation Event
{
"SituationEvent" [
{
Activity="sitting",
ActivityDuration="1hour"
}
]
}
Lifelog Data
Data Fetcher
Receive SE Req.
Parse & Send SE
Data Req./Resp.
Prepare Data Req.
Knowledge Loading Interface
Convert SE to JSON
Send/Rec SE JSON to/from KCL
Send Rules to RBR & DF


Recommendation Building Scenario
Service Curation Layer
Recommendation Builder
4
JSON for Situation Event
{
"SituationEvent"
[
{
Activity="sitting",
ActivityDuration="1hour"
}
]
}
4 KLI transforms SE to JSON format and sends KCL for rules
5 KCL share indexed-based rules for the SE
5
Rules JSON for Situation Event
[
{
"ruleConclusion": "Take a five-minute walk.",
"conclusionList": [
{
"conclusionKey": "Current Activity",
"conclusionValue": "Walking",
"conclusionOperator": "=",
}
],
"conditionList": [
{
"conditionKey": "Current Activity",
"conditionValue": "Sitting",
"conditionType": "String",
"conditionValueOperator": "=",
Data Fetcher
Receive SE Req.
Parse & Send SE
Data Req./Resp.
Prepare Data Req.
Knowledge Loading Interface
Convert SE to JSON
4
Send/Rec SE JSON to/from KCL
5
Send Rules to RBR & DF
Data Curation Layer
Knowledge Curation Layer

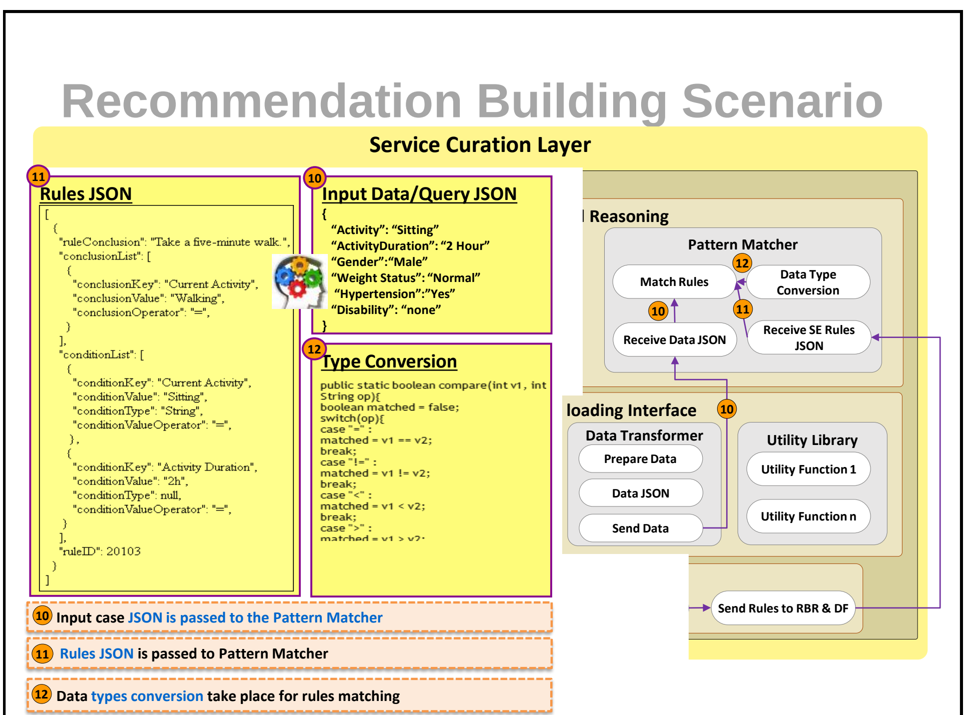

Recommendation Building Scenario
Service Curation Layer
Recommendation Builder
Rule-based Reasoning
5 6
Rules JSON for Situation Event
[
{
"ruleConclusion": "Take a five-minute walk.",
"conclusionList": [
{
"conclusionKey": "Current Activity",
"conclusionValue": "Walking",
"conclusionOperator": "=",
}
],
"conditionList": [
{
"conditionKey": "Current Activity",
"conditionValue": "Sitting",
"conditionType": "String",
"conditionValueOperator": "=",
5a KCL share indexed-based rules conditions f information with Data Fetcher for request generation
6 Rules are provided to Pattern Matcher for reasoning
Pattern Matcher
Match Rules
Data Type Conversion
Receive Data JSON
Receive SE Rules JSON
Lifelog Data loading Interface
Data Fetcher
Receive SE Req.
Parse & Send SE
Data Req./Resp.
Prepare Data Req.
5a
6
5a
Rules Conditions
{
"conditionList" [
{
"Weight Status",
"Hypertension"
}
]
}
Knowledge Loading Interface
Convert SE to JSON
Send/Rec SE JSON to/from KCL
5
Send Rules to RBR & DF


Recommendation Building Scenario
Service Curation Layer
Service Orchestrator
SCL Services
EP. 1
EP. 2
Input/output Adapter
RB Data Req/Resp.
Recom. Receive/Send
5a Rules are provided to Data Fetcher for data request preparation
6 Data Fetcher prepares data request from DCL data
7 Data request is forwarded to SO, for loading data from DCL
7a Data is loaded by SO, and provided back to Data fetcher
5a Rules Conditions
{
"conditionList" [{
"Weight Status",
"Hypertension"
}]}
6 Preparing Data Request
{
Uid =39
Risk Factor?
Physiological Factor?
}
7 7a Data Request/Response
{ Uid =39
Gender = Male, Height= 153cm,
Weight = 65Kg, Disease = Hypertension
Disability = None
}
Lifelog Data
Data Fetcher
Receive SE Req.
Parse & Send SE
Data Req./Resp.
Prepare Data Req.
Knowledge Loading Interface
Convert SE to JSON
Send/Rec SE JSON to/from KCL
Send Rules to RBR & DF
Data Curation Layer

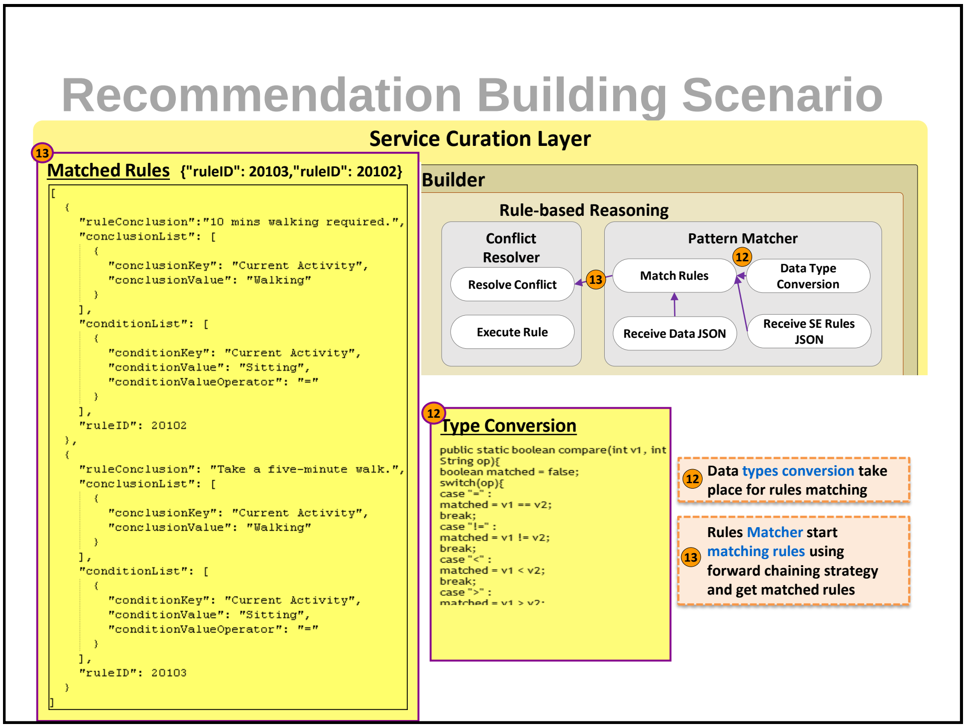

Recommendation Building Scenario
Service Curation Layer
8 Data for Transformation
• height
• weight
8a Transform Data
BMI = computeBMI (height, weight)
Weight Status = Normal
9 Prepared Data
Uid =39
Activity = Sitting
Duration = 1 Hour
Gender = Male
Weight Status = Normal
Disease = Hypertension
Disability = None
10 Input Data/Query JSON
{
"Activity": "Sitting"
"ActivityDuration": "2 Hour"
"Gender":"Male"
"Weight Status": "Normal"
"Hypertension":"Yes"
"Disability": "none"
}
8 Data required for computation is provided to Data Transformation module
8a Transformation for composite conditions is performed using utility library
9 Input/query case is prepared from the computed data
10 Input case is transformed to input JSON
Lifelog Data loading Interface
Data Fetcher
Receive SE Req.
Parse & Send SE
Data Req./Resp.
Prepare Data Req.
Data Transformer
Prepare Data
Data JSON
Send Data
Utility Library
Utility Function 1
Utility Function n



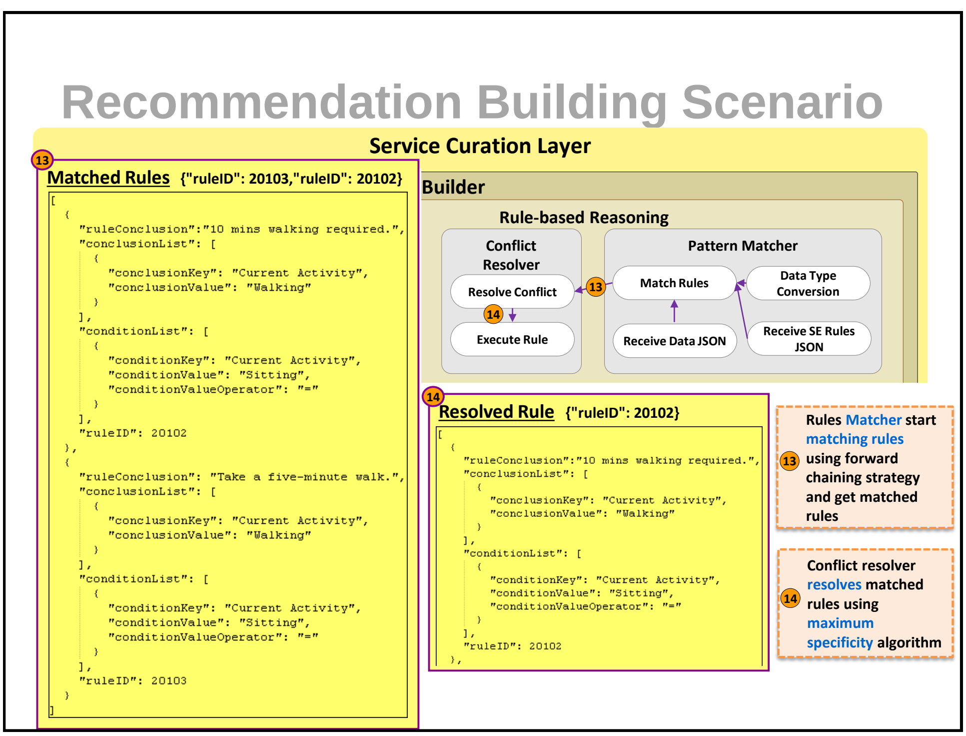
Recommendation Building Scenario
Service Curation Layer
6
Rules JSON
[
{
"ruleConclusion": "Take a five-minute walk.",
"conclusionList": [
{
"conclusionKey": "Current Activity",
"conclusionValue": "Walking",
"conclusionOperator": "=",
}
],
"conditionList": [
{
"conditionKey": "Current Activity",
"conditionValue": "Sitting",
"conditionType": "String",
"conditionValueOperator": "=",
},
{
"conditionKey": "Activity Duration",
"conditionValue": "2h",
"conditionType": null,
"conditionValueOperator": "=",
}
],
"ruleID": 20103
}
]
10
Input Data/Query JSON
{
"Activity": "Sitting"
"ActivityDuration": "2 Hour"
"Gender":"Male"
"Weight Status":"Normal"
"Hypertension":"Yes"
"Disability": "none"
}
6 Rules JSON is passed to Pattern Matcher
10 Input case is transformed to input JSON
Reasoning
Pattern Matcher
Match Rules
Data Type Conversion
Receive Data JSON
Receive SE Rules JSON
loading Interface
10
Data Transformer
Prepare Data
Data JSON
10
Send Data
Utility Library
Utility Function 1
Utility Function n
6
Send Rules to RBR & DF



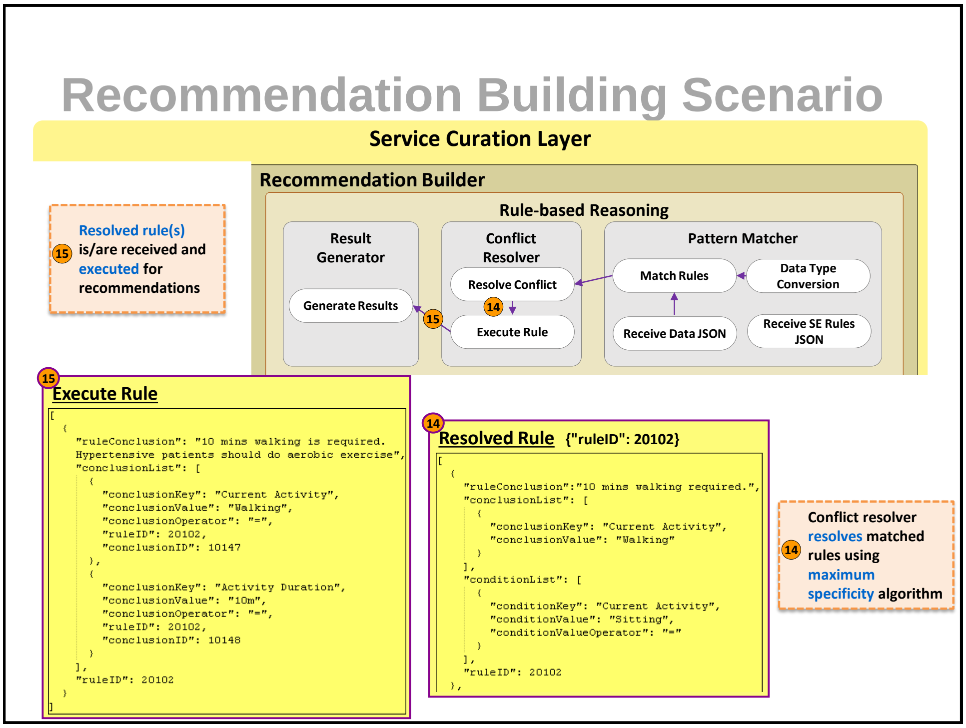
Recommendation Building Scenario
Service Curation Layer
11
Rules JSON
[
{
"ruleConclusion": "Take a five-minute walk.",
"conclusionList": [
{
"conclusionKey": "Current Activity",
"conclusionValue": "Walking",
"conclusionOperator": "=",
}
],
"conditionList": [
{
"conditionKey": "Current Activity",
"conditionValue": "Sitting",
"conditionType": "String",
"conditionValueOperator": "=",
},
{
"conditionKey": "Activity Duration",
"conditionValue": "2h",
"conditionType": null,
"conditionValueOperator": "=",
}
],
"ruleID": 20103
}
]
10
Input Data/Query JSON
{
"Activity": "Sitting"
"ActivityDuration": "2 Hour"
"Gender":"Male"
"Weight Status":"Normal"
"Hypertension":"Yes"
"Disability": "none"
}
12
Type Conversion
public static boolean compare(int v1 , int String op){
boolean matched = false;
switch(op){
case "=" :
matched = v1 == v2;
break;
case "!=" :
matched = v1 != v2;
break;
case "<" :
matched = v1 < v2;
break;
case ">" :
Reasoning
Pattern Matcher
12
Match Rules
Data Type Conversion
10
11
Receive Data JSON
Receive SE Rules JSON
loading Interface
10
Data Transformer
Prepare Data
Data JSON
Send Data
Utility Library
Utility Function 1
Utility Function n
Send Rules to RBR & DF
10 Input case JSON is passed to the Pattern Matcher
11 Rules JSON is passed to Pattern Matcher
12 Data types conversion take place for rules matching

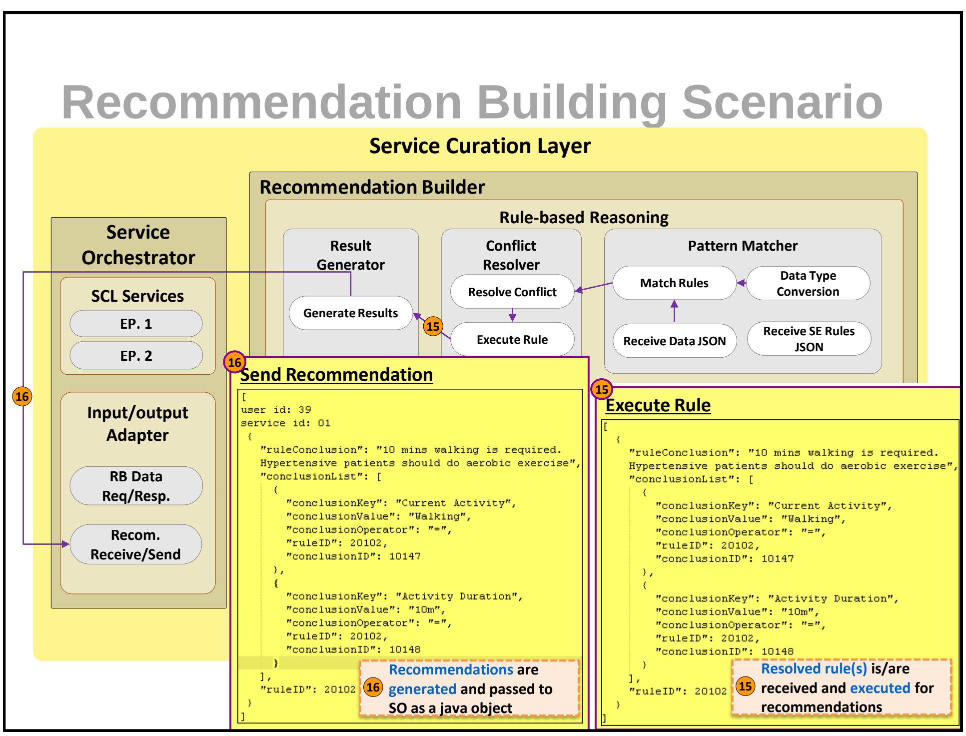


Recommendation Building Scenario
Service Curation Layer
13
Matched Rules {"ruleID": 20103,"ruleID": 20102}
[
{
"ruleConclusion":"10 mins walking required.",
"conclusionList": [
{
"conclusionKey": "Current Activity",
"conclusionValue": "Walking"
}
],
"conditionList": [
{
"conditionKey": "Current Activity",
"conditionValue": "Sitting",
"conditionValueOperator": "="
}
],
"ruleID": 20102
},
{
"ruleConclusion": "Take a five-minute walk.",
"conclusionList": [
{
"conclusionKey": "Current Activity",
"conclusionValue": "Walking"
}
],
"conditionList": [
{
"conditionKey": "Current Activity",
"conditionValue": "Sitting",
"conditionValueOperator": "="
}
],
"ruleID": 20103
}
]
Builder
Rule-based Reasoning
Conflict Resolver
Resolve Conflict
Execute Rule
Pattern Matcher
Match Rules
Data Type Conversion
Receive Data JSON
Receive SE Rules JSON
12
13
12
Type Conversion
public static boolean compare(int v1, int
String op){
boolean matched = false;
switch(op){
case "=" :
matched = v1 == v2;
break;
case "!=" :
matched = v1 != v2;
break;
case "<" :
matched = v1 < v2;
break;
case ">" :
12
Data types conversion take place for rules matching
13
Rules Matcher start matching rules using forward chaining strategy and get matched rules



Recommendation Building Scenario
Service Curation Layer
13
Matched Rules {"ruleID": 20103,"ruleID": 20102}
[
{
"ruleConclusion":"10 mins walking required.",
"conclusionList": [
{
"conclusionKey": "Current Activity",
"conclusionValue": "Walking"
}
],
"conditionList": [
{
"conditionKey": "Current Activity",
"conditionValue": "Sitting",
"conditionValueOperator": "="
}
],
"ruleID": 20102
},
{
"ruleConclusion": "Take a five-minute walk.",
"conclusionList": [
{
"conclusionKey": "Current Activity",
"conclusionValue": "Walking"
}
],
"conditionList": [
{
"conditionKey": "Current Activity",
"conditionValue": "Sitting",
"conditionValueOperator": "="
}
],
"ruleID": 20103
}
]
Builder
Rule-based Reasoning
Conflict Resolver
Resolve Conflict
14
Execute Rule
Pattern Matcher
Match Rules
13
Data Type Conversion
Receive Data JSON
Receive SE Rules JSON
14
Resolved Rule {"ruleID": 20102}
[
{
"ruleConclusion":"10 mins walking required.",
"conclusionList": [
{
"conclusionKey": "Current Activity",
"conclusionValue": "Walking"
}
],
"conditionList": [
{
"conditionKey": "Current Activity",
"conditionValue": "Sitting",
"conditionValueOperator": "="
}
],
"ruleID": 20102
},
13
Rules Matcher start matching rules using forward chaining strategy and get matched rules
14
Conflict resolver resolves matched rules using maximum specificity algorithm


Recommendation Building Scenario
Service Curation Layer
Recommendation Builder
Rule-based Reasoning
Result Generator
Generate Results
Conflict Resolver
Resolve Conflict
Execute Rule
Pattern Matcher
Match Rules
Data Type Conversion
Receive Data JSON
Receive SE Rules JSON
15
Resolved rule(s) is/are received and executed for recommendations
14
15
15
Execute Rule
[{ "ruleConclusion": "10 mins walking is required. Hypertensive patients should do aerobic exercise", "conclusionList": [{ "conclusionKey": "Current Activity", "conclusionValue": "Walking", "conclusionOperator": "=", "ruleID": 20102, "conclusionID": 10147 }, { "conclusionKey": "Activity Duration", "conclusionValue": "10m", "conclusionOperator": "=", "ruleID": 20102, "conclusionID": 10148 }], "ruleID": 20102 }]
14
Resolved Rule {"ruleID": 20102}
[{ "ruleConclusion":"10 mins walking required.", "conclusionList": [{ "conclusionKey": "Current Activity", "conclusionValue": "Walking" }], "conditionList": [{ "conditionKey": "Current Activity", "conditionValue": "Sitting", "conditionValueOperator": "=" }], "ruleID": 20102 },
14
Conflict resolver resolves matched rules using maximum specificity algorithm


Recommendation Building Scenario
Service Curation Layer
Recommendation Builder
Rule-based Reasoning
Service Orchestrator
SCL Services
EP. 1
EP. 2
Input/output Adapter
RB Data Req/Resp.
Recom. Receive/Send
16
Result Generator
Generate Results
Conflict Resolver
Resolve Conflict
Execute Rule
15
Pattern Matcher
Match Rules
Data Type Conversion
Receive Data JSON
Receive SE Rules JSON
16
Send Recommendation
[user id: 39 service id: 01 { "ruleConclusion": "10 mins walking is required. Hypertensive patients should do aerobic exercise", "conclusionList": [{ "conclusionKey": "Current Activity", "conclusionValue": "Walking", "conclusionOperator": "=", "ruleID": 20102, "conclusionID": 10147 }, { "conclusionKey": "Activity Duration", "conclusionValue": "10m", "conclusionOperator": "=", "ruleID": 20102, "conclusionID": 10148 }], "ruleID": 20102 }]
16
Recommendations are generated and passed to SO as a java object
15
Execute Rule
[{ "ruleConclusion": "10 mins walking is required. Hypertensive patients should do aerobic exercise", "conclusionList": [{ "conclusionKey": "Current Activity", "conclusionValue": "Walking", "conclusionOperator": "=", "ruleID": 20102, "conclusionID": 10147 }, { "conclusionKey": "Activity Duration", "conclusionValue": "10m", "conclusionOperator": "=", "ruleID": 20102, "conclusionID": 10148 }], "ruleID": 20102 }]
15
Resolved rule(s) is/are received and executed for recommendations

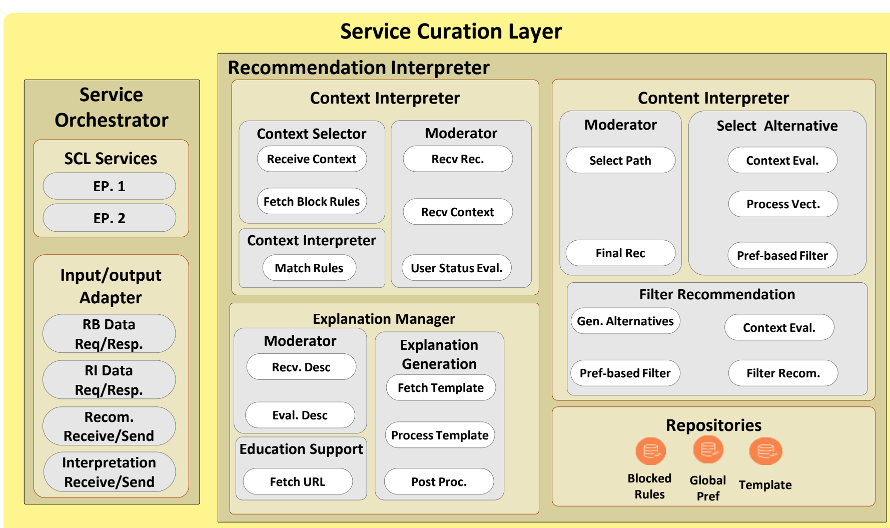
Recommendation Interpreter Scenario

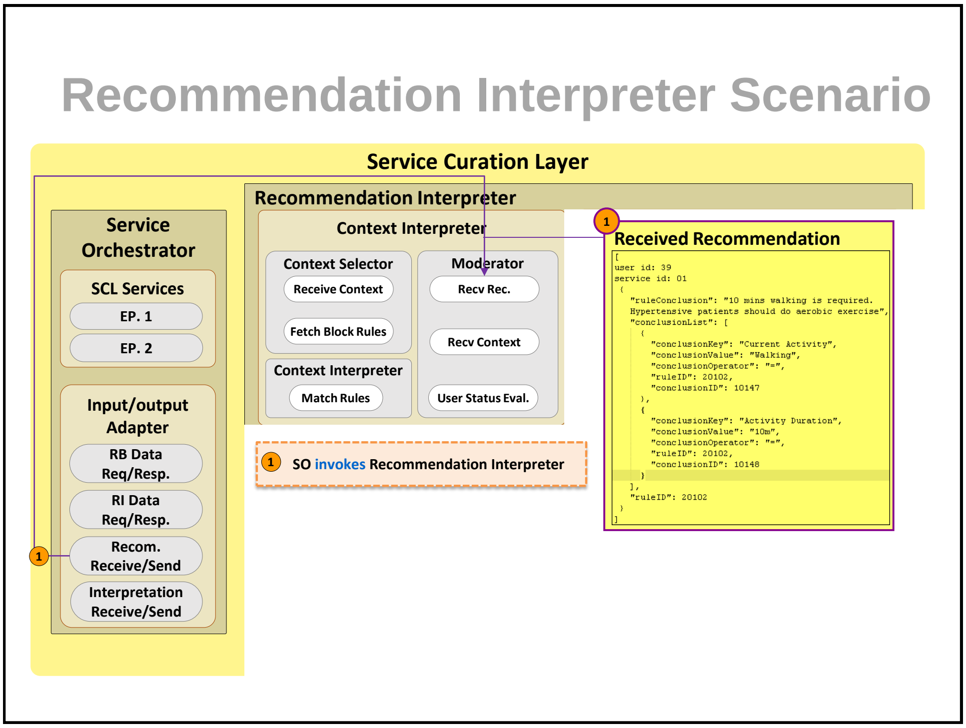
Scenario 2: Recommendation Interpreting

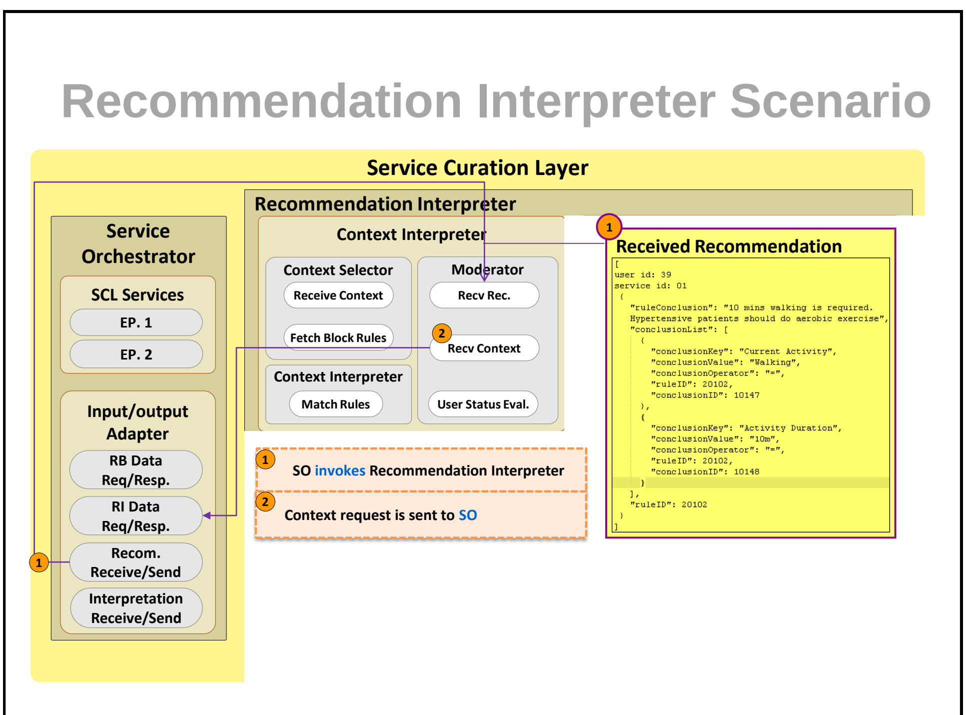
Desc : **Interpretation of Recommendations for Personalization.**

1. Recommendation received from Service Orchestrator (SO) to Recommendation Interpreter (RI).
2. RI interprets the recommendations Based on the location, high level, and weather context retrieved from DCL.
3. RI sends the personalized recommendation back to SO.
4. SO forwards the personalized recommendation to DCL for persistence and SL for presentation.

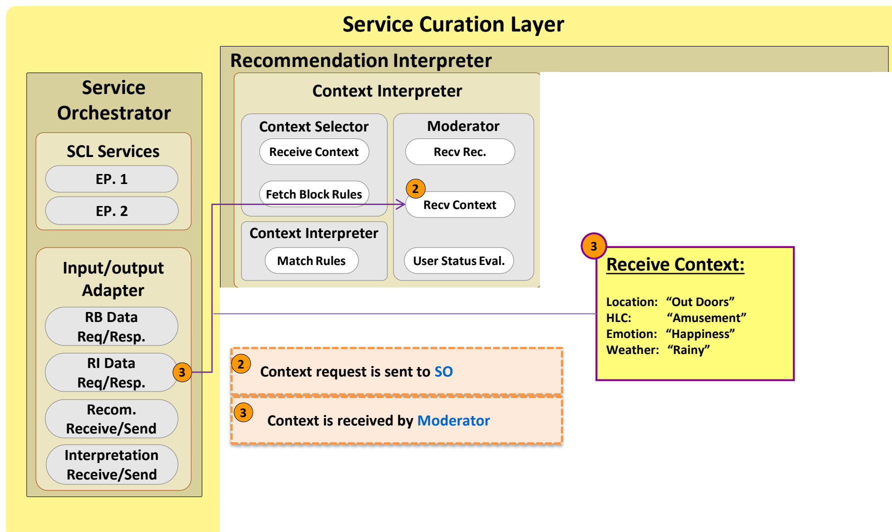
Recommendation Interpreter Scenario



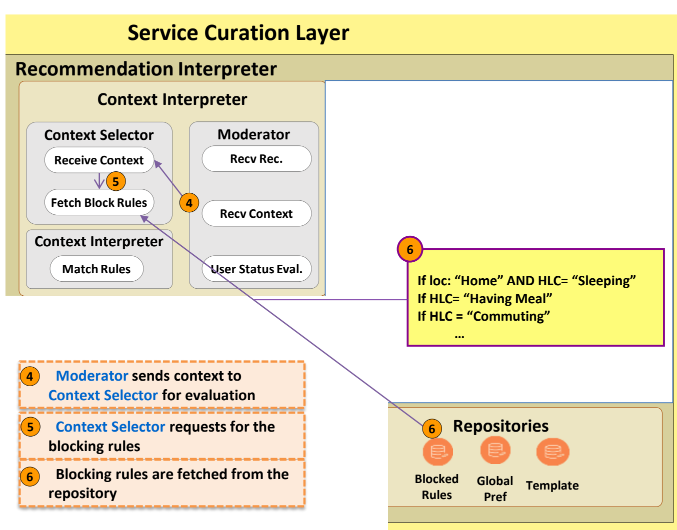

Recommendation Interpreter Scenario
Service Curation Layer
Service Orchestrator
SCL Services
EP. 1
EP. 2
Input/output Adapter
RB Data Req/Resp.
RI Data Req/Resp.
Recom. Receive/Send
Interpretation Receive/Send
Recommendation Interpreter
Context Interpreter
Context Selector
Receive Context
Fetch Block Rules
Context Interpreter
Match Rules
Moderator
Recv Rec.
Recv Context
User Status Eval.
SO invokes Recommendation Interpreter
Received Recommendation
[
user id: 39
service id: 01
{
"ruleConclusion": "10 mins walking is required. Hypertensive patients should do aerobic exercise",
"conclusionList": [
{
"conclusionKey": "Current Activity",
"conclusionValue": "Walking",
"conclusionOperator": "=",
"ruleID": 20102,
"conclusionID": 10147
},
{
"conclusionKey": "Activity Duration",
"conclusionValue": "10m",
"conclusionOperator": "=",
"ruleID": 20102,
"conclusionID": 10148
}
],
"ruleID": 20102
}
]

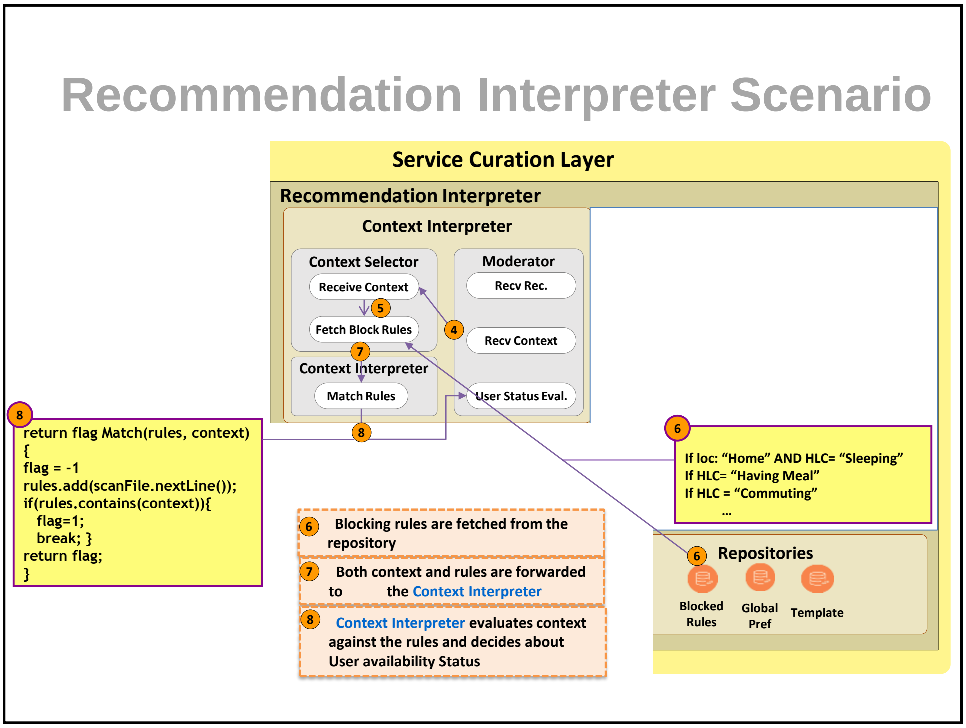

Recommendation Interpreter Scenario
Service Curation Layer
Service Orchestrator
SCL Services
EP. 1
EP. 2
Input/output Adapter
RB Data Req/Resp.
RI Data Req/Resp.
Recom. Receive/Send
Interpretation Receive/Send
Recommendation Interpreter
Context Interpreter
Context Selector
Receive Context
Fetch Block Rules
Context Interpreter
Match Rules
Moderator
Recv Rec.
Recv Context
User Status Eval.
SO invokes Recommendation Interpreter
Context request is sent to SO
Received Recommendation
[
user id: 39
service id: 01
{
"ruleConclusion": "10 mins walking is required. Hypertensive patients should do aerobic exercise",
"conclusionList": [
{
"conclusionKey": "Current Activity",
"conclusionValue": "Walking",
"conclusionOperator": "=",
"ruleID": 20102,
"conclusionID": 10147
},
{
"conclusionKey": "Activity Duration",
"conclusionValue": "10m",
"conclusionOperator": "=",
"ruleID": 20102,
"conclusionID": 10148
}
],
"ruleID": 20102
}
]

Recommendation Interpreter Scenario

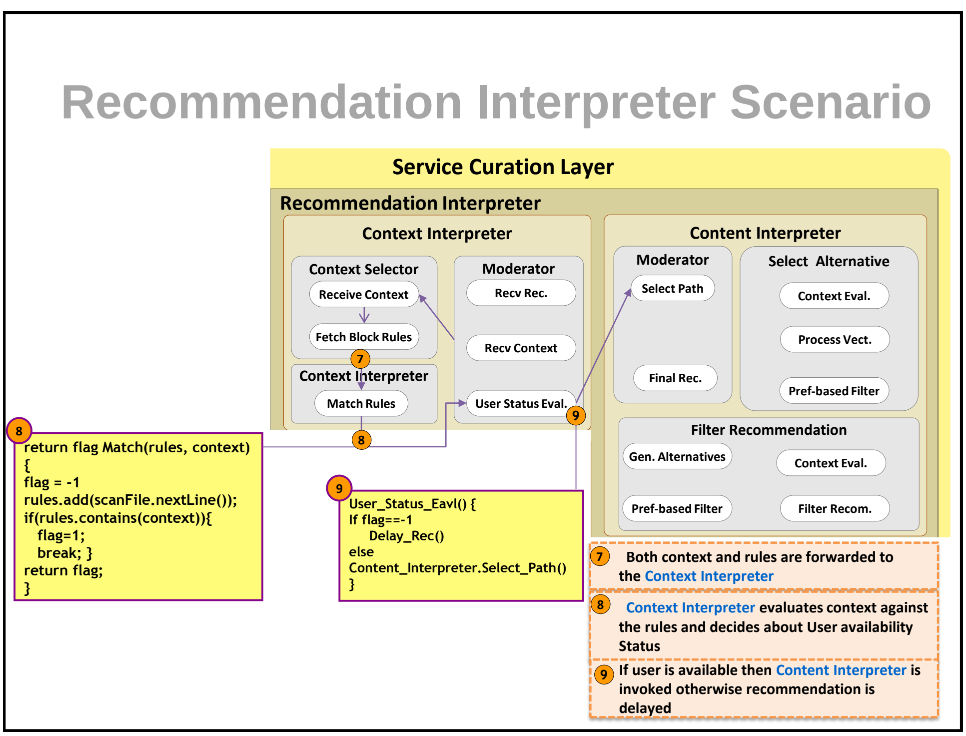


Recommendation Interpreter Scenario



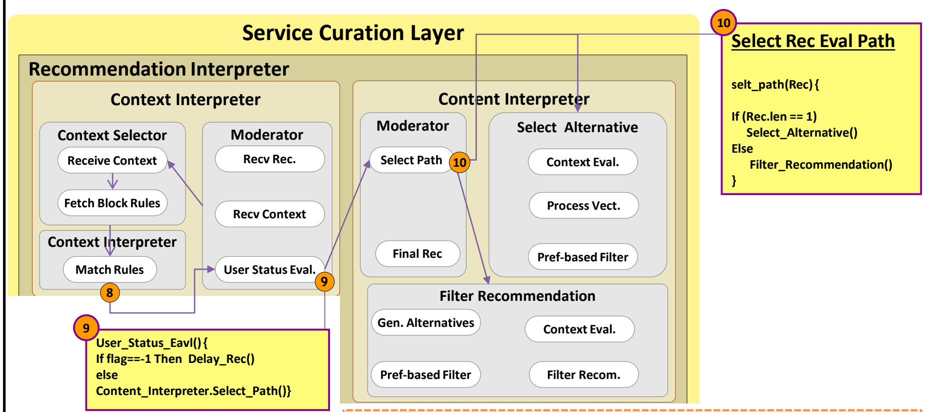


Recommendation Interpreter Scenario
Service Curation Layer
Recommendation Interpreter
Context Interpreter
Context Selector
Receive Context
Fetch Block Rules
Context Interpreter
Match Rules
Moderator
Recv Rec.
Recv Context
User Status Eval.
5
4
7
8
8
return flag Match(rules, context)
{
flag = -1
rules.add(scanFile.nextLine());
if(rules.contains(context)){
flag=1;
break; }
return flag;
}
6
If loc: "Home" AND HLC= "Sleeping"
If HLC= "Having Meal"
If HLC = "Commuting"
...
6 Repositories
Blocked Rules
Global Pref
Template
6 Blocking rules are fetched from the repository
7 Both context and rules are forwarded to the Context Interpreter
8 Context Interpreter evaluates context against the rules and decides about User availability Status



Recommendation Interpreter Scenario
Service Curation Layer
Recommendation Interpreter
Context Interpreter
Context Selector
Receive Context
Fetch Block Rules
Context Interpreter
Match Rules
Moderator
Recv Rec.
Recv Context
User Status Eval.
Content Interpreter
Moderator
Select Path
Final Rec.
Select Alternative
Context Eval.
Process Vect.
Pref-based Filter
Filter Recommendation
Gen. Alternatives
Context Eval.
Pref-based Filter
Filter Recom.
7
8
9
8
return flag Match(rules, context)
{
flag = -1
rules.add(scanFile.nextLine());
if(rules.contains(context)){
flag=1;
break; }
return flag;
}
9
User_Status_Eavl() {
If flag==-1
Delay_Rec()
else
Content_Interpreter.Select_Path()
}
7 Both context and rules are forwarded to the Context Interpreter
8 Context Interpreter evaluates context against the rules and decides about User availability Status
9 If user is available then Content Interpreter is invoked otherwise recommendation is delayed

Recommendation Interpreter Scenario

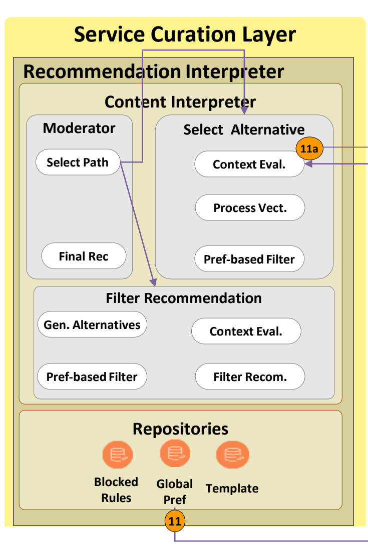


8 Context Interpreter evaluates context against the rules and decides about User availability Status

9 If user is available then Content Interpreter is invoked otherwise recommendation is delayed

10 If a single recommendation is received then "Select Alternative" is invoked otherwise "Filter Recommendation" is called for further processing

Recommendation Interpreter Scenario



11a **Prepared Context Matrix**

Contextual Matrix					
Context/Rec	Walking	Running	Stretching	Cycling	Sitting
Out doors	1	1	1	1	1
Amusement	1	0	1	1	1
Sunny	1	1	1	1	1
Happiness	1	1	1	1	1
Aggregate	1	0	1	1	1

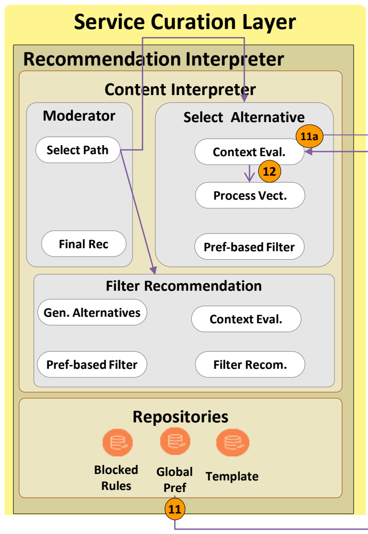
11 Global preferences are fetched from the local repository for cross-context recommendation

11a Contextual Matrix is generated to evaluate recommendation against the current context

11 **Get Global Preferences**

```
get_loc_pref(user_id);
get_hlc_pref(user_id);
get_weather_pref(user_id);
get_emotion_pref(user_id);
```

Recommendation Interpreter Scenario



11a **Prepared Context Matrix**

Contextual Matrix					
Context/Rec	Walking	Running	Stretching	Cycling	Sitting
Out doors	1	1	1	1	1
Amusement	1	0	1	0	1
Sunny	1	1	1	1	1
Happiness	1	1	1	1	1
Aggregate	1	0	1	1	1

Running
5m

11 Global preferences are fetched from the local repository for cross-context recommendation

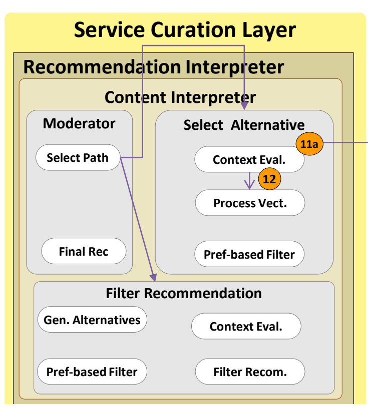
11a Contextual Matrix is generated to evaluate recommendation against the current context

12 Recommendation (running) is unsuitable in the current context

11 **Get Global Preferences**

get_loc_pref (user_id);
get_hlc_pref (user_id);
get_weather_pref (user_id);
get_emotion_pref (user_id);

Recommendation Interpreter Scenario



11a **Prepared Context Matrix**

Contextual Matrix					
Context/Rec	Walking	Running	Stretching	Cycling	Sitting
Out doors	1	1	1	1	1
Amusement	1	0	1	0	1
Sunny	1	1	1	1	1
Happiness	1	1	1	1	1
Aggregate	1	0	1	1	1

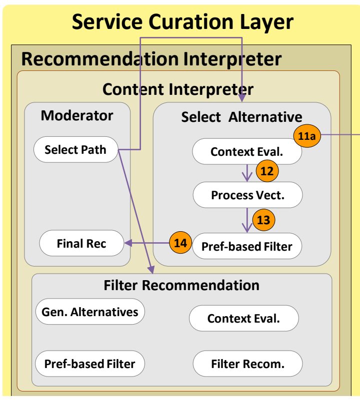
Walking
10m

Stretching
15m

11a Contextual Matrix is generated to evaluate recommendation against the current context

12 Recommendation (running) is unsuitable in the current context
In case of unsuitability, Alternative Recommendations are searched in the Contextual Matrix

Recommendation Interpreter Scenario



14 **Prepared Context Matrix**

Contextual Matrix					
Context/Rec	Walking	Running	Stretching	Cycling	Sitting
Out doors	1	1	1	1	1
Amusement	1	0	1	0	1
Sunny	1	1	1	1	1
Happiness	1	1	1	1	1
Aggregate	1	0	1	1	1

Walking 10m
Stretching 15m

get_pref (user_id);

12 Recommendation (running) is unsuitable in the current context
In case of unsuitability, Alternative Recommendations are searched in the Contextual Matrix

13 For multiple alternative recommendations, User's Preferences are weighed in

14 "Walking" is preferred over "Stretching" by the user therefore "Walking" is treated as final recommendation

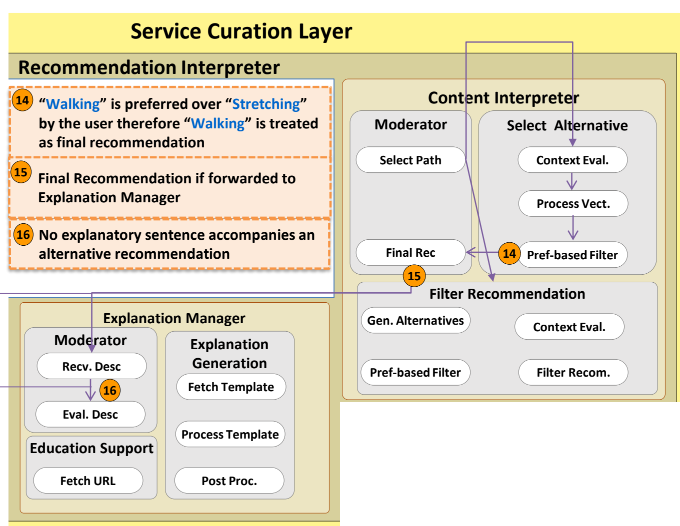
Recommendation Interpreter Scenario

15 **Received Description**

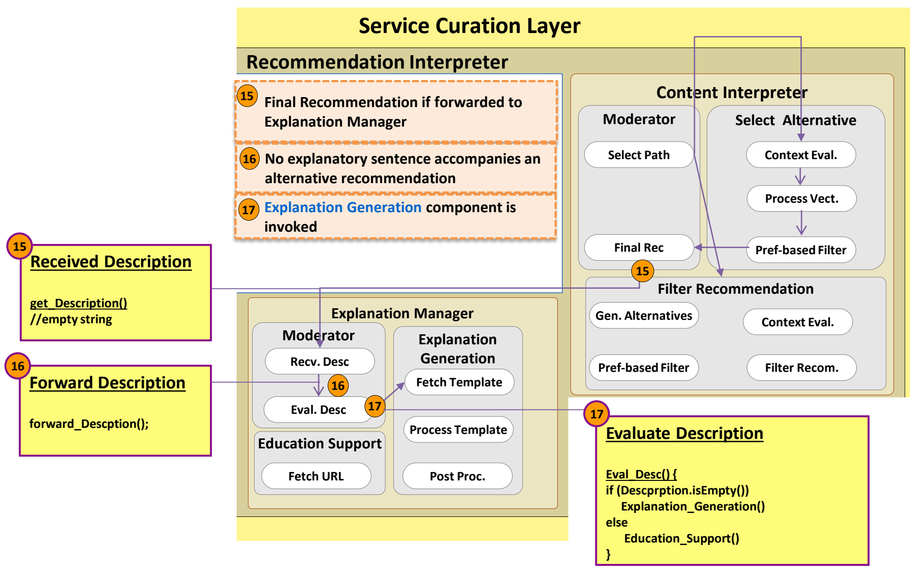
get_Description()
//empty string

16 **Forward Description**

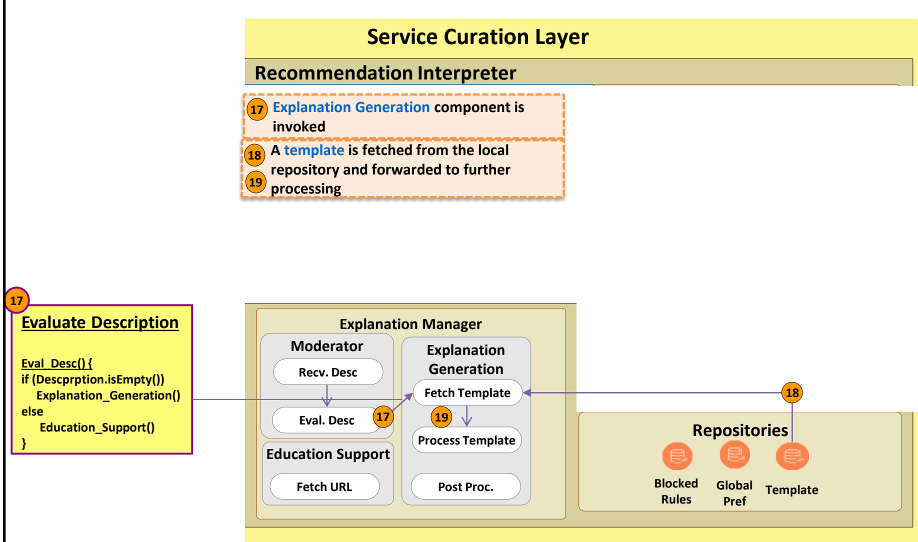
forward_Descption();

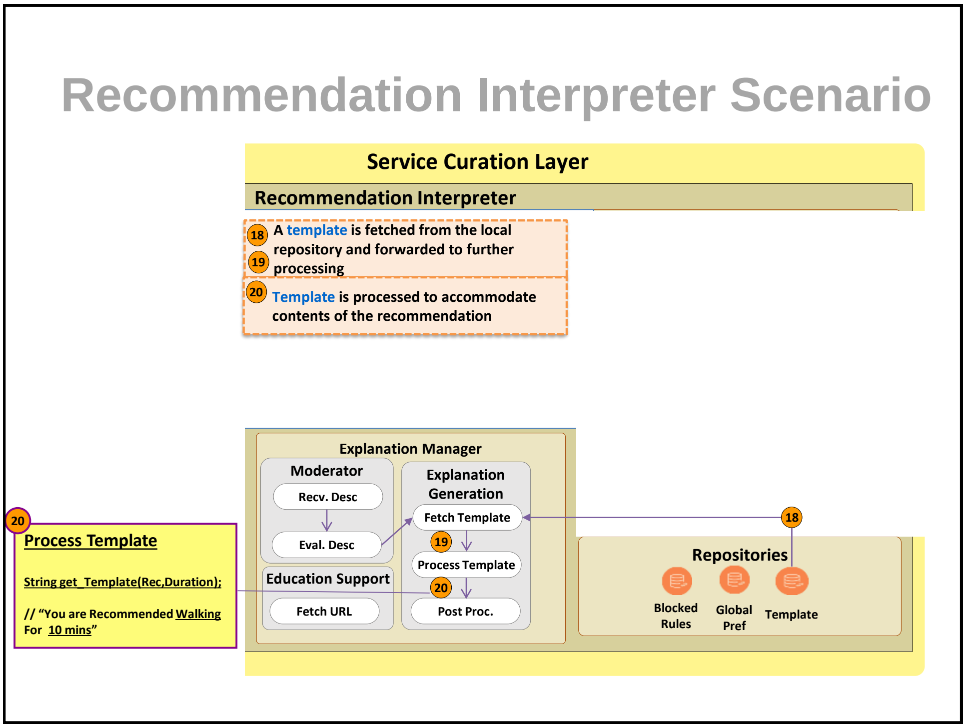


Recommendation Interpreter Scenario

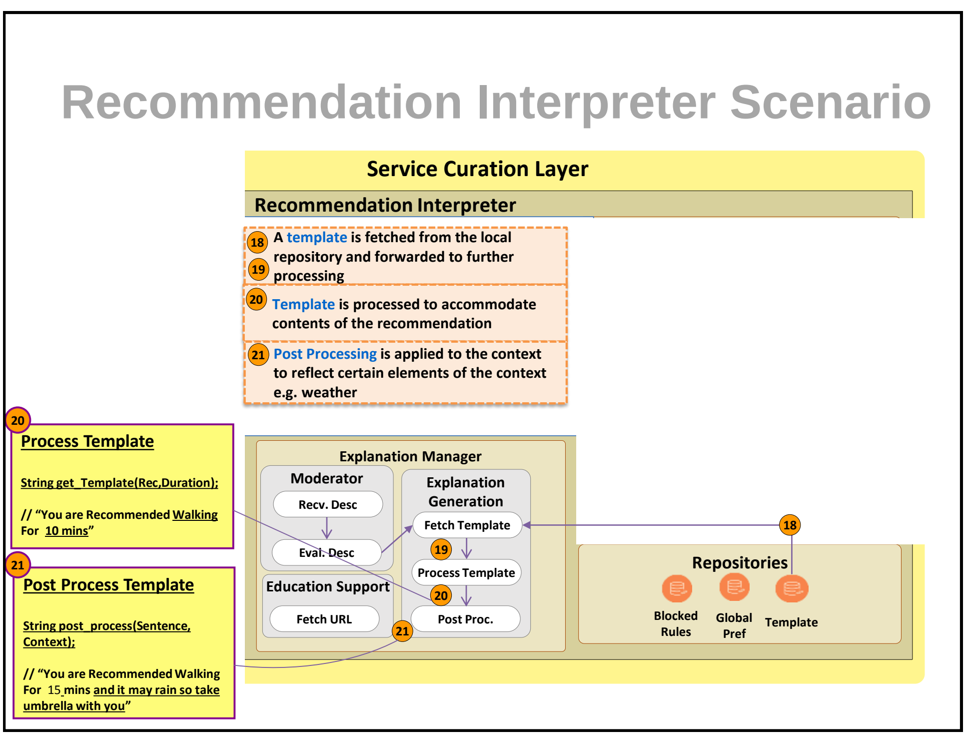


Recommendation Interpreter Scenario

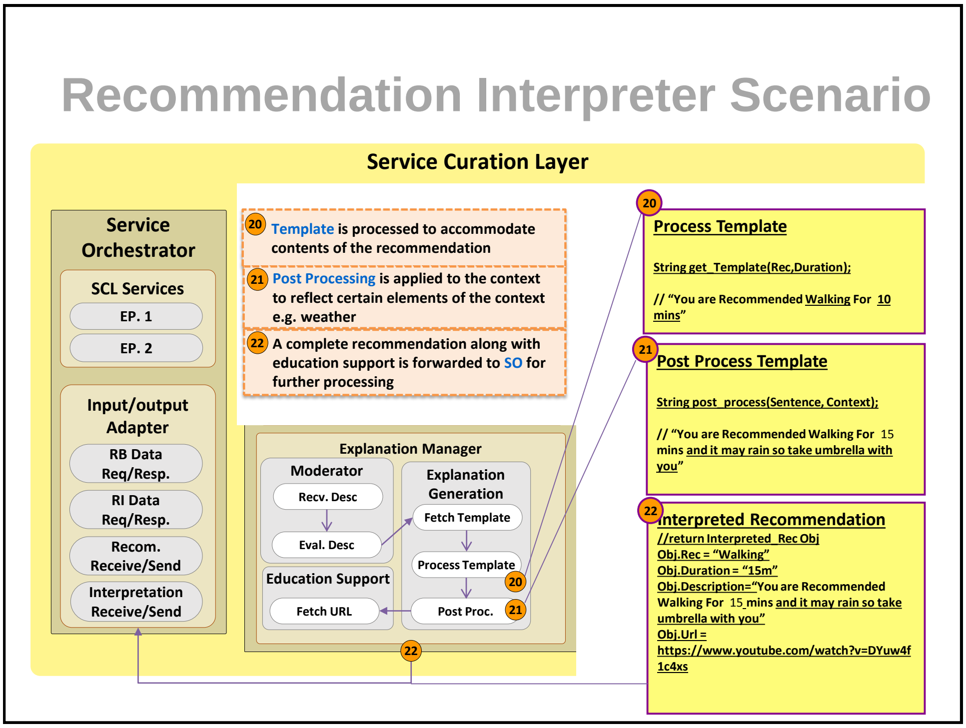


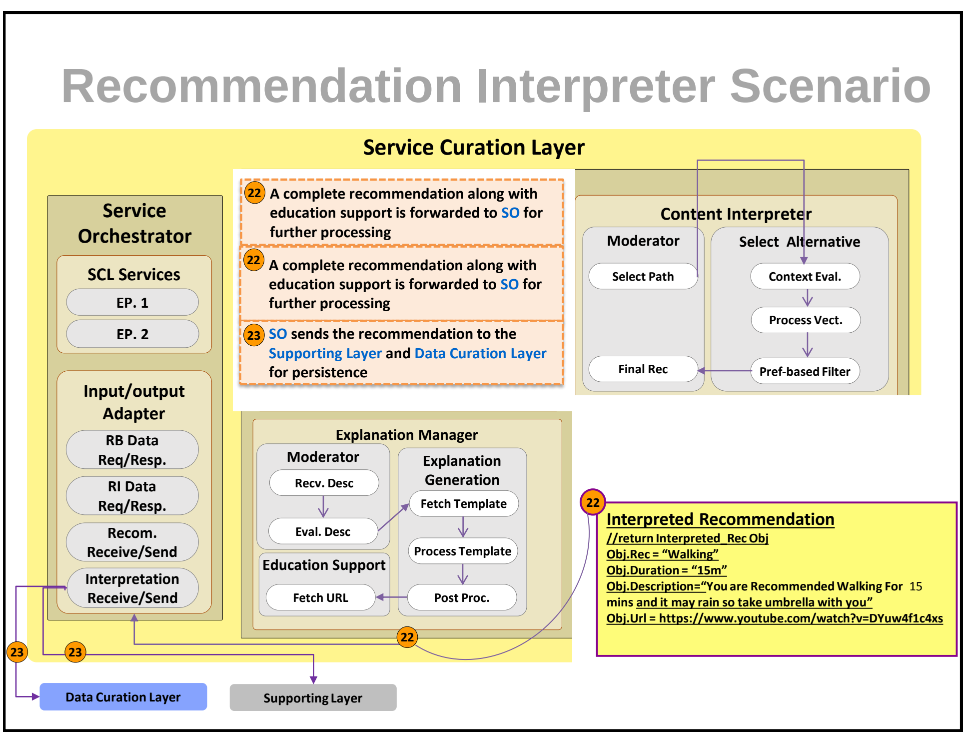


Recommendation Interpreter Scenario
Service Curation Layer
Recommendation Interpreter
18 A template is fetched from the local repository and forwarded to further
19 processing
20 Template is processed to accommodate contents of the recommendation
20 Process Template
String get_Template(Rec,Duration);
// "You are Recommended Walking For 10 mins"
Explanation Manager
Moderator
Recv. Desc
Eval. Desc
Education Support
Fetch URL
Explanation Generation
Fetch Template
19
Process Template
20
Post Proc.
18
Repositories
Blocked Rules
Global Pref
Template



Recommendation Interpreter Scenario
Service Curation Layer
Recommendation Interpreter
18 A template is fetched from the local repository and forwarded to further
19 processing
20 Template is processed to accommodate contents of the recommendation
21 Post Processing is applied to the context to reflect certain elements of the context e.g. weather
20 Process Template
String get_Template(Rec,Duration);
// "You are Recommended Walking For 10 mins"
21 Post Process Template
String post_process(Sentence, Context);
// "You are Recommended Walking For 15 mins and it may rain so take umbrella with you"
Explanation Manager
Moderator
Recv. Desc
Eval. Desc
Education Support
Fetch URL
Explanation Generation
Fetch Template
19
Process Template
20
Post Proc.
21
18
Repositories
Blocked Rules
Global Pref
Template

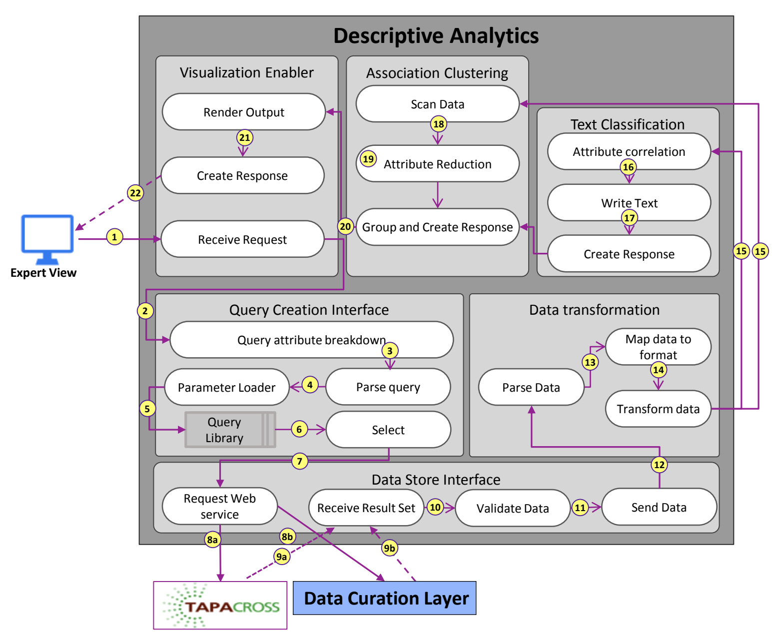

Recommendation Interpreter Scenario
Service Curation Layer
Service Orchestrator
SCL Services
EP. 1
EP. 2
Input/output Adapter
RB Data Req/Resp.
RI Data Req/Resp.
Recom. Receive/Send
Interpretation Receive/Send
20 Template is processed to accommodate contents of the recommendation
21 Post Processing is applied to the context to reflect certain elements of the context e.g. weather
22 A complete recommendation along with education support is forwarded to SO for further processing
Explanation Manager
Moderator
Recv. Desc
Eval. Desc
Explanation Generation
Fetch Template
Process Template
Post Proc.
Education Support
Fetch URL
20 Process Template
String get_Template(Rec,Duration);
// "You are Recommended Walking For 10 mins"
21 Post Process Template
String post_process(Sentence, Context);
// "You are Recommended Walking For 15 mins and it may rain so take umbrella with you"
22 Interpreted Recommendation
//return Interpreted_Rec Obj
Obj.Rec = "Walking"
Obj.Duration = "15m"
Obj.Description="You are Recommended Walking For 15 mins and it may rain so take umbrella with you"
Obj.Url = https://www.youtube.com/watch?v=DYuw4f1c4xs


Recommendation Interpreter Scenario
Service Curation Layer
Service Orchestrator
SCL Services
EP. 1
EP. 2
Input/output Adapter
RB Data Req/Resp.
RI Data Req/Resp.
Recom. Receive/Send
Interpretation Receive/Send
22 A complete recommendation along with education support is forwarded to SO for further processing
22 A complete recommendation along with education support is forwarded to SO for further processing
23 SO sends the recommendation to the Supporting Layer and Data Curation Layer for persistence
Content Interpreter
Moderator
Select Path
Final Rec
Select Alternative
Context Eval.
Process Vect.
Pref-based Filter
Explanation Manager
Moderator
Recv. Desc
Eval. Desc
Explanation Generation
Fetch Template
Process Template
Post Proc.
Education Support
Fetch URL
22 Interpreted Recommendation
//return Interpreted_Rec Obj
Obj.Rec = "Walking"
Obj.Duration = "15m"
Obj.Description="You are Recommended Walking For 15 mins and it may rain so take umbrella with you"
Obj.Url = https://www.youtube.com/watch?v=DYuw4f1c4xs
Data Curation Layer
Supporting Layer

STENCILS for SL Micro-Demo

Descriptive Analytics

DESCRIPTIVE ANALYTICS DETAILED ARCHITECTURE

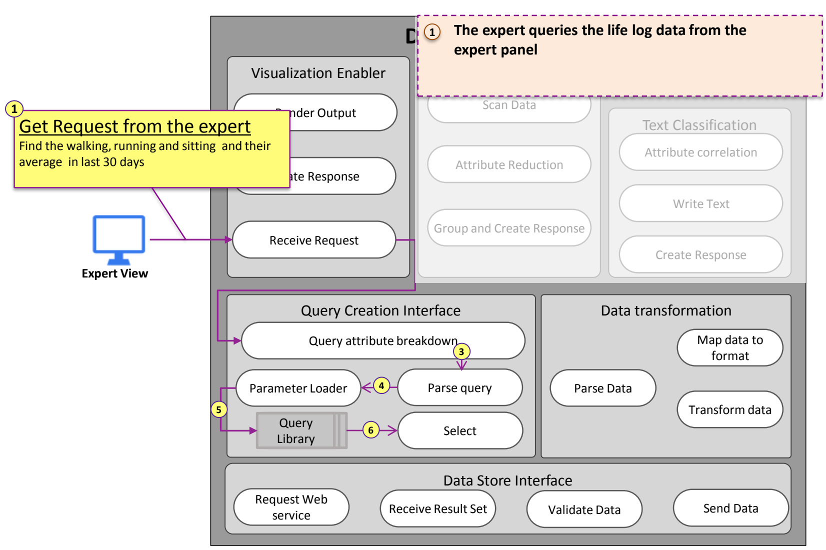


Scenario 1: Visualization and Analytics from Lifelog data and SNS

1. The request is **parsed** from the expert.
2. The request is converted into a **queries** from the library.
3. Data is integrated and **transformed** into a predefined format
4. Trend analysis is done through **calculating facts** and **grouping** data
5. The data is **visualized** in graphs and data facts are presented.

QUERY PARSING AND CREATION

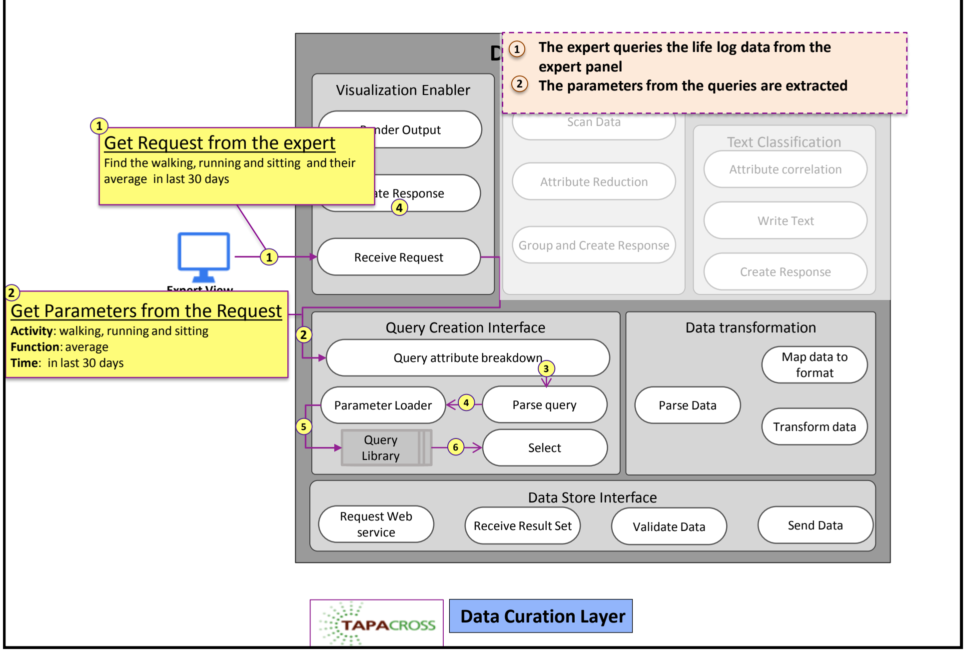
1/6



Data Curation Layer

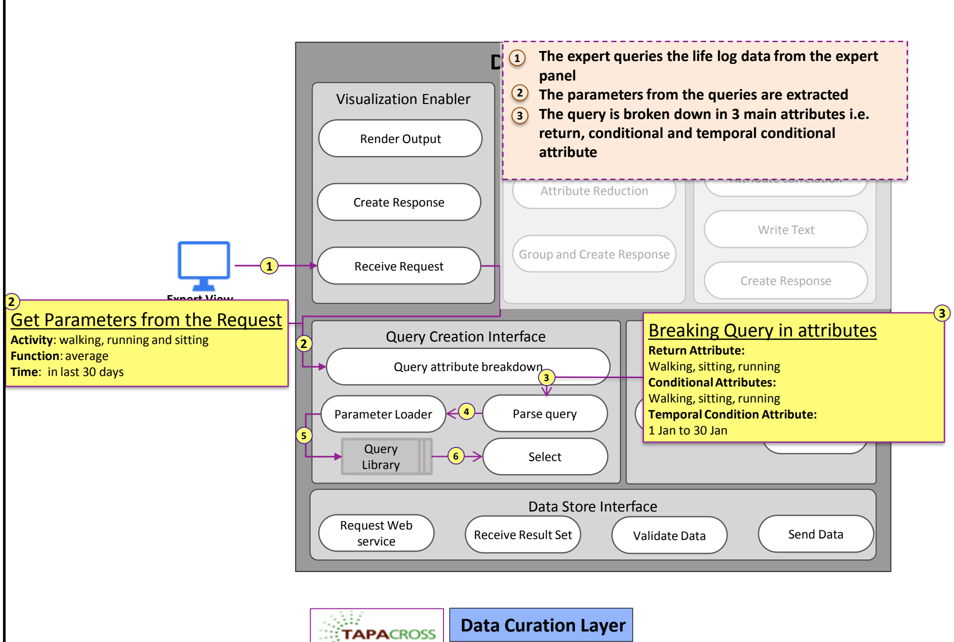
QUERY PARSING AND CREATION

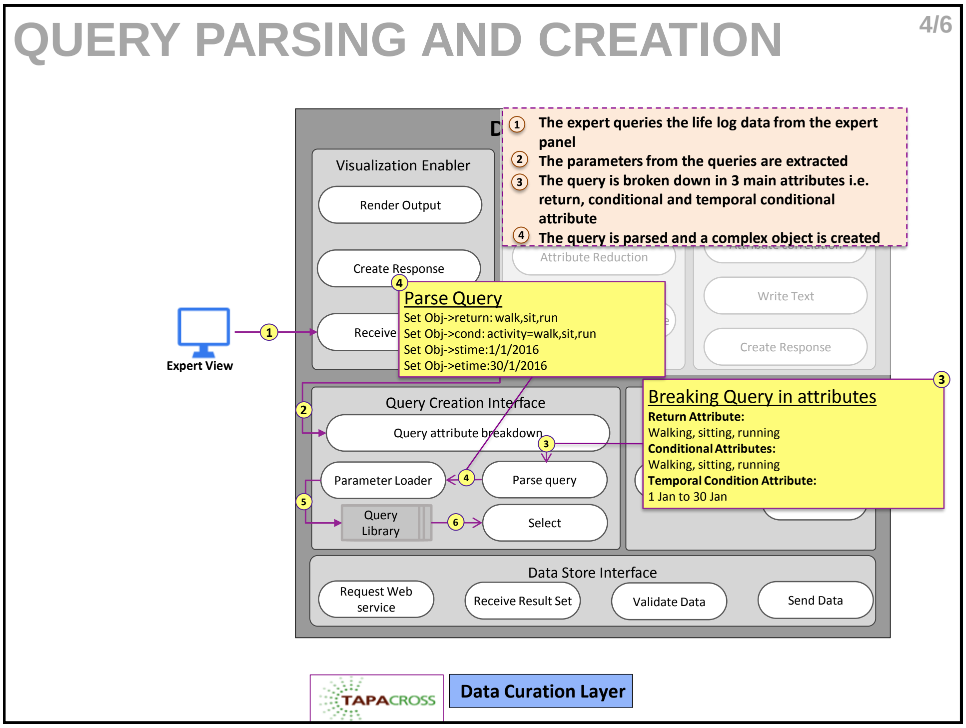
2/6

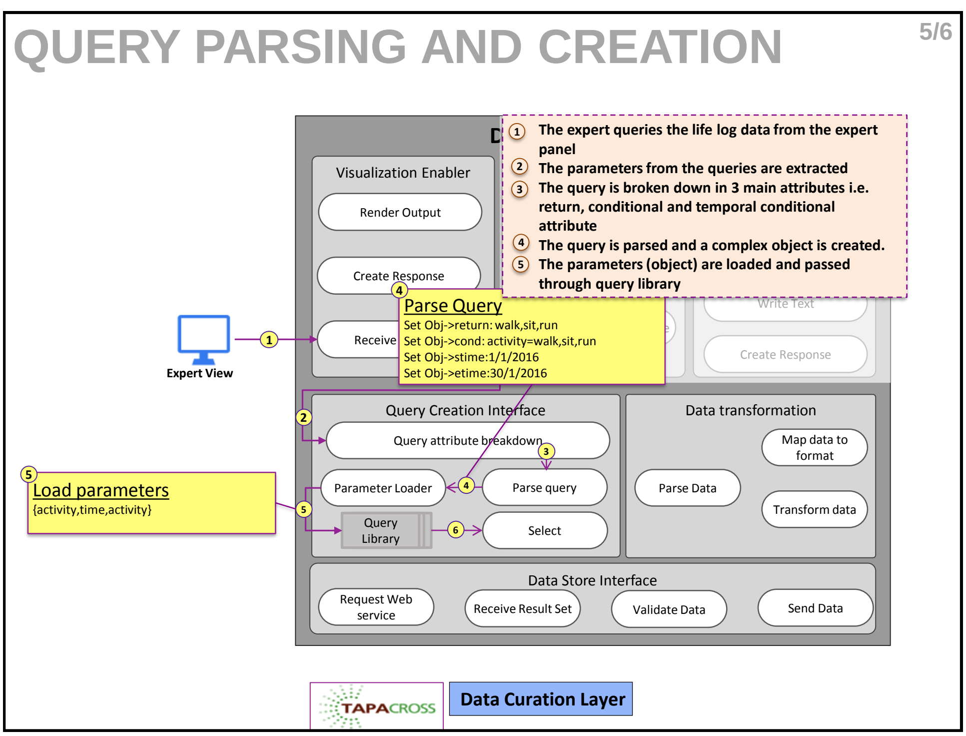


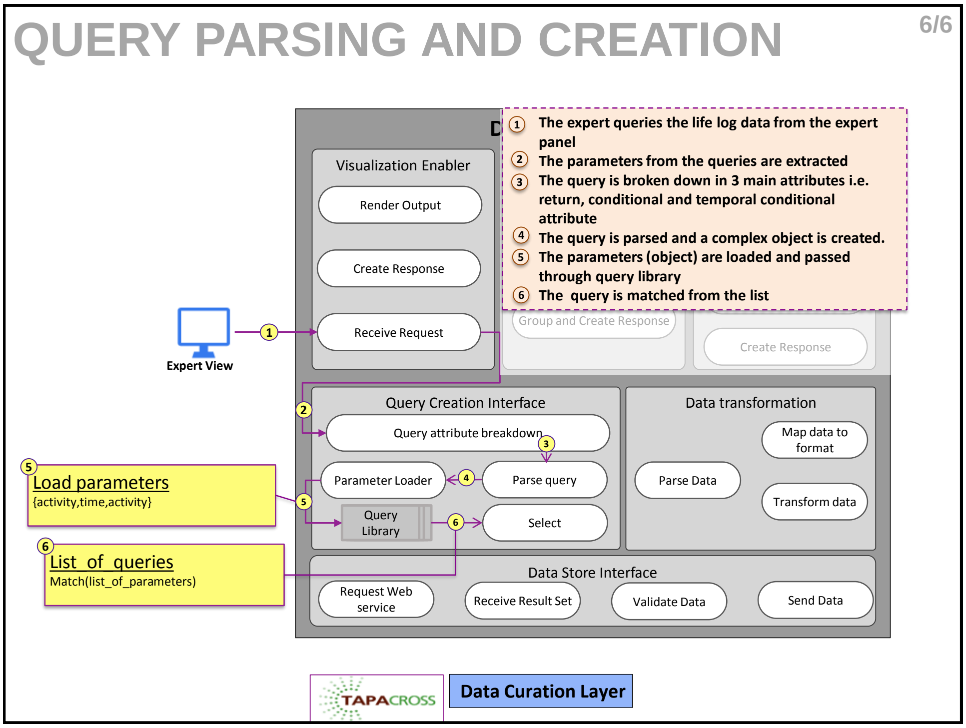
QUERY PARSING AND CREATION

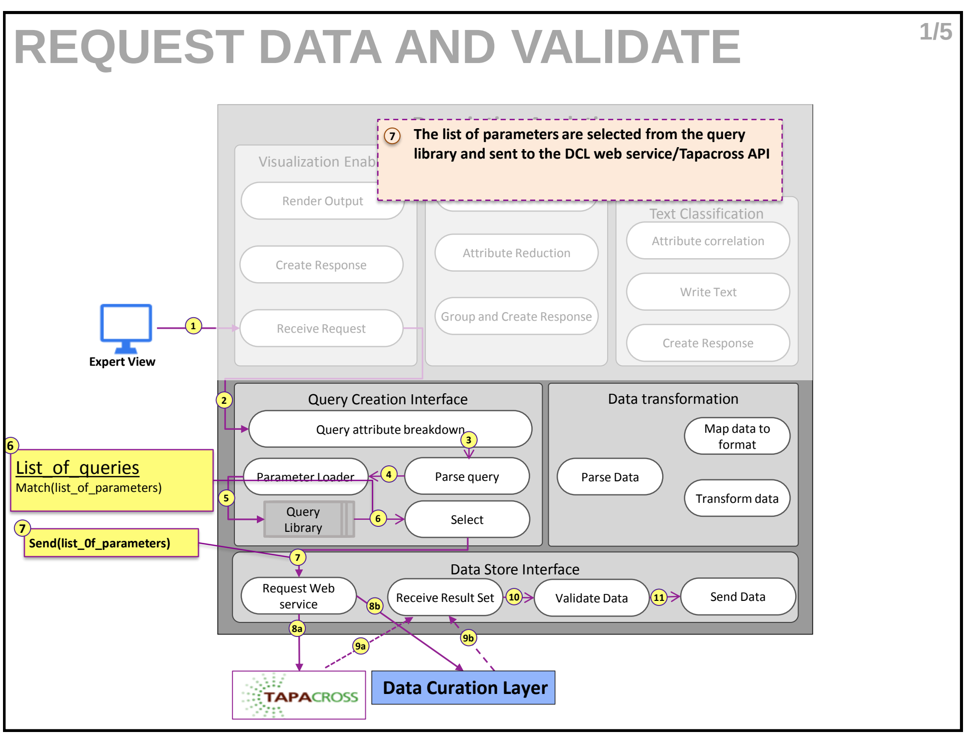
3/6




QUERY PARSING AND CREATION
4/6
The expert queries the life log data from the expert panel
The parameters from the queries are extracted
The query is broken down in 3 main attributes i.e. return, conditional and temporal conditional attribute
The query is parsed and a complex object is created
Visualization Enabler
Render Output
Create Response
Receive
Parse Query
Set Obj->return: walk,sit,run
Set Obj->cond: activity=walk,sit,run
Set Obj->stime:1/1/2016
Set Obj->etime:30/1/2016
Attribute Reduction
Write Text
Create Response
Expert View
Query Creation Interface
Query attribute breakdown
Parameter Loader
Parse query
Query Library
Select
Breaking Query in attributes
Return Attribute:
Walking, sitting, running
Conditional Attributes:
Walking, sitting, running
Temporal Condition Attribute:
1 Jan to 30 Jan
Data Store Interface
Request Web service
Receive Result Set
Validate Data
Send Data
TAPACROSS
Data Curation Layer

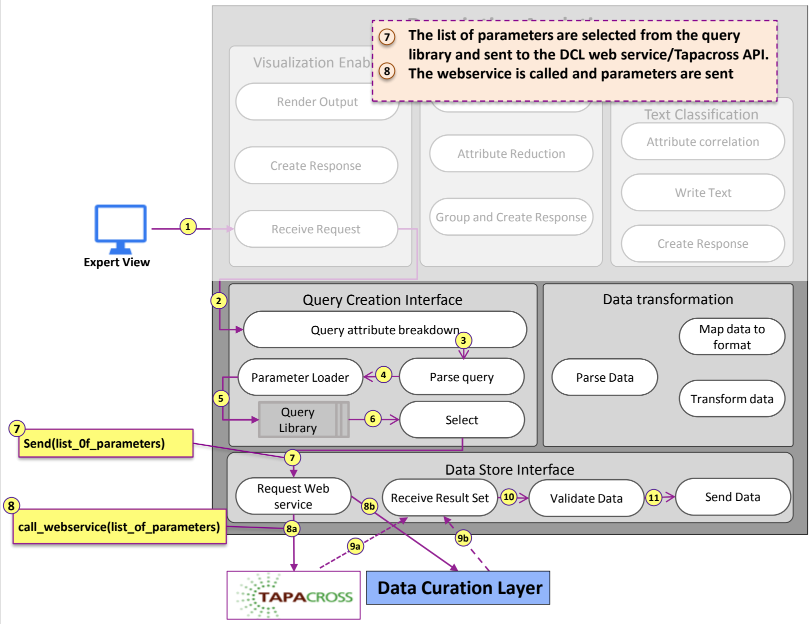

QUERY PARSING AND CREATION
5/6
The expert queries the life log data from the expert panel
The parameters from the queries are extracted
The query is broken down in 3 main attributes i.e. return, conditional and temporal conditional attribute
The query is parsed and a complex object is created.
The parameters (object) are loaded and passed through query library
Visualization Enabler
Render Output
Create Response
Receive
Parse Query
Set Obj->return: walk,sit,run
Set Obj->cond: activity=walk,sit,run
Set Obj->stime:1/1/2016
Set Obj->etime:30/1/2016
Write Text
Create Response
Expert View
Load parameters
{activity,time,activity}
Query Creation Interface
Query attribute breakdown
Parameter Loader
Parse query
Query Library
Select
Data transformation
Map data to format
Parse Data
Transform data
Data Store Interface
Request Web service
Receive Result Set
Validate Data
Send Data
TAPACROSS
Data Curation Layer


QUERY PARSING AND CREATION
6/6
① The expert queries the life log data from the expert panel
② The parameters from the queries are extracted
③ The query is broken down in 3 main attributes i.e. return, conditional and temporal conditional attribute
④ The query is parsed and a complex object is created.
⑤ The parameters (object) are loaded and passed through query library
⑥ The query is matched from the list
Visualization Enabler
Render Output
Create Response
Receive Request
Group and Create Response
Create Response
Expert View
Query Creation Interface
Query attribute breakdown
Parameter Loader
Parse query
Query Library
Select
Data transformation
Map data to format
Parse Data
Transform data
Load parameters
{activity,time,activity}
List_of_queries
Match(list_of_parameters)
Data Store Interface
Request Web service
Receive Result Set
Validate Data
Send Data
TAPACROSS
Data Curation Layer


REQUEST DATA AND VALIDATE
1/5
⑦ The list of parameters are selected from the query library and sent to the DCL web service/Tapacross API
Visualization Enab
Render Output
Create Response
Receive Request
Attribute Reduction
Group and Create Response
Text Classification
Attribute correlation
Write Text
Create Response
Expert View
Query Creation Interface
Query attribute breakdown
Parameter Loader
Parse query
Query Library
Select
Data transformation
Map data to format
Parse Data
Transform data
List_of_queries
Match(list_of_parameters)
Send(list_0f_parameters)
Data Store Interface
Request Web service
Receive Result Set
Validate Data
Send Data
TAPACROSS
Data Curation Layer

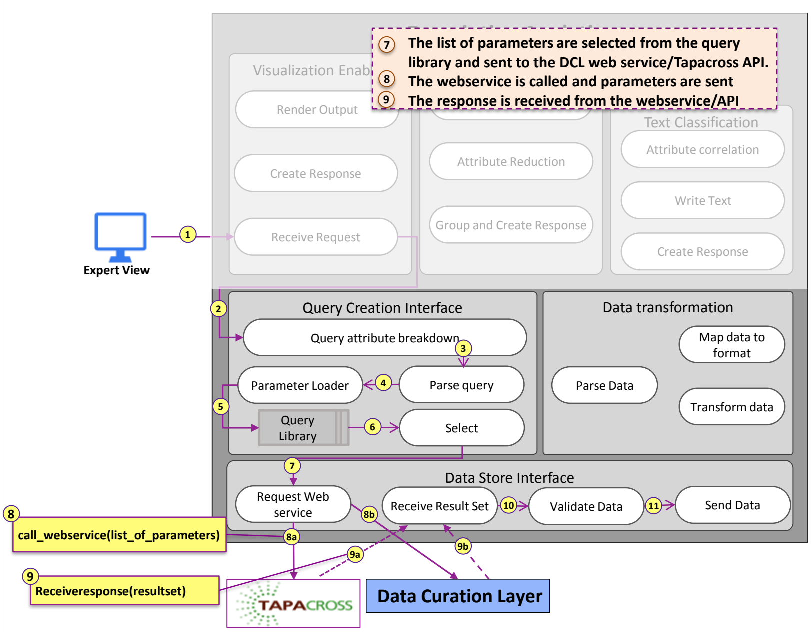
REQUEST DATA AND VALIDATE

2/5



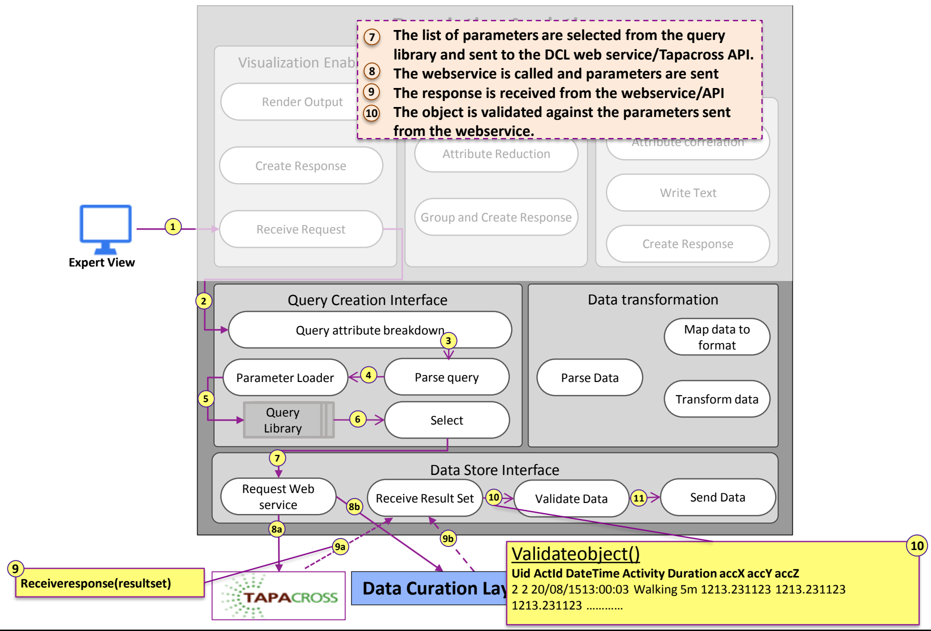
REQUEST DATA AND VALIDATE

3/5



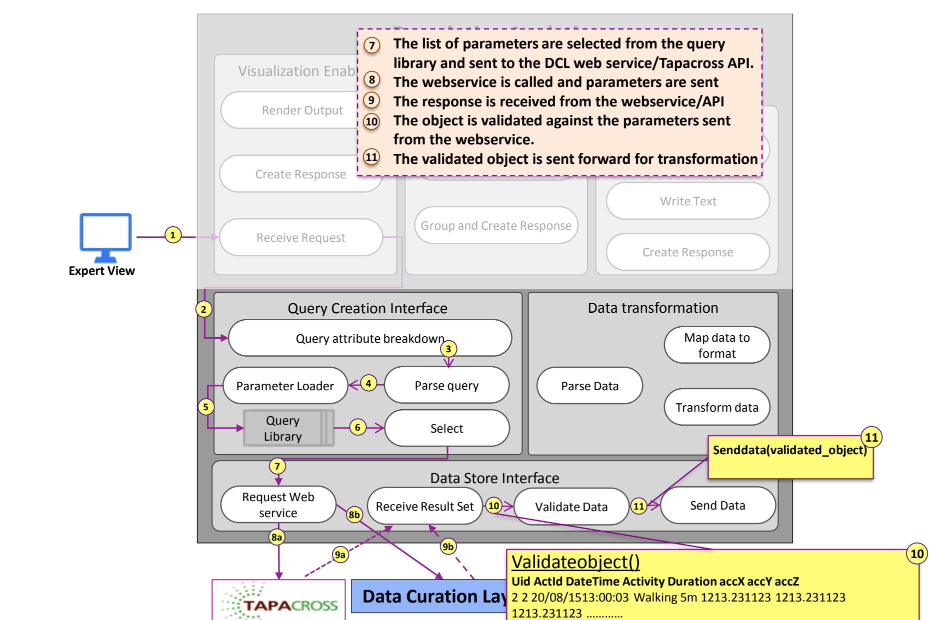
REQUEST DATA AND VALIDATE

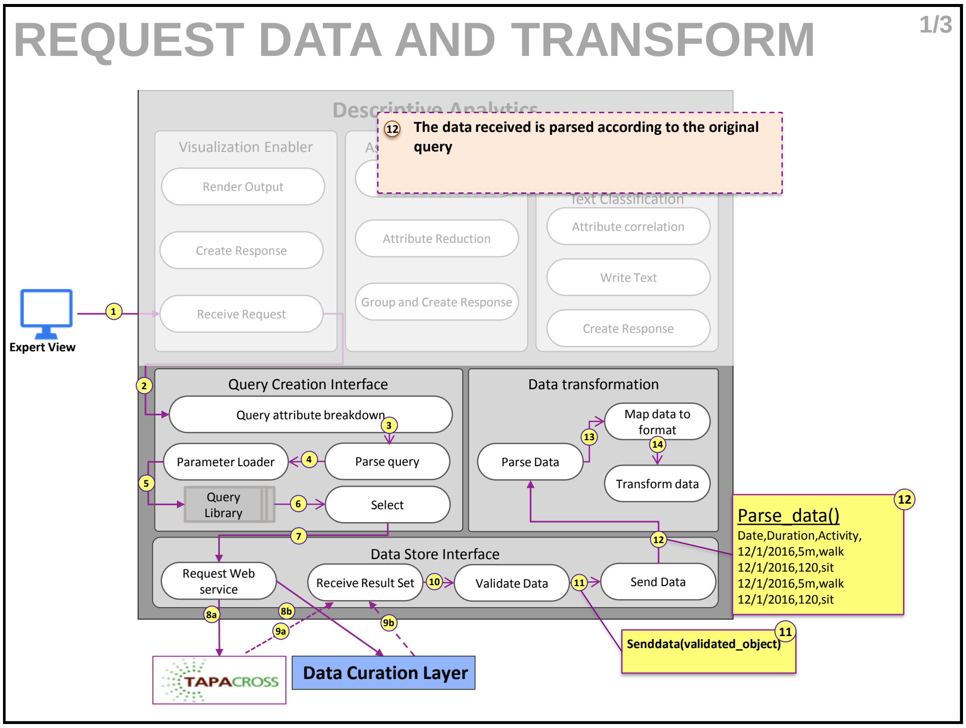
4/5

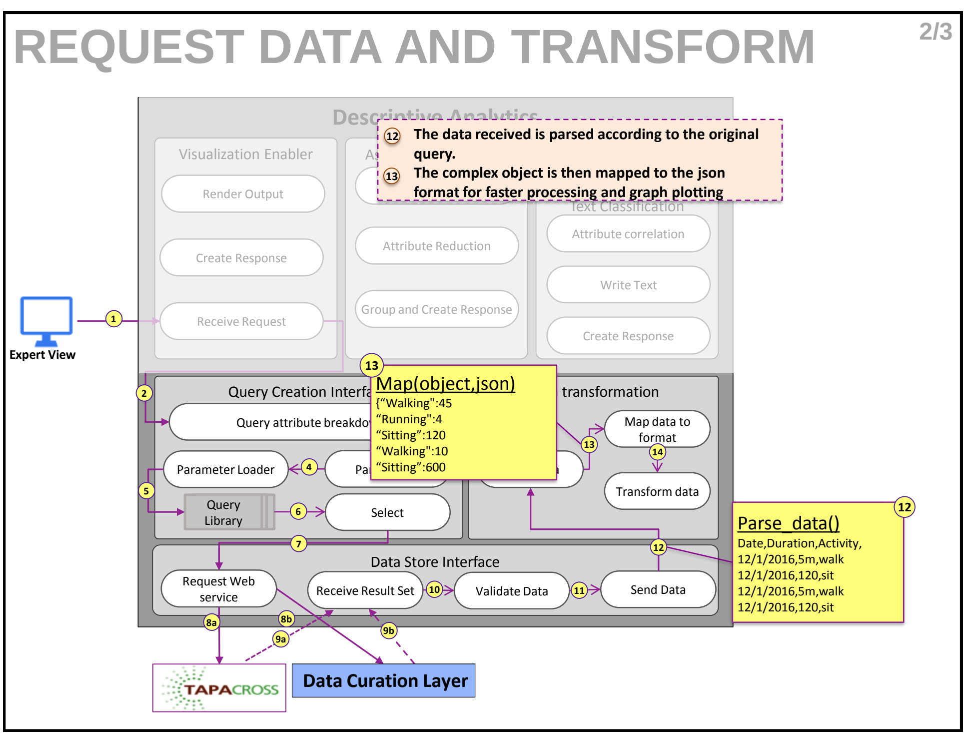


REQUEST DATA AND VALIDATE

5/5

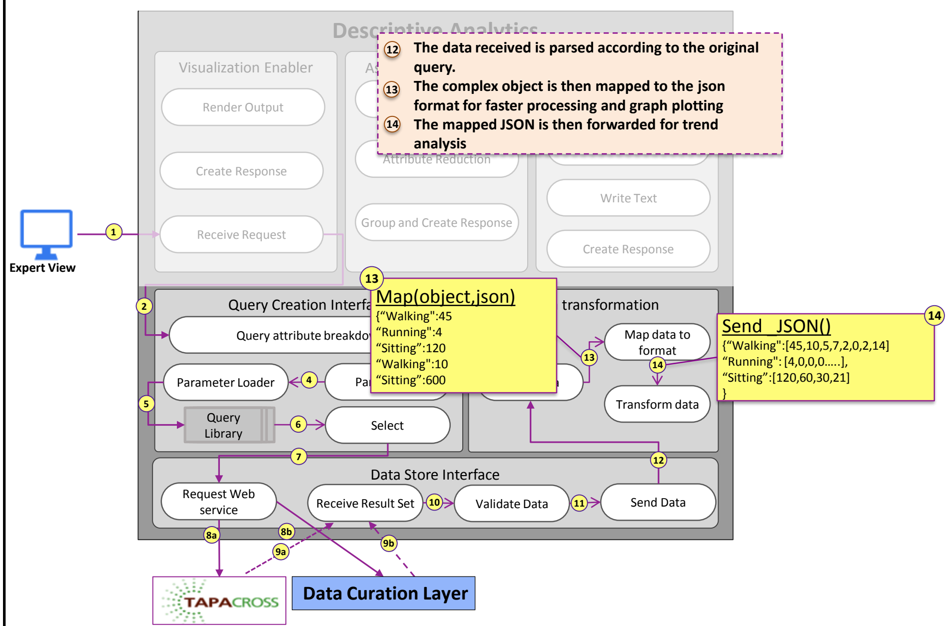



REQUEST DATA AND TRANSFORM
1/3
Descriptive Analytics
12 The data received is parsed according to the original query
Visualization Enabler
Render Output
Create Response
Receive Request
Text Classification
Attribute correlation
Write Text
Create Response
Attribute Reduction
Group and Create Response
Expert View
Query Creation Interface
Query attribute breakdown
Parameter Loader
Parse query
Query Library
Select
Data transformation
Map data to format
Parse Data
Transform data
Data Store Interface
Request Web service
Receive Result Set
Validate Data
Send Data
Parse_data()
Date,Duration,Activity,
12/1/2016,5m,walk
12/1/2016,120,sit
12/1/2016,5m,walk
12/1/2016,120,sit
Senddata(validated_object)
TAPACROSS
Data Curation Layer


REQUEST DATA AND TRANSFORM
2/3
Descriptive Analytics
12 The data received is parsed according to the original query.
13 The complex object is then mapped to the json format for faster processing and graph plotting
Visualization Enabler
Render Output
Create Response
Receive Request
Text Classification
Attribute correlation
Write Text
Create Response
Attribute Reduction
Group and Create Response
Expert View
Map(object,json)
{"Walking":45
"Running":4
"Sitting":120
"Walking":10
"Sitting":600
Query attribute breakdo
Parameter Loader
Query Library
Select
transformation
Map data to format
Transform data
Data Store Interface
Request Web service
Receive Result Set
Validate Data
Send Data
Parse_data()
Date,Duration,Activity,
12/1/2016,5m,walk
12/1/2016,120,sit
12/1/2016,5m,walk
12/1/2016,120,sit
TAPACROSS
Data Curation Layer

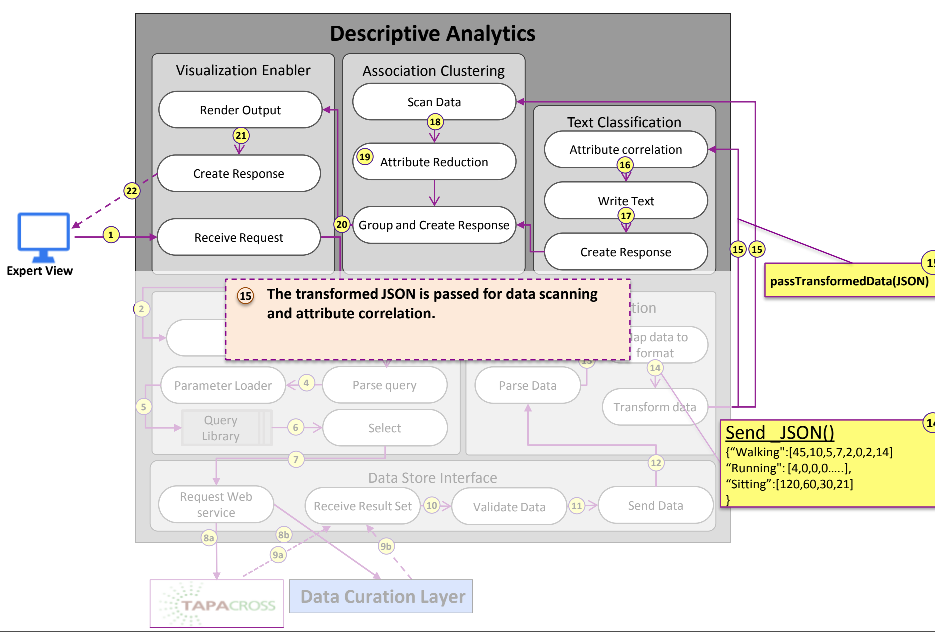
REQUEST DATA AND TRANSFORM

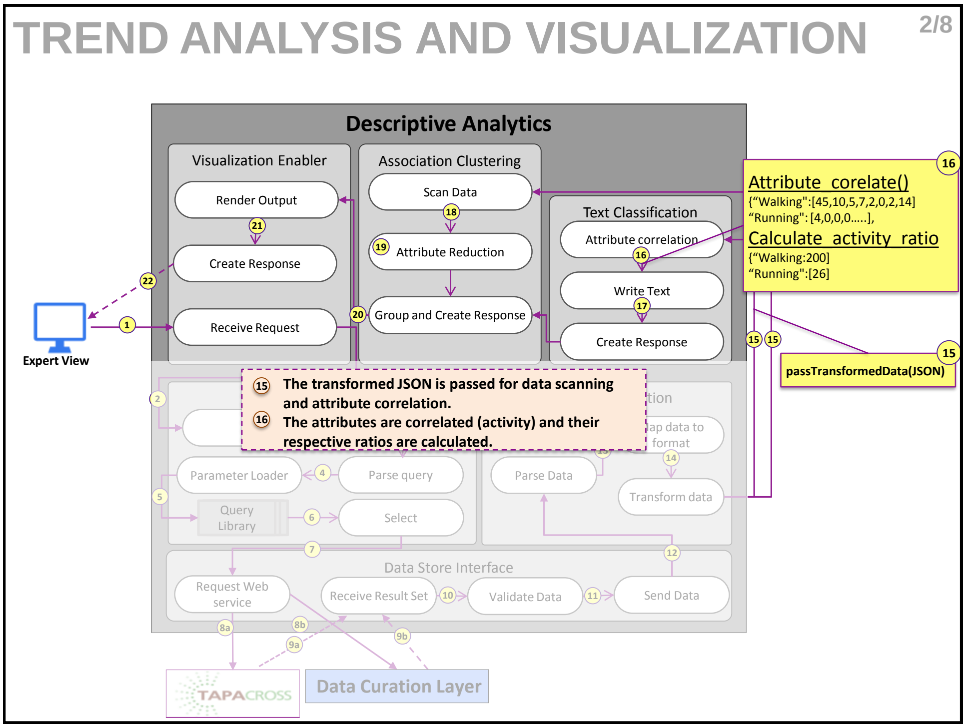
3/3

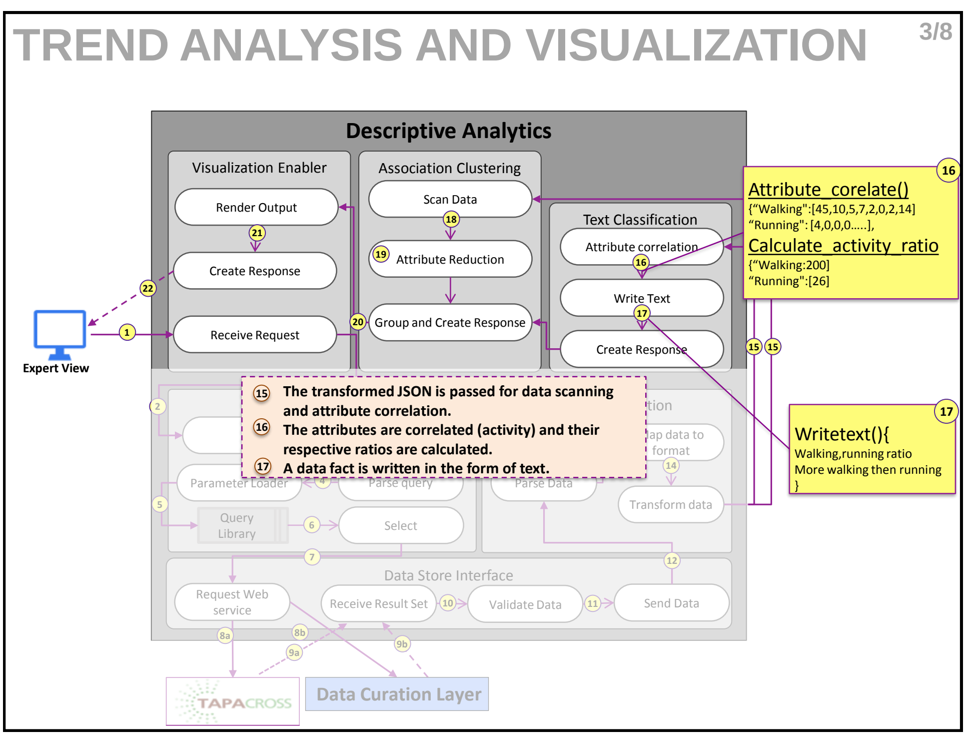


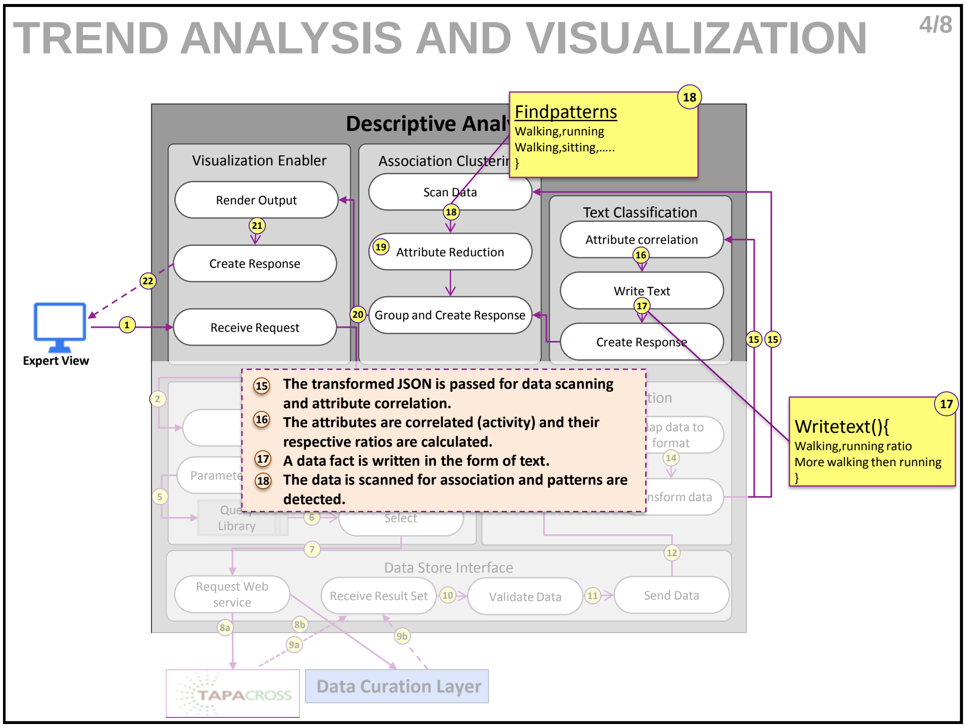
TREND ANALYSIS AND VISUALIZATION

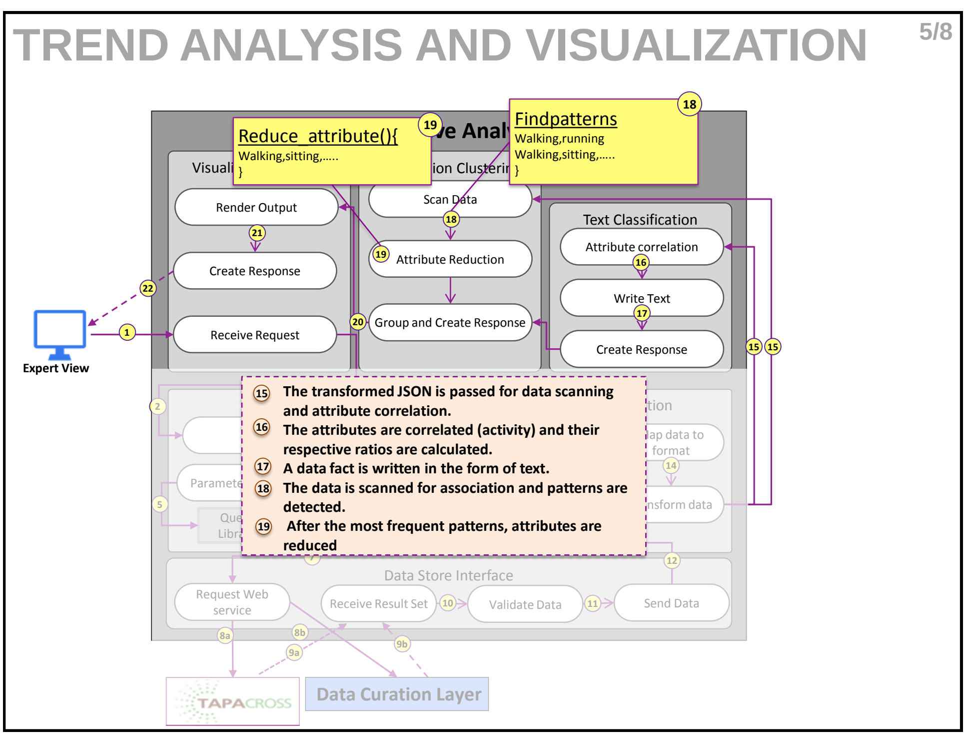
1/8

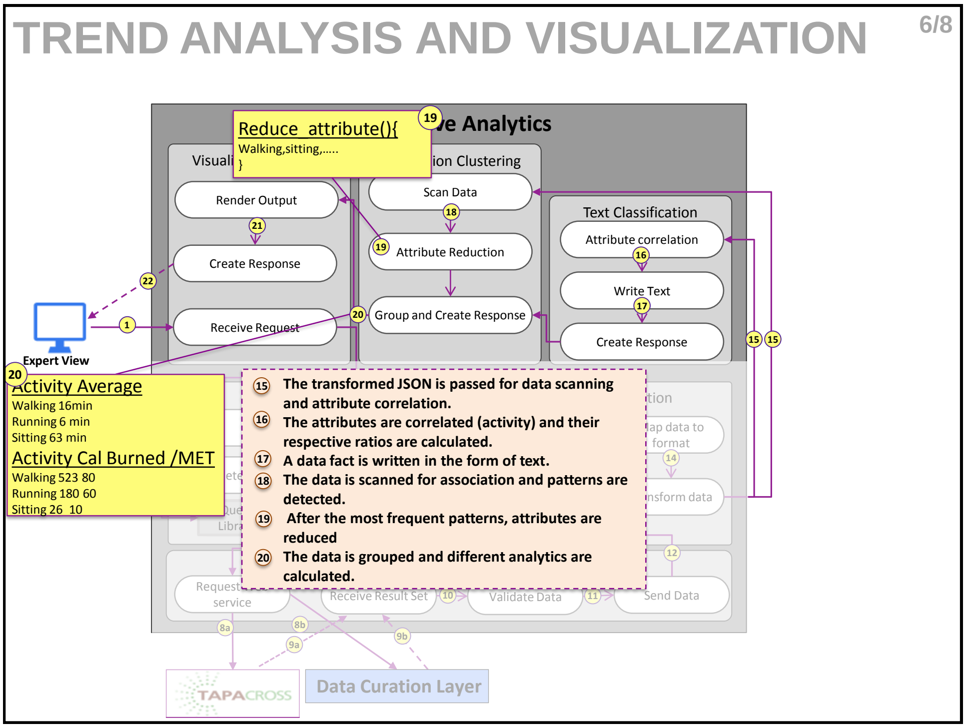



TREND ANALYSIS AND VISUALIZATION
2/8
Descriptive Analytics
Visualization Enabler
Render Output
Create Response
Receive Request
Association Clustering
Scan Data
Attribute Reduction
Group and Create Response
Text Classification
Attribute correlation
Write Text
Create Response
Expert View
Attribute_corelate()
{"Walking":[45,10,5,7,2,0,2,14]
"Running":[4,0,0,0.....],
Calculate_activity_ratio
{"Walking:200]
"Running":[26]
passTransformedData(JSON)
The transformed JSON is passed for data scanning and attribute correlation.
The attributes are correlated (activity) and their respective ratios are calculated.
Parameter Loader
Parse query
Query Library
Select
Parse Data
Transform data
Data Store Interface
Request Web service
Receive Result Set
Validate Data
Send Data
TAPACROSS
Data Curation Layer

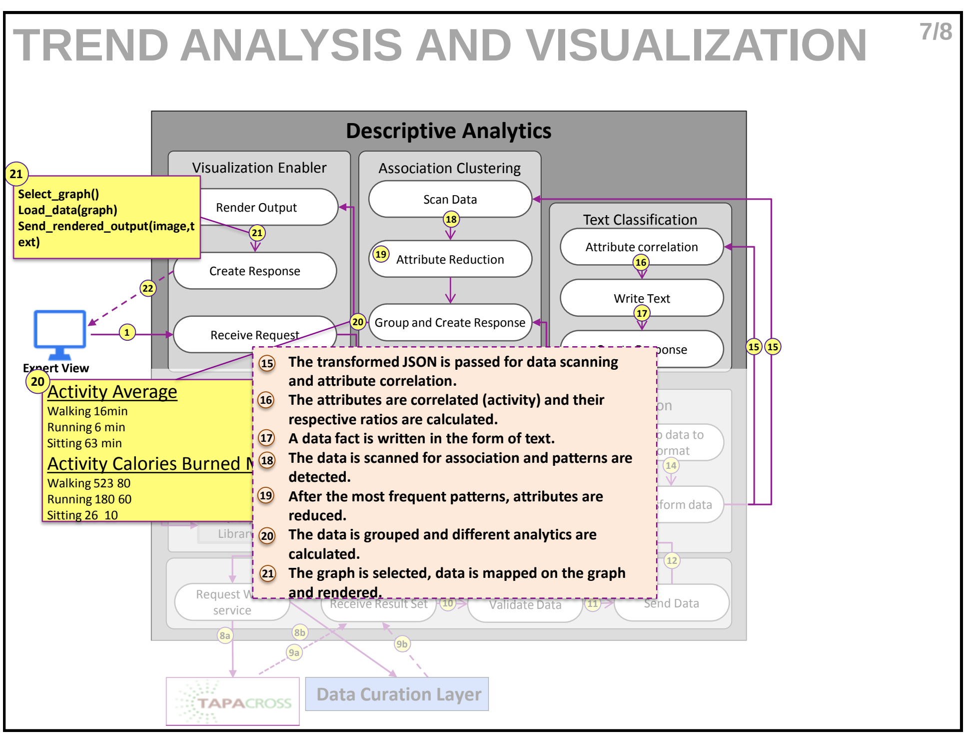

TREND ANALYSIS AND VISUALIZATION
3/8
Descriptive Analytics
Visualization Enabler
Render Output
Create Response
Receive Request
Association Clustering
Scan Data
Attribute Reduction
Group and Create Response
Text Classification
Attribute correlation
Write Text
Create Response
Expert View
Attribute_corelate()
{"Walking":[45,10,5,7,2,0,2,14]
"Running":[4,0,0,0.....],
Calculate_activity_ratio
{"Walking:200]
"Running":[26]
Writetext(){
Walking,running ratio
More walking then running
}
The transformed JSON is passed for data scanning and attribute correlation.
The attributes are correlated (activity) and their respective ratios are calculated.
A data fact is written in the form of text.
Parameter Loader
Parse query
Query Library
Select
Parse Data
Transform data
Data Store Interface
Request Web service
Receive Result Set
Validate Data
Send Data
TAPACROSS
Data Curation Layer


TREND ANALYSIS AND VISUALIZATION
4/8
Descriptive Anal
Findpatterns
Walking,running
Walking,sitting,.....
}
18
Visualization Enabler
Association Clusterir
Render Output
Scan Data
18
21
Attribute Reduction
19
Create Response
Text Classification
Attribute correlation
16
Write Text
17
Create Response
22
Group and Create Response
20
Receive Request
1
Expert View
15
15
15 The transformed JSON is passed for data scanning and attribute correlation.
16 The attributes are correlated (activity) and their respective ratios are calculated.
17 A data fact is written in the form of text.
18 The data is scanned for association and patterns are detected.
Writetext(){
Walking,running ratio
More walking then running
}
17
Data Store Interface
Request Web service
Receive Result Set
Validate Data
Send Data
TAPACROSS
Data Curation Layer

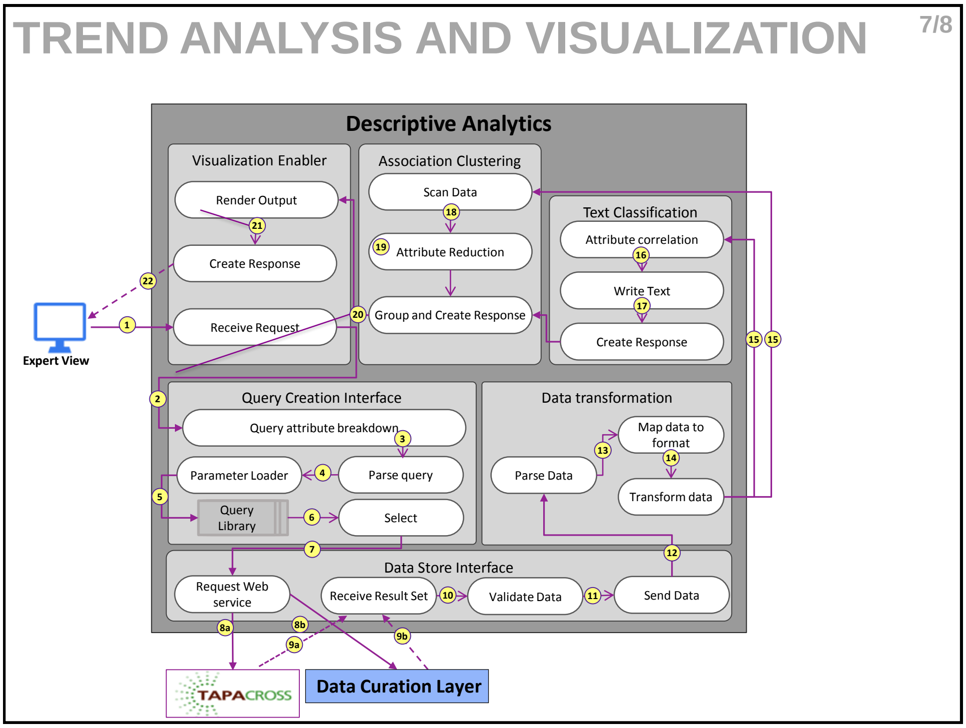

TREND ANALYSIS AND VISUALIZATION
5/8
Reduce_attribute(){
Walking,sitting,.....
}
19
Findpatterns
Walking,running
Walking,sitting,.....
}
18
Render Output
Scan Data
18
21
Attribute Reduction
19
Create Response
Text Classification
Attribute correlation
16
Write Text
17
Create Response
22
Group and Create Response
20
Receive Request
1
Expert View
15
15
15 The transformed JSON is passed for data scanning and attribute correlation.
16 The attributes are correlated (activity) and their respective ratios are calculated.
17 A data fact is written in the form of text.
18 The data is scanned for association and patterns are detected.
19 After the most frequent patterns, attributes are reduced
Data Store Interface
Request Web service
Receive Result Set
Validate Data
Send Data
TAPACROSS
Data Curation Layer

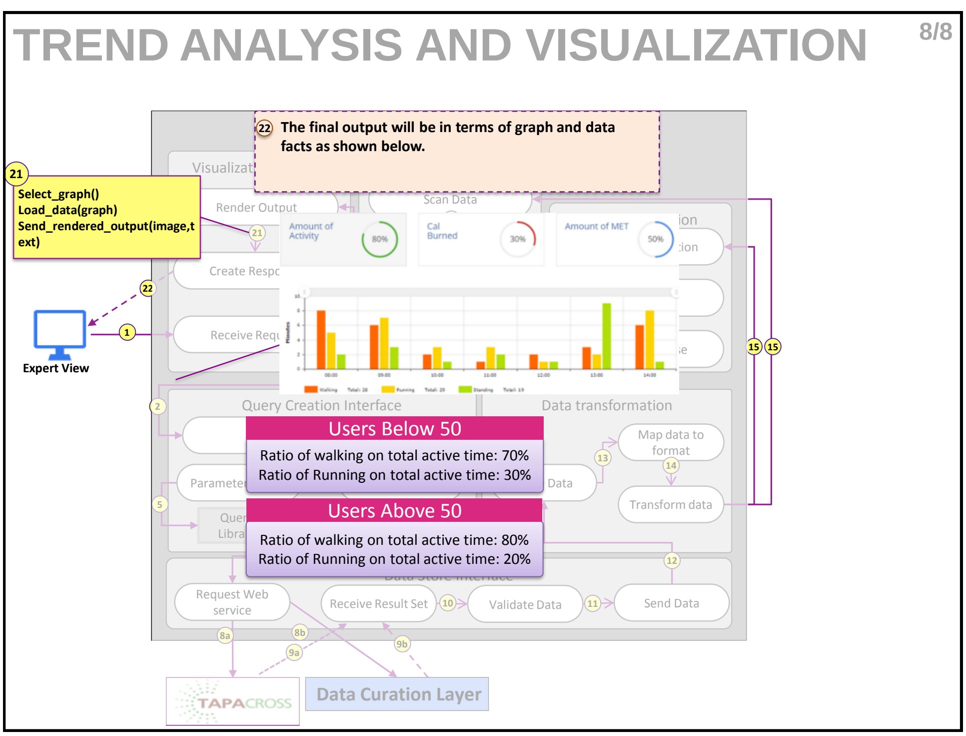


TREND ANALYSIS AND VISUALIZATION
6/8
Reduce_attribute(){
Walking,sitting,.....
}
ve Analytics
ion Clustering
Render Output
Create Response
Receive Request
Scan Data
Attribute Reduction
Group and Create Response
Text Classification
Attribute correlation
Write Text
Create Response
Expert View
Activity Average
Walking 16min
Running 6 min
Sitting 63 min
Activity Cal Burned /MET
Walking 523 80
Running 180 60
Sitting 26 10
15 The transformed JSON is passed for data scanning and attribute correlation.
16 The attributes are correlated (activity) and their respective ratios are calculated.
17 A data fact is written in the form of text.
18 The data is scanned for association and patterns are detected.
19 After the most frequent patterns, attributes are reduced
20 The data is grouped and different analytics are calculated.
Receive Result Set
Validate Data
Send Data
Data Curation Layer
TAPACROSS



TREND ANALYSIS AND VISUALIZATION
7/8
Descriptive Analytics
Visualization Enabler
Association Clustering
Select_graph()
Load_data(graph)
Send_rendered_output(image,text)
Render Output
Create Response
Receive Request
Scan Data
Attribute Reduction
Group and Create Response
Text Classification
Attribute correlation
Write Text
Expert View
Activity Average
Walking 16min
Running 6 min
Sitting 63 min
Activity Calories Burned
Walking 523 80
Running 180 60
Sitting 26 10
15 The transformed JSON is passed for data scanning and attribute correlation.
16 The attributes are correlated (activity) and their respective ratios are calculated.
17 A data fact is written in the form of text.
18 The data is scanned for association and patterns are detected.
19 After the most frequent patterns, attributes are reduced.
20 The data is grouped and different analytics are calculated.
21 The graph is selected, data is mapped on the graph and rendered.
Receive Result Set
Validate Data
Send Data
Data Curation Layer
TAPACROSS

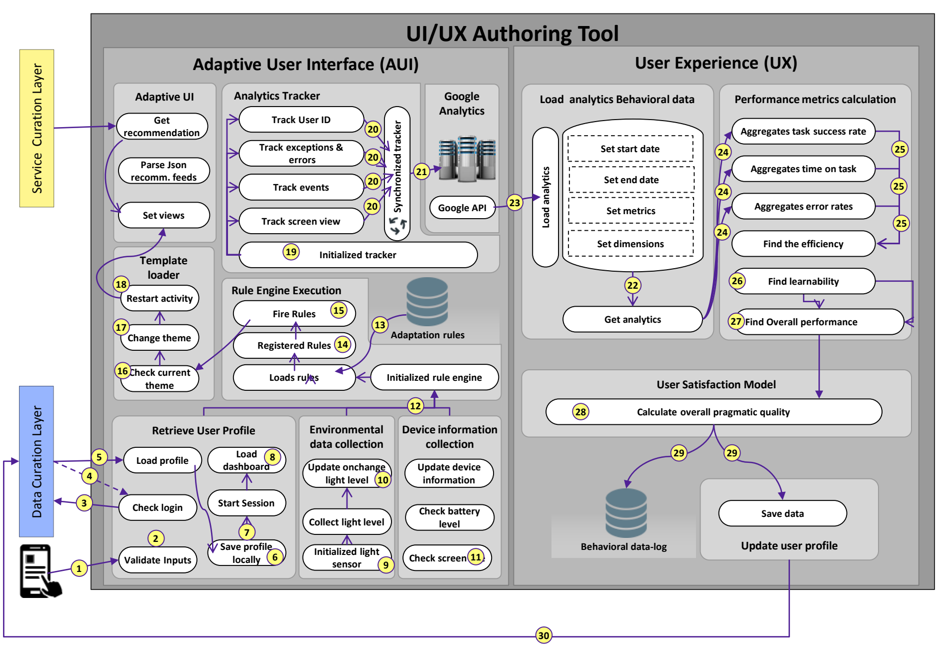

TREND ANALYSIS AND VISUALIZATION
7/8
Descriptive Analytics
Visualization Enabler
Render Output
Create Response
Receive Request
Association Clustering
Scan Data
Attribute Reduction
Group and Create Response
Text Classification
Attribute correlation
Write Text
Create Response
Query Creation Interface
Query attribute breakdown
Parameter Loader
Parse query
Query Library
Select
Data transformation
Map data to format
Parse Data
Transform data
Data Store Interface
Request Web service
Receive Result Set
Validate Data
Send Data
Expert View
TAPACROSS
Data Curation Layer


TREND ANALYSIS AND VISUALIZATION
8/8
22 The final output will be in terms of graph and data facts as shown below.
21
Select_graph()
Load_data(graph)
Send_rendered_output(image,text)
Expert View
Users Below 50
Ratio of walking on total active time: 70%
Ratio of Running on total active time: 30%
Users Above 50
Ratio of walking on total active time: 80%
Ratio of Running on total active time: 20%
Data Curation Layer

STENCILS for SL Micro-Animation

UI/UX Authoring Tool

UI/UX Detailed Architecture

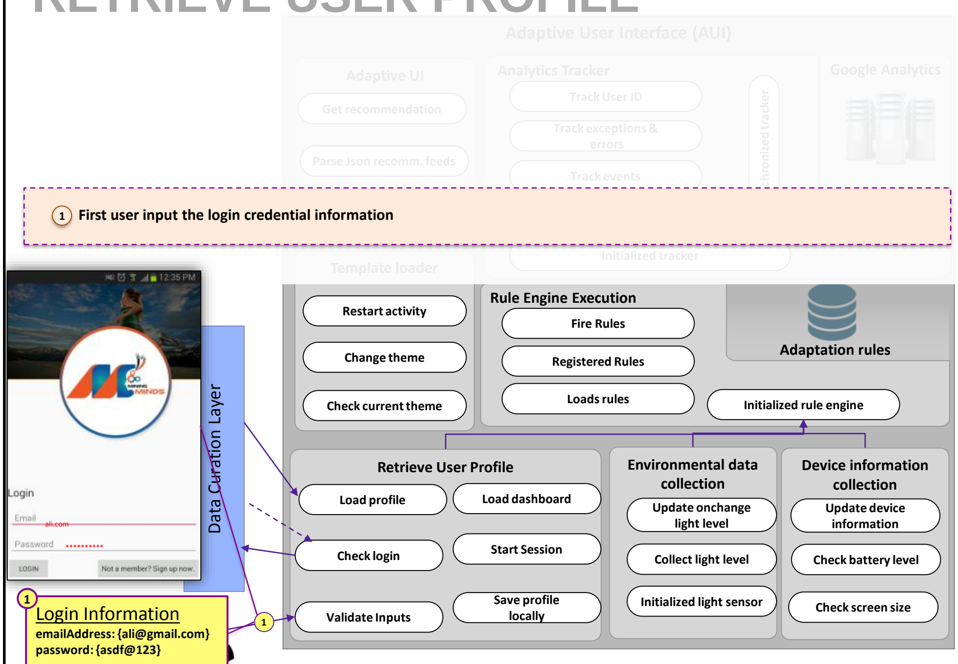


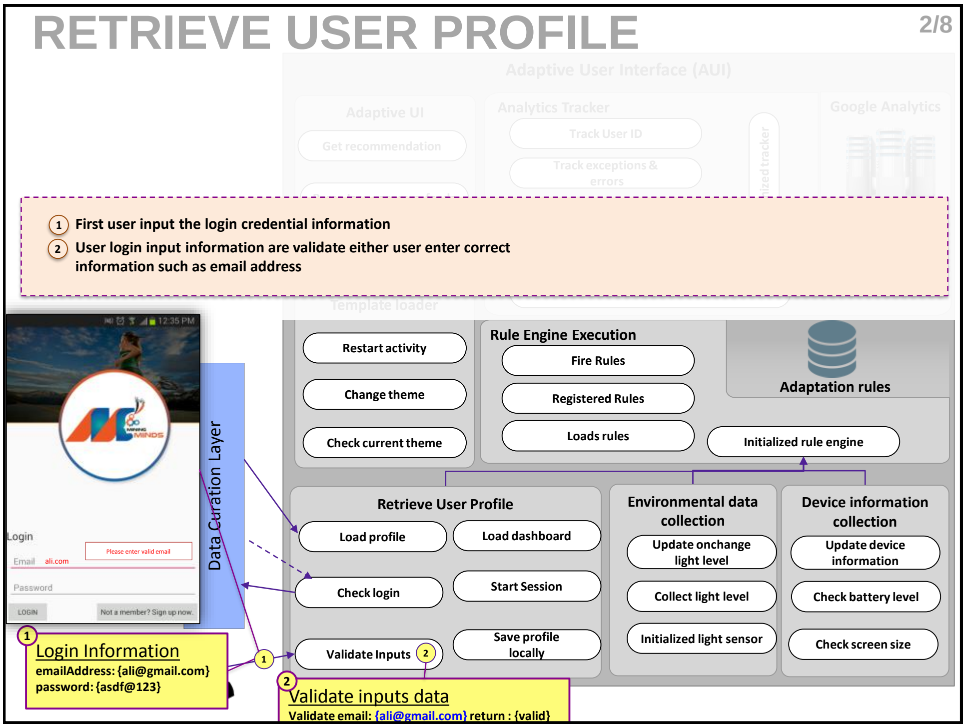
Scenario 1: Adaptive User Interface

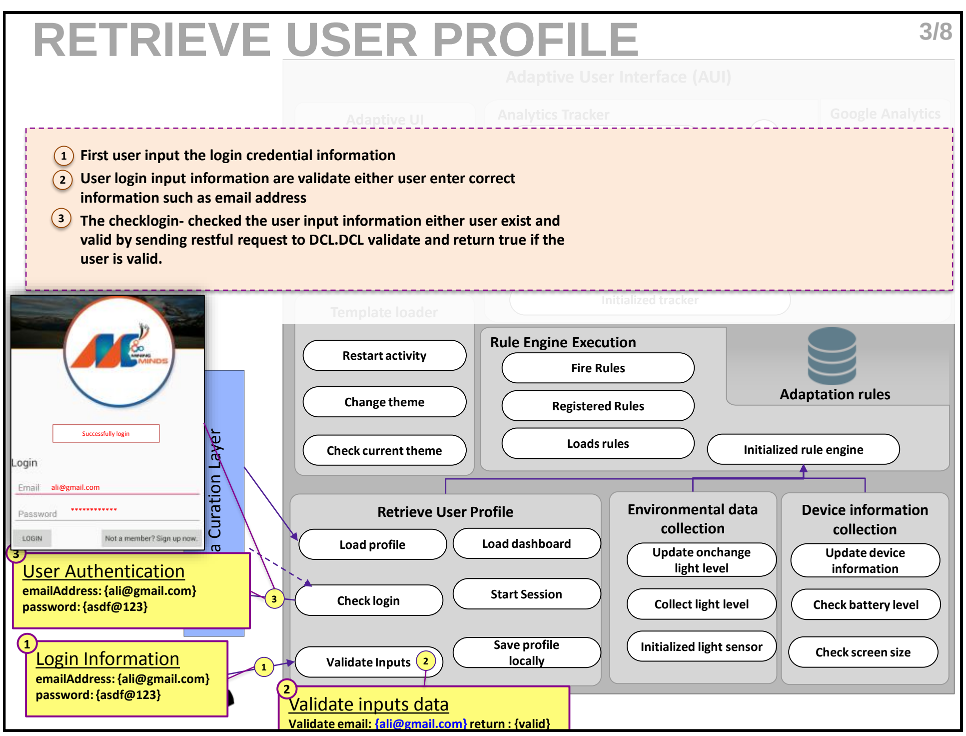
Desc. : **User information**, **environmental** information, and device information are acquired for **rule engine** to **adapt the UI** accordingly.

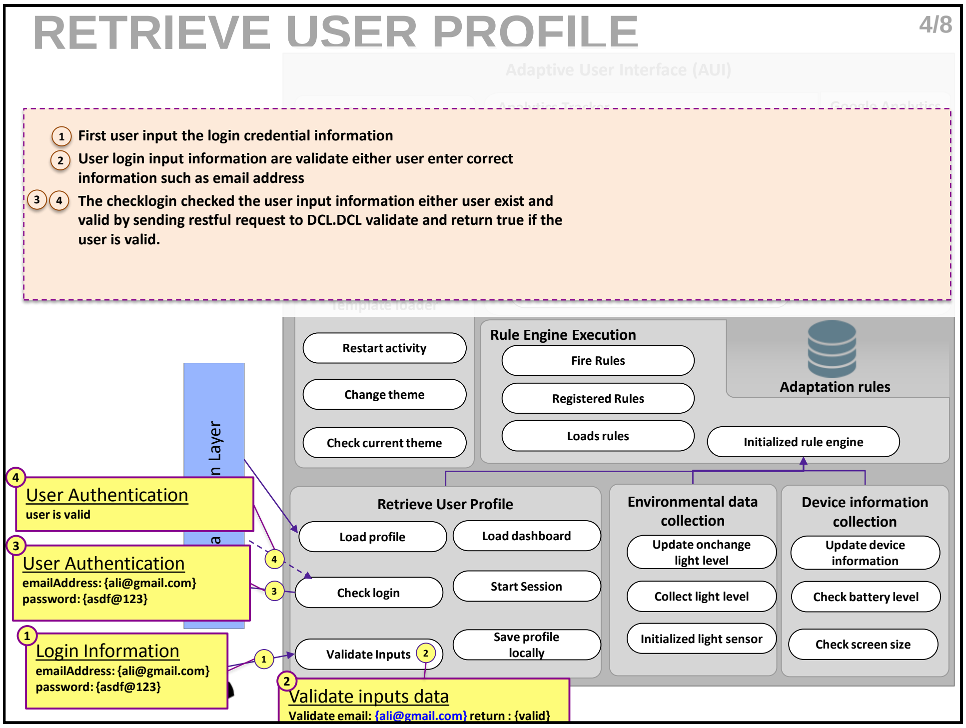
RETRIEVE USER PROFILE

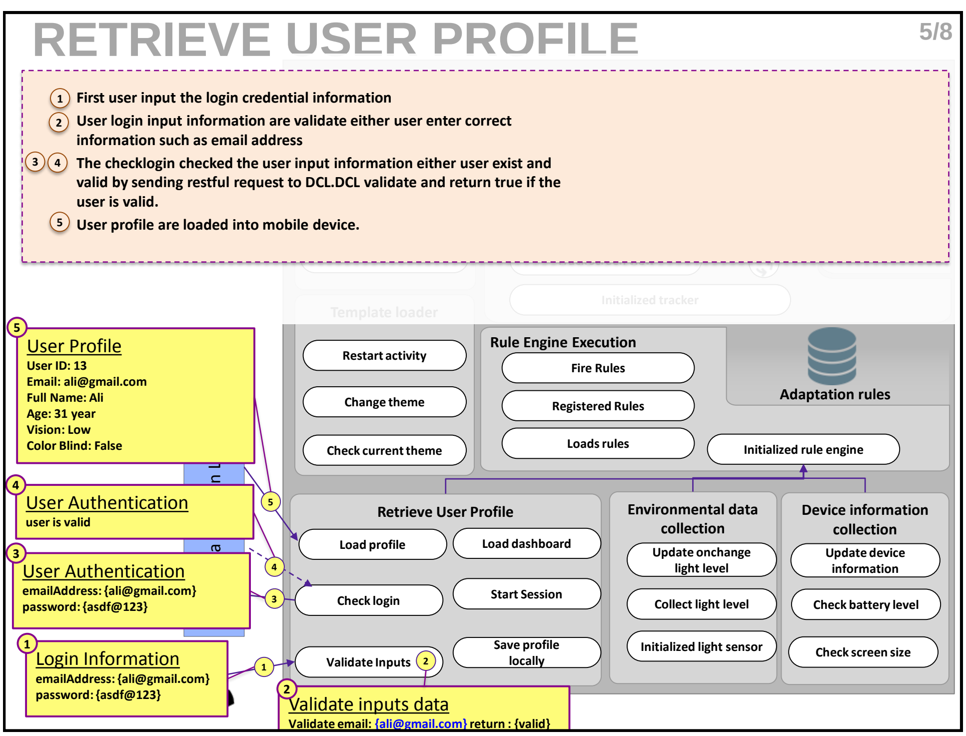
1/8




RETRIEVE USER PROFILE
2/8
Adaptive User Interface (AUI)
Adaptive UI
Get recommendation
Analytics Tracker
Track User ID
Track exceptions & errors
Google Analytics
1 First user input the login credential information
2 User login input information are validate either user enter correct information such as email address
Template loader
Login
Email ali.com
Please enter valid email
Password
LOGIN
Not a member? Sign up now.
Data Curation Layer
Restart activity
Change theme
Check current theme
Rule Engine Execution
Fire Rules
Registered Rules
Loads rules
Adaptation rules
Initialized rule engine
Retrieve User Profile
Load profile
Load dashboard
Check login
Start Session
Validate Inputs
Save profile locally
Environmental data collection
Update onchange light level
Collect light level
Initialized light sensor
Device information collection
Update device information
Check battery level
Check screen size
Login Information
emailAddress: {ali@gmail.com}
password: {asdf@123}
Validate inputs data
Validate email: {ali@gmail.com} return : {valid}


RETRIEVE USER PROFILE
3/8
Adaptive User Interface (AUI)
Adaptive UI
Analytics Tracker
Google Analytics
1 First user input the login credential information
2 User login input information are validate either user enter correct information such as email address
3 The checklogin- checked the user input information either user exist and valid by sending restful request to DCL.DCL validate and return true if the user is valid.
Initialized tracker
Template loader
Successfully login
Login
Email ali@gmail.com
Password
LOGIN
Not a member? Sign up now.
Data Curation Layer
User Authentication
emailAddress: {ali@gmail.com}
password: {asdf@123}
Restart activity
Change theme
Check current theme
Rule Engine Execution
Fire Rules
Registered Rules
Loads rules
Adaptation rules
Initialized rule engine
Retrieve User Profile
Load profile
Load dashboard
Check login
Start Session
Validate Inputs
Save profile locally
Environmental data collection
Update onchange light level
Collect light level
Initialized light sensor
Device information collection
Update device information
Check battery level
Check screen size
Login Information
emailAddress: {ali@gmail.com}
password: {asdf@123}
Validate inputs data
Validate email: {ali@gmail.com} return : {valid}


RETRIEVE USER PROFILE
4/8
1 First user input the login credential information
2 User login input information are validate either user enter correct information such as email address
3 4 The checklogin checked the user input information either user exist and valid by sending restful request to DCL.DCL validate and return true if the user is valid.
Restart activity
Change theme
Check current theme
Rule Engine Execution
Fire Rules
Registered Rules
Loads rules
Adaptation rules
Initialized rule engine
4 User Authentication
user is valid
3 User Authentication
emailAddress: {ali@gmail.com}
password: {asdf@123}
1 Login Information
emailAddress: {ali@gmail.com}
password: {asdf@123}
Retrieve User Profile
Load profile
Load dashboard
Check login
Start Session
Validate Inputs
Save profile locally
Environmental data collection
Update onchange light level
Collect light level
Initialized light sensor
Device information collection
Update device information
Check battery level
Check screen size
2 Validate inputs data
Validate email: {ali@gmail.com} return : {valid}


RETRIEVE USER PROFILE
5/8
1 First user input the login credential information
2 User login input information are validate either user enter correct information such as email address
3 4 The checklogin checked the user input information either user exist and valid by sending restful request to DCL.DCL validate and return true if the user is valid.
5 User profile are loaded into mobile device.
5 User Profile
User ID: 13
Email: ali@gmail.com
Full Name: Ali
Age: 31 year
Vision: Low
Color Blind: False
4 User Authentication
user is valid
3 User Authentication
emailAddress: {ali@gmail.com}
password: {asdf@123}
1 Login Information
emailAddress: {ali@gmail.com}
password: {asdf@123}
Restart activity
Change theme
Check current theme
Rule Engine Execution
Fire Rules
Registered Rules
Loads rules
Adaptation rules
Initialized rule engine
Retrieve User Profile
Load profile
Load dashboard
Check login
Start Session
Validate Inputs
Save profile locally
Environmental data collection
Update onchange light level
Collect light level
Initialized light sensor
Device information collection
Update device information
Check battery level
Check screen size
2 Validate inputs data
Validate email: {ali@gmail.com} return : {valid}

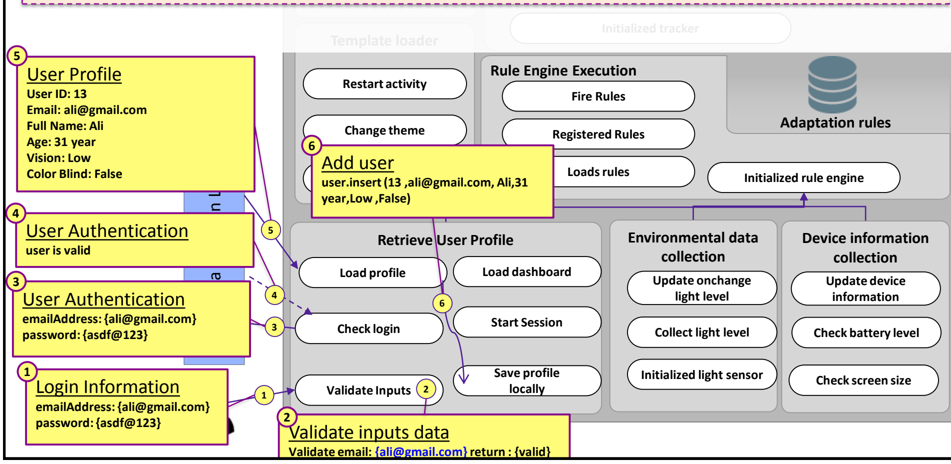
RETRIEVE USER PROFILE

6/8

1. First user input the login credential information
2. User login input information are validate either user enter correct information such as email address

3 4 The checklogin checked the user input information either user exist and valid by sending restful request to DCL.DCL validate and return true if the user is valid.

5. User profile are loaded into mobile device.
6. Then user profile data are locally save in mobile device



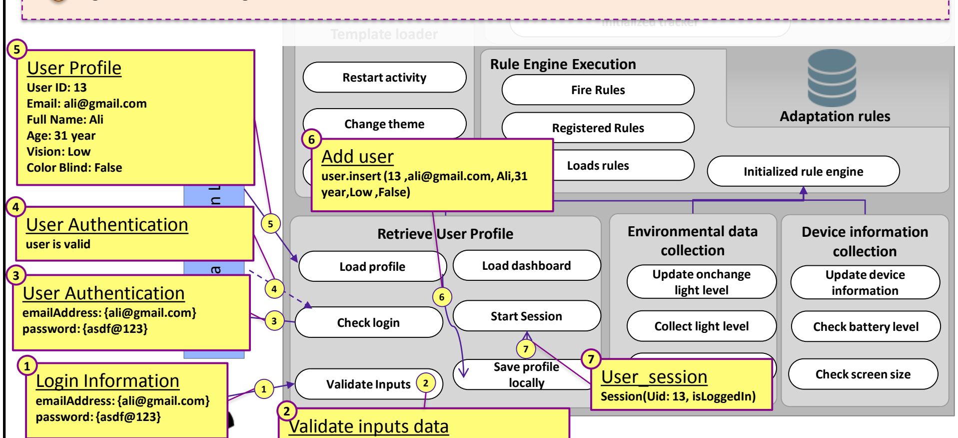
RETRIEVE USER PROFILE

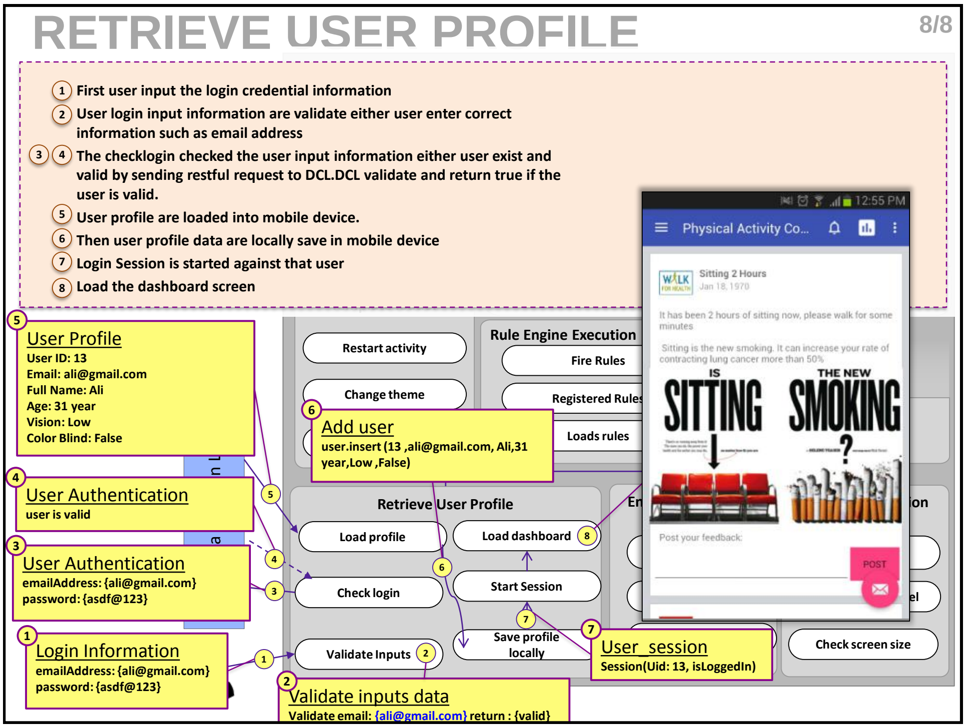
7/8

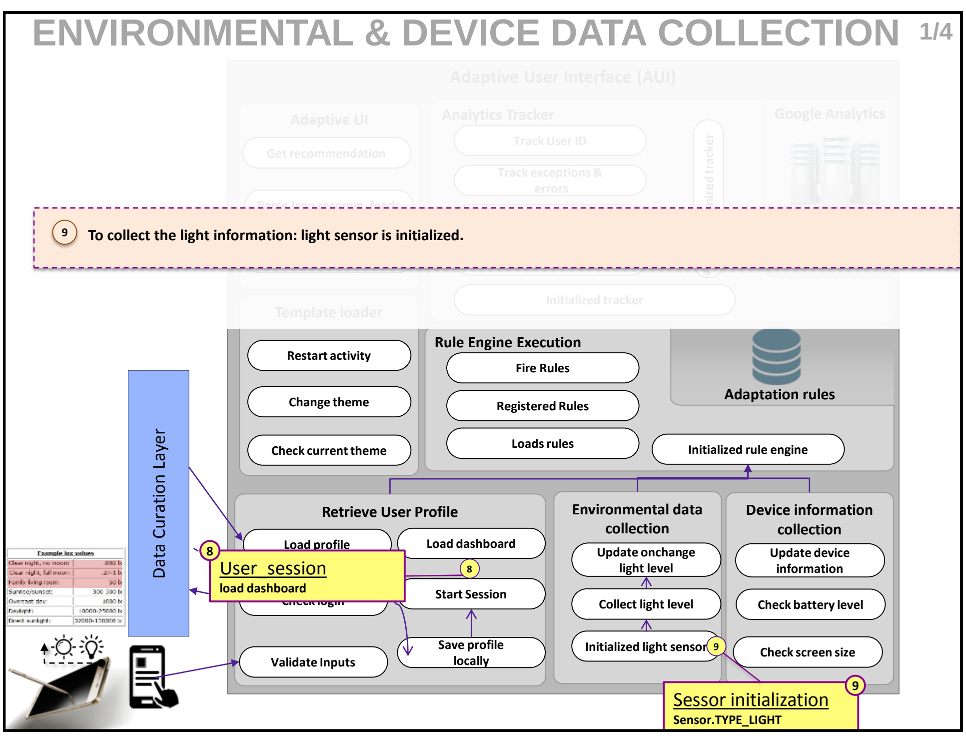
1. First user input the login credential information
2. User login input information are validate either user enter correct information such as email address

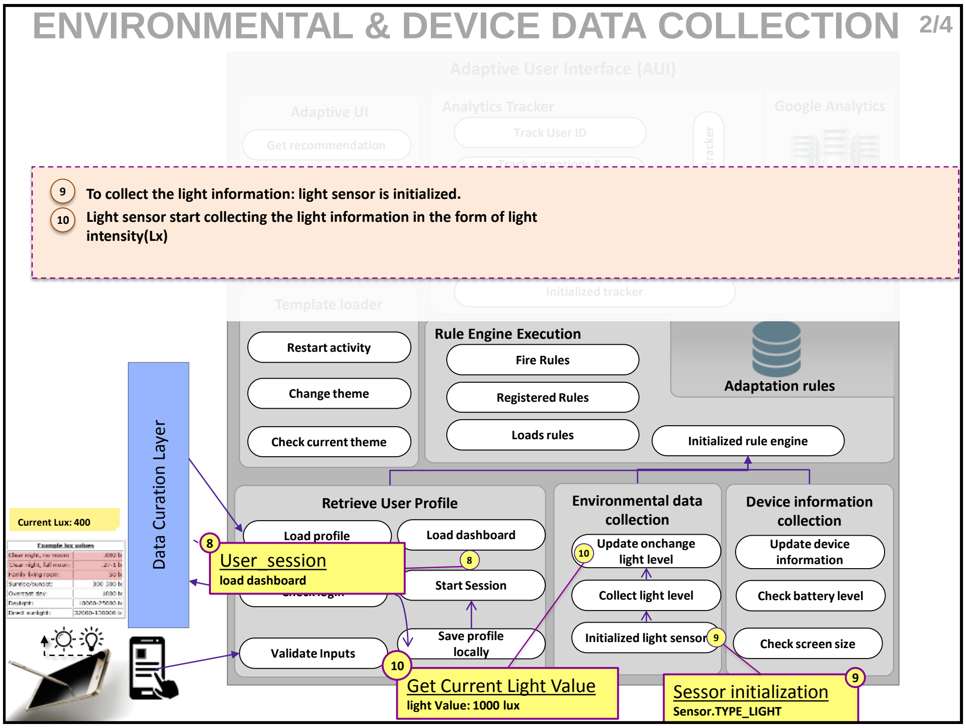
3 4 The checklogin checked the user input information either user exist and valid by sending restful request to DCL.DCL validate and return true if the user is valid.

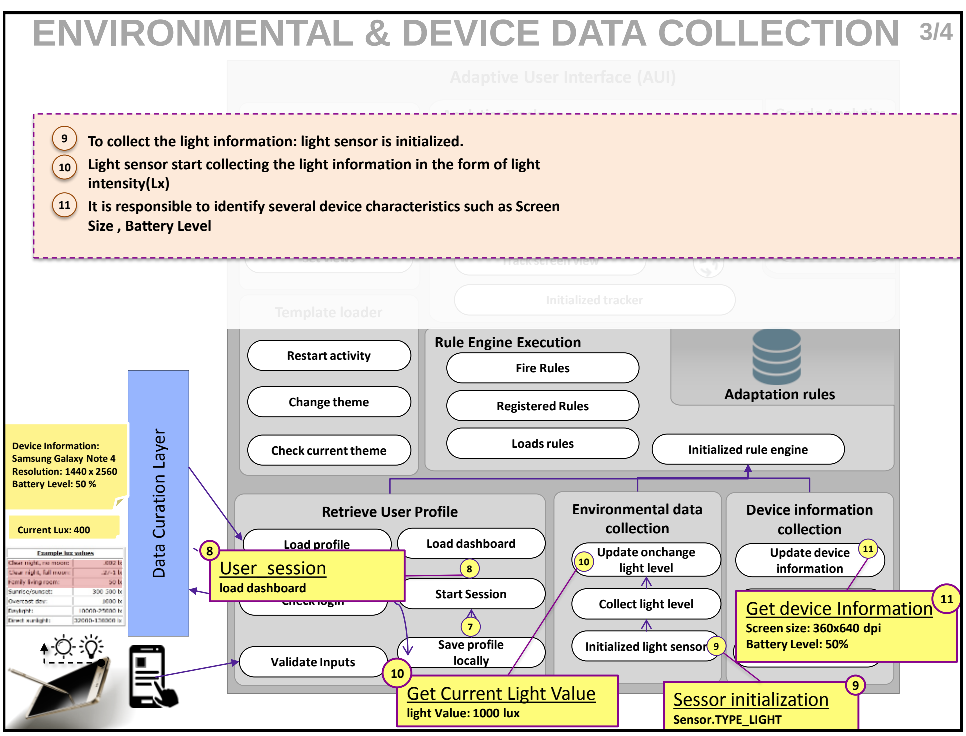
5. User profile are loaded into mobile device.
6. Then user profile data are locally save in mobile device
7. Login Session is started against that user

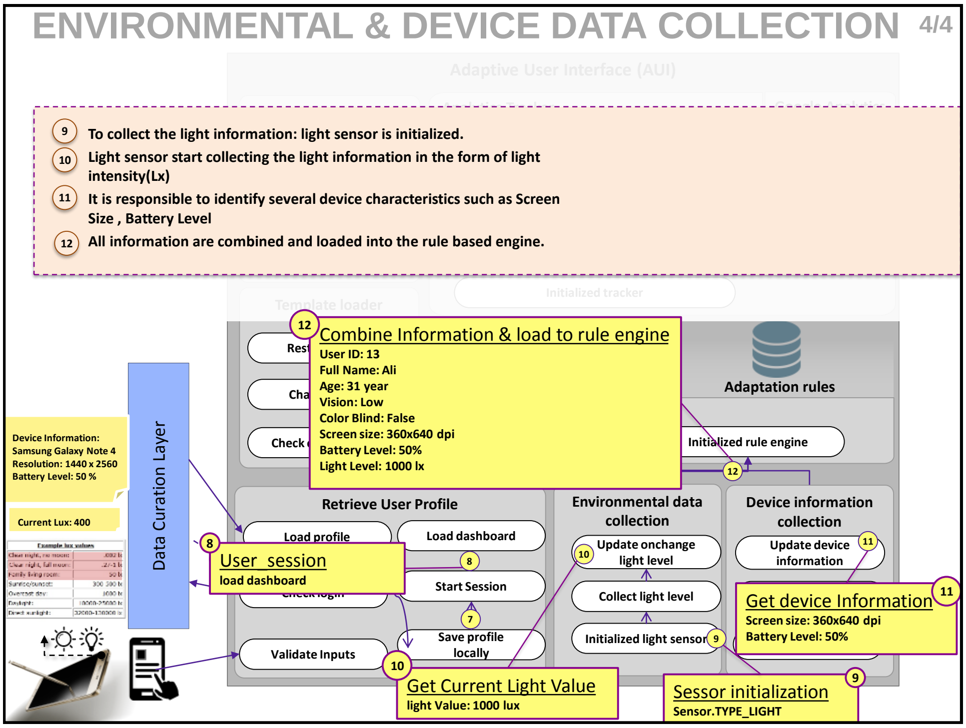



RETRIEVE USER PROFILE
8/8
1 First user input the login credential information
2 User login input information are validate either user enter correct information such as email address
3 4 The checklogin checked the user input information either user exist and valid by sending restful request to DCL.DCL validate and return true if the user is valid.
5 User profile are loaded into mobile device.
6 Then user profile data are locally save in mobile device
7 Login Session is started against that user
8 Load the dashboard screen
5 User Profile
User ID: 13
Email: ali@gmail.com
Full Name: Ali
Age: 31 year
Vision: Low
Color Blind: False
4 User Authentication
user is valid
3 User Authentication
emailAddress: {ali@gmail.com}
password: {asdf@123}
1 Login Information
emailAddress: {ali@gmail.com}
password: {asdf@123}
Restart activity
Change theme
6 Add user
user.insert (13 ,ali@gmail.com, Ali,31 year,Low ,False)
Rule Engine Execution
Fire Rules
Registered Rules
Loads rules
Retrieve User Profile
Load profile
Load dashboard 8
Check login
Start Session
Validate Inputs 2
Save profile locally
7 User_session
Session(Uid: 13, isLoggedIn)
Check screen size
2 Validate inputs data
Validate email: {ali@gmail.com} return : {valid}
12:55 PM
Physical Activity Co...
Sitting 2 Hours
Jan 18, 1970
It has been 2 hours of sitting now, please walk for some minutes
Sitting is the new smoking. It can increase your rate of contracting lung cancer more than 50%
IS SITTING THE NEW SMOKING ?
Post your feedback
POST

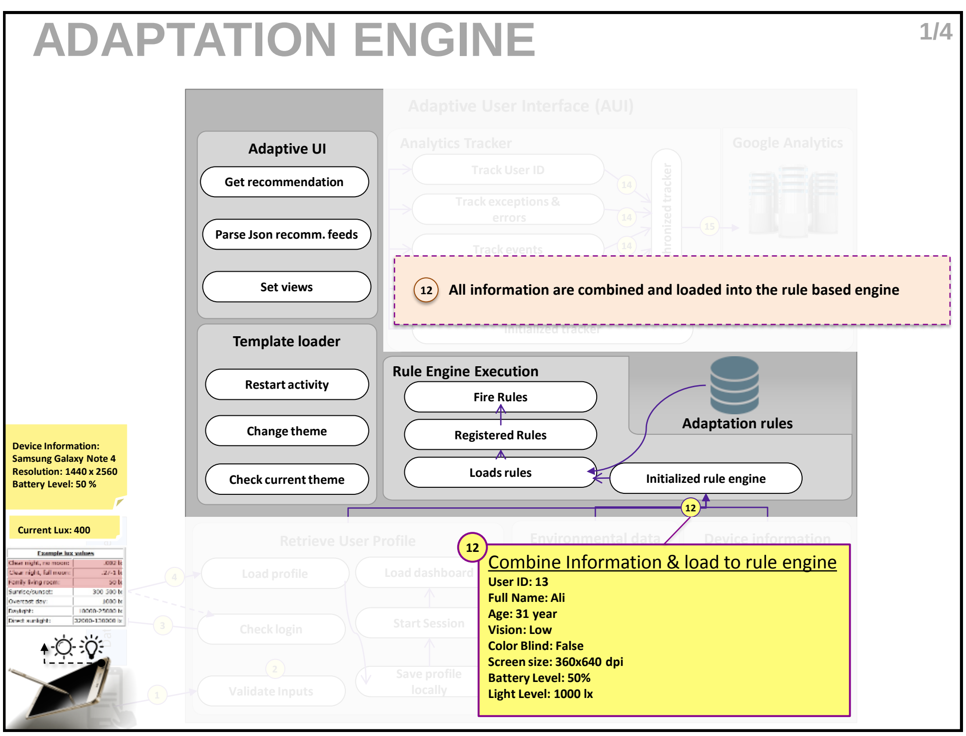

ENVIRONMENTAL & DEVICE DATA COLLECTION
1/4
Adaptive User Interface (AUI)
9 To collect the light information: light sensor is initialized.
Data Curation Layer
Restart activity
Change theme
Check current theme
Rule Engine Execution
Fire Rules
Registered Rules
Loads rules
Adaptation rules
Initialized rule engine
Retrieve User Profile
Load profile
Load dashboard
8 User_session
load dashboard
Start Session
Validate Inputs
Save profile locally
Environmental data collection
Update onchange light level
Collect light level
Initialized light sensor 9
Device information collection
Update device information
Check battery level
Check screen size
9 Sessor initialization
Sensor.TYPE_LIGHT


ENVIRONMENTAL & DEVICE DATA COLLECTION 2/4
9 To collect the light information: light sensor is initialized.
10 Light sensor start collecting the light information in the form of light intensity(Lx)
Restart activity
Change theme
Check current theme
Rule Engine Execution
Fire Rules
Registered Rules
Loads rules
Adaptation rules
Initialized rule engine
Data Curation Layer
Current Lux: 400
Retrieve User Profile
Load profile
Load dashboard
User_session
load dashboard
Start Session
Save profile locally
Validate Inputs
Environmental data collection
Update onchange light level
Collect light level
Initialized light sensor
Device information collection
Update device information
Check battery level
Check screen size
Get Current Light Value
light Value: 1000 lux
Sessor initialization
Sensor.TYPE_LIGHT

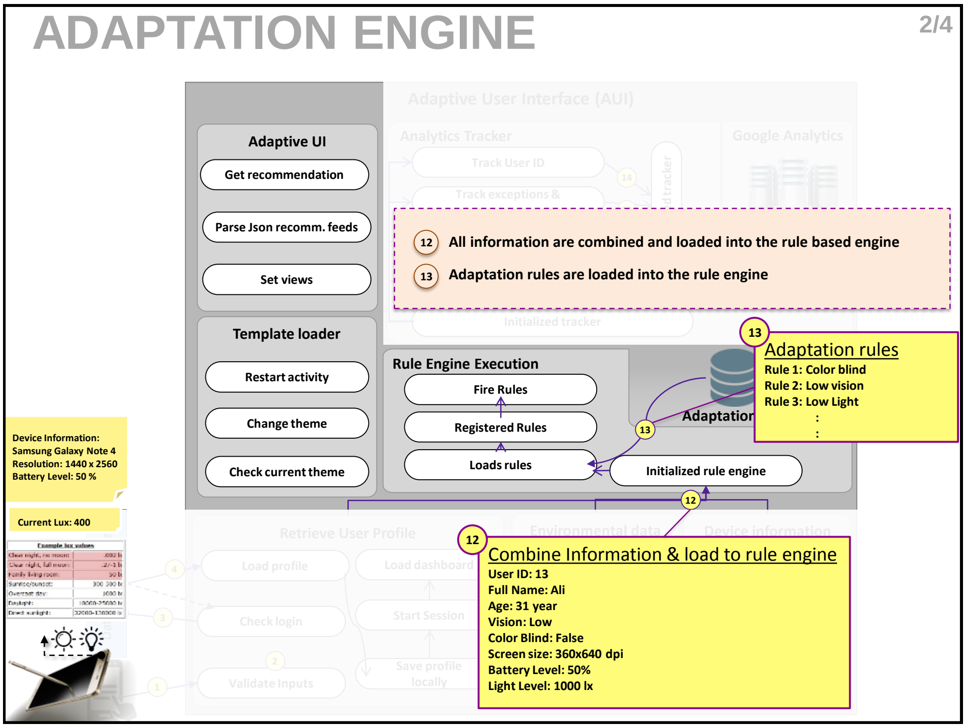

ENVIRONMENTAL & DEVICE DATA COLLECTION 3/4
9 To collect the light information: light sensor is initialized.
10 Light sensor start collecting the light information in the form of light intensity(Lx)
11 It is responsible to identify several device characteristics such as Screen Size , Battery Level
Restart activity
Change theme
Check current theme
Rule Engine Execution
Fire Rules
Registered Rules
Loads rules
Adaptation rules
Initialized rule engine
Device Information:
Samsung Galaxy Note 4
Resolution: 1440 x 2560
Battery Level: 50 %
Current Lux: 400
Data Curation Layer
Retrieve User Profile
Load profile
Load dashboard
User_session
load dashboard
Start Session
Save profile locally
Validate Inputs
Environmental data collection
Update onchange light level
Collect light level
Initialized light sensor
Device information collection
Update device information
Get device Information
Screen size: 360x640 dpi
Battery Level: 50%
Get Current Light Value
light Value: 1000 lux
Sessor initialization
Sensor.TYPE_LIGHT

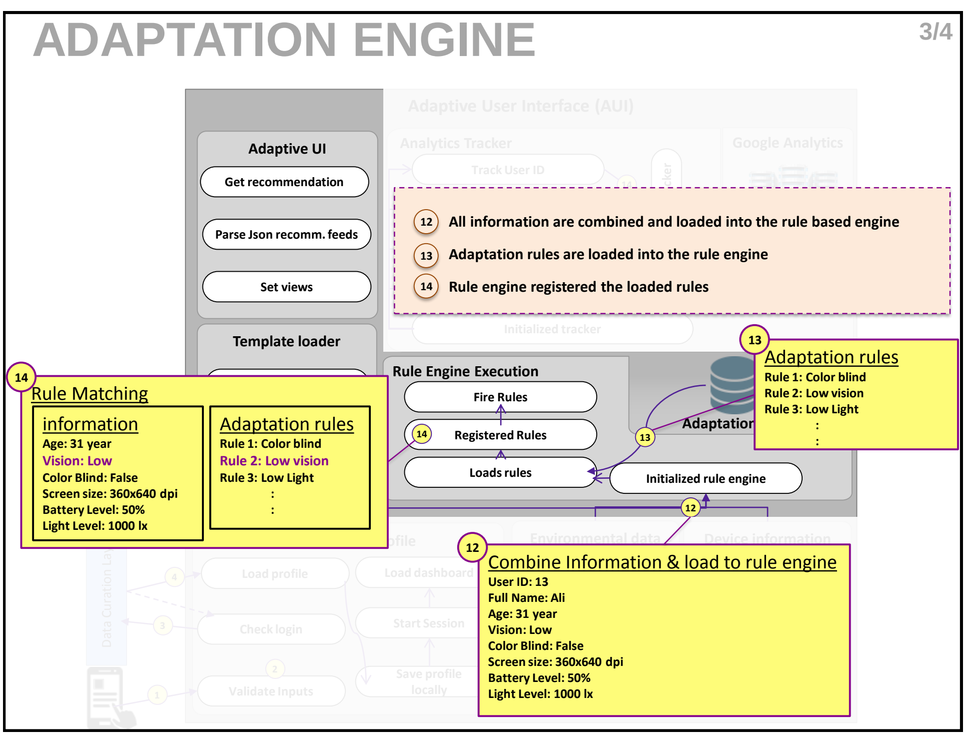


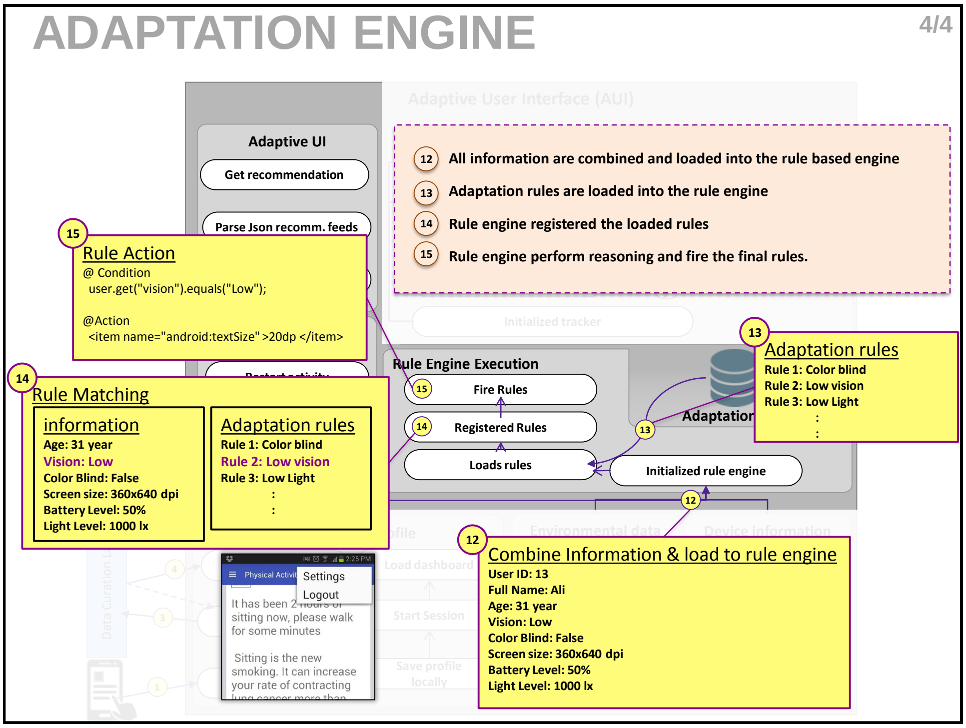
ENVIRONMENTAL & DEVICE DATA COLLECTION
4/4
9 To collect the light information: light sensor is initialized.
10 Light sensor start collecting the light information in the form of light intensity(Lx)
11 It is responsible to identify several device characteristics such as Screen Size , Battery Level
12 All information are combined and loaded into the rule based engine.
Combine Information & load to rule engine
User ID: 13
Full Name: Ali
Age: 31 year
Vision: Low
Color Blind: False
Screen size: 360x640 dpi
Battery Level: 50%
Light Level: 1000 lx
Adaptation rules
Initialized rule engine
Device Information:
Samsung Galaxy Note 4
Resolution: 1440 x 2560
Battery Level: 50 %
Current Lux: 400
Data Curation Layer
Retrieve User Profile
Load profile
Load dashboard
User session
load dashboard
Start Session
Save profile locally
Validate Inputs
Environmental data collection
Update onchange light level
Collect light level
Initialized light sensor
Device information collection
Update device information
Get device Information
Screen size: 360x640 dpi
Battery Level: 50%
Get Current Light Value
light Value: 1000 lux
Sessor initialization
Sensor.TYPE_LIGHT

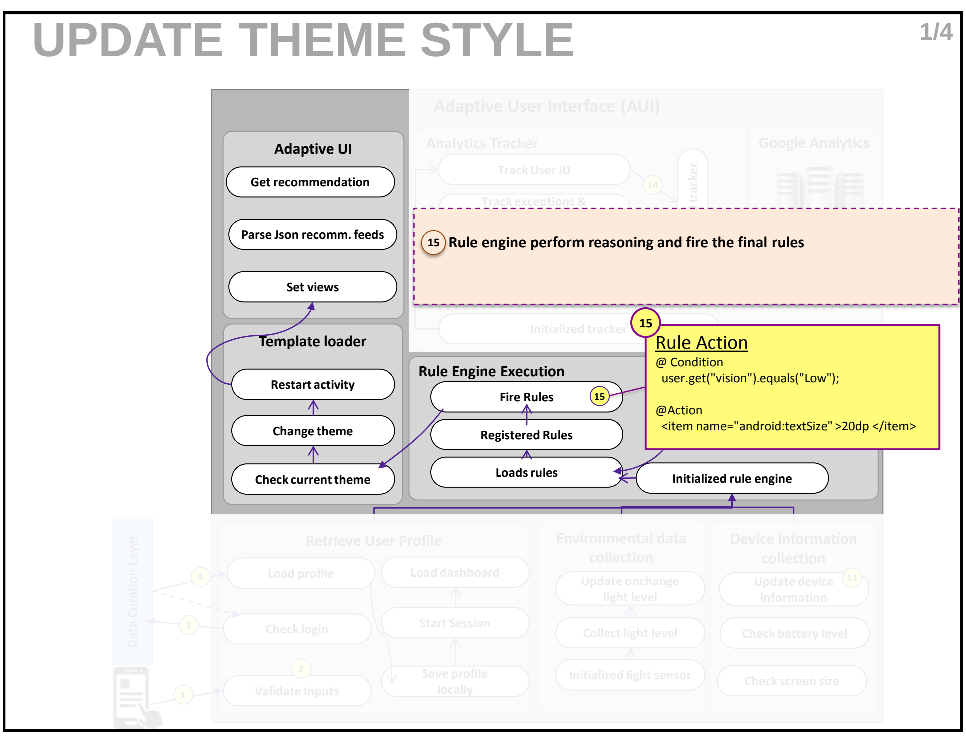


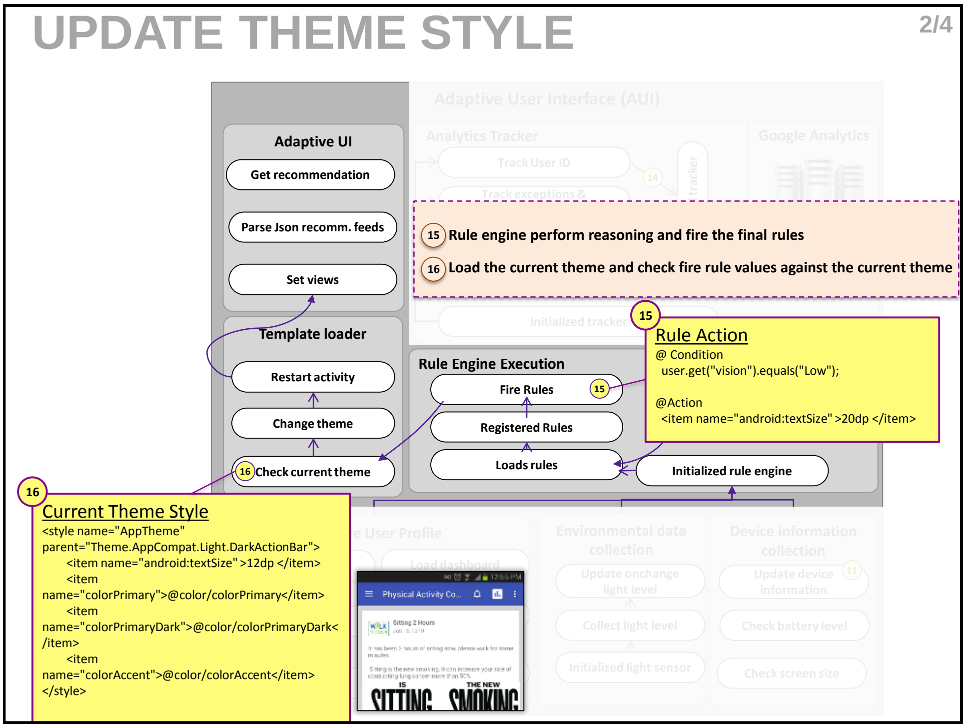
ADAPTATION ENGINE
1/4
Adaptive UI
Get recommendation
Parse Json recomm. feeds
Set views
Template loader
Restart activity
Change theme
Check current theme
12 All information are combined and loaded into the rule based engine
Rule Engine Execution
Fire Rules
Registered Rules
Loads rules
Adaptation rules
Initialized rule engine
Combine Information & load to rule engine
User ID: 13
Full Name: Ali
Age: 31 year
Vision: Low
Color Blind: False
Screen size: 360x640 dpi
Battery Level: 50%
Light Level: 1000 lx
Device Information:
Samsung Galaxy Note 4
Resolution: 1440 x 2560
Battery Level: 50 %
Current Lux: 400

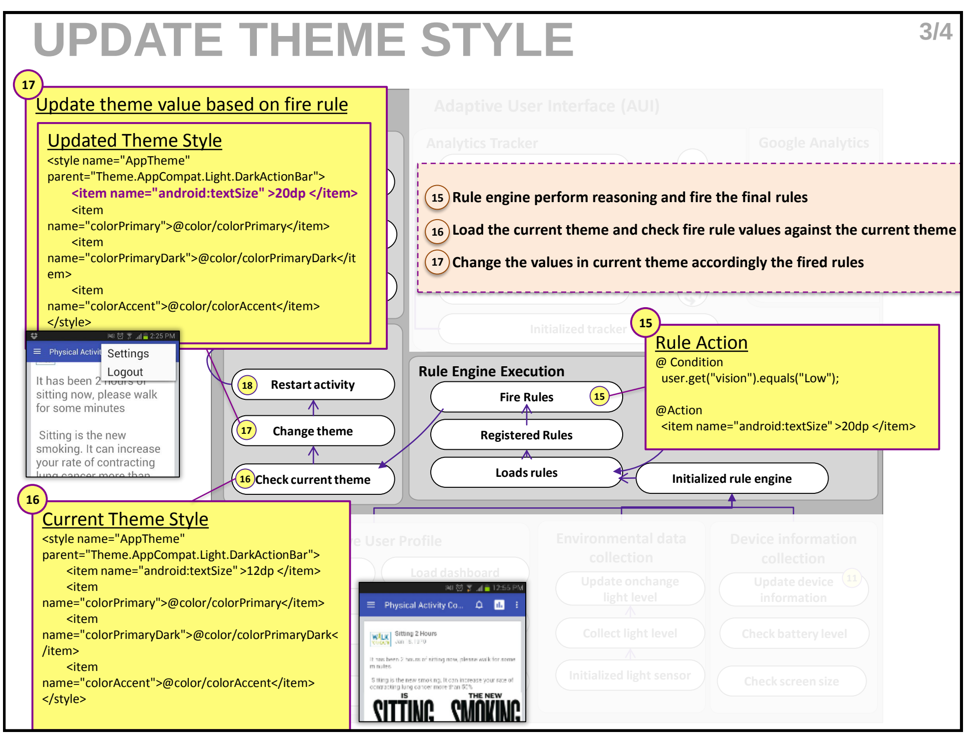

ADAPTATION ENGINE
2/4
Adaptive UI
Get recommendation
Parse Json recomm. feeds
Set views
Template loader
Restart activity
Change theme
Check current theme
12 All information are combined and loaded into the rule based engine
13 Adaptation rules are loaded into the rule engine
Rule Engine Execution
Fire Rules
Registered Rules
Loads rules
Initialized rule engine
Adaptation
13 Adaptation rules
Rule 1: Color blind
Rule 2: Low vision
Rule 3: Low Light
Device Information:
Samsung Galaxy Note 4
Resolution: 1440 x 2560
Battery Level: 50 %
Current Lux: 400
12 Combine Information & load to rule engine
User ID: 13
Full Name: Ali
Age: 31 year
Vision: Low
Color Blind: False
Screen size: 360x640 dpi
Battery Level: 50%
Light Level: 1000 lx


ADAPTATION ENGINE
3/4
Adaptive UI
Get recommendation
Parse Json recomm. feeds
Set views
Template loader
12 All information are combined and loaded into the rule based engine
13 Adaptation rules are loaded into the rule engine
14 Rule engine registered the loaded rules
Rule Engine Execution
Fire Rules
14 Registered Rules
Loads rules
Initialized rule engine
Adaptation
13 Adaptation rules
Rule 1: Color blind
Rule 2: Low vision
Rule 3: Low Light
14 Rule Matching
information
Age: 31 year
Vision: Low
Color Blind: False
Screen size: 360x640 dpi
Battery Level: 50%
Light Level: 1000 lx
Adaptation rules
Rule 1: Color blind
Rule 2: Low vision
Rule 3: Low Light
12 Combine Information & load to rule engine
User ID: 13
Full Name: Ali
Age: 31 year
Vision: Low
Color Blind: False
Screen size: 360x640 dpi
Battery Level: 50%
Light Level: 1000 lx


ADAPTATION ENGINE
4/4
Adaptive UI
Get recommendation
Parse Json recomm. feeds
12 All information are combined and loaded into the rule based engine
13 Adaptation rules are loaded into the rule engine
14 Rule engine registered the loaded rules
15 Rule engine perform reasoning and fire the final rules.
15 Rule Action
@ Condition
user.get("vision").equals("Low");
@Action
<item name="android:textSize" >20dp </item>
14 Rule Matching
information
Age: 31 year
Vision: Low
Color Blind: False
Screen size: 360x640 dpi
Battery Level: 50%
Light Level: 1000 lx
Adaptation rules
Rule 1: Color blind
Rule 2: Low vision
Rule 3: Low Light
Rule Engine Execution
Fire Rules
Registered Rules
Loads rules
Initialized rule engine
13 Adaptation rules
Rule 1: Color blind
Rule 2: Low vision
Rule 3: Low Light
12 Combine Information & load to rule engine
User ID: 13
Full Name: Ali
Age: 31 year
Vision: Low
Color Blind: False
Screen size: 360x640 dpi
Battery Level: 50%
Light Level: 1000 lx
Settings
Logout
It has been 2 hours of sitting now, please walk for some minutes
Sitting is the new smoking. It can increase your rate of contracting

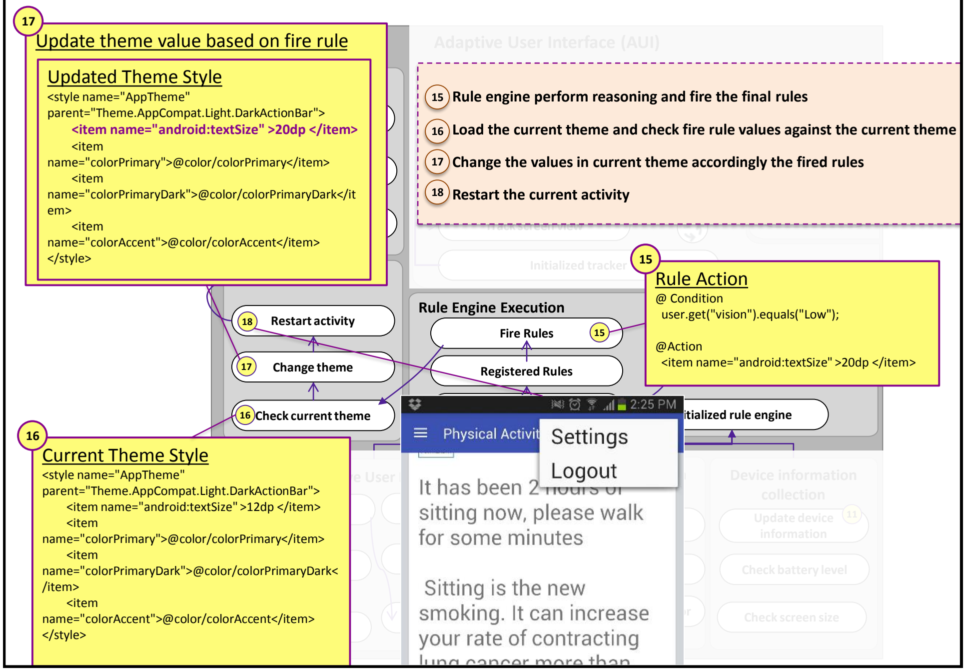

UPDATE THEME STYLE
1/4
Adaptive UI
Get recommendation
Parse Json recomm. feeds
Set views
Template loader
Restart activity
Change theme
Check current theme
15 Rule engine perform reasoning and fire the final rules
15 Rule Action
@ Condition
user.get("vision").equals("Low");
@Action
<item name="android:textSize" >20dp </item>
Rule Engine Execution
Fire Rules
Registered Rules
Loads rules
Initialized rule engine


UPDATE THEME STYLE
2/4
Adaptive UI
Get recommendation
Parse Json recomm. feeds
Set views
Template loader
Restart activity
Change theme
16 Check current theme
15 Rule engine perform reasoning and fire the final rules
16 Load the current theme and check fire rule values against the current theme
15
Rule Action
@ Condition
user.get("vision").equals("Low");
@Action
<item name="android:textSize" >20dp </item>
Rule Engine Execution
Fire Rules 15
Registered Rules
Loads rules
Initialized rule engine
16
Current Theme Style
<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
<item name="android:textSize" >12dp </item>
<item name="colorPrimary">@color/colorPrimary</item>
<item name="colorPrimaryDark">@color/colorPrimaryDark</item>
<item name="colorAccent">@color/colorAccent</item>
</style>


UPDATE THEME STYLE
3/4
17
Update theme value based on fire rule
Updated Theme Style
<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
<item name="android:textSize" >20dp </item>
<item name="colorPrimary">@color/colorPrimary</item>
<item name="colorPrimaryDark">@color/colorPrimaryDark</item>
<item name="colorAccent">@color/colorAccent</item>
</style>
Settings
Logout
15 Rule engine perform reasoning and fire the final rules
16 Load the current theme and check fire rule values against the current theme
17 Change the values in current theme accordingly the fired rules
15
Rule Action
@ Condition
user.get("vision").equals("Low");
@Action
<item name="android:textSize" >20dp </item>
18 Restart activity
17 Change theme
16 Check current theme
Rule Engine Execution
Fire Rules 15
Registered Rules
Loads rules
Initialized rule engine
16
Current Theme Style
<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
<item name="android:textSize" >12dp </item>
<item name="colorPrimary">@color/colorPrimary</item>
<item name="colorPrimaryDark">@color/colorPrimaryDark</item>
<item name="colorAccent">@color/colorAccent</item>
</style>

UPDATE THEME STYLE

4/4



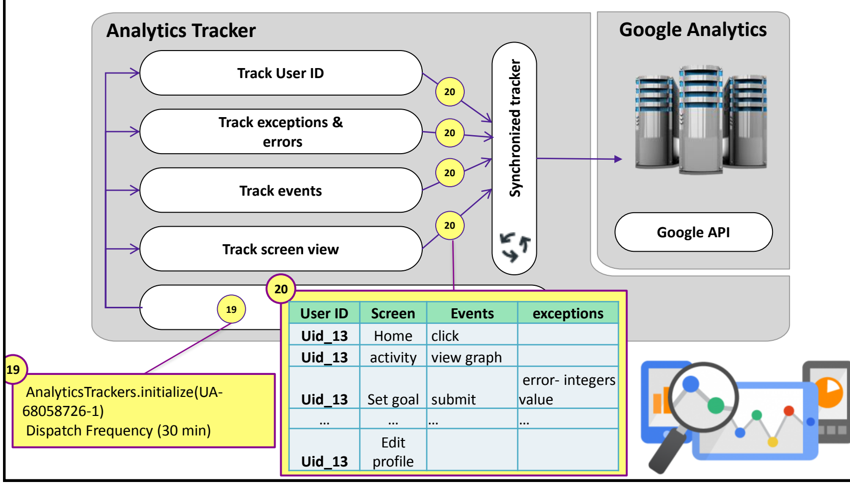
Scenario 2: User Experience evaluation

Desc.: **User experience evaluation** based on user **interaction** data that collects by **analytics tracker**, to finding the different **usability metric**. Finally the **user satisfaction** is calculating based on those metrics.

ANALYTICS TRACKER

1/3

(19) Google analytics account tracker ID is added in tracker to initialized the tracker



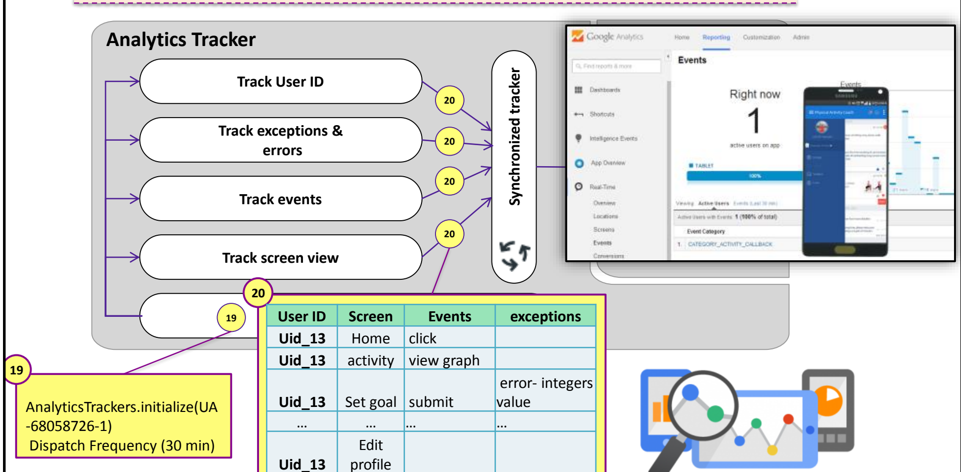
User ID	Screen	Events	exceptions
Uid_13	Home	click	
Uid_13	activity	view graph	
Uid_13	Set goal	submit	error- integers value
…	…	…	…
Uid_13	Edit profile		

ANALYTICS TRACKER

2/3

(19) Google analytics account tracker ID is added in tracker to initialized the tracker

(20) Tracker start tracking the user, events, screen views of the login user



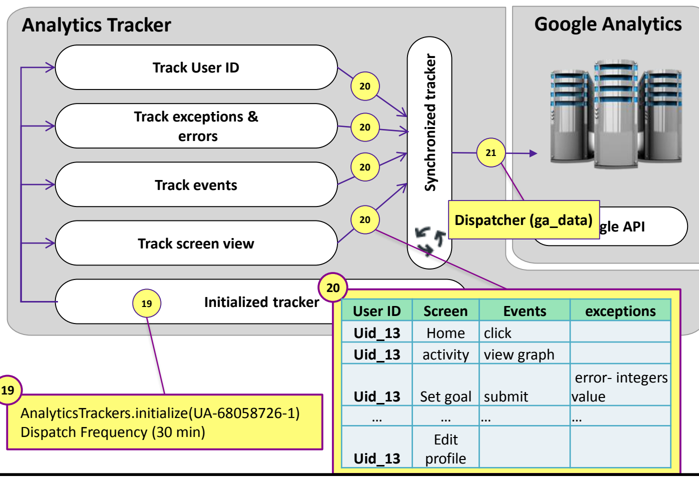
User ID	Screen	Events	exceptions
Uid_13	Home	click	
Uid_13	activity	view graph	
Uid_13	Set goal	submit	error- integers value
…	…	…	…
Uid_13	Edit profile		

ANALYTICS TRACKER

19 Google analytics account tracker ID is added in tracker to initialized the tracker

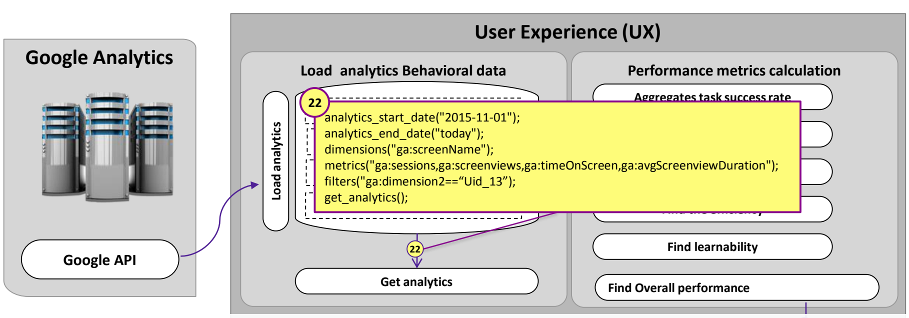
20 Tracker start tracking the user, events, screen views of the login user

21 Tracker synchronized the data by calling the dispatch function with google analytics server.

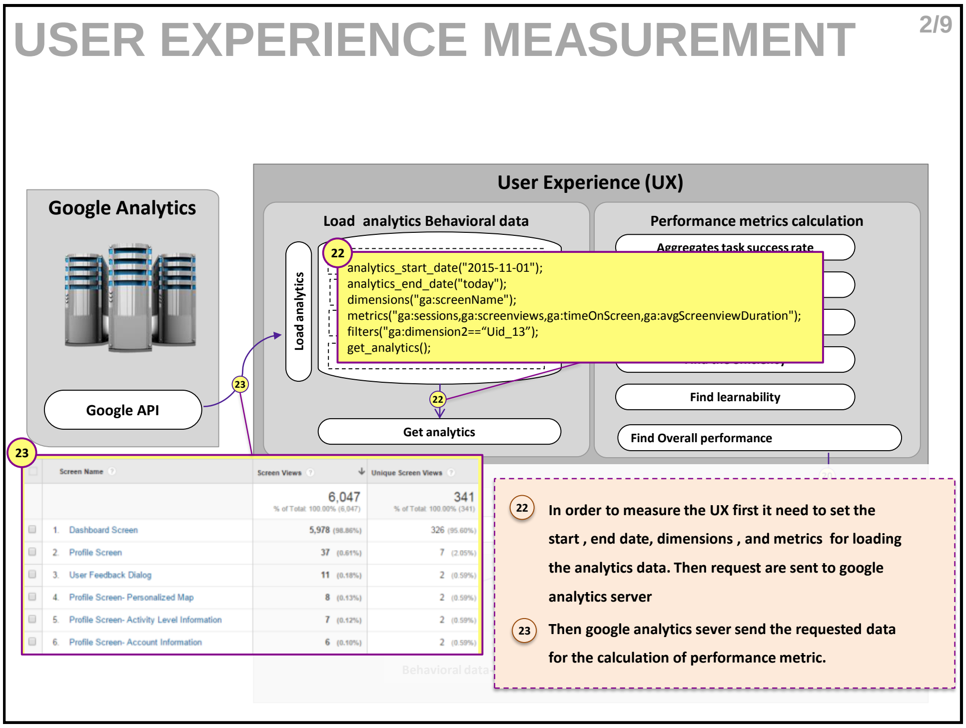


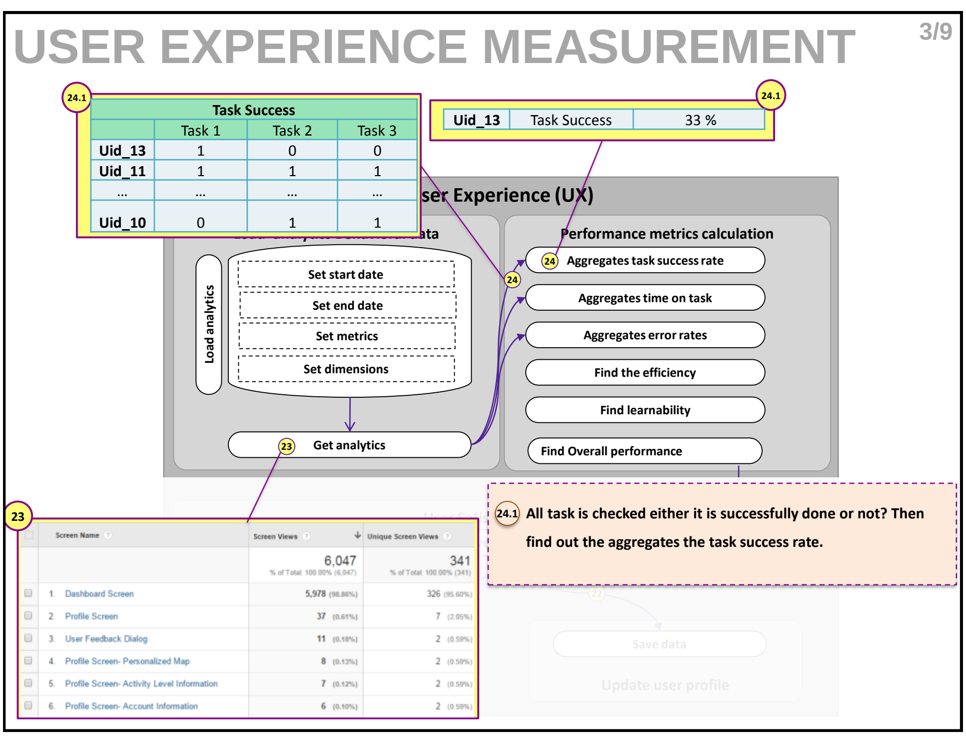
User ID	Screen	Events	exceptions
Uid_13	Home	click	
Uid_13	activity	view graph	
Uid_13	Set goal	submit	error- integers value
…	…	…	…
Uid_13	Edit profile		

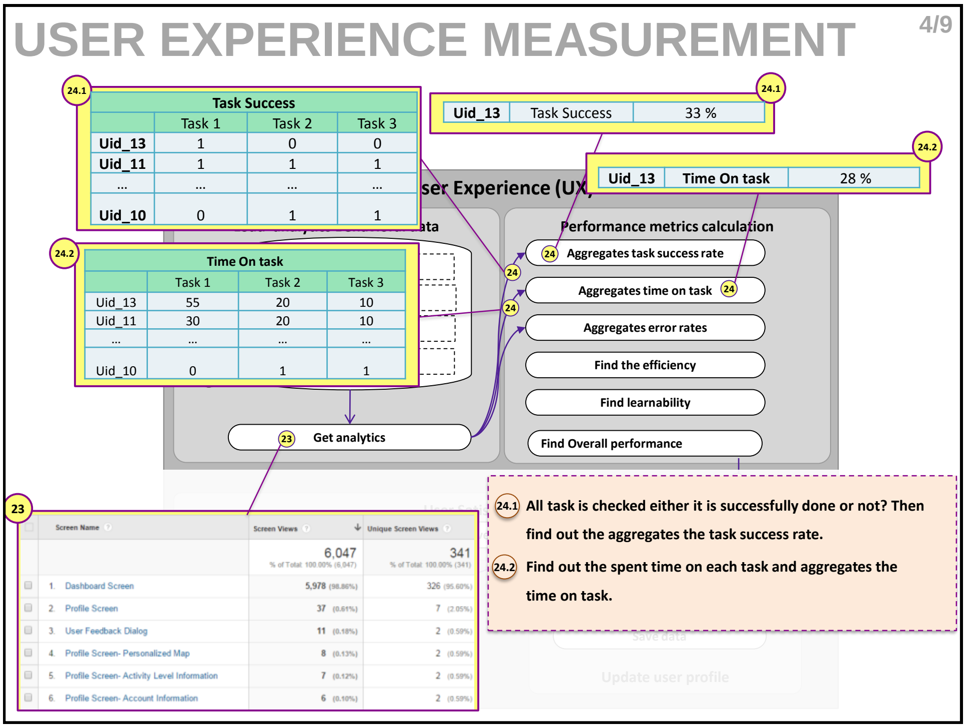
USER EXPERIENCE MEASUREMENT



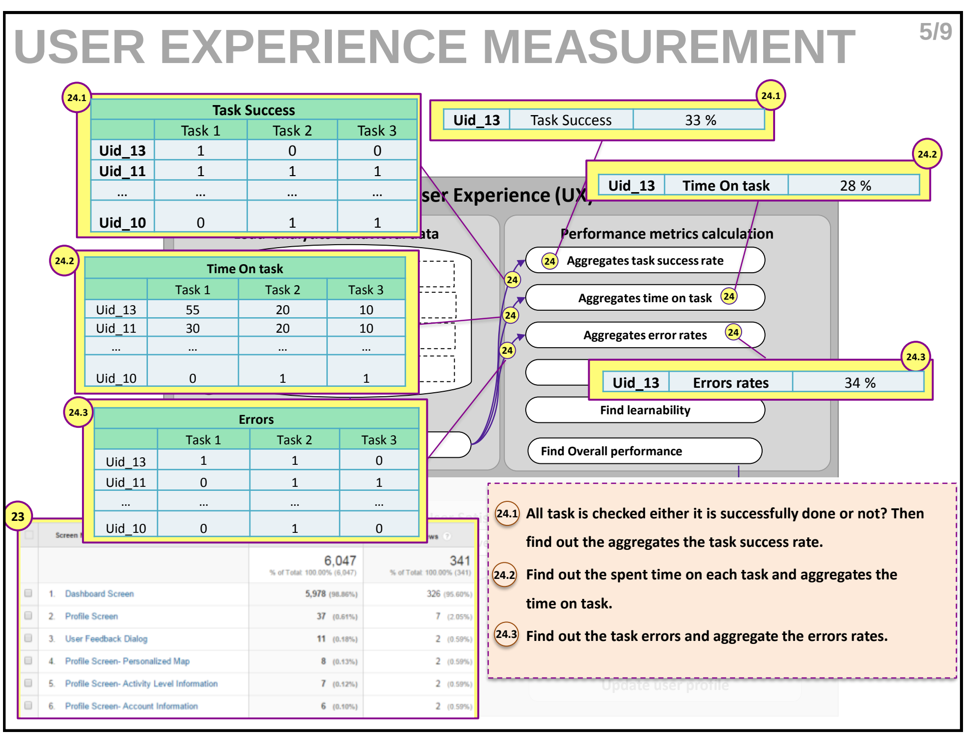
22 In order to measure the UX first it need to set the start , end date, dimensions , and metrics for loading the analytics data. Then request are sent to google analytics server


USER EXPERIENCE MEASUREMENT
2/9
Google Analytics
Google API
User Experience (UX)
Load analytics Behavioral data
Load analytics
analytics_start_date("2015-11-01");
analytics_end_date("today");
dimensions("ga:screenName");
metrics("ga:sessions,ga:screenviews,ga:timeOnScreen,ga:avgScreenviewDuration");
filters("ga:dimension2=="Uid_13");
get_analytics();
Get analytics
Performance metrics calculation
Aggregates task success rate
Find learnability
Find Overall performance
Screen Name
Screen Views
Unique Screen Views
6,047
341
1. Dashboard Screen 5,978 326
2. Profile Screen 37 7
3. User Feedback Dialog 11 2
4. Profile Screen- Personalized Map 8 2
5. Profile Screen- Activity Level Information 7 2
6. Profile Screen- Account Information 6 2
22 In order to measure the UX first it need to set the start , end date, dimensions , and metrics for loading the analytics data. Then request are sent to google analytics server
23 Then google analytics sever send the requested data for the calculation of performance metric.

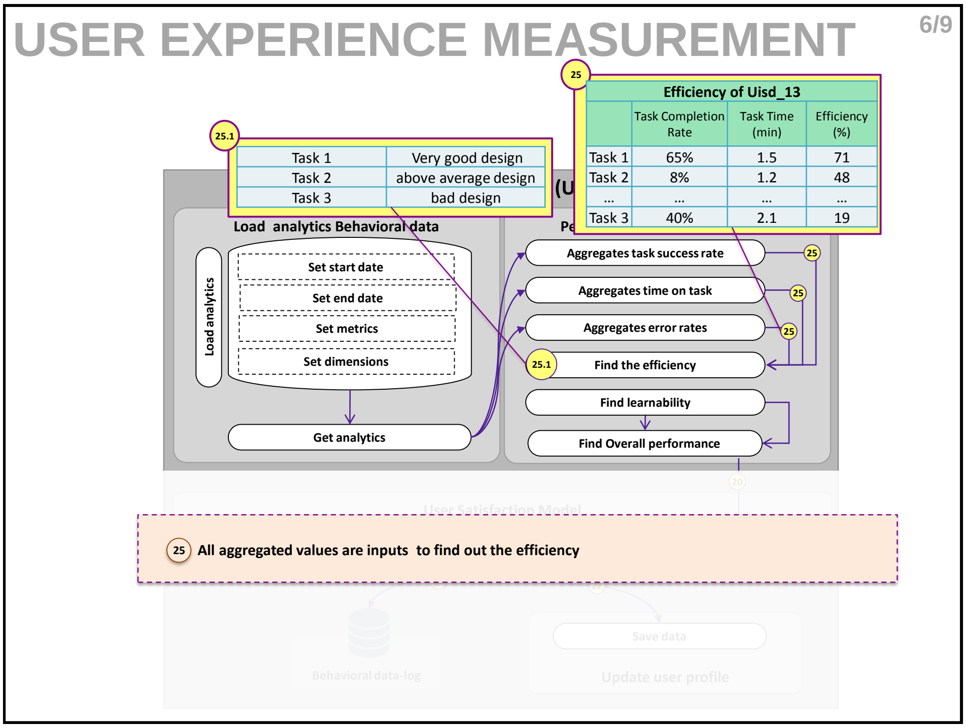

USER EXPERIENCE MEASUREMENT
3/9
Task Success
Task 1 Task 2 Task 3
Uid_13 1 0 0
Uid_11 1 1 1
... ...
Uid_10 0 1 1
Uid_13 Task Success 33 %
User Experience (UX)
Load analytics
Set start date
Set end date
Set metrics
Set dimensions
Get analytics
Performance metrics calculation
Aggregates task success rate
Aggregates time on task
Aggregates error rates
Find the efficiency
Find learnability
Find Overall performance
Screen Name
Screen Views
Unique Screen Views
6,047
341
1. Dashboard Screen 5,978 326
2. Profile Screen 37 7
3. User Feedback Dialog 11 2
4. Profile Screen- Personalized Map 8 2
5. Profile Screen- Activity Level Information 7 2
6. Profile Screen- Account Information 6 2
24.1 All task is checked either it is successfully done or not? Then find out the aggregates the task success rate.

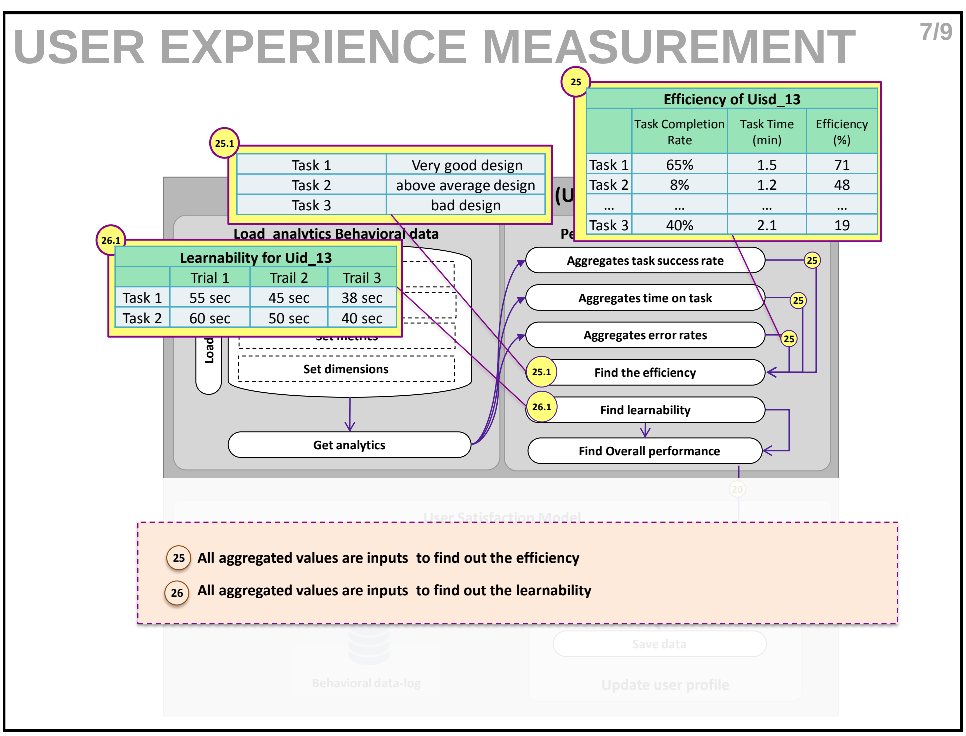


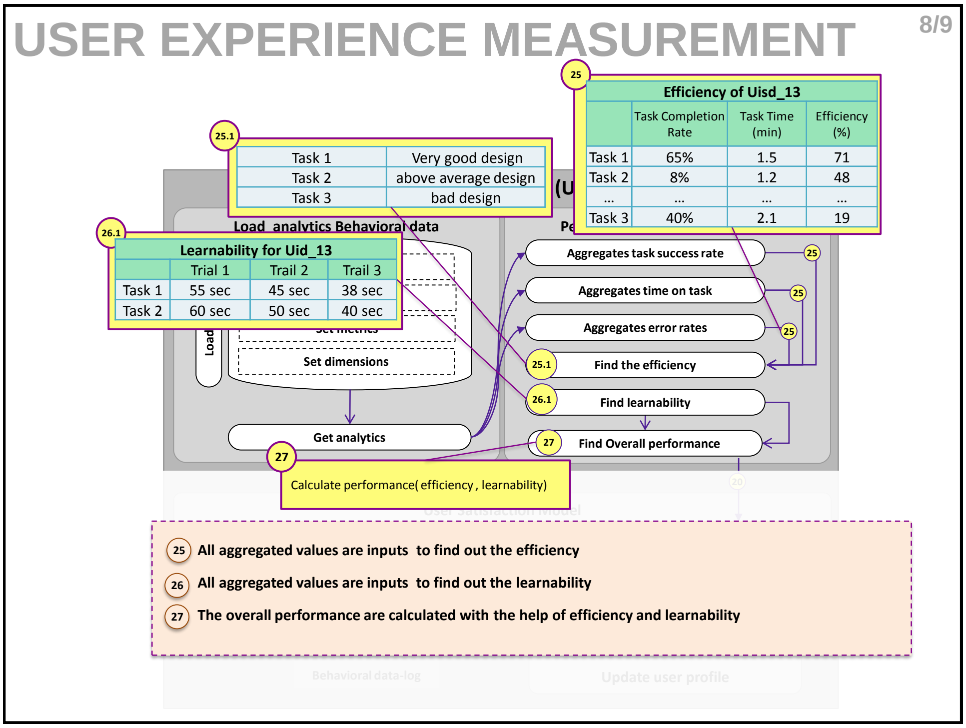
USER EXPERIENCE MEASUREMENT
4/9
24.1
Task Success
	Task 1	Task 2	Task 3
Uid_13	1	0	0
Uid_11	1	1	1
...	...	...	...
Uid_10	0	1	1
24.1
Uid_13	Task Success	33 %
24.2
Uid_13	Time On task	28 %
24.2
Time On task
	Task 1	Task 2	Task 3
Uid_13	55	20	10
Uid_11	30	20	10
...	...	...	...
Uid_10	0	1	1
ser Experience (UX)
Performance metrics calculation
24 Aggregates task success rate
Aggregates time on task 24
Aggregates error rates
Find the efficiency
Find learnability
Find Overall performance
23 Get analytics
24
24
23
	Screen Name	Screen Views	Unique Screen Views
		6,047 % of Total: 100.00% (6,047)	341 % of Total: 100.00% (341)
1.	Dashboard Screen	5,978 (98.86%)	326 (95.60%)
2.	Profile Screen	37 (0.61%)	7 (2.05%)
3.	User Feedback Dialog	11 (0.18%)	2 (0.59%)
4.	Profile Screen- Personalized Map	8 (0.13%)	2 (0.59%)
5.	Profile Screen- Activity Level Information	7 (0.12%)	2 (0.59%)
6.	Profile Screen- Account Information	6 (0.10%)	2 (0.59%)
24.1 All task is checked either it is successfully done or not? Then find out the aggregates the task success rate.
24.2 Find out the spent time on each task and aggregates the time on task.

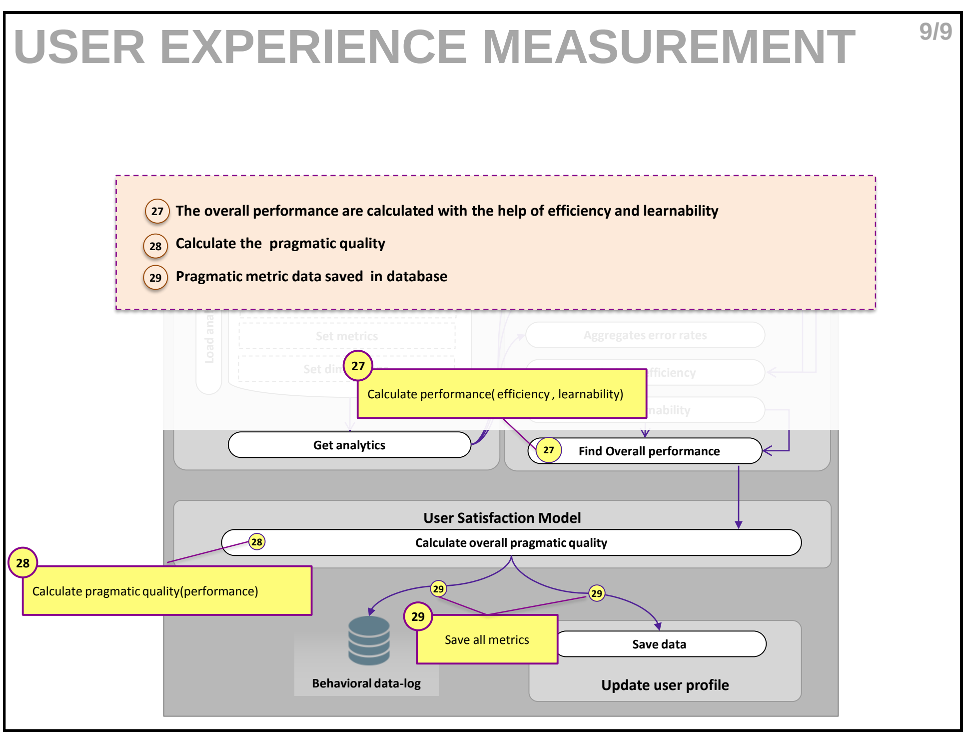


USER EXPERIENCE MEASUREMENT
5/9
24.1
Task Success
	Task 1	Task 2	Task 3
Uid_13	1	0	0
Uid_11	1	1	1
...	...	...	...
Uid_10	0	1	1
24.1
Uid_13	Task Success	33 %
24.2
Uid_13	Time On task	28 %
24.2
Time On task
	Task 1	Task 2	Task 3
Uid_13	55	20	10
Uid_11	30	20	10
...	...	...	...
Uid_10	0	1	1
24.3
Errors
	Task 1	Task 2	Task 3
Uid_13	1	1	0
Uid_11	0	1	1
...	...	...	...
Uid_10	0	1	0
ser Experience (UX)
Performance metrics calculation
24 Aggregates task success rate
Aggregates time on task 24
Aggregates error rates 24
24.3
Uid_13	Errors rates	34 %
Find learnability
Find Overall performance
24
24
24
23
	Screen Name	Screen Views	Unique Screen Views
		6,047 % of Total: 100.00% (6,047)	341 % of Total: 100.00% (341)
1.	Dashboard Screen	5,978 (98.86%)	326 (95.60%)
2.	Profile Screen	37 (0.61%)	7 (2.05%)
3.	User Feedback Dialog	11 (0.18%)	2 (0.59%)
4.	Profile Screen- Personalized Map	8 (0.13%)	2 (0.59%)
5.	Profile Screen- Activity Level Information	7 (0.12%)	2 (0.59%)
6.	Profile Screen- Account Information	6 (0.10%)	2 (0.59%)
24.1 All task is checked either it is successfully done or not? Then find out the aggregates the task success rate.
24.2 Find out the spent time on each task and aggregates the time on task.
24.3 Find out the task errors and aggregate the errors rates.


USER EXPERIENCE MEASUREMENT
6/9
25
Efficiency of Uisd_13
Task Completion Rate
Task Time (min)
Efficiency (%)
Task 1 65% 1.5 71
Task 2 8% 1.2 48
... ...
Task 3 40% 2.1 19
25.1
Task 1 Very good design
Task 2 above average design
Task 3 bad design
Load analytics Behavioral data
Load analytics
Set start date
Set end date
Set metrics
Set dimensions
Get analytics
Aggregates task success rate
Aggregates time on task
Aggregates error rates
Find the efficiency
Find learnability
Find Overall performance
25 All aggregated values are inputs to find out the efficiency


USER EXPERIENCE MEASUREMENT
7/9
25
Efficiency of Uisd_13
Task Completion Rate
Task Time (min)
Efficiency (%)
Task 1 65% 1.5 71
Task 2 8% 1.2 48
... ...
Task 3 40% 2.1 19
25.1
Task 1 Very good design
Task 2 above average design
Task 3 bad design
Load analytics Behavioral data
26.1
Learnability for Uid_13
Trial 1 Trail 2 Trail 3
Task 1 55 sec 45 sec 38 sec
Task 2 60 sec 50 sec 40 sec
Set dimensions
Get analytics
Aggregates task success rate
Aggregates time on task
Aggregates error rates
Find the efficiency
Find learnability
Find Overall performance
25 All aggregated values are inputs to find out the efficiency
26 All aggregated values are inputs to find out the learnability


USER EXPERIENCE MEASUREMENT
8/9
Efficiency of Uisd_13
Task Completion Rate | Task Time (min) | Efficiency (%)
Task 1 | 65% | 1.5 | 71
Task 2 | 8% | 1.2 | 48
... | ... | ... | ...
Task 3 | 40% | 2.1 | 19
Task 1 | Very good design
Task 2 | above average design
Task 3 | bad design
Learnability for Uid_13
Trial 1 | Trial 2 | Trial 3
Task 1 | 55 sec | 45 sec | 38 sec
Task 2 | 60 sec | 50 sec | 40 sec
Load analytics Behavioral data
Set dimensions
Get analytics
Aggregates task success rate
Aggregates time on task
Aggregates error rates
Find the efficiency
Find learnability
Find Overall performance
Calculate performance(efficiency , learnability)
25 All aggregated values are inputs to find out the efficiency
26 All aggregated values are inputs to find out the learnability
27 The overall performance are calculated with the help of efficiency and learnability

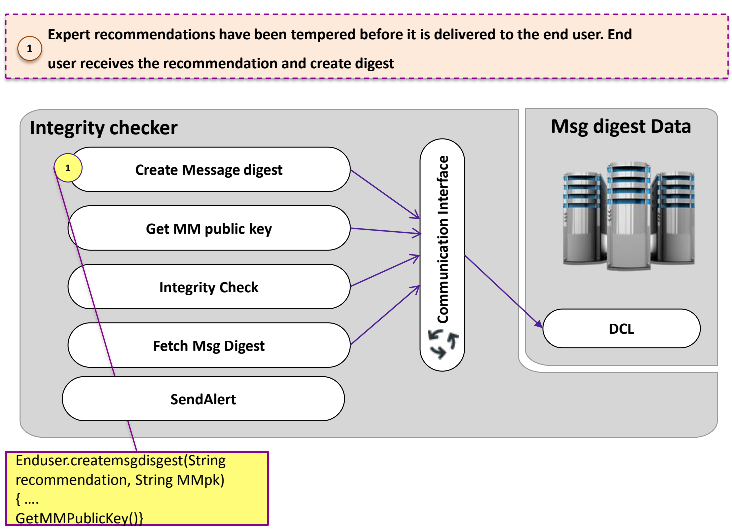

USER EXPERIENCE MEASUREMENT
9/9
27 The overall performance are calculated with the help of efficiency and learnability
28 Calculate the pragmatic quality
29 Pragmatic metric data saved in database
Calculate performance(efficiency , learnability)
Get analytics
Find Overall performance
User Satisfaction Model
Calculate overall pragmatic quality
Calculate pragmatic quality(performance)
Save all metrics
Save data
Behavioral data-log
Update user profile

Scenario 3: Man in the Middle Attack for Recommendation

Desc.: Assuming the **ability of an attacker** to **modify** the recommendations generated by an expert before it is delivered to the user. In case of **tempered recommendations**, user is refrained from following the recommendation and expert is asked for **re-sending** the new recommendation with updated password

Recommendation Integrity Check

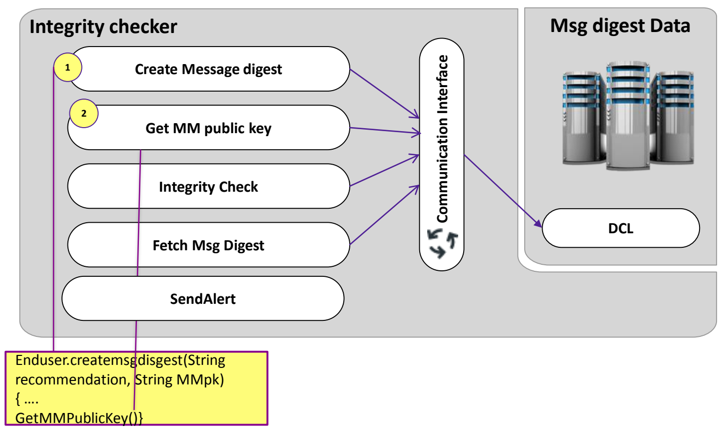
1/8



Recommendation Integrity Check

2/8

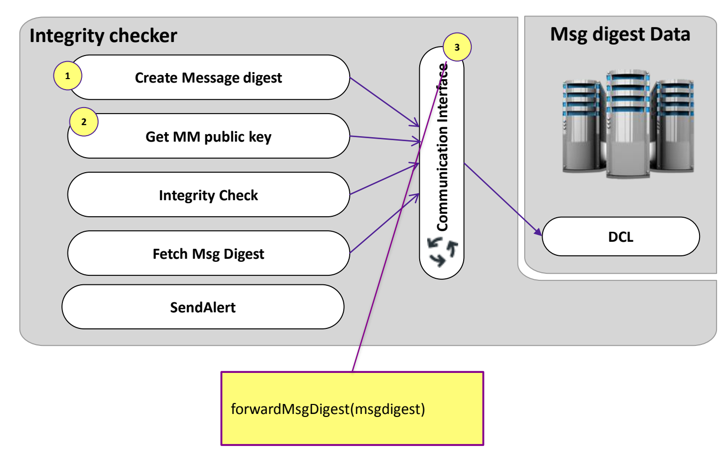
1. Expert recommendations have been tempered before it is delivered to the end user
2. Get public key of Mining Mind to create msg digest



Recommendation Integrity Check

3/8

1. Expert recommendations have been tempered before it is delivered to the end user
2. Get public key of Mining Mind to create msg digest
3. To preserve the message digest, DCL communication interface is invoked



4/8

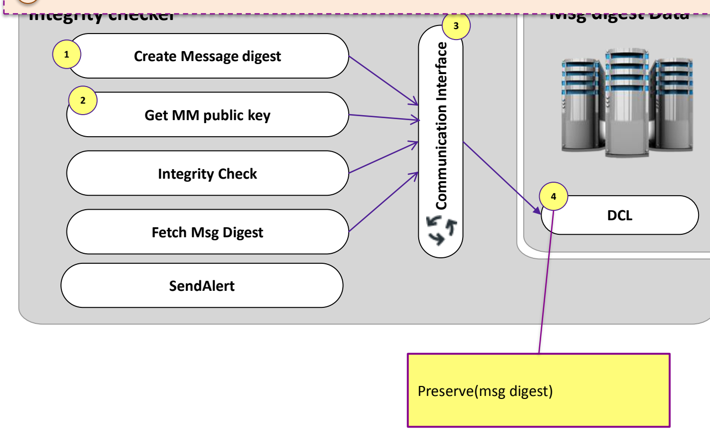
Recommendation Integrity Check

1 Expert recommendations have been tempered before it is delivered to the end user

2 Get public key of Mining Mind to create msg digest

3 To preserve the message (recommendation) digest, DCL communication interface is

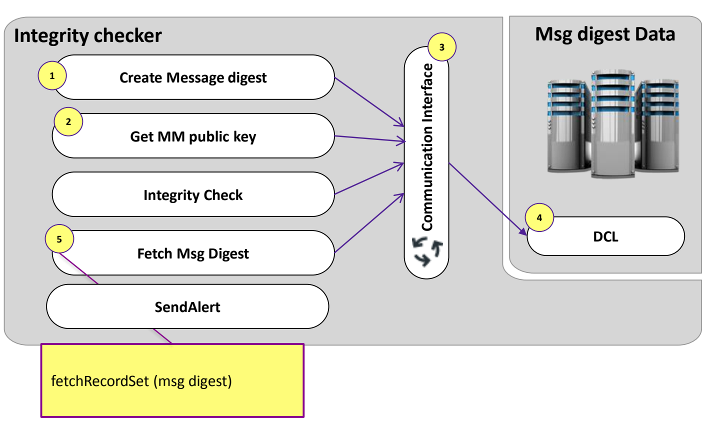
4 DCL preserve the message digest



5/8

Recommendation Integrity Check

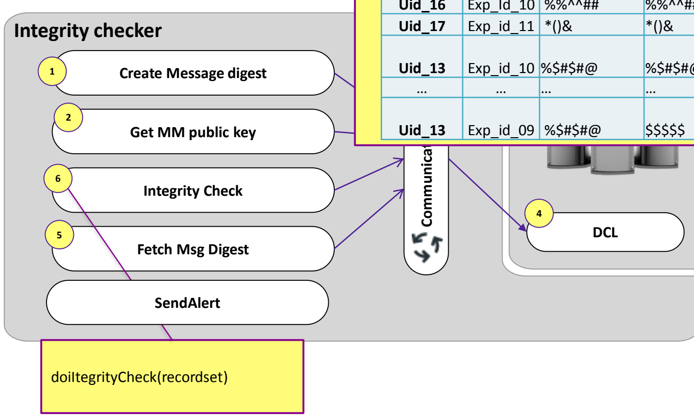
5 Fetch Msg digest reterives the msg digest from both expert and the end user



Recommendation Integrity Check

(6) fetchMsgDigest() retrieves the msg digest from both expert and the end user from the persisted records

User ID	Expert ID	Exp-Msg-dgst	User Msg digest	Status
Uid_16	Exp_Id_10	%%^^##	%%^^##	Ok
Uid_17	Exp_id_11	*()&	*()&	Ok
Uid_13	Exp_id_10	%$#$#@	%$#$#@	Ok
…	…	…	…	Ok
Uid_13	Exp_id_09	%$#$#@	$$$$$	tempered

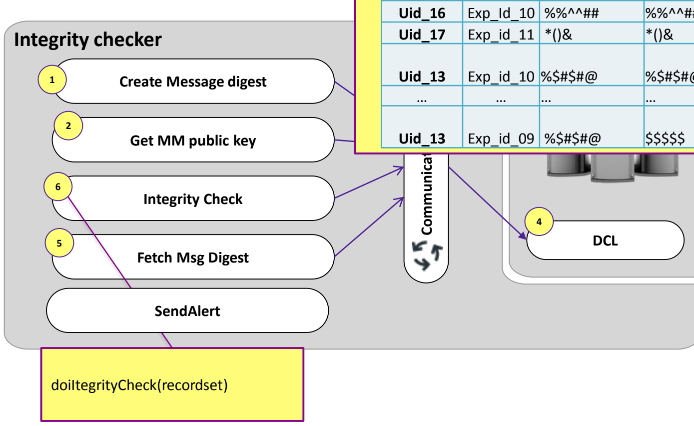


Recommendation Integrity Check

(5) fetchMsgDigest() retrieves the msg digest from both expert and the end user from the persisted records

(6) The last table shows the mismatch status of two msg digest which sends

User ID	Expert ID	Exp-Msg-dgst	User Msg digest	Status
Uid_16	Exp_Id_10	%%^^##	%%^^##	Ok
Uid_17	Exp_id_11	*()&	*()&	Ok
Uid_13	Exp_id_10	%$#$#@	%$#$#@	Ok
…	…	…	…	Ok
Uid_13	Exp_id_09	%$#$#@	$$$$$	tempered



Recommendation Integrity Check

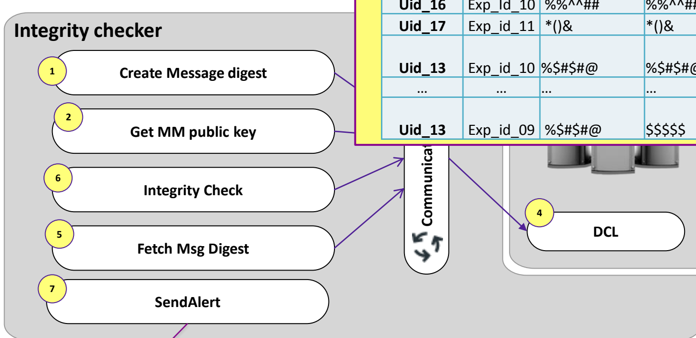
8/8

(5) fetchMsgDigest() retrieves the msg digest from both expert and the end user from the persisted records

(6) The last table shows the mismatch status of two msg digest which sends

(6)

User ID	Expert ID	Exp-Msg-dgst	User Msg digest	Status
Uid_16	Exp_Id_10	%%^^##	%%^^##	Ok
Uid_17	Exp_id_11	*()&	*()&	Ok
Uid_13	Exp_id_10	%$#$#@	%$#$#@	Ok
…	…	…	…	Ok
Uid_13	Exp_id_09	%$#$#@	$$$$$	tempered



sendAlert(user,Expert)
User.msg(discardRecommendation)
Expert.msg(update.pwd)
Expert.msg(resendRecommendation)

(7) Respective msgs are sent to the expert and user. Expert is notified with reset password and re-sending of new message. End user is notified to discard the last received recommendation

Thanks