Proceedings of Mining Minds Release Version 2.0

Research & Technical Specifications

Principal Investigator:

Professor Sungyoung Lee

December 30th, 2015

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, Soung Ho Cho, Usman Akhtar, Taqdir Ali, Jinhyuk Gong

ICL Team

Dr. Oresti Baños (Team Lead)
Claudia Villalonga, Le Ba Vui, Taeho Hur, Sangbeom Park
Jaehun Bang, Thien Huynh The, Donguk Kang
Muhammad Asif Razzaq

KCL Team

Maqbool Hussain (Team Lead)
Taqdir Ali, Maqbool Ali, Mahmood Ahmad, Sangho Lee

SCL Team

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

SL Team

Dr. Wajahat Ali Khan (Team Lead)
Shujaat Hussain, Jamil Hussain

Facilitators

Professor Byeong Ho Kang (University of Tasmania, Australia)

Document History

Created By:	Bilal Amin	**Last Updated By:**	Bilal Amin
Date Created:	July 13, 2015	**Last Revision Date:**	Dec 15, 2015

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

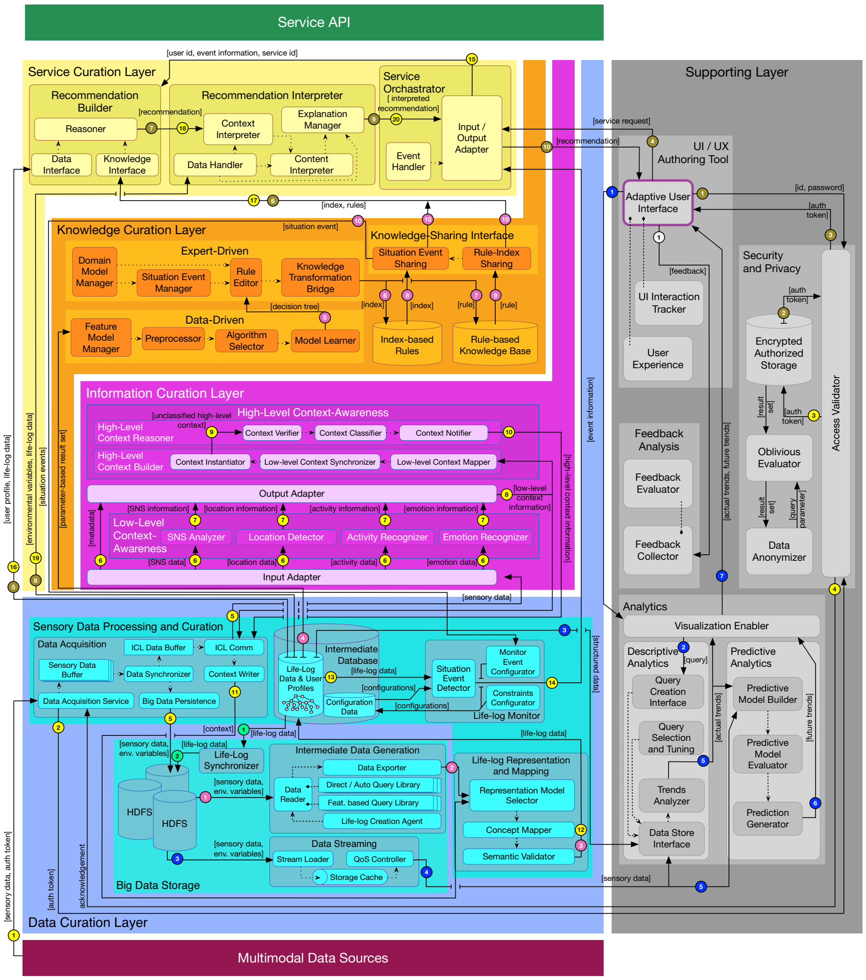


Figure 1: Mining Minds Framework Architecture and Operational diagram.

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

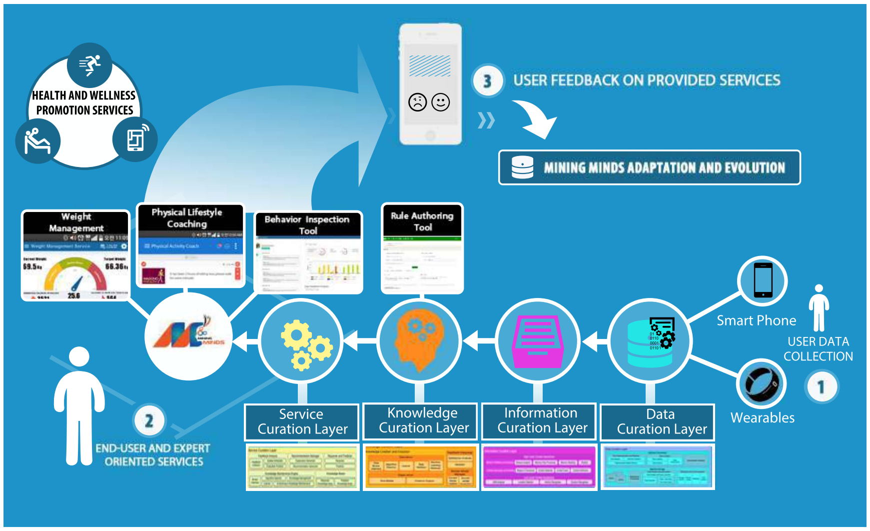


Figure 2: Mining Minds health and wellness service scenario.

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

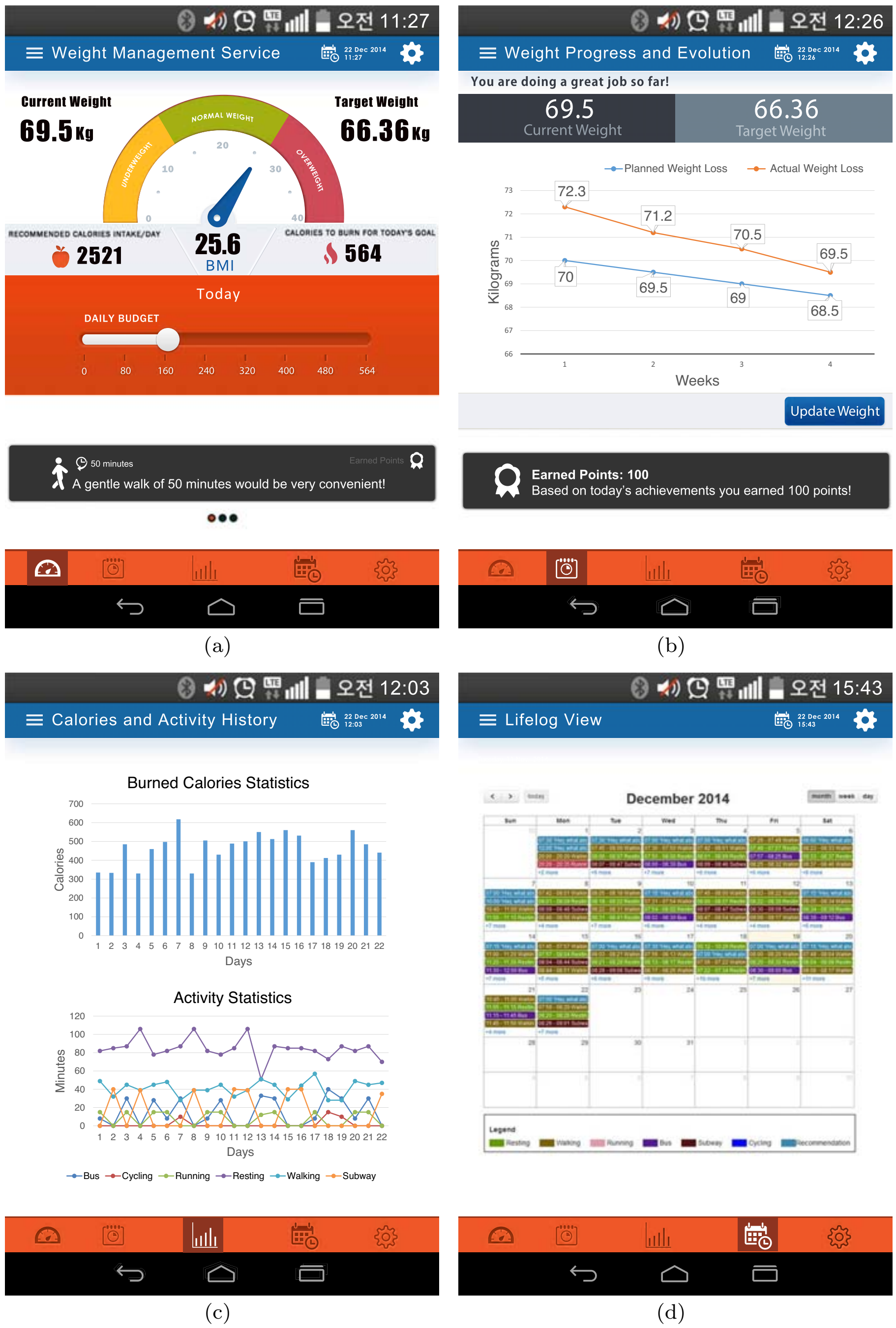


Figure 3: Personalized weight management app.

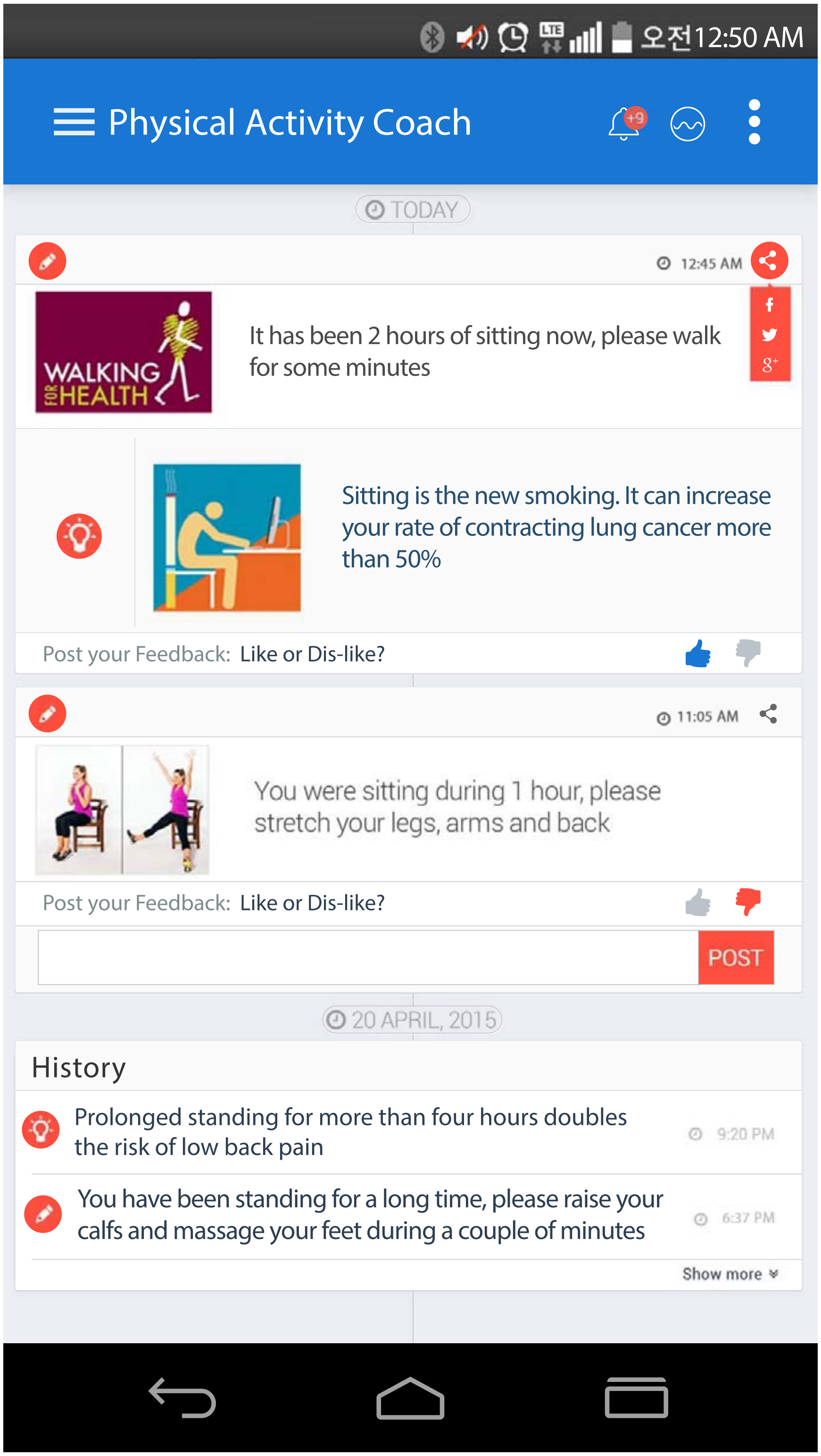


Figure 4: Physical lifestyle coaching app.

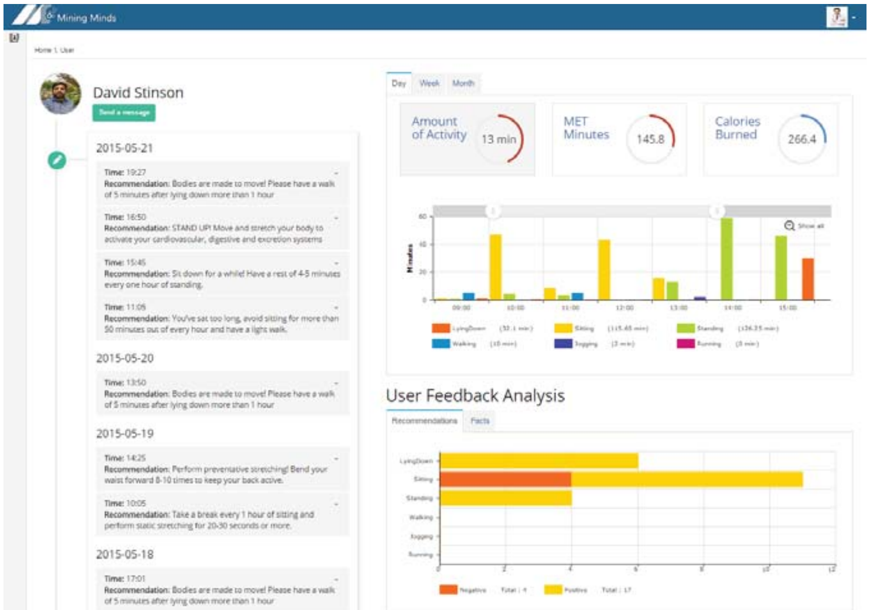


Figure 5: Behavior inspection tool.

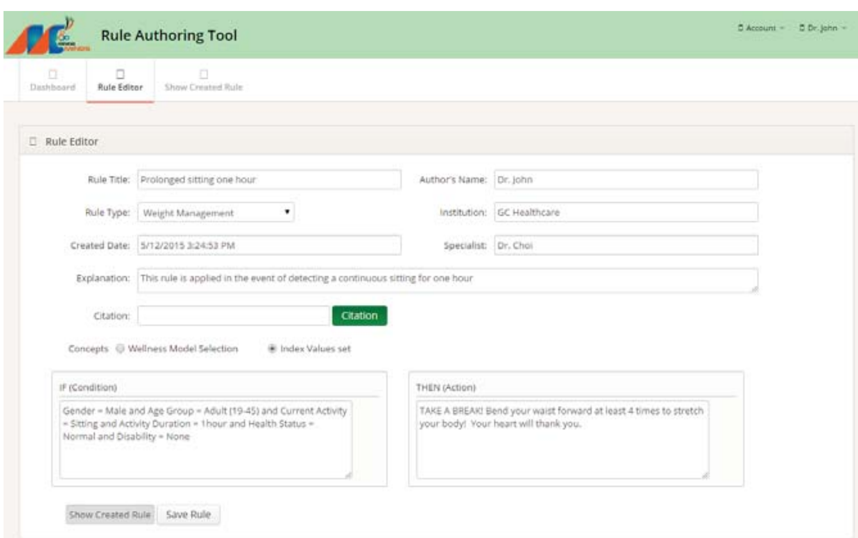


Figure 6: Rule authoring tool.

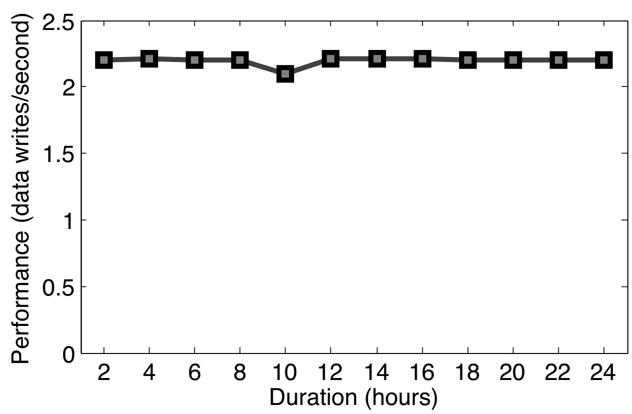


Figure 7: Performance of the data persistence process.

Section 2

Mining Minds Version 2.0 Overview and Demonstration

Mining Minds V2.0

Ubiquitous Computing Lab (UCLab)
Kyung Hee University, South Korea

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

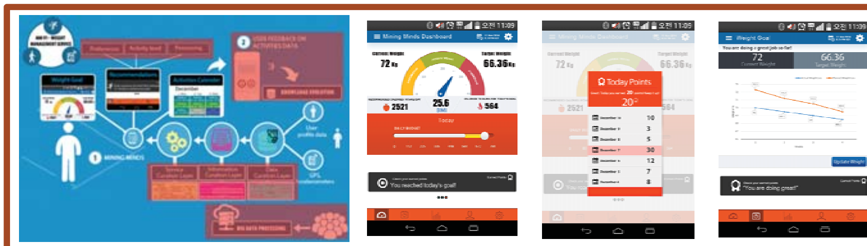
Agenda

- The History of MM
- Software Engineering Principles
- Architecture
- Platform Operation
- Overall Uniqueness
- Technical Contributions
- Microdemos for Proof of Technical Contributions
- Future Plan
- Conclusions
- End-to-End Live Demonstration

History of Mining Minds

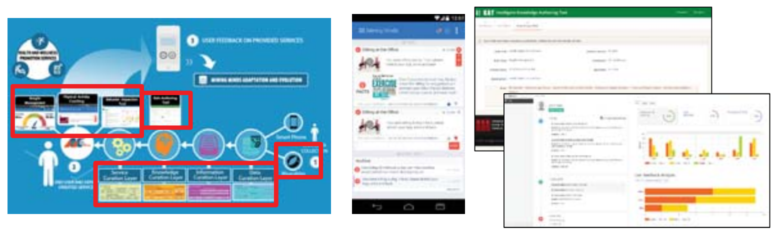
MM V1.0
Dec 2014

Weight Management App
Management Tool

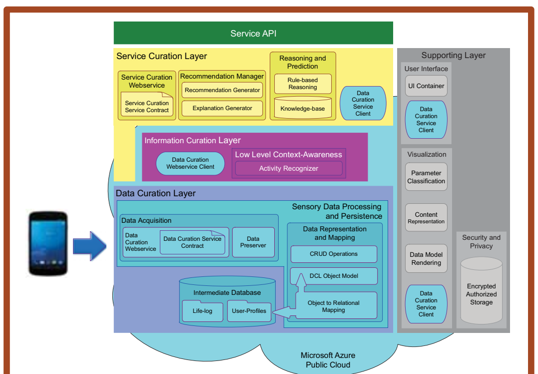


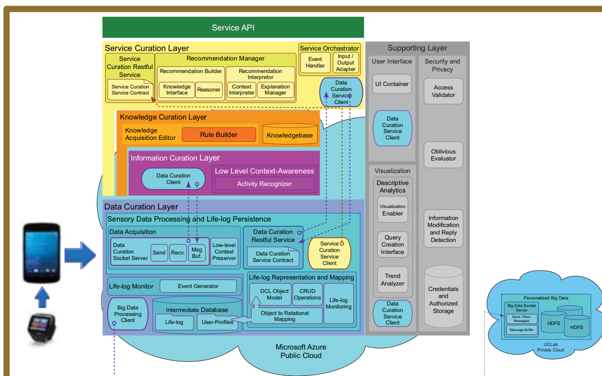
MM V1.5
May 2015

Personalized Lifestyle Coaching App
Rule Authoring Tool
Behavior Inspection Tool



History of Mining Minds





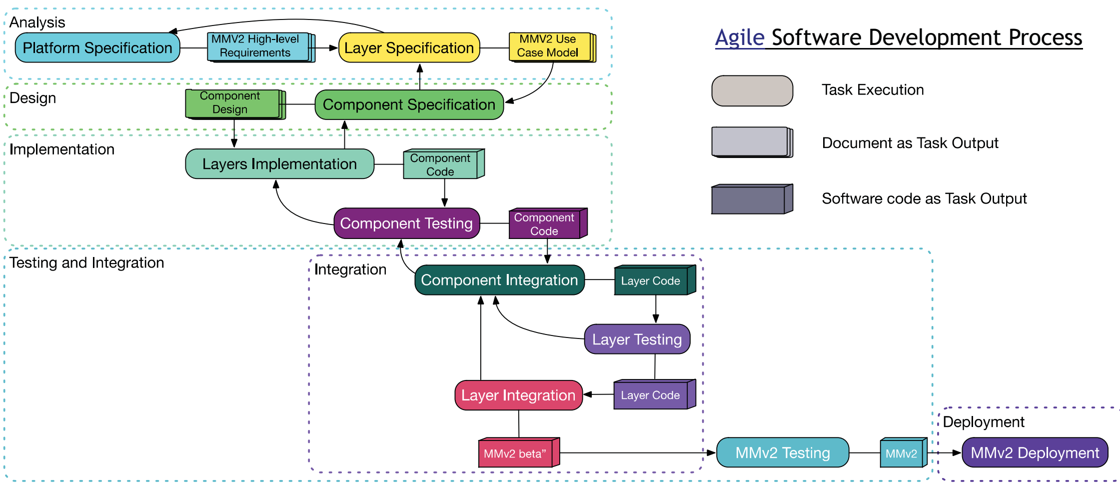
MM V1.0
Dec 2014

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)

Software Engineering Principles



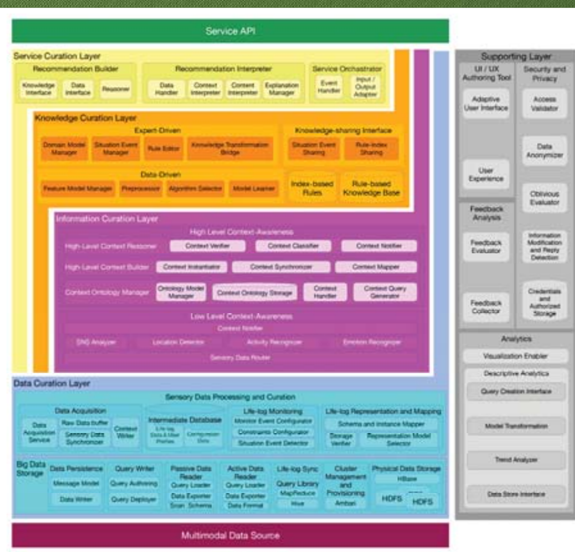
MMv2.0: Architecture

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

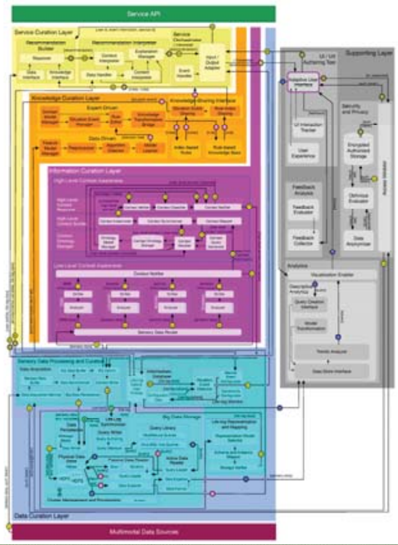
Converts the data obtained from the user interaction with the real and cyberworld, into abstract concepts or categories, such as physical activities, emotional states, locations and social patterns, which are intelligently combined to determine and track context and behavior

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



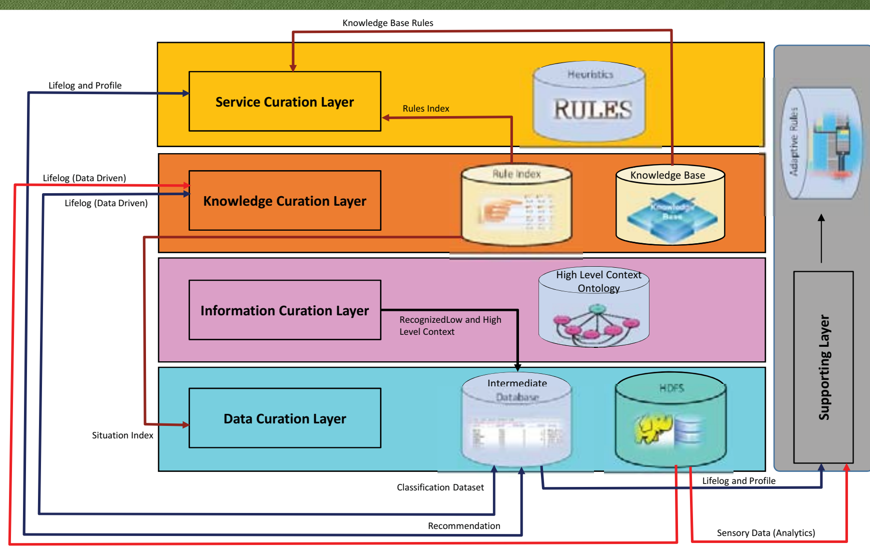
Facilitates information to the users in the most intuitive manner, in a secure environment reflecting their personal needs and preferences

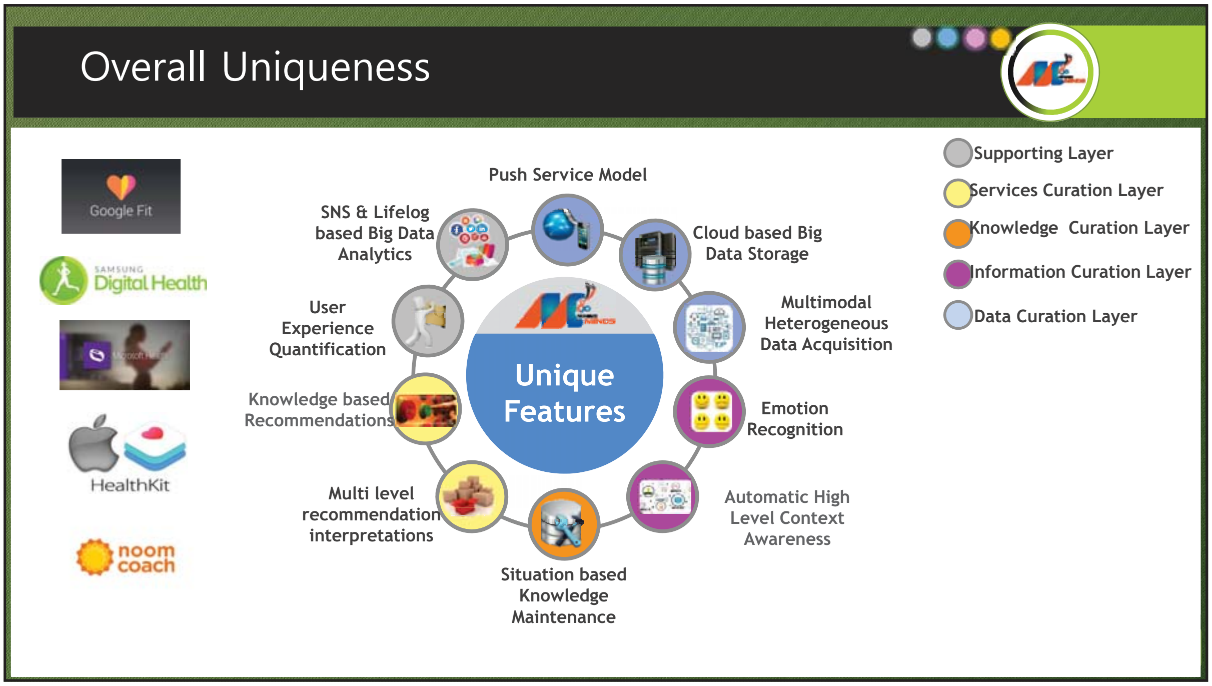
MMv2.0: Operation Flow

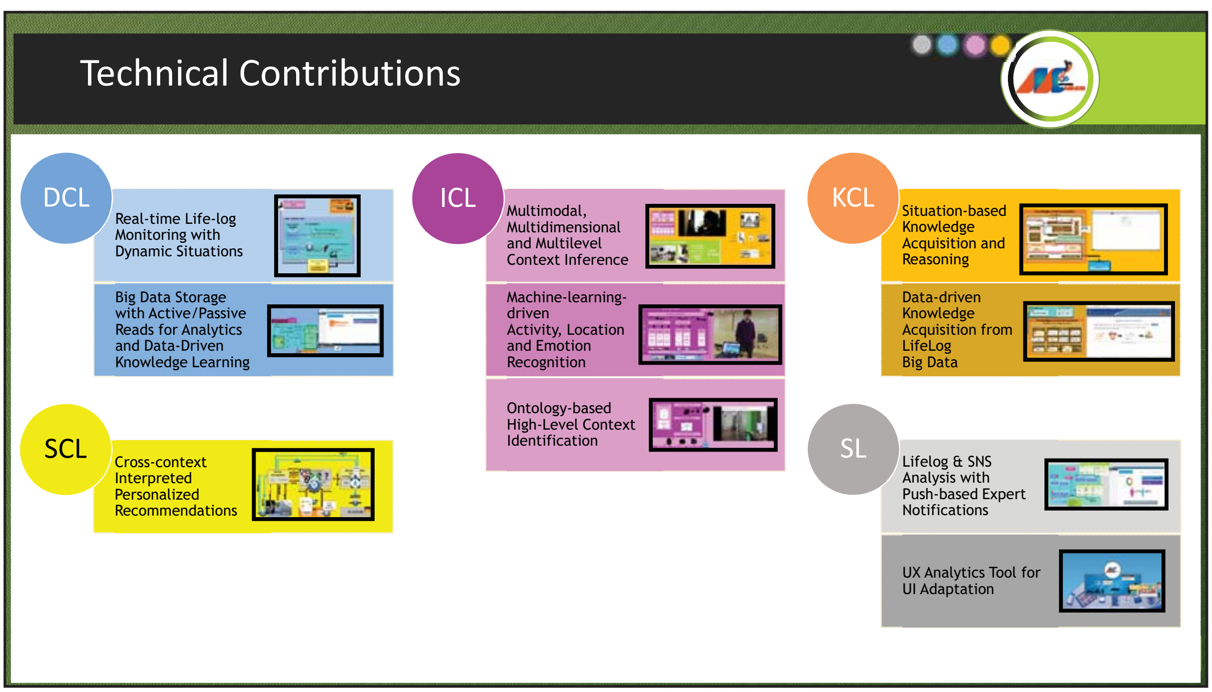


Legend
- Real-time Heterogenous Data Acquisition and Push-model Recommendation Generation
- Visualization and Analytics Request
- Feedback Request
- Recommendation Request (Pull-model)
- Life-log Data Synchronization
- Data-driven & Expert-driven Rules Creation

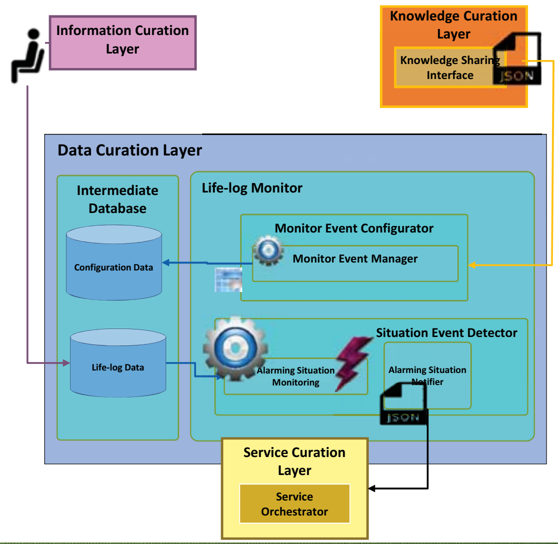
MMv2.0: Distributed Database Model



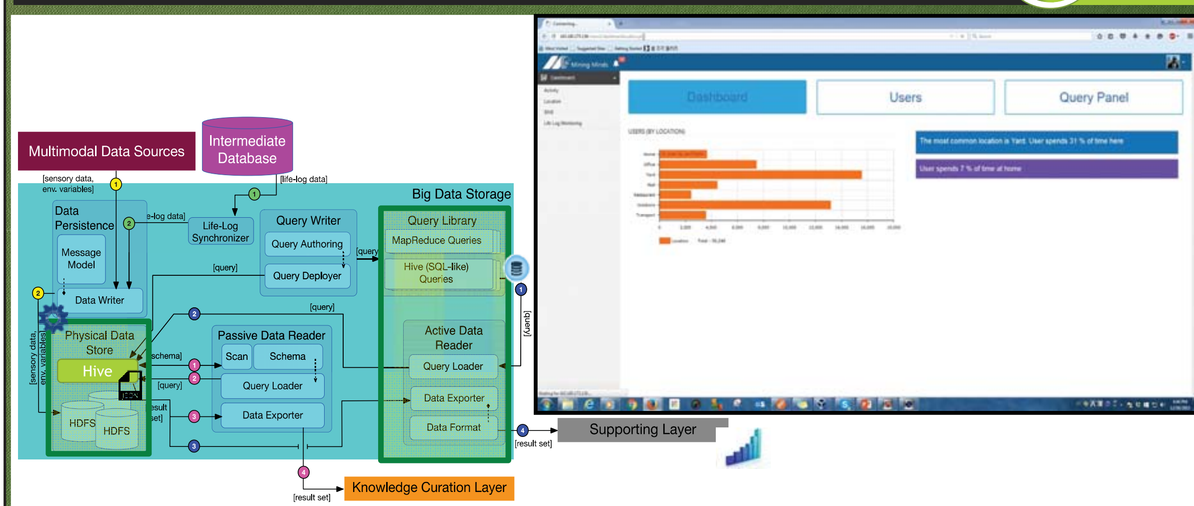

Overall Uniqueness
Google Fit
SAMSUNG Digital Health
HealthKit
noom coach
Push Service Model
SNS & Lifelog based Big Data Analytics
Cloud based Big Data Storage
User Experience Quantification
Multimodal Heterogeneous Data Acquisition
Unique Features
Knowledge based Recommendations
Emotion Recognition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


Technical Contributions
DCL
Real-time Life-log Monitoring with Dynamic Situations
Big Data Storage with Active/Passive Reads for Analytics and Data-Driven Knowledge Learning
ICL
Multimodal, Multidimensional and Multilevel Context Inference
Machine-learning-driven Activity, Location and Emotion Recognition
Ontology-based High-Level Context Identification
KCL
Situation-based Knowledge Acquisition and Reasoning
Data-driven Knowledge Acquisition from LifeLog Big Data
SCL
Cross-context Interpreted Personalized Recommendations
SL
Lifelog & SNS Analysis with Push-based Expert Notifications
UX Analytics Tool for UI Adaptation

Real-time Life-log Monitoring with Dynamic Situations

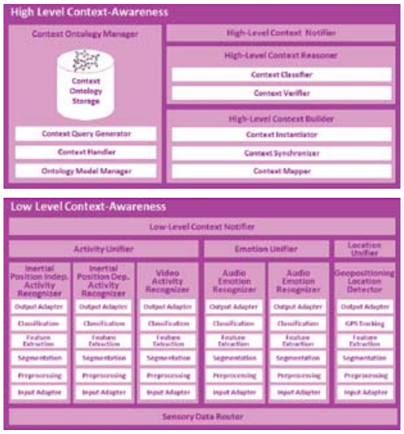


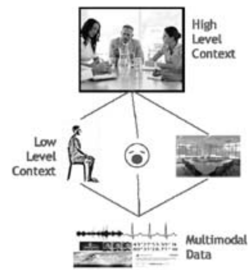
Big Data Storage with Active and Passive Reads for Analytics and Data-Driven Knowledge Model Learning



ICL

Multimodal, Multidimensional and Multilevel Context Inference





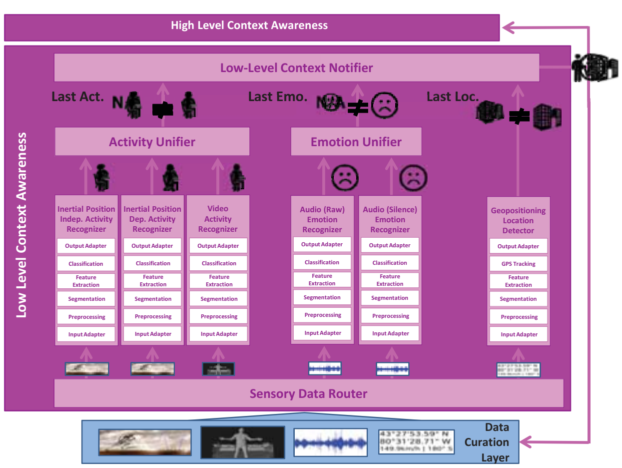
Mining Minds Multimodal Dataset

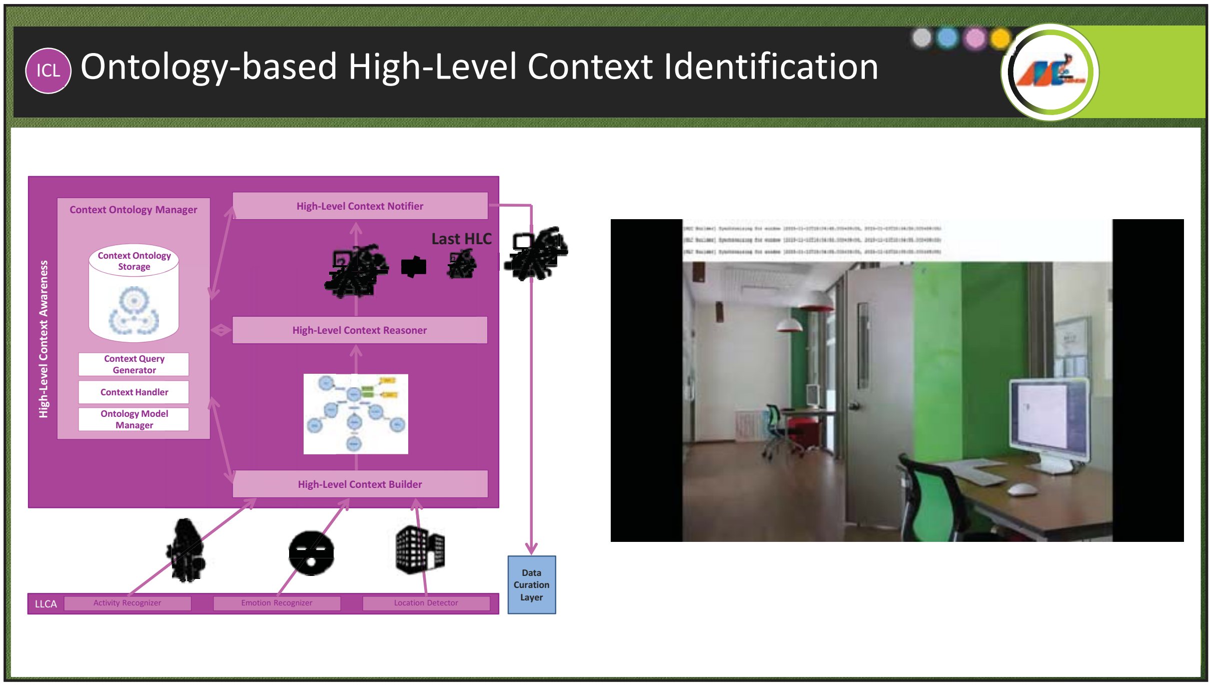
10 subjects
12 activities
8 locations
4 emotions
98 sensor signals

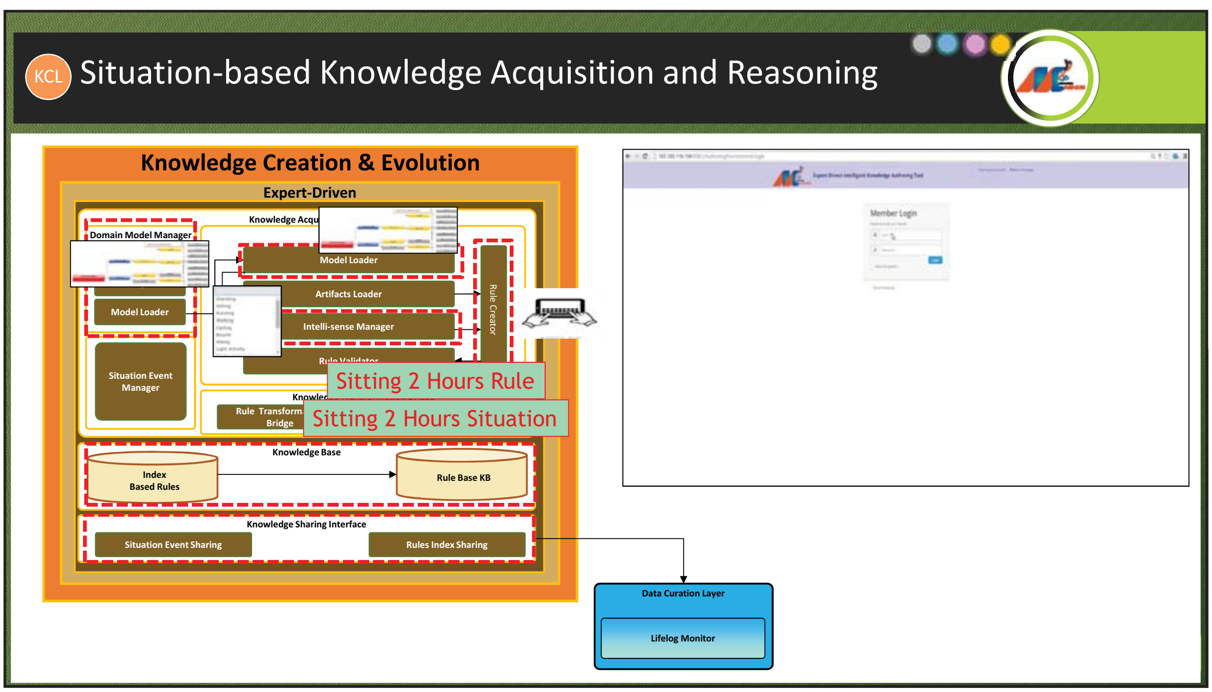
>2GB of data

ICL

Machine-learning-driven Activity, Location and Emotion Recognition

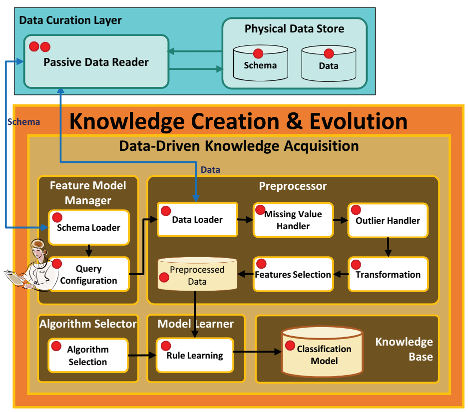


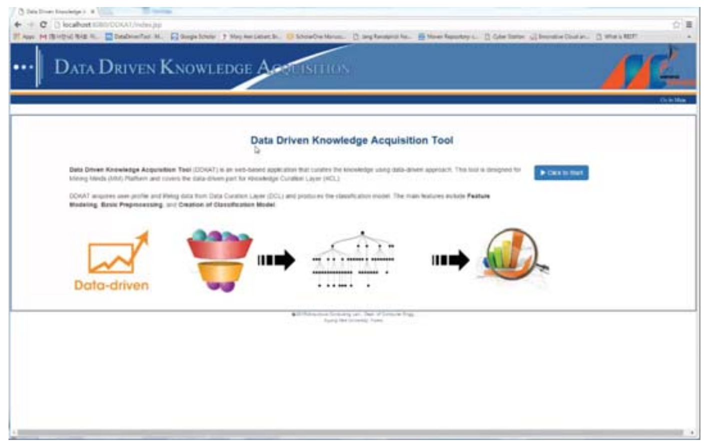

ICL
Ontology-based High-Level Context Identification
High-Level Context Awareness
Context Ontology Manager
Context Ontology Storage
Context Query Generator
Context Handler
Ontology Model Manager
High-Level Context Notifier
Last HLC
High-Level Context Reasoner
High-Level Context Builder
LLCA
Activity Recognizer
Emotion Recognizer
Location Detector
Data Curation Layer


KCL
Situation-based Knowledge Acquisition and Reasoning
Knowledge Creation & Evolution
Expert-Driven
Domain Model Manager
Model Loader
Situation Event Manager
Model Loader
Artifacts Loader
Intelli-sense Manager
Rule Creator
Sitting 2 Hours Rule
Sitting 2 Hours Situation
Rule Transform Bridge
Knowledge Base
Index Based Rules
Rule Base KB
Knowledge Sharing Interface
Situation Event Sharing
Rules Index Sharing
Data Curation Layer
Lifelog Monitor
Member Login

KCL

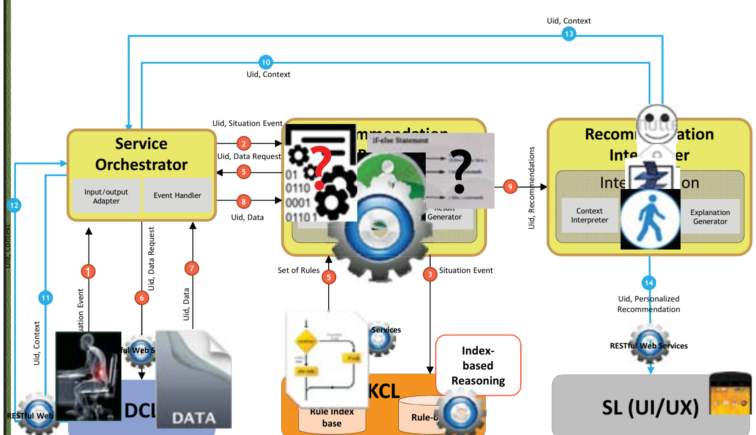
Data-driven Knowledge Acquisition from LifeLog Big Data



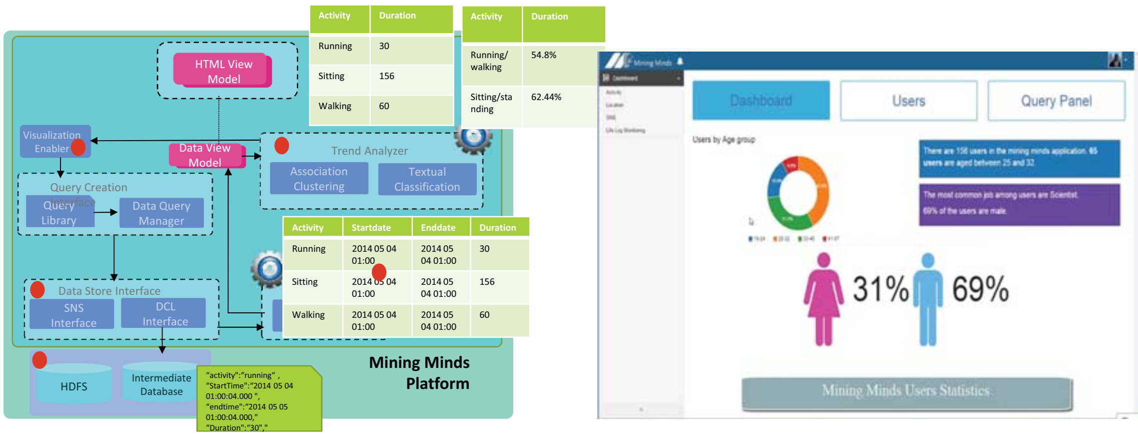


SCL

Cross-context Interpreted Personalized Recommendations



Lifelog & SNS Analysis with Push-based Expert Notifications



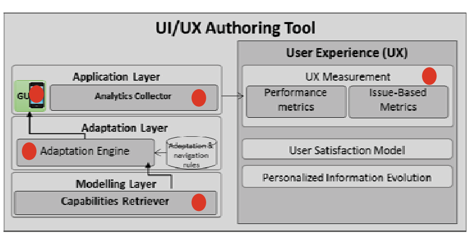
Activity	Duration
Running	30
Sitting	156
Walking	60

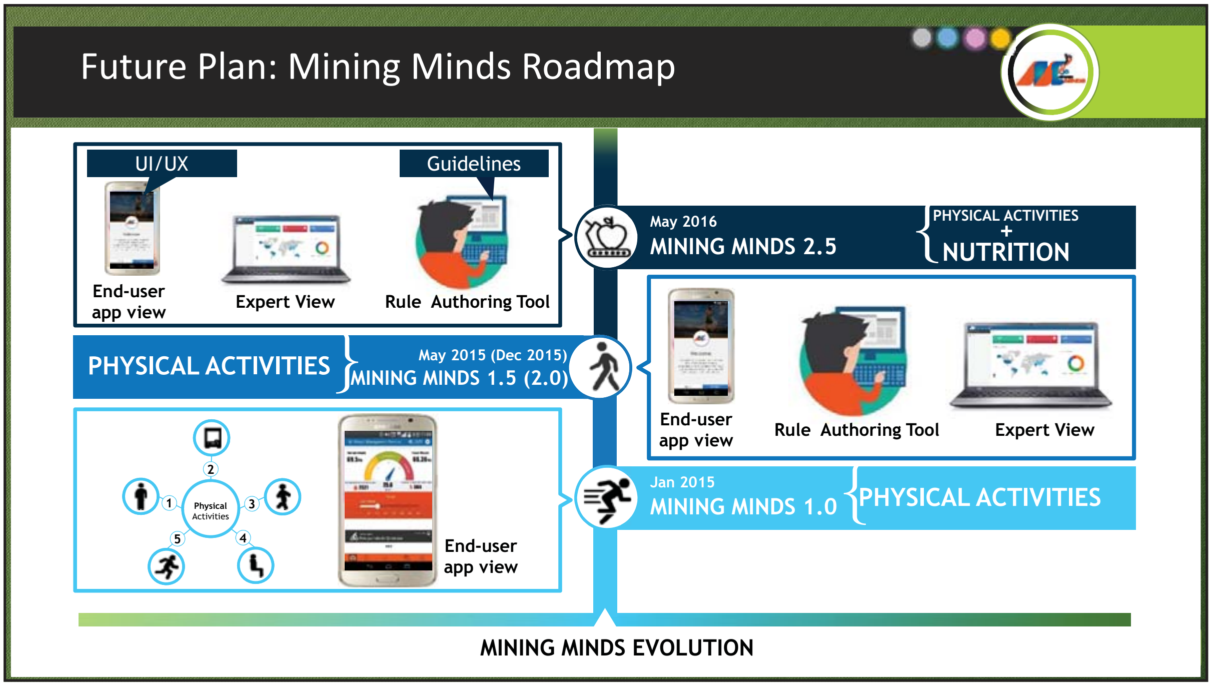
Activity	Duration
Running/ walking	54.8%
Sitting/standing	62.44%

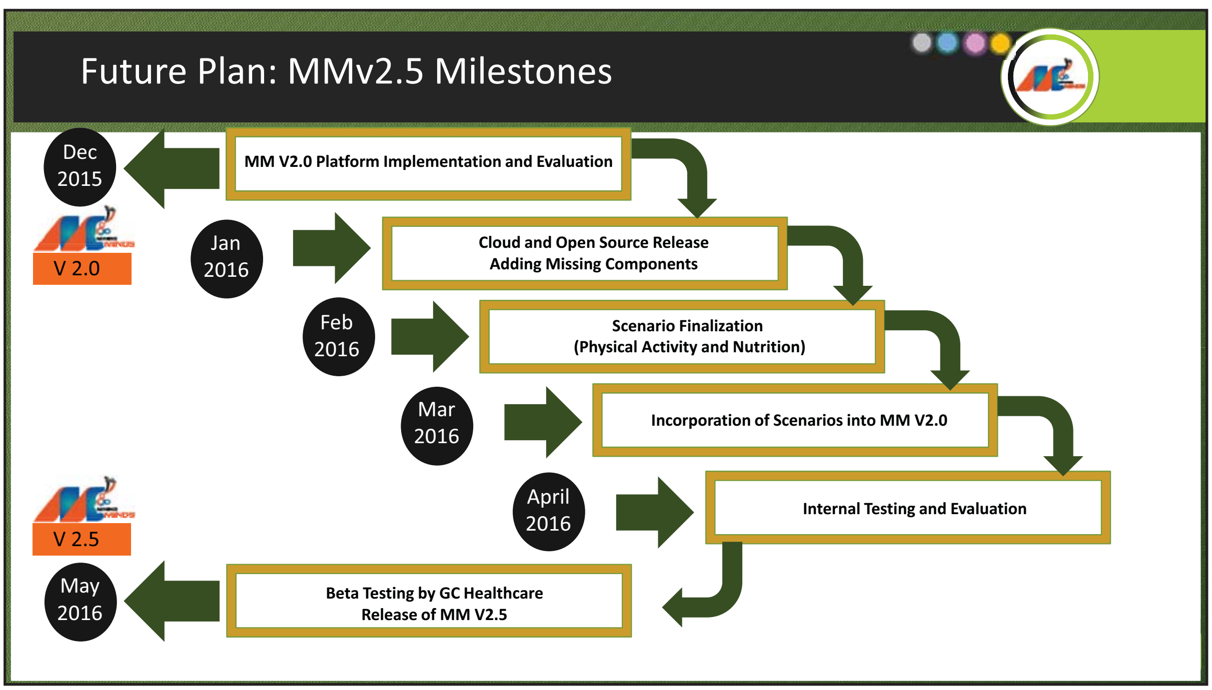
Activity	Startdate	Enddate	Duration
Running	2014 05 04 01:00	2014 05 04 01:00	30
Sitting	2014 05 04 01:00	2014 05 04 01:00	156
Walking	2014 05 04 01:00	2014 05 04 01:00	60

"activity":"running" , "StartTime":"2014 05 04 01:00:04.000 ", "endtime":"2014 05 05 01:00:04.000," "Duration":"30","

UX Analytics Tool for UI Adaptation




Future Plan: Mining Minds Roadmap
UI/UX
Guidelines
End-user app view
Expert View
Rule Authoring Tool
May 2016
MINING MINDS 2.5
PHYSICAL ACTIVITIES + NUTRITION
PHYSICAL ACTIVITIES
May 2015 (Dec 2015)
MINING MINDS 1.5 (2.0)
End-user app view
Rule Authoring Tool
Expert View
Physical Activities
End-user app view
Jan 2015
MINING MINDS 1.0
PHYSICAL ACTIVITIES
MINING MINDS EVOLUTION


Future Plan: MMv2.5 Milestones
Dec 2015
MM V2.0 Platform Implementation and Evaluation
V 2.0
Jan 2016
Cloud and Open Source Release
Adding Missing Components
Feb 2016
Scenario Finalization
(Physical Activity and Nutrition)
Mar 2016
Incorporation of Scenarios into MM V2.0
April 2016
Internal Testing and Evaluation
V 2.5
May 2016
Beta Testing by GC Healthcare
Release of MM V2.5

Future Plan: MMv2.5 Technical Steps

DCL
- **Dynamic situations** with **non-linear life-log monitoring**
- **Collection** and enrichment of raw sensory data for a **large-scale** comprehensive **data set**

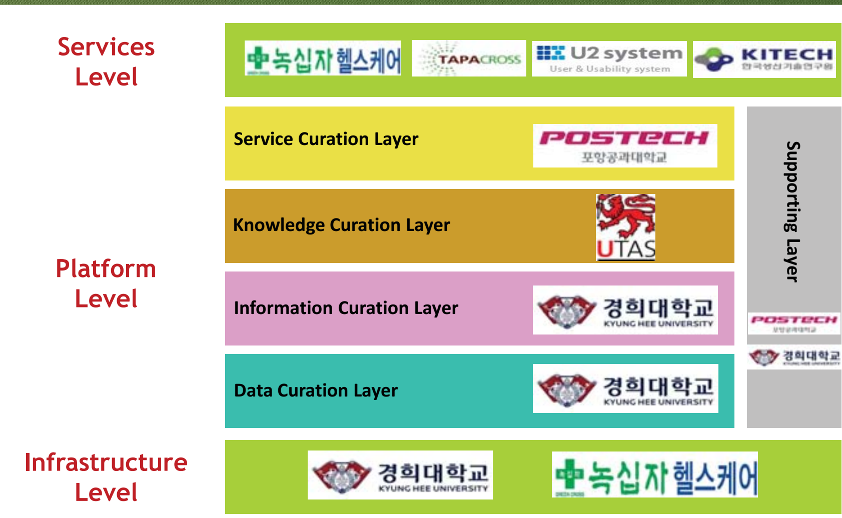
ICL
- Include **new sensing modalities** for the low-level context identification (heart rate, ambient audio)
- **Increase the number of low-level contexts and extend the** Mining Minds **Context Ontology** for enriching the high-level context inference process

KCL
- **Classification** model **and feature model** creation and integration with expert-driven knowledge
- **Validation mechanisms for** the existing **rule creation,** and population of **physical activity and nutrition rules**

SCL
- Improve the performance of rule-based reasoner with **dynamic data integration and conflict resolution strategies**
- **Augment the** capabilities of **interpreter** for personalizing recommendations **based on multiple context** information

SL
- **User satisfaction measurement and behavior change** based on **UIUX analysis and adaptation**
- Advanced and **complex analytics** based on **life log and big data**

Future Plan: MMv2.5 New Roles



Conclusion

Evolution of Mining Minds platform from V1.0 to V2.0

Developed the baseline infrastructure to support current and future services

V2.5 (May 2016)

Physical Activity + Nutrition combined scenario

Established overall uniqueness compared to similar existing systems

Demonstrated platform-oriented and layer-wise technical contributions

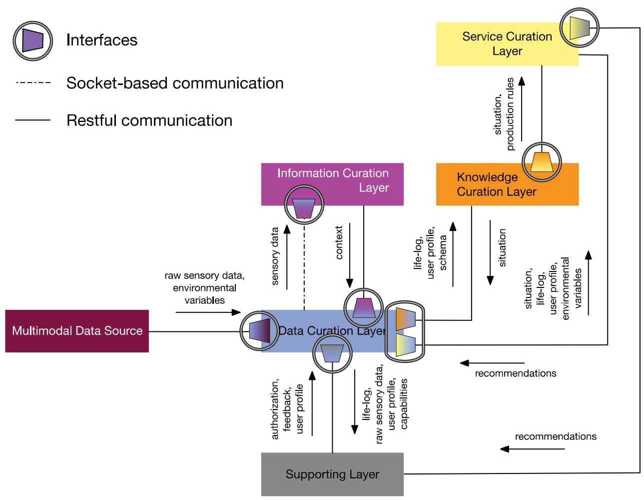
Appendix

경희대학교
KYUNG HEE UNIVERSITY

Software Engineering Principles

Low-coupling via interfaces

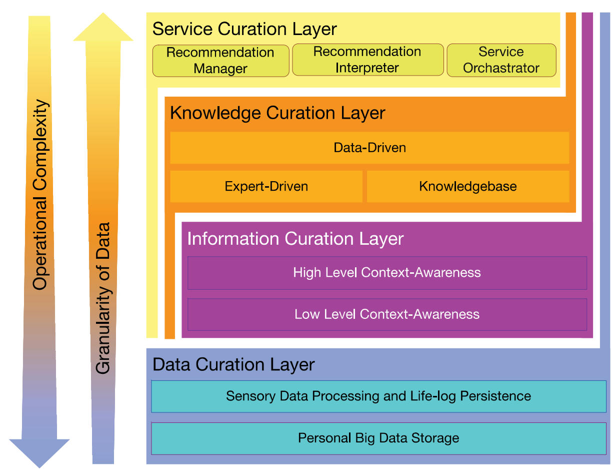
- Each layer provides access via interfaces
- Plug-n-play for more layers and 3rd party APIs
- Provides scalable and re-usable design to Mining Mind Platform



Software Engineering Principles

Highly-cohesive Implementation

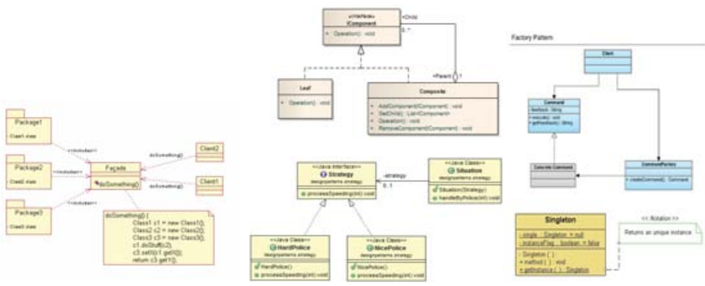
- Each layer works with its own abstraction of data
- Each layer is autonomous by operations

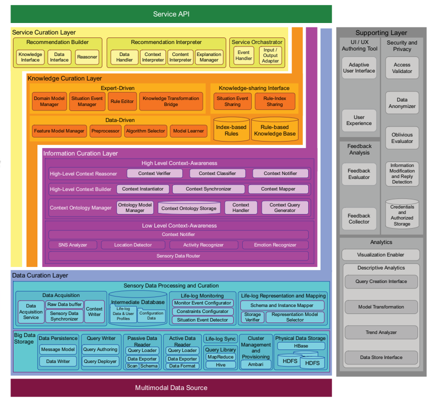


Software Engineering Principles

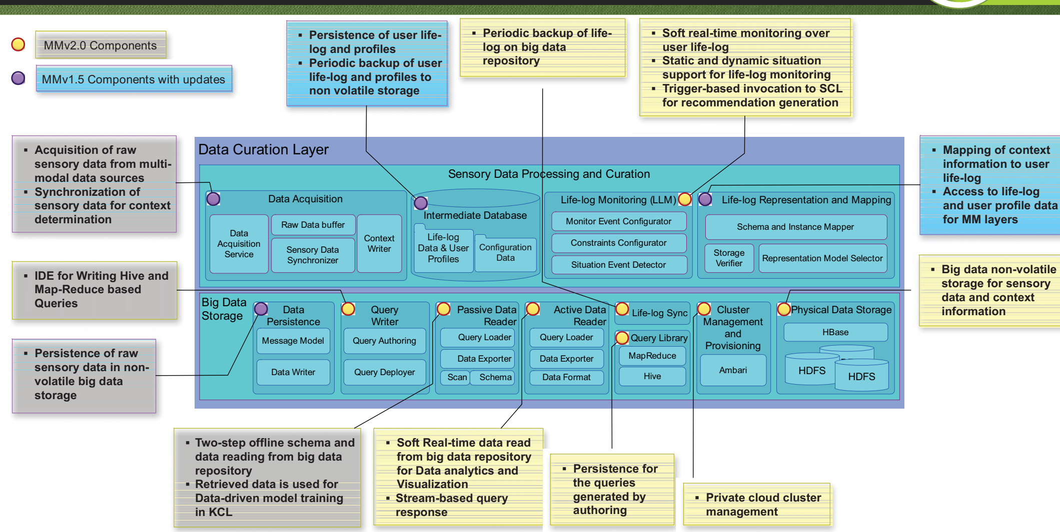
Agility at the Core of Implementation

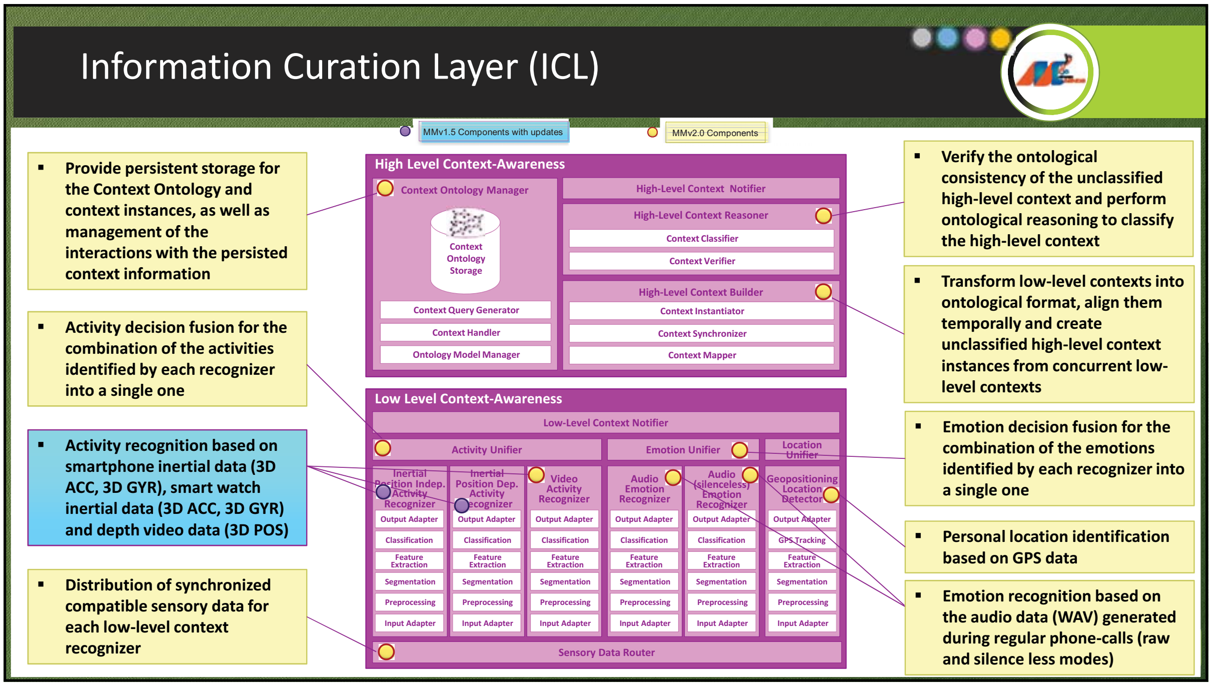
- Object Oriented Design Patterns are used for agility, re-usability, and extensibility

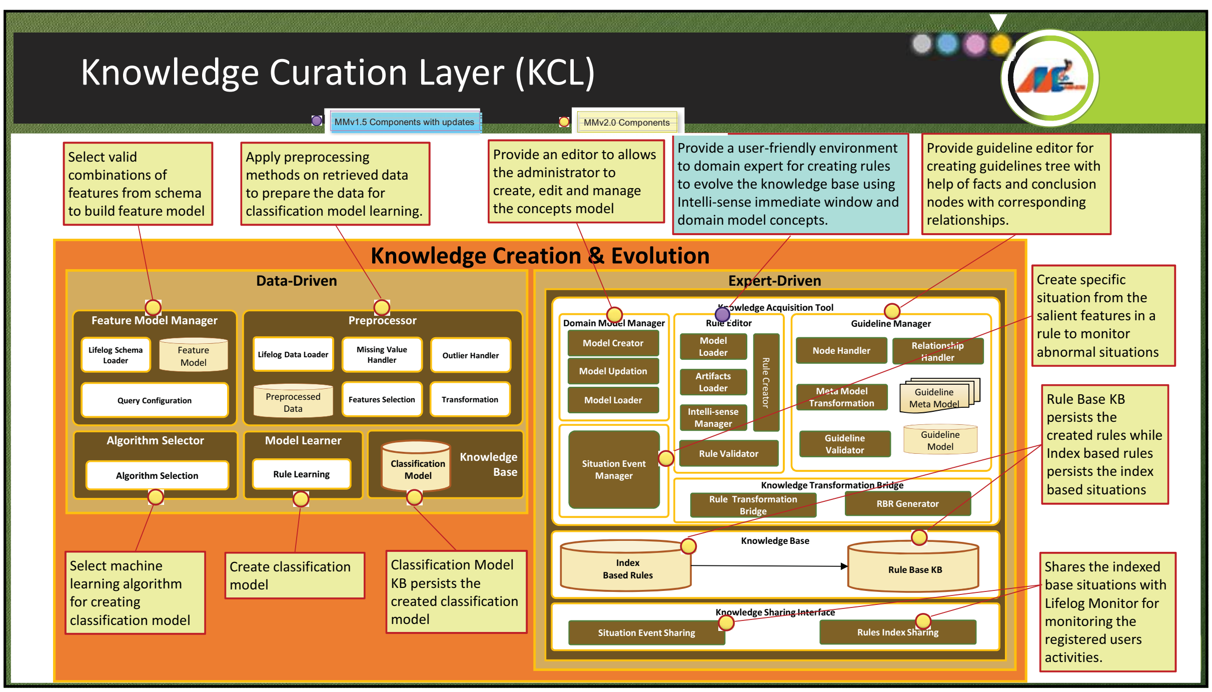


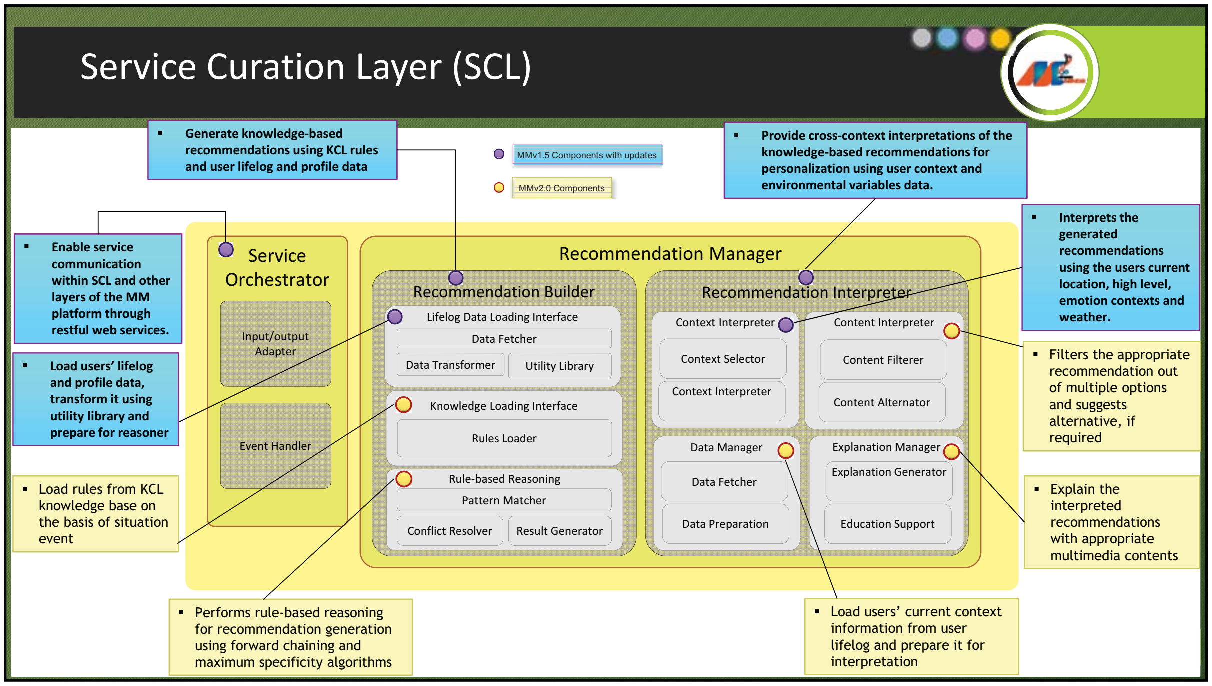


Data Curation Layer (DCL)

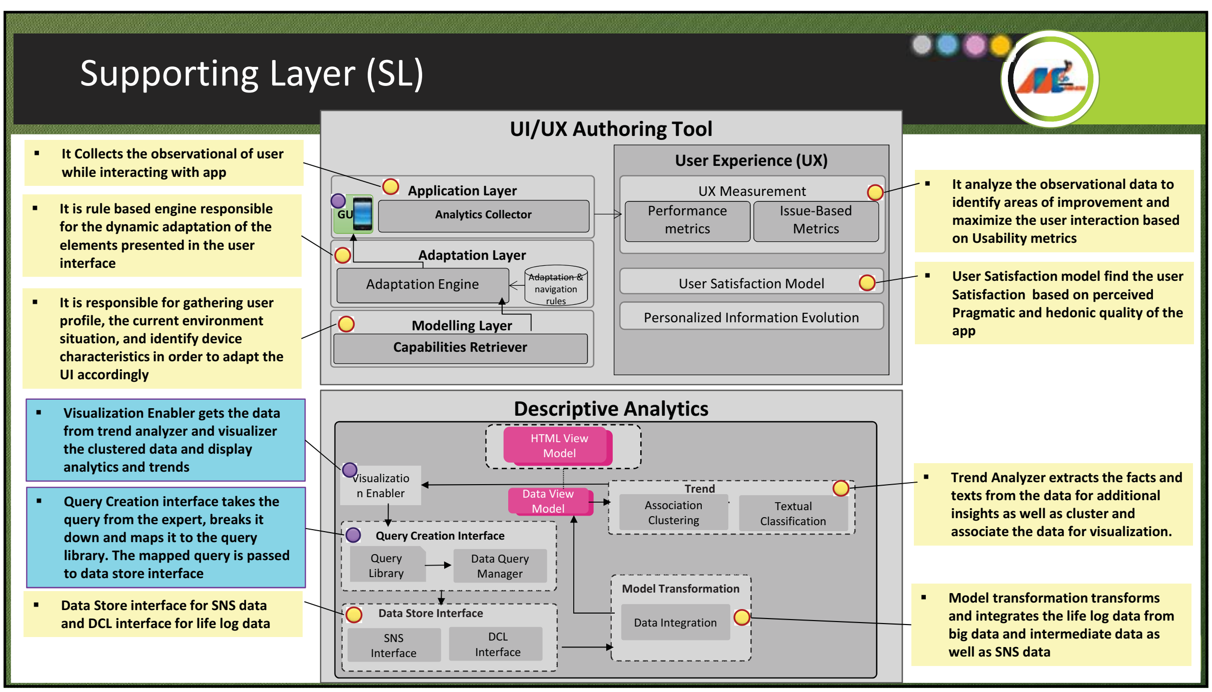



Information Curation Layer (ICL)
MMv1.5 Components with updates
MMv2.0 Components
Provide persistent storage for the Context Ontology and context instances, as well as management of the interactions with the persisted context information
Activity decision fusion for the combination of the activities identified by each recognizer into a single one
Activity recognition based on smartphone inertial data (3D ACC, 3D GYR), smart watch inertial data (3D ACC, 3D GYR) and depth video data (3D POS)
Distribution of synchronized compatible sensory data for each low-level context recognizer
High Level Context-Awareness
Context Ontology Manager
Context Ontology Storage
Context Query Generator
Context Handler
Ontology Model Manager
High-Level Context Notifier
High-Level Context Reasoner
Context Classifier
Context Verifier
High-Level Context Builder
Context Instantiator
Context Synchronizer
Context Mapper
Low Level Context-Awareness
Low-Level Context Notifier
Activity Unifier
Emotion Unifier
Location Unifier
Inertial Position Indep. Activity Recognizer
Inertial Position Dep. Activity Recognizer
Video Activity Recognizer
Audio Emotion Recognizer
Audio (silenceless) Emotion Recognizer
Geopositioning Location Detector
Output Adapter
Classification
GPS Tracking
Feature Extraction
Segmentation
Preprocessing
Input Adapter
Sensory Data Router
Verify the ontological consistency of the unclassified high-level context and perform ontological reasoning to classify the high-level context
Transform low-level contexts into ontological format, align them temporally and create unclassified high-level context instances from concurrent low-level contexts
Emotion decision fusion for the combination of the emotions identified by each recognizer into a single one
Personal location identification based on GPS data
Emotion recognition based on the audio data (WAV) generated during regular phone-calls (raw and silence less modes)


Knowledge Curation Layer (KCL)
MMv1.5 Components with updates
MMv2.0 Components
Select valid combinations of features from schema to build feature model
Apply preprocessing methods on retrieved data to prepare the data for classification model learning.
Provide an editor to allows the administrator to create, edit and manage the concepts model
Provide a user-friendly environment to domain expert for creating rules to evolve the knowledge base using Intelli-sense immediate window and domain model concepts.
Provide guideline editor for creating guidelines tree with help of facts and conclusion nodes with corresponding relationships.
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
Classification Model
Knowledge Base
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
Create specific situation from the salient features in a rule to monitor abnormal situations
Rule Base KB persists the created rules while Index based rules persists the index based situations
Shares the indexed base situations with Lifelog Monitor for monitoring the registered users activities.
Select machine learning algorithm for creating classification model
Create classification model
Classification Model KB persists the created classification model

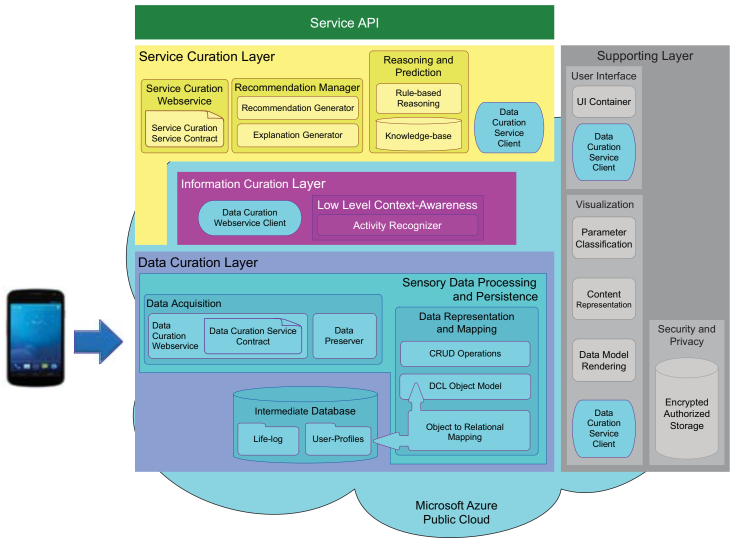


Service Curation Layer (SCL)
Generate knowledge-based recommendations using KCL rules and user lifelog and profile data
MMv1.5 Components with updates
MMv2.0 Components
Provide cross-context interpretations of the knowledge-based recommendations for personalization using user context and environmental variables data.
Enable service communication within SCL and other layers of the MM platform through restful web services.
Load users' lifelog and profile data, transform it using utility library and prepare for reasoner
Load rules from KCL knowledge base on the basis of situation event
Interprets the generated recommendations using the users current location, high level, emotion contexts and weather.
Filters the appropriate recommendation out of multiple options and suggests alternative, if required
Explain the interpreted recommendations with appropriate multimedia contents
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
Content Alternator
Data Manager
Data Fetcher
Data Preparation
Explanation Manager
Explanation Generator
Education Support
Performs rule-based reasoning for recommendation generation using forward chaining and maximum specificity algorithms
Load users' current context information from user lifelog and prepare it for interpretation



Supporting Layer (SL)
UI/UX Authoring Tool
It Collects the observational of user while interacting with app
It is rule based engine responsible for the dynamic adaptation of the elements presented in the user interface
It is responsible for gathering user profile, the current environment situation, and identify device characteristics in order to adapt the UI accordingly
Application Layer
GUI
Analytics Collector
Adaptation Layer
Adaptation Engine
Adaptation & navigation rules
Modelling Layer
Capabilities Retriever
User Experience (UX)
UX Measurement
Performance metrics
Issue-Based Metrics
User Satisfaction Model
Personalized Information Evolution
It analyze the observational data to identify areas of improvement and maximize the user interaction based on Usability metrics
User Satisfaction model find the user Satisfaction based on perceived Pragmatic and hedonic quality of the app
Descriptive Analytics
Visualization Enabler gets the data from trend analyzer and visualizer the clustered data and display analytics and trends
Query Creation interface takes the query from the expert, breaks it down and maps it to the query library. The mapped query is passed to data store interface
Data Store interface for SNS data and DCL interface for life log data
HTML View Model
Visualization Enabler
Data View Model
Trend
Association Clustering
Textual Classification
Query Creation Interface
Query Library
Data Query Manager
Data Store Interface
SNS Interface
DCL Interface
Model Transformation
Data Integration
Trend Analyzer extracts the facts and texts from the data for additional insights as well as cluster and associate the data for visualization.
Model transformation transforms and integrates the life log data from big data and intermediate data as well as SNS data

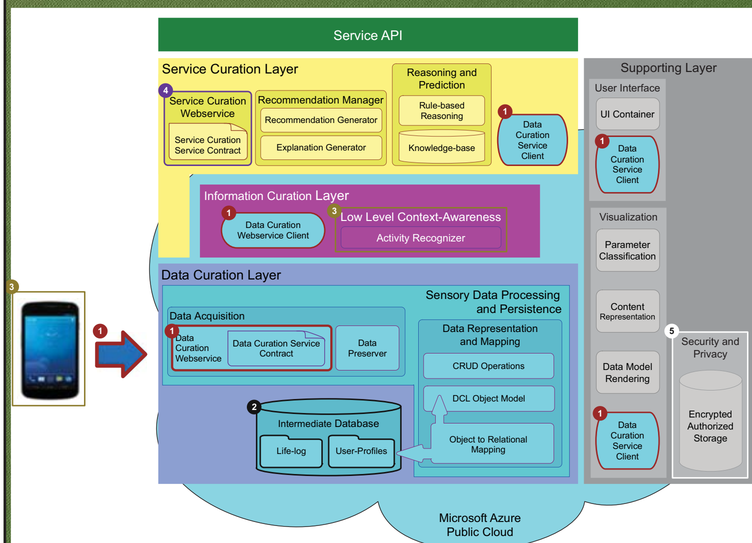
MM Ver. 1.0 (Release Jan 2015)



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

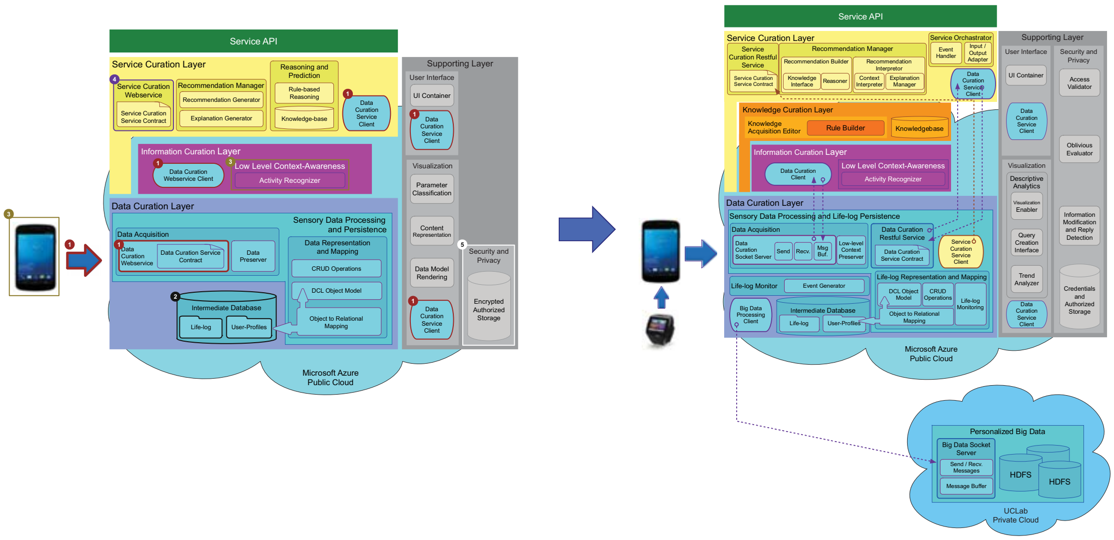
MM Ver. 1.0 (Release Jan 2015)



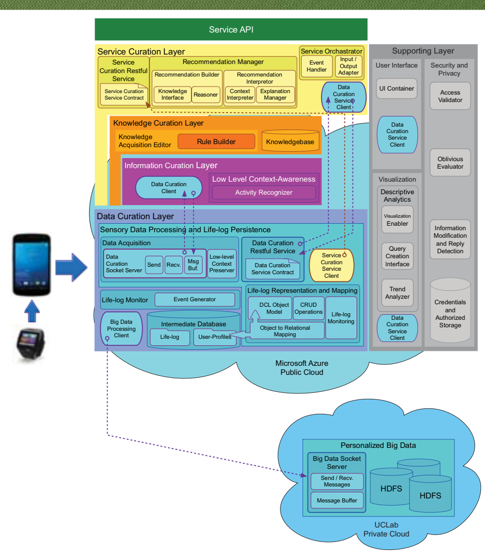
Technological Limitations

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

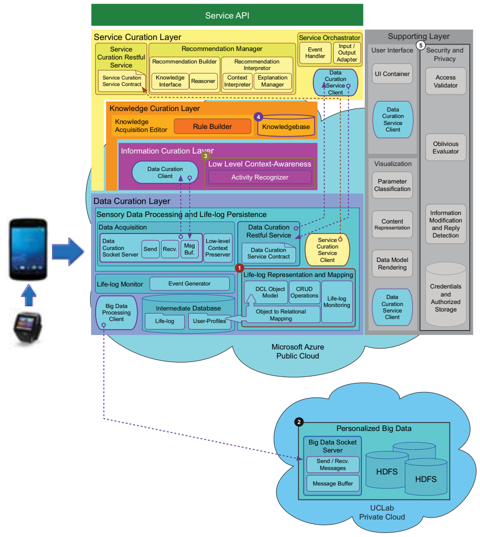


MM Ver. 1.5 (Release Apr 2015)



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

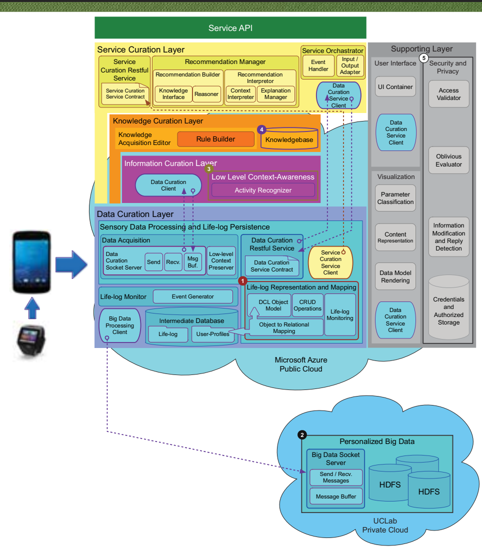
MM Ver. 1.5 (Release Apr 2015)

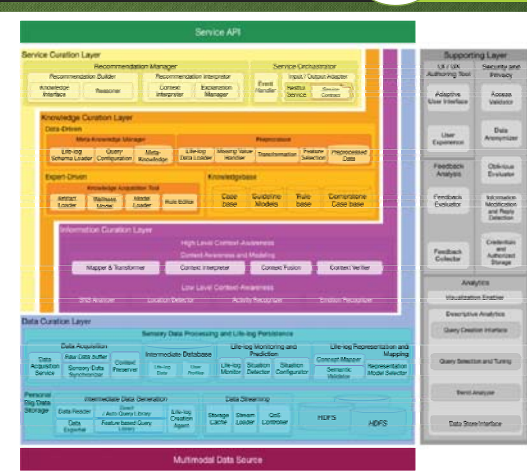


Limitations

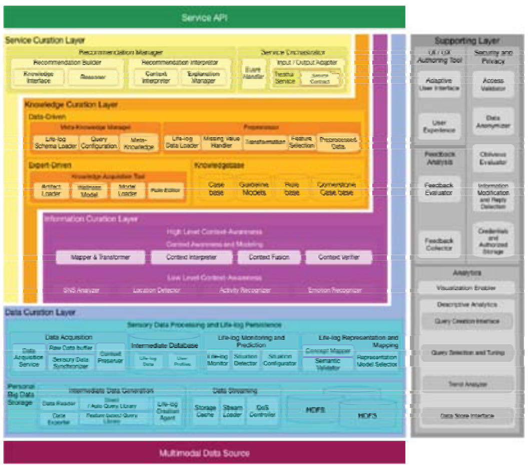
1. Object Oriented and Relational life-log models
2. Big data is only used for persistence
3. Lack of High-level context determination
4. Only static rules as knowledge
5. Anonymized Access

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



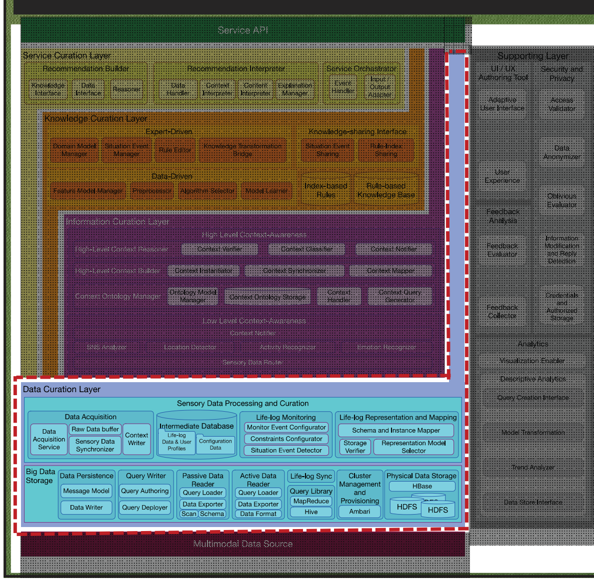


MM Ver. 2.0 (Release Dec 2015)



1. Input Data : Sensors (Smartphone-based Accelerometer & GPS, Wearable device) + 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. Feedback Analysis
    4. Personalized Predictions
    5. Descriptive Analytics
    6. User Experience (UX)

Data Curation Layer (DCL)



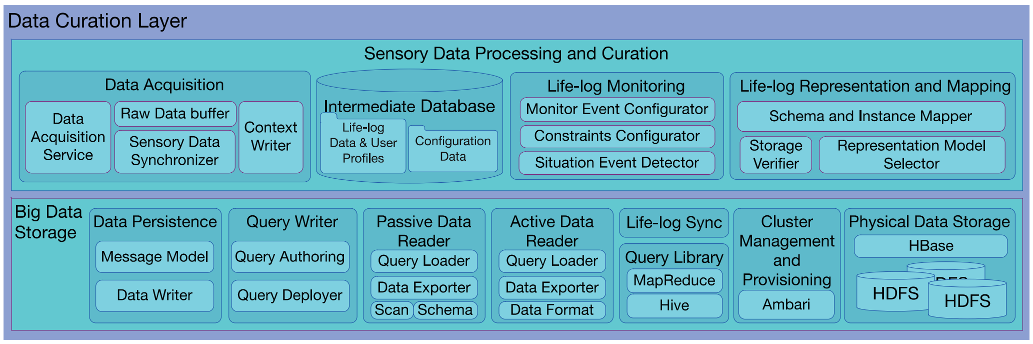
DCL Goal

To curate large-volume of heterogeneous data representing human-activities and behavior such that it can be be used for situation monitoring and decision making regarding personal health and wellness.

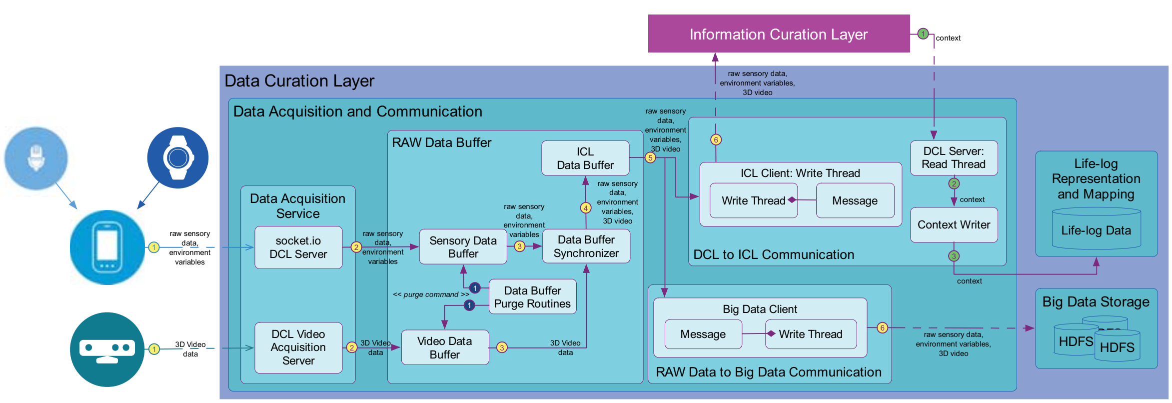
DCL Objectives

- **Acquisition of multimodal sensory data in real-time with support to its large volume**
- **Curation of recognized activities from multimodal data sources as a human life-log.**
- **Analysis of life-log instances for situation detection**
- **High Performance Inter- and intra-layer communication for sensory and life-log data read and write**

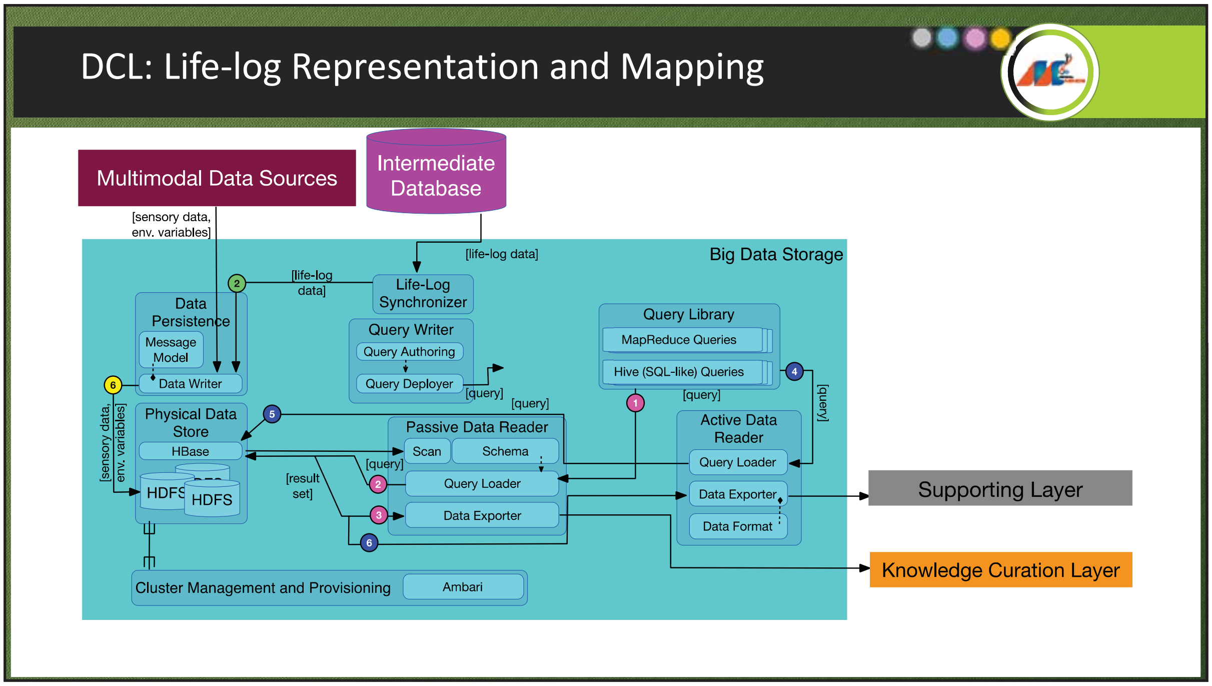
Data Curation Layer (DCL)

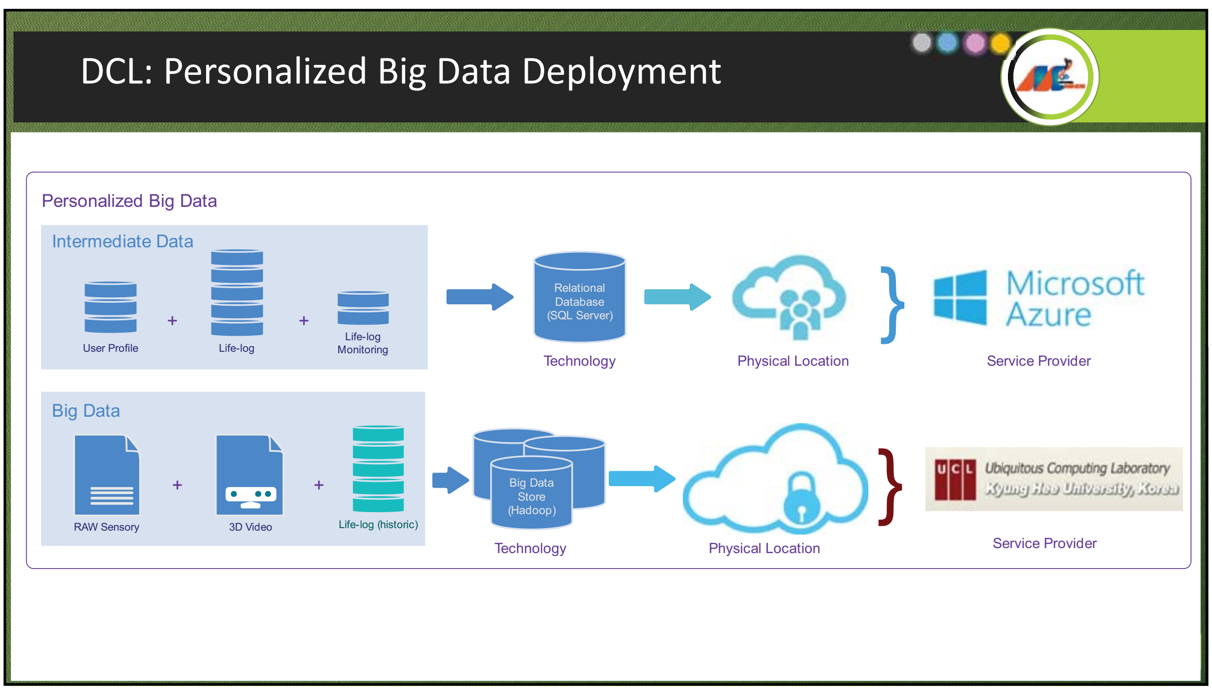


DCL: Data Acquisition and Communication

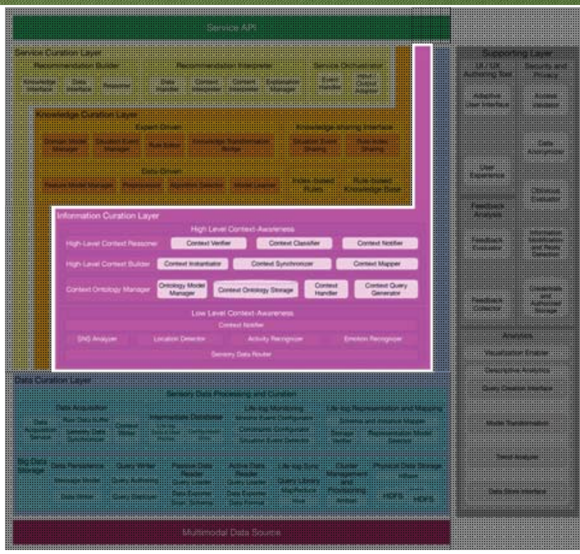


Asynchronous Non-Blocking communication


DCL: Life-log Representation and Mapping
Multimodal Data Sources
Intermediate Database
[sensory data, env. variables]
[life-log data]
Big Data Storage
[life-log data]
Life-Log Synchronizer
Data Persistence
Message Model
Data Writer
Query Writer
Query Authoring
Query Deployer
[query]
Query Library
MapReduce Queries
Hive (SQL-like) Queries
[query]
[query]
[query]
[sensory data, env. variables]
Physical Data Store
HBase
HDFS
HDFS
Passive Data Reader
Scan
Schema
[query]
Query Loader
Data Exporter
[result set]
Active Data Reader
Query Loader
Data Exporter
Data Format
Supporting Layer
Knowledge Curation Layer
Cluster Management and Provisioning
Ambari


DCL: Personalized Big Data Deployment
Personalized Big Data
Intermediate Data
User Profile
+
Life-log
+
Life-log Monitoring
Relational Database (SQL Server)
Technology
Physical Location
Microsoft Azure
Service Provider
Big Data
RAW Sensory
+
3D Video
+
Life-log (historic)
Big Data Store (Hadoop)
Technology
Physical Location
Ubiquitous Computing Laboratory
Kyung Hee University, Korea
Service Provider

Information Curation Layer (ICL)



ICL Goal

To convert the wide-spectrum of data obtained from the user interaction with the real and cyber-world, into concepts or categories, such as physical activities, emotional states, locations, social patterns and abstract contexts

ICL Objectives

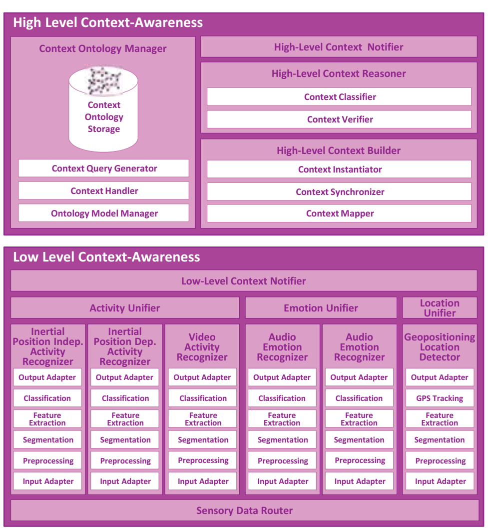
High Level Context Awareness

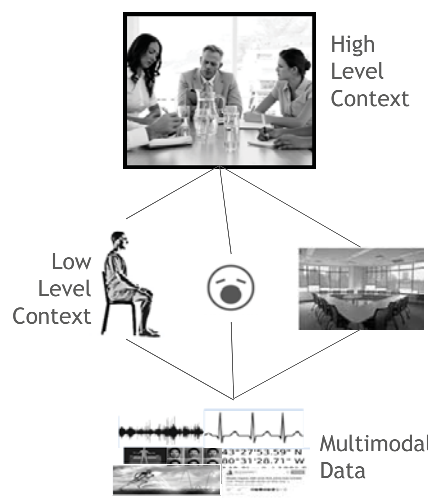
- Context Modeling – define the relationships among contexts of diverse granularity level
- Context Inference – identify the abstract context from low-level information

Low Level Context Awareness

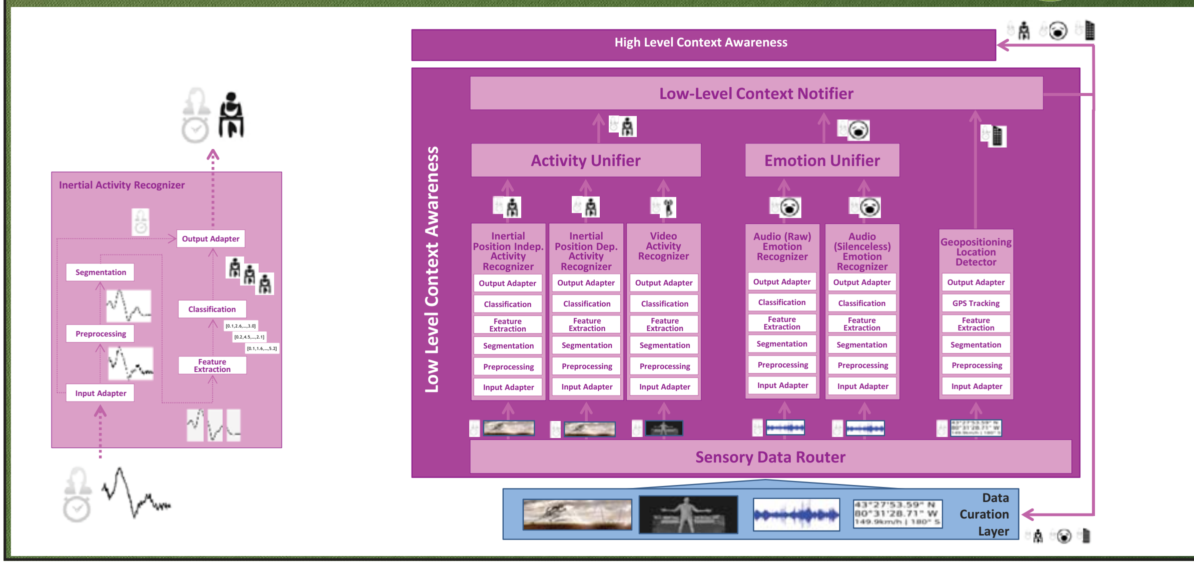
- Acquisition – collect and process sensory and social data
- Recognition – recognize different physical activities, location and emotional states of the user
- Fusion – combine decisions from different recognizers of the same category

Information Curation Layer (ICL)

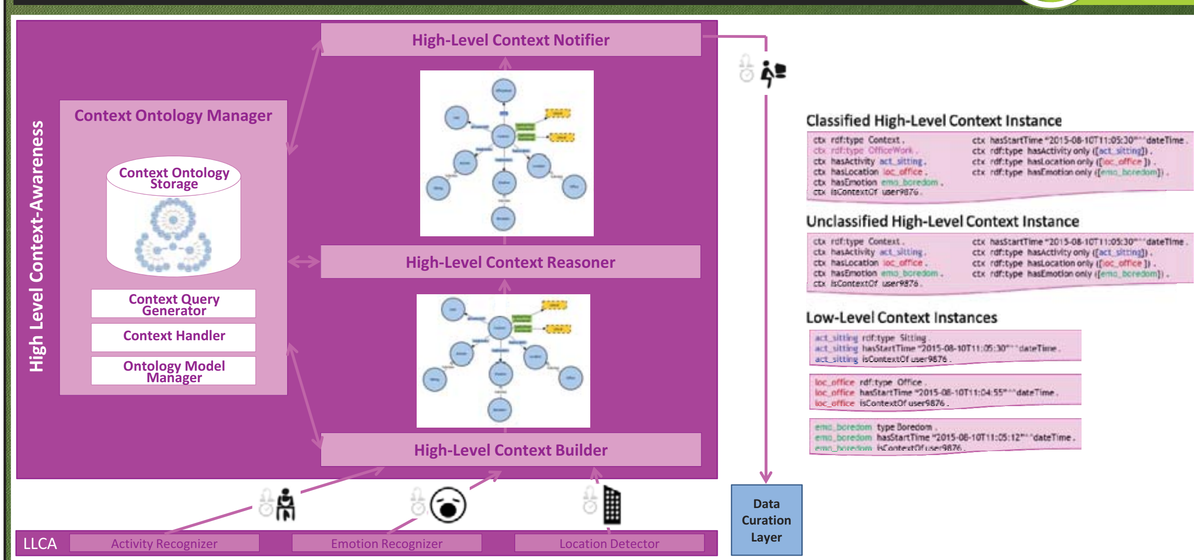




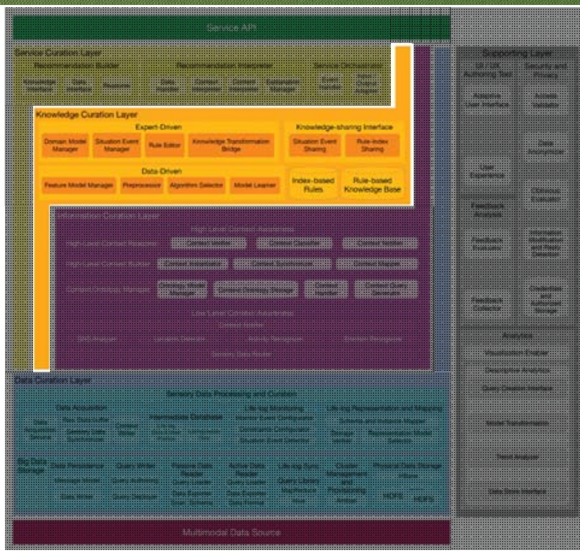
ICL 2.0: Low Level Context Awareness



ICL: High Level Context Awareness



Knowledge Curation Layer (KCL)



KCL Goal

To provide knowledge acquisition tools to incorporate data-driven knowledge acquisition model and support guidelines and expert knowledge for open knowledge

KCL Objectives

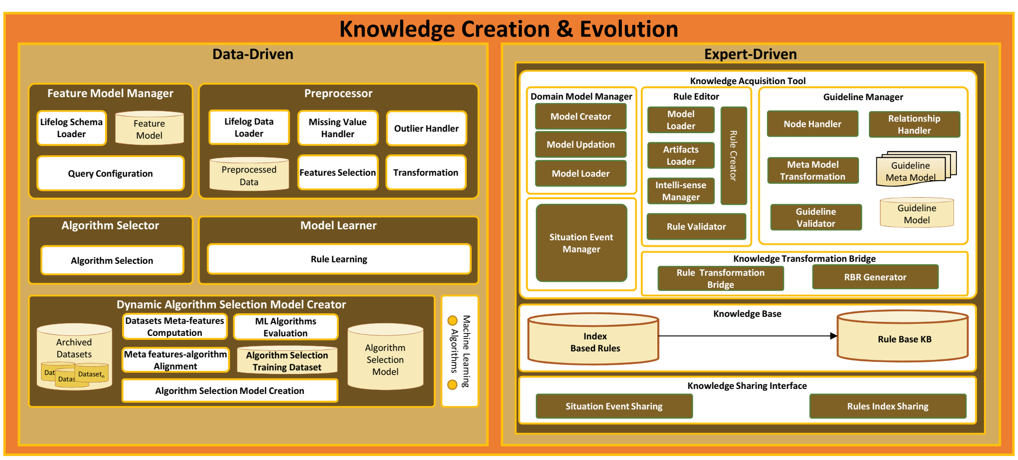
Data-Driven Knowledge Acquisition

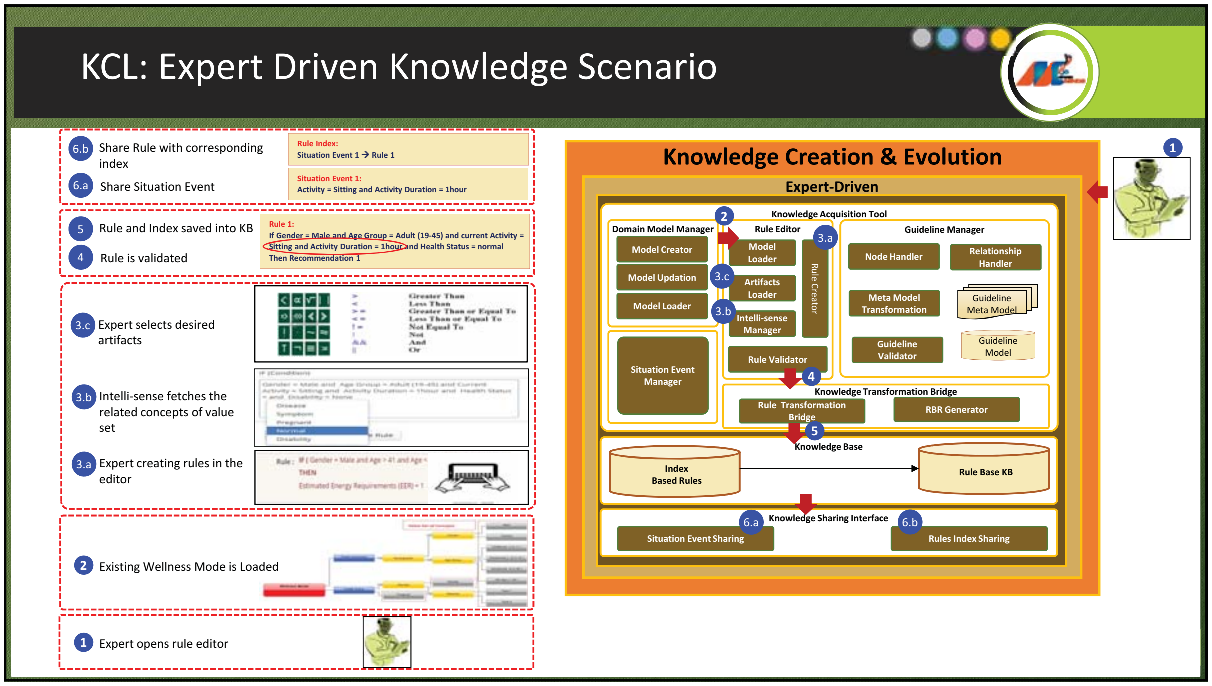
- Enabling data driven knowledge acquisition using life-log meta knowledge
- Providing support of Feature Modeling to facilitate expert in creating classification model

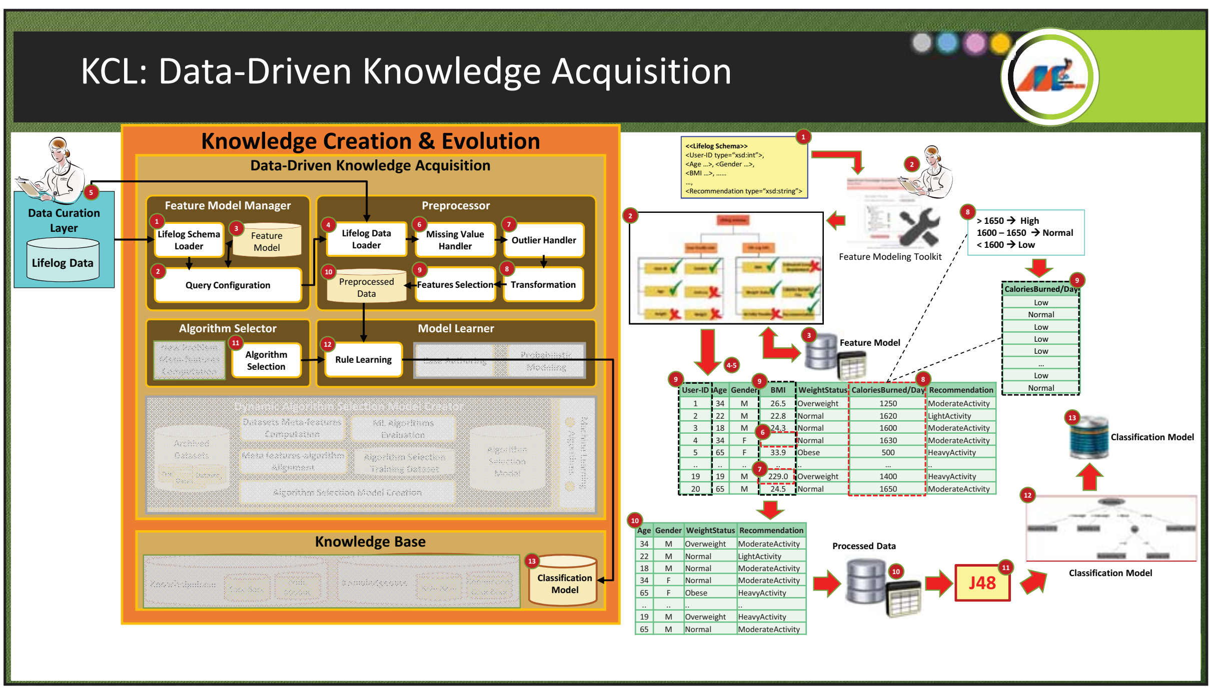
Expert-Driven Knowledge Acquisition

- Enabling rule editor to facilitate domain expert in creating rules
- Incorporating guidelines into knowledge using easy to manageable editor

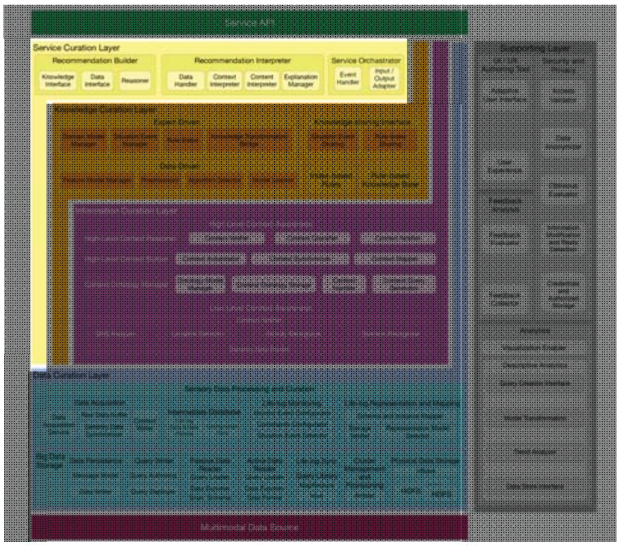
KCL: Architecture




KCL: Expert Driven Knowledge Scenario
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
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


KCL: Data-Driven Knowledge Acquisition
Knowledge Creation & Evolution
Data-Driven Knowledge Acquisition
Data Curation Layer
Lifelog Data
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
Knowledge Base
Classification Model
Feature Modeling Toolkit
Feature Model
> 1650 → High
1600 – 1650 → Normal
< 1600 → Low
Processed Data
J48
Classification Model

Service Curation Layer (SCL)



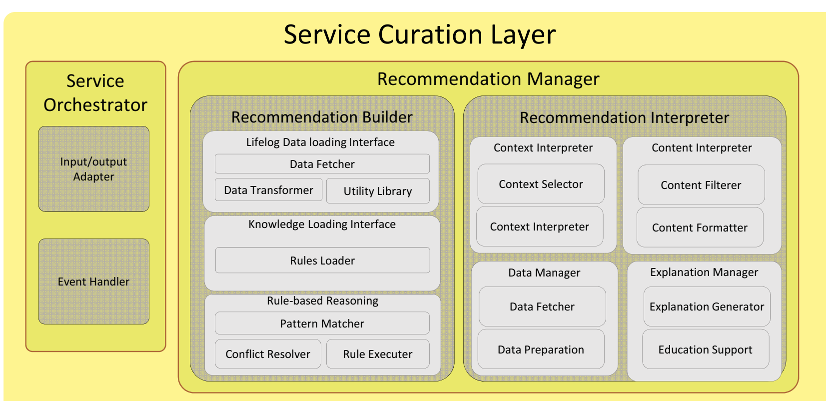
SCL Goal

To provide knowledgeable and personalized wellbeing recommendations using domain knowledge, users' conditions, context, and preferences.

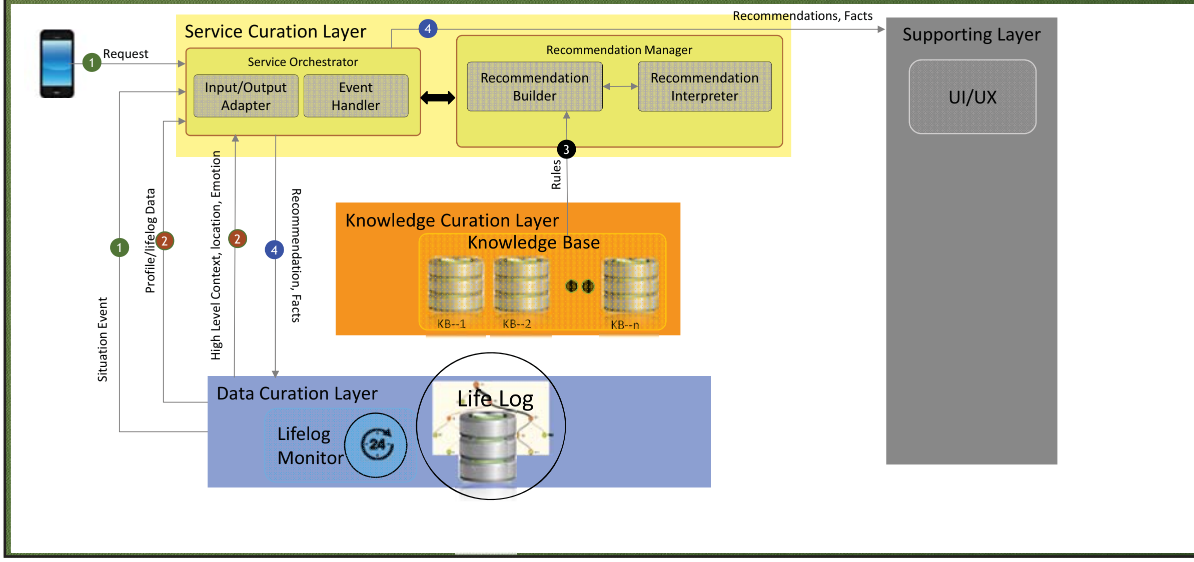
SCL Objectives

- Enabling service communication within SCL and other layers of the MM platform through restful web services.
- Generating knowledgeable contents for physical activities and nutrition recommendations, using users profile information
- Interpreting physical activities and nutrition recommendations contents for personalization and explanation, using user context and environmental variables.

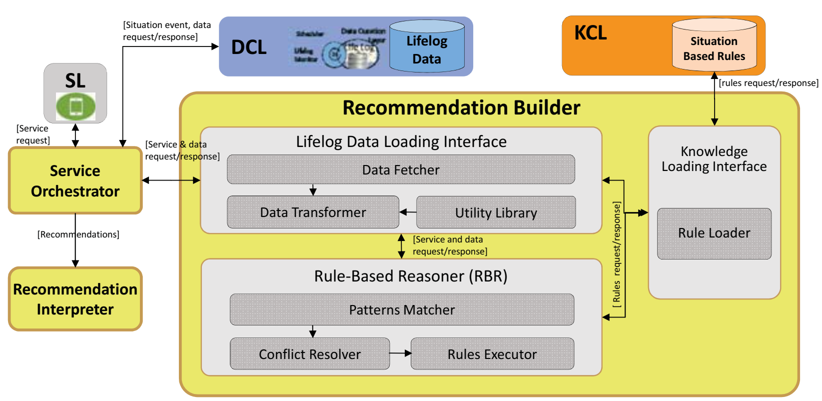
Service Curation Layer (SCL)



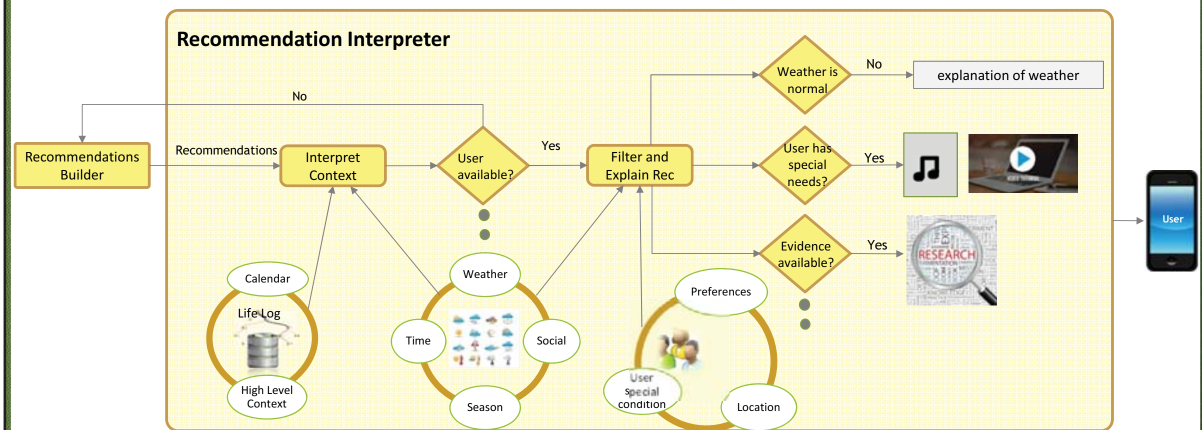
SCL: Service Orchestrator



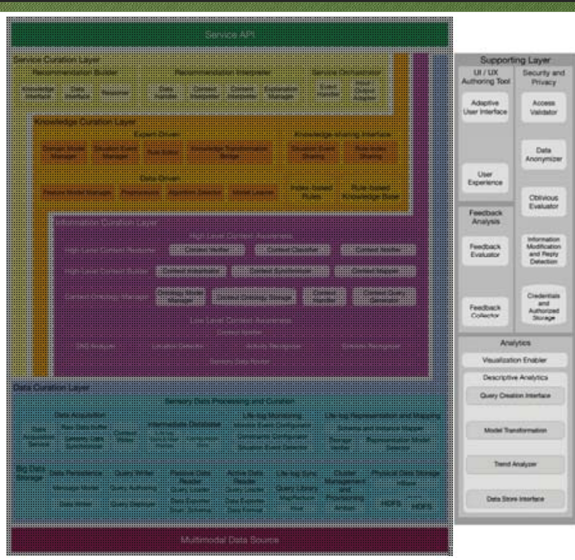
SCL: Recommendation Builder



SCL: Recommendation Interpreter



Supporting Layer (SL)



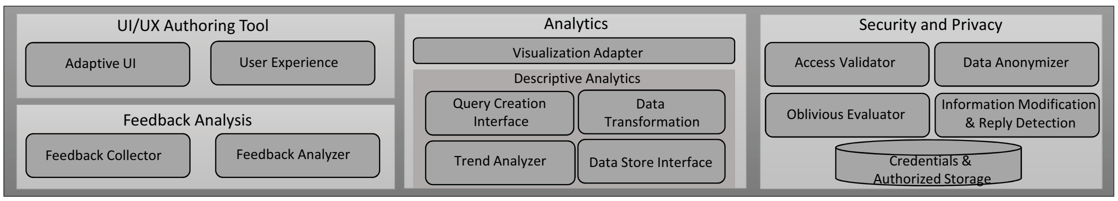
SL Goal

To provide intuitive user interface with the support of analytics and security and privacy for obtaining user feedback in improving the user experience.

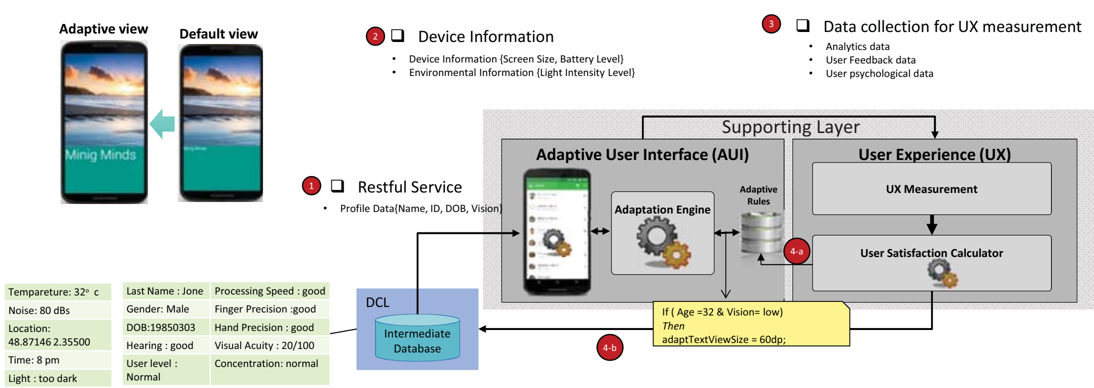
SL Objectives

- Continuous evolution of Adaptive user interface with contextual information change to provide better user experience
- User satisfaction as a feedback metric
- High entropy solution with efficient execution time named “Reflection”.
- Visualization for latest trends to provide decision making ability to domain expert
- Parameter Classification and structuring attributes for intuitive analytics

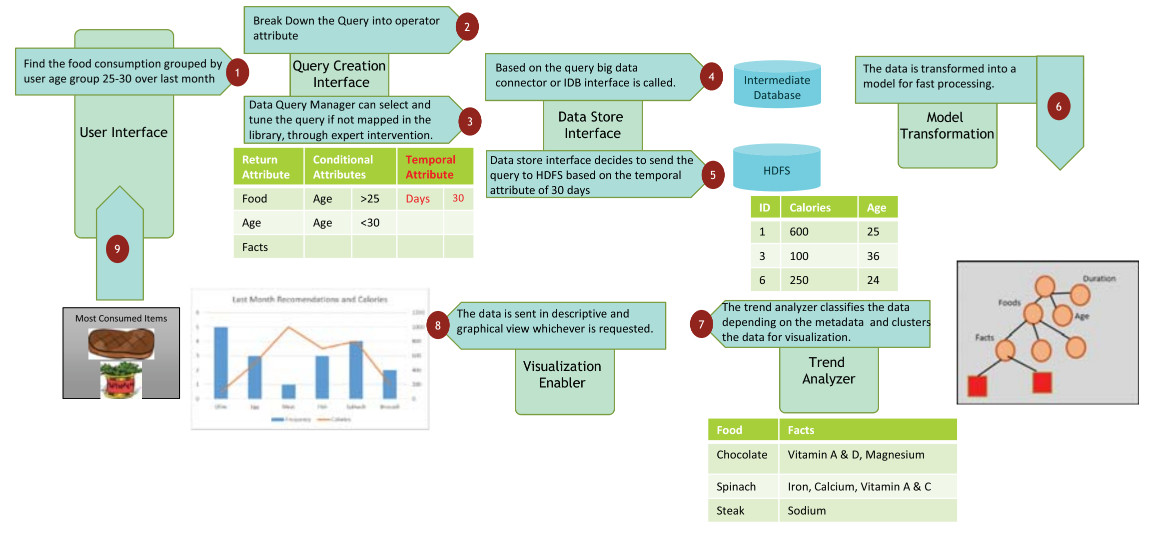
Supporting Layer (SL)



SL: UI/UX Adaptation



SL: Descriptive Analytics



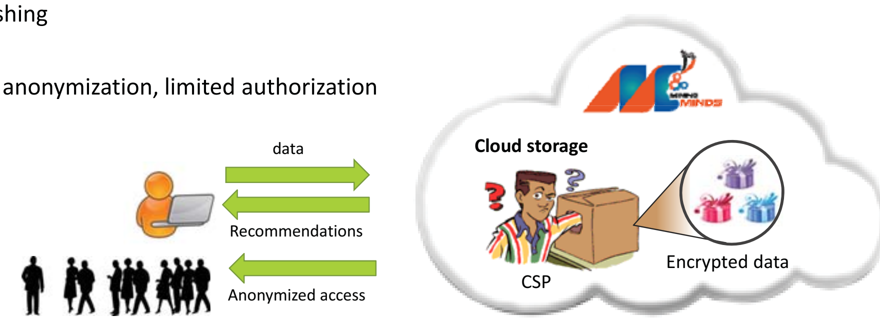
Return Attribute	Conditional Attributes		Temporal Attribute	
Food	Age	>25	Days	30
Age	Age	<30		
Facts				

ID	Calories	Age
1	600	25
3	100	36
6	250	24

Food	Facts
Chocolate	Vitamin A & D, Magnesium
Spinach	Iron, Calcium, Vitamin A & C
Steak	Sodium

SL: Privacy and security

- Mining Mind
  - Personalized information
  - Generate recommendations using this information
- To achieve **security** over personal data
  - Encryption, Hashing
- To Achieve **privacy**
  - Partial sharing, anonymization, limited authorization



경희대학교
KYUNG HEE UNIVERSITY

Future Plan: MMv2.5 Technical Steps

DCL

- Improve the **data acquisition** implementation with **soft-realtime and data synchronization** support
- Extended support for **dynamic situations** with **non-linear life-log monitoring**
- **Collection** and enrichment of raw sensory data for a **large-scale** comprehensive **data set**
- **Extended authoring of queries for** active data read over **big data** storage

ICL

- **Improve the performance of** the low-level context **recognition** models (particularly for the emotions)
- Include **new sensing modalities** for the low-level context identification (heart rate, ambient audio)
- **Increase the number of** identifiable low and high-level **contexts** (activities, emotions, locations, high-level)
- **Extend the** Mining Minds **Context Ontology** for enriching the high-level context inference process

KCL

- **Classification** model **and feature model** creation
- Data preprocessing and **integration with big data** repository
- **Domain model manager** to create different domain models
- **Validation mechanisms for** the existing **rule creation,** and population of **physical activity and nutrition rules**

SCL

- Improve the performance of rule-based reasoner with **dynamic data integration and conflict resolution strategies**
- Enable the rule-based reasoner framework to support the **generation to nutrition recommendations**
- **Augment the** capabilities of **interpreter** for personalizing recommendations **based on multiple context** information
- **Extend the explanatory capabilities** of interpreter for easier understanding of the users

SL

- **User satisfaction measurement** based on user experience analytics and feedback
- **Behavior change** based on **UIUX adaptation**
- Advanced and **complex analytics** based on **life log and big data**
- **More comprehensive expert view** in terms of query library, visualization and statistics, and data facts for every user

Section 3

Mining Minds Features Evolution

Mining Minds Version 2

Principal Investigator:
Professor Sungyoung Lee

Contents

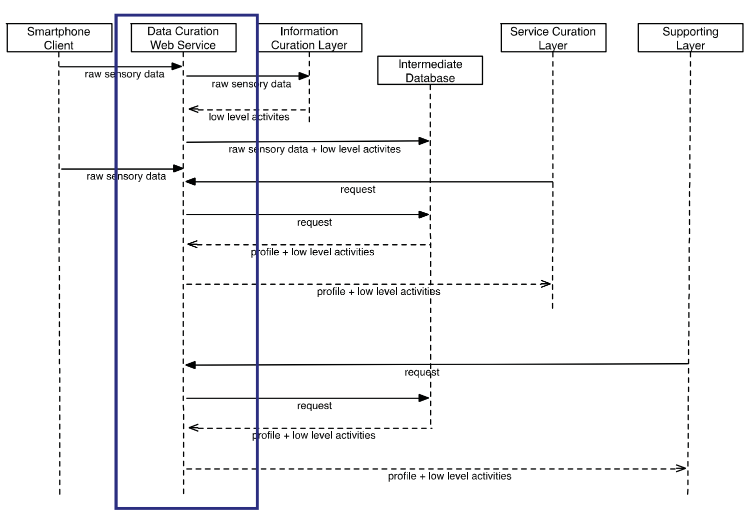
- Data Curation Layer (DCL 2.0)
- Information Curation Layer (ICL 2.0)
- Knowledge Curation Layer (KCL 2.0)
- Service Curation Layer (SCL 2.0)
- Supporting Layer (SL 2.0)

Data Curation Layer (DCL)

Challenges and solutions (MMV2-DCL)

1. Communication

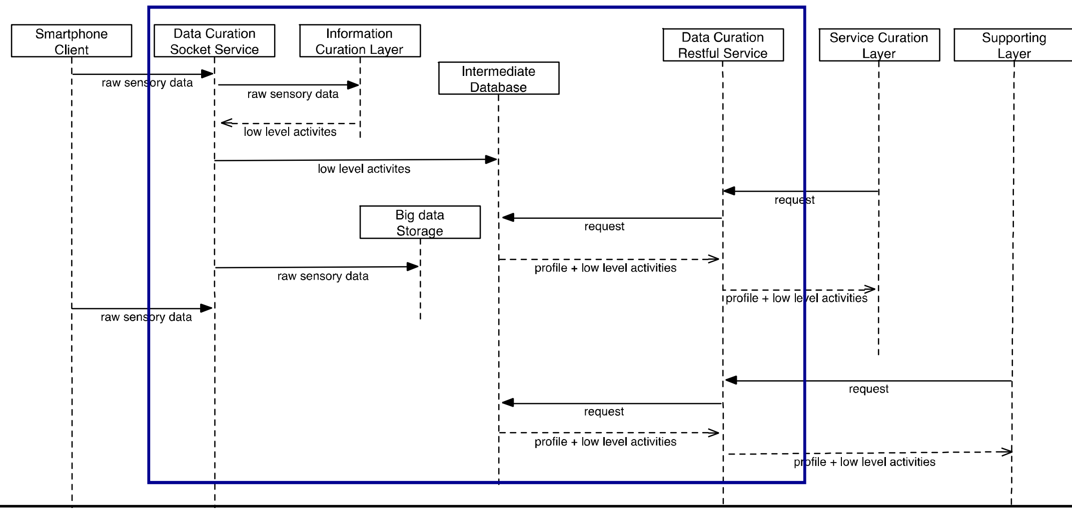
	Benefit		Limitation
V1. SOAP-based Web services →	Faster Implementation	→	Performance bottleneck



Challenges and solutions (MMV2-DCL)

1. Communication

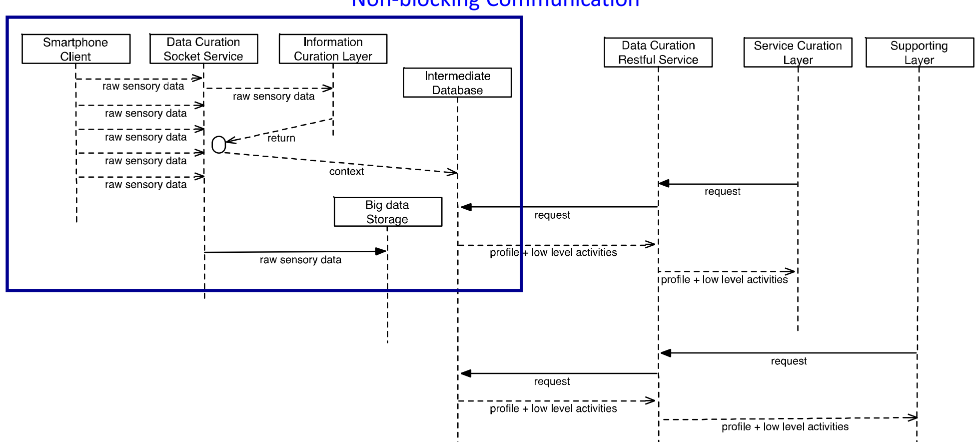
		Benefit		Limitation
V1. SOAP-based Web services	→	Faster Implementation	→	Performance bottleneck
V1.5 Sockets + Restful Services	→	Performance	→	Scalability (Blocking Communication)



Challenges and solutions (MMV2-DCL)

1. Communication

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

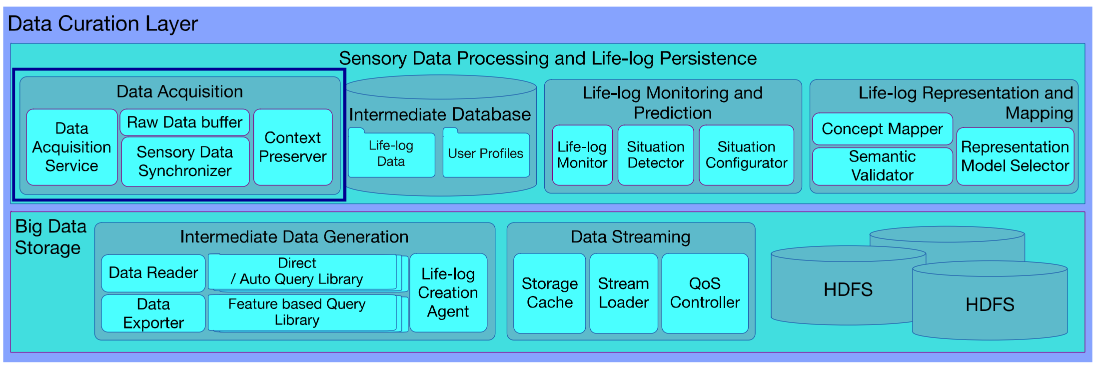


Challenges and solutions (MMV2-DCL)

1. Communication

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

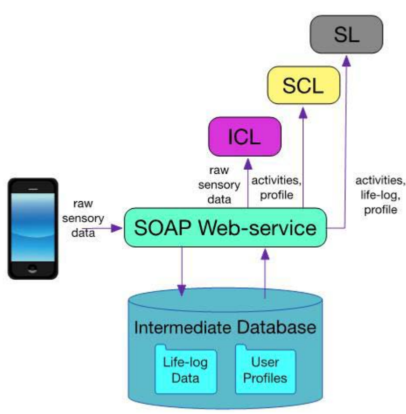
Confidence {Medium → High}



Challenges and solutions (MMV2-DCL)

2. Big Data + Intermediate DB

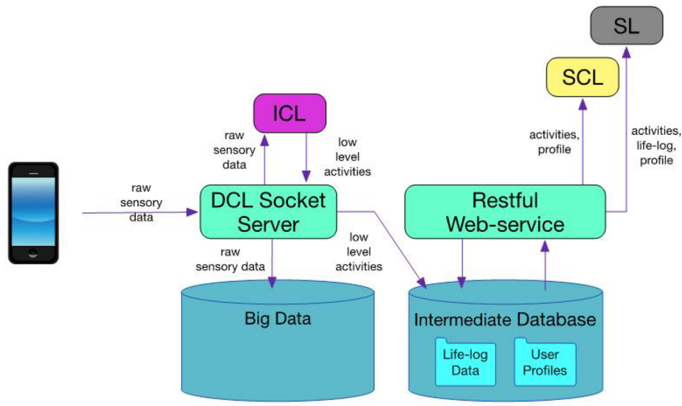
		Benefit		Limitation
V1. Only Intermediate DB	→	Faster Implementation	→	Intermediate DB Stress (Performance Bottleneck)



Challenges and solutions (MMV2-DCL)

2. Big Data + Intermediate DB

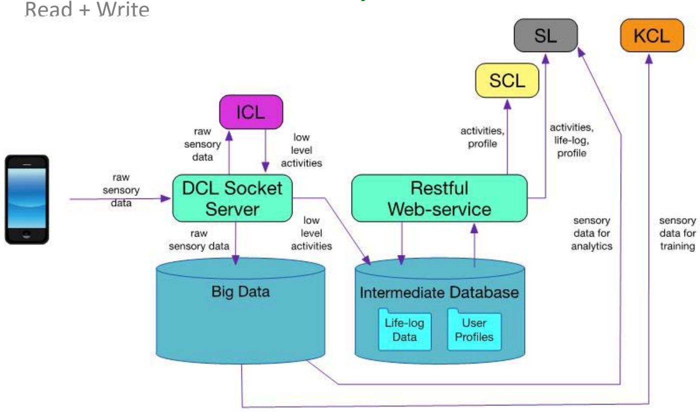
		Benefit		Limitation
V1. Only Intermediate DB	→	Faster Implementation	→	Intermediate DB Stress (Performance Bottleneck)
V1.5 Big Data For Sensory Data Write	→	Performance	→	No Analytics on Sensory Data + No Data for Training



Challenges and solutions (MMV2-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		1. Upfront Requirement Analysis for Big Data Queries 2. File distribution structure for Big Data Fragmentation 3. Ontology / Model for faster file location indexing

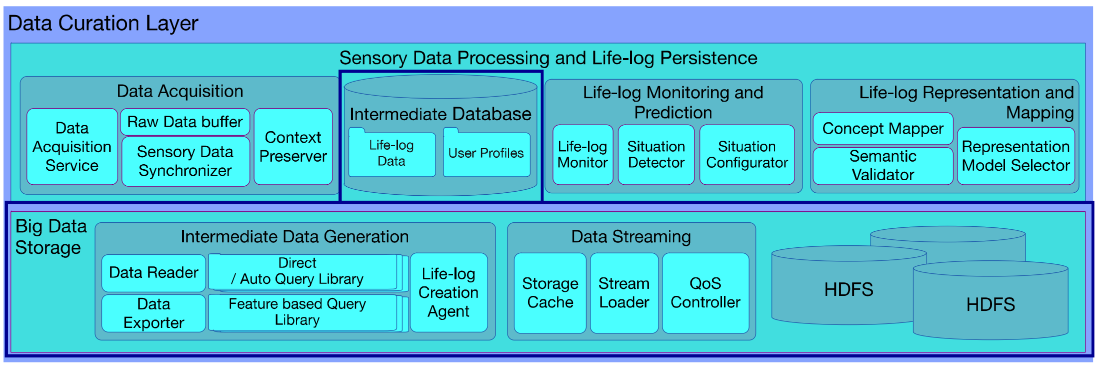


Challenges and solutions (MMV2-DCL)

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	{	1. Upfront Requirement Analysis for Big Data Queries 2. File distribution structure for Big Data Fragmentation 3. Ontology / Model for faster file location indexing

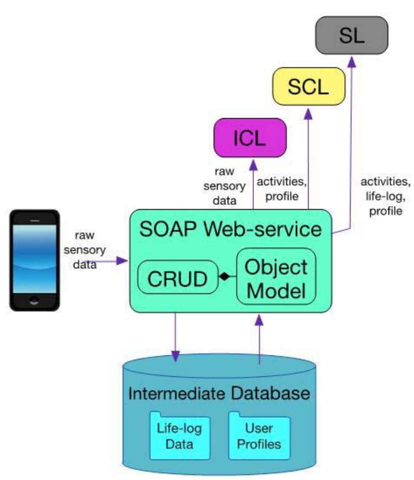
Confidence {Low → Medium}



Challenges and solutions (MMV2-DCL)

3. Life-log Representation and Mapping

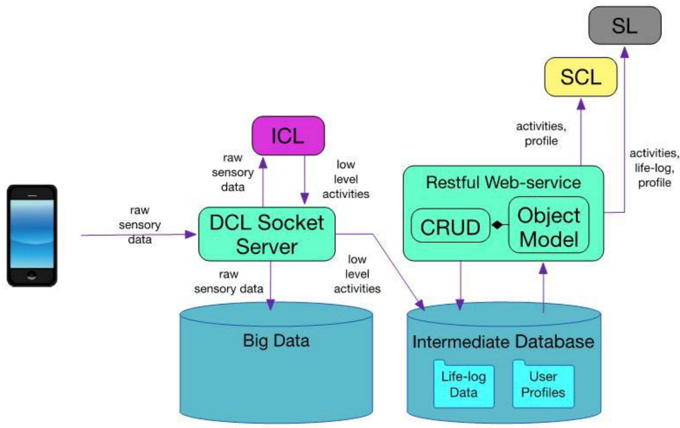
		Benefit		Limitation
V1. Object Model	→	Performance	→	No Inference (No High Level Context in ICL)



Challenges and solutions (MMV2-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)

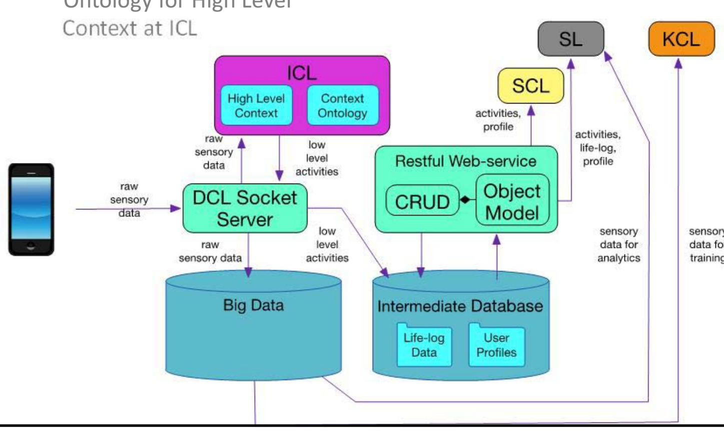


Challenges and solutions (MMV2-DCL)

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 Object Oriented Life-log + Ontology for High Level Context at ICL	→	Performance + Inference		

1. Only ICL requires to Inference , ICL can host an ontology natively
2. Triples will be stored in Intermediate DB; however KCL and SCL will read data via Interface (XML, JSON, etc.)

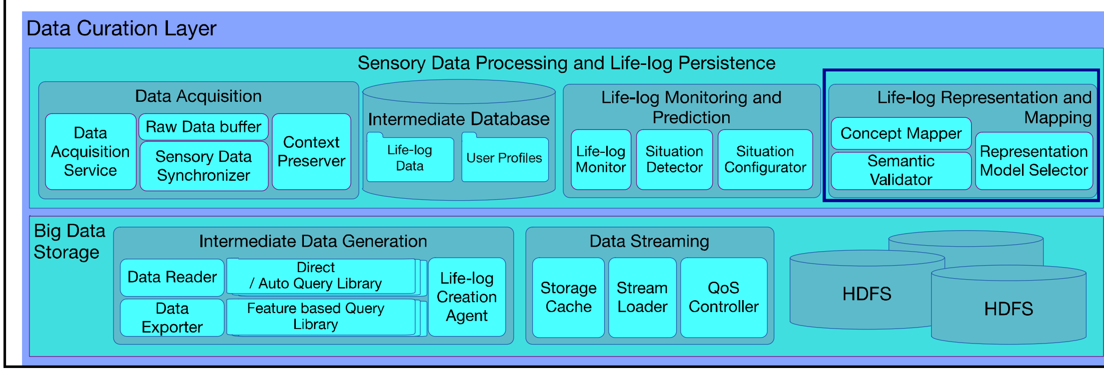


Challenges and solutions (MMV2-DCL)

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 Object Oriented Life-log + Ontology for High Level Context at ICL	→	Performance + Inference	{	1. Only ICL requires to Inference , ICL can host an ontology natively 2. Triples will be stored in Intermediate DB; however KCL and SCL will read data via Interface (XML, JSON, etc.)

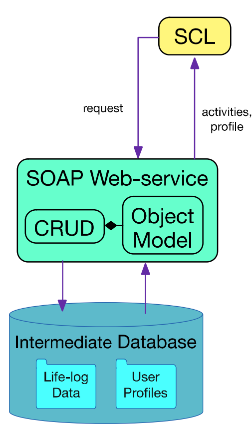
Confidence {Low}



Challenges and solutions (MMV2-DCL)

4. Life-log Monitoring and Prediction

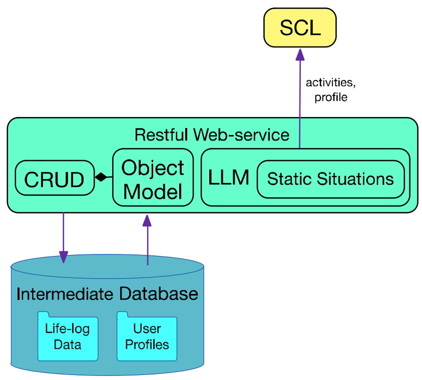
		Benefit		Limitation
V1. NA	→	NA	→	No trigger/situation based recommendations



Challenges and solutions (MMV2-DCL)

4. Life-log Monitoring and Prediction

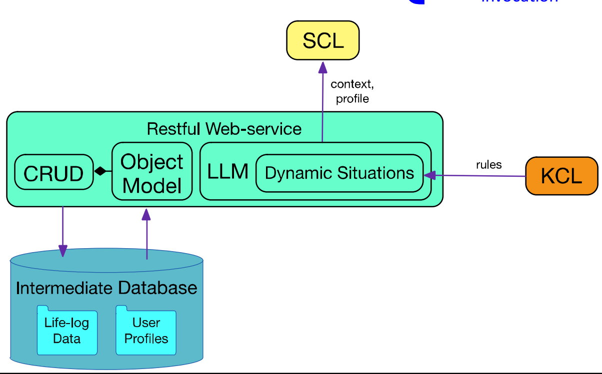
		Benefit		Limitation
V1. NA	→	NA	→	No trigger/situation based recommendations
V1.5 Life-log Monitoring (LLM)	→	Trigger-based recommendation generation	→	Design-time static situations



Challenges and solutions (MMV2-DCL)

4. Life-log Monitoring and Prediction

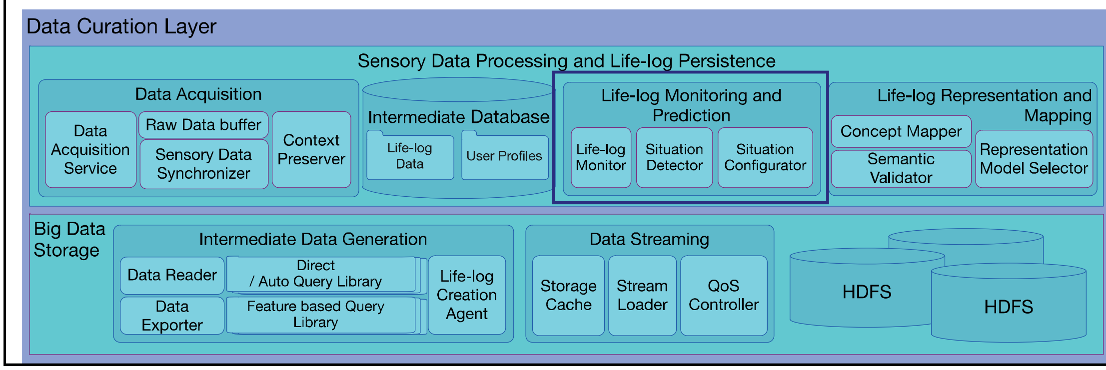
		Benefit		Limitation
V1. NA	→	NA	→	No trigger/situation based recommendations
V1.5 Life-log Monitoring (LLM)	→	Trigger-based recommendation generation	→	Design-time static situations
V2 Life-log Monitoring (LLM)	→	Dynamic situations with runtime-invocation		1. A configuration utility will be created for situation generation by KCL 2. LLM will load the new situation at runtime for invocation



Challenges and solutions (MMV2-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	{	1. A configuration utility will be created for situation generation by KCL 2. LLM will load the new situation at runtime for invocation

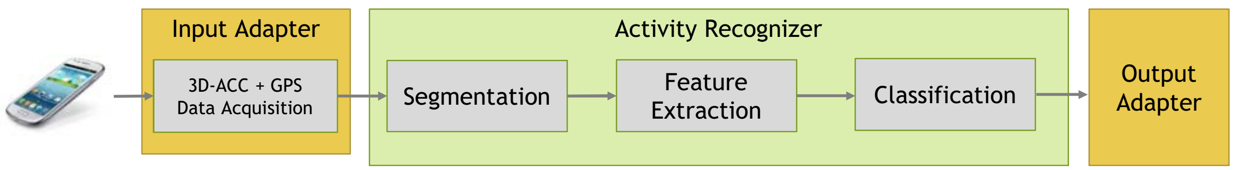


Information Curation Layer (ICL)

Challenges and solutions (MMV2-ICL)

1. Human activity recognition (AR)

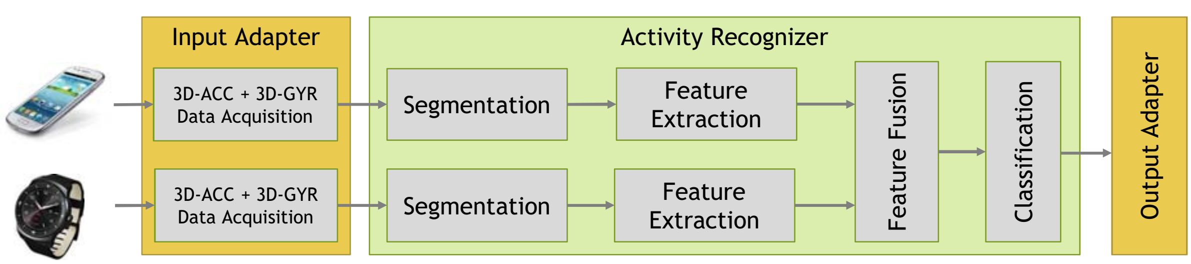
	Benefit	Limitation
V1. Smartphone-based AR	→ High accuracy	→ Only 5 activities



Challenges and solutions (MMV2-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”)

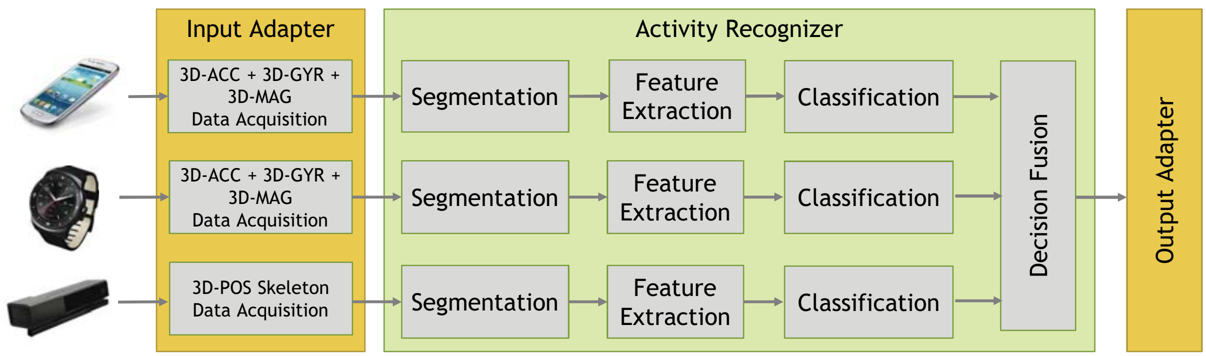


Challenges and solutions (MMV2-ICL)

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)	→ More activities and high accuracy	

Confidence level: Medium

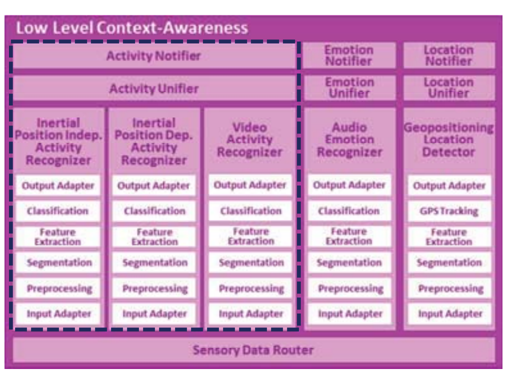


Challenges and solutions (MMV2-ICL)

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)	→ More activities and high accuracy	

Confidence level: Medium

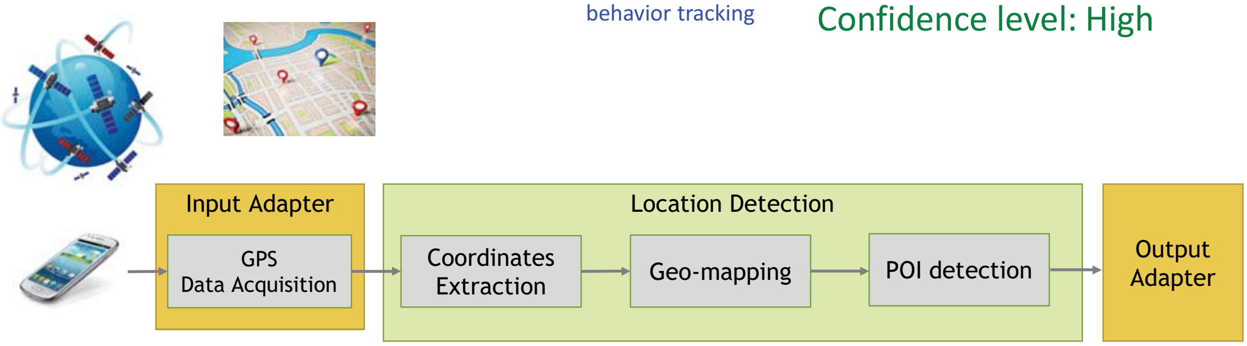


Challenges and solutions (MMV2-ICL)

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

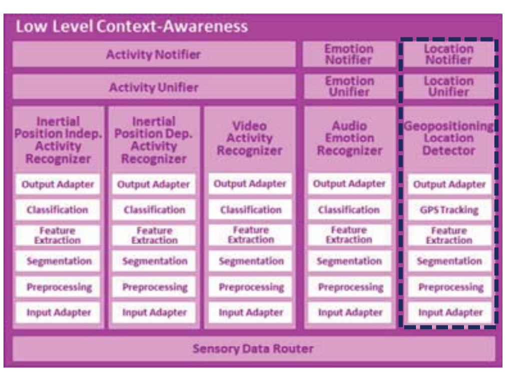


Challenges and solutions (MMV2-ICL)

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

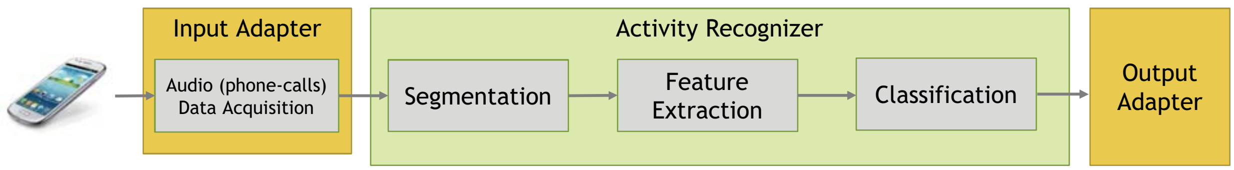


Challenges and solutions (MMV2-ICL)

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	

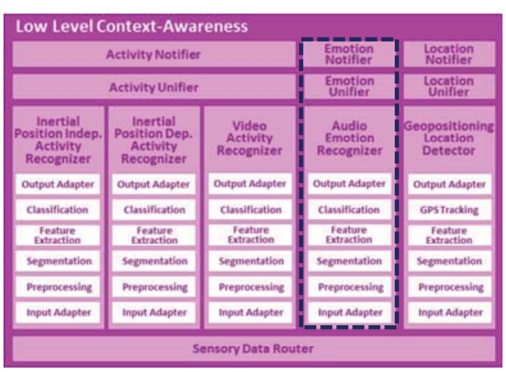
Confidence level: Low-Medium



Challenges and solutions (MMV2-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	



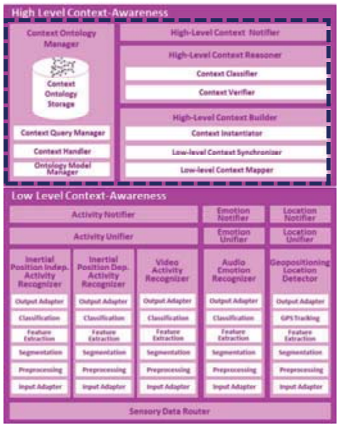
Confidence level: Low-Medium

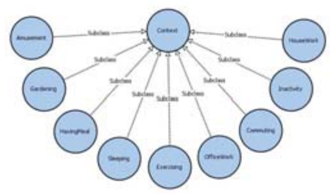
Challenges and solutions (MMV2-ICL)

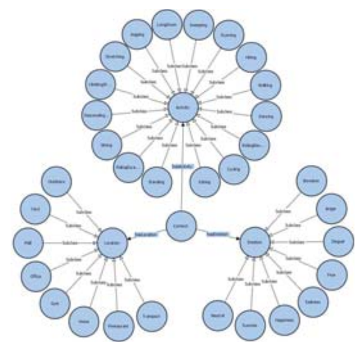
4. High-Level Context Awareness (HLCA)

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

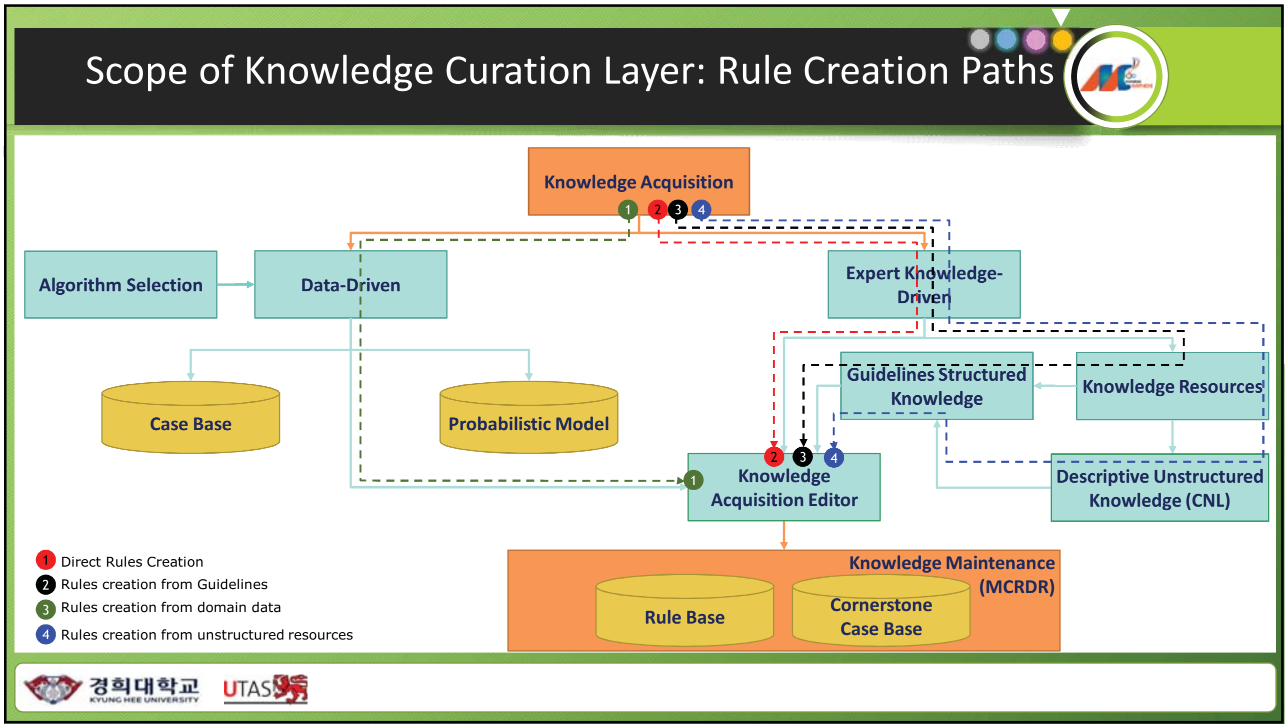
Confidence level: High

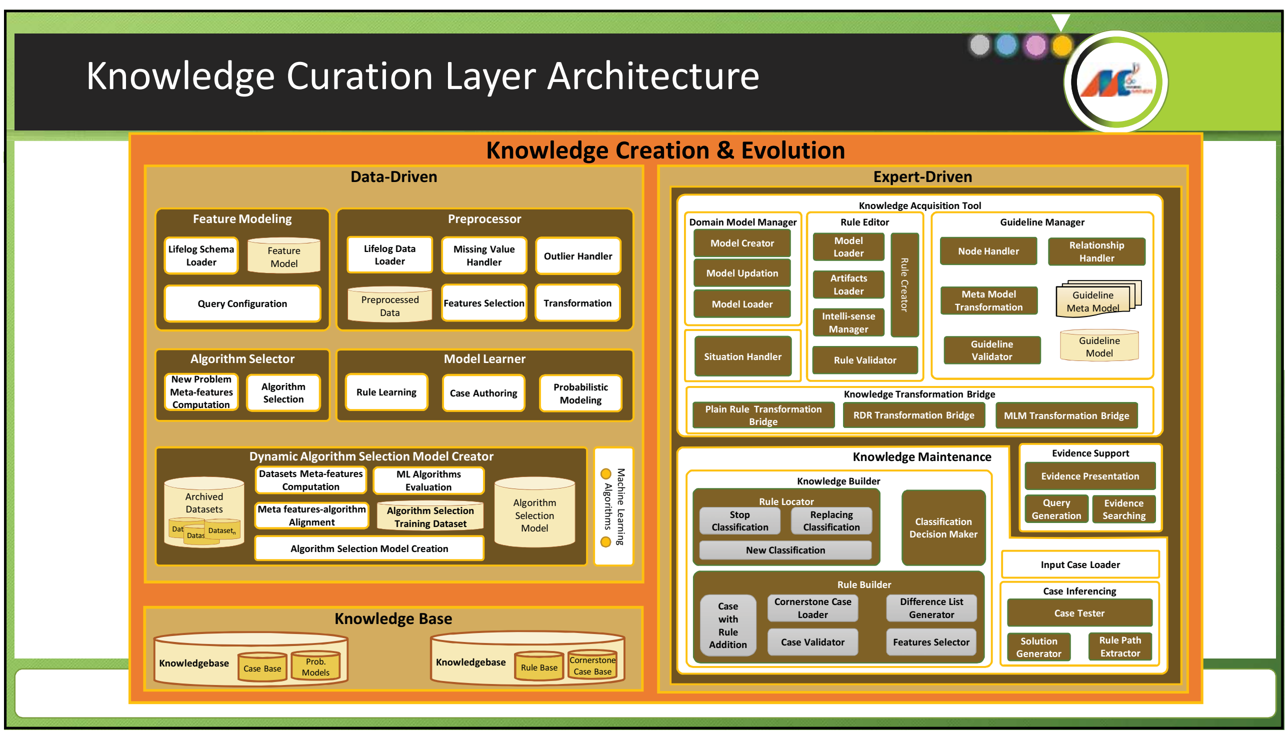






Knowledge Curation Layer (KCL)

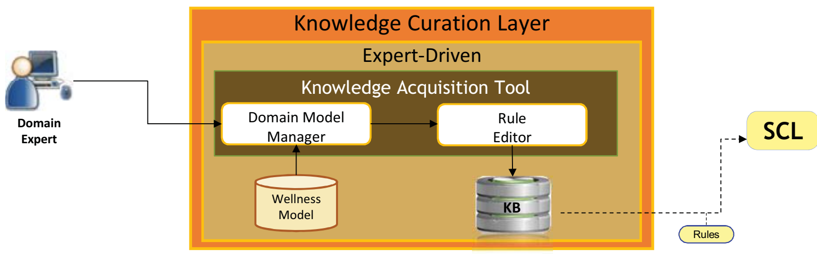

Scope of Knowledge Curation Layer: Rule Creation Paths
Knowledge Acquisition
Algorithm Selection
Data-Driven
Expert Knowledge-Driven
Case Base
Probabilistic Model
Guidelines Structured Knowledge
Knowledge Resources
Knowledge Acquisition Editor
Descriptive Unstructured Knowledge (CNL)
1 Direct Rules Creation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


Knowledge Curation Layer Architecture
Knowledge Creation & Evolution
Data-Driven
Feature Modeling
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
Knowledge Base
Knowledgebase
Case Base
Prob. Models
Knowledgebase
Rule Base
Cornerstone Case Base
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
Situation Handler
Rule Validator
Guideline Manager
Node Handler
Relationship Handler
Meta Model Transformation
Guideline Meta Model
Guideline Validator
Guideline Model
Knowledge Transformation Bridge
Plain Rule Transformation Bridge
RDR Transformation Bridge
MLM Transformation Bridge
Knowledge Maintenance
Knowledge Builder
Rule Locator
Stop Classification
Replacing Classification
New Classification
Classification Decision Maker
Rule Builder
Case with Rule Addition
Cornerstone Case Loader
Difference List Generator
Case Validator
Features Selector
Evidence Support
Evidence Presentation
Query Generation
Evidence Searching
Input Case Loader
Case Inferencing
Case Tester
Solution Generator
Rule Path Extractor

Challenges and solutions (MMV2-KCL)

1. Expert Driven Knowledge Acquisition

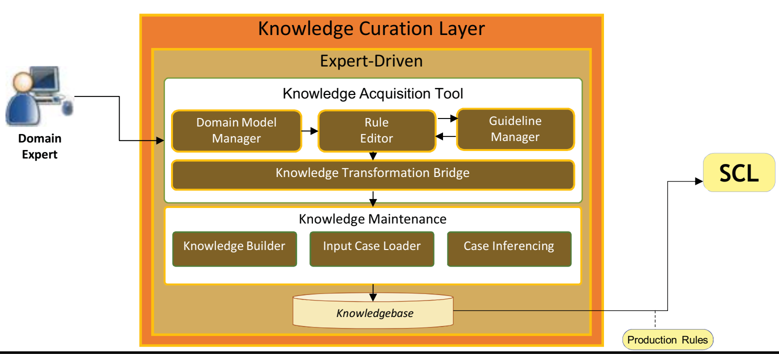
MM Version	Benefits	Limitations
V1. (Not considered)	→ ( )	→ ( )
V1.5: Expert base knowledge (Path-I)	→ Rule base KA	→ • No Validation • No SCL direct Integration



Challenges and solutions (MMV2-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: Guideline Integration(Path-II)	→ • Guideline Template Model (DTs) • RDR base KA and SCL Integration	→ ( )	→ • Only Structured guideline handling

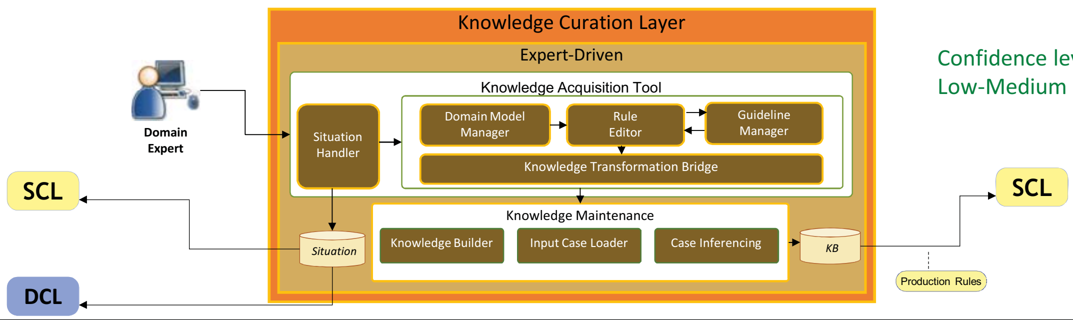


Confidence level: Low-Medium

Challenges and solutions (MMV2-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

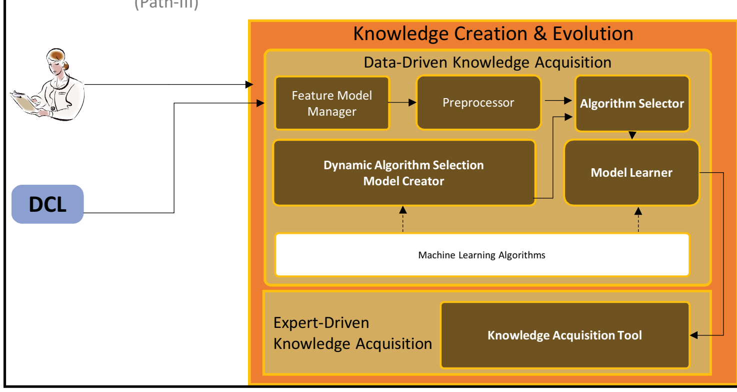


Confidence level:
Low-Medium

Challenges and solutions (MMV2-KCL)

2. Data Driven Knowledge Acquisition

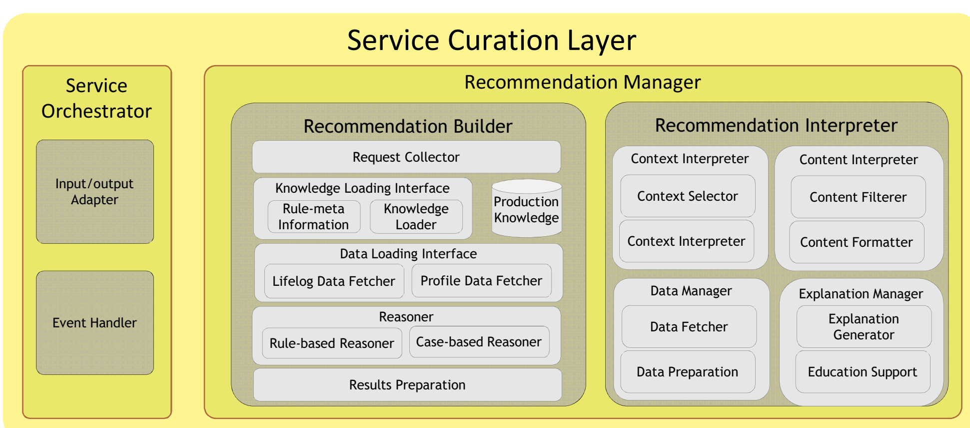
MM Version	Benefits	Limitations	Challenges
V1: (Not considered)	→ ( )	→ ( )	→ ( )
V1.5: (Not considered)	→ ( )	→ ( )	→ ( )
V2.0: Data acquisition based on Mining mind life log data (Path-III)	→ • DTs from lifelog data • Feature Modeling for different services	→ ( )	→ • Overall accuracy depends on available data • Big data handling



Confidence level:
Low-Medium

Service Curation Layer (SCL)

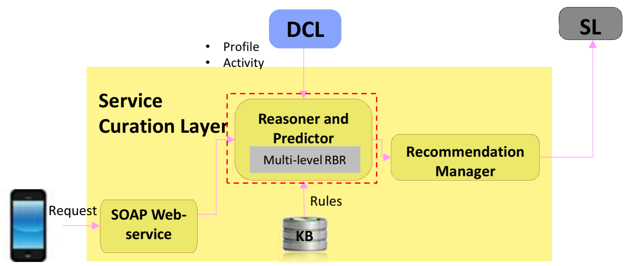
MMV2-SCL Architecture



Challenges and solutions (MMV2-SCL)

1. Reasoning

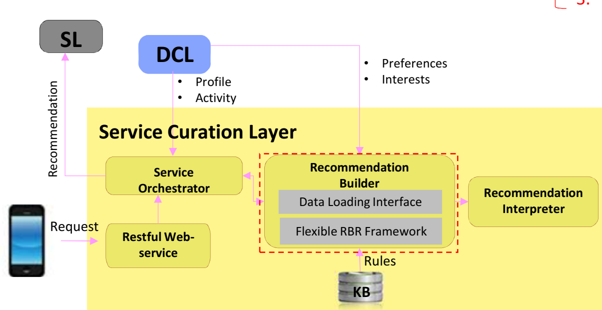
	Benefit	Limitation
V1. Rule-based Reasoning (Reasoner and Predictor) →	1. Granular personalization →	1. Less flexible 2. Service and Data Management 3. No Automatic Rules Retrieval 4. Handcrafted Rules (not scalable)



Challenges and solutions (MMV2-SCL)

1. Reasoning

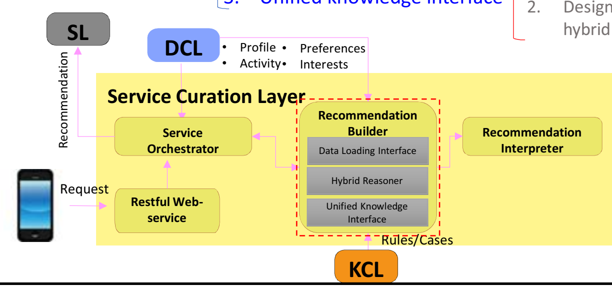
	Benefit	Limitation
V1. Rule-based Reasoning (Reasoner and Predictor) →	1. Granular personalization →	1. Less flexible 2. Service and Data Management 3. Handcrafted Rules (not scalable) 4. No Automatic Rules Retrieval
V1.5. Rule-based Reasoning (Recommendation Builder) →	1. Flexibility 2. Improved data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



Challenges and solutions (MMV2-SCL)

1. Reasoning

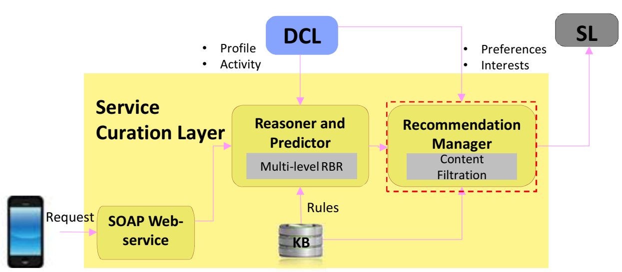
		Benefit		Limitation
V1. Rule-based Reasoning (Reasoner and Predictor)	→	1. Granular personalization	→	1. Less flexible 2. Service and Data Management 3. Handcrafted Rules (not scalable) 4. No Automatic Rules Retrieval
V1.5. Rule-based Reasoning (Recommendation Builder)	→	1. Flexibility 2. Improved data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



Challenges and solutions (MMV2-SCL)

2. Interpretation and Explanation

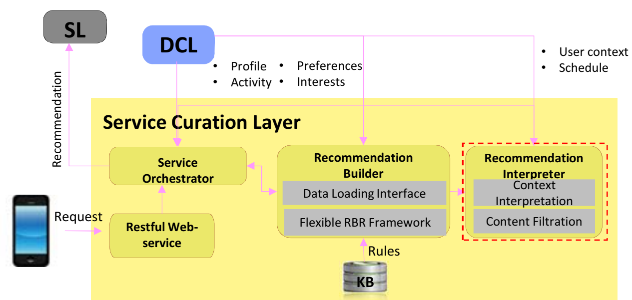
		Benefit		Limitation
V1. Content Filtration (Recommendation Manager)	→	• Preference-based Personalization	→	1. Lack of understandability 2. No context interpretation 3. No explanation



Challenges and solutions (MMV2-SCL)

2. Interpretation and Explanation

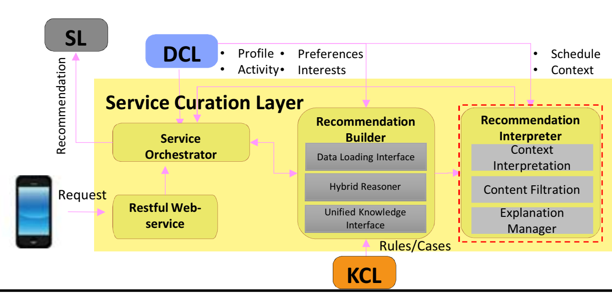
		Benefit		Limitation
V1. Content Filtration (Recommendation Manager)	→	1. Preference-based Personalization	→	1. Lack of understandability 2. No context interpretation 3. No explanation
V1.5. Context Interpretation (Recommendation Interpreter)	→	1. Understandable 2. Contextually (time-based) interpreted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



Challenges and solutions (MMV2-SCL)

2. Interpretation and Explanation

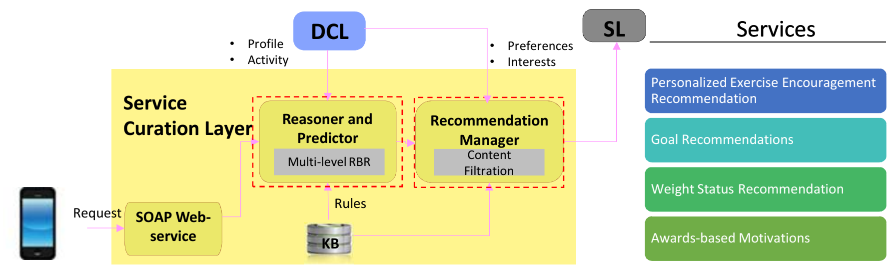
		Benefit		Limitation
V1. Content Filtration (Recommendation Manager)	→	1. Preference-based Personalization	→	1. Lack of understandability 2. No context interpretation 3. No explanation
V1.5. Context Interpretation (Recommendation Interpreter)	→	1. Understandable 2. Contextually (time-based) interpreted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



Challenges and solutions (MMV2-SCL)

3. Services Scope and Types

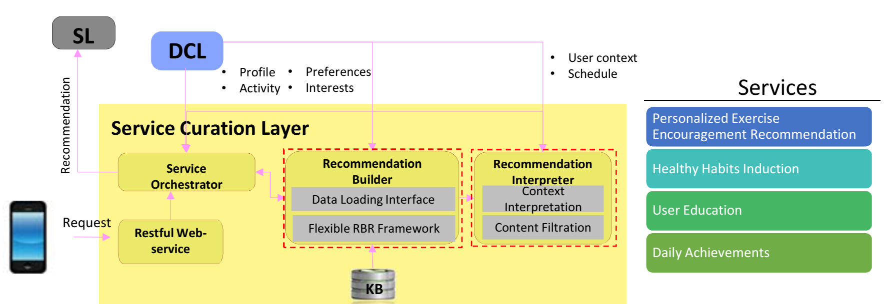
	Benefit	Limitation
V1. Physical Activity Recommendations (Calories Based)	1. User encouragement for phy. activity rec. using goal and award-based services	1. Less practical in terms of user understandability



Challenges and solutions (MMV2-SCL)

3. Services Scope and Types

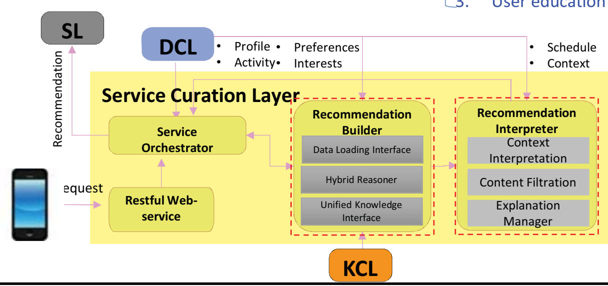
	Benefit	Limitation
V1. Physical Activity Recommendations (Calories Based)	1. User encouragement for phy. activity rec. using goal and award-based services	1. Less practical in terms of user understandability
V1.5 Physical Activity Recommendations and Educational Facts	1. Physical actv. rec 2. Avoiding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



Challenges and solutions (MMV2-SCL)

3. Services Scope and Types

	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 • V2.0 Nutrition Recommendations →	1. Avoiding abnormal sedentary behaviors 2. Food-integrated phy. act. recommendations 3. User education →	Challenges: 1. Increased Service dependencies management 2. Integration of food and physical activity services



Services

- Personalized Exercise Encouragement Recommendation
- Goal Recommendations
- Healthy Habits Induction
- User Education
- Daily Achievements
- Nutrition Services

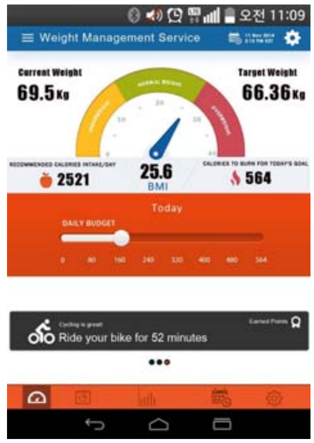
Supporting Layer (SL)

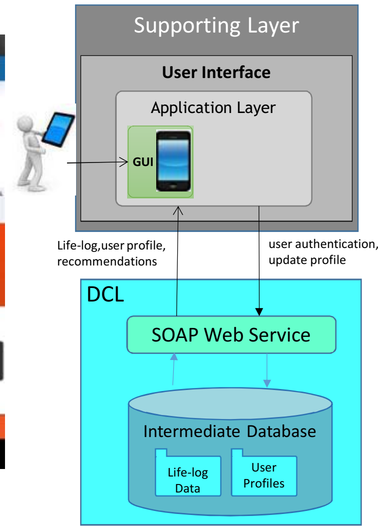
Challenges and solutions (MMV2-SL)

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 users activities of the users and system



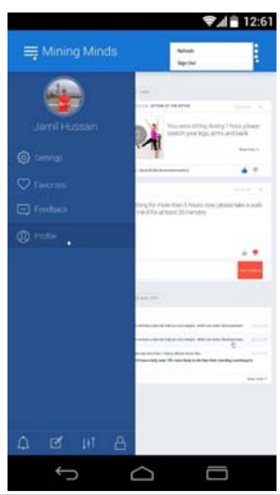


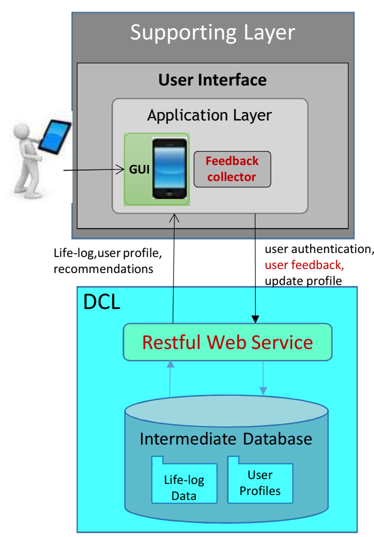
Challenges and solutions (MMV2-SL)

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



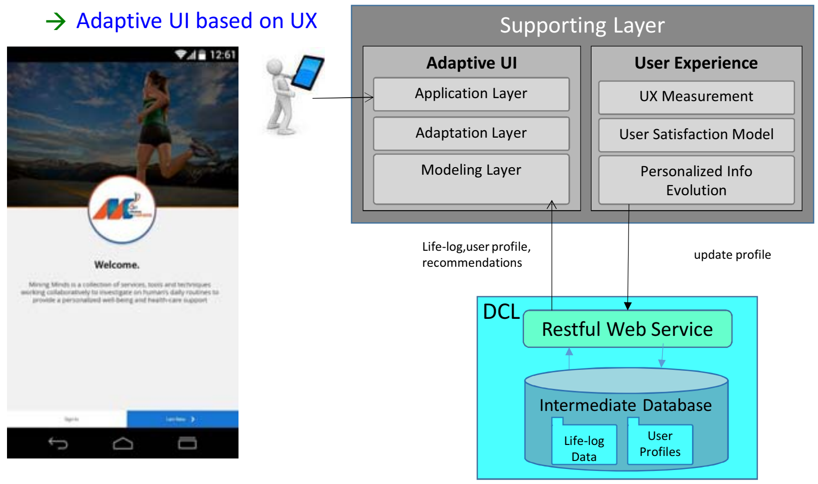


Challenges and solutions (MMV2-DCL)

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	

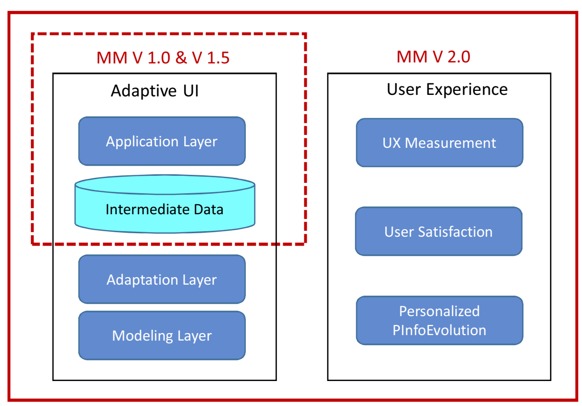
- User interfaces for weight management scenario
- User satisfaction modeling for user experience
- Feedback for UX information collection

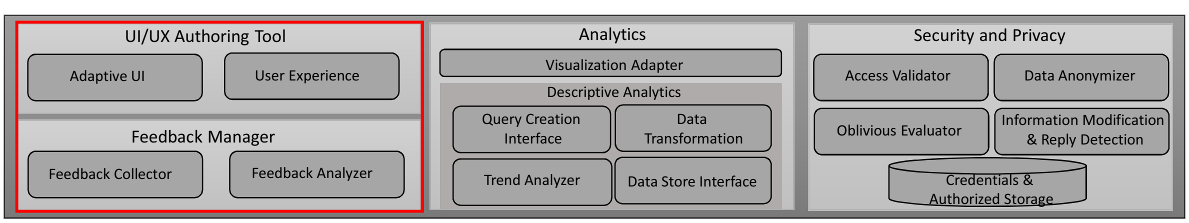


Challenges and solutions (MMV2-SL)

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	



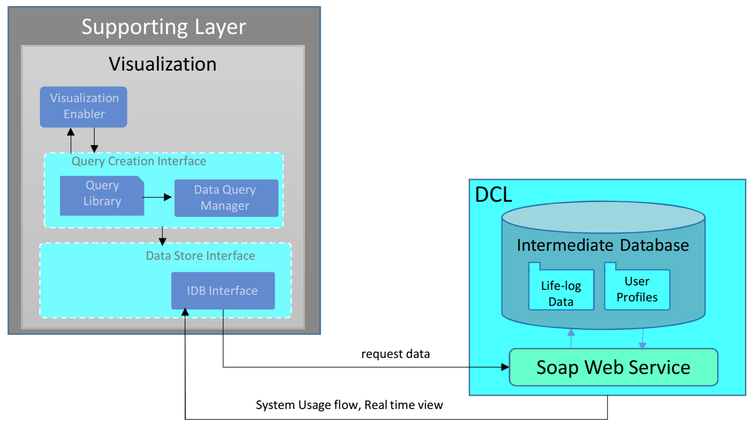


Challenges and solutions (MMV2-SL)

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

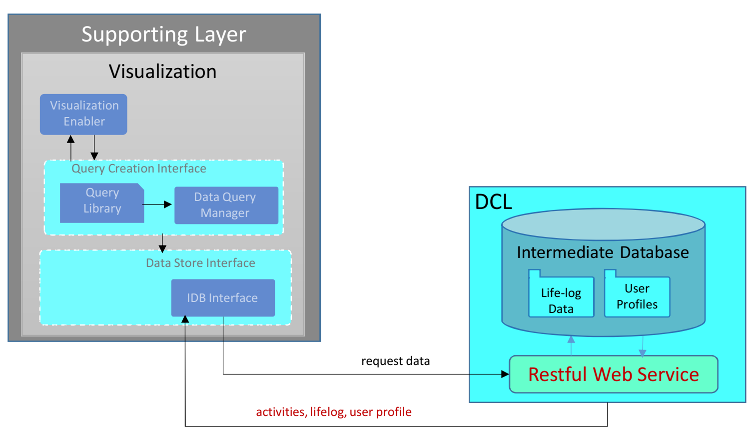


Challenges and solutions (MMV2-SL)

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

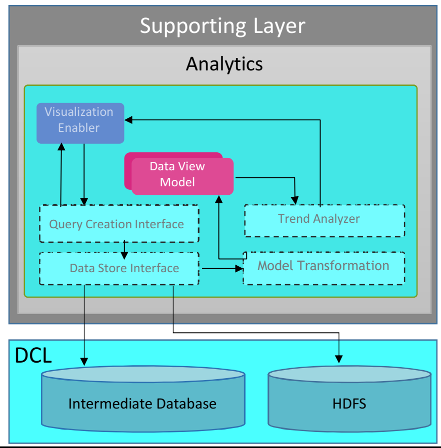


Challenges and solutions (MMV2-SL)

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.	

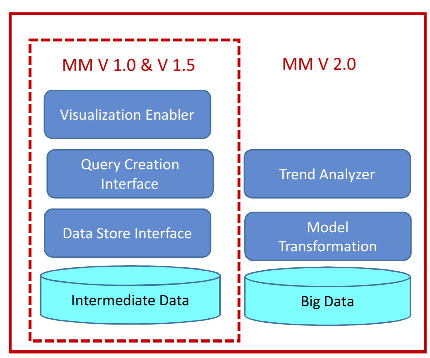
- Analytics based on statistics, clustering and association
- Insights from the big data repository and communication between both repositories
- Query library for the big data repository

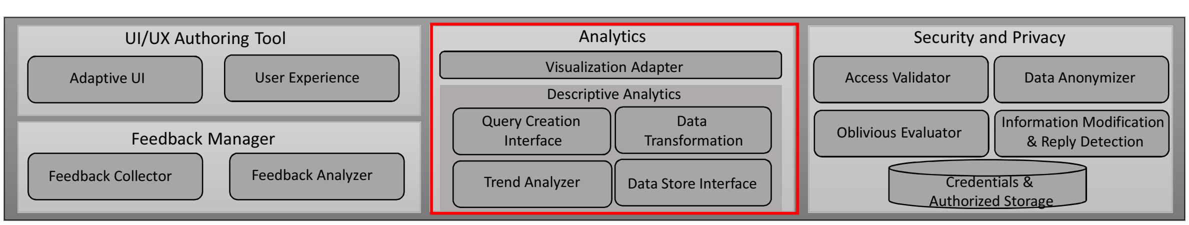


Challenges and solutions (MMV2-SL)

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.	

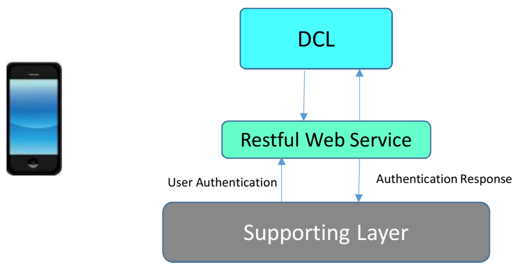




Challenges and solutions (MMV2-SL)

3. Security

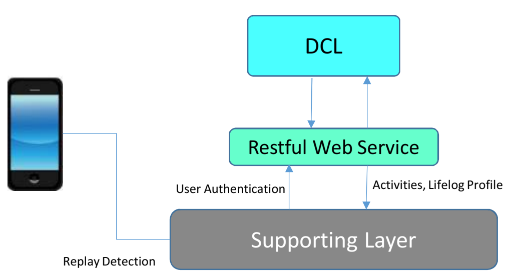
	Benefit	Limitation
V1.5 Expert View Authentication	→ User Authentication based on access layer protocol for expert panel Data encryption and decryption for databases	→ No secure communication between databases and mobile application



Challenges and solutions (MMV2-SL)

3. Security

	Benefit	Limitation
V1.5 Expert View Authentication	→ User Authentication based on access layer protocol for expert panel data encryption and decryption for databases	→ No secure communication between databases and mobile application
V2 Communication between user application and data repositories	→ Replay Detection implementation for secure communication. Data encryption and decryption for databases	



Section 4

Mining Minds Version 2.0
Design and Development Methodology

Section 4-A

Requirements Specifications

Introduction

MMV2 is the 3rd major integration for Mining Minds Platform, constituting upon the duration of 6 months, i.e., (Jun 2015 – Dec 2015). MMV2 utilizes the analysis, design and implementation efforts of previous versions, i.e., MMV1 and 1.5, for its benefit and builds on the top of already identified layered abstractions and primary components. The overall architecture of MMV2 is similar to previous iterations; however, components have been added to incorporate newer requirements.

At Data level, Data Curation Layer of MMV2 (DCL 2), is incorporating not only activity data from smartphone and smart watch, but also acquiring video stream from a camera, making the data input more heterogeneous per user. This data acquisition is purely in real-time with asynchronous, non-blocking communication from data source to the DCL 2. Furthermore, DCL 2 is also providing the read access to sensory data persisted in big data storage for 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.

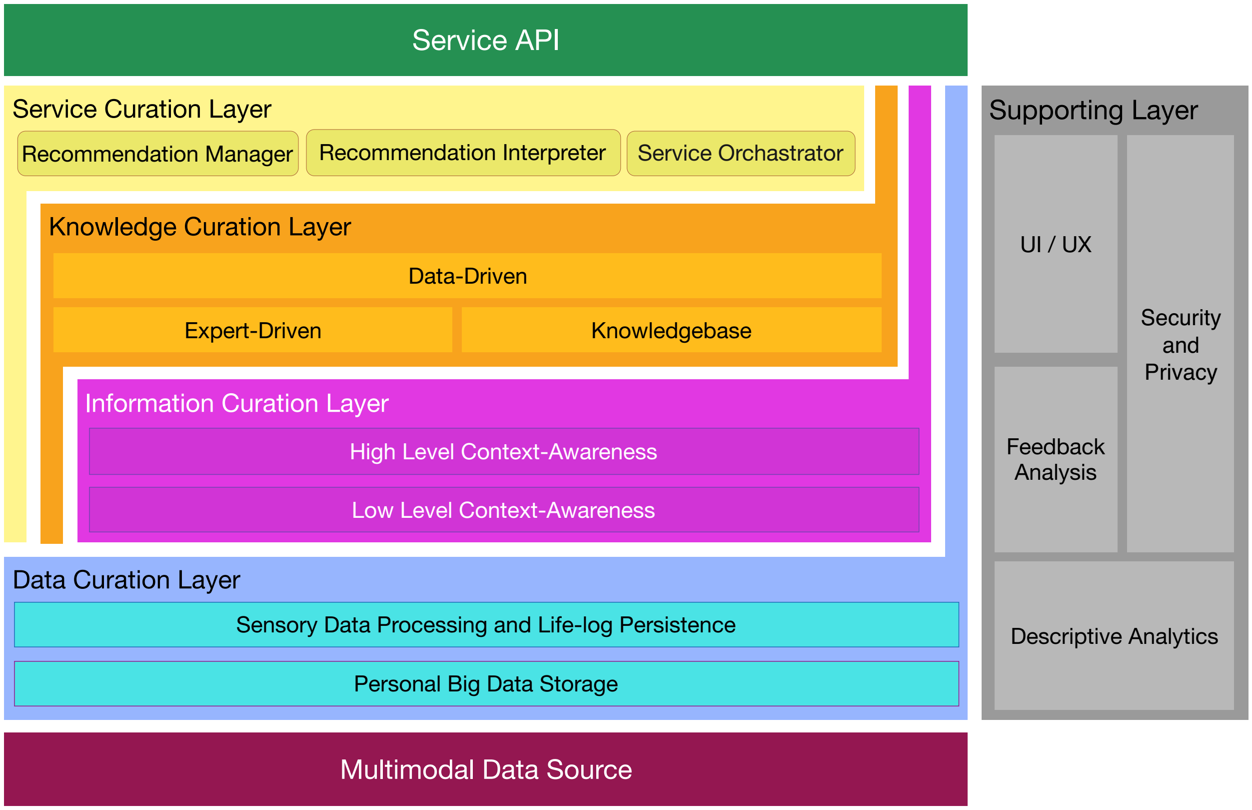
Knowledge curation, introduced in MMV1.5 is updated in MMV2 as Knowledge Curation Layer (KCL 2). For data driven knowledge acquisition, this layer now incorporates training of classification models directly from the data persisted as life-log in the intermediate database and big data storage. For expert driven knowledge acquisition, KCL 2 provides production rules for service curation components and derive situations to be hosted at DCL 2 and serviced at high-level service curation.

At service level, Service Curation Layer (SCL 2) 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 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, and are delivered to the Supporting Layer i.e. SL 2 in order to serve the service requester.

Supporting layer (SL 2) deals with providing services to every other layer of MMV2. It is responsible to provide personalized recommendations, trend analytics through adaptive user interface with added services of privacy and security. SL 2 provides security services for securing the communication among different layers in MMV2. It handles the recommendations from SCL 2 and stores the feedback of the user in DCL 2. Also, for analysis of the data stored as big data, SL 2 provides services of analytics by communicating with DCL 2.

This document provides requirement specifications for MMV2 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-22	The platform shall identify the unresolved user requests and notify the corresponding layer for missing knowledge
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

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)
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

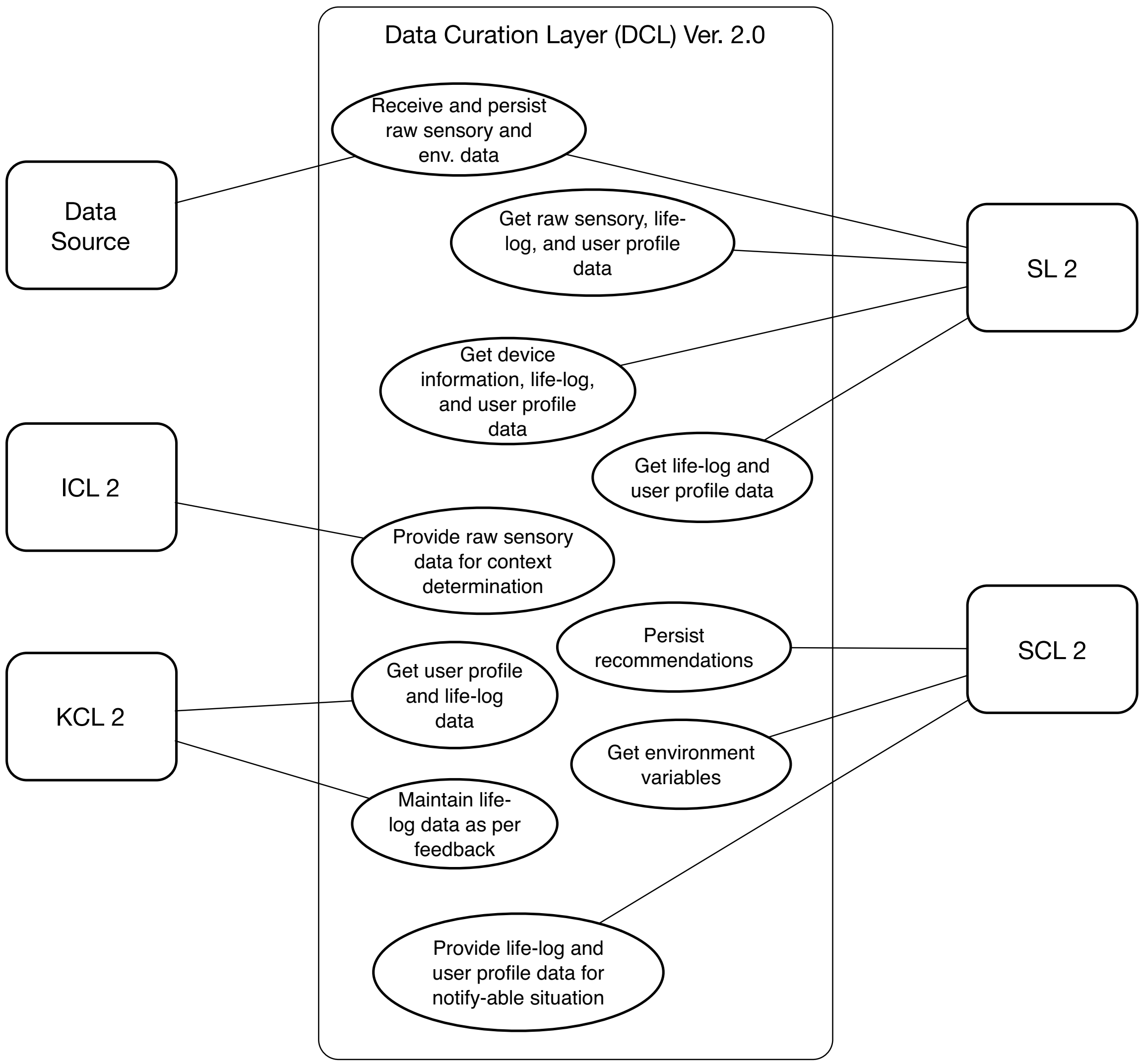
High Level Use cases

Data Curation Layer Ver. 2.0 (DCL 2)

Use case List

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

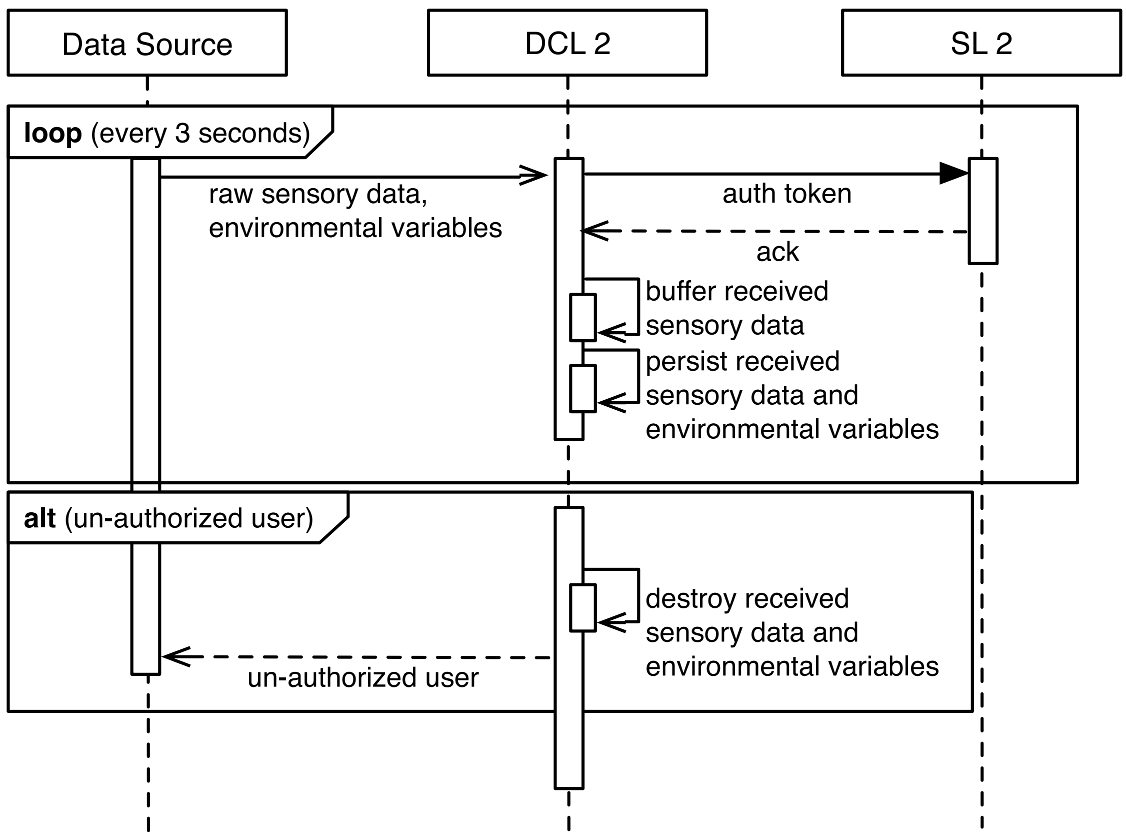
Use case Diagram



Detailed Use case

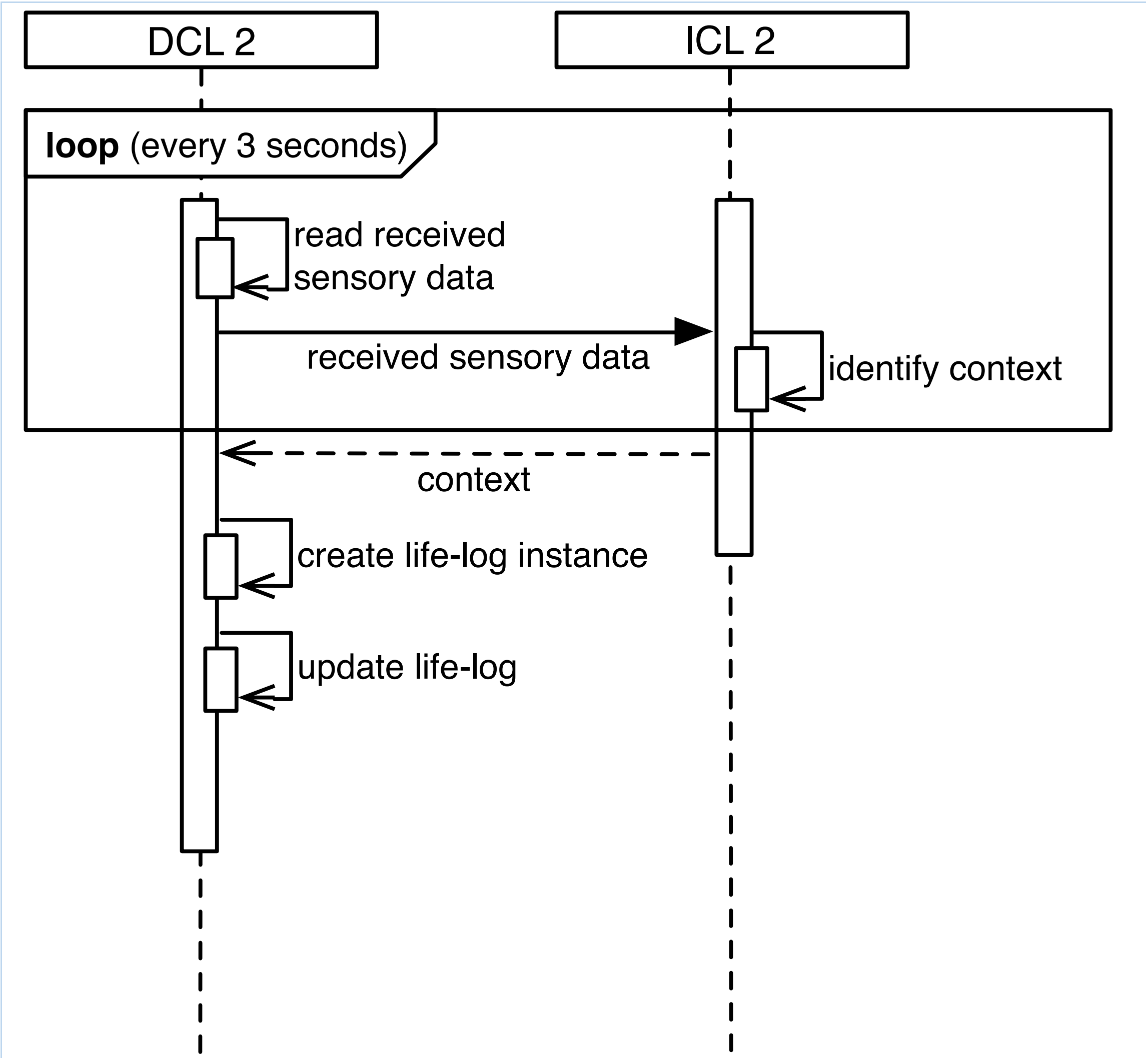
Use Case ID:	*DCL2-UC-01*		
Use Case Name:	Receive and persist raw sensory data and environmental variables from the data source		
FR ID:	MM-FR-01, MM-FR-02, MM-FR-04		
Created By:	Bilal Amin	**Last Updated By:**	Bilal Amin
Date Created:	06 July 2015	**Last Revision Date:**	10 July 2015
Actors:	Data Source, SL 2		
Description:	DCL 2 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 2. Raw sensory data with environmental variables is persisted in a non-volatile storage by the DCL 2		
Normal Flow:	1. DCL 2 receives the accumulated raw sensory data with environmental variables (e.g., temperature, weather, etc.) sent from the data source 2. DCL 2 authenticates the sensory data source 3. DCL 2 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 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 is secure
Notes and Issues:	NA
Sequence Diagram:	



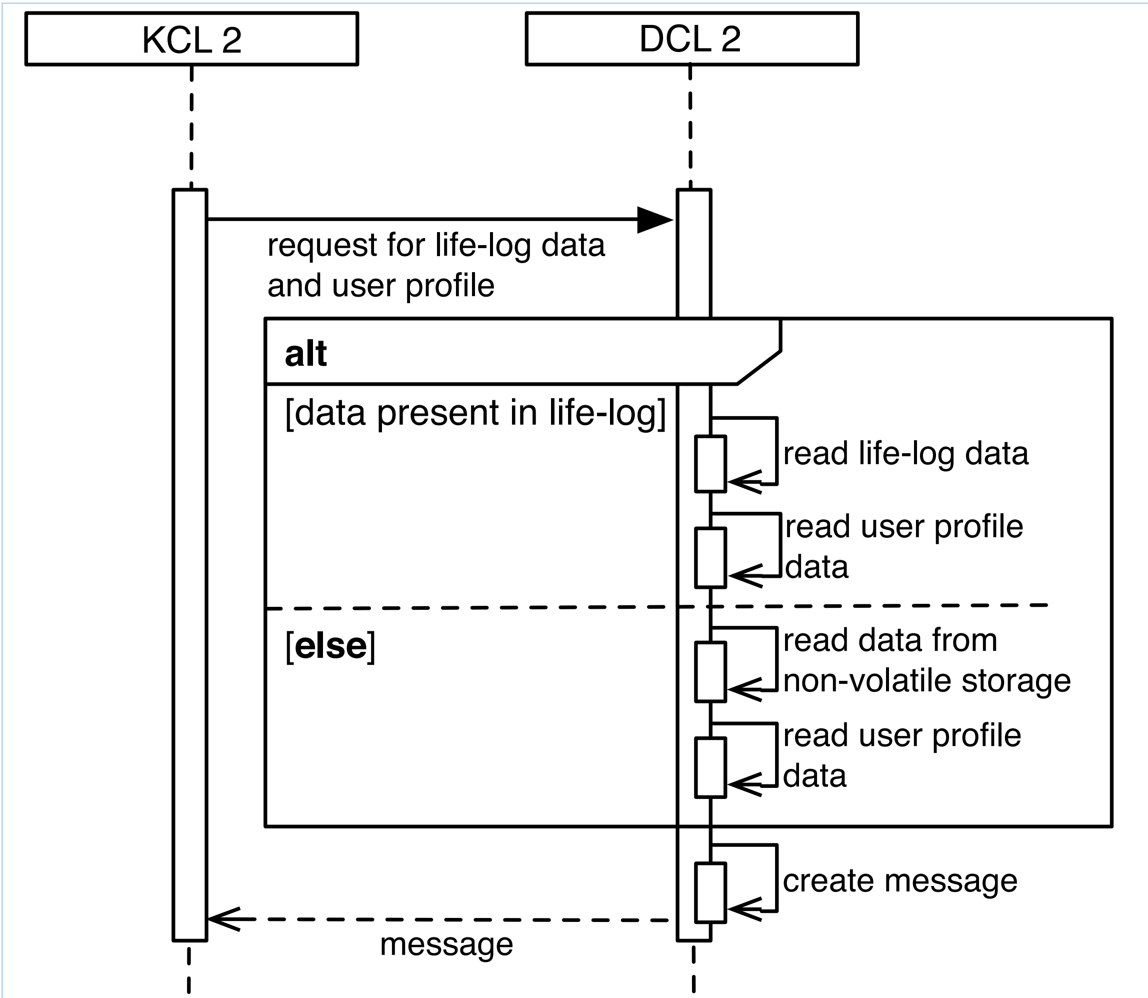
Use Case ID:	*DCL2-UC-02*		
Use Case Name:	Provide raw sensory data for context determination		
FR ID:	MM-FR-03, MM-FR-04		
Created By:	Bilal Amin	**Last Updated By:**	Bilal Amin
Date Created:	06 July 2015	**Last Revision Date:**	10 July 2015
Actors:	DCL 2, ICL 2		

Description:	Received sensory data from data source is communicated with ICL 2 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
Post-conditions:	1. Context have been received by DCL 2 2. Life-log has been updated
Normal Flow:	1. DCL 2 reads the buffered raw sensory data and sends it to ICL 2 2. ICL 2 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:	



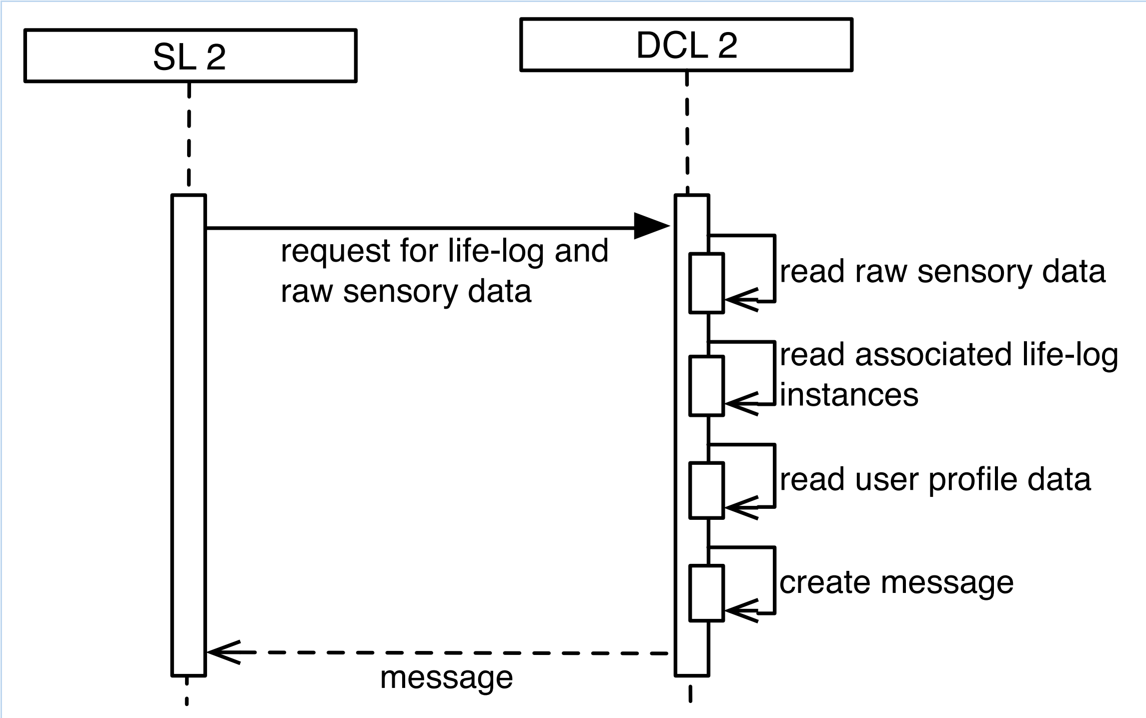
Use Case ID:	*DCL2-UC-03*		
Use Case Name:	Get user profile and life-log data (offline process)		
FR ID:	MM-FR-04, MM-FR-05		
Created By:	Bilal Amin	**Last Updated By:**	Bilal Amin
Date Created:	06 July 2015	**Last Revision Date:**	10 July 2015
Actors:	KCL 2		
Description:	DCL 2 gets user profile and life-log data for the learning of classification models in KCL 2		
Trigger:	Data request from KCL 2 (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
Normal Flow:	1. DCL 2 receives request for user profile and life-log data from KCL 2. DCL 2 reads life-log data from the storage depending upon the attributes provided by KCL 2 3. DCL 2 creates data message 4. Message is sent to KCL 2 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 and KCL is defined
Notes and Issues:	NA
Sequence Diagram:	



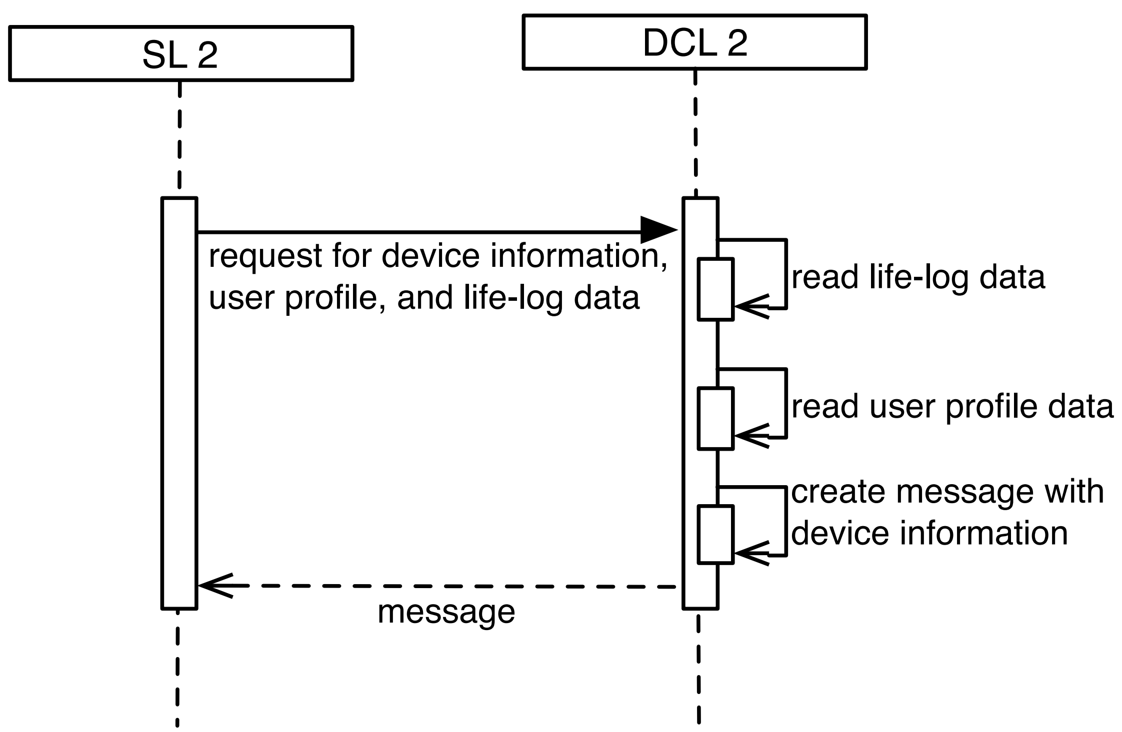
Use Case ID:	*DCL2-UC-04*		
Use Case Name:	Get raw sensory, life-log, and user profile data (online process)		
FR ID:	MM-FR-03, MM-FR-04, MM-FR-05		
Created By:	Bilal Amin	**Last Updated By:**	Bilal Amin
Date Created:	06 July 2015	**Last Revision Date:**	10 July 2015
Actors:	SL 2		
Description:	DCL 2 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
Normal Flow:	1. DCL 2 receives request for Life-log instances, user profile, and persisted raw sensory data from SL 2 2. DCL 2 reads data from non-volatile storage depending upon the attributes provided by SL 2 3. DCL 2 reads associated user life-log instances 4. DCL 2 reads user profile data 5. DCL 2 creates a message containing raw sensory data, Life-log instances, and user profile 6. Message is sent to SL 2 as a response
Alternative Flows:	NA
Exceptions:	NA
Includes:	NA
Frequency of Use:	Frequent: request by SL 2
NFR ID:	MM-NFR-04
Assumptions:	Service contract between DCL 2 and SL 2 is defined
Notes and Issues:	NA
Sequence Diagram:	



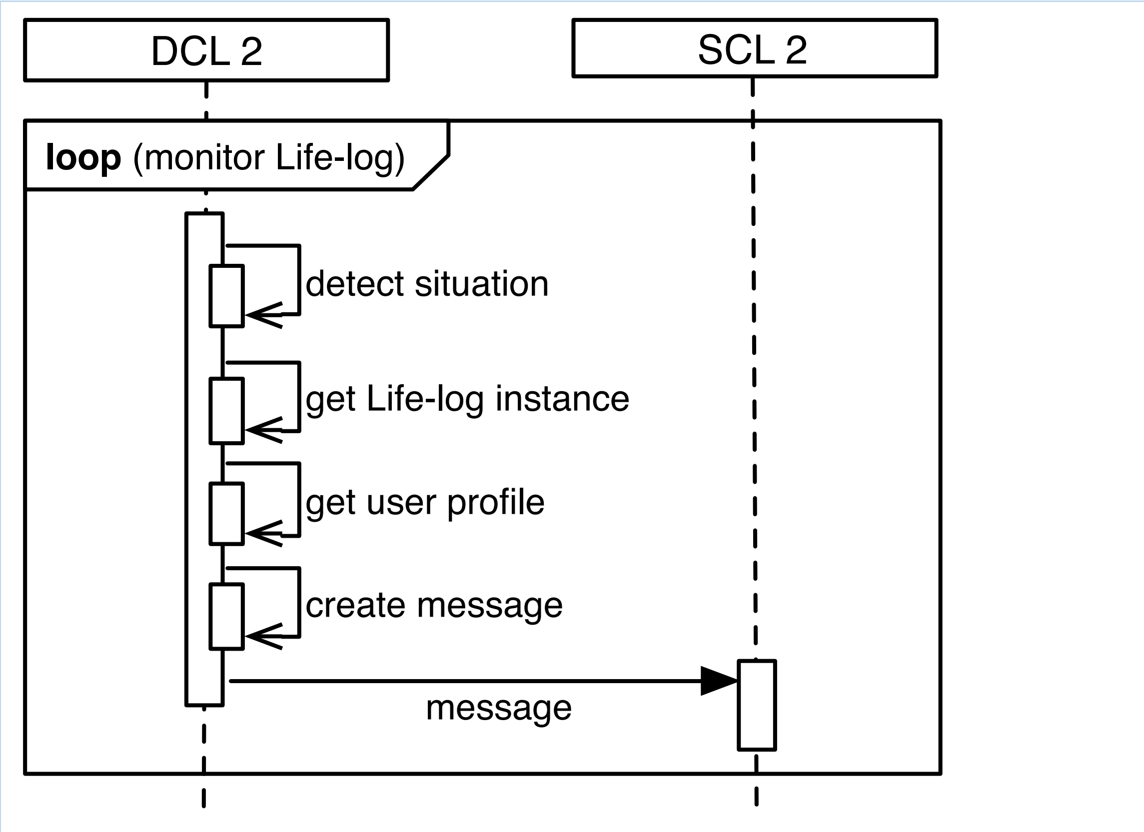
Use Case ID:	*DCL2-UC-05*		
Use Case Name:	Get device information, life-log and user profile data		
FR ID:	MM-FR-04, MM-FR-05		
Created By:	Bilal Amin	**Last Updated By:**	Bilal Amin
Date Created:	06 July 2015	**Last Revision Date:**	10 July 2015
Actors:	SL 2		
Description:	DCL 2 provides device information,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

	log data with user profile and device information 5. Message is sent to SCL 2
Alternative Flows:	NA
Exceptions:	NA
Includes:	NA
Frequency of Use:	Frequent: request by SL 2
NFR ID:	MM-NFR-04
Assumptions:	Service contract between DCL 2 and SL 2 is defined
Notes and Issues:	NA
Sequence Diagram:	



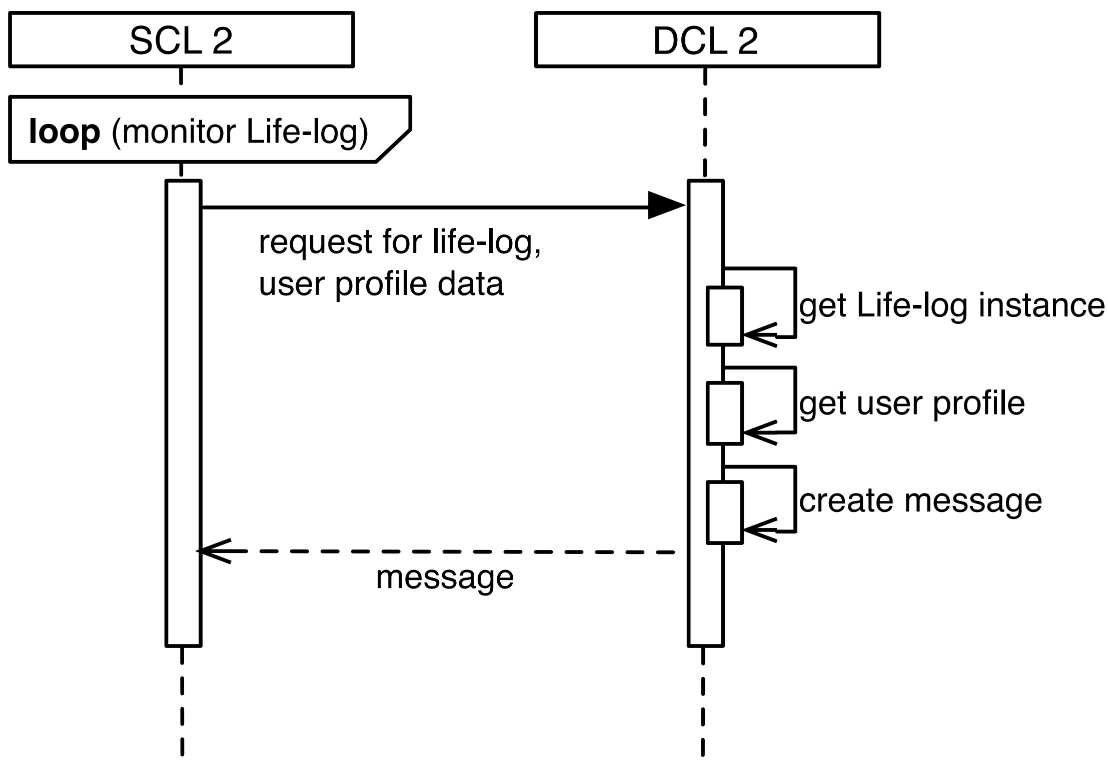
Use Case ID:	*DCL2-UC-06*		
Use Case Name:	Provide life-log and user profile data (trigger) for notify-able situation		
FR ID:	MM-FR-04, MM-FR-05, MM-FR-08		
Created By:	Bilal Amin	**Last Updated By:**	Bilal Amin
Date Created:	06 July 2015	**Last Revision Date:**	10 July 2015

Actors:	DCL 2, SCL 2
Description:	DCL 2 detects a situation over life-log data and triggers SCL 2 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
Normal Flow:	1. DCL 2 performs continuous monitoring of life-log data 2. DCL 2 detects situation over life-log data 3. DCL 2 creates message by accumulating related life-log data and user profile 4. Message is sent to SCL 2 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 and SCL 2 is defined
Notes and Issues:	NA
Sequence Diagram:	



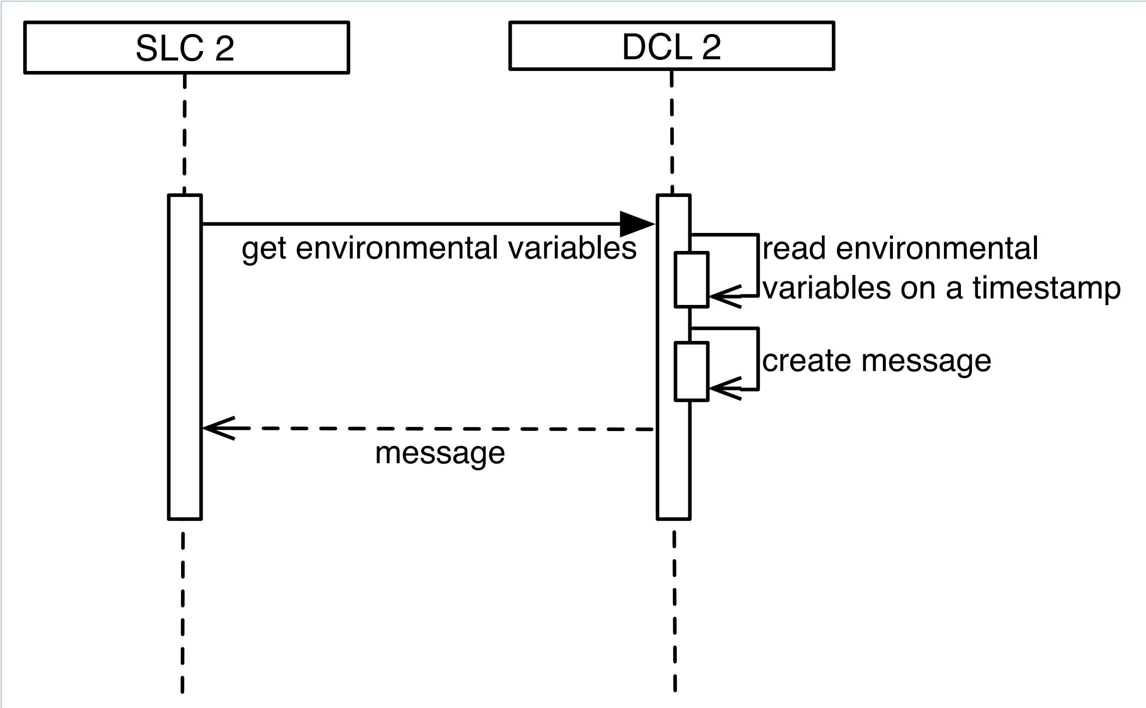
Use Case ID:	*DCL2-UC-07*		
Use Case Name:	Get life-log and user profile data		
FR ID:	MM-FR-04, MM-FR-05		
Created By:	Bilal Amin	**Last Updated By:**	Bilal Amin
Date Created:	06 July 2015	**Last Revision Date:**	10 July 2015
Actors:	SCL 2		
Description:	DCL 2 provides life-log and user profile data to SCL 2 for recommendation generation		
Trigger:	Request for recommendation generation by SCL 2		
Pre-conditions:	Context data is persisted as Life-log instances		
Post-conditions:	Required life-log instance and user profile is sent to SCL		
Normal Flow:	1. DCL 2 receives request for life-log and user profile data from SCL 2 2. DCL 2 creates message by accumulating		

	related life-log data and user profile 3. Message is sent to SCL 2 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 and SCL 2 is defined
Notes and Issues:	NA
Sequence Diagram:	



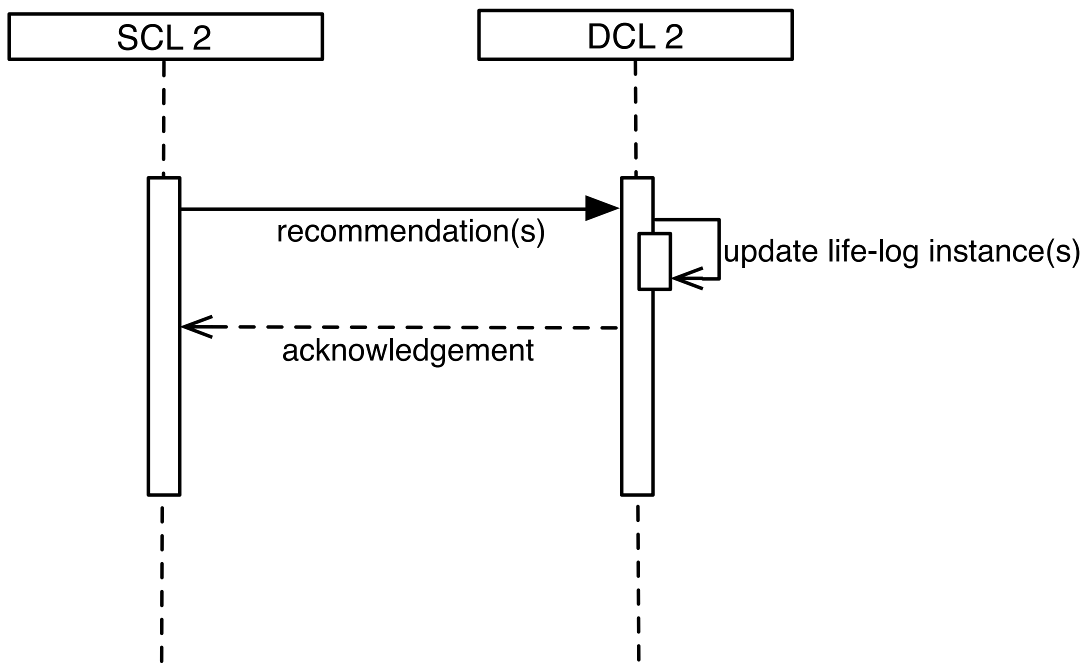
Use Case ID:	*DCL2-UC-08*		
Use Case Name:	Get environmental variables data of the user		
FR ID:	MM-FR-04		
Created By:	Bilal Amin	**Last Updated By:**	Bilal Amin

Date Created:	06 July 2015	**Last Revision Date:**	10 July 2015
Actors:	SCL 2		
Description:	DCL 2 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		
Normal Flow:	1. DCL 2 receives request from SCL 2 for environmental variables 2. DCL 2 retrieves environmental variables based on timestamp from user profile 3. DCL 2 sends the environmental variables as a message to SCL 2		
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 and SCL 2 is defined		
Notes and Issues:	NA		
Sequence Diagram:			



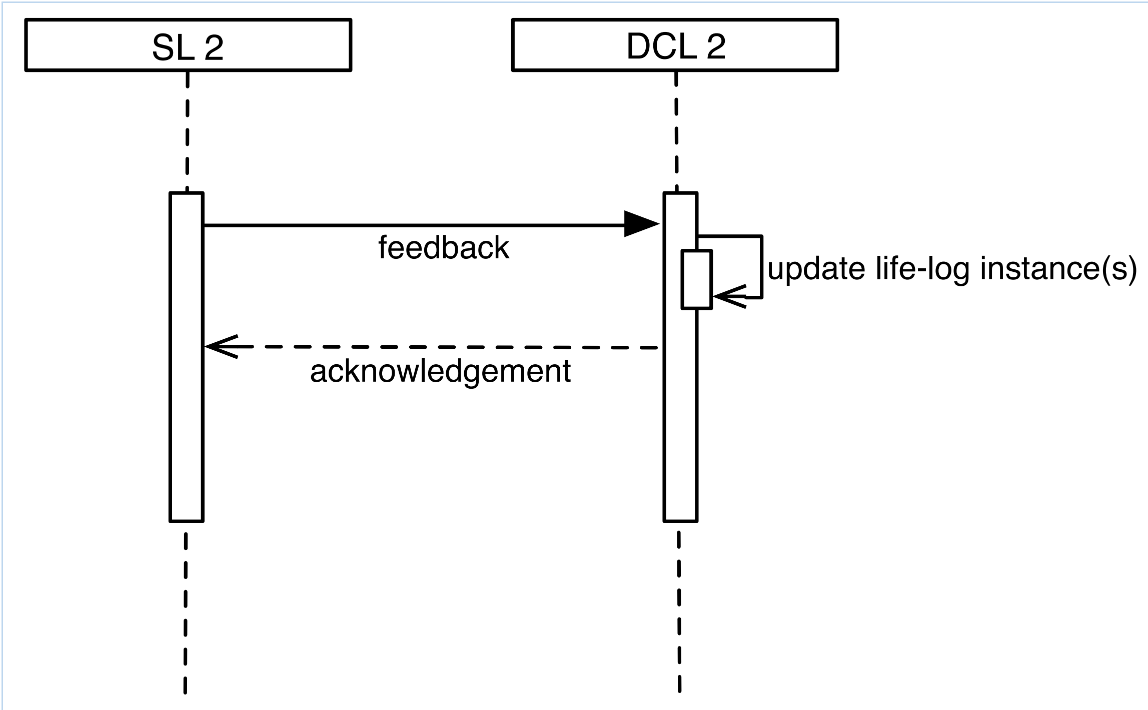
Use Case ID:	*DCL2-UC-09*		
Use Case Name:	Persist the recommendation in life-log		
FR ID:	MM-FR-05		
Created By:	Bilal Amin	**Last Updated By:**	Bilal Amin
Date Created:	06 July 2015	**Last Revision Date:**	10 July 2015
Actors:	SCL 2		
Description:	DCL 2 updates the life-log instance with the recommendations 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 receives generated recommendations from the SCL 2 2. DCL 2 updates the associated life-log instance(s) 3. DCL 2 acknowledges the update to SLC 2		
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 and SCL 2 is defined
Notes and Issues:	NA
Sequence Diagram:	



Use Case ID:	*DCL2-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 July 2015
Actors:	SL 2		
Description:	DCL 2 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 receives feedback from the SL 2 2. DCL 2 updates the associated life-log instance(s) 3. DCL 2 acknowledges the update to SL 2
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 and SL 2 is defined
Notes and Issues:	NA
Sequence Diagram:	

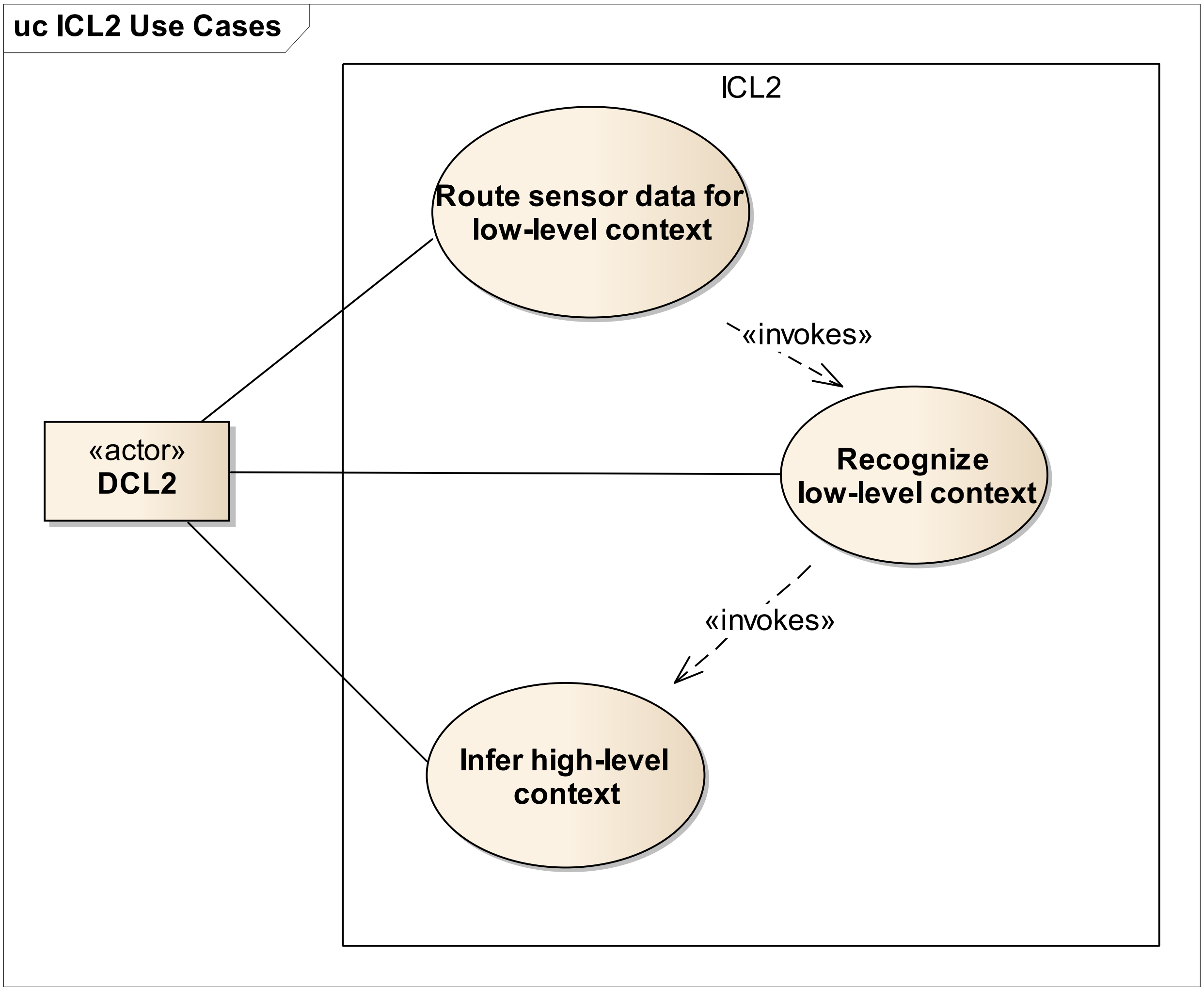

SL 2
DCL 2
feedback
update life-log instance(s)
acknowledgement

Information Curation Layer Ver. 2.0 (ICL 2)

Use case List

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

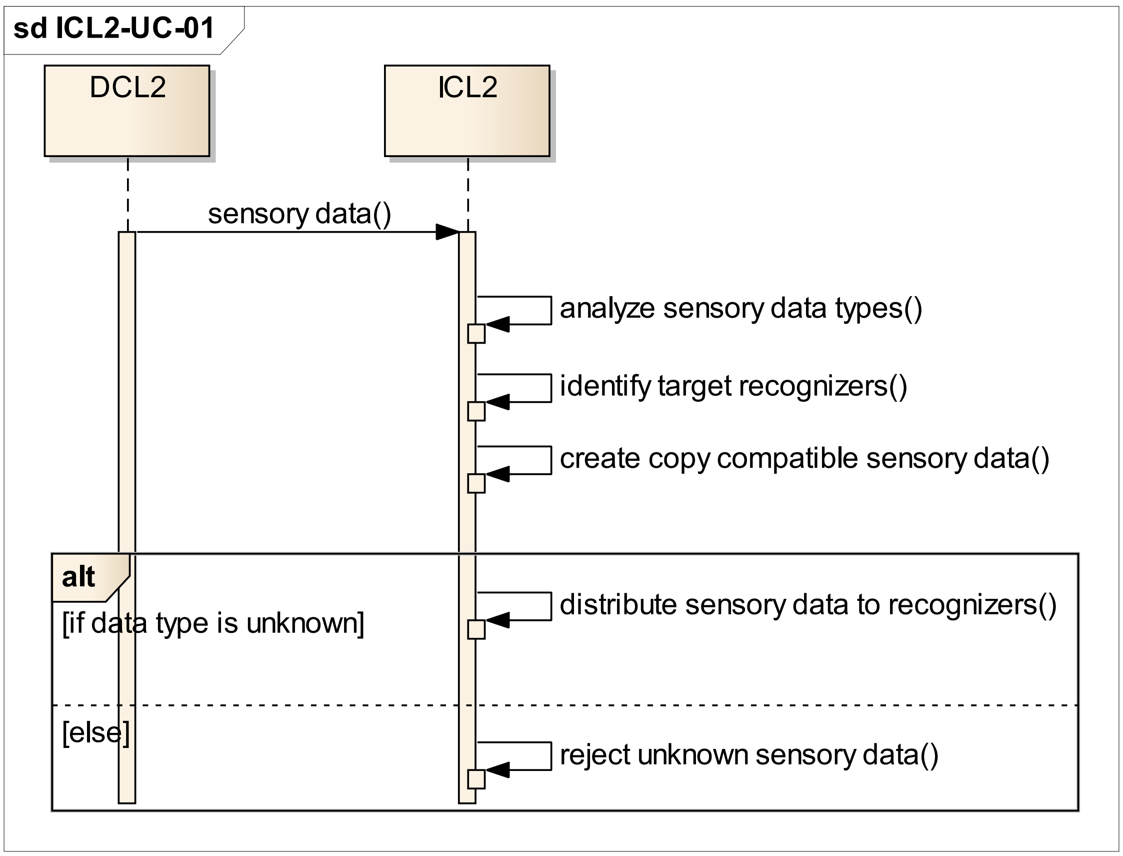
Use case Diagram



Detailed Use case

Use Case ID:	*ICL2-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:**	Oresti Banos
Date Created:	06 July 2015	**Last Revision Date:**	10 July 2015
Actors:	DCL 2		
Description:	Sensory data is received from DCL2 and it is distributed to the corresponding low-level context recognizer based on the data type(s).		
Trigger:	Receive sensory data send by DCL2 to ICL2		
Pre-conditions:	DCL2 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		

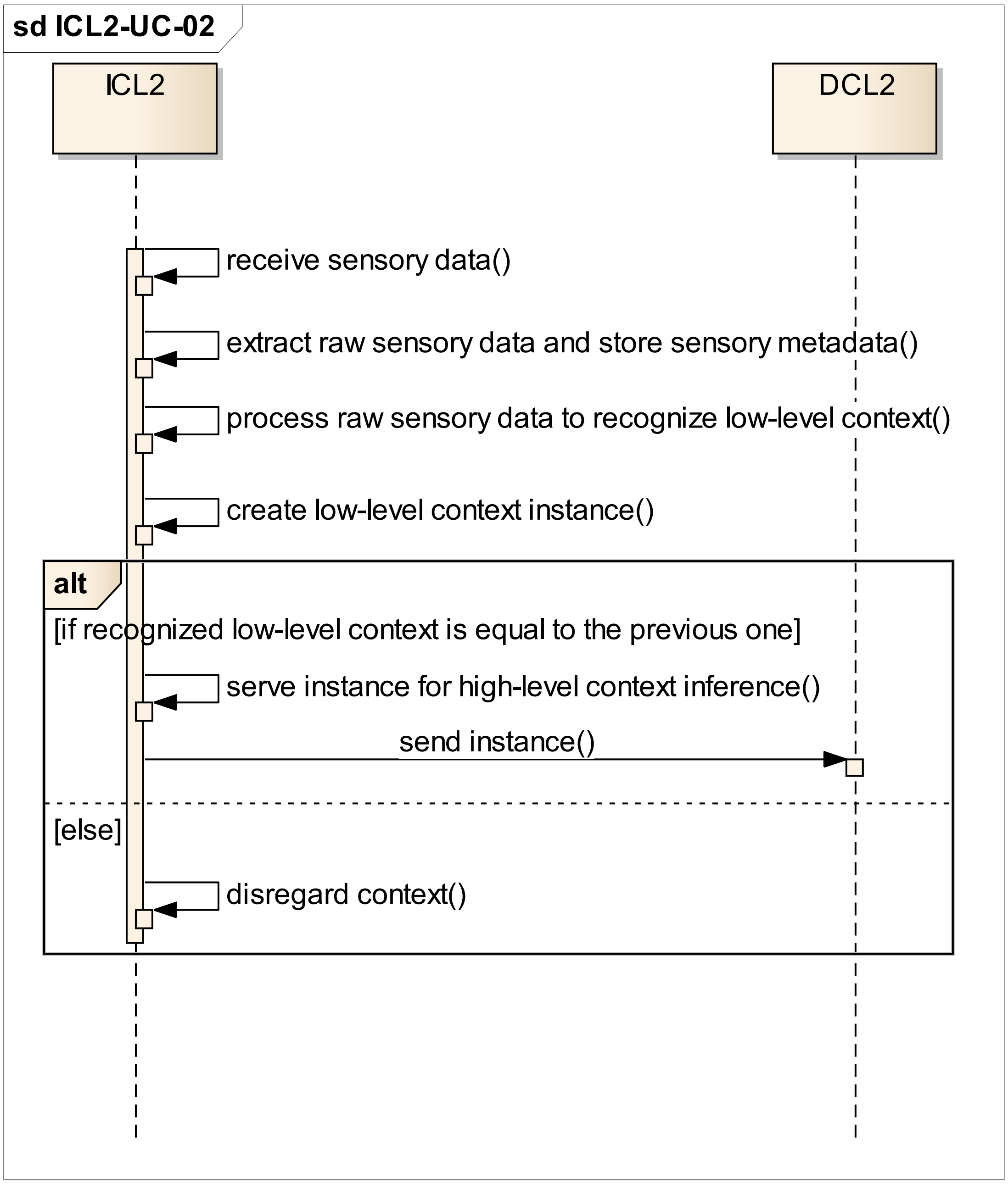
NFR ID:	NA
Assumptions:	1. There is an established communication between DCL 2 and ICL 2 2. The communication channel between the DCL 2 and the ICL 2 is secure 3. Incoming sensory data is already preprocessed (i.e., without missing samples and with synchronized streams)
Notes and Issues:	NA
Sequence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



Use Case ID:	*ICL2-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:**	Oresti Banos

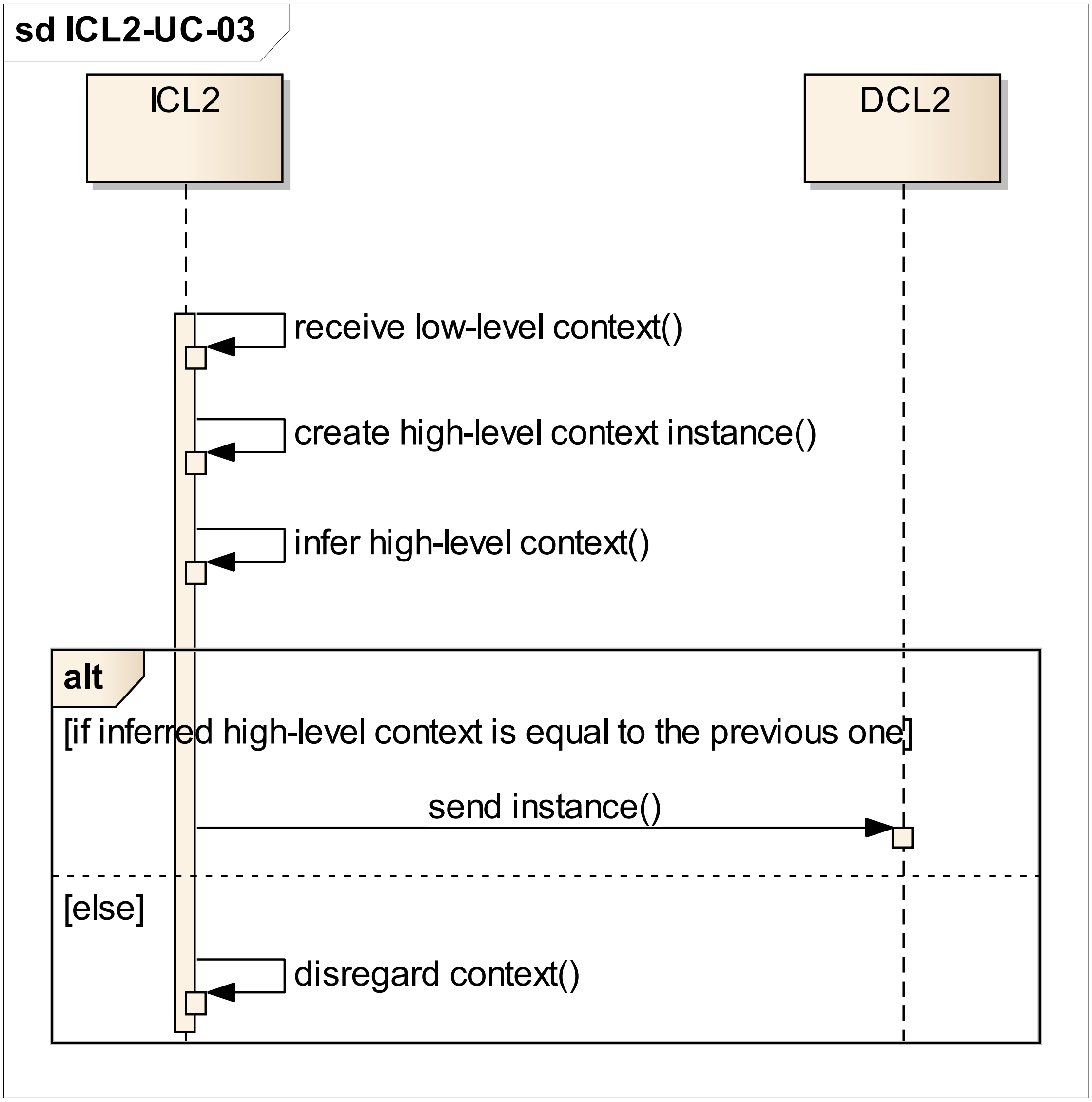
Date Created:	06 July 2015	**Last Revision Date:**	10 July 2015
Actors:	ICL 2, DCL 2		
Description:	Low-level contexts are recognized for a given user based on the received sensory data. The identified contexts are served for the high-level context inference and also communicated to DCL 2 for the generation 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		
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 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-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:**	Oresti Banos
Date Created:	06 July 2015	**Last Revision Date:**	14 July 2015
Actors:	ICL 2, DCL 2		
Description:	High-level contexts are recognized for a given user based on the identified low-level contexts. High-level context(s) are communicated to DCL2 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		
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 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:

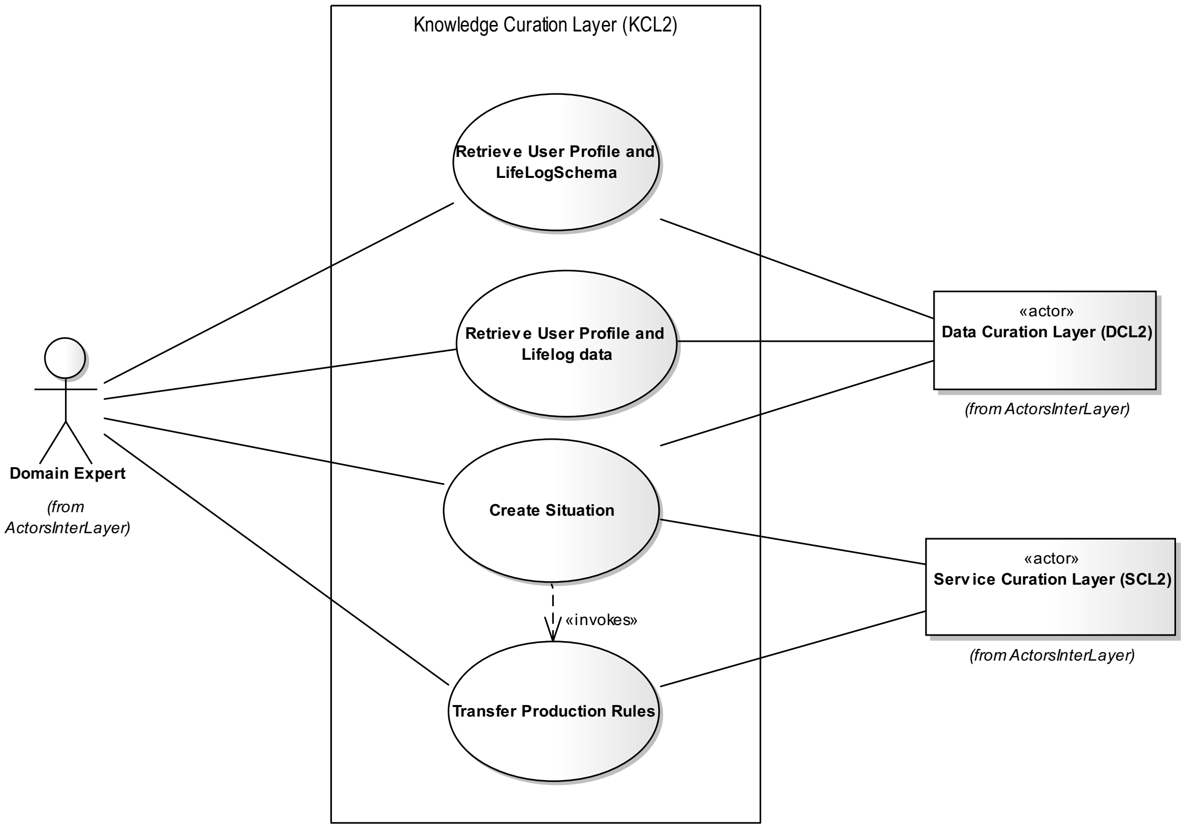


Knowledge Curation Layer Ver. 2.0 (KCL 2)

Use case List

Use case ID#	Name
KCL2-UC-01	Retrieve User Profile and Life-log Schema
KCL2-UC-02	Retrieve User Profile and Life-log data
KCL2-UC-03	Transfer Production Rules
KCL2-UC-04	Create Situation

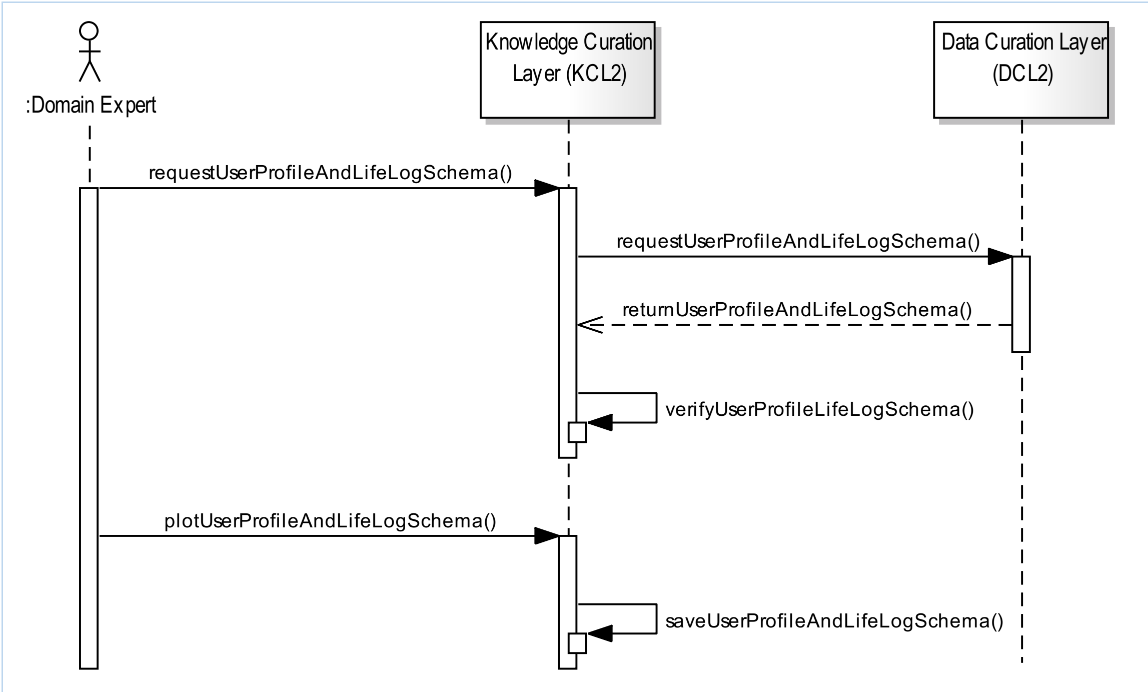
Use case Diagram



Detailed Use case

Use Case ID:	*KCL2-UC-01*		
Use Case Name:	Retrieve User Profile and Life-log Schema		
FR ID:	MM-FR-15		
Created By:	Maqbool Ali	**Last Updated By:**	Maqbool Hussain
Date Created:	06 July 2015	**Last Revision Date:**	14 July 2015
Actors:	Domain Expert, DCL 2		
Description:	In domain engineering, features modeling enables domain experts to capture variability in a domain. Life-log 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		
Pre-conditions:	1. KCL 2 has access through service interface to retrieve life-log and user profile schema from DCL 2 2. KCL 2 and DCL 2 has agreement on common schema representation format 3. DCL 2 has capability to share life-log and user profile schema in secure environment.		
Post-conditions:	KCL 2 will receive life-log and user profile schema conform to its representation scheme.		
Normal Flow:	1. KCL 2 connects to DCL 2 via unified service interface and sends request for life-log and user profile schema. 2. DCL 2 retrieves life-log and user profile schema, transform into common representation format and sends to KCL 2. 3. KCL 2 receives life-log and user profile schema and verifies its conformance and sends received acknowledgement to DCL 2. 4. Domain expert plots the schema and saves it after verification, for further process.		

Alternative Flows:	NA
Exceptions:	1a. KCL 2 unable to connect to DCL 2 1. KCL 2 connection is failed during retrieving life-log and user profile schema 2. KCL 2 hold and will retry after sometime to connect to DCL 2 and retrieve the life-log and user profile schema 3a. KCL 2 unable to verify life-log and user profile schema conformance 1. KCL 2 fail to conform the schema representation from DCL 2 2. KCL 2 will send message to DCL 2 about incompatible schema format
Includes:	NA
Frequency of Use:	Less Frequent: When new service is required and mining mind have sufficient data for classification model creation
NFR ID:	NA
Assumptions:	NA
Notes and Issues:	If DCL 2 is unable to send life-log and user profile schema in required format, then alternate strategy has to be considered.
Sequence Diagram:	

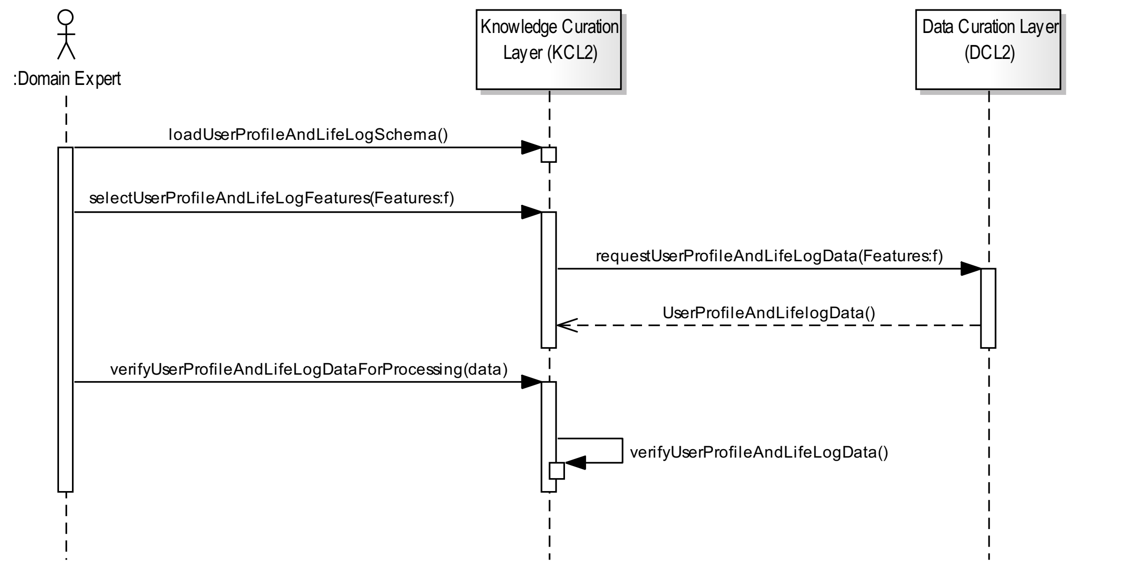


<table>
<tr><td>Use Case ID:</td><td colspan="3">KCL2-UC-02</td></tr>
<tr><td>Use Case Name:</td><td colspan="3">Retrieve User Profile and Life-log data</td></tr>
<tr><td>FR ID:</td><td colspan="3">MM-FR-16</td></tr>
<tr><td>Created By:</td><td>Maqbool Ali</td><td>Last Updated By:</td><td>Maqbool Hussain</td></tr>
<tr><td>Date Created:</td><td>06 July 2015</td><td>Last Revision Date:</td><td>14 July 2015</td></tr>
<tr><td>Actors:</td><td colspan="3">Domain Expert, DCL 2</td></tr>
<tr><td>Description:</td><td colspan="3">Life-log and user profile data has hidden knowledge and it is important to load the life-log information for exploring the hidden knowledge. Furthermore, life-log and user profile data are used for model learning to explore the hidden knowledge.</td></tr>
<tr><td>Trigger:</td><td colspan="3">Prior to classification model creation needed for required domain</td></tr>
<tr><td>Pre-conditions:</td><td colspan="3">1. KCL 2 has access through service interface to retrieve life-log and user profile data from DCL 2
2. DCL 2 has capability to share life-log and user profile data in secure environment.</td></tr>
</table>

	3. KCL 2 has already loaded the previously imported life-log and user profile schema
Post-conditions:	KCL 2 will receive life-log and user profile data based on selected schema.
Normal Flow:	1. Domain expert selects the required features of life-log and user profile schema from whole DCL 2 provided schema 2. KCL 2 connects to DCL 2 via unified service interface and sends request for life-log and user profile data based on selected features of schema. 3. DCL 2 retrieves life-log and user profile data and sends to KCL 2. 4. KCL 2 receives life-log and user profile data for further processing and sends received acknowledgement to DCL 2.
Alternative Flows:	NA
Exceptions:	2a. KCL 2 unable to connect to DCL 2 1. KCL 2 connection is failed during retrieving life-log and user profile data 2. KCL 2 hold and will retry after sometime to connect to DCL 2 and retrieve the life-log and user profile data 3a. KCL 2 receives irrelevant data 1. KCL 2 detects the irrelevant data sent by DCL 2. 2. KCL 2 request again DCL 2 to make sure that data received is according to feature selected.
Includes:	NA
Frequency of Use:	Less Frequent: When new service is required and mining mind have sufficient data for classification model creation
NFR ID:	NA
Assumptions:	NA
Notes and Issues:	If DCL 2 is unable to send life-log and user profile schema in required format, then alternate strategy has

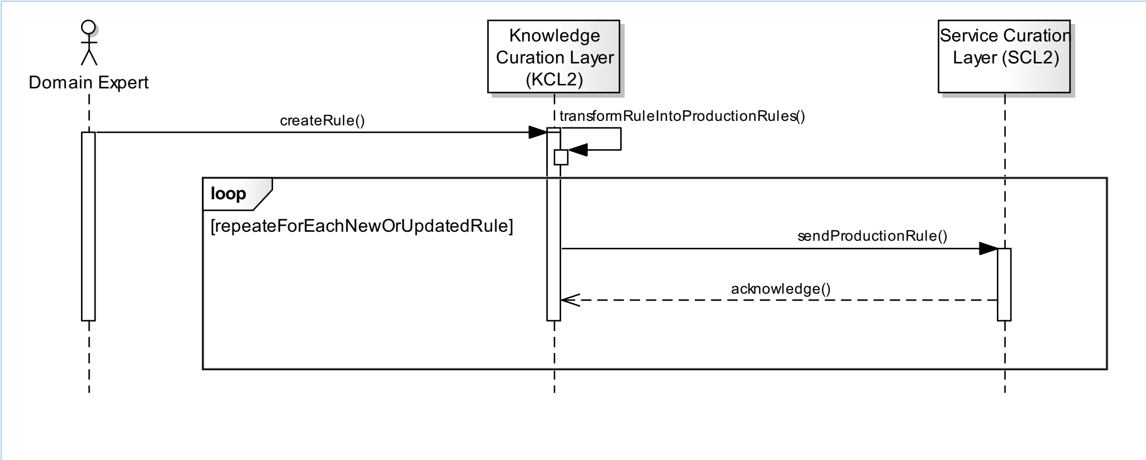
	to be considered.

Sequence Diagram:



Use Case ID:	*KCL2-UC-03*		
Use Case Name:	Transfer Production Rules		
FR ID:	MM-FR-17		
Created By:	Maqbool Ali	**Last Updated By:**	Maqbool Hussain
Date Created:	06 July 2015	**Last Revision Date:**	14 July 2015
Actors:	Domain Expert, SCL 2		
Description:	Transfer Production Rules use case transfer final production rules to SCL 2. SCL 2 integrates these rules to support recommendations for different services.		
Trigger:	New rule creation Update existing rule		
Pre-conditions:	SCL 2 and KCL 2 should agree on common rule representation for final reasoning process		
Post-conditions:	SCL 2 will have latest knowledge from KCL 2		
Normal Flow:	1. Domain expert using KCL 2 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 into production rule(s) representation scheme 3. The transform production rule(s) is/are stored by KCL 2 for maintaining knowledge base and establish connection with SCL 2 for sharing the production rule(s). 4. The production rule(s) is/are transferred to SCL 2. 5. SCL 2 receives the production rule(s) and acknowledge with updated knowledge at SCL 2.
Alternative Flows:	1a. Domain expert modify existing rule(s) using KCL 2 authoring tool 1. Updated rule(s) is/are transformed by KCL 2 into production rule(s) representation scheme 2. Step 2-5 of normal flow is used by indicating updated rule(s) in step 4 to SCL 2.
Exceptions:	3a. KCL 2 unable to connect to SCL 2 1. KCL 2 connection is failed during sharing production rule(s) 2. KCL 2 hold and will retry after sometime to connect to SCL 2 and transfer the production rule(s)
Includes:	NA
Frequency of Use:	Less Frequent: Once per change in knowledge base
NFR ID:	MM-NFR-07, MM-NFR-08
Assumptions:	NA
Notes and Issues:	Common knowledge representation scheme is challenging to implement.
Sequence Diagram:	



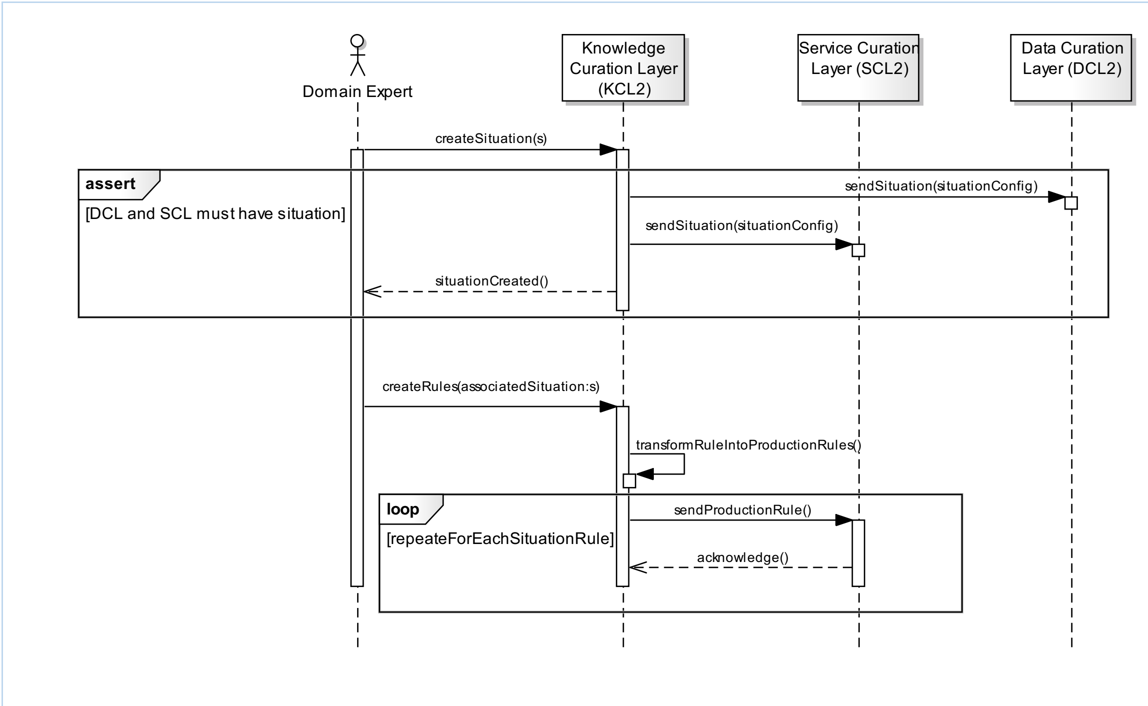
Use Case ID:	*KCL2-UC-04*		
Use Case Name:	Create Situation		
FR ID:	MM-FR-18		
Created By:	Maqbool Ali	**Last Updated By:**	Maqbool Hussain
Date Created:	06 July 2015	**Last Revision Date:**	14 July 2015
Actors:	Domain Expert, DCL 2, SCL 2		
Description:	Situation is important features of mining mind which includes set of associated recommendation rules. Moreover, situation configurations are needed to be shared with DCL 2 to take care and notify SCL 2 to handle it if situation is observed. SCL 2 handle the situation and use KCL 2 production rule(s) to provide appropriate recommendations.		
Trigger:	Situation is identified by domain expert		
Pre-conditions:	KCL 2, DCL 2 and SCL 2 should agree on common representation of sharing Situation configuration		
Post-conditions:	1. KCL 2 has configured situation, shared with DCL 2 2. All associated rules are created and shared with SCL 2		
Normal Flow:	1. Domain expert uses authoring tool of KCL 2 and create situation. 2. KCL 2 connects to DCL 2 and send the newly		

	created situation in common configuration format.
	3. DCL 2 receives the situation configuration and responds with acknowledgement message.
	4. Domain expert creates all associate rules and apply following steps;
	    a. Shares situation configuration with SCL 2
	    b. performs step 2-5 of Transfer Production Rules by transferring the production rule(s) to SCL 2.

Alternative Flows:	NA

Exceptions:	2a. KCL 2 unable to connect to SCL 2
	    a. KCL 2 connection is failed during sharing production rule(s)
	    b. KCL 2 hold and will retry after sometime to connect to SCL 2 and transfer the production rule(s)

Includes:	Transfer Production Rules
Frequency of Use:	Less Frequent: Invoked per situation creation
NFR ID:	MM-NFR-07, MM-NFR-08
Assumptions:	NA
Notes and Issues:	Common configuration format is challenging task.
Sequence Diagram:	

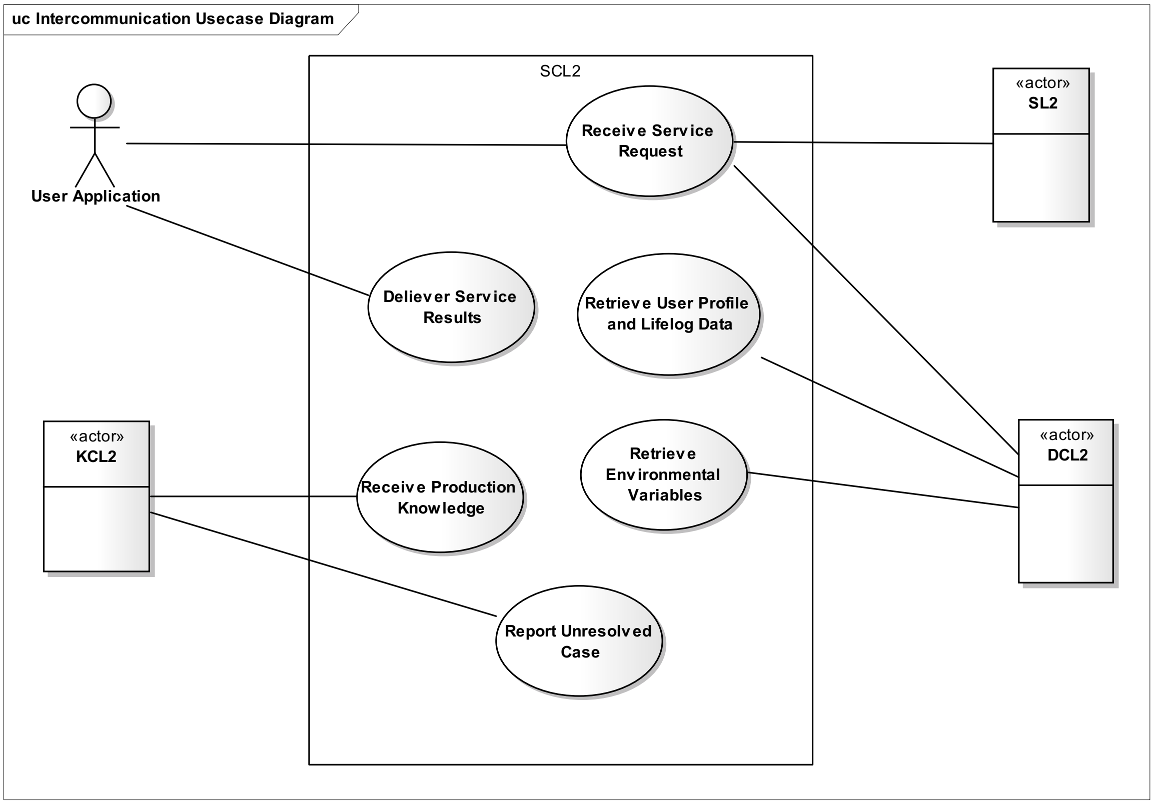

Domain Expert
Knowledge Curation Layer (KCL2)
Service Curation Layer (SCL2)
Data Curation Layer (DCL2)
createSituation(s)
assert
[DCL and SCL must have situation]
sendSituation(situationConfig)
sendSituation(situationConfig)
situationCreated()
createRules(associatedSituation:s)
transformRuleIntoProductionRules()
loop
[repeateForEachSituationRule]
sendProductionRule()
acknowledge()

Service Curation Layer Ver. 2.0 (SCL 2)

Use case List

Use case ID#	Name
SCL2-UC-01	Receive service request from user application or mining mind generated events
SCL2-UC-02	Retrieve user profile and life-log data for reasoning from intermediate database
SCL2-UC-03	Retrieve Production Knowledge from the knowledge base created by knowledge curation layer
SCL2-UC-04	Report unresolved case for acquiring the missing knowledge
SCL2-UC-05	Retrieve environment variable data for interpretations from intermediate database
SCL2-UC-06	Deliver service results to service requester and to intermediate database for persistence

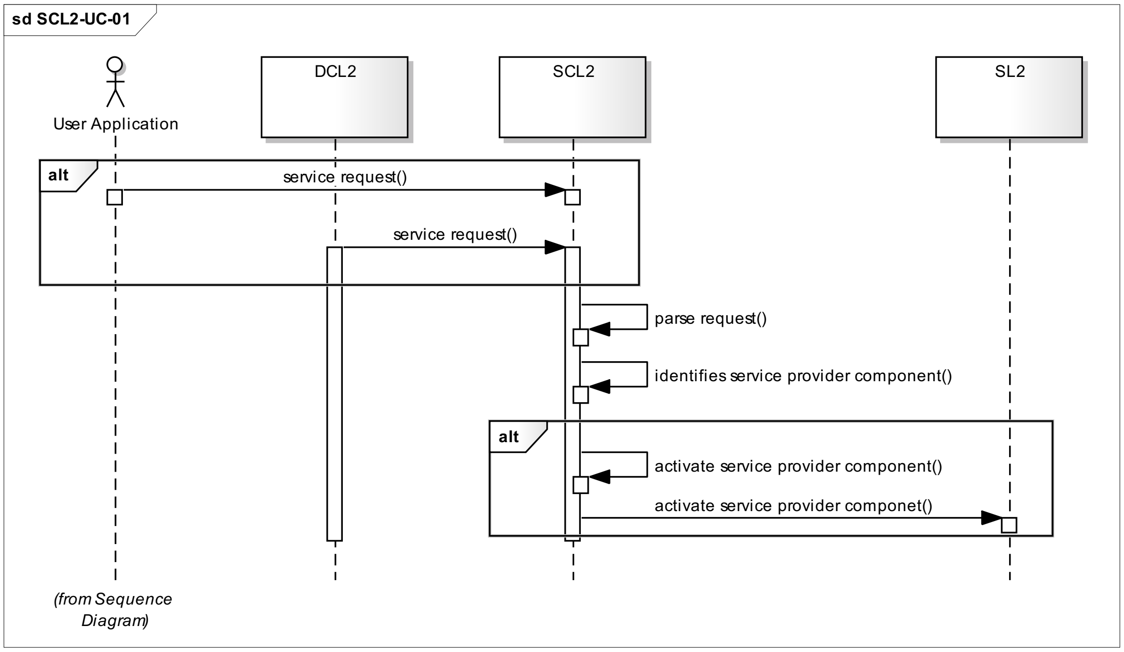
Use case Diagram



Detailed Use case

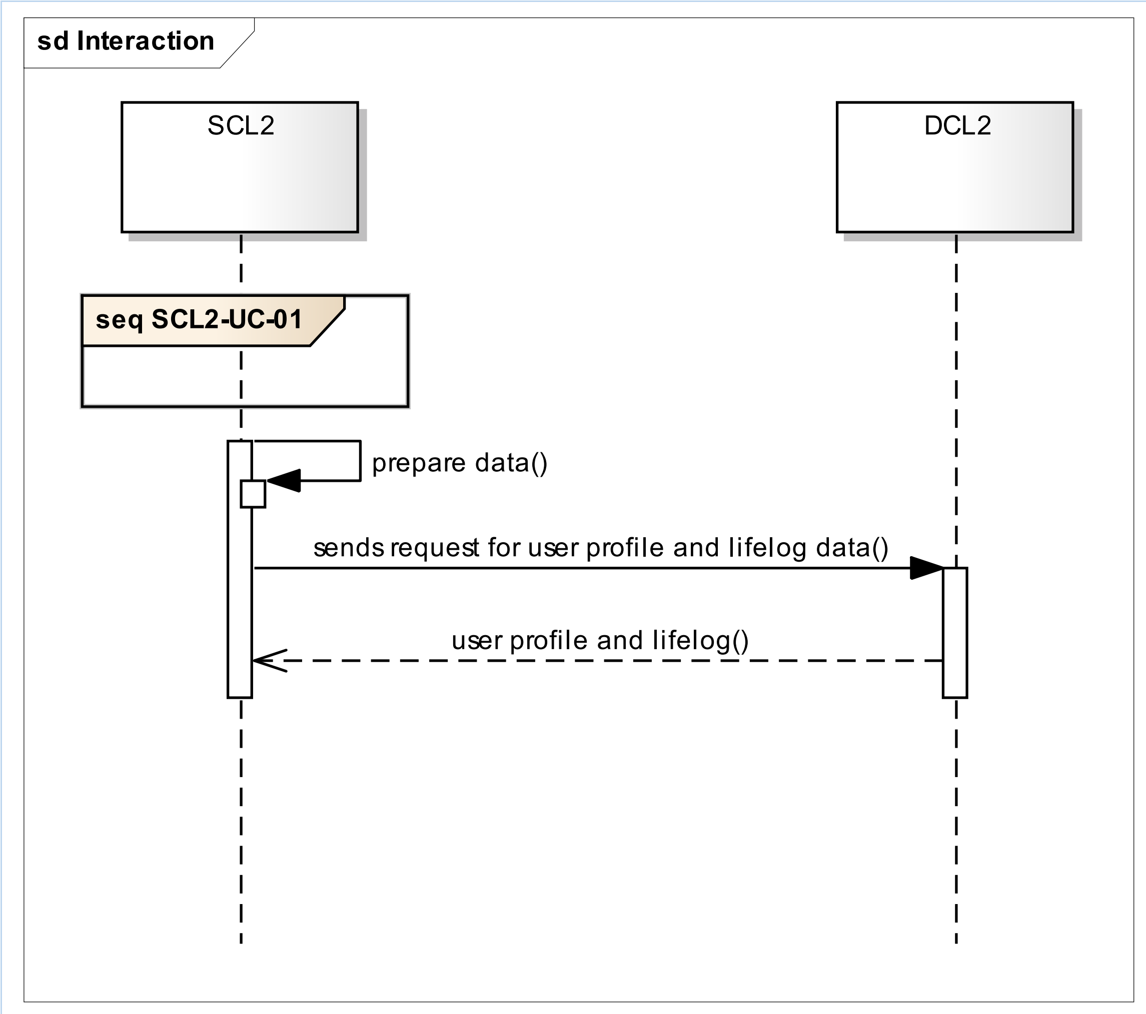
Use Case ID:	*SCL2-UC-01*		
Use Case Name:	Receive service request from user application or mining mind generated events		
FR ID:	MM-FR-19		
Created By:	Muhammad Afzal	**Last Updated By:**	Muhammad Afzal
Date Created:	06 July 2015	**Last Revision Date:**	14 July 2015
Actors:	User Application, DCL 2		
Description:	SCL 2 allows to receive request from the user application, or DCL 2 for a service. SCL 2 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		
Normal Flow:	1. SCL 2 receives the service request from user application 2. SCL 2 parses the request 3. SCL 2 identifies the service requirements 4. SCL 2 passes the message to invoked the service in SCL 2		
Alternative Flows:	1.a SCL 2 receives the request as an interrupt from DCL 2 whenever a situation occurs 4.a SCL 2 passes the message to invoke the service in SL 2		
Exceptions:	NA		
Includes:	NA		
Frequency of Use:	Very frequent: at every service request		
NFR ID:	MM-NFR-09		
Assumptions:	SCL 2 and SL 2/DCL 2 contract is agreed.		

Notes and Issues:	NA
Sequence Diagram:	



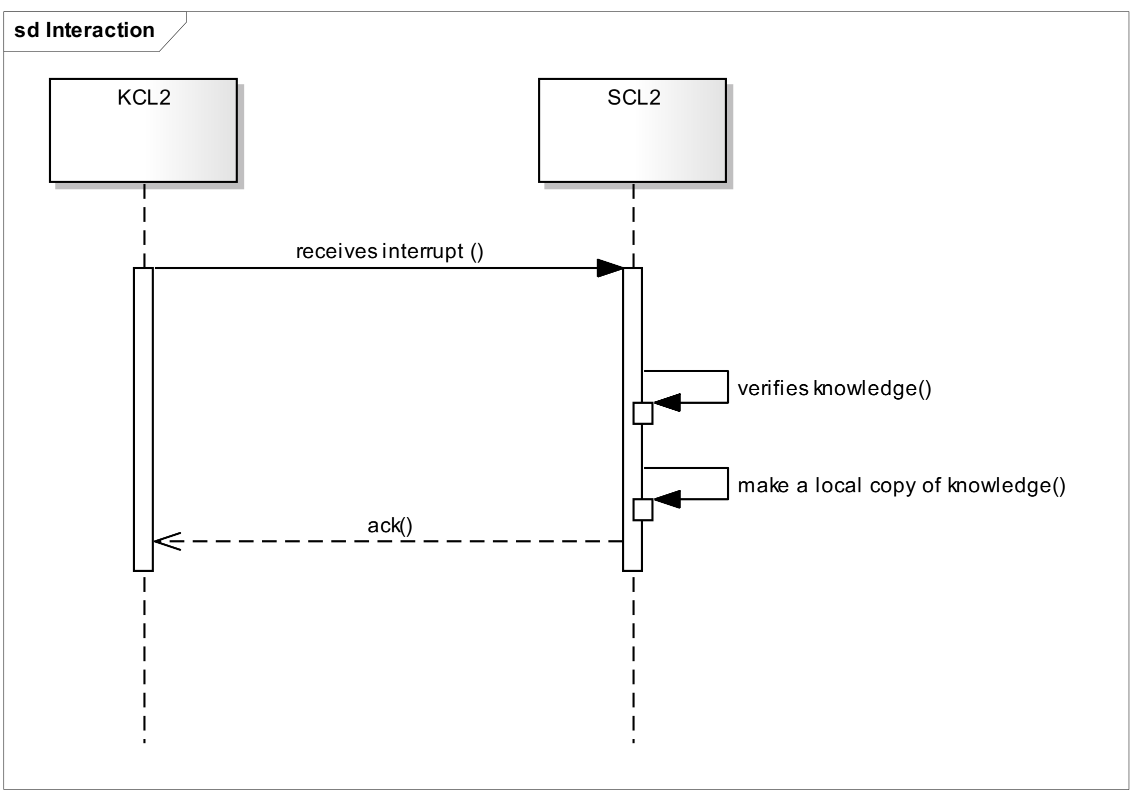
Use Case ID:	*SCL2-UC-02*		
Use Case Name:	Retrieve user profile/life-log data from user life-log		
FR ID:	MM-FR-20		
Created By:	Muhammad Afzal	**Last Updated By:**	Muhammad Afzal
Date Created:	06 July 2015	**Last Revision Date:**	14 July 2015
Actors:	SCL 2, DCL 2		
Description:	Retrieving user profile and life-log data is required for reasoning over knowledge. Since the data resides in DCL 2, so SCL 2 as a primary actor originates the request to DCL 2 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 to process.		

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

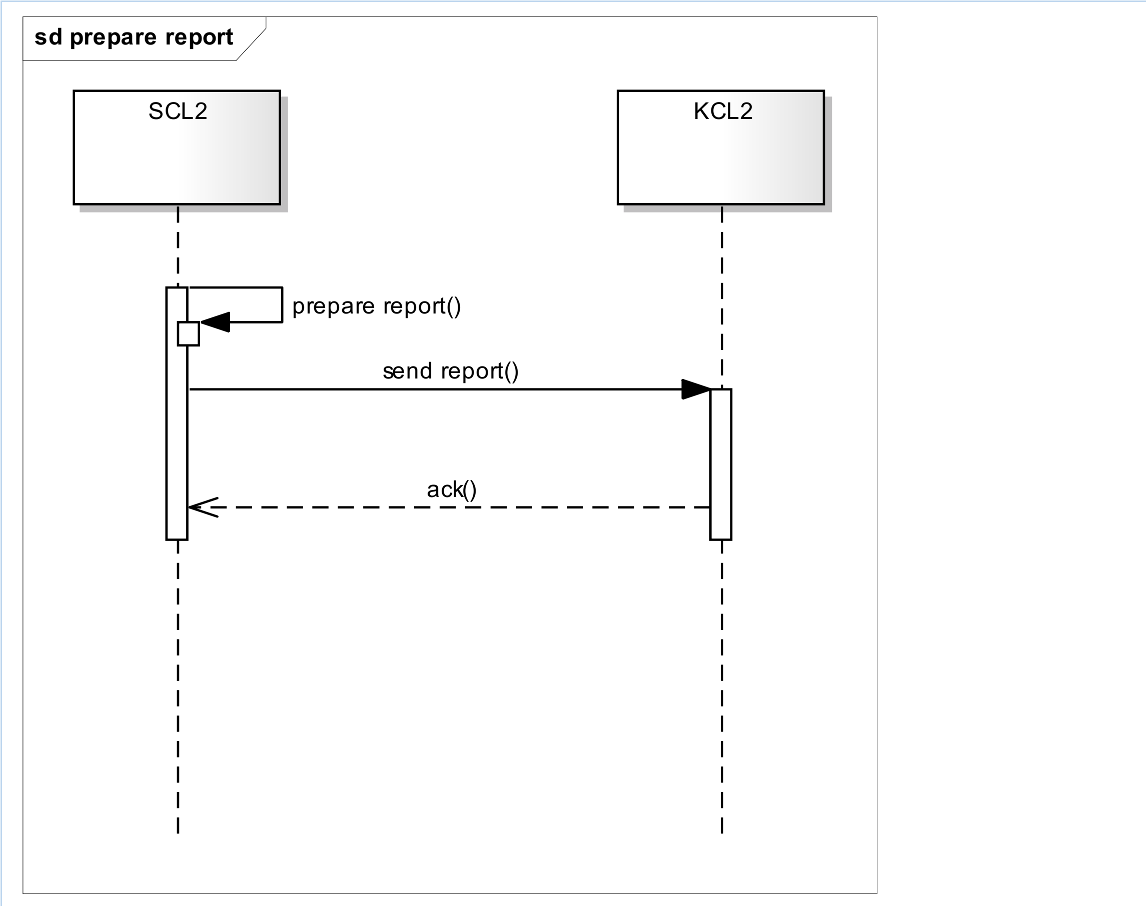


Use Case ID:	*SCL2-UC-03*		
Use Case Name:	Receive Production Knowledge from knowledge curation layer		
FR ID:	MM-FR-21		
Created By:	Muhammad Afzal	**Last Updated By:**	Muhammad Afzal
Date Created:	06 July 2015	**Last Revision Date:**	14 July 2015
Actors:	KCL 2, SCL 2		
Description:	The knowledge is originated by KCL 2 and is transfer to SCL 2 to keep a local copy of the production knowledge.		

Trigger:	At knowledge creation/update time
Pre-conditions:	SCL 2 and KCL 2 has a common representation agreement
Post-conditions:	The SCL 2 copy of knowledge is updated and is synchronized with KCL 2
Normal Flow:	1. SCL 2 receives interrupt from KCL 2 2. SCL 2 verifies the knowledge 3. SCL 2 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:	

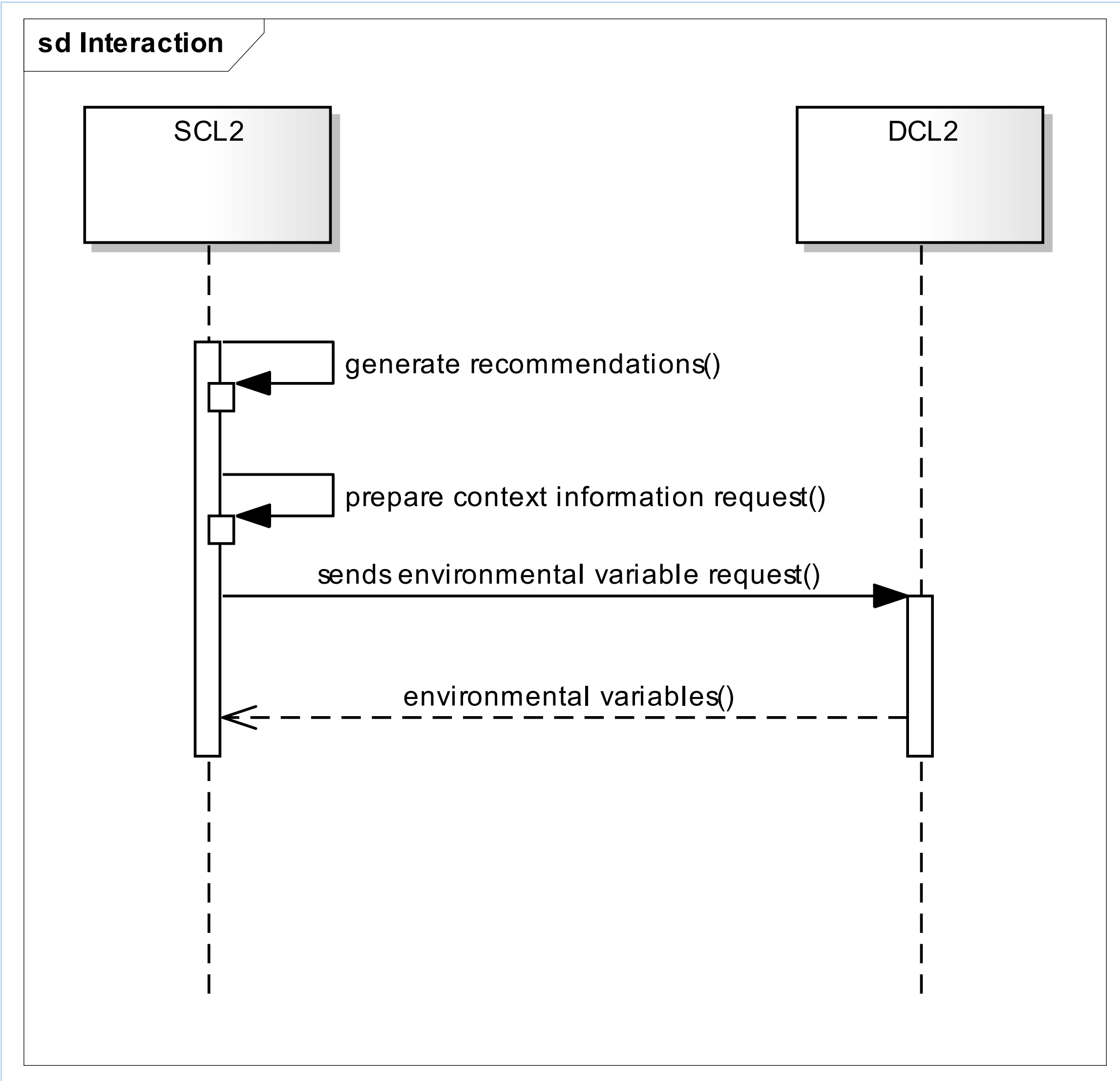


Use Case ID:	*SCL2-UC-04*		
Use Case Name:	Report unresolved case for acquiring the missing knowledge		
FR ID:	MM-FR-22		
Created By:	Muhammad Afzal	**Last Updated By:**	Muhammad Afzal
Date Created:	06 July 2015	**Last Revision Date:**	14 July 2015
Actors:	SCL 2, KCL 2		
Description:	Notifying KCL 2 that reasoner is incapable to generate the recommendation for the service request. KCL 2 may be able to acquired new knowledge for such service request to handle in future.		
Trigger:	At the time when reasoner is not capable to generate recommendation because of insufficient knowledge in the KB.		
Pre-conditions:	Reasoner has completed the reasoning process.		
Post-conditions:	The message with reason is successfully reported to KCL 2		
Normal Flow:	1. SCL 2 prepare the report 2. SCL 2 sends the report to KCL 2 3. SCL 2 receives the acknowledgement		
Alternative Flows:	NA		
Exceptions:	NA		
Includes:	NA		
Frequency of Use:	Less frequent: when reasoner detects new case not handled with existing knowledge.		
NFR ID:	MM-NFR-09		
Assumptions:	NA		
Notes and Issues:	NA		
Sequence Diagram:			



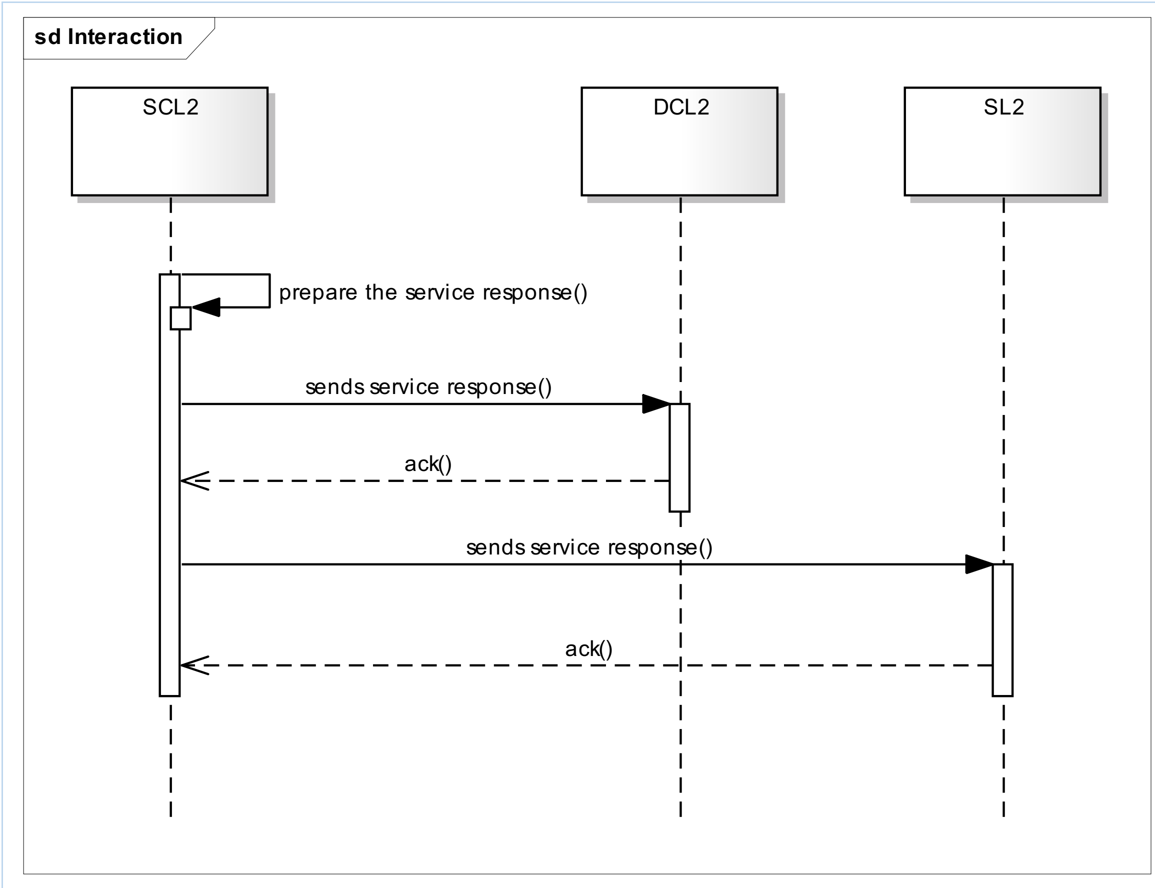
Use Case ID:	*SCL2-UC-05*		
Use Case Name:	Retrieve environmental variables from intermediate database		
FR ID:	MM-FR-20		
Created By:	Muhammad Afzal	**Last Updated By:**	Muhammad Afzal
Date Created:	06 July 2015	**Last Revision Date:**	14 July 2015
Actors:	SCL 2, DCL 2		
Description:	Retrieving contextual data is required by Recommendation Interpreter component to perform its interpretations. Since the data resides in DCL 2, so SCL 2 as a primary actor originates the request to DCL 2 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 to process.
Normal Flow:	1. SCL 2 generates the recommendations 2. SCL 2 prepare the information request 3. SCL 2 sends the request to DCL 2 for environmental variables 4. SCL 2 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-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:	06 July 2015	**Last Revision Date:**	14 July 2015
Actors:	SCL 2, SL2, DCL 2		
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 and DCL 2
Normal Flow:	1. SCL 2 prepares the recommendation output 2. SCL 2 sends the output to DCL for storage 3. SCL 2 receives the acknowledgement 4. SCL 2 sends the output to SL 5. SCL 2 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 has the agreed contract with SL 2 and DCL 2
Notes and Issues:	NA
Sequence Diagram:	

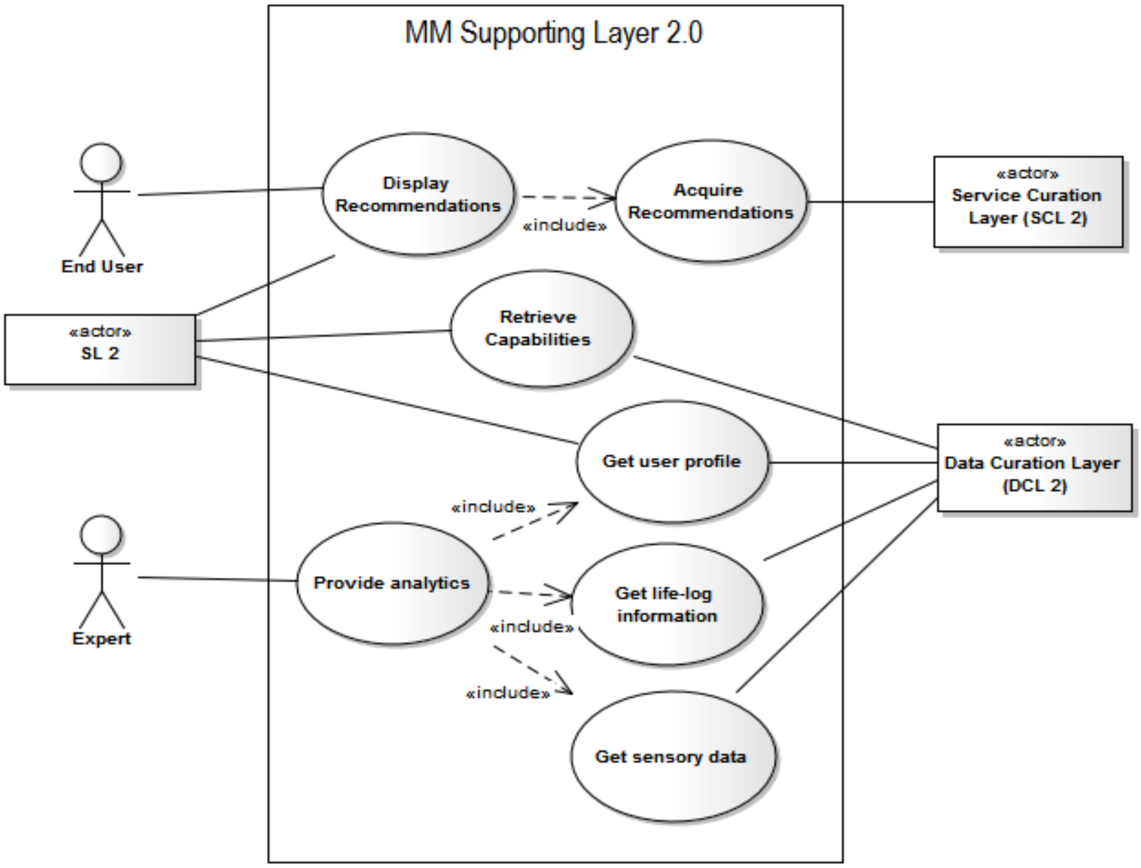

sd Interaction
SCL2
DCL2
SL2
prepare the service response()
sends service response()
ack()
sends service response()
ack()

Supporting Layer Ver. 2.0 (SL 2)

Use case List

Use case ID#	Description
SL2-UC-01	Acquire Recommendations for displaying to end user
SL2-UC-02	Retrieve Capabilities for user interface adaption
SL2-UC-03	Collect user profile data for user satisfaction
SL2-UC-04	Get user profile, life-log and sensory data for analytics

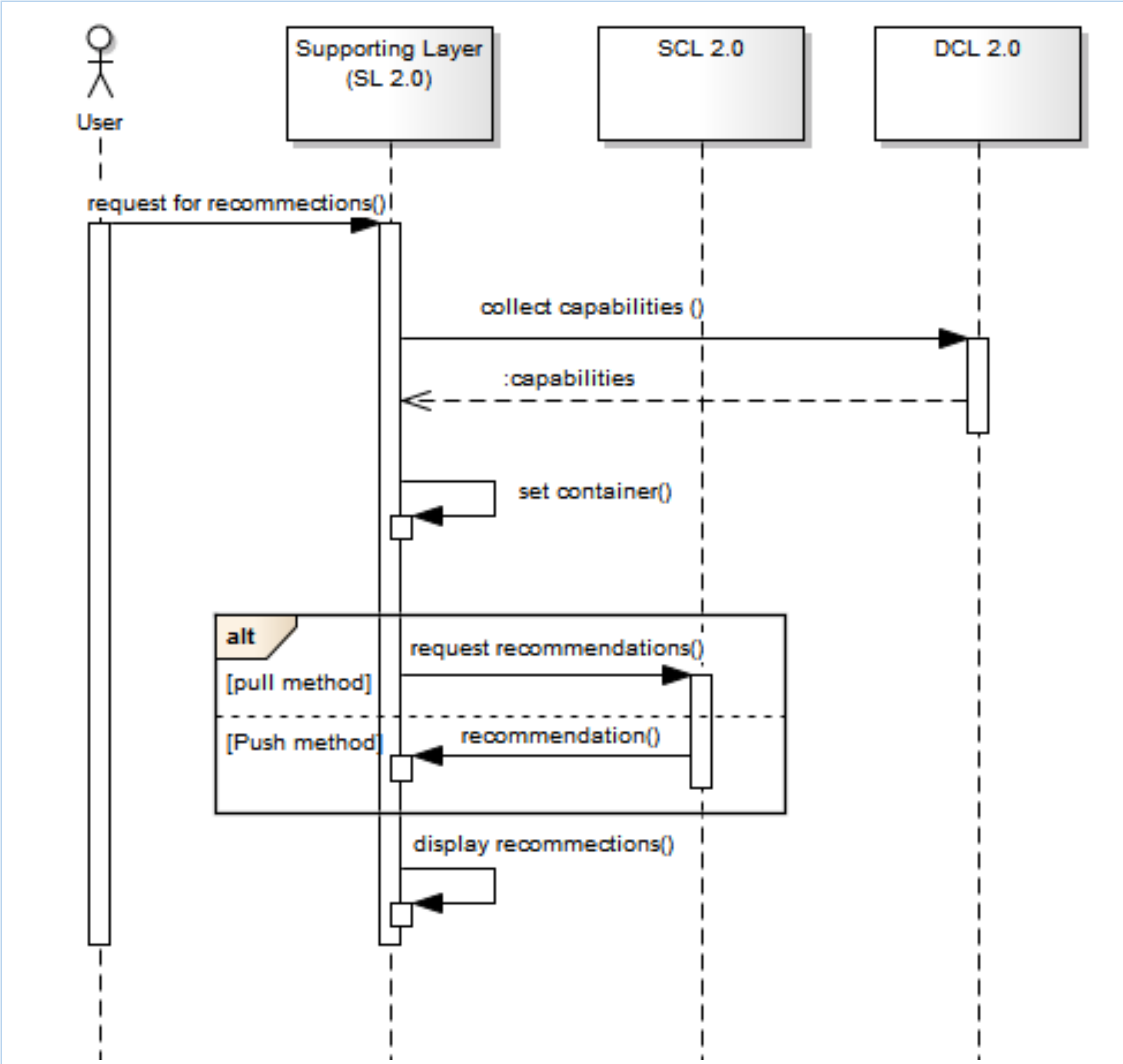
Use case Diagram



Detailed Use case

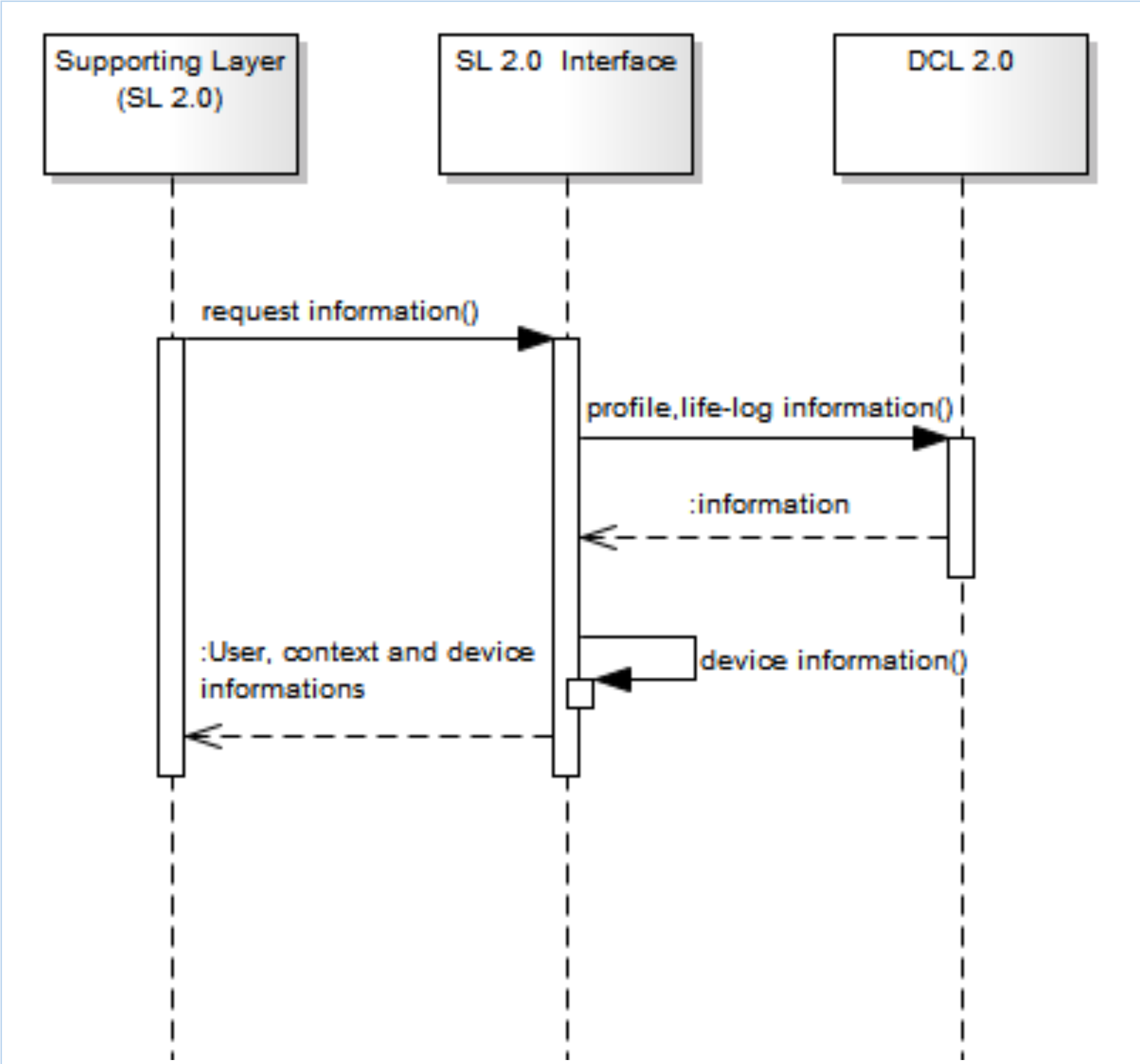
Use Case ID:	*SL2-UC-01*		
Use Case Name:	Acquire Recommendations for displaying to end user		
FR ID:	MM-FR-24		
Created By:	Jamil Hussain	**Last Updated By:**	Wajahat Ali
Date Created:	06 July 2015	**Last Revision Date:**	10 July 2015
Actors:	End-user, SCL 2, DCL 2		
Description:	This use case collects the recommendations generated by SCL 2 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.		
Trigger:	SCL 2 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 generate the recommendations and provide it to user interface 2. The SCL 2 recommendations are acquired by the SL 2 3. SL 2 investigates the user capabilities, context of use, and device characteristics by obtaining from DCL 2 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 recommendations are acquired by the SL 2 1. user request for recommendations (pull method) 2b. In step 2. The SCL 2 recommendations are acquired by the SL		

	1. SL 2 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 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
NFR ID:	MM-NFR-10
Assumptions:	The user profile data and context information should exist in the DCL 2
Notes and Issues:	NA
Sequence Diagram:	



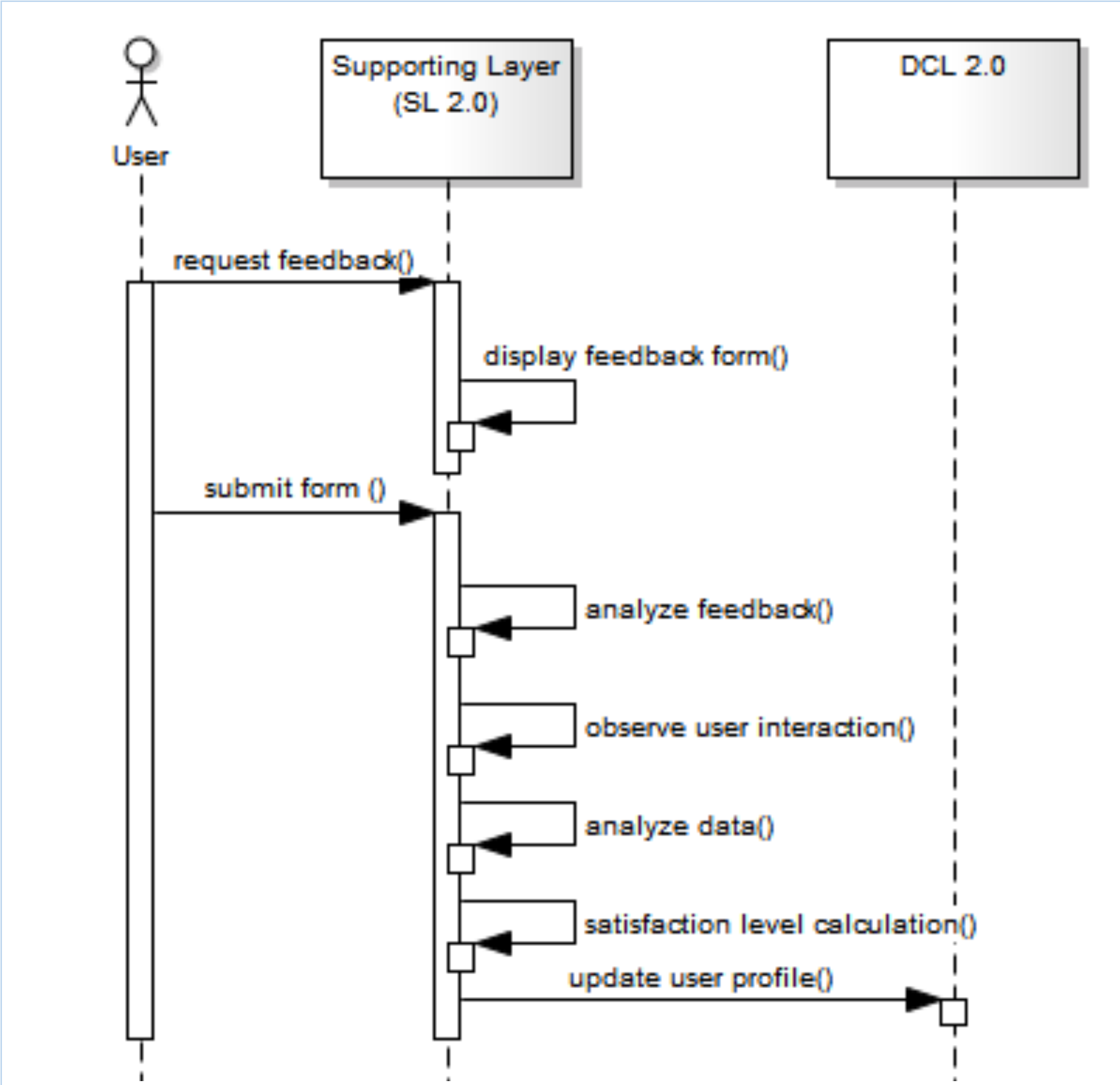
Use Case ID:	SL2-UC-02		
Use Case Name:	Retrieve Capabilities for user interface adaptation		
FR ID:	MM-FR-25		
Created By:	Jamil Hussain	**Last Updated By:**	Wajahat Ali
Date Created:	06 July 2015	**Last Revision Date:**	10 July 2015
Actors:	SL 2, DCL 2		
Description:	This use case focuses on the retrieval 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 provide the access to required information
Post-conditions:	All required capabilities are successfully collected.
Normal Flow:	1. SL 2 generates request for user, device, and context information collection from DCL 2 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.
Notes and Issues:	N/A
Sequence Diagram:	



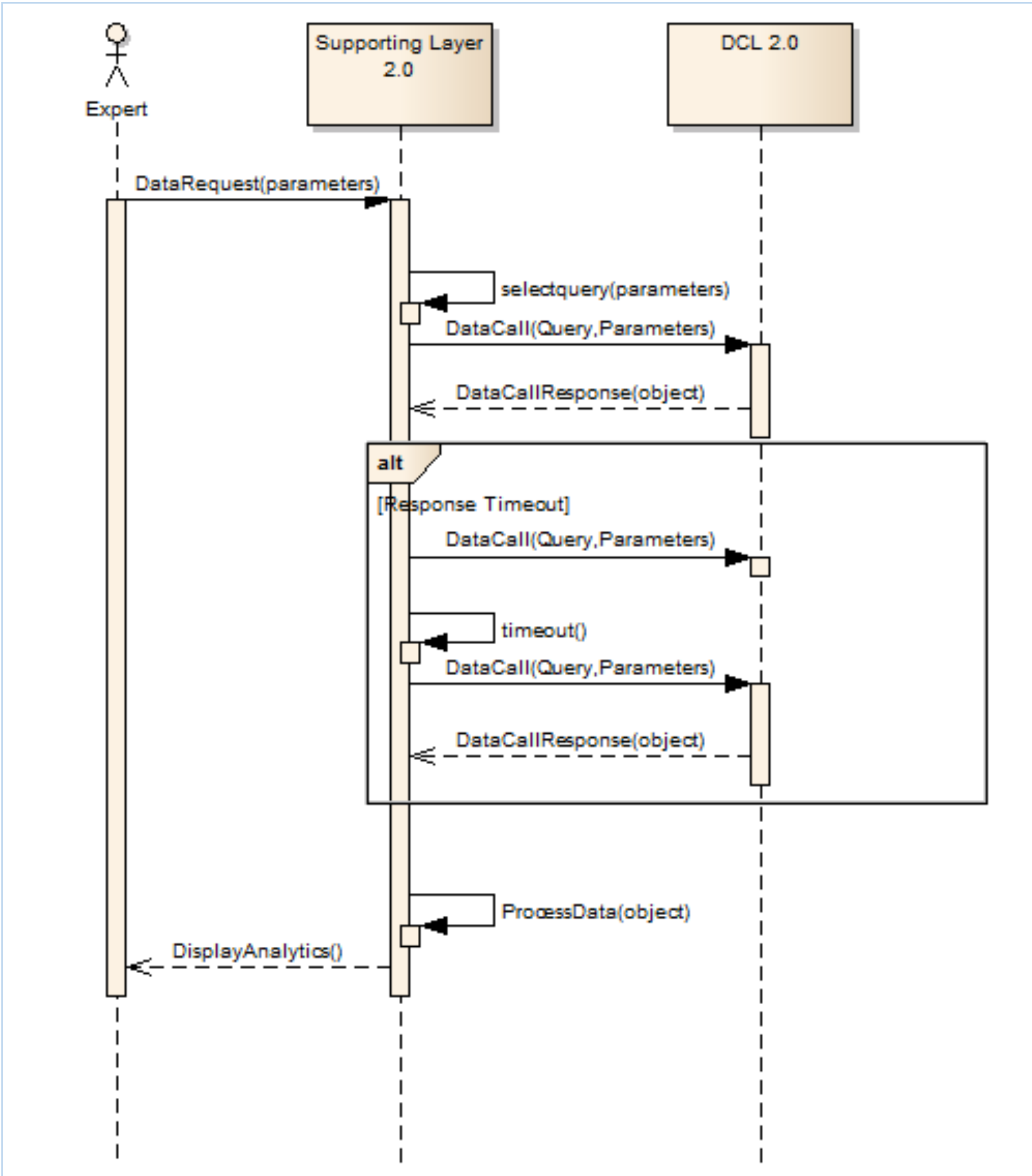
Use Case ID:	*SL2-UC-03*		
Use Case Name:	Collect user profile data for user satisfaction		
FR ID:	MM-FR-26		
Created By:	Jamil Hussain	**Last Updated By:**	Wajahat Ali
Date Created:	06 July 2015	**Last Revision Date:**	10 July 2015
Actors:	SL 2, DCL 2		
Description:	This use case focuses on the collection 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
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-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:**	Wajahat Ali
Date Created:	06 July 2015	**Last Revision Date:**	10 July 2015
Actors:	Domain Expert, DCL 2		
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:	


Expert
Supporting Layer 2.0
DCL 2.0
DataRequest(parameters)
selectquery(parameters)
DataCall(Query,Parameters)
DataCallResponse(object)
alt
[Response Timeout]
DataCall(Query,Parameters)
timeout()
DataCall(Query,Parameters)
DataCallResponse(object)
ProcessData(object)
DisplayAnalytics()

Section 4-B

Design Document

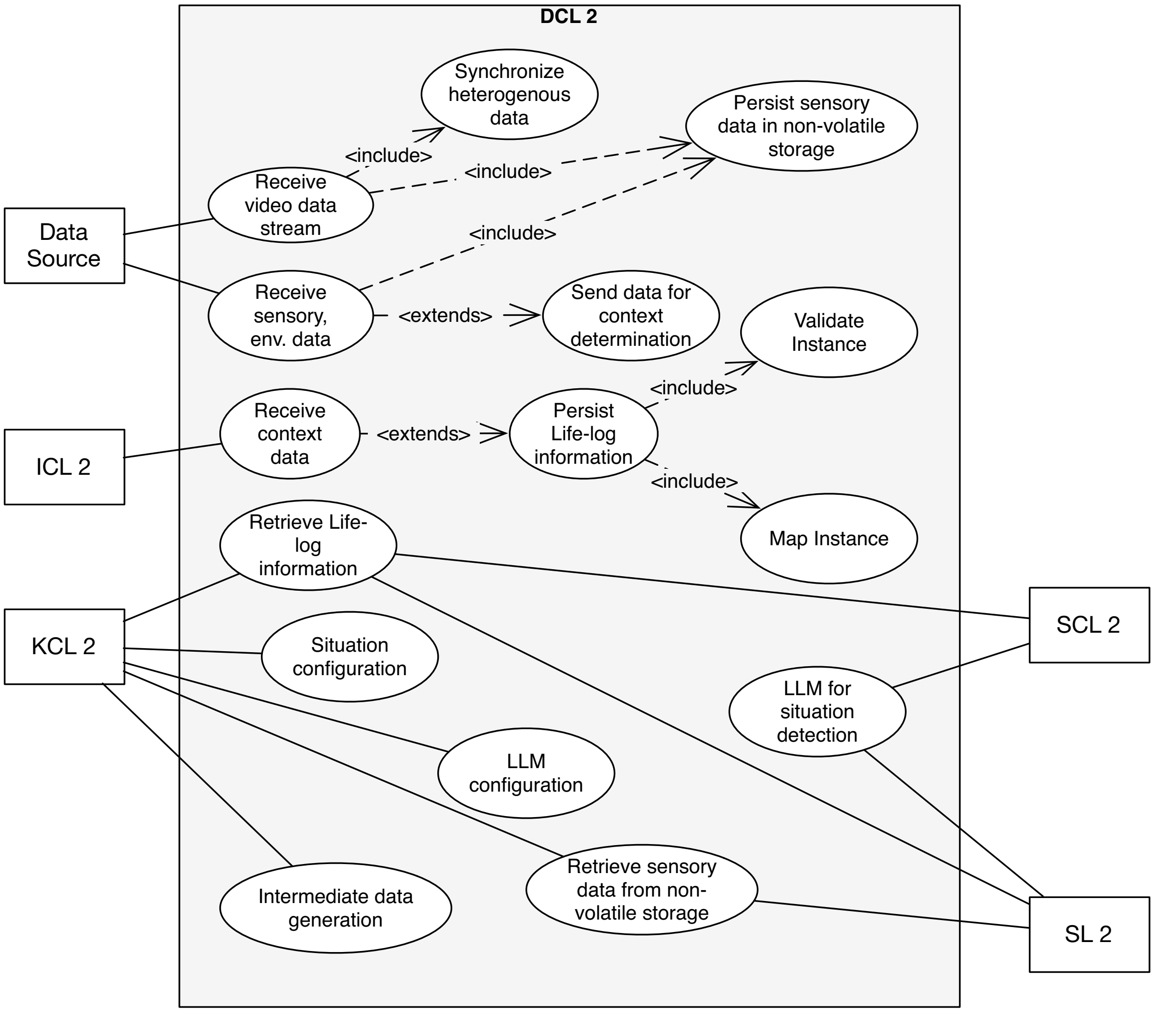
Data Curation Layer (DCL Ver. 2)

System Level Use cases

List of Use cases

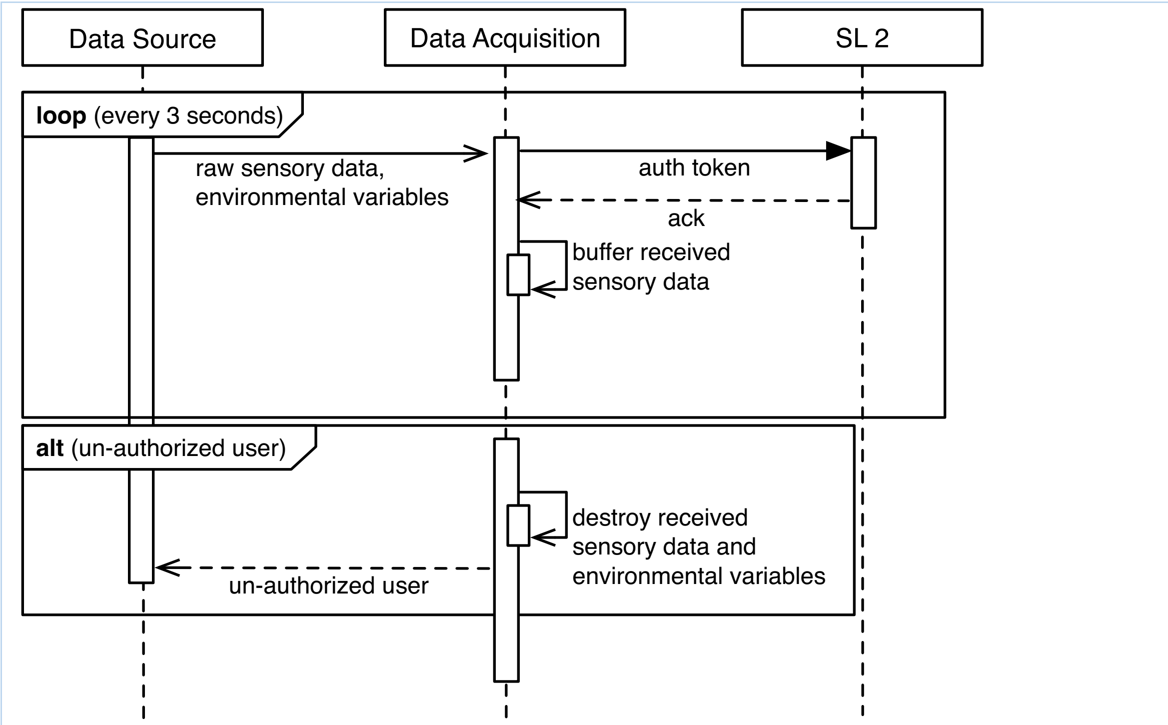
Use case ID#	Name
DCL2-SUC-01	Receive sensory and environmental data from data source
DCL2-SUC-02	Receive video data stream from data source
DCL2-SUC-03	Synchronize heterogeneous user data
DCL2-SUC-04	Send data for context determination
DCL2-SUC-05	Receive context data
DCL2-SUC-06	Retrieve Life-log Information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

Use case Diagram



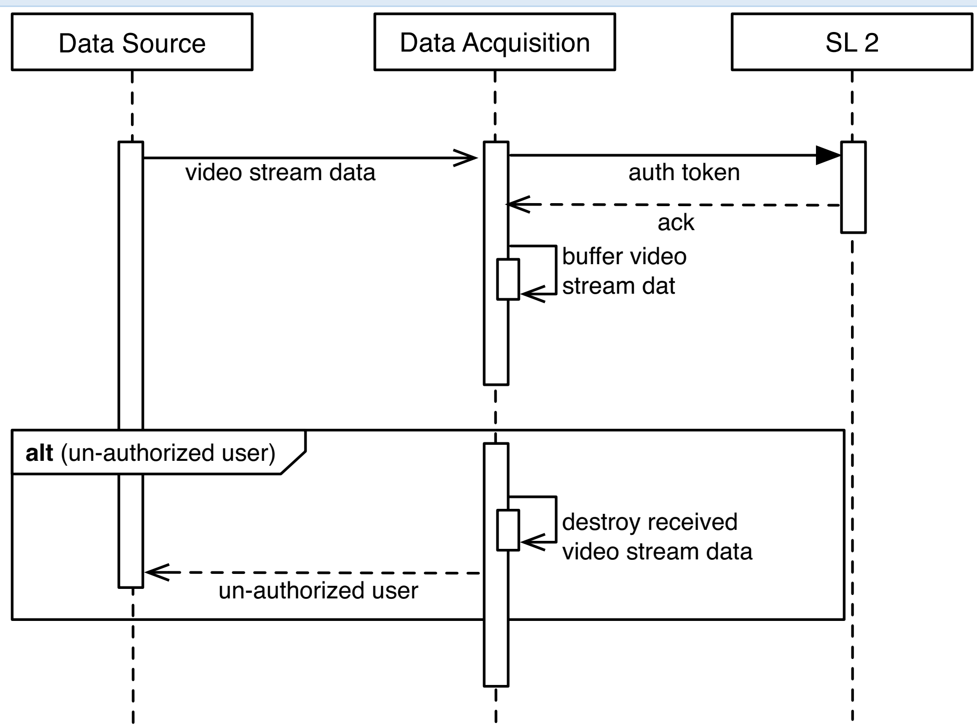
Use case Description

Use Case ID:	*DCL2-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 July 2015
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-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 July 2015
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:	

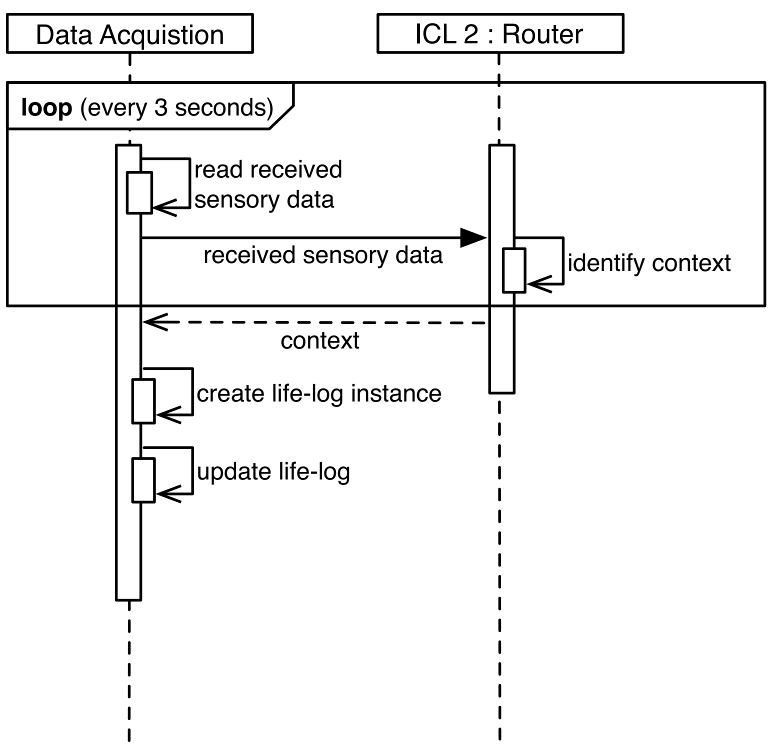


Use Case ID:	*DCL2-SUC-03*		
Use Case Name:	Synchronize heterogeneous user data		
Created By:	Bilal Amin	**Last Updated By:**	Bilal Amin
Date Created:	15 July 2015	**Last Revision Date:**	20 July 2015
Actors:	DCL 2		

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-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 July 2015
Actors:	Context sender		

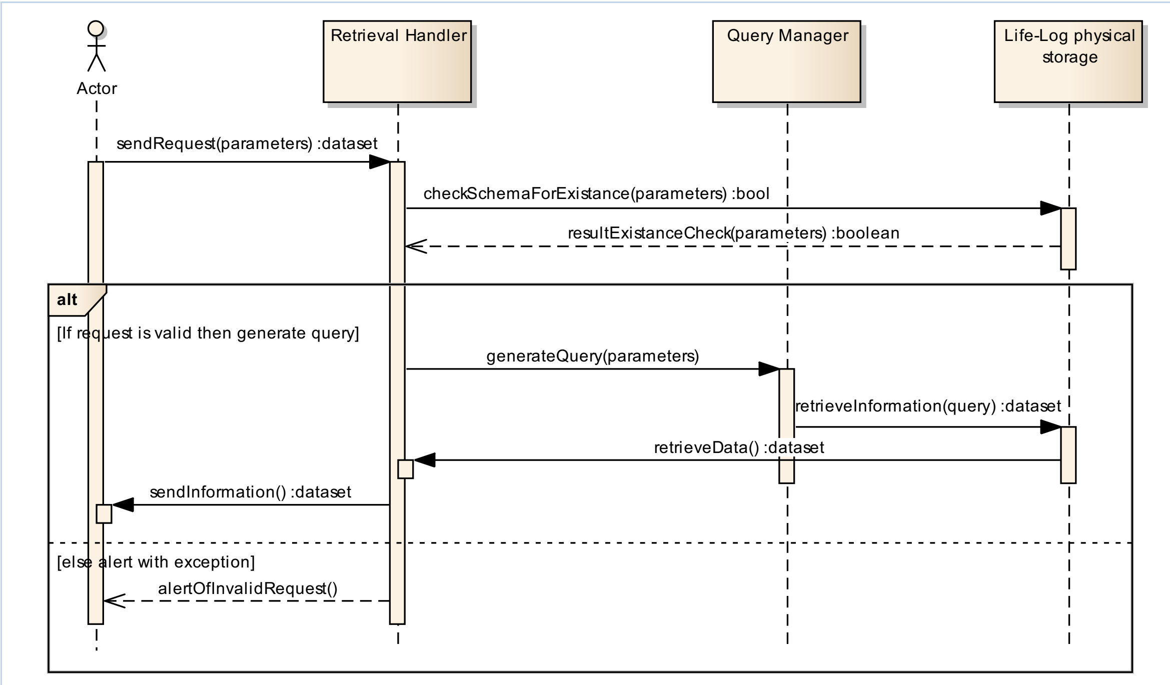
Description:	Sensory data buffer is sent to ICL 2 for context determination
Trigger:	Sensory data is available for context determination
Pre-conditions:	Sensory data is persisted in the buffer
Post-conditions:	Sensory data is sent for context determination
Normal Flow:	1. Context sender reads sensory data from the sensory data buffer 2. Context sender creates communication object by serialization 3. Communication object is sent to the ICL 2 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-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 July 2015
Actors:	ICL 2 server		
Description:	After the determination by ICL 2, 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:			

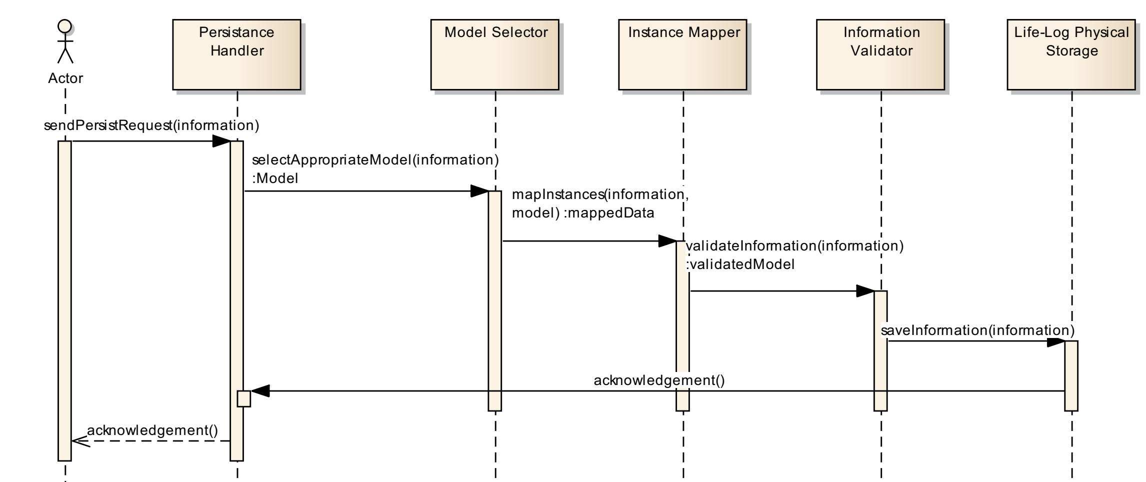
Use Case ID:	*DCL2-SUC-06*		
Use Case Name:	Retrieve Life-log Information		
Created By:	Taqdir Ali	**Last Updated By:**	Taqdir Ali
Date Created:	15 July 2015	**Last Revision Date:**	15 July 2015
Actors:	SCL 2, KCL 2, SL 2		

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:	



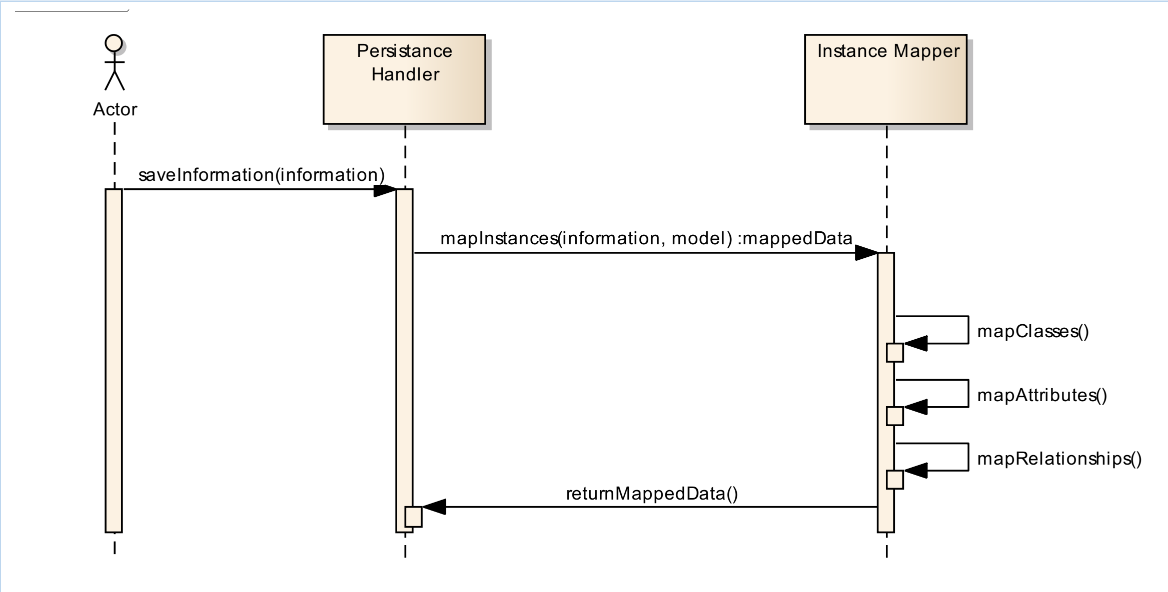
Use Case ID:	*DCL2-SUC-07*		
Use Case Name:	Persist Life-log Information		
Created By:	Taqdir Ali	**Last Updated By:**	Taqdir Ali
Date Created:	15 July 2015	**Last Revision Date:**	15 July 2015
Actors:	ICL 2		
Description:	Each actor performed some specific operations on incoming data from external resources 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:	



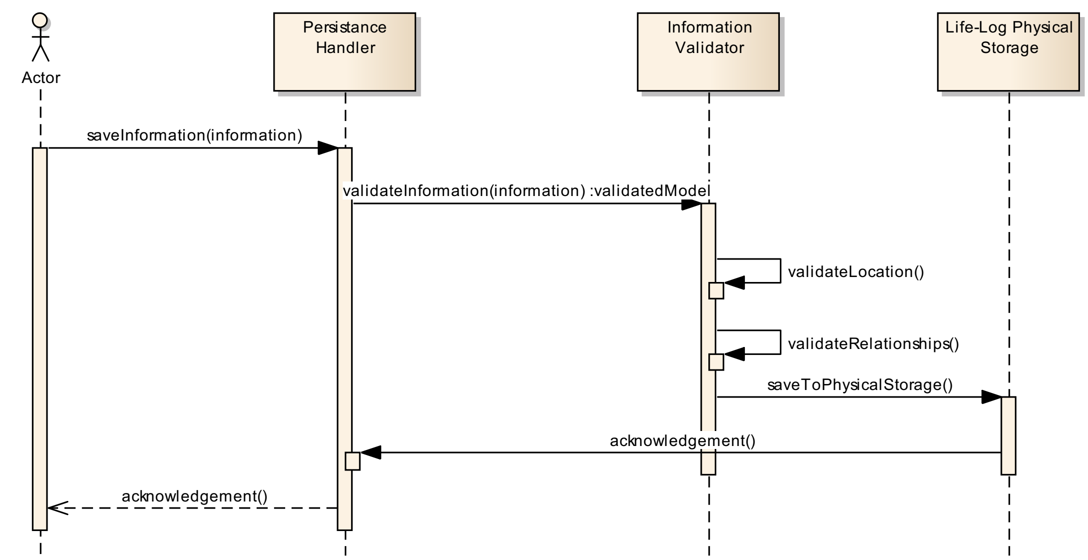
Use Case ID:	*DCL2-SUC-08*		
Use Case Name:	Map Instances		
Created By:	Taqdir Ali	**Last Updated By:**	Taqdir Ali
Date Created:	15 July 2015	**Last Revision Date:**	15 July 2015
Actors:	DCL2, ICL2, SCL2, SL2, and Life-Log physical storage		
Description:	The information produced by each actor shall be 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:	



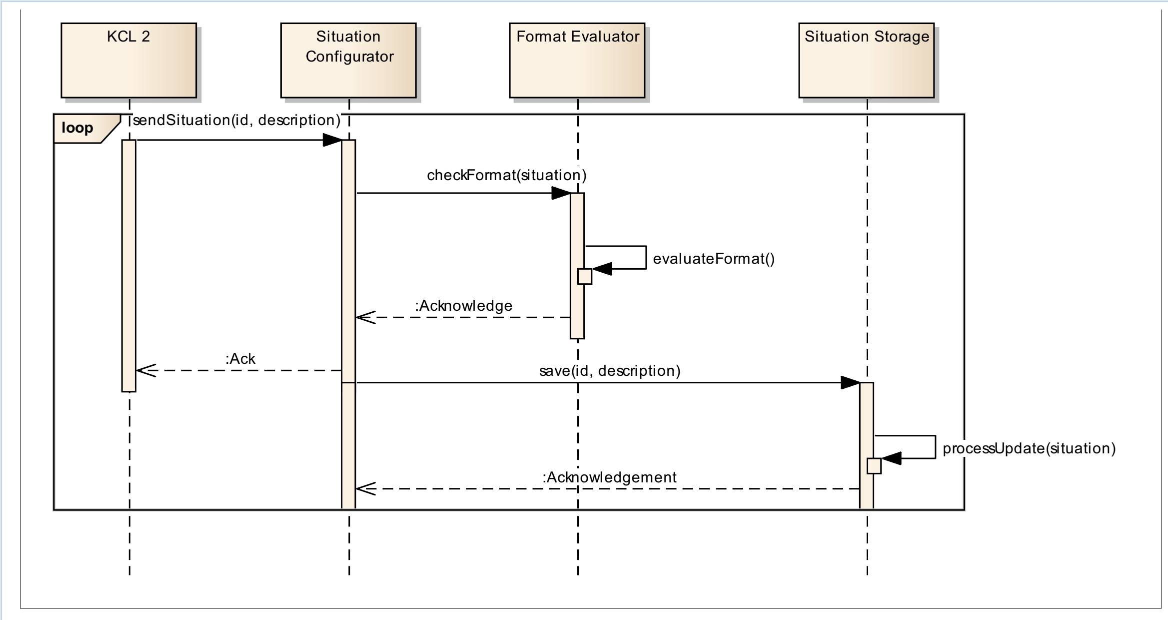
Use Case ID:	*DCL2-SUC-09*		
Use Case Name:	Validate Instances		
Created By:	Taqdir Ali	**Last Updated By:**	Taqdir Ali
Date Created:	15 July 2015	**Last Revision Date:**	15 July 2015
Actors:	DCL2, ICL2, SCL2, SL2, and Life-Log physical storage		
Description:	The mapped information in previous use case 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:	



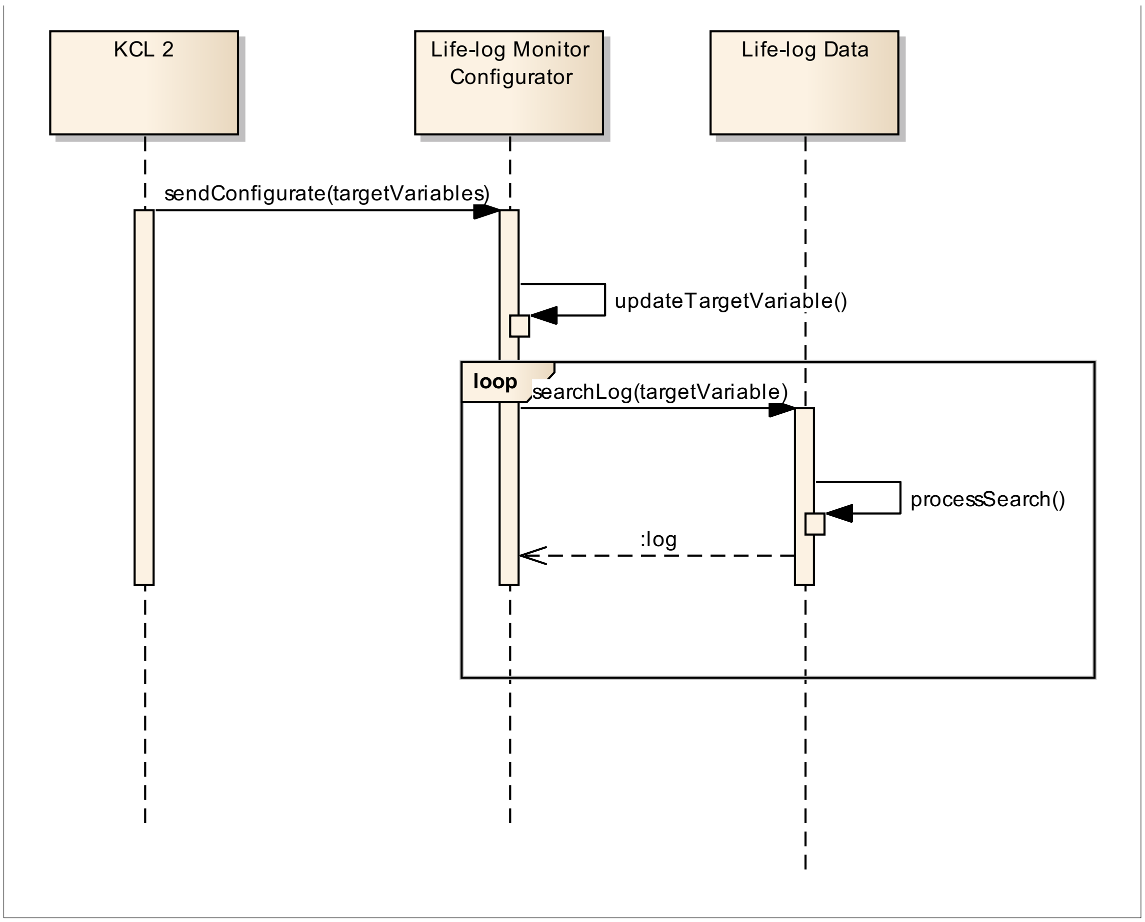
Use Case ID:	*DCL2-SUC-10*		
Use Case Name:	Situation Configuration		
Created By:	Bilal Ali	**Last Updated By:**	Bilal Ali
Date Created:	15 July 2015	**Last Revision Date:**	15 July 2015
Actors:	KCL 2		
Description:	Situation is determined by experts and is communicated to DCL 2 for monitoring the Life-log.		

Trigger:	Creation of new rule to capture a situation.
Pre-conditions:	KCL 2 and DCL 2 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 connects to DCL 2 and send the newly created situation in common configuration format. 2. DCL 2 evaluates the format of received situation configuration. 3. DCL 2 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:	



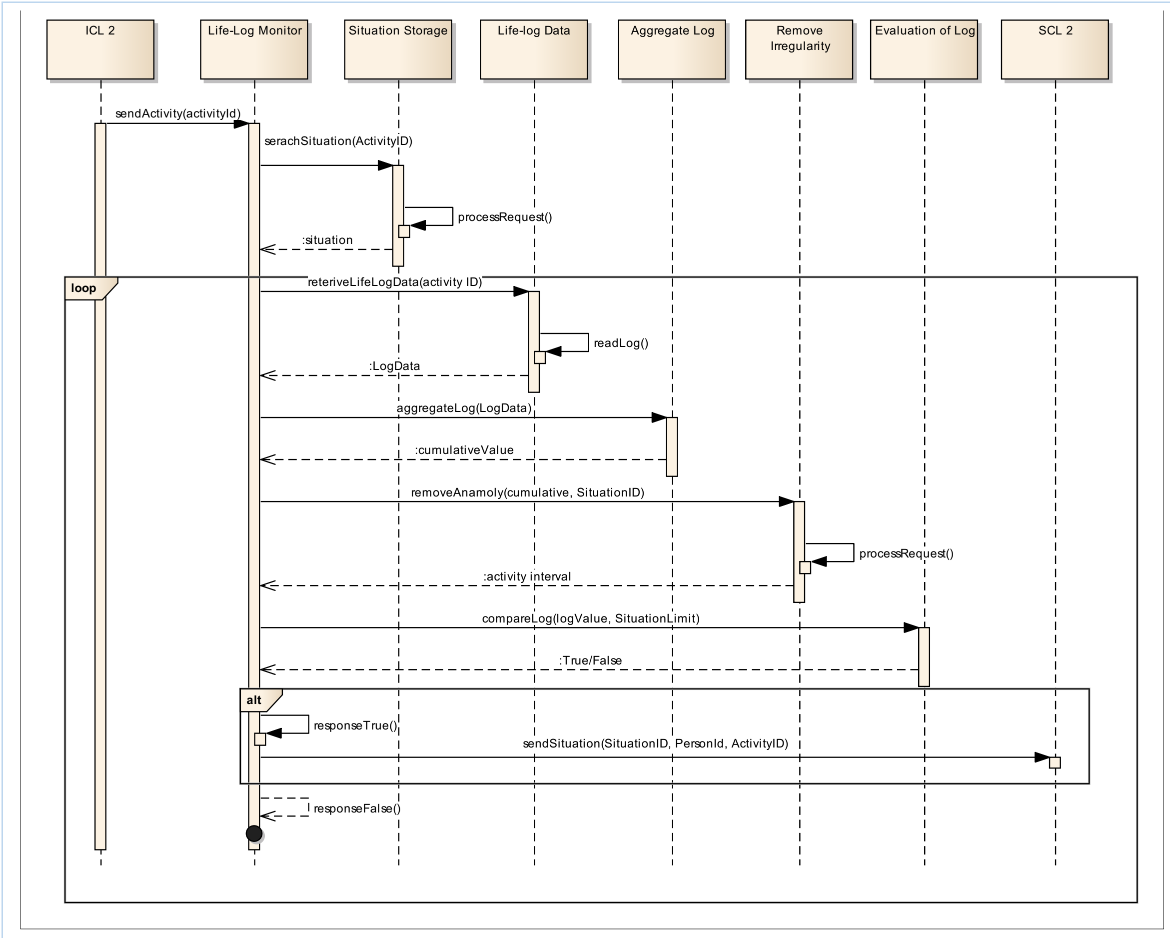
Use Case ID:	*DCL2-SUC-11*		
Use Case Name:	LLM Configuration for target Variables		
Created By:	Bilal Ali	**Last Updated By:**	Bilal Ali
Date Created:	15 July 2015	**Last Revision Date:**	15 July 2015
Actors:	KCL-2, Experts		
Description:	Configure the Life-log monitor for the screening of the target 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 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:	



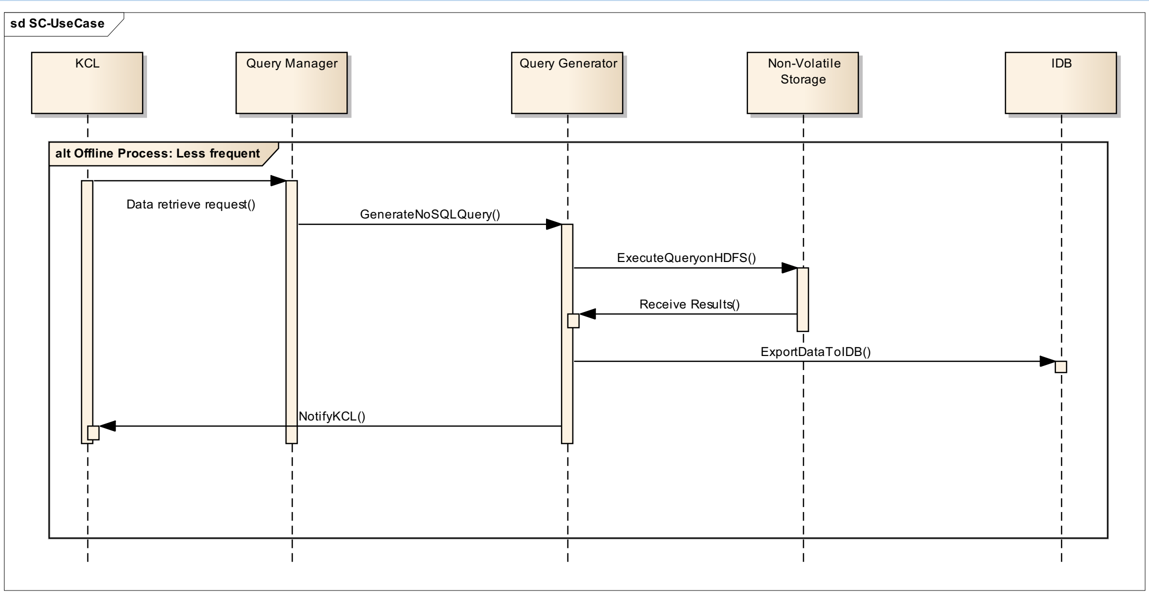
Use Case ID:	*DCL2-SUC-12*		
Use Case Name:	LLM for Situation Detection		
Created By:	Bilal Ali	**Last Updated By:**	Bilal Ali
Date Created:	15 July 2015	**Last Revision Date:**	15 July 2015

Actors:	Life-log Data, SCL-2, ICL 2
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 with situation and user.
Normal Flow:	1. ICL 2 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 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.
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:	



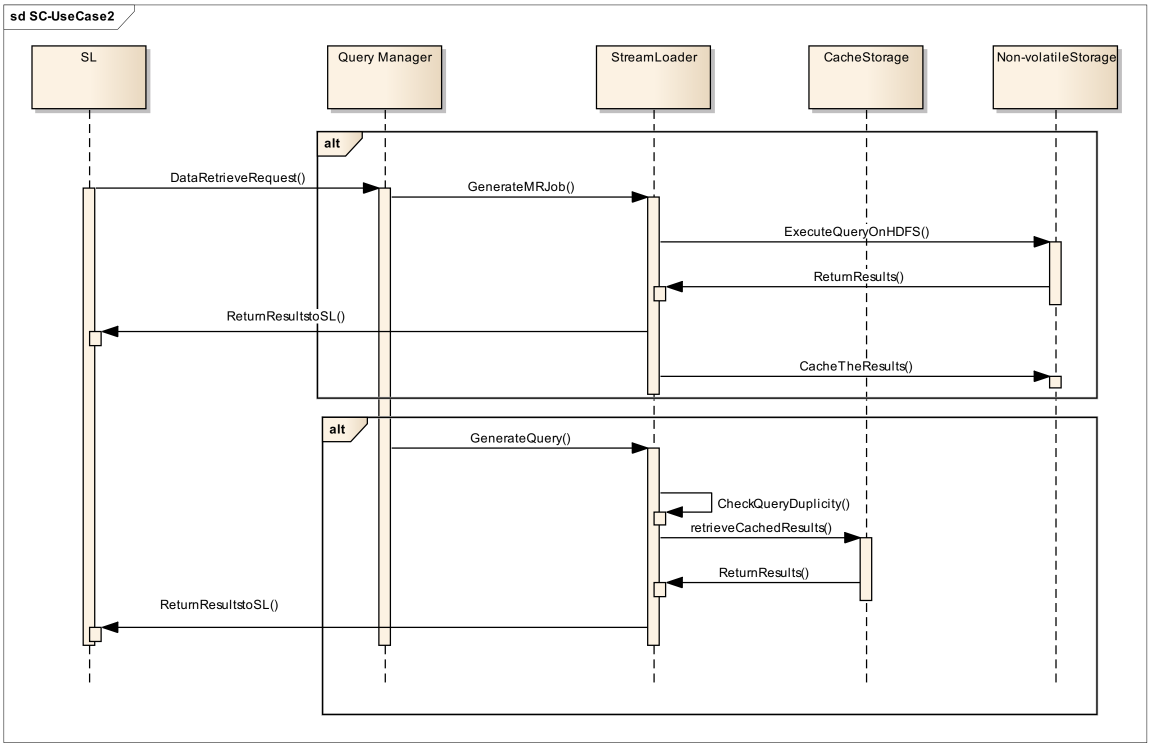
Use Case ID:	*DCL2-SUC-13*		
Use Case Name:	Retrieve sensory data from non-volatile storage for intermediate data generation (offline)		
Created By:	Muhammad Idris	**Last Updated By:**	Bilal Amin
Date Created:	15 July 2015	**Last Revision Date:**	20 July 2015
Actors:	KCL		
Description:	Raw data in HDFS is retrieved based on the request from KCL and converted to relational format and stored in IDB.		
Trigger:	KCL 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 is informed
Normal Flow:	1. SC receives requests from KCL and creates a NoSQL query 2. NoSQL query is executed using Apache Hive on HDFS to retrieve the data 3. Retrieved data is processed and transformed to relational format based on IDB schema 4. Transformed data is exported to IDB
Alternative Flows:	NA
Exceptions:	NA
Includes:	NA
Frequency of Use:	Less frequent, offline process, may be executed once or on change in the IDB schema
Assumptions:	NoSQL query results are easy to transform to relational format
Notes and Issues:	NA
Sequence Diagram:	



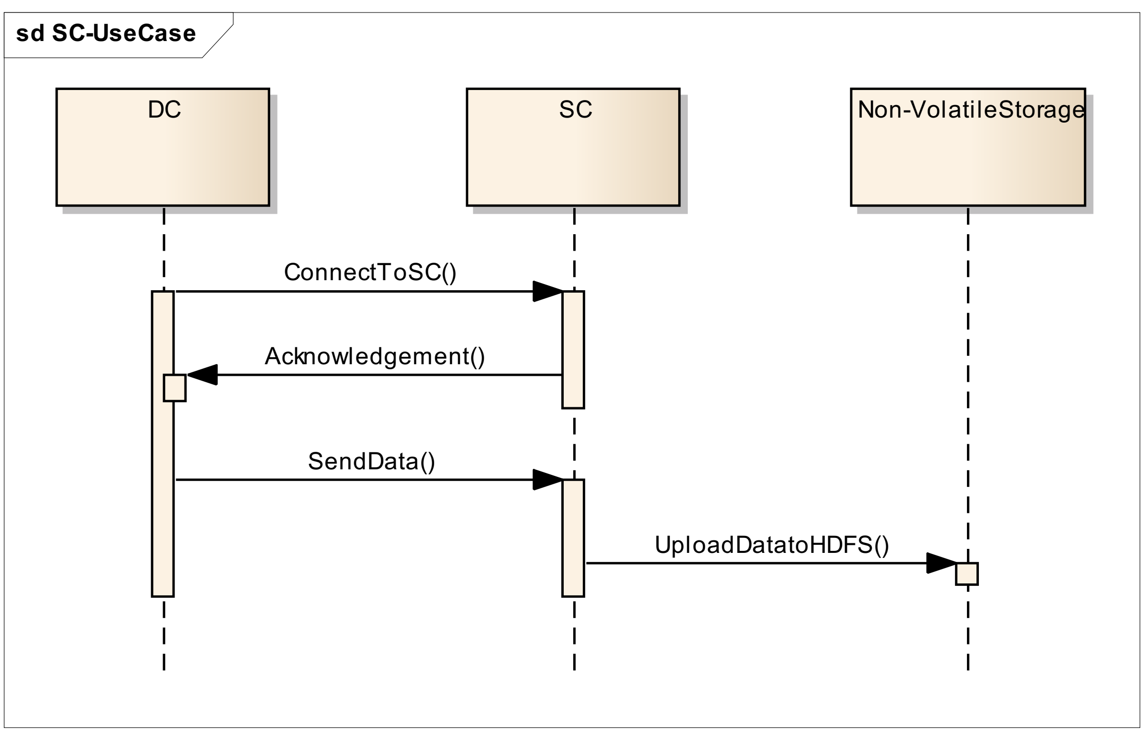
Use Case ID:	*DCL2-SUC-14*		
Use Case Name:	Retrieve sensory data from non-volatile storage (online)		
Created By:	Muhammad Idris	**Last Updated By:**	Bilal Amin
Date Created:	15 July 2015	**Last Revision Date:**	20 July 2015
Actors:	SL		
Description:	Raw sensory data in HDFS is retrieved and provided to analytics component of SL in an online process		
Trigger:	Request for data from descriptive analytics is received		
Pre-conditions:	Raw sensory exists and persisted in HDFS (non-volatile storage)		
Post-conditions:	1. Required data is retrieved from HDFS 2. Data is cached locally in SC 3. Required data is communicated to descriptive analytics in SL directly (online)		
Normal Flow:	1. Descriptive analytics in SL requests data from SC 2. SC receives the request and maintains its log. 3. SC transforms request to a MapReduce Job. 4. MapReduce job is executed on HDFS and results are retrieved. 5. Retrieved results are cached locally in volatile storage 6. Results are forwarded directly to descriptive analytics		
Alternative Flows:	1. Descriptive analytics requests data from SC 2. SC receives the request and examines it for duplicity(a request coming more than one time) 3. Duplicate request results are directly forwarded from the cached results 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 (SC)
Notes and Issues:	NA
Sequence Diagram:	



Use Case ID:	*DCL2-SUC-15*		
Use Case Name:	Persist sensory data in non-volatile storage		
Created By:	Muhammad Idris	**Last Updated By:**	Bilal Amin
Date Created:	15 July 2015	**Last Revision Date:**	20 July 2015
Actors:	DC		
Description:	Receive and persist raw sensory data from DC in to HDFS		
Trigger:	DC request to upload data, every 3 seconds		
Pre-conditions:	1. Data storage structure, directory structure in HDFS defined 2. File formats and data formats in HDFS known		

	3. Raw sensory data is received from DC 4. Big Data server is already running
Post-conditions:	1. Raw sensory 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 DC 2. DC connects and sends data to SC 3. SC uploads received data to HDFS
Alternative Flows:	NA
Exceptions:	NA
Includes:	NA
Frequency of Use:	Very frequent, every 3 seconds
Assumptions:	Data format and specifications already defined between DC and SC
Notes and Issues:	NA
Sequence Diagram:	



Information Curation Layer (ICL Ver. 2)

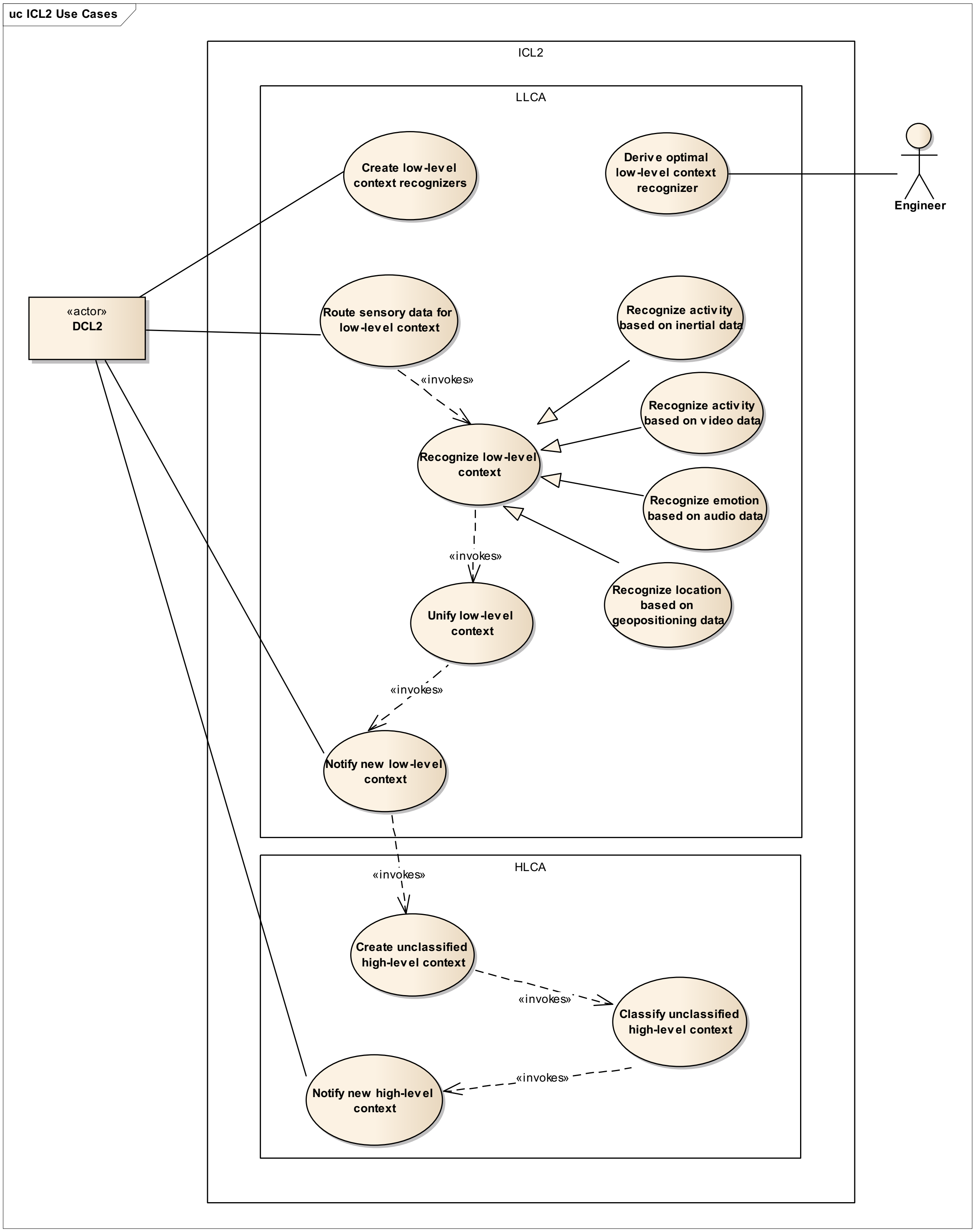
System Level Use cases

This section presents the system specifications of the Information Curation Layer for Mining Minds Version 2. The list of specifications presented in this document may be updated along with the design of the system, as we will follow an iterative approach.

List of Use cases

Use case ID#	Name
ICL2-SUC-01	Derive optimal low-level context recognizer
ICL2-SUC-02	Create low-level context recognizers
ICL2-SUC-03	Route sensory data for the low-level context identification
ICL2-SUC-04	Recognize user low-level context
ICL2-SUC-05	Recognize user activity based on inertial raw sensory data
ICL2-SUC-06	Recognize user activity based on video raw sensory data
ICL2-SUC-07	Recognize user location based on geopositioning raw sensory data
ICL2-SUC-08	Recognize user emotion based on audio raw sensory data
ICL2-SUC-09	Unify low-level contexts
ICL2-SUC-10	Notify new low-level context
ICL2-SUC-11	Create unclassified high-level context instance
ICL2-SUC-12	Classify high-level context instance
ICL2-SUC-13	Notify new high-level context

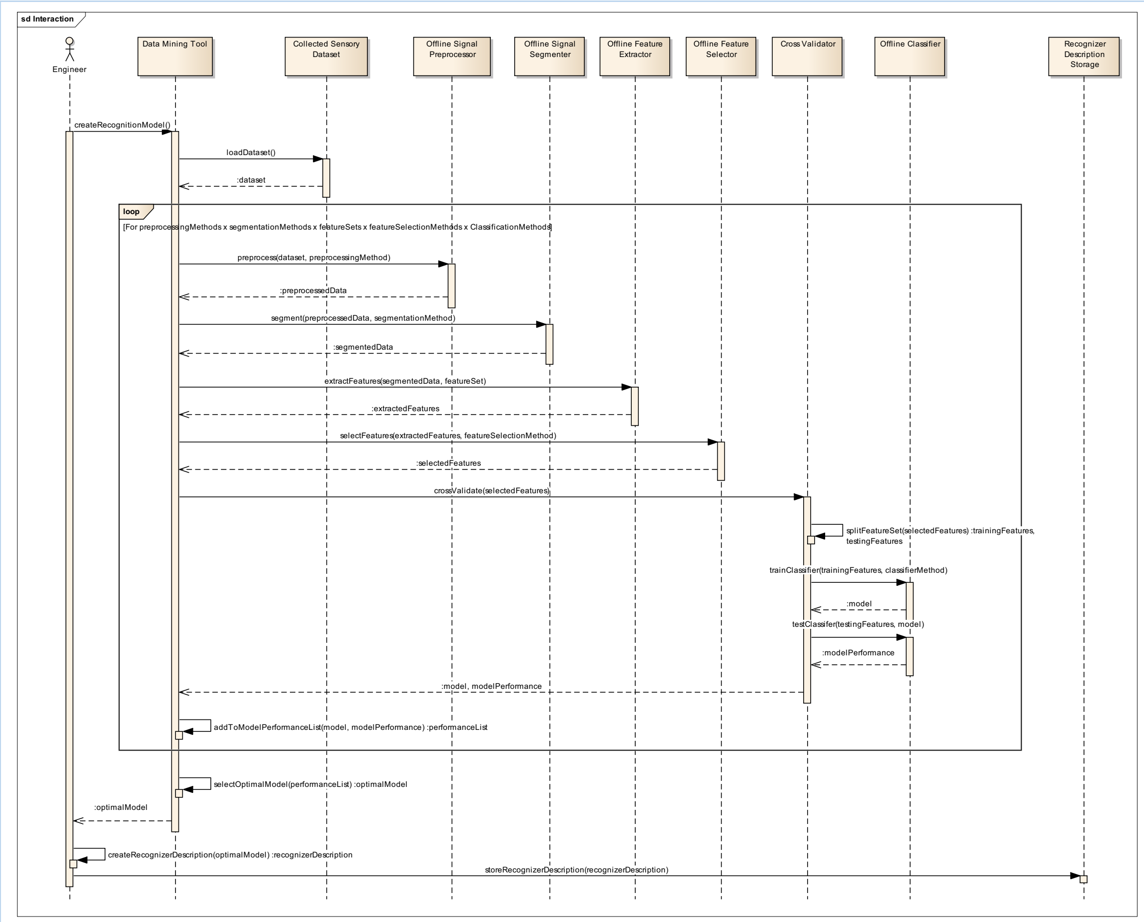
Use case Diagram



Use case Description

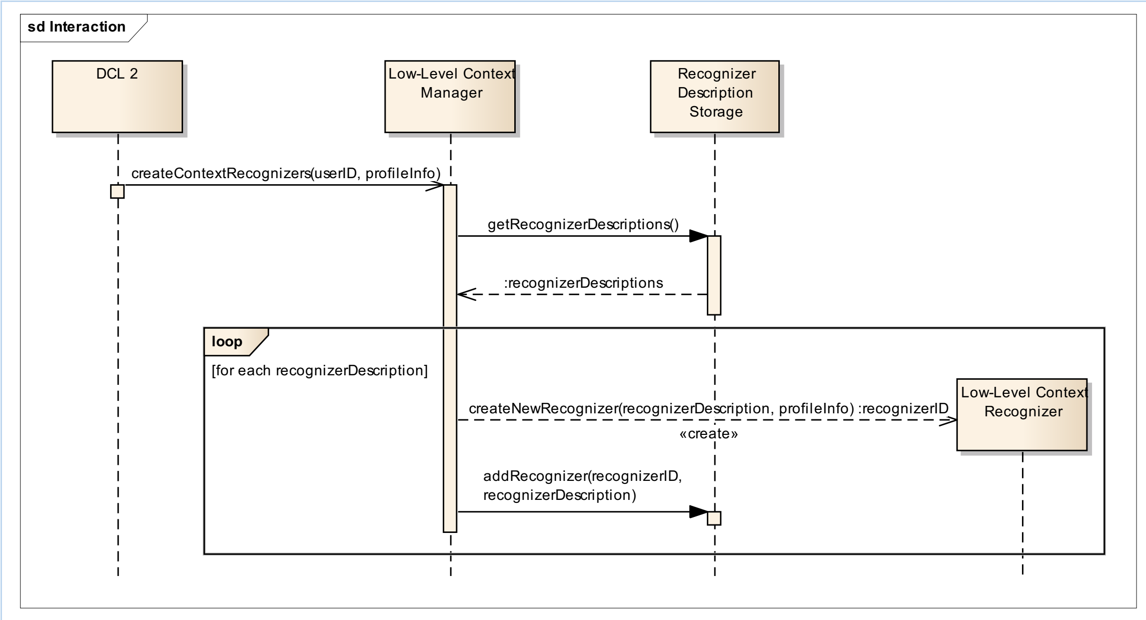
Use Case ID:	*ICL2-SUC-01*		
Use Case Name:	Derive optimal low-level context recognizer		
FR ID:	MM-FR-11		
Created By:	Oresti Banos	**Last Updated By:**	Oresti Banos
Date Created:	22 July 2015	**Last Revision Date:**	24 July 2015
Actors:	Engineer		
Description:	Create an optimal recognition model through 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:	



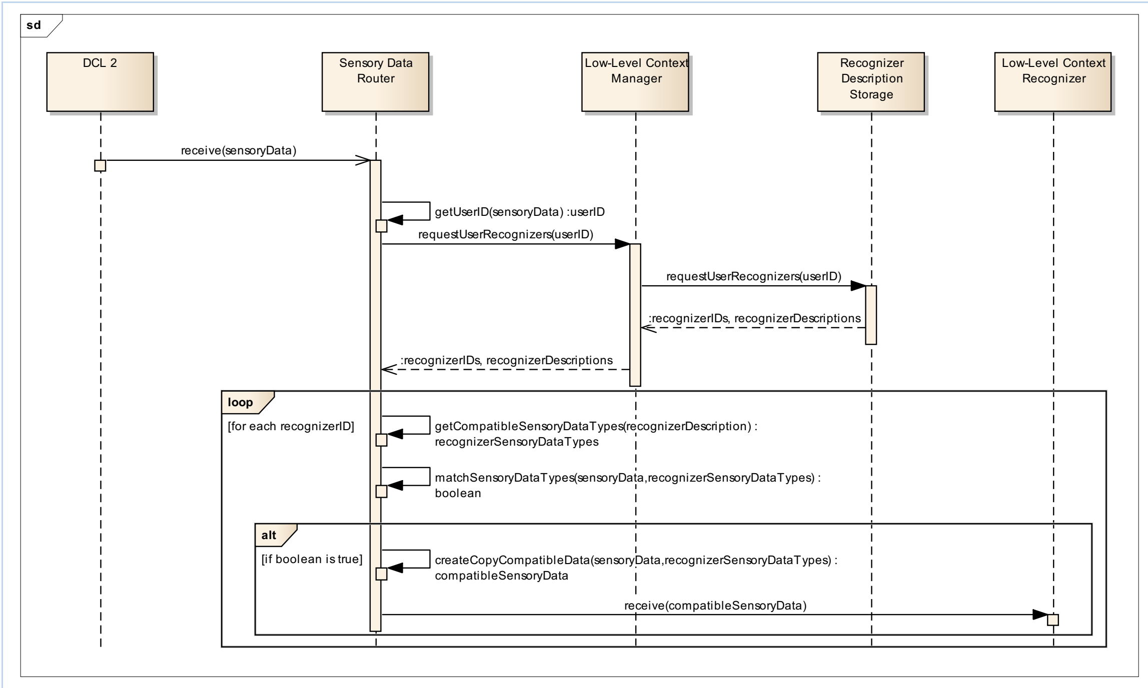
Use Case ID:	*ICL2-SUC-02*		
Use Case Name:	Create low-level context recognizers		
FR ID:	MM-FR-11		
Created By:	Oresti Banos	**Last Updated By:**	Oresti Banos
Date Created:	22 July 2015	**Last Revision Date:**	24 July 2015
Actors:	DCL 2		
Description:	Sensory data is received from DCL 2 and it is distributed 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 to ICL2		
Pre-conditions:	DCL 2 sends the userID and part of the user profile information to ICL 2 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 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:	



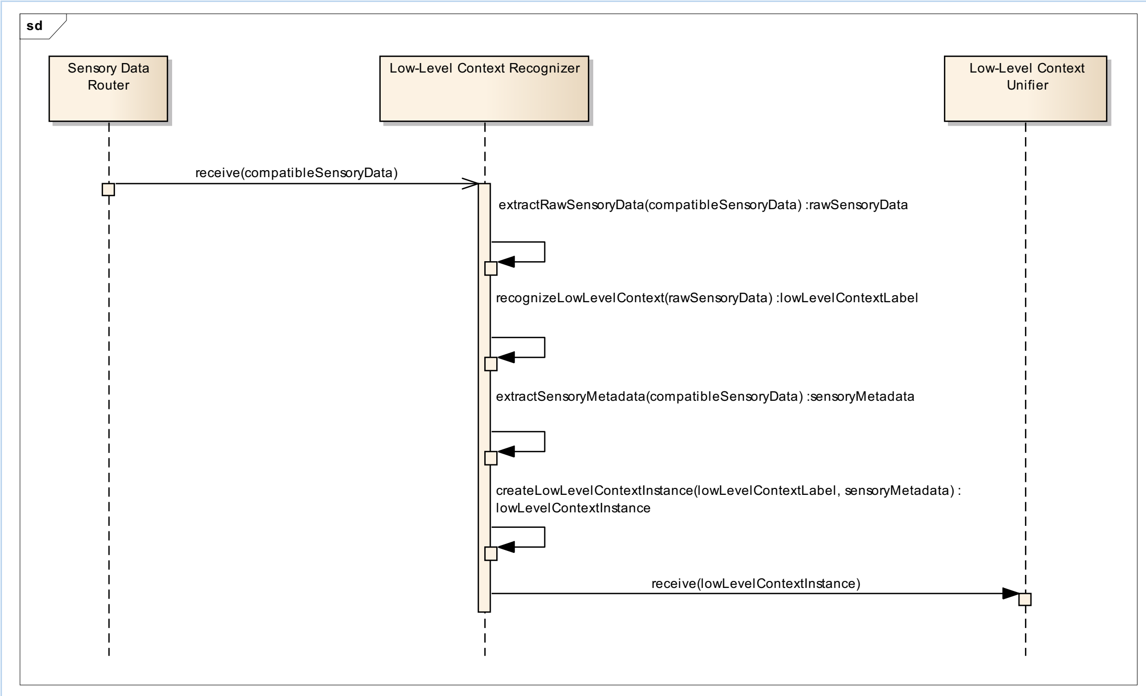
Use Case ID:	*ICL2-SUC-03*		
Use Case Name:	Route sensory data for the low-level context identification		
FR ID:	MM-FR-10		
Created By:	Oresti Banos	**Last Updated By:**	Oresti Banos
Date Created:	14 July 2015	**Last Revision Date:**	24 July 2015
Actors:	DCL 2, Low-Level Context Recognizer		
Description:	Sensory data is received from DCL 2 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 to ICL2		
Pre-conditions:	DCL 2 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
NFR ID:	NA
Assumptions:	• There is an established communication between DCL 2 and the Sensory Data Router • The communication channel between the DCL 2 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:	



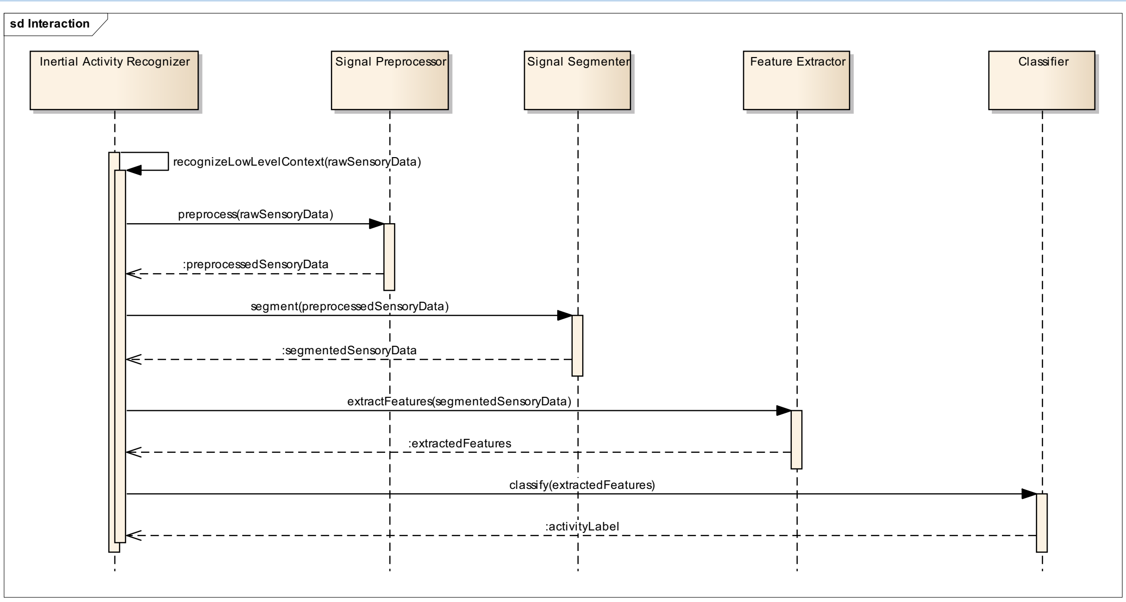
Use Case ID:	*ICL2-SUC-04*		
Use Case Name:	Recognize user low-level context		
FR ID:	MM-FR-11		
Created By:	Oresti Banos	**Last Updated By:**	Oresti Banos
Date Created:	14 July 2015	**Last Revision Date:**	20 July 2015
Actors:	Sensory Data Router, Low-Level Context Unifier		
Description:	The low-level 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:	



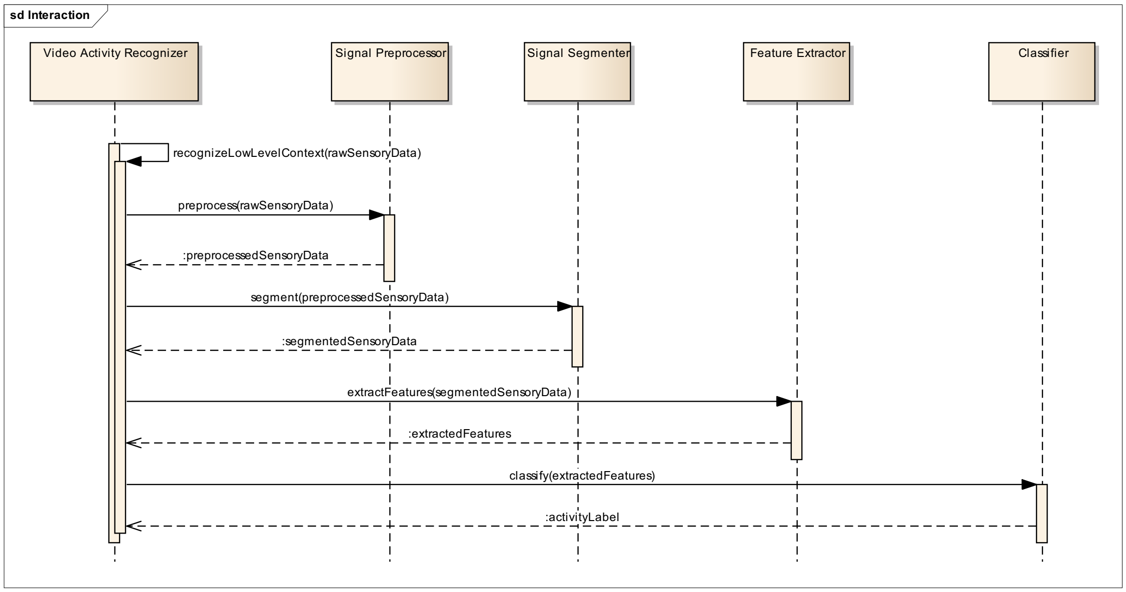
Use Case ID:	*ICL2-SUC-05*		
Use Case Name:	Recognize user activity based on inertial raw sensory data		
FR ID:	MM-FR-11		
Created By:	Oresti Banos	**Last Updated By:**	Oresti Banos
Date Created:	14 July 2015	**Last Revision Date:**	20 July 2015
Actors:	ICL2-SUC-04		
Description:	Identification of the 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:	



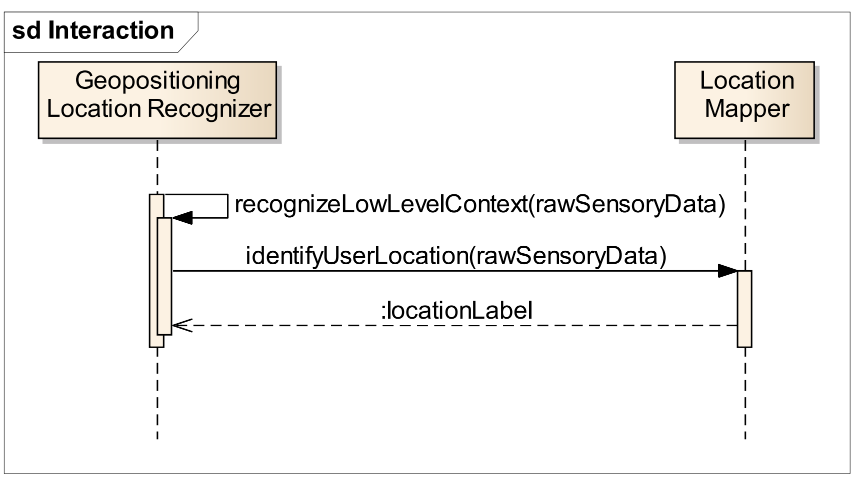
Use Case ID:	*ICL2-SUC-06*		
Use Case Name:	Recognize user activity based on video raw sensory data		
FR ID:	MM-FR-11		
Created By:	Oresti Banos	**Last Updated By:**	Oresti Banos
Date Created:	17 July 2015	**Last Revision Date:**	20 July 2015
Actors:	ICL2-SUC-04		
Description:	Identification of the user 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:	



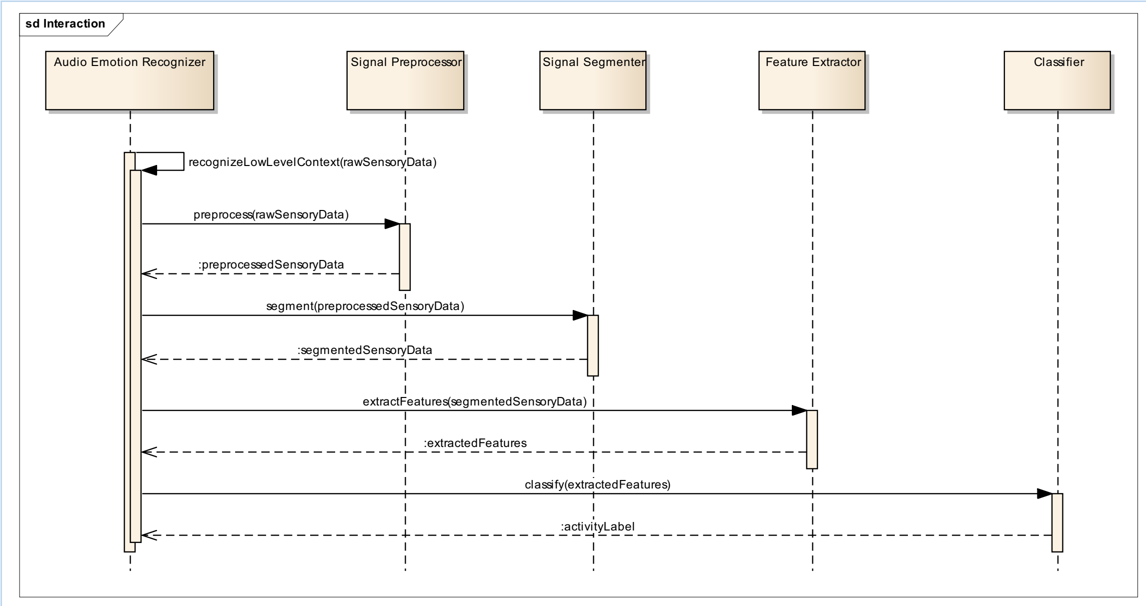
Use Case ID:	*ICL2-SUC-07*		
Use Case Name:	Recognize user location based on geopositioning raw sensory data		
FR ID:	MM-FR-11		
Created By:	Oresti Banos	**Last Updated By:**	Oresti Banos
Date Created:	17 July 2015	**Last Revision Date:**	20 July 2015
Actors:	ICL2-SUC-04		
Description:	Identification 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:	



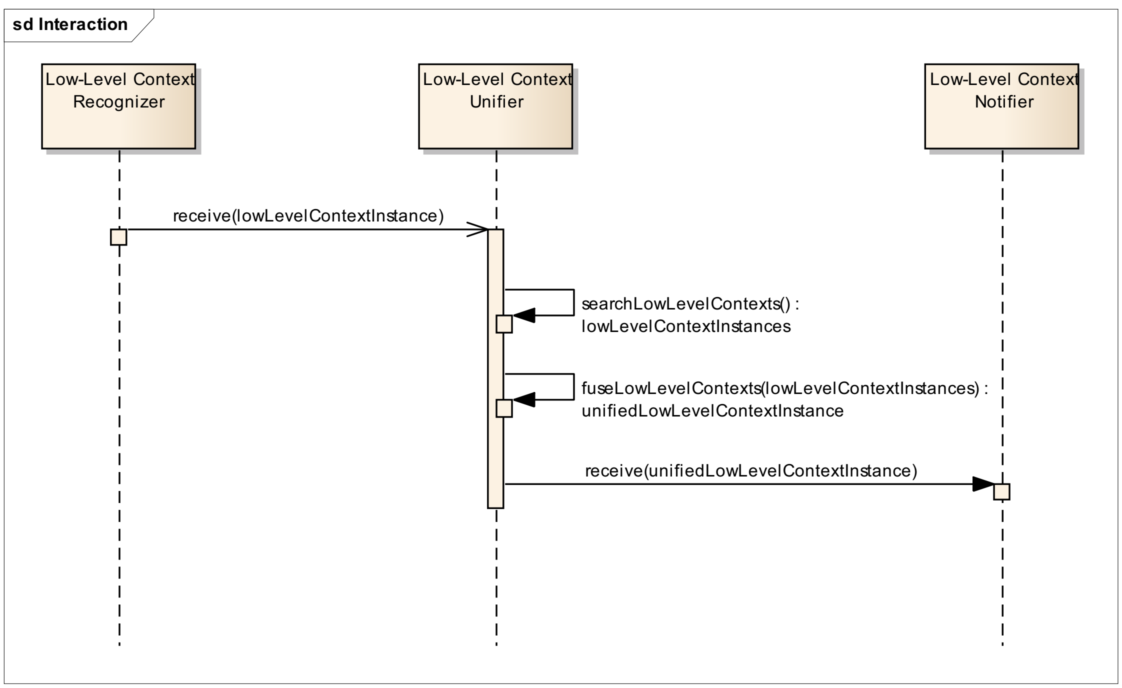
Use Case ID:	*ICL2-SUC-08*
Use Case Name:	Recognize user emotion based on audio raw sensory data
FR ID:	MM-FR-11

Created By:	Oresti Banos	**Last Updated By:**	Oresti Banos
Date Created:	17 July 2015	**Last Revision Date:**	20 July 2015
Actors:	ICL2-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:			



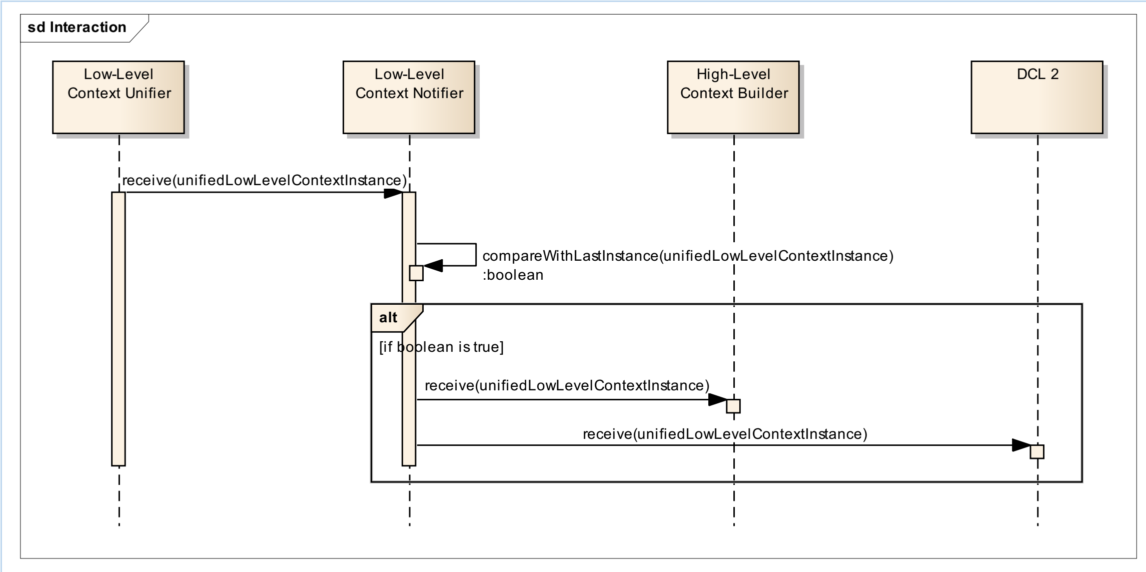
Use Case ID:	*ICL2-SUC-09*		
Use Case Name:	Unify low-level contexts		
FR ID:	MM-FR-11		
Created By:	Oresti Banos	**Last Updated By:**	Oresti Banos
Date Created:	20 July 2015	**Last Revision Date:**	20 July 2015
Actors:	Low-Level Context Recognizer, Low-Level Context Notifier		
Description:	Aggregation 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:	



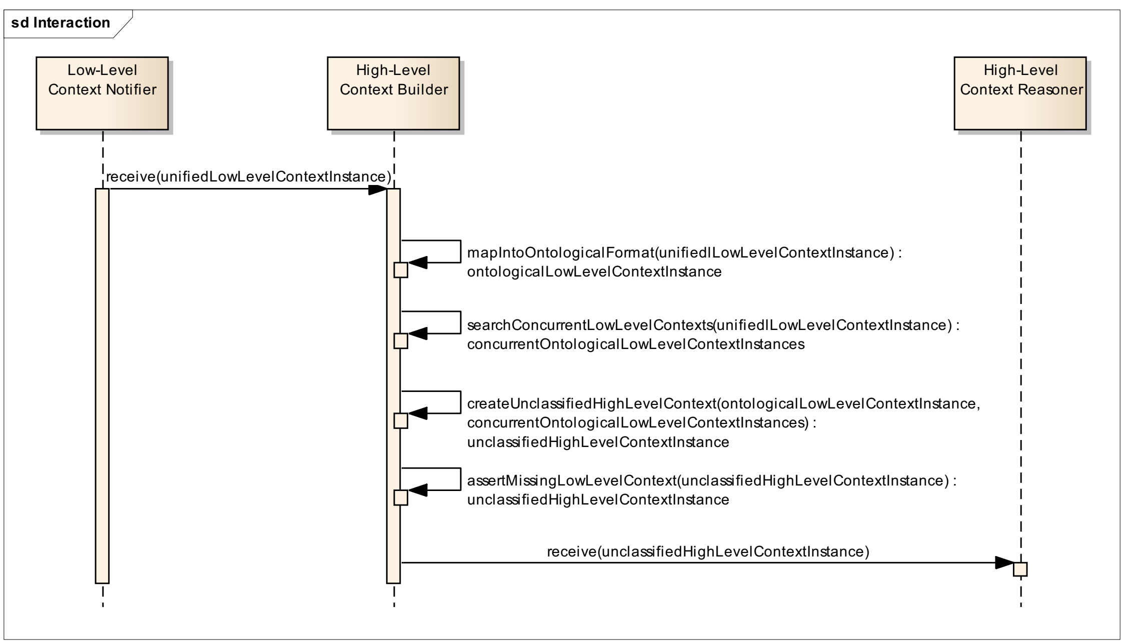
Use Case ID:	*ICL2-SUC-10*		
Use Case Name:	Notify new low-level context		
FR ID:	MM-FR-13		
Created By:	Oresti Banos	**Last Updated By:**	Oresti Banos

Date Created:	14 July 2015	**Last Revision Date:**	20 July 2015
Actors:	Low-Level Context Unifier, High-Level Context Builder, DCL 2		
Description:	Serve the newly recognized low-level context for the identification of high-level context and also communicate it to DCL 2 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		
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		
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:			



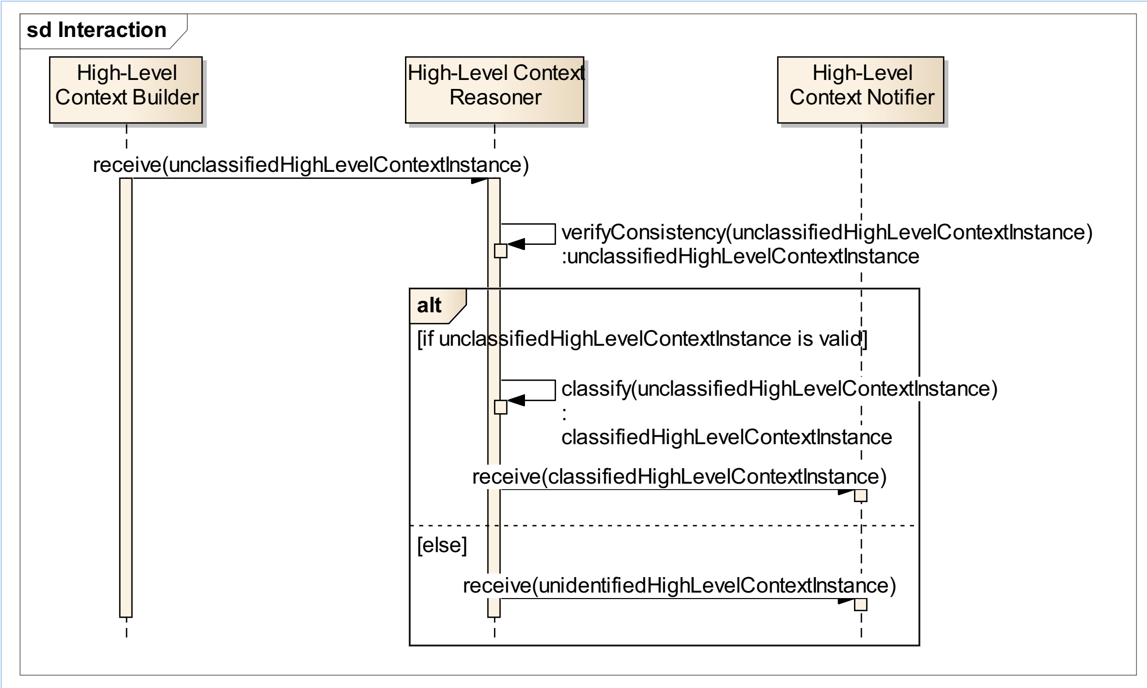
Use Case ID:	*ICL2-SUC-11*		
Use Case Name:	Create unclassified high-level context instance		
FR ID:	MM-FR-12		
Created By:	Claudia Villalonga	**Last Updated By:**	Oresti Banos
Date Created:	14 July 2015	**Last Revision Date:**	20 July 2015
Actors:	Low-Level Context Notifier, High-Level Context Reasoner		
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:	



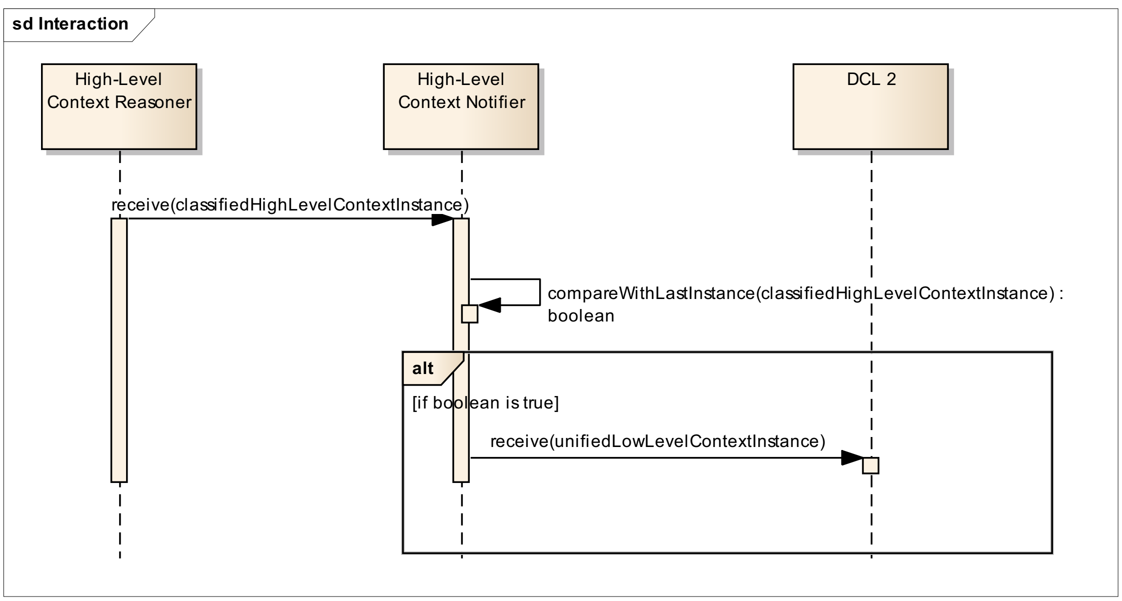
Use Case ID:	*ICL2-SUC-12*		
Use Case Name:	Classify high-level context instance		
FR ID:	MM-FR-12		
Created By:	Claudia Villalonga	**Last Updated By:**	Oresti Banos

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-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		
Description:	Communicate the newly recognized high-level context to DCL 2 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		
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		

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:	



Knowledge Curation Layer (KCL Ver. 2)

System Level Use cases

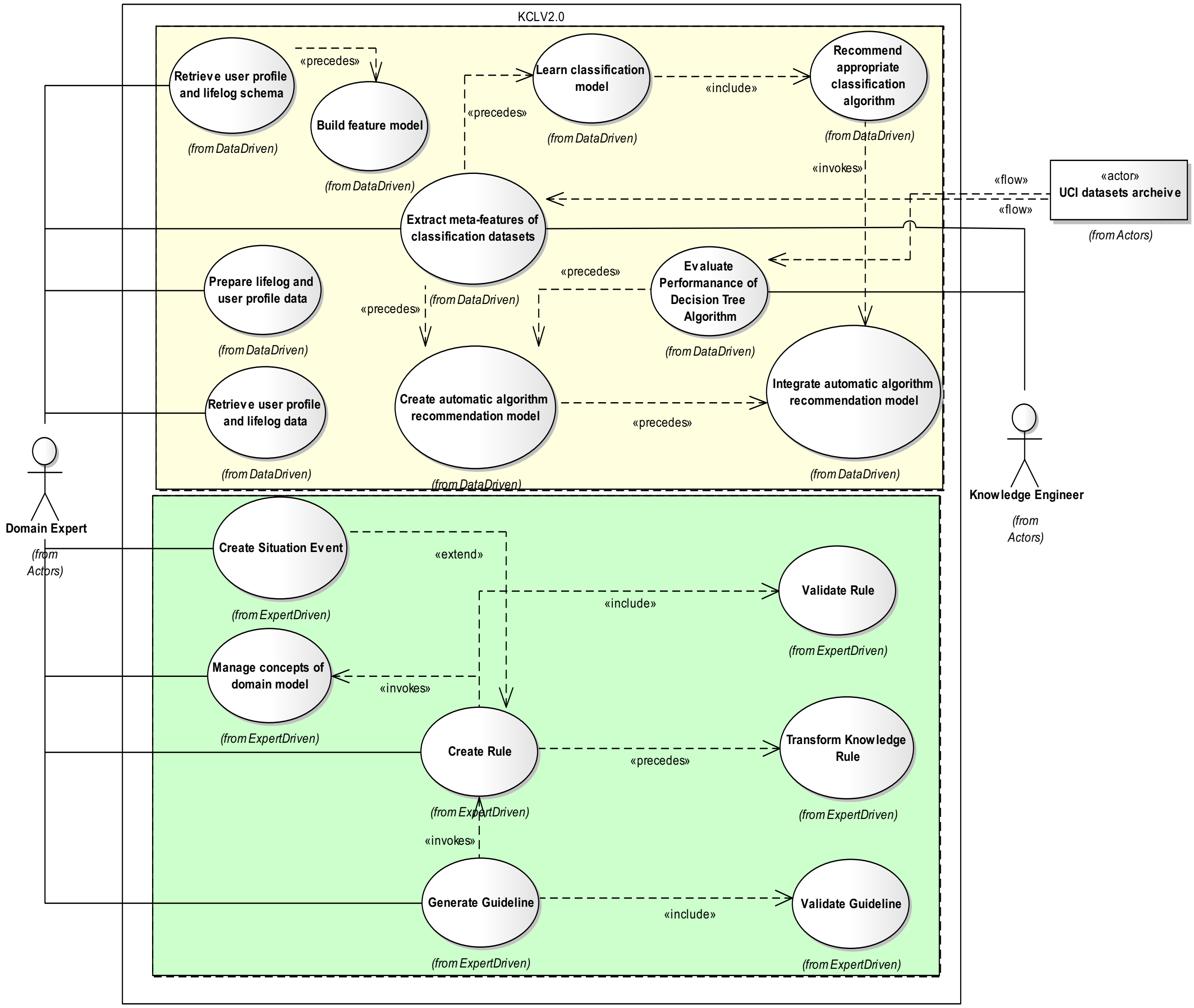
List of Use cases

Use case ID#	Name
KCL2-SUC-01	Select valid combinations of features from lifelog and user

	profile schema to build feature model for yielding correct classification model.
KCL2-SUC-02	Apply preprocessing methods on retrieved lifelog and user profile data to prepare the data for classification model learning.
KCL2-SUC-03	Expert generate guidelines to utilize their practices to create rules in the knowledge bases.
KCL2-SUC-04	System validate the guidelines in tree structure to maintain the rules.
KCL2-SUC-05	User profile and lifelog schema is needed to be known before feature modeling and creation of classification model.
KCL2-SUC-06	Retrieve user profile and lifelog data for creation of classification model.
KCL2-SUC-07	Extract meta-features of classification datasets.
KCL2-SUC-08	Evaluate performance of decision tree algorithms (i.e, f-measure)
KCL2-SUC-09	Create automatic algorithm recommendation model (AARM) from offline datasets. AARM will be used as recommendation model for algorithm selection.
KCL2-SUC-10	Create rules to enhance the knowledge base of the system to generate recommendations in easy manner.
KCL2-SUC-11	Rule validation avoid the duplication of rules in the knowledge base and enhance the maintainability of knowledge base.
KCL2-SUC-12	It integrates AARM dataset in Mining Minds Data Driven knowledge acquisition approach for recommendation of automatic algorithm on given dataset.
KCL2-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-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-SUC-15	Domain model is used in creation of rule. It manages the domain model for creating rule.
KCL2-SUC-16	It transforms the rules or guidelines into executable knowledge representation.
KCL2-SUC-17	It creates situation event and index the rule based on situation event.

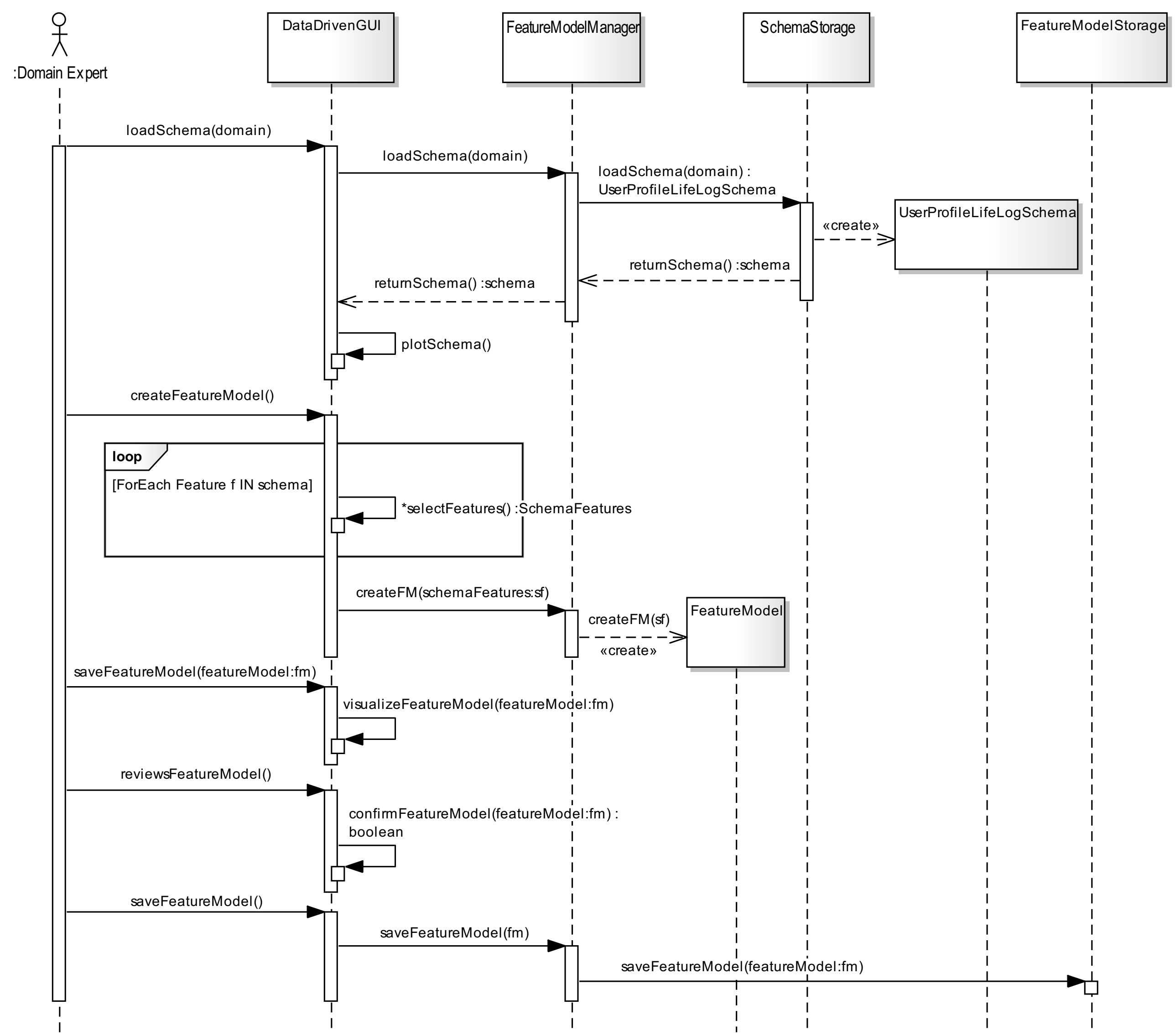
Use case Diagram



Use case Description

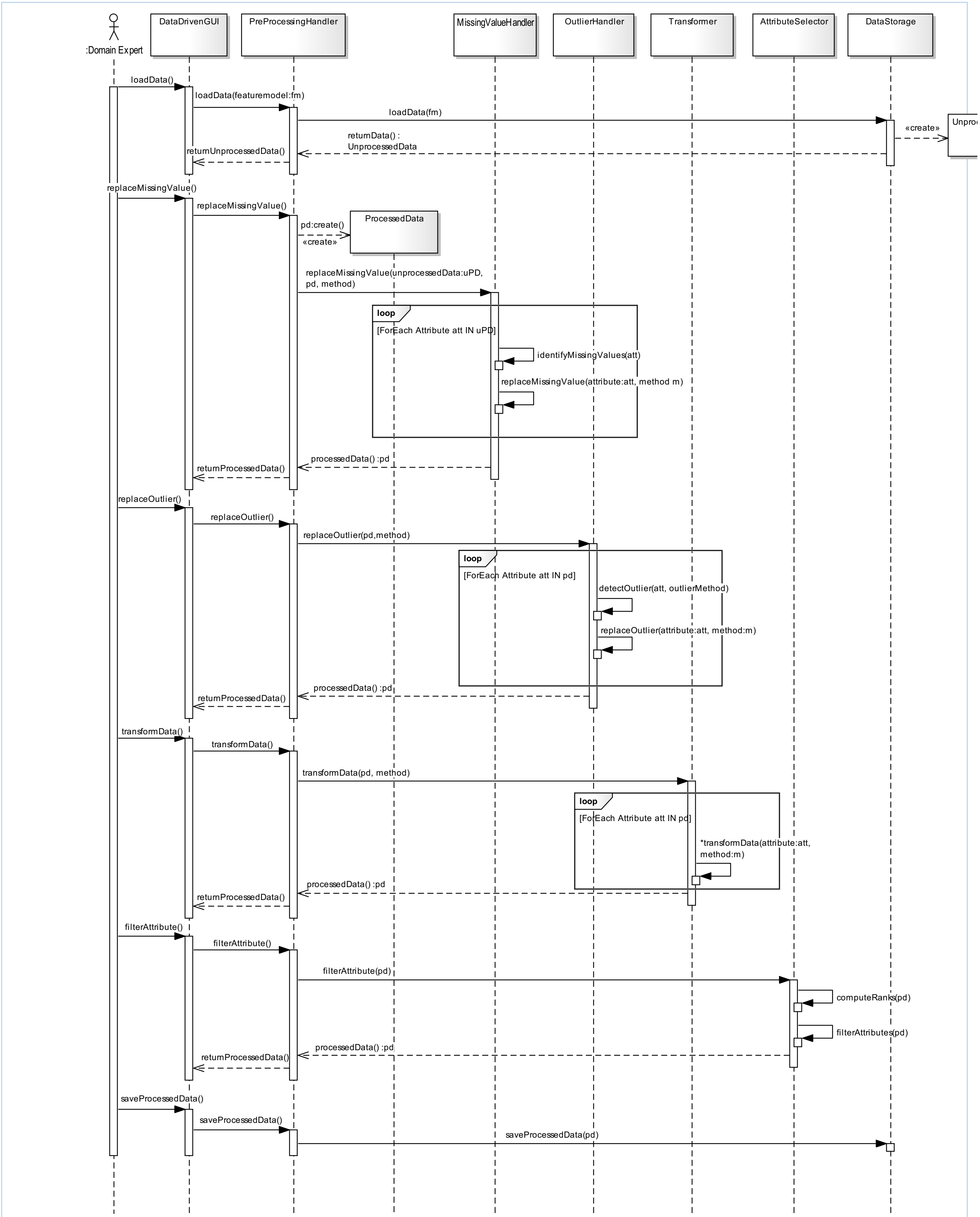
Use Case ID:	*KCL2-SUC-01*		
Use Case Name:	Build feature model		
Created By:	Maqbool Ali	**Last Updated By:**	Maqbool Hussain
Date Created:	11 July 2015	**Last Revision Date:**	27 July 2015
Actors:	Domain Expert		
Description:	A feature model defines the valid combinations of features in a domain that 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. • 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:	



Use Case ID:	*KCL2-SUC-02*		
Use Case Name:	Prepare lifelog and user profile data		
Created By:	Maqbool Ali	**Last Updated By:**	Maqbool Hussain
Date Created:	10 July 2015	**Last Revision Date:**	15 July 2015
Actors:	Domain Expert		
Description:	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.		
Trigger:	Prior to classification model creation needed for high accuracy of model learning		
Preconditions:	• System has retrieved the data from DCL2, which is unprocessed data.		
Postconditions:	System will prepare and store the data		
Normal Flow:	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		

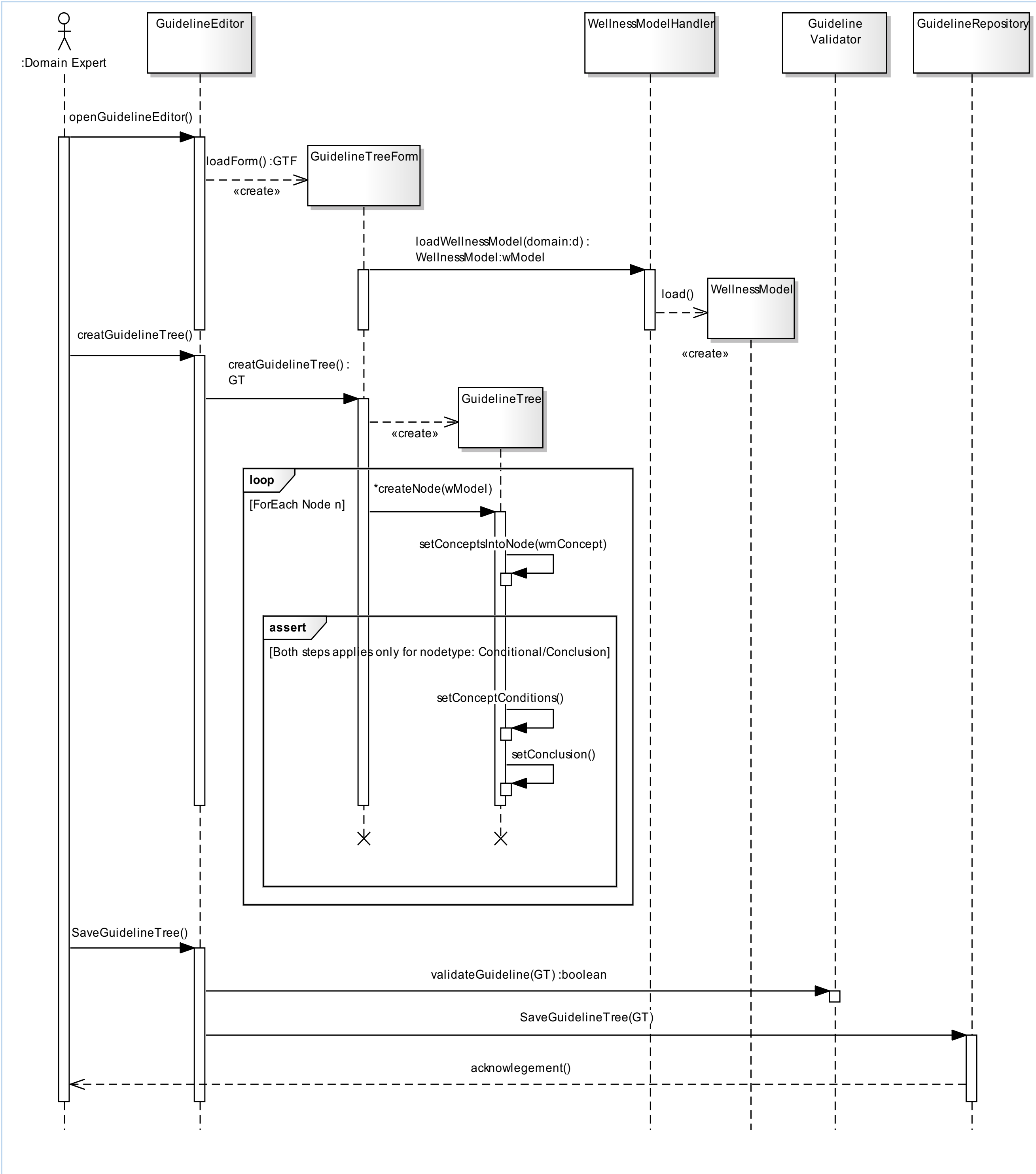
	d. System replace the outlier using selected method. 5. For each attribute: a. Domain expert identifies, normalizes the non-transformed values, and updates the dataset. b. System modifies the values set and update the dataset 6. Domain expert applies the attributes filtration techniques (i.e ranking) 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 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:	



:Domain Expert
DataDrivenGUI
PreProcessingHandler
MissingValueHandler
OutlierHandler
Transformer
AttributeSelector
DataStorage
loadData()
loadData(featuremodel:fm)
loadData(fm)
«create»
returnData() : UnprocessedData
returnUnprocessedData()
replaceMissingValue()
replaceMissingValue()
ProcessedData
pd:create()
«create»
replaceMissingValue(unprocessedData:uPD, pd, method)
loop
[ForEach Attribute att IN uPD]
identifyMissingValues(att)
replaceMissingValue(attribute:att, method m)
processedData() :pd
returnProcessedData()
replaceOutlier()
replaceOutlier()
replaceOutlier(pd,method)
loop
[ForEach Attribute att IN pd]
detectOutlier(att, outlierMethod)
replaceOutlier(attribute:att, method:m)
processedData() :pd
returnProcessedData()
transformData()
transformData()
transformData(pd, method)
loop
[ForEach Attribute att IN pd]
*transformData(attribute:att, method:m)
processedData() :pd
returnProcessedData()
filterAttribute()
filterAttribute()
filterAttribute(pd)
computeRanks(pd)
filterAttributes(pd)
processedData() :pd
returnProcessedData()
saveProcessedData()
saveProcessedData()
saveProcessedData(pd)

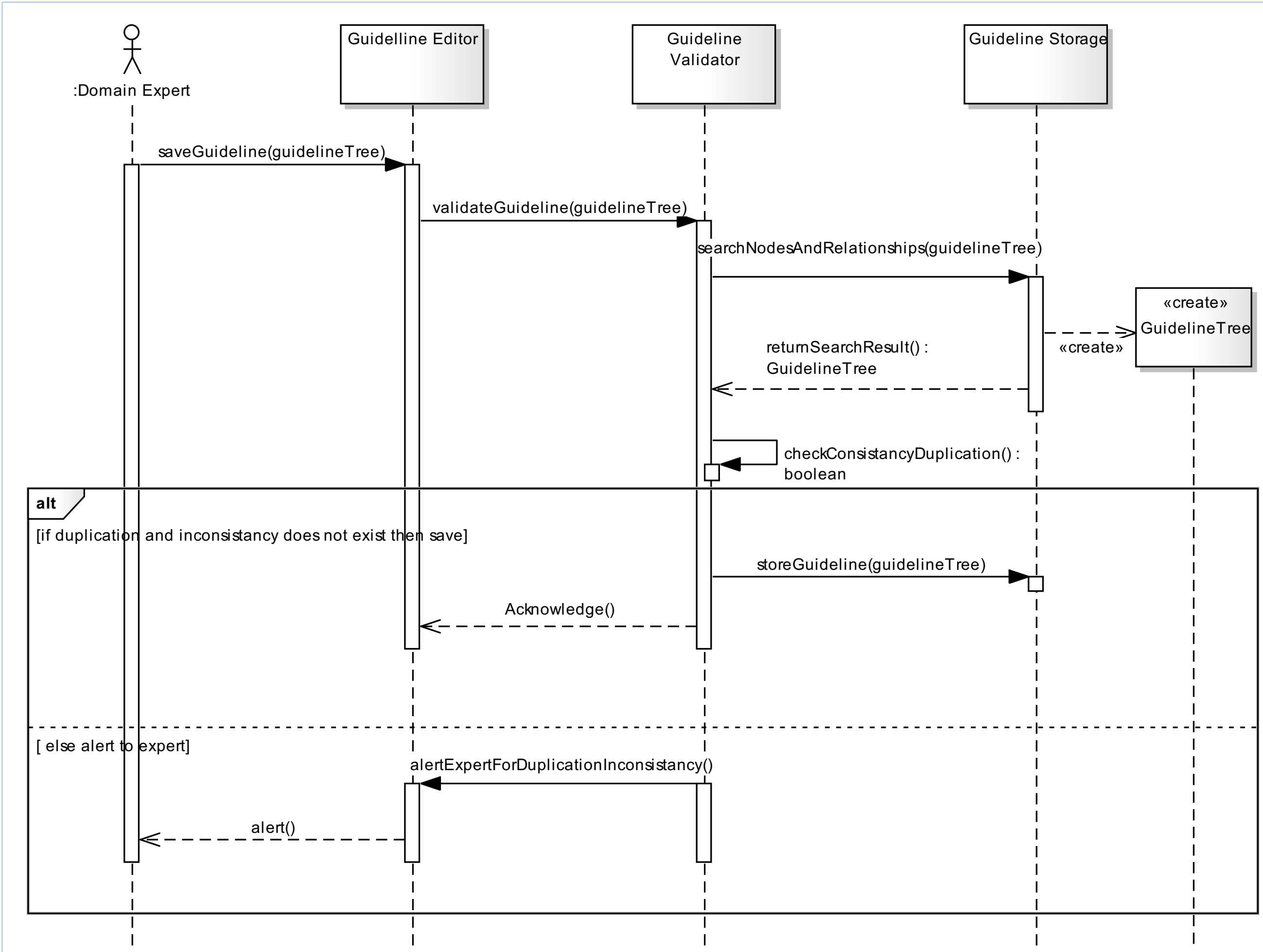
Use Case ID:	*KCL2-SUC-03*		
Use Case Name:	Generate Guideline		
Created By:	Taqdir Ali and Maqbool Hussain	**Last Updated By:**	Taqdir Ali, Maqbool Hussain
Date Created:	11 July 2015	**Last Revision Date:**	15 July 2015
Actors:	Domain Expert		
Description:	Guidelines are the combination of one or multiple 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.		
Trigger:	Whenever domain expert wants to generate new guideline or update the existing one.		
Preconditions:	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.		
Postconditions:	• The expert shall generate guidelines to acquire their knowledge into the system.		
Normal Flow:	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-SUC-04” and save into guideline repository.
Alternative Flows:	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.
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:	


:Domain Expert
GuidelineEditor
WellnessModelHandler
Guideline Validator
GuidelineRepository
openGuidelineEditor()
loadForm() :GTF
«create»
GuidelineTreeForm
loadWellnessModel(domain:d) : WellnessModel:wModel
load()
WellnessModel
«create»
creatGuidelineTree()
creatGuidelineTree() : GT
GuidelineTree
«create»
loop
[ForEach Node n]
*createNode(wModel)
setConceptsIntoNode(wmConcept)
assert
[Both steps applies only for nodetype: Conditional/Conclusion]
setConceptConditions()
setConclusion()
SaveGuidelineTree()
validateGuideline(GT) :boolean
SaveGuidelineTree(GT)
acknowlegement()

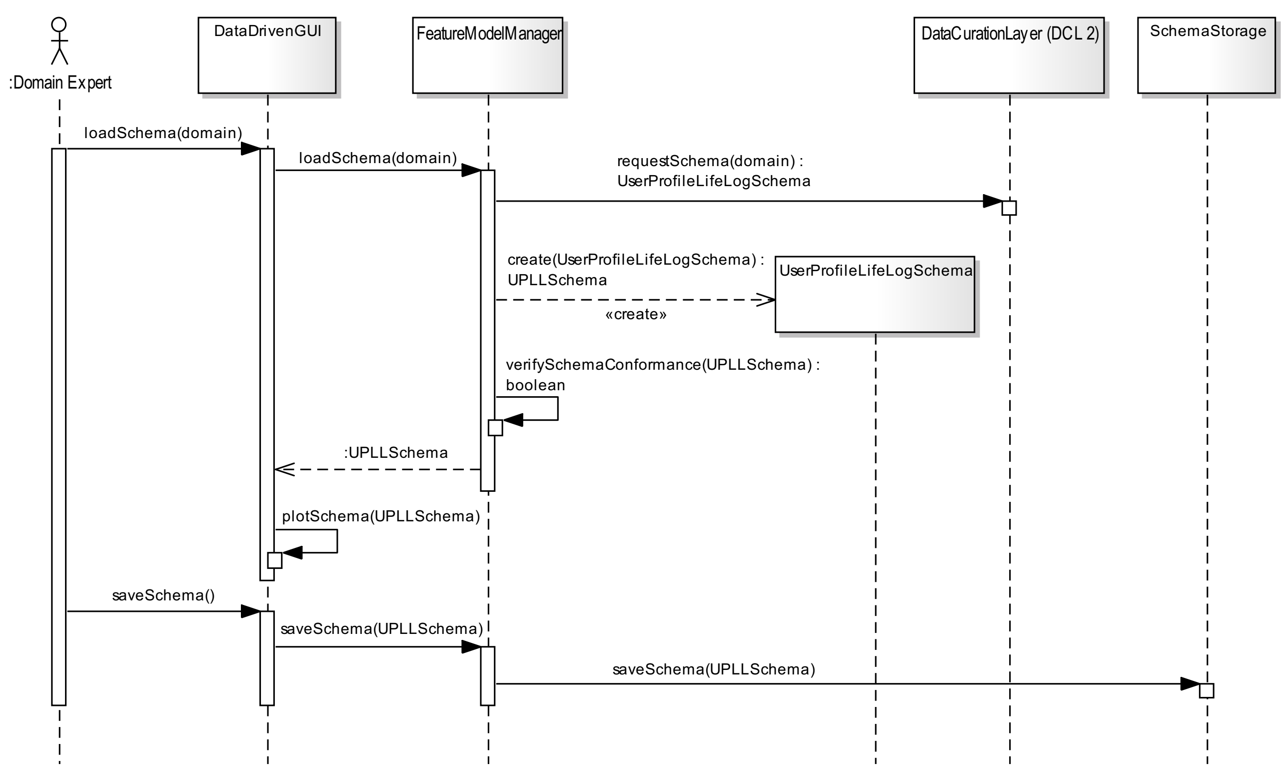
<table>
<tr><td>Use Case ID:</td><td colspan="3">KCL2-SUC-04</td></tr>
<tr><td>Use Case Name:</td><td colspan="3">Validate Guideline</td></tr>
<tr><td>Created By:</td><td>Taqdir Ali and Maqbool Hussain</td><td>Last Updated By:</td><td>Taqdir Ali, Maqbool Hussain</td></tr>
<tr><td>Date Created:</td><td>11 July 2015</td><td>Last Revision Date:</td><td>15 July 2015</td></tr>
<tr><td>Actors:</td><td colspan="3">Domain Expert</td></tr>
<tr><td>Description:</td><td colspan="3">Guidelines have different facts and conclusions in form of nodes related with different relationships. The guidelines tree shall be validated for the possible duplication.</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. System shall be running
2. The domain expert shall be authenticated with full access on the guideline management.</td></tr>
<tr><td>Postconditions:</td><td colspan="3">• Validated guideline tree</td></tr>
<tr><td>Normal Flow:</td><td colspan="3">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.</td></tr>
<tr><td>Alternative Flows:</td><td colspan="3">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</td></tr>
</table>

	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:	


:Domain Expert
Guidelline Editor
Guideline Validator
Guideline Storage
saveGuideline(guidelineTree)
validateGuideline(guidelineTree)
searchNodesAndRelationships(guidelineTree)
«create»
GuidelineTree
«create»
returnSearchResult() : GuidelineTree
checkConsistancyDuplication() : boolean
alt
[if duplication and inconsistancy does not exist then save]
storeGuideline(guidelineTree)
Acknowledge()
[else alert to expert]
alertExpertForDuplicationInconsistancy()
alert()

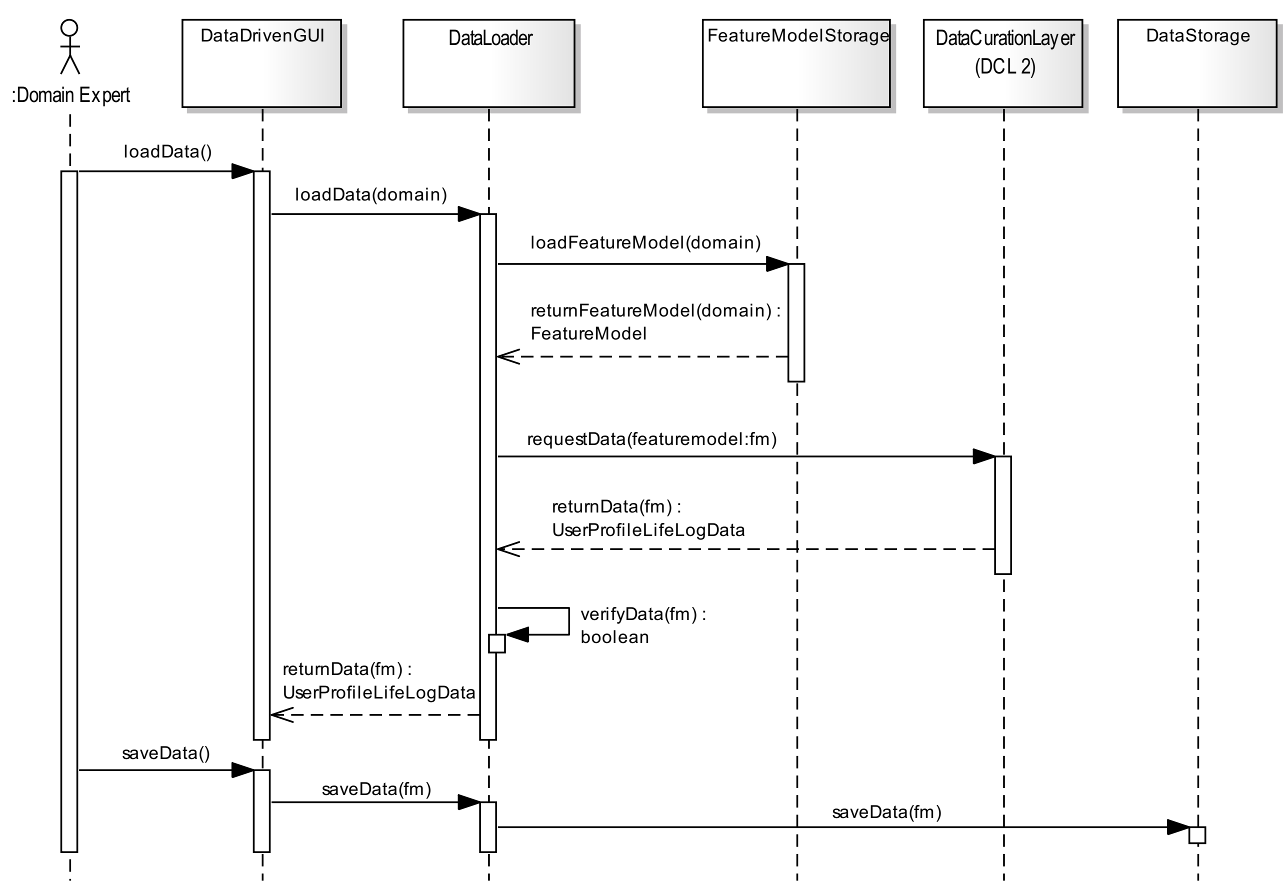
Use Case ID:	*KCL2-SUC-05*		
Use Case Name:	Retrieve user profile and lifelog schema		
Created By:	Maqbool Ali	**Last Updated By:**	Maqbool Hussain
Date Created:	10 July 2015	**Last Revision Date:**	27 July 2015
Actors:	Domain Expert, DCL 2		
Description:	User profile and lifelog schema retrieval help domain expert to view all available features for building feature model.		
Trigger:	Prior to classification model creation needed for required domain		
Preconditions:	• System has access through service interface to retrieve user profile and lifelog schema from DCL 2 • System and DCL 2 has agreement on common schema representation format • DCL 2 has capability to share user profile and lifelog schema in secure environment.		
Postconditions:	System will receive user profile and lifelog schema conform to its representation scheme.		
Normal Flow:	1. Domain expert selects the domain and sends requests to DCL 2 for user profile and lifelog schema. 2. DCL 2 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		
Alternative Flows:	N/A		
Exceptions:	1a. System unable to connect to DCL 2 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 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 b. System will send message to DCL 2 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 is unable to send user profile and lifelog schema in required format, then alternate strategy has to be considered.
Sequence Diagram:	



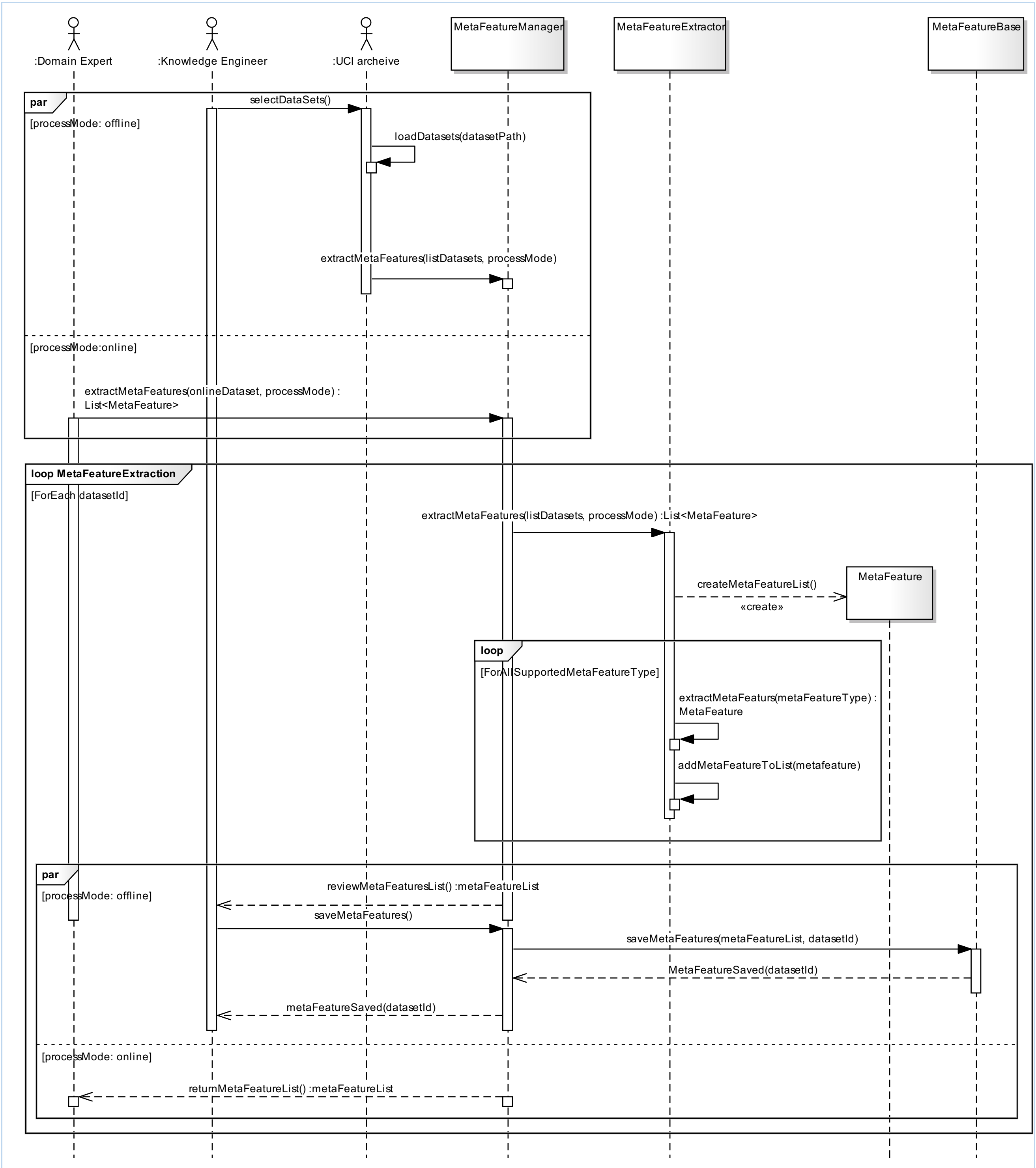
Use Case ID:	*KCL2-SUC-06*		
Use Case Name:	Retrieve user profile and lifelog data		
Created By:	Maqbool Ali	**Last Updated By:**	Maqbool Hussain
Date Created:	10 July 2015	**Last Revision Date:**	27 July 2015
Actors:	Domain Expert, DCL 2		
Description:	User profile and lifelog data has hidden knowledge that can be explored after loading from data storage		
Trigger:	Prior to classification model creation needed for required data		
Preconditions:	• System has access through service interface to retrieve user profile and lifelog data from DCL 2 • DCL 2 has capability to share user profile and lifelog data in secure environment. • System has already loaded the previously imported user profile and lifelog schema		
Postconditions:	System will receive user profile and lifelog data based on selected schema		
Normal Flow:	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 for user profile and lifelog data based on loaded feature model 4. DCL 2 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		
Alternative Flows:			
Exceptions:	3a. System unable to connect to DCL 2 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 and retrieve the user profile and lifelog data 6a. System receives irrelevant data		

	a. System detects the irrelevant data sent by DCL 2. b. System request again DCL 2 to make sure that data received is according to feature selected.
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:	If DCL 2 is unable to send data based on dynamic feature selection from schema, then alternate strategy has to be considered.
Sequence Diagram:	



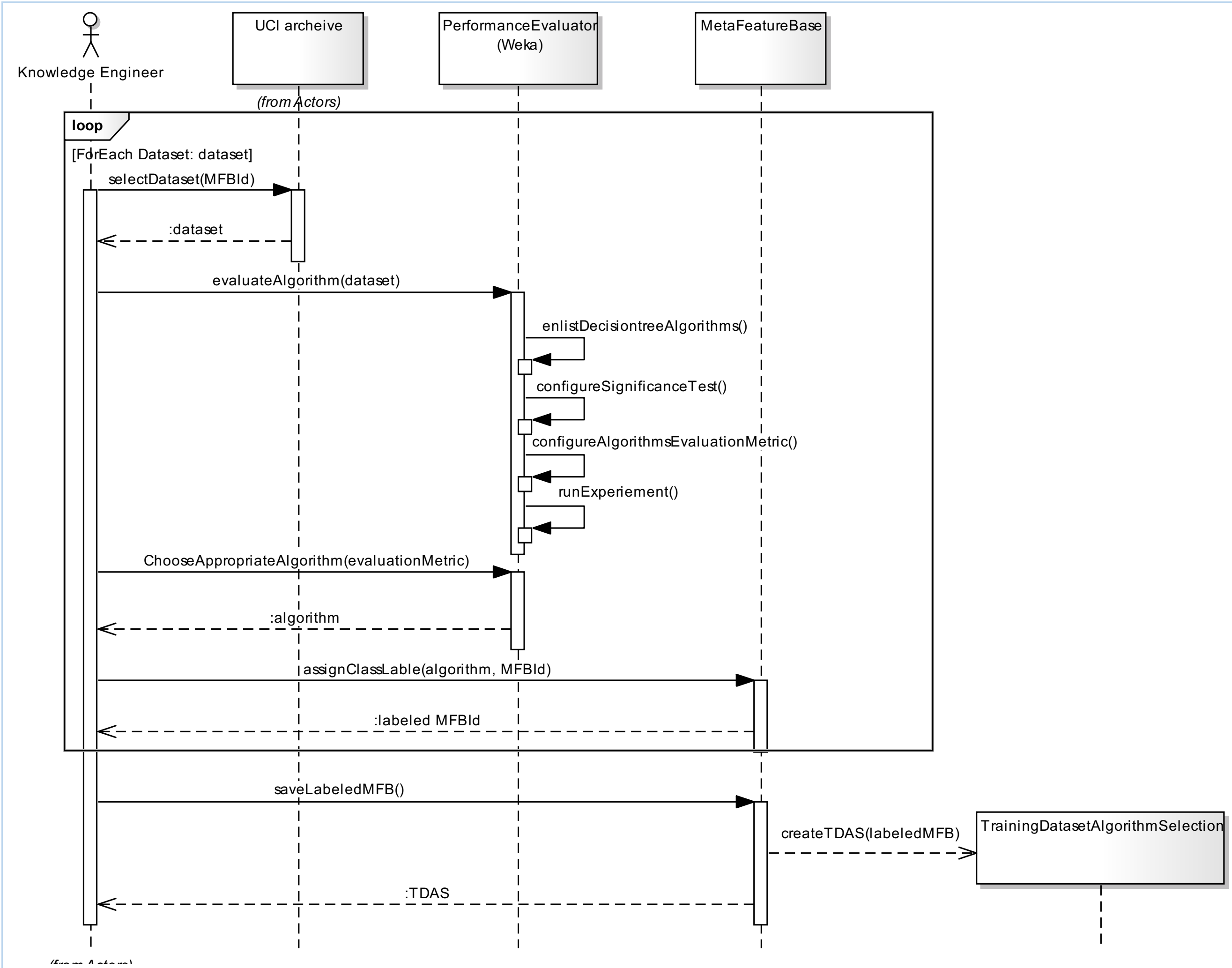
Use Case ID:	*KCL2-SUC-07*		
Use Case Name:	Extract meta-features of classification datasets		
Created By:	Rahman Ali	**Last Updated By:**	Maqbool Hussain
Date Created:	16 July 2015	**Last Revision Date:**	20 July 2015
Actors:	Knowledge Engineer/Domain Expert, UCI archive		
Description:	Datasets have 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:	


:Domain Expert
:Knowledge Engineer
:UCI archeive
MetaFeatureManager
MetaFeatureExtractor
MetaFeatureBase
par
[processMode: offline]
selectDataSets()
loadDatasets(datasetPath)
extractMetaFeatures(listDatasets, processMode)
[processMode:online]
extractMetaFeatures(onlineDataset, processMode) : List<MetaFeature>
loop MetaFeatureExtraction
[ForEach datasetId]
extractMetaFeatures(listDatasets, processMode) :List<MetaFeature>
createMetaFeatureList()
«create»
MetaFeature
loop
[ForAll SupportedMetaFeatureType]
extractMetaFeaturs(metaFeatureType) : MetaFeature
addMetaFeatureToList(metafeature)
par
[processMode: offline]
reviewMetaFeaturesList() :metaFeatureList
saveMetaFeatures()
saveMetaFeatures(metaFeatureList, datasetId)
MetaFeatureSaved(datasetId)
metaFeatureSaved(datasetId)
[processMode: online]
returnMetaFeatureList() :metaFeatureList

Use Case ID:	*KCL2-SUC-08*		
Use Case Name:	Evaluate performance of decision tree algorithm		
Created By:	Rahman Ali	**Last Updated By:**	Maqbool Hussain
Date Created:	16 July 2015	**Last Revision Date:**	20 July 2015
Actors:	Knowledge Engineer, UCI archive		
Description:	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.		
Trigger:	In the offline process, when the algorithm selection model is to build the first time.		
Preconditions:	• 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)		
Postconditions:	All datasets records in Meta-Feature Base (MFB) will be assigned with optimal decision tree algorithm class label.		
Normal Flow:	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).		

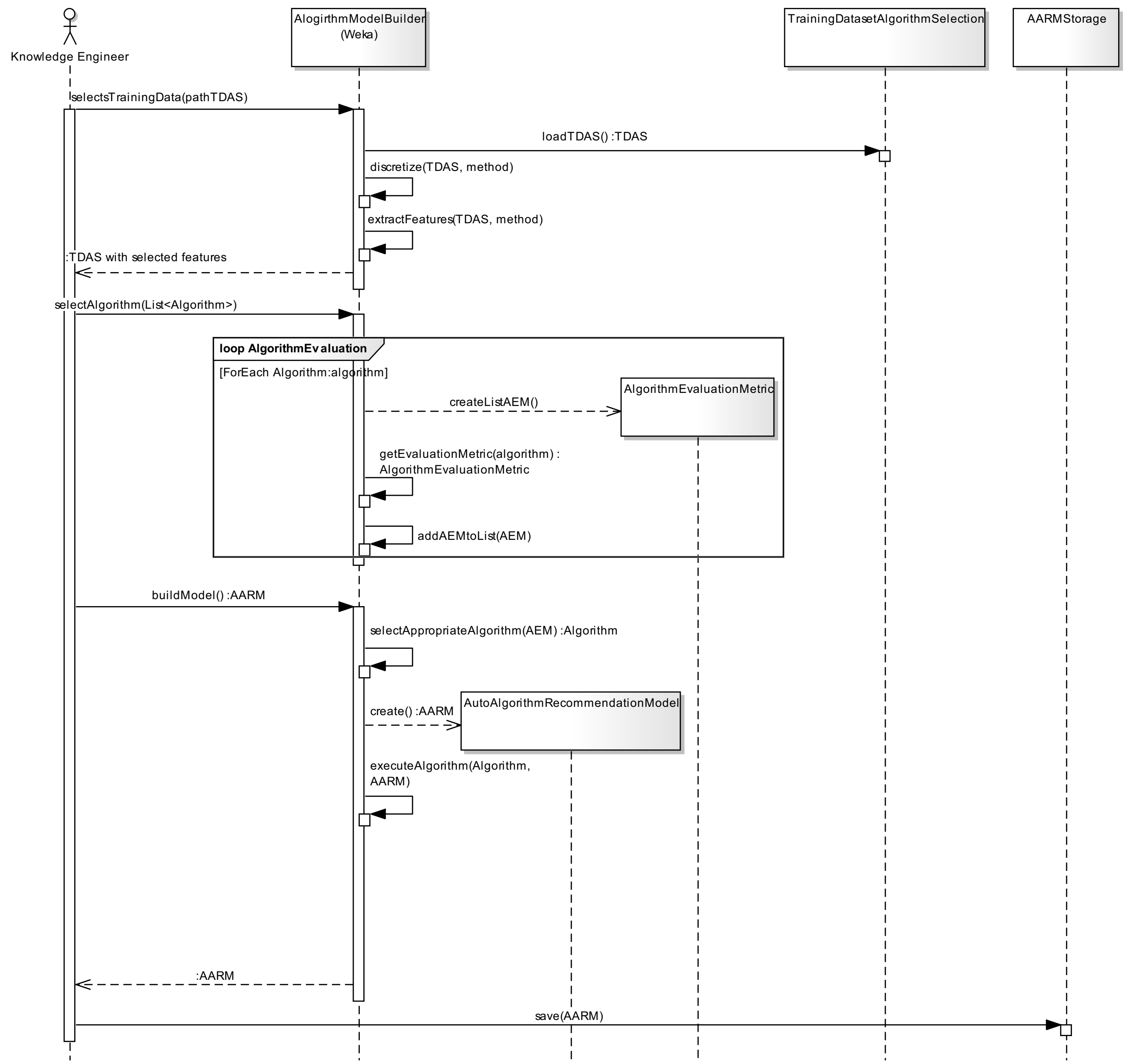
	6. System saves the updated records in MFB as final TDAS.
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Frequency of Use:	• Rarely, once enough new datasets are added to the system
Special Requirements:	N/A
Assumptions:	• The decision tree-based algorithms are implemented in Weka.
Notes and Issues:	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:	



Knowledge Engineer
UCI archeive
(from Actors)
PerformanceEvaluator (Weka)
MetaFeatureBase
loop
[ForEach Dataset: dataset]
selectDataset(MFBId)
:dataset
evaluateAlgorithm(dataset)
enlistDecisiontreeAlgorithms()
configureSignificanceTest()
configureAlgorithmsEvaluationMetric()
runExperiement()
ChooseAppropriateAlgorithm(evaluationMetric)
:algorithm
assignClassLable(algorithm, MFBId)
:labeled MFBId
saveLabeledMFB()
createTDAS(labeledMFB)
TrainingDatasetAlgorithmSelection
:TDAS

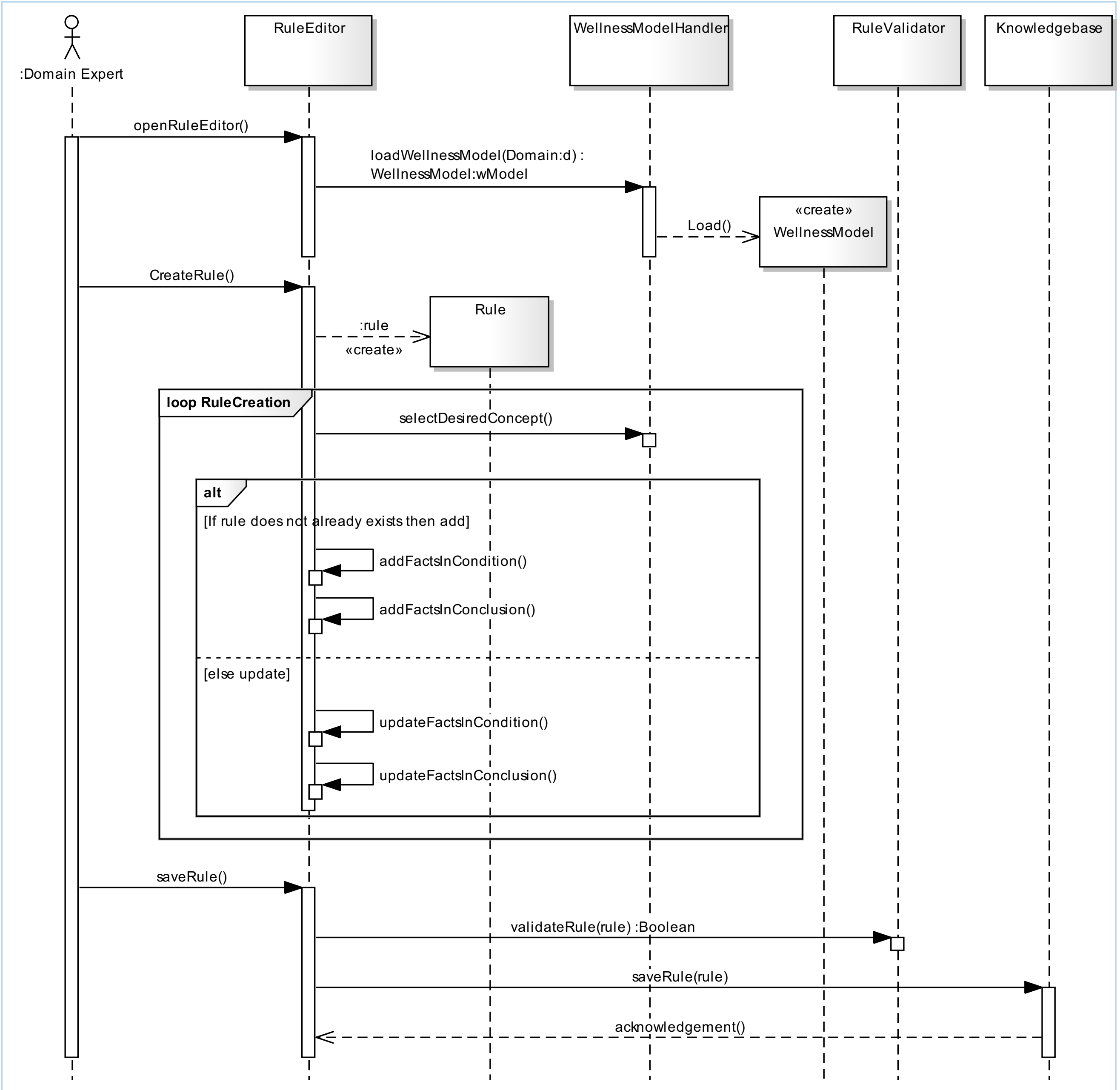
<table>
<tr><td>Use Case ID:</td><td colspan="3">KCL2-SUC-09</td></tr>
<tr><td>Use Case Name:</td><td colspan="3">Create automatic algorithm recommendation model</td></tr>
<tr><td>Created By:</td><td>Rahman Ali</td><td>Last Updated By:</td><td>Maqbool Hussain</td></tr>
<tr><td>Date Created:</td><td>10 July 2015</td><td>Last Revision Date:</td><td>22 July 2015</td></tr>
<tr><td>Actors:</td><td colspan="3">Knowledge Engineer/Domain Expert</td></tr>
<tr><td>Description:</td><td colspan="3">An automatic algorithm selection model enables knowledge engineer to automatically select appropriate algorithm for building classification model for his new dataset</td></tr>
<tr><td>Trigger:</td><td colspan="3">When the training dataset comprising datasets meta-features and algorithms performance evaluation are made available.</td></tr>
<tr><td>Preconditions:</td><td colspan="3">The datasets meta-features and algorithms performance training datasets is made available.</td></tr>
<tr><td>Postconditions:</td><td colspan="3">The automatic algorithm recommendation model (AARM) is ready to integrate in Mining Minds for real time algorithm selection.</td></tr>
<tr><td>Normal Flow:</td><td colspan="3">1. Knowledge Engineer selects TDAS.
2. System (Weka) loads TDAS.
3. Knowledge Engineer performs preprocessing of the TDAS (i.e., discretization, and features selection).
4. System refines the datasets with appropriate features.
5. Knowledge Engineer select an algorithm from decision tree (DT) or Rules learner (RL) for building AARM.
6. System executes selected algorithm on TDAS and produces evaluation matrix.
7. Knowledge Engineer records performs following tasks;
    a. Records evaluation matrix and repeats step 5-6 till all algorithms finished.
    b. Knowledge Engineer selects the appropriate algorithm.
8. System builds AARM model using the selected algorithm.
9. Knowledge Engineer saves AARM.
10. System saves the AARM into AARM storage.</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, when TDAS is updated with new datasets or algorithms</td></tr>
</table>

Special Requirements:	Availability of records > 60 for minimum acceptable accuracy
Assumptions:	N/A
Notes and Issues:	N/A
Sequence Diagram:	



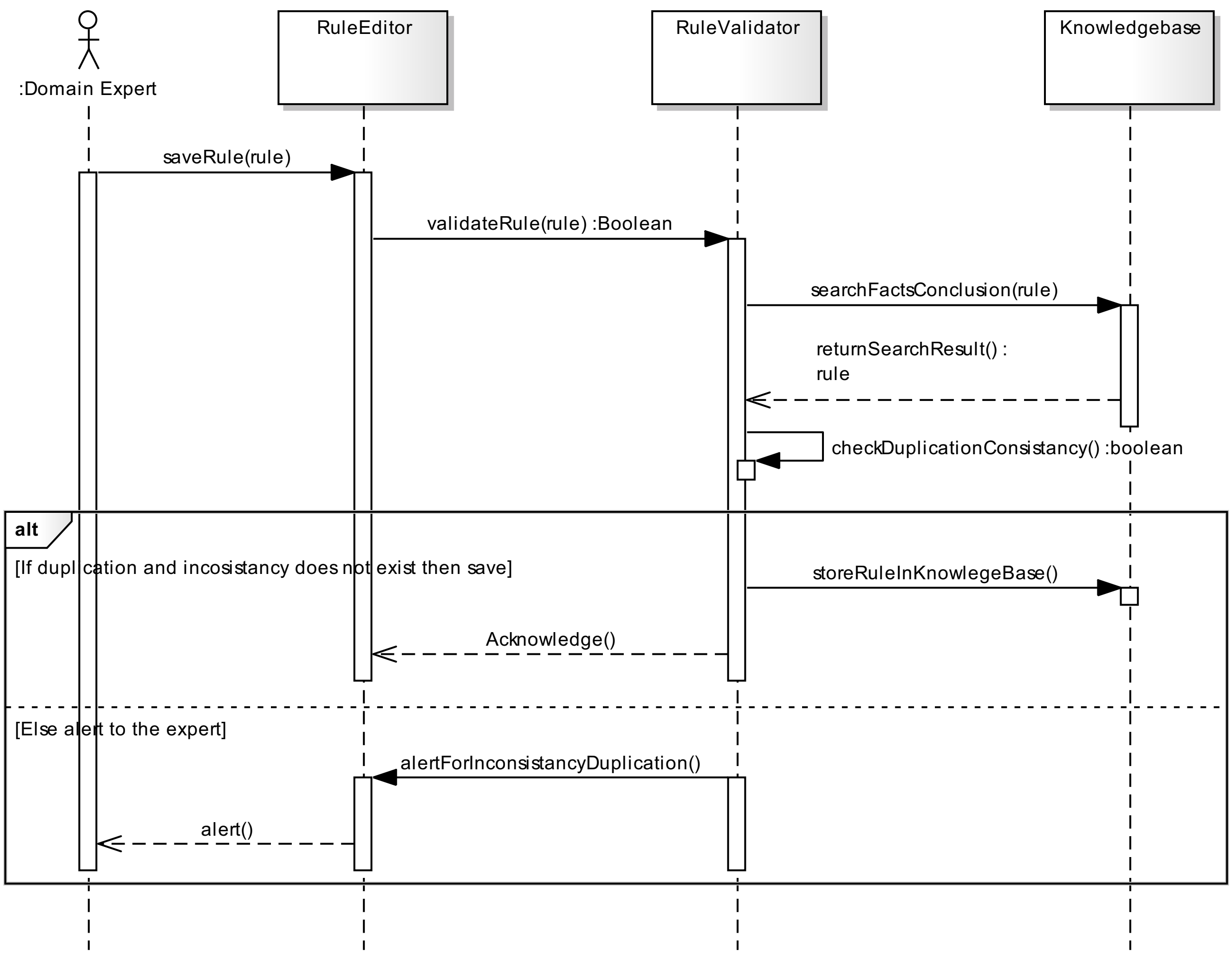
Use Case ID:	*KCL2-SUC-10*		
Use Case Name:	Create Rule		
Created By:	Taqdir Ali	**Last Updated By:**	Maqbool Hussain
Date Created:	11 July 2015	**Last Revision Date:**	29 July 2015
Actors:	Domain Expert		
Description:	Knowledge bases need to enhance with up-to-date knowledge for correct recommendation. 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; a. Using wellness model, dragging concepts and facts into conditional or conclusion part of the rule editor. b. 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 a. Add new facts of the rule in condition. b. 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; a. The system validates the rule using "KCL2-SUC-11". b. The system stores the validated rule into the knowledge base		
Alternative Flows:	6a. System founds the rule is already exists in rule repository c. Domain expert review the existing facts and		

	conclusion. d. 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:	


:Domain Expert
RuleEditor
WellnessModelHandler
RuleValidator
Knowledgebase
openRuleEditor()
loadWellnessModel(Domain:d) :
WellnessModel:wModel
«create»
WellnessModel
Load()
CreateRule()
Rule
:rule
«create»
loop RuleCreation
selectDesiredConcept()
alt
[If rule does not already exists then add]
addFactsInCondition()
addFactsInConclusion()
[else update]
updateFactsInCondition()
updateFactsInConclusion()
saveRule()
validateRule(rule) :Boolean
saveRule(rule)
acknowledgement()

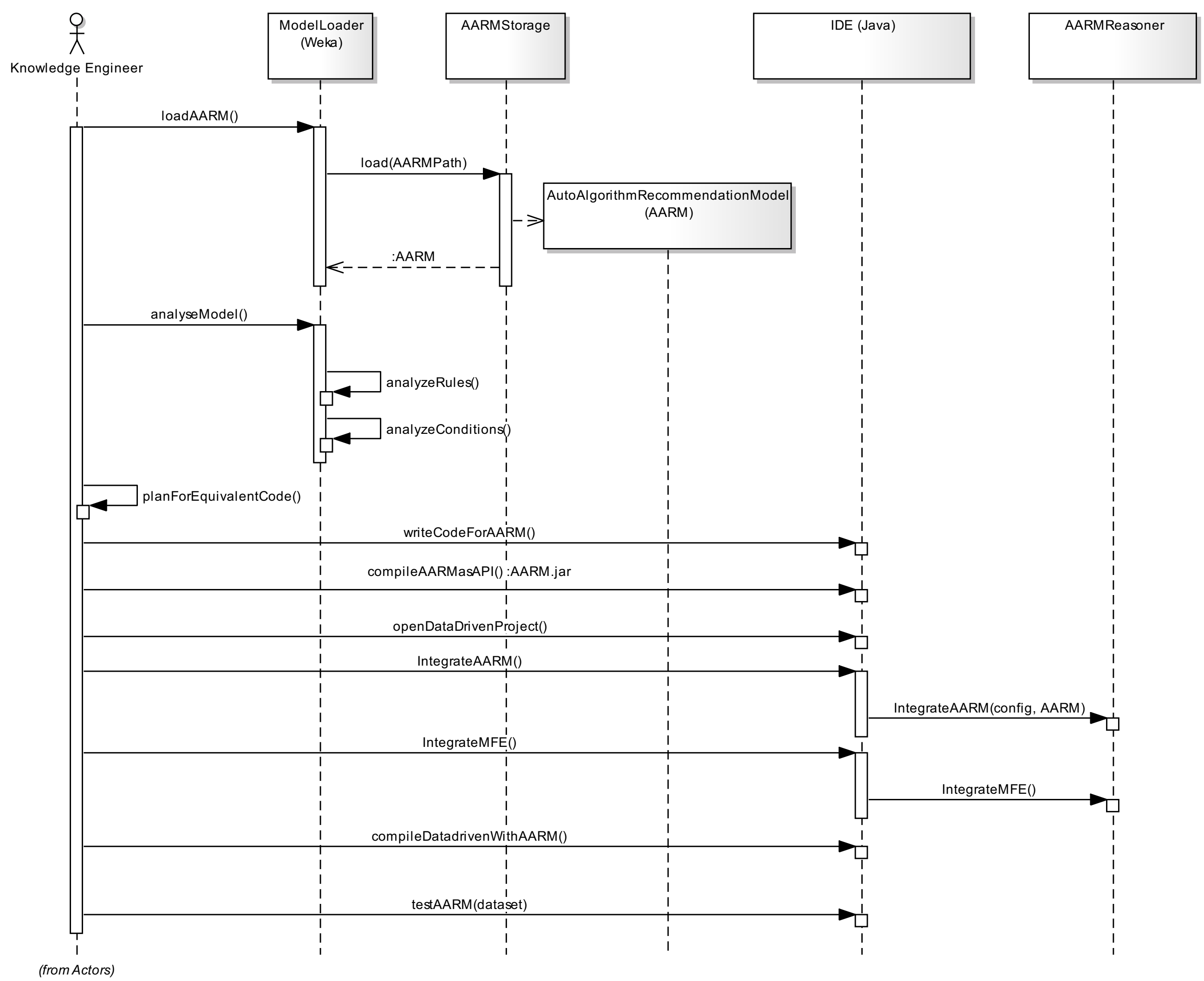
Use Case ID:	*KCL2-SUC-11*		
Use Case Name:	Validate Rule		
Created By:	Taqdir Ali	**Last Updated By:**	Maqbool Hussain
Date Created:	11 July 2015	**Last Revision Date:**	29 July 2015
Actors:	Domain Expert		
Description:	In new rules creation and editing existing rules, duplication and inconsistency may occur. The validation is needed to validate and find the duplication and inconsistency among the rules.		
Trigger:	• When new rule is going to create. • When existing rule is going to update		
Preconditions:	The rule creation and editing process completed by physician successfully.		
Postconditions:	The validated rule shall be saved into knowledge base.		
Normal Flow:	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.		
Alternative Flows:	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.		
Exceptions:	N/A		
Includes:	N/A		
Frequency of Use:	Whenever domain expert want to save the rule		

Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	Finding duplication using facts of condition and conclusion in existing rules is challenging task.
Sequence Diagram:	



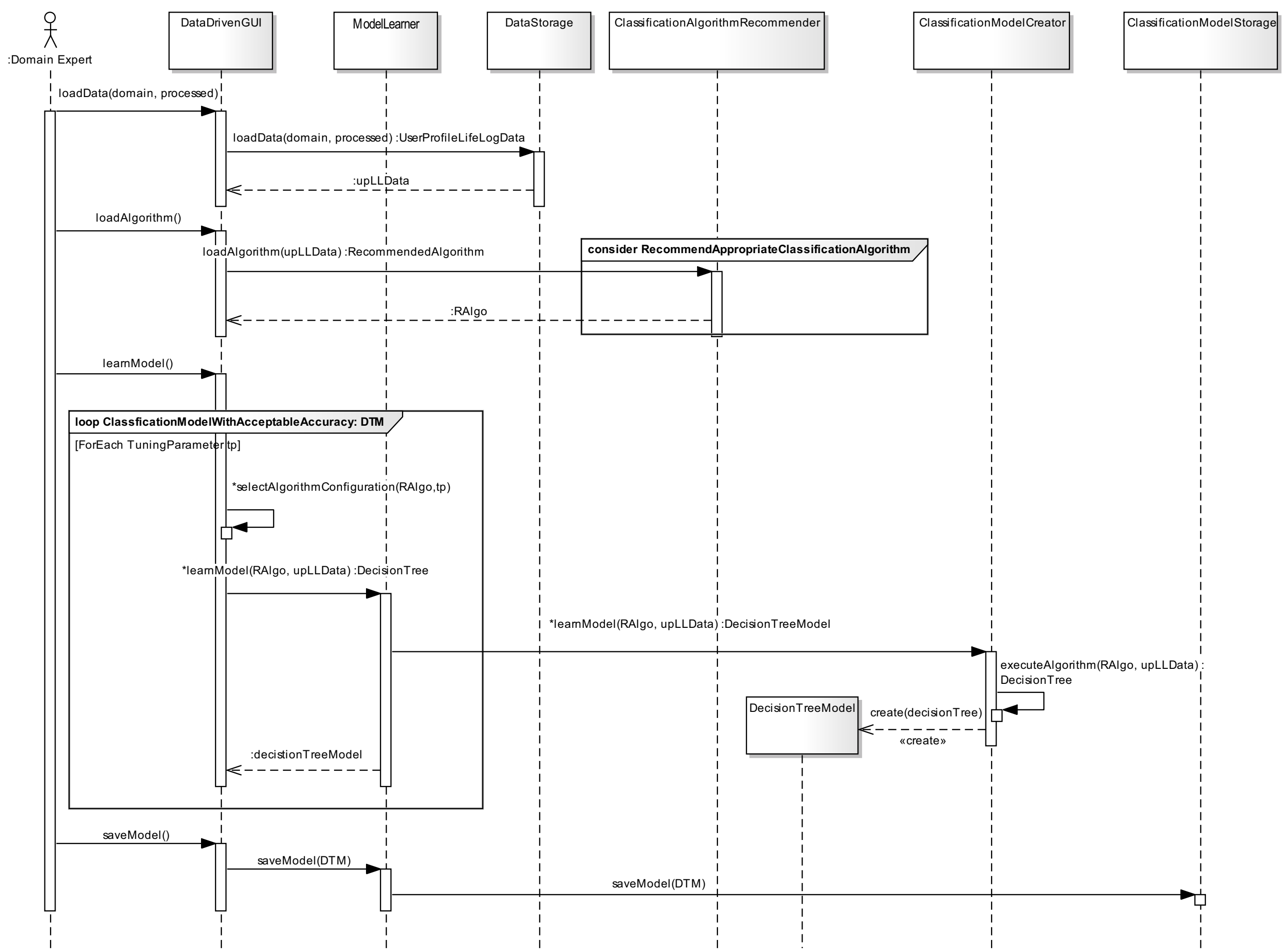
Use Case ID:	*KCL2-SUC-12*		
Use Case Name:	Integrate automatic algorithm recommendation model		
Created By:	Rahman Ali	**Last Updated By:**	Maqbool Hussain
Date Created:	10 July 2015	**Last Revision Date:**	28 July 2015
Actors:	Knowledge Engineer		
Description:	For real time recommendation of classification 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:	



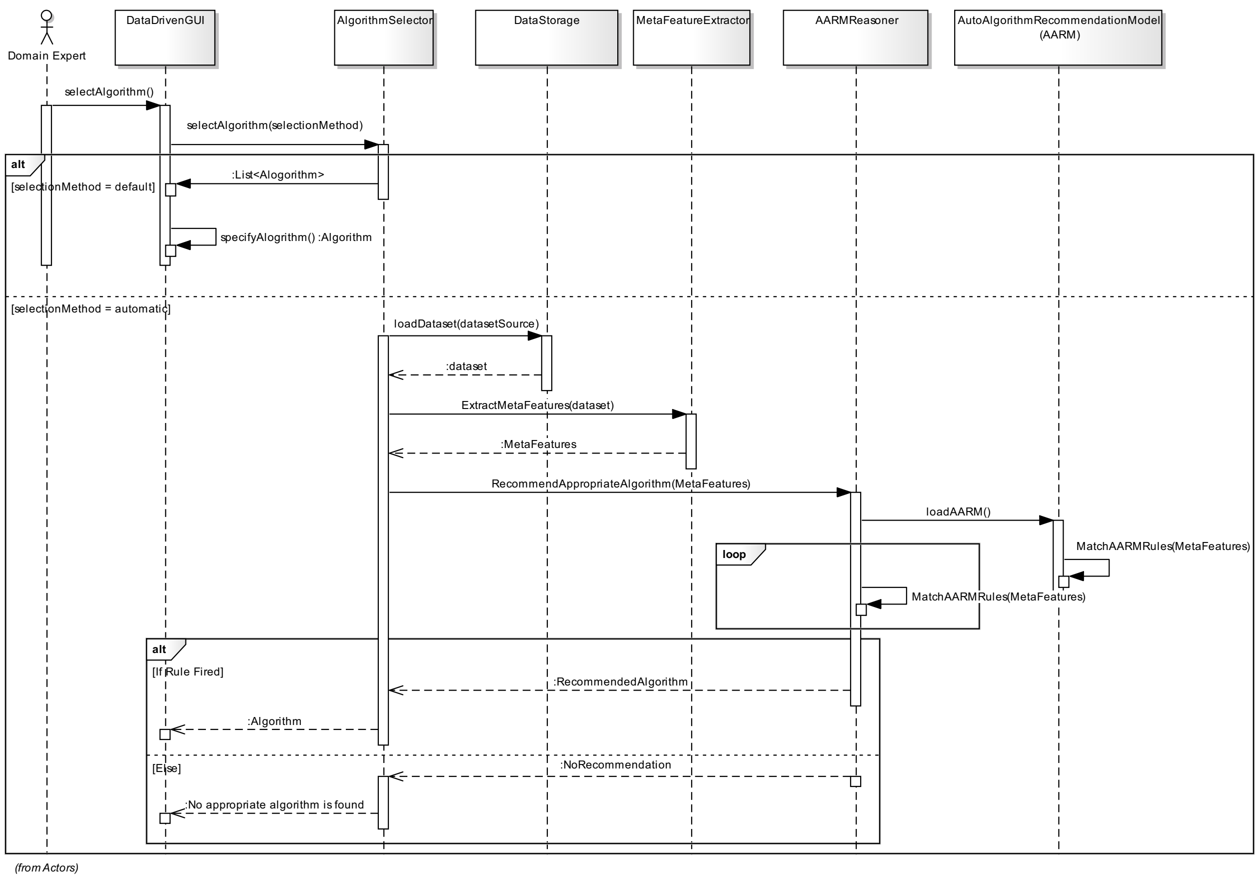
Use Case ID:	*KCL2-SUC-13*		
Use Case Name:	Learn classification model		
Created By:	Maqbool Ali	**Last Updated By:**	Maqbool Hussain
Date Created:	10 July 2015	**Last Revision Date:**	27 July 2015
Actors:	Domain Expert		
Description:	An expert wants to see hidden knowledge from user profile lifelog 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-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><th>Use Case ID:</th><td colspan="3">KCL2-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 July 2015</td><td>Last Revision Date:</td><td>27 July 2015</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-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-SUC-07 (alternate flow)</td></tr>
<tr><td>Frequency of Use:</td><td colspan="3">Frequently, when domain expert needs to select appropriate</td></tr>
</table>

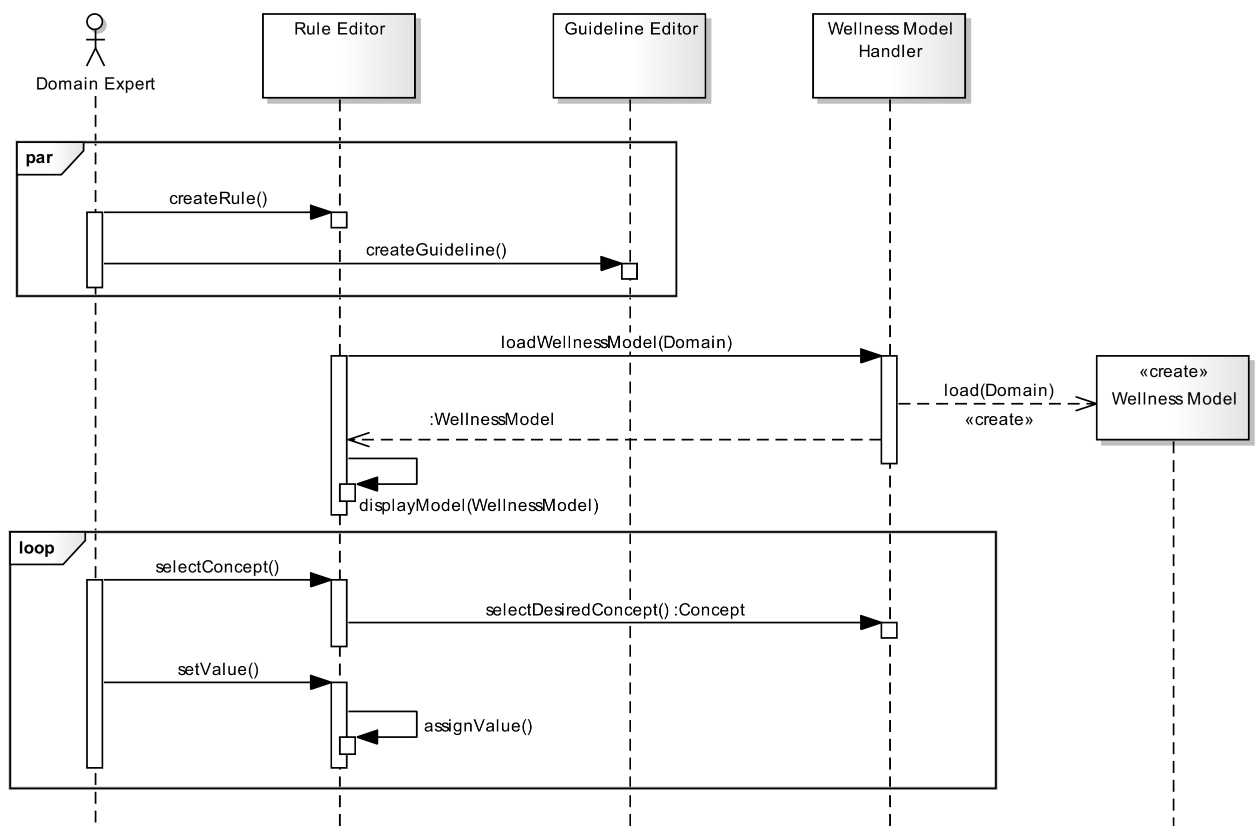
	algorithm for his/her dataset
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A
Sequence Diagram:	



Use Case ID:	*KCL2-SUC-15*		
Use Case Name:	Manage concepts of domain model		
Created By:	Taqdir Ali	**Last Updated By:**	Taqdir Ali, Maqbool Hussain
Date Created:	27 July 2015	**Last Revision Date:**	27 July 2015
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-SUC-10) or creates guideline (using KCL2-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		

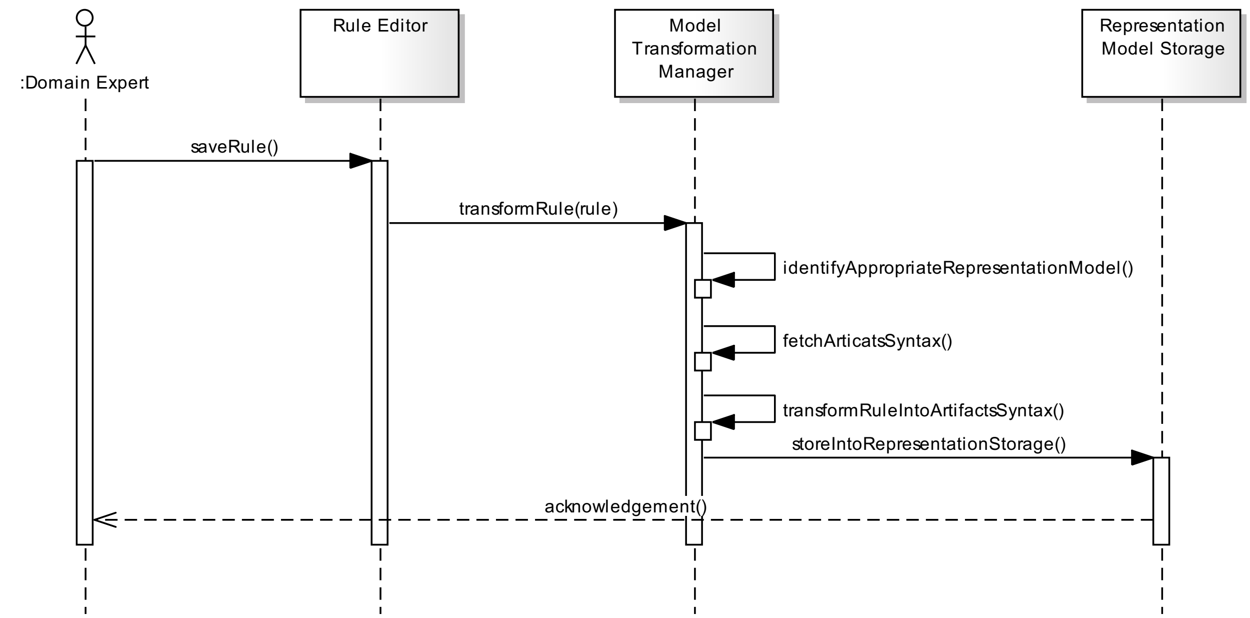
Assumptions:	Wellness model repository in the system is exist.
Notes and Issues:	If wellness model storage does not exist in system the administrator shall build the wellness model storage first and configure with system.

Sequence Diagram:



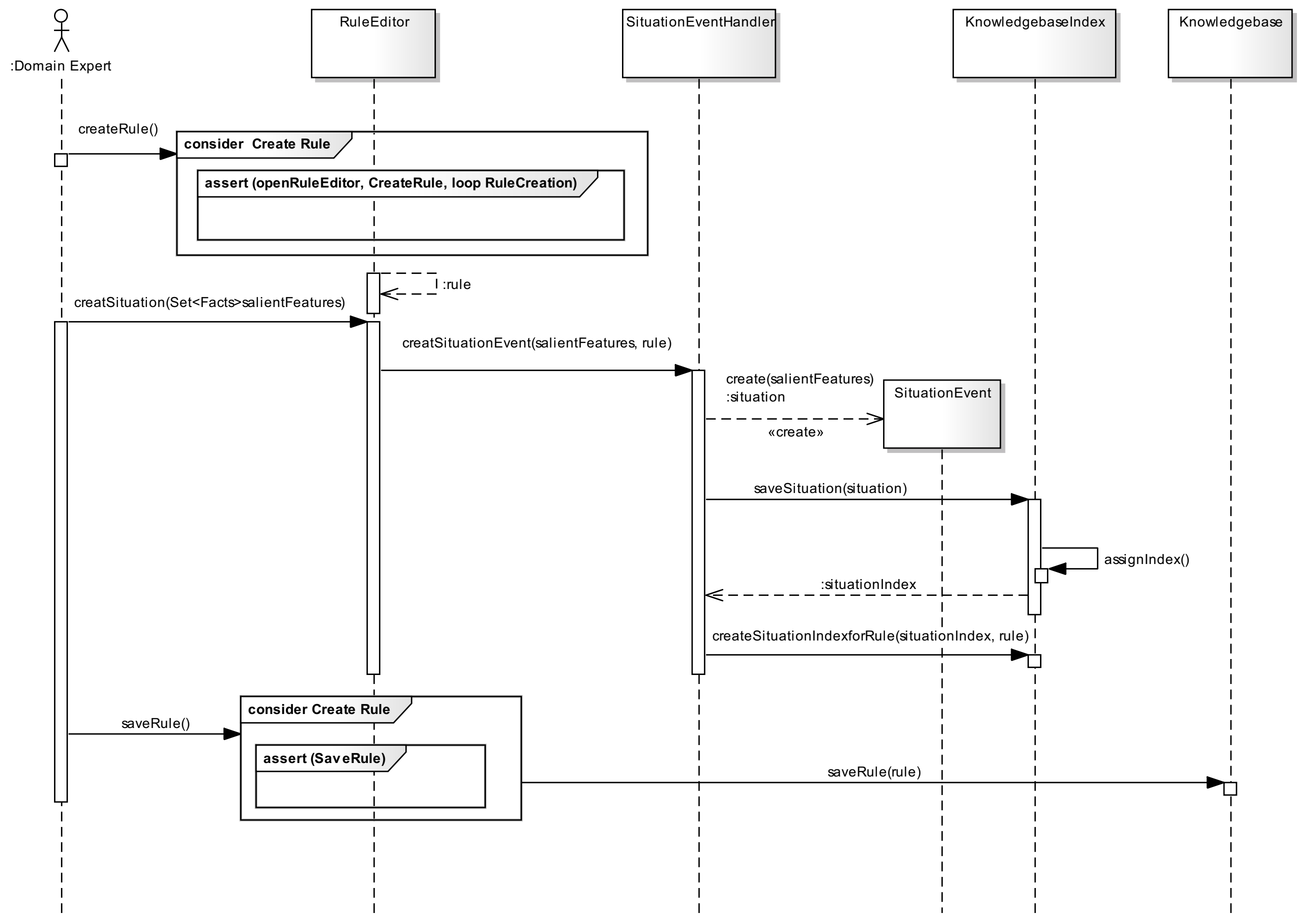
Use Case ID:	*KCL2-SUC-16*		
Use Case Name:	Transform Knowledge Rule		
Created By:	Taqdir Ali and Maqbool Hussain	**Last Updated By:**	Maqbool Hussain
Date Created:	27 July 2015	**Last Revision Date:**	27 July 2015
Actors:	Domain Expert		
Description:	The new created rules are needed to transform to 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:



Use Case ID:	*KCL2-SUC-17*		
Use Case Name:	Create Situation Event		
Created By:	Maqbool Hussain	**Last Updated By:**	Maqbool Hussain
Date Created:	29 July 2015	**Last Revision Date:**	29 July 2015
Actors:	Domain Expert		
Description:	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.		
Trigger:	Whenever domain expert wants to store the created or updated rule.		
Preconditions:	The rule has salient features based on which the rule can be indexed.		
Postconditions:	• Rule is saved into knowledgebase • Rule is indexed based on the created situation event		
Normal Flow:	1. Domain expert start creating rule; a. Performs steps 1-5 in KCL2-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-SUC-10. 4. System saves the rule and index of the rule.		
Alternative Flows:	N/A		
Exceptions:	N/A		
Extends:	Create Rule (KCL2-SUC-10)		
Frequency of Use:	Whenever domain expert want to save rule		
Special Requirements:	N/A		
Assumptions:	N/A		

Notes and Issues:	
Sequence Diagram:	



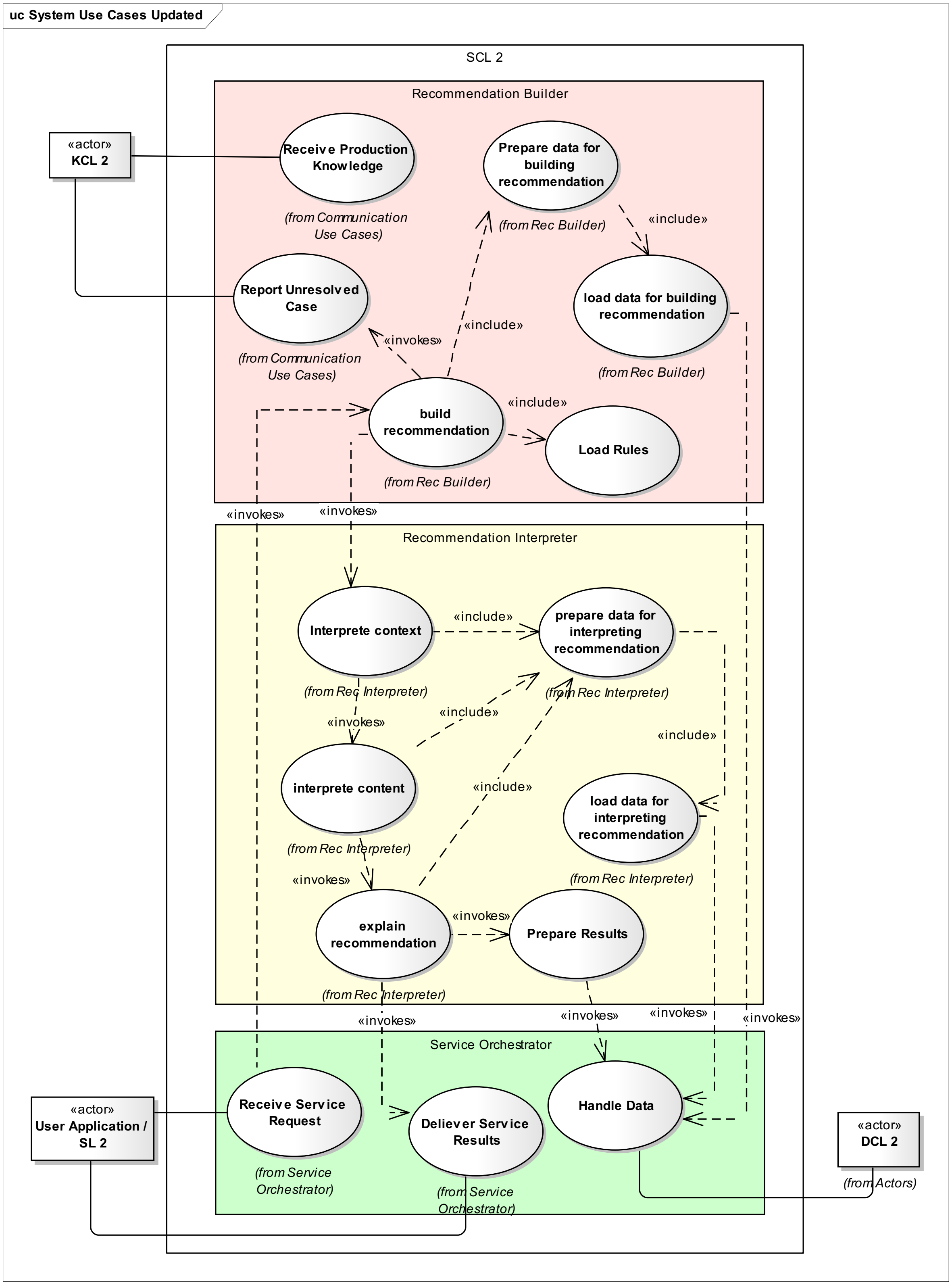
Service Curation Layer (SCL Ver. 2)

System Level Use cases

List of Use cases

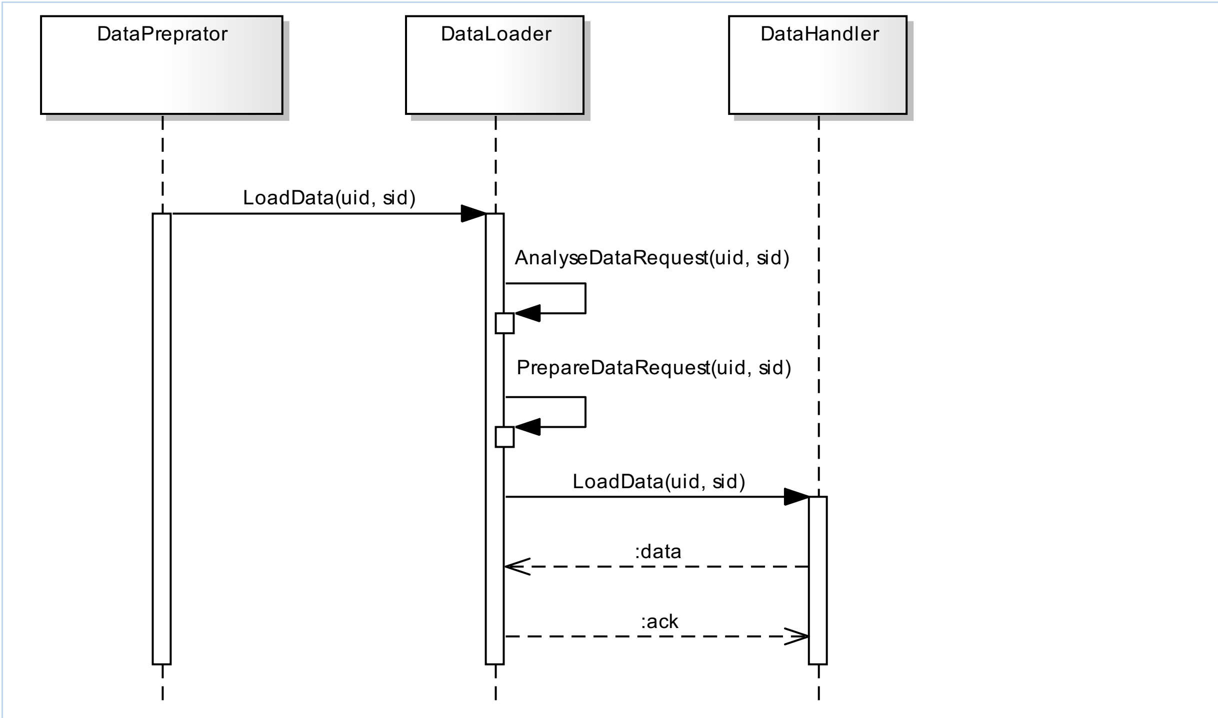
Use Case #ID	Name
SCL2-SUC-01	Load data for building recommendation
SCL2-SUC-02	Prepare data for building recommendation
SCL2-SUC-03	Load Rules
SCL2-SUC-04	Build Recommendation
SCL2-SUC-05	Receive Production Knowledge
SCL2-SUC-06	Report Unresolved Case
SCL2-SUC-07	Load data for interpreting recommendation
SCL2-SUC-08	Prepare data for interpreting recommendation
SCL2-SUC-09	Interpret Context
SCL2-SUC-10	Interpret Content
SCL2-SUC-11	Explain recommendations
SCL2-SUC-12	Prepare Results
SCL2-SUC-13	Receive service request
SCL2-SUC-14	Handle Data
SCL2-SUC-15	Deliver service results

Use case Diagram

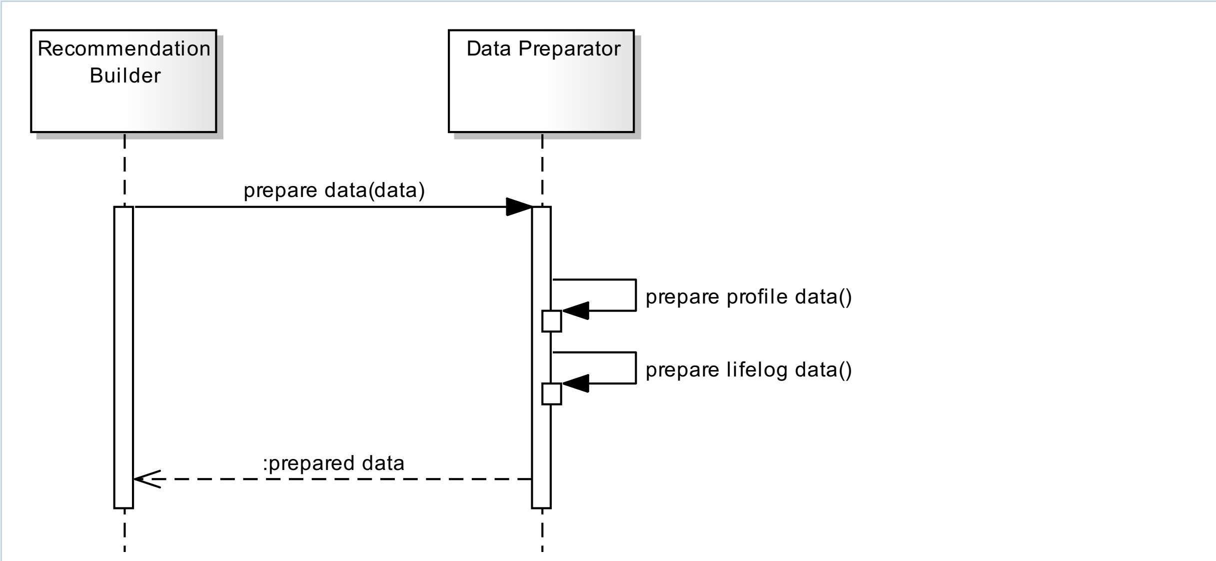


Use case Description

Use Case ID:	*SCL2-SUC-01*		
Use Case Name:	Load data for building recommendation		
Created By:	Rahman Ali	**Last Updated By:**	Rahman Ali
Date Created:	14 July 2015	**Last Revision Date:**	28 July 2015
Actors:	SCL2-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.		
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:			

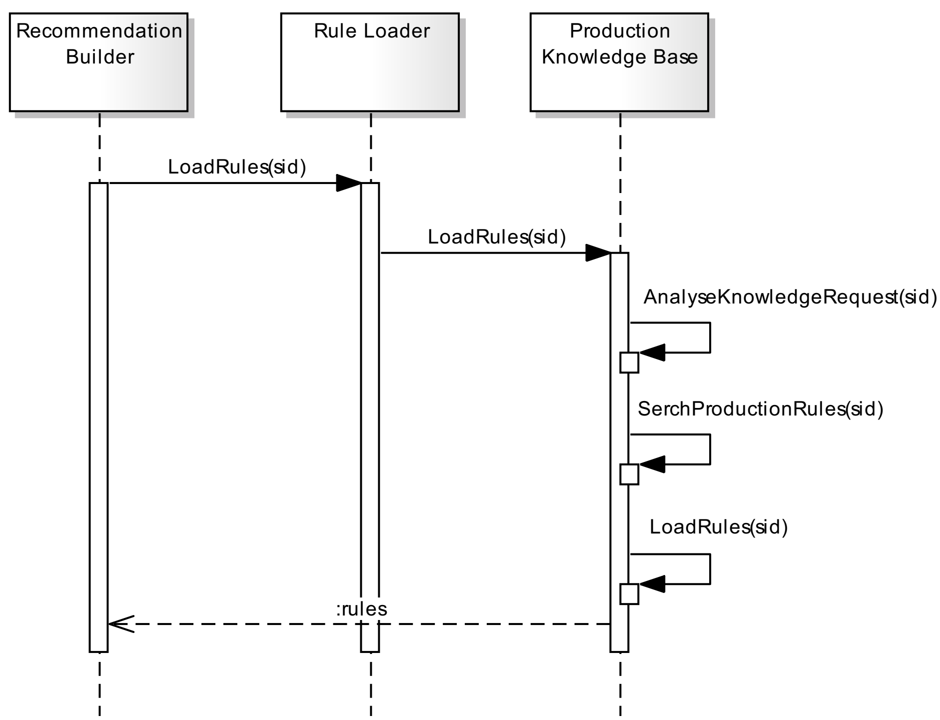

DataPreprator
DataLoader
DataHandler
LoadData(uid, sid)
AnalyseDataRequest(uid, sid)
PrepareDataRequest(uid, sid)
LoadData(uid, sid)
:data
:ack

Use Case ID:	SCL2-SUC-02		
Use Case Name:	Prepare data for building recommendation		
Created By:	Rahman Ali	**Last Updated By:**	Rahman Ali
Date Created:	14 July 2015	**Last Revision Date:**	28 July 2015
Actors:	SCL2-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-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:			



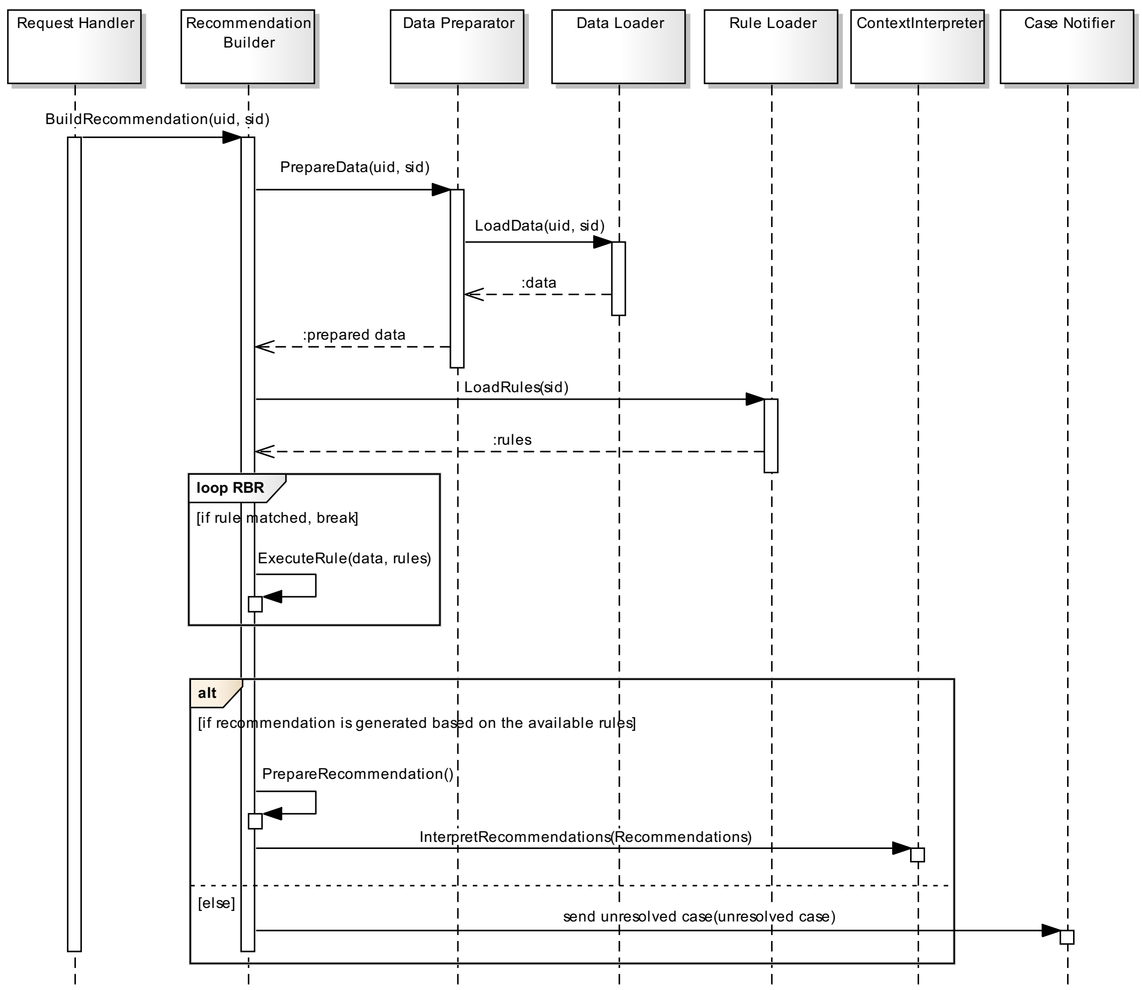
Use Case ID:	*SCL2-SUC-03*		
Use Case Name:	Load Rules		
Created By:	Rahman Ali	**Last Updated By:**	Rahman Ali
Date Created:	28 July 2015	**Last Revision Date:**	28 July 2015
Actors:	SCL2-SUC-04 (Build Recommendation)		
Description:	Rule-based reasoned needs knowledge rules to perform reasoning using 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 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:	

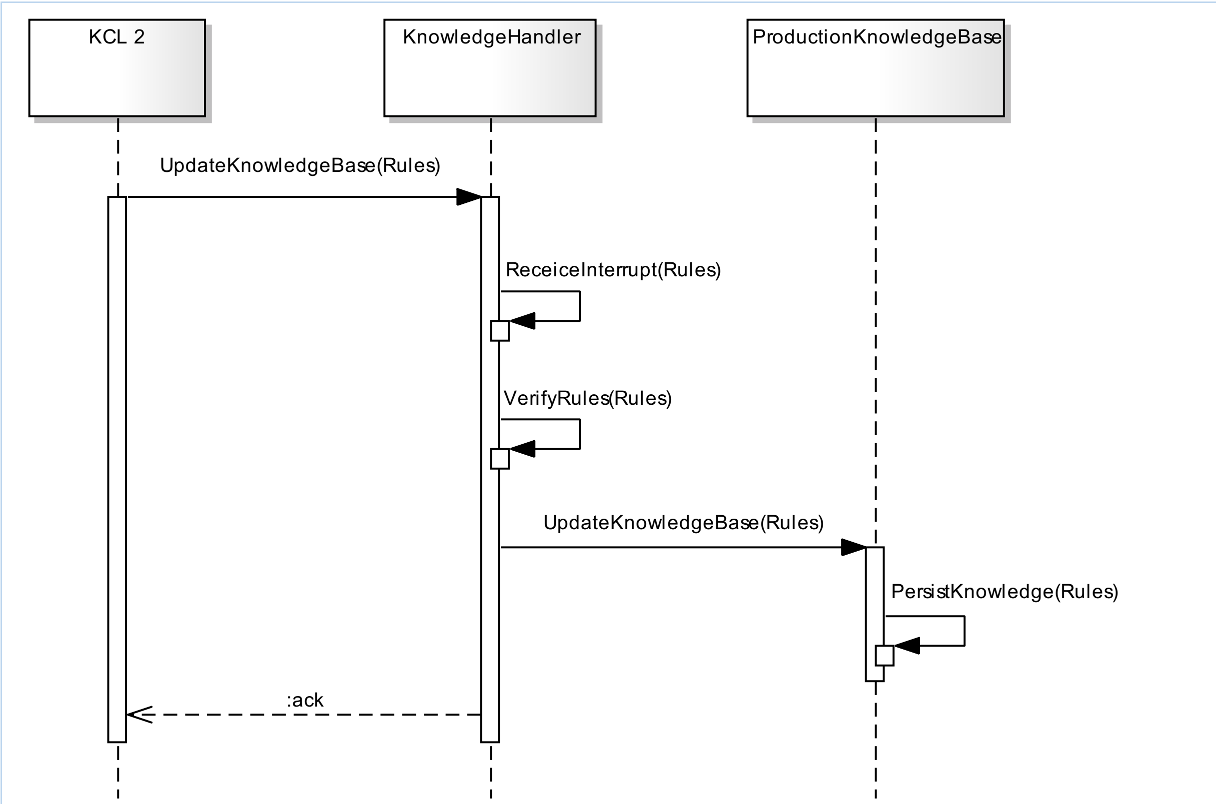


Use Case ID:	*SCL2-SUC-04*		
Use Case Name:	Build recommendations		
Created By:	Rahman Ali	**Last Updated By:**	Rahman Ali
Date Created:	15 July 2015	**Last Revision Date:**	15 July 2015
Actors:	SCL2-SUC-13 (Request Handler), SCL2-SUC-04, SCL2-SUC-02, SCL2-SUC-01, SCL2-SUC-03, SCL2-SUC-09 (Interpret 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:	5a. The system could not find rule to execute a. Recommendation Builder sends message along with Unresolved Case to Case Notifier		
Exceptions:	N/A		
Includes:	SCL2-SUC-02, SCL2-SUC-03		
Frequency of Use:	Frequent: when recommendation builder is invoked for		

	generating recommendation.
Special Requirements:	N/A
Assumptions:	KCL 2 and RB 2 agree on common format of production rules.
Notes and Issues:	N/A
Sequence Diagram:	

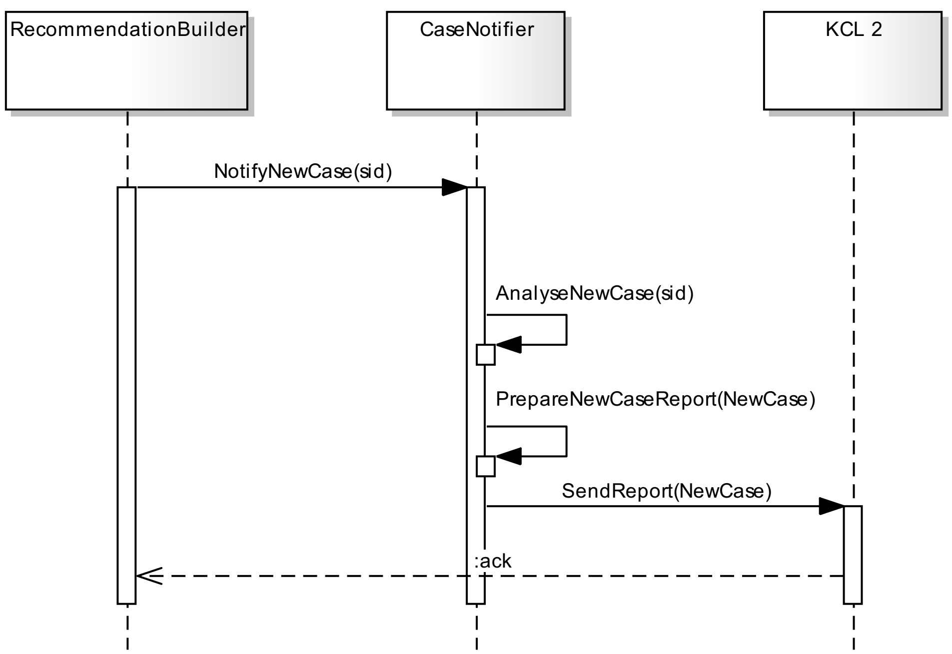


Use Case ID:	*SCL2-SUC-05*		
Use Case Name:	Receive Production Knowledge		
Created By:	Muhammad Afzal	**Last Updated By:**	Rahman Ali, Muhammad Afzal
Date Created:	5 July 2015	**Last Revision Date:**	28 July 2015
Actors:	KCL 2		
Description:	The knowledge is originated by KCL 2 and is transfer to SCL 2 to keep a local copy of the production knowledge.		
Trigger:	At knowledge creation/update time		
Preconditions:	SCL 2 and KCL 2 has a common representation agreement		
Postconditions:	The SCL 2 copy of knowledge is updated and is synchronized with KCL 2		
Normal Flow:	1. KCL interrupt Knowledge Handler for new knowledge 2. Knowledge Handler verifies the knowledge 3. Knowledge Handler make a local of the received knowledge in the Production Knowledge Base 4. Knowledge is persisted in Production Knowledge Base 5. KCL 2 is acknowledged of the knowledge receipt		
Alternative Flows:	N/A		
Exceptions:	N/A		
Includes:	N/A		
Frequency of Use:	Less frequent: at knowledge creation/update time		
Special Requirements:	N/A		
Assumptions:	N/A		
Notes and Issues:	N/A		
Sequence Diagram:			

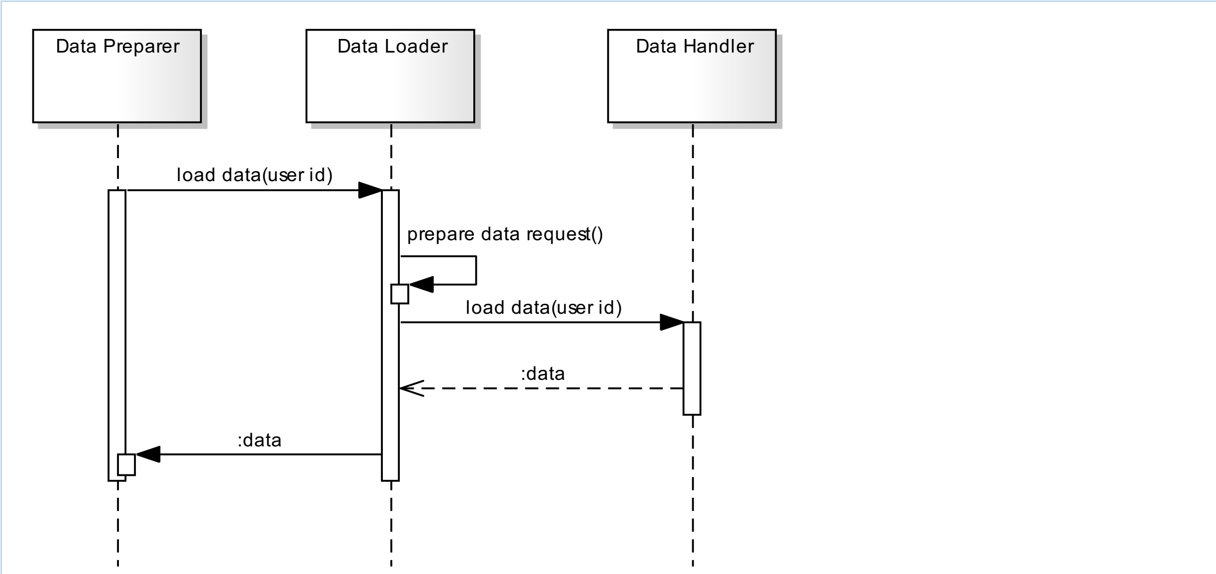


Use Case ID:	*SCL2-SUC-06*		
Use Case Name:	Report unresolved case		
Created By:	Muhammad Afzal	**Last Updated By:**	Rahman Ali, Muhammad Afzal
Date Created:	5 July 2015	**Last Revision Date:**	28 July 2015
Actors:	SCL2-SUC-04 (Build Recommendation), KCL 2		
Description:	Notifying KCL 2 that reasoner is incapable to generate recommendation for the service request. KCL 2 may be able to acquired new knowledge for such service request to handle in future.		
Trigger:	At the time when reasoner is not capable to generate recommendation because of insufficient knowledge in the KB.		
Preconditions:	Reasoner has completed the reasoning process		
Postconditions:	The message with reason is successfully reported to KCL 2		
Normal Flow:	1. Recommendation Builder notify unresolved case as new		

	case prepare the report 2. Case Notifier analyses the new case 3. Case Notifier prepare the new case report 4. Case Notifier sends new case report to KCL 2 5. KCL 2 acknowledges the new case receipt
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Frequency of Use:	Less frequent: when reasoner detects new case not handled with existing knowledge.
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	N/A
Sequence Diagram:	

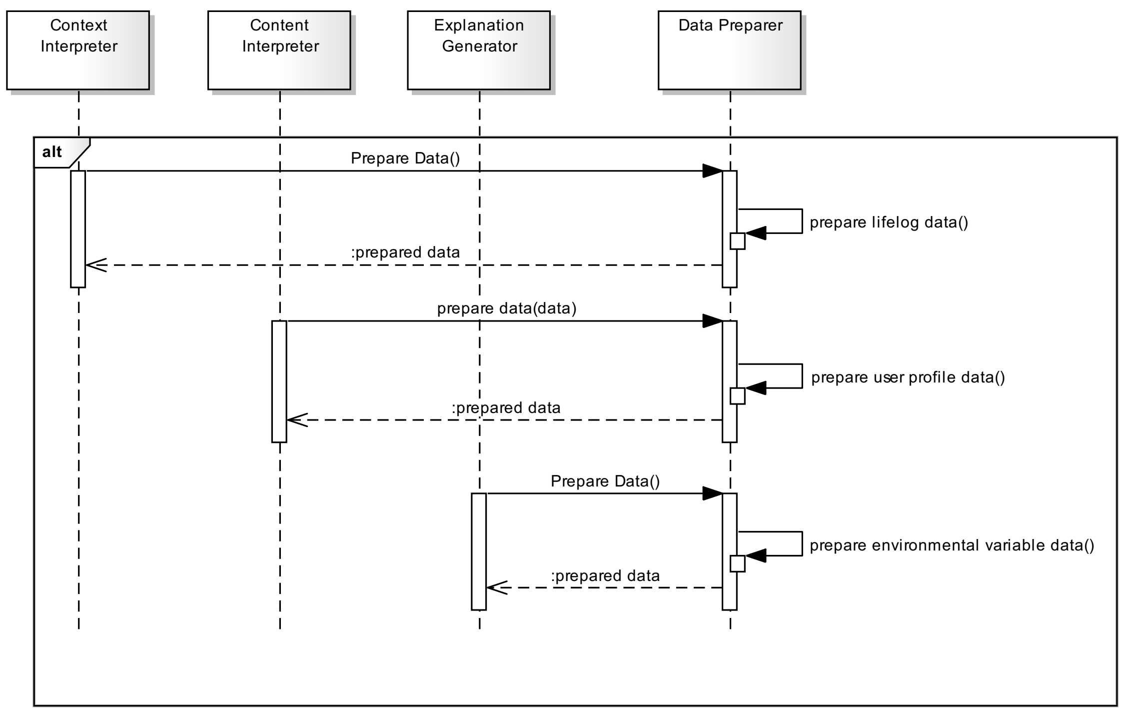


Use Case ID:	*SCL2-SUC-07*		
Use Case Name:	Load data for interpreting recommendations		
Created By:	Muhammad Afzal	**Last Updated By:**	Rahman Ali, Muhammad Afzal
Date Created:	15 July 2015	**Last Revision Date:**	28 July 2015
Actors:	Data Preparer		
Description:	The data is loaded from DCL 2 through 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:			



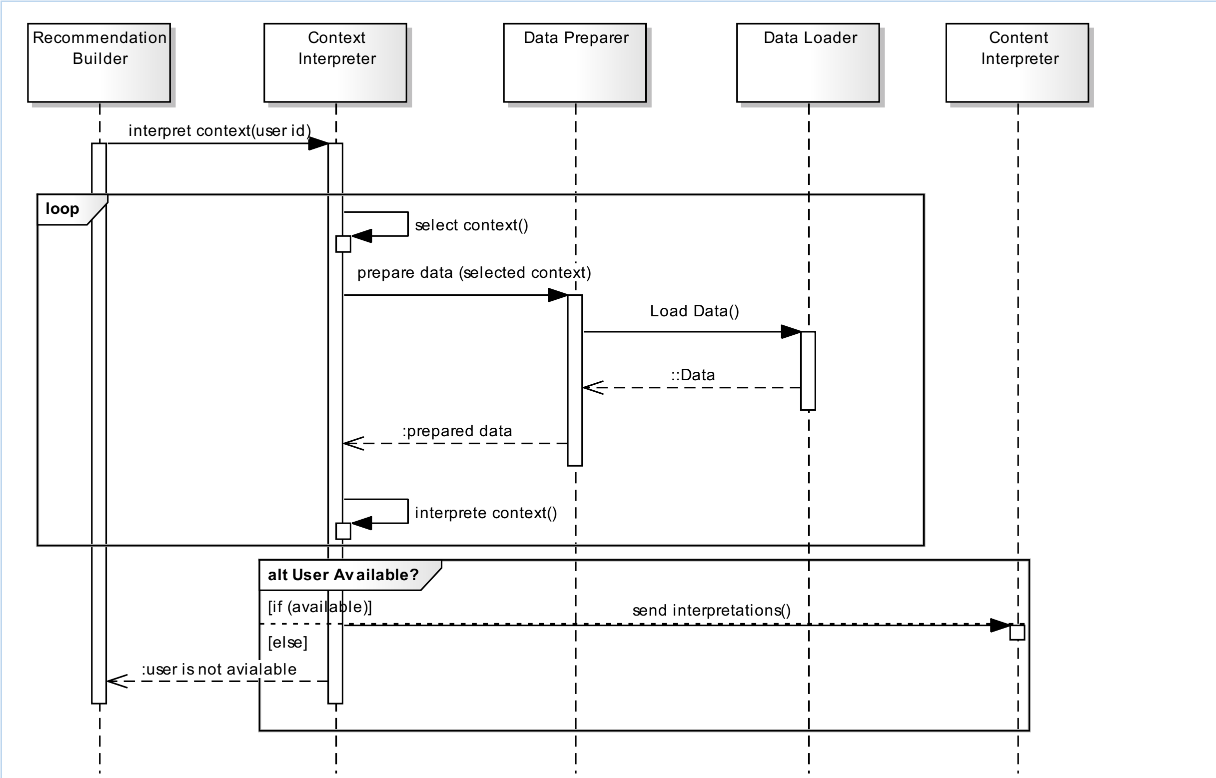
Use Case ID:	*SCL2-SUC-08*		
Use Case Name:	Prepare data for interpreting recommendations		
Created By:	Muhammad Afzal	**Last Updated By:**	Rahman Ali, Muhammad Afzal
Date Created:	15 July 2015	**Last Revision Date:**	28 July 2015
Actors:	Context Interpreter		
Description:	The loaded data is prepared 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-SUC-07
Frequency of Use:	Very frequent: at every service request
Special Requirements:	N/A
Assumptions:	N/A
Notes and Issues:	
Sequence Diagram:	



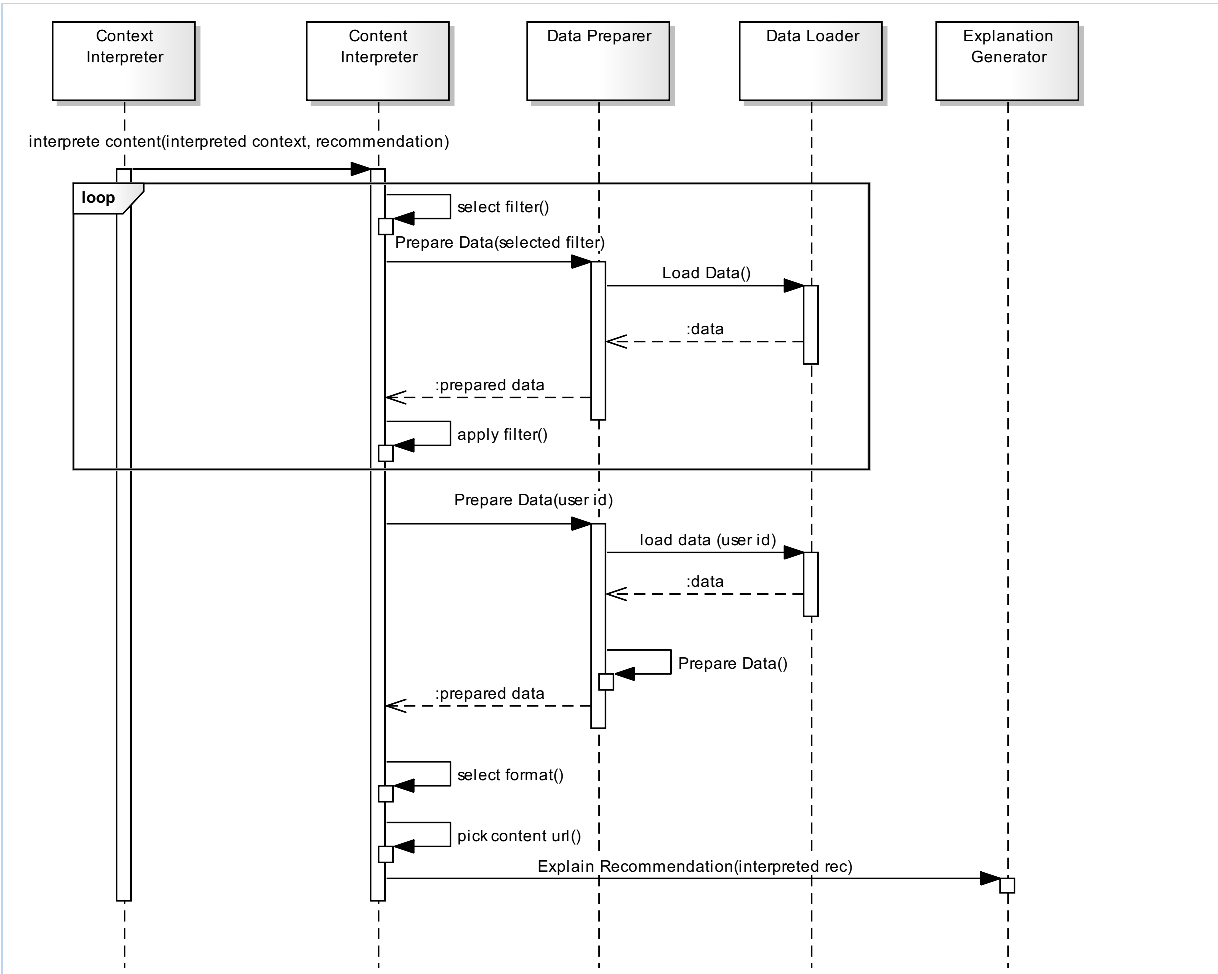
Use Case ID:	*SCL2-SUC-09*		
Use Case Name:	Interpret context		
Created By:	Muhammad	**Last Updated By:**	Rahman Ali,

	Afzal		Muhammad Afzal
Date Created:	15 July 2015	**Last Revision Date:**	28 July 2015
Actors:	Build Recommendation (SCL2-SUC-4)		
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-SUC-08		
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-SUC-10*		
Use Case Name:	Interpret contents		
Created By:	Muhammad Afzal	**Last Updated By:**	Rahman Ali
Date Created:	15 July 2015	**Last Revision Date:**	28 July 2015
Actors:	Interpret Context (SCL2-SUC-09)		
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-SUC-08
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:	

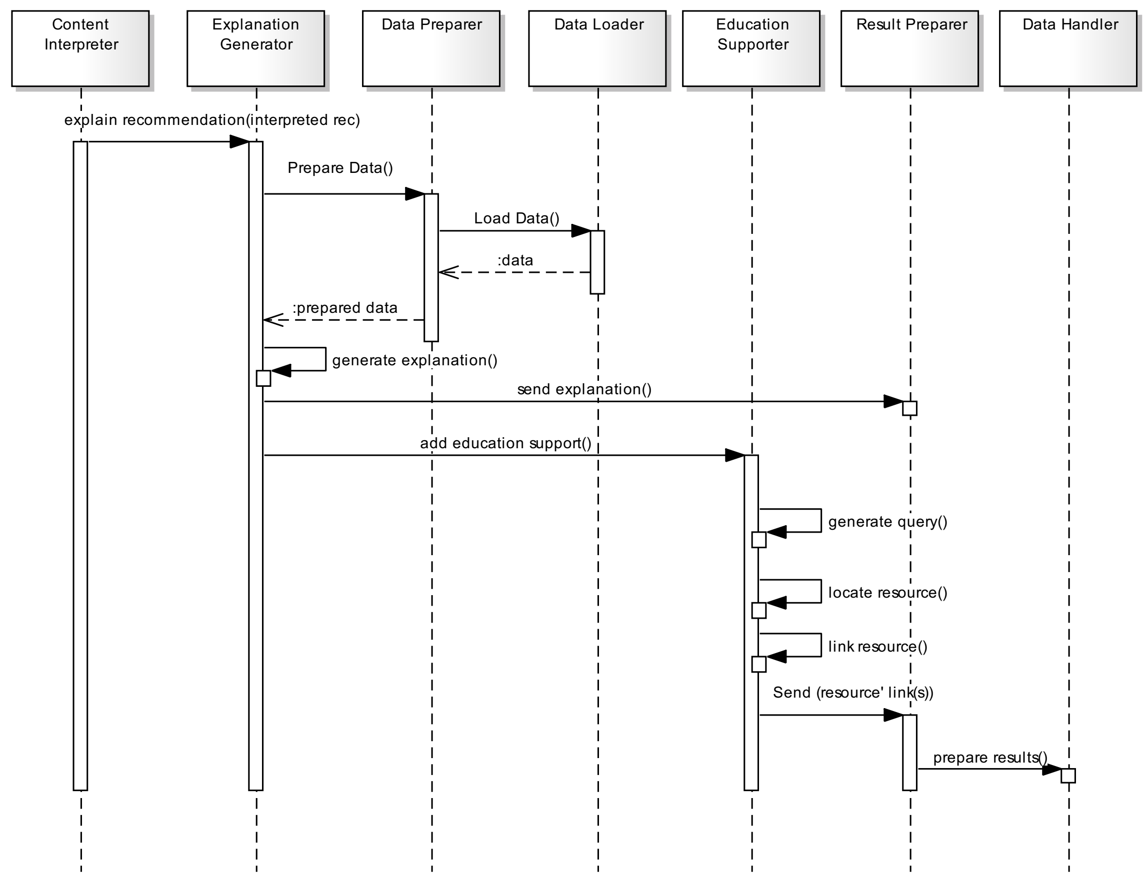

Context Interpreter
Content Interpreter
Data Preparer
Data Loader
Explanation Generator
interprete content(interpreted context, recommendation)
loop
select filter()
Prepare Data(selected filter)
Load Data()
:data
:prepared data
apply filter()
Prepare Data(user id)
load data (user id)
:data
Prepare Data()
:prepared data
select format()
pick content url()
Explain Recommendation(interpreted rec)

Use Case ID:	*SCL2-SUC-11*		
Use Case Name:	Explain Recommendations		
Created By:	Muhammad Afzal	**Last Updated By:**	Rahman Ali
Date Created:	15 July 2015	**Last Revision Date:**	20 July 2015
Actors:	Interpret content (SCL2-SUC-10)		
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.		

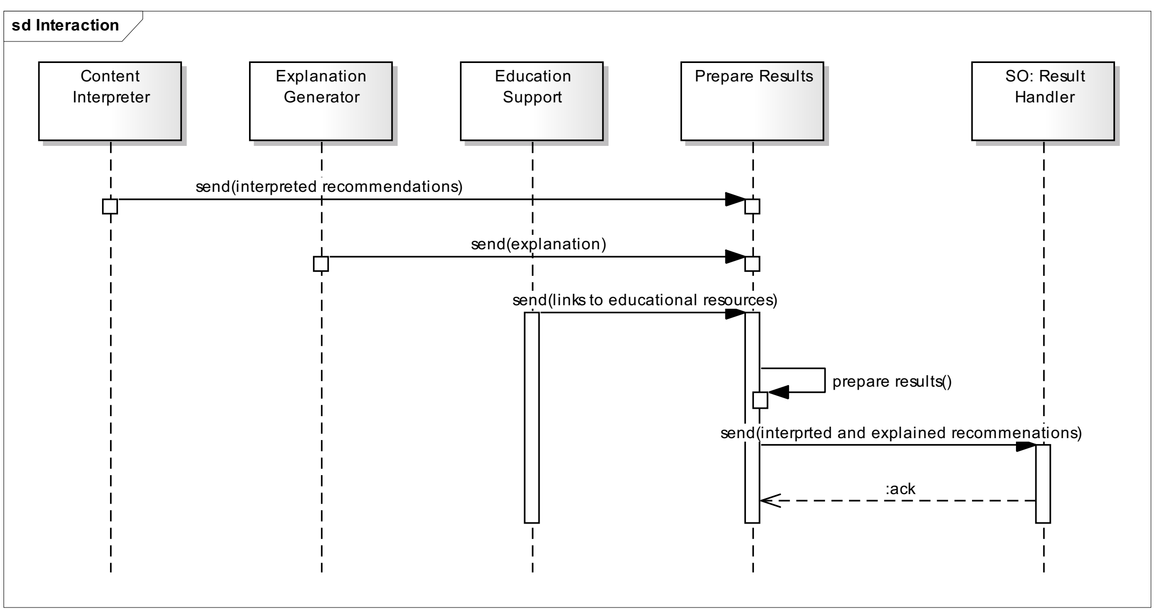
Alternative Flows:	N/A		
Exceptions:	N/A		
Includes:	SCL2-SUC-08		
Frequency of Use:	Very frequent: when recommendations are interpreted		
Special Requirements:	N/A		
Assumptions:	N/A		

Notes and Issues:	N/A
Sequence Diagram:	



Use Case ID:	*SCL2-SUC-12*		
Use Case Name:	Prepare Results		
Created By:	Muhammad Afzal	**Last Updated By:**	Rahman Ali, Muhammad Afzal
Date Created:	5 July 2015	**Last Revision Date:**	20 July 2015
Actors:	Explain Recommendation (SCL2-SUC-11), Interpret Contents (SCL2-SUC-10)		
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		

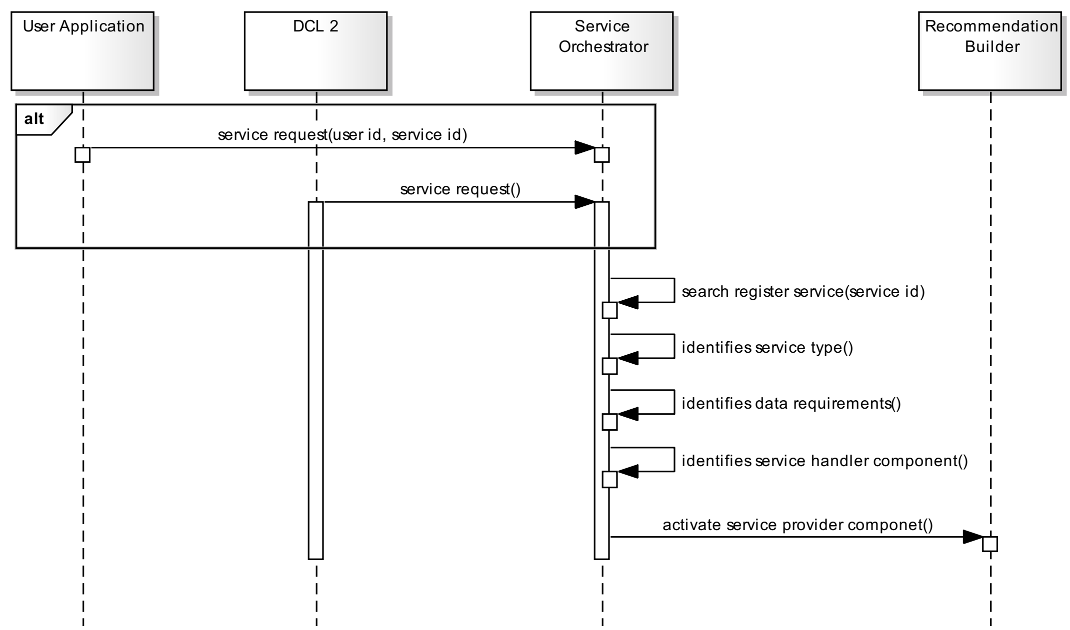
Preconditions:	The recommendation are interpreted and explained
Postconditions:	The results are forwarded to service orchestrator.
Normal Flow:	1. Result Preparer receives outputs from content interpreter and/or explanation generator as well as education support. 2. Result Preparer combines the received 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-SUC-13*

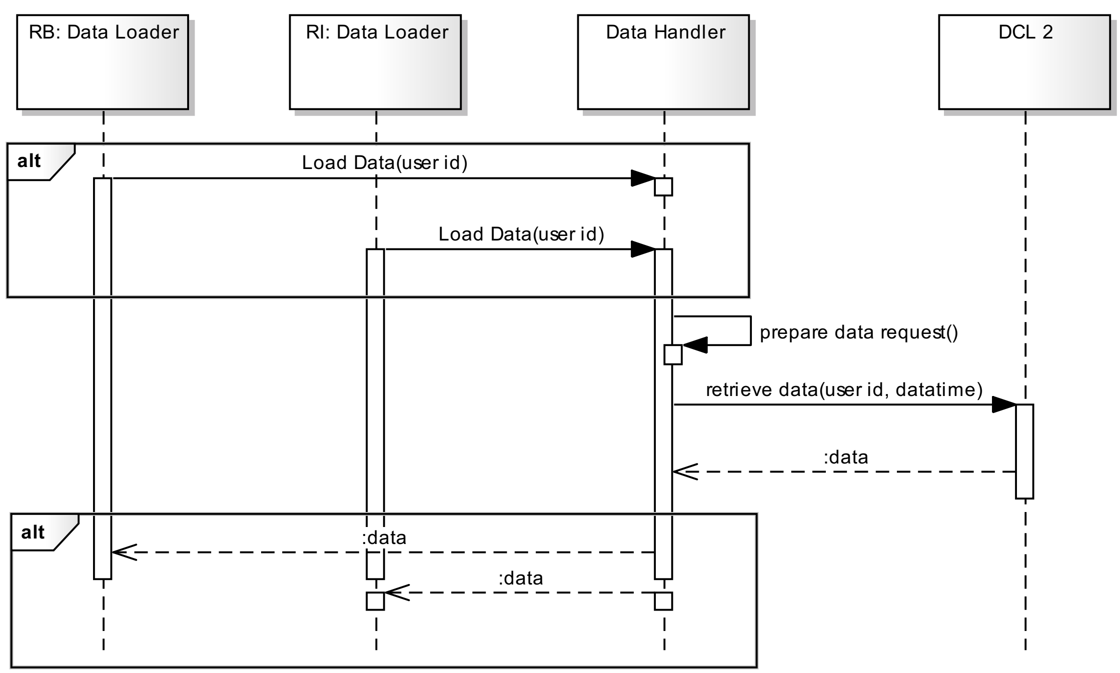
<table>
<tr><td>Use Case Name:</td><td colspan="3">Receive service request</td></tr>
<tr><td>Created By:</td><td>Muhammad Afzal</td><td>Last Updated By:</td><td>Rahman Ali, Muhammad Afzal</td></tr>
<tr><td>Date Created:</td><td>5 July 2015</td><td>Last Revision Date:</td><td>20 July 2015</td></tr>
<tr><td>Actors:</td><td colspan="3">User Application / SL 2</td></tr>
<tr><td>Description:</td><td colspan="3">Service orchestrator receives request from user application, or DCL 2 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">Service orchestrator receives the service request from user application
System parses the request<ol type="a">
Search for the registered service
Identifies the service type
Identifies data requirements of the service
Identifies the appropriate handling module
Service Orchestrator passes the request to recommendation builder of SCL 2 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, 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</td><td colspan="3">N/A</td></tr>
</table>

Requirements:	
Assumptions:	Service orchestrator and DCL 2 agreed on service contract.
Notes and Issues:	
Sequence Diagram:	



Use Case ID:	*SCL2-SUC-14*		
Use Case Name:	Handle Data		
Created By:	Muhammad Afzal	**Last Updated By:**	Muhammad Afzal
Date Created:	28 July 2015	**Last Revision Date:**	28 July 2015
Actors:	DCL 2		
Description:	This use case receives data request from recommendation builder and recommendation interpreter. It makes request from DCL 2 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		

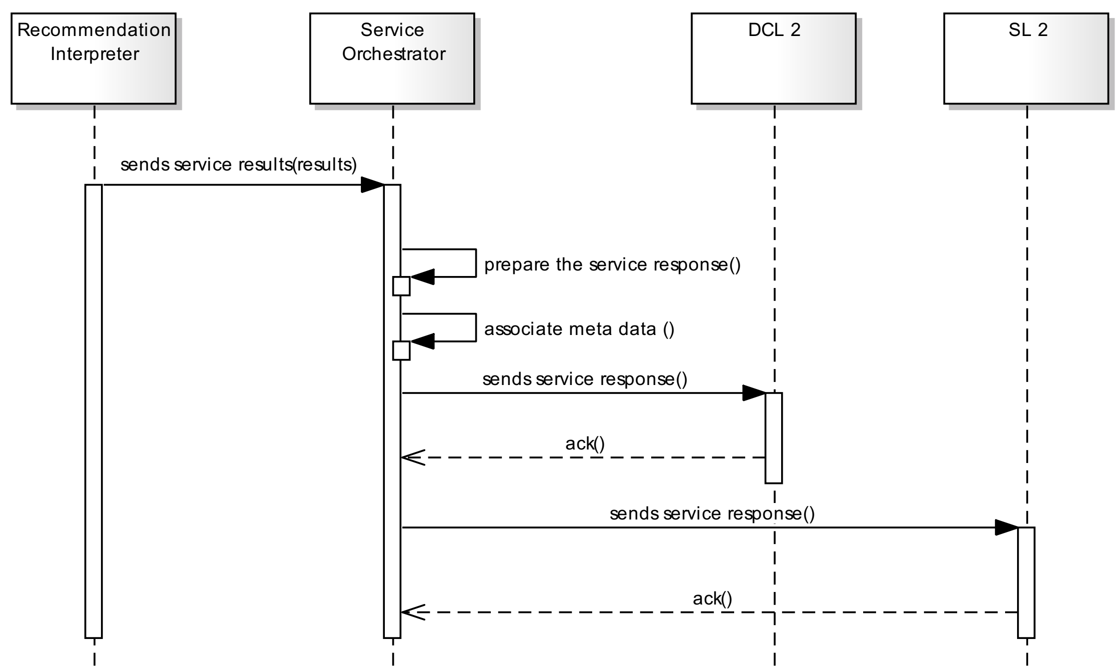
	2. Prepare data request 3. Retrieve data from DCL 2 4. Send data to RB: Data Loader
Alternative Flows:	1a. Data Handler in Service Orchestrator received data loading request from recommendation interpreter Step 2-3 of normal flow 4a. Send data to RI: Data Loader
Exceptions:	N/A
Includes:	N/A
Frequency 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-SUC-15*
Use Case Name:	Deliver service results

Created By:	Muhammad Afzal	**Last Updated By:**	Rahman Ali, Muhammad Afzal
Date Created:	5 July 2015	**Last Revision Date:**	20 July 2015
Actors:	Application / SL 2		
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 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 for persistence 4. Service Orchestrator receives acknowledgement of storage 5. Service Orchestrator sends interpreted recommendations to SL 2 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 and user application.		

Notes and Issues:	N/A
Sequence Diagram:	



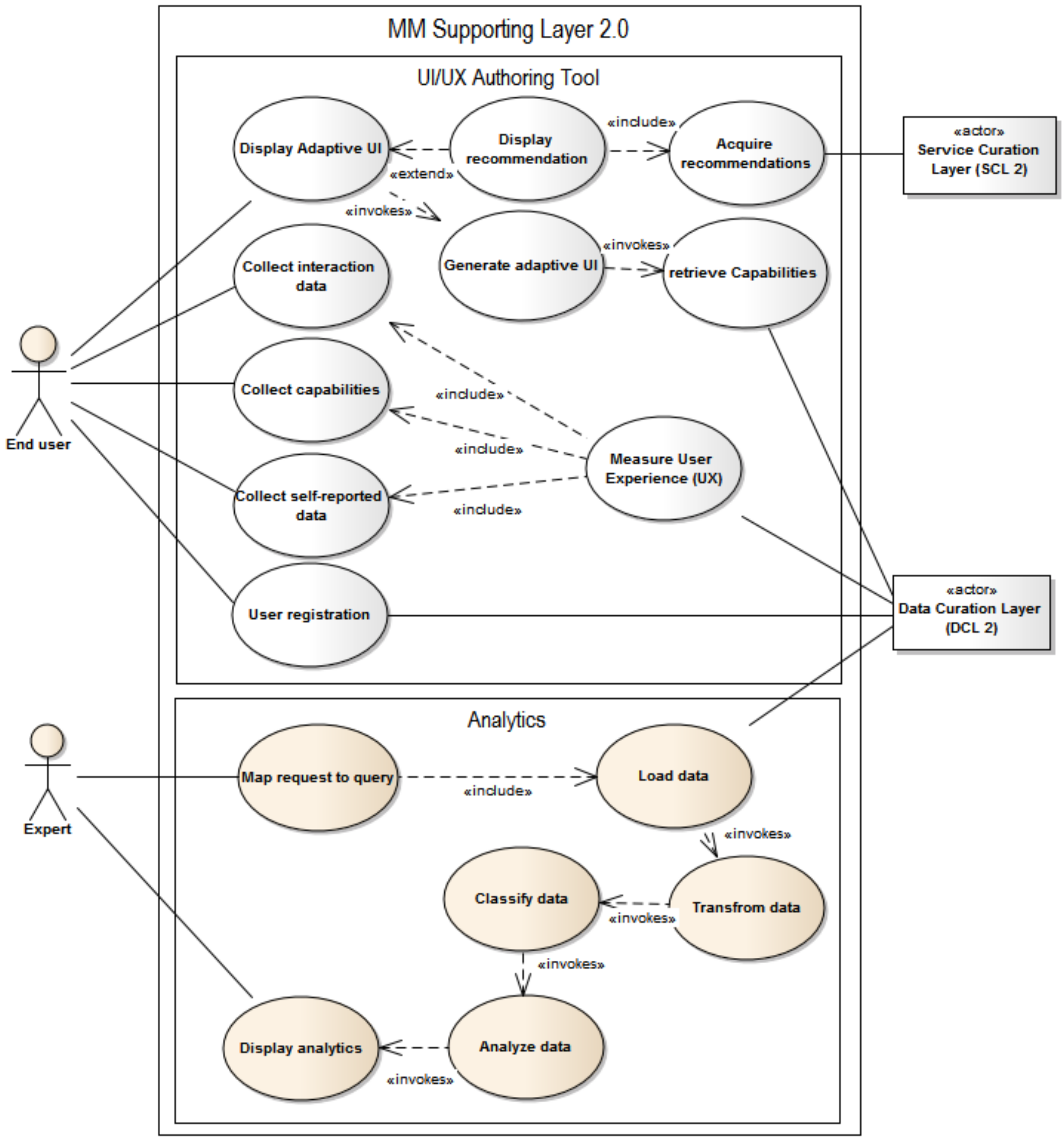
Supporting Layer (SL Ver. 2)

System Level Use cases

List of Use cases

Use case ID#	Name
SL2-UCS-01	User registration
SL2-UCS-02	Retrieve capabilities for user interface adaptation
SL2-UCS-03	Mapping the user capability information into model
SL2-UCS-04	Adapt user interface based on user profile, context and device
SL2-UCS-05	User capabilities collection
SL2-UCS-06	Self-reporting user experience measurement
SL2-UCS-07	Collect and analyze observational data
SL2-UCS-08	Acquire Recommendations for displaying to end user
SL2-UCS-09	Map Request to Query
SL2-UCS-10	Transform Data
SL2-UCS-11	Classify Data
SL2-UCS-12	Analyze Data
SL2-UCS-13	Display Analytics

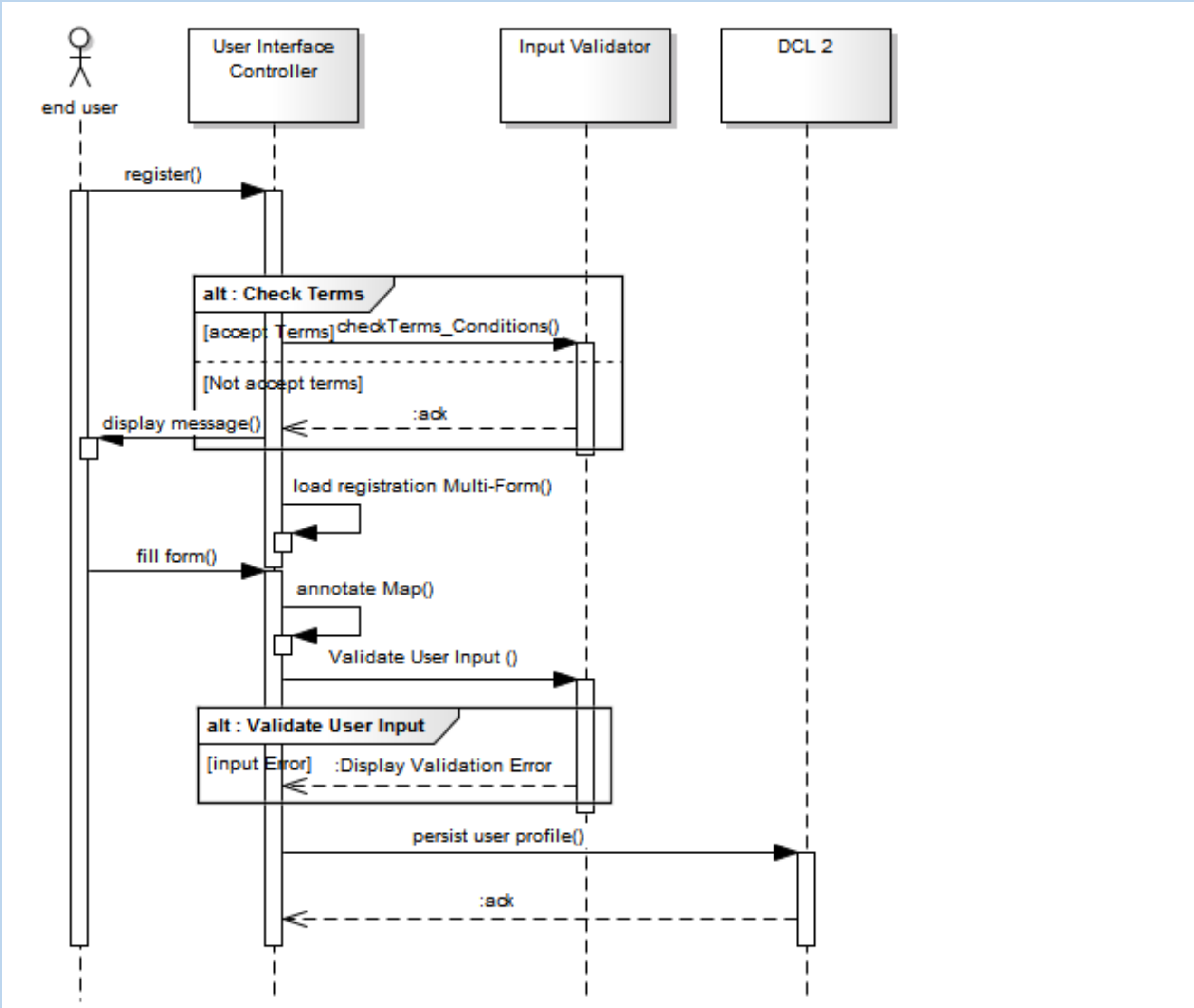
Use case Diagram



Use case Description

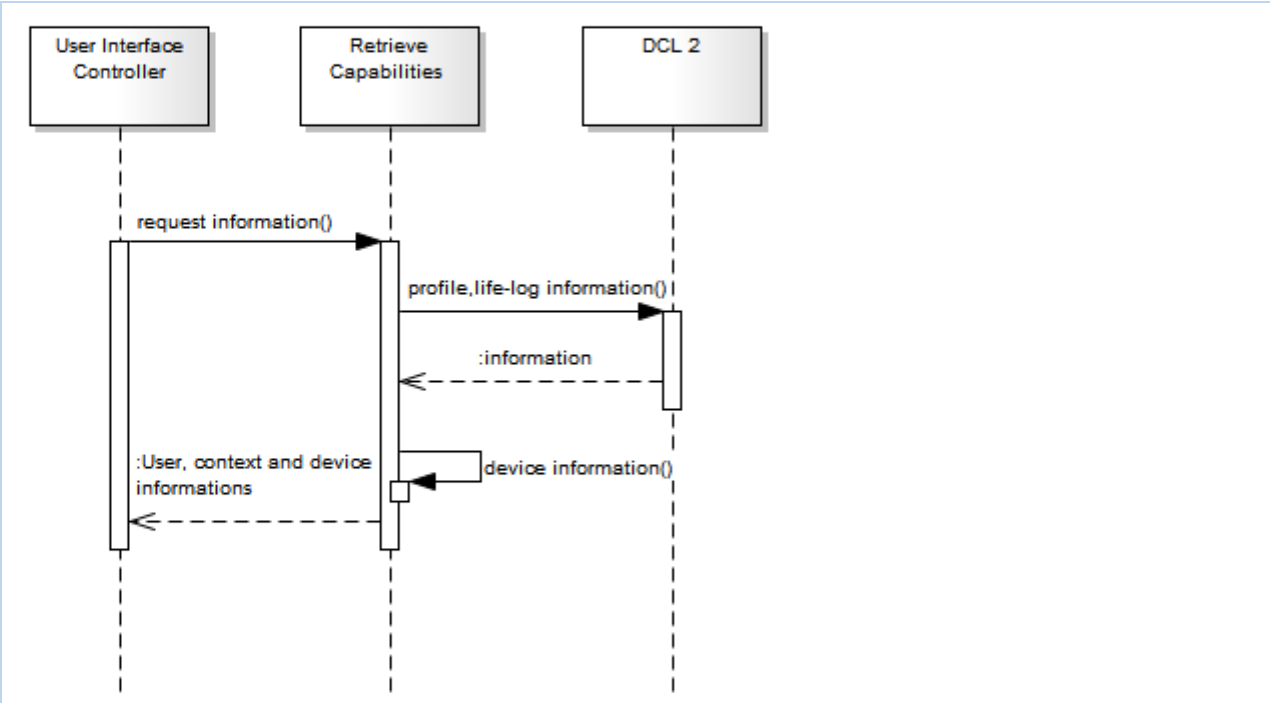
<table>
<tr><th>*Use Case ID:*</th><th colspan="3">*SL2-UCS-01*</th></tr>
<tr><td>**Use Case Name:**</td><td colspan="3">User registration</td></tr>
<tr><td>**Created By:**</td><td>Jamil Husain</td><td>**Last Updated By:**</td><td>Wajahat Ali Khan</td></tr>
<tr><td>**Date Created:**</td><td>14 July 2015</td><td>**Last Revision Date:**</td><td>15 July 2015</td></tr>
<tr><td>**Actors:**</td><td colspan="3">Primary: End-user
Secondary: DCL 2</td></tr>
<tr><td>**Description:**</td><td colspan="3">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.</td></tr>
<tr><td>**Trigger:**</td><td colspan="3">End user</td></tr>
<tr><td>**Preconditions:**</td><td colspan="3">The non-register user asked the application to register to it.</td></tr>
<tr><td>**Postconditions:**</td><td colspan="3">The user successfully registered to the application and can access its functionality</td></tr>
<tr><td>**Normal Flow:**</td><td colspan="3">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.</td></tr>
<tr><td>**Alternative Flows:**</td><td colspan="3">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</td></tr>
</table>

	• 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:	



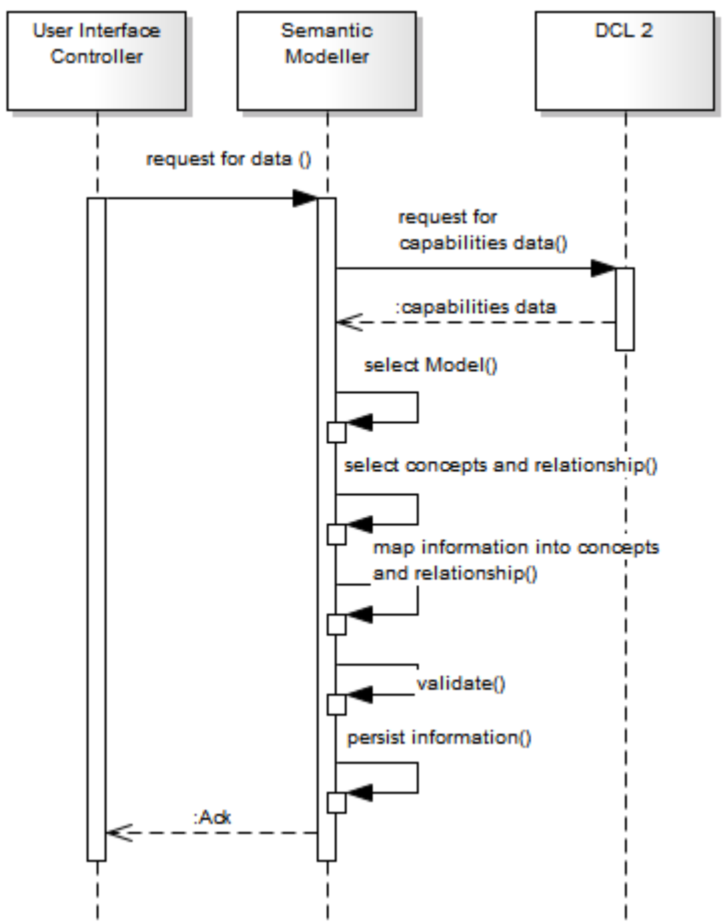
Use Case ID:	*SL2-UCS-02*		
Use Case Name:	Retrieve capabilities for user interface adaptation		
Created By:	Jamil Hussain	**Last Updated By:**	Wajahat Ali Khan
Date Created:	14 July 2015	**Last Revision Date:**	14 July 2015
Actors:	DCL 2		
Description:	This use case focuses on the retrieval of the capabilities for user interface adaptation. The capabilities includes user profile information, context information and device information. It is utilized for adaptation based on changes or observational data.		

Trigger:	
Preconditions:	The DCL 2 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 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:	

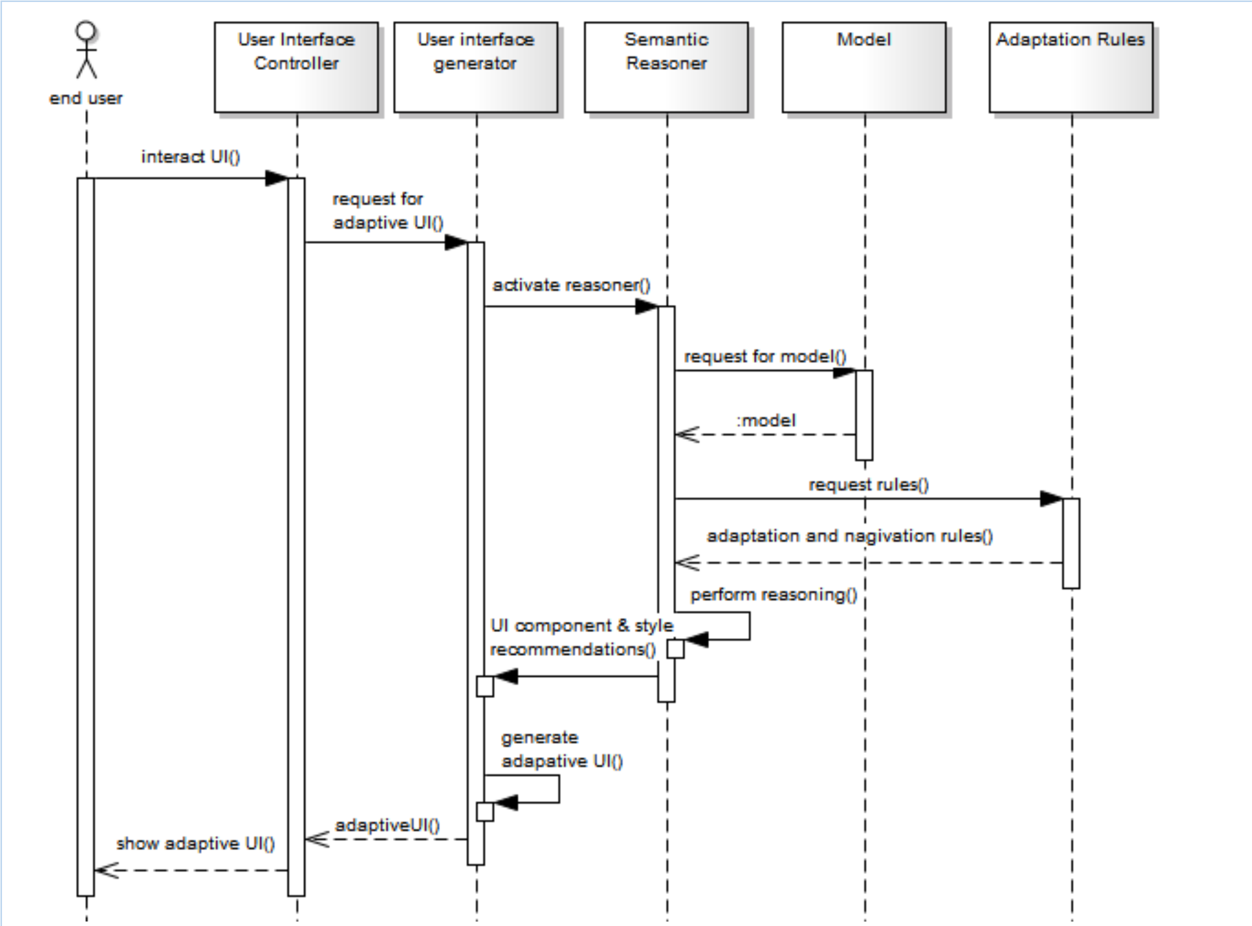


Use Case ID:	*SL2-UCS-03*		
Use Case Name:	Mapping the user capability information into model		
Created By:	Jamil Hussain	**Last Updated By:**	Wajahat Ali Khan
Date Created:	14 July 2015	**Last Revision Date:**	14 July 2015
Actors:	DCL 2		
Description:	The collected capabilities information from DCL 2.0 shall be mapped against the hierarchical structure of the model		
Trigger:	SL 2 initiate communication with DCL 2.		
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 2.0 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 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-UCS-01
Frequency of Use:	Always when the application is running [High]
Assumptions:	N/A
Notes and Issues:	N/A
Sequence Diagram:	

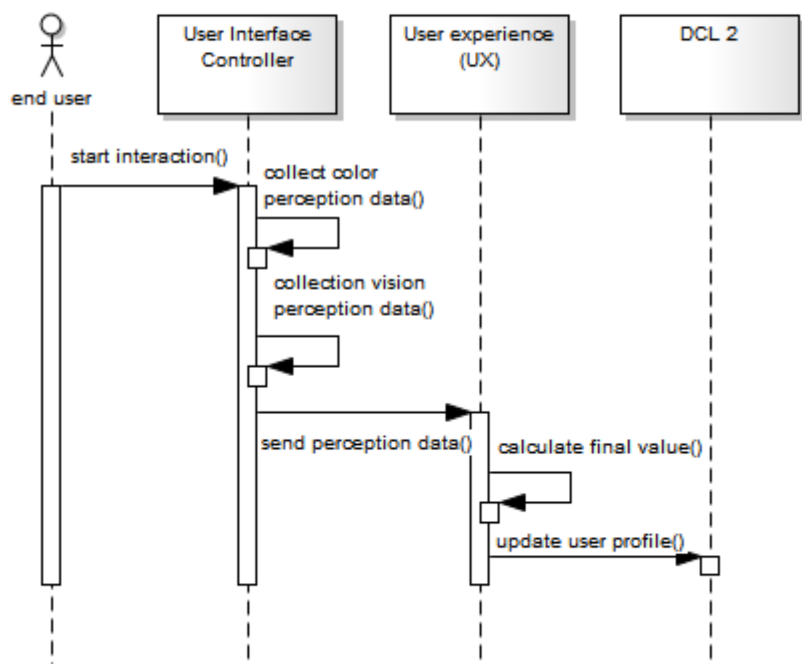


Use Case ID:	*SL2-UCS-04*		
Use Case Name:	Adapt user interface based on user profile, context and device		
Created By:	Jamil Hussain	**Last Updated By:**	Wajahat Ali Khan
Date Created:	14 July 2015	**Last Revision Date:**	15 July 2015
Actors:	DCL 2, end user		
Description:	The collected information of user profile, context of use and device characteristics from DCL 2 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:			

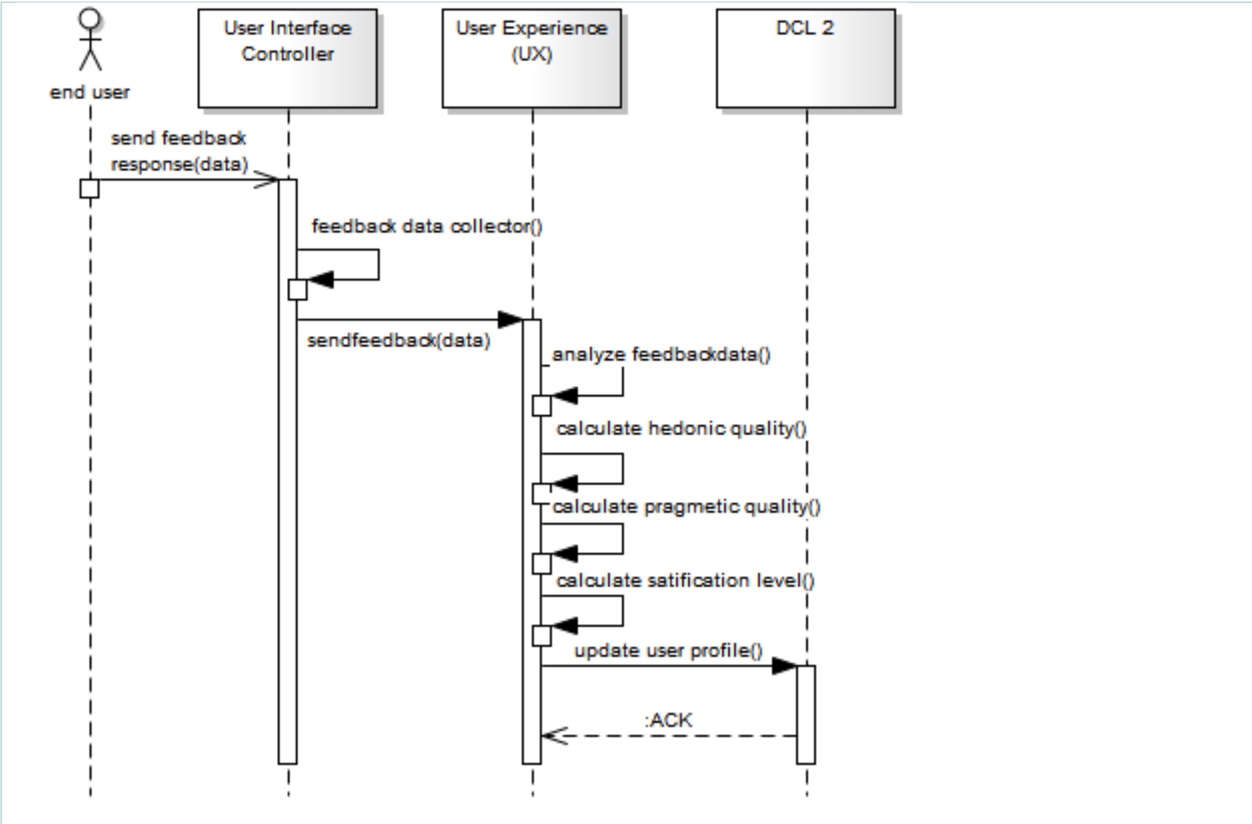


Use Case ID:	*SL2-UCS-05*		
Use Case Name:	User capabilities collection		
Created By:	Jamil Hussain	**Last Updated By:**	Wajahat Ali Khan
Date Created:	14 July 2015	**Last Revision Date:**	15 July 2015
Actors:	End user, DCL 2		
Description:	The UI/UX collects the user capabilities information's by analyzing 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		

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.
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 and DCL is defined
Notes and Issues:	N/A
Sequence Diagram:	

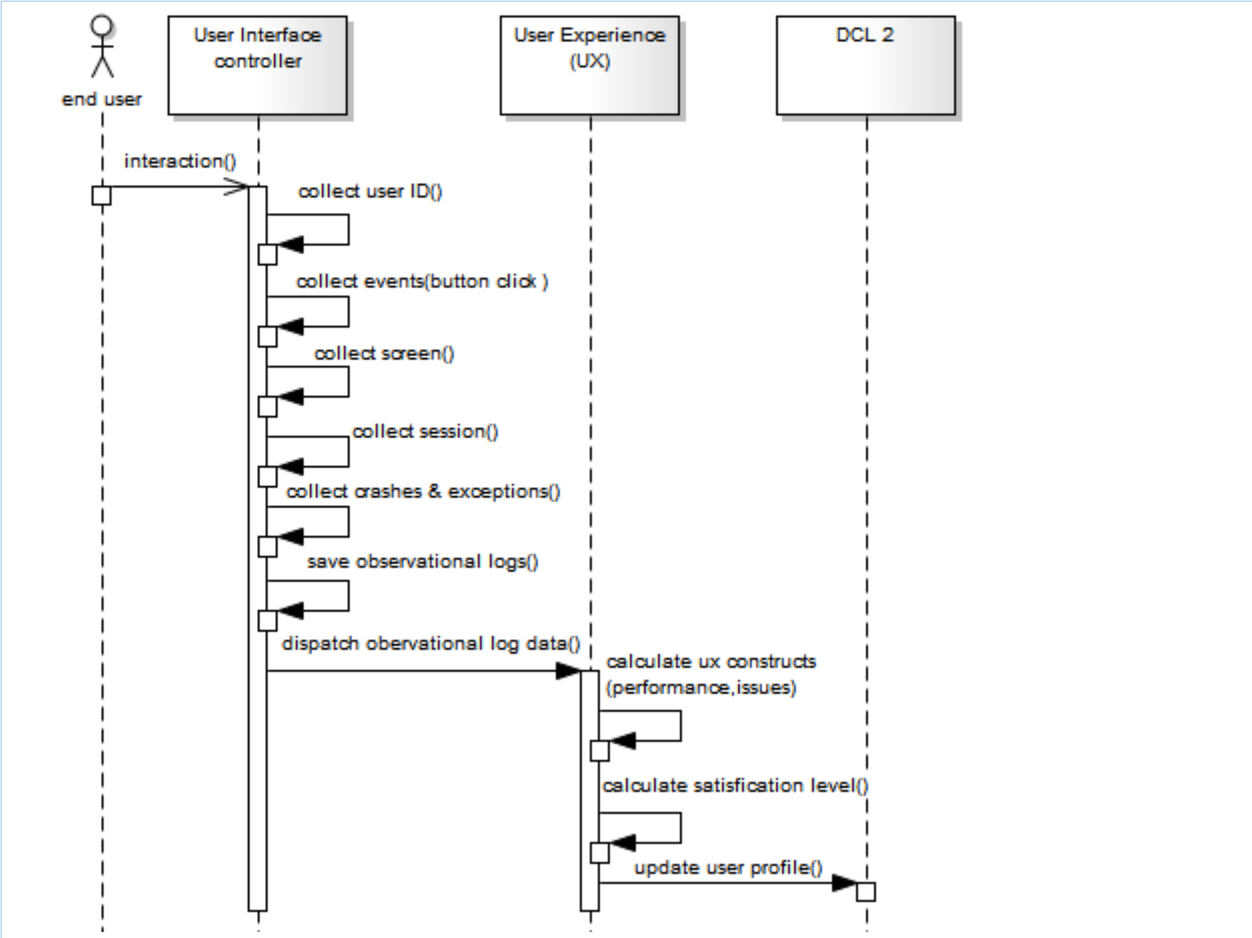


Use Case ID:	*SL2-UCS-06*		
Use Case Name:	Self-reporting user experience measurement		
Created By:	Jamil Hussain	**Last Updated By:**	Wajahat Ali Khan
Date Created:	14 July 2015	**Last Revision Date:**	14 July 2015
Actors:	End user, DCL 2		
Description:	The UI/UX shall collect feedback about how users 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		
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:			



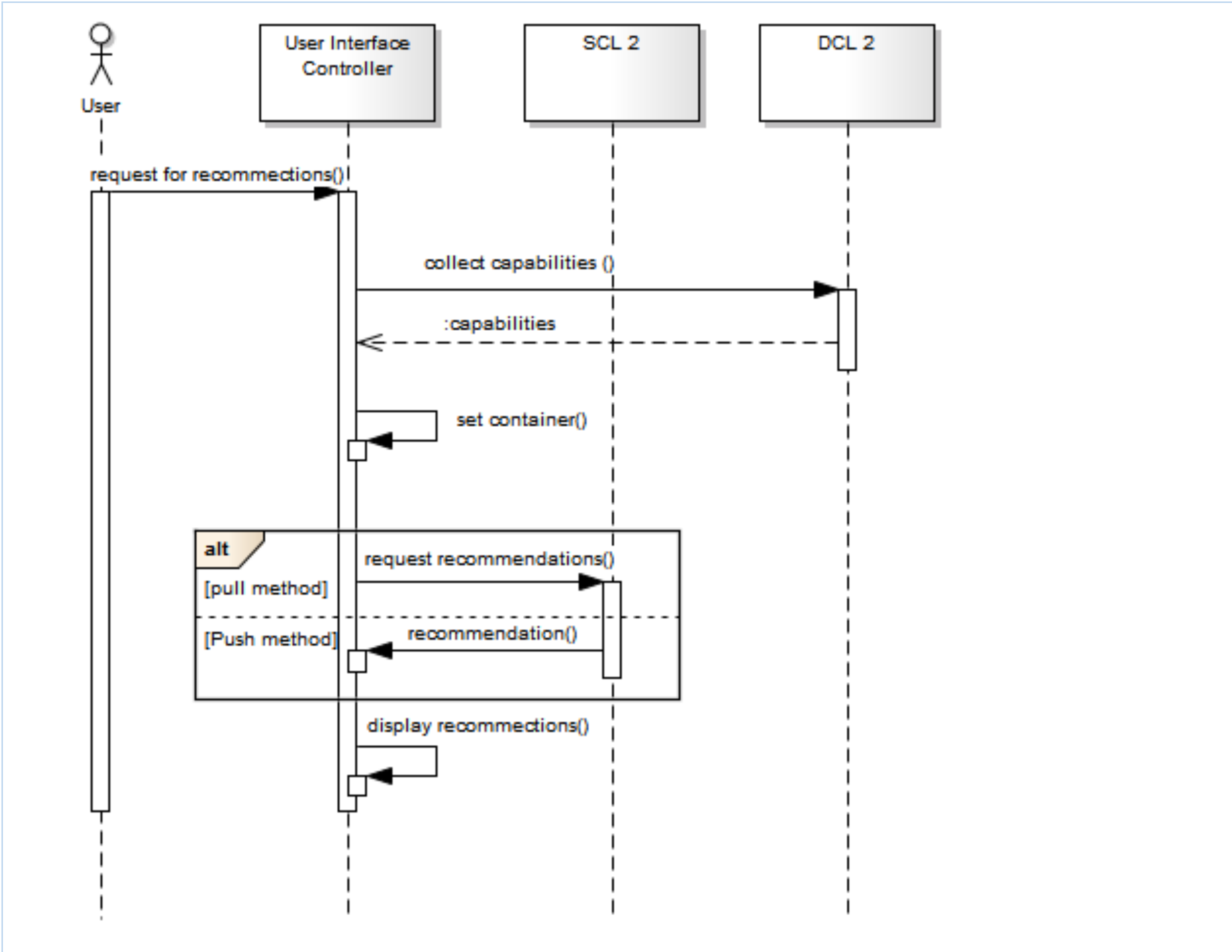
Use Case ID:	*SL2-UCS-07*		
Use Case Name:	Collect and analyze observational data		
Created By:	Jamil Hussain	**Last Updated By:**	Wajahat Ali Khan
Date Created:	14 July 2015	**Last Revision Date:**	15 July 2015
Actors:	End user, DCL 2		
Description:	The UI/UX shall identify areas of improvement and 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 DCL2 for storage/updating in user profile.
Alternative Flows:	N/A
Exceptions:	N/A
Includes:	N/A
Frequency of Use:	Frequent, request by SL 2
Special Requirements:	N/A
Assumptions:	Service contract between DCL 2 and SL 2 is defined
Notes and Issues:	N/A
Sequence Diagram:	



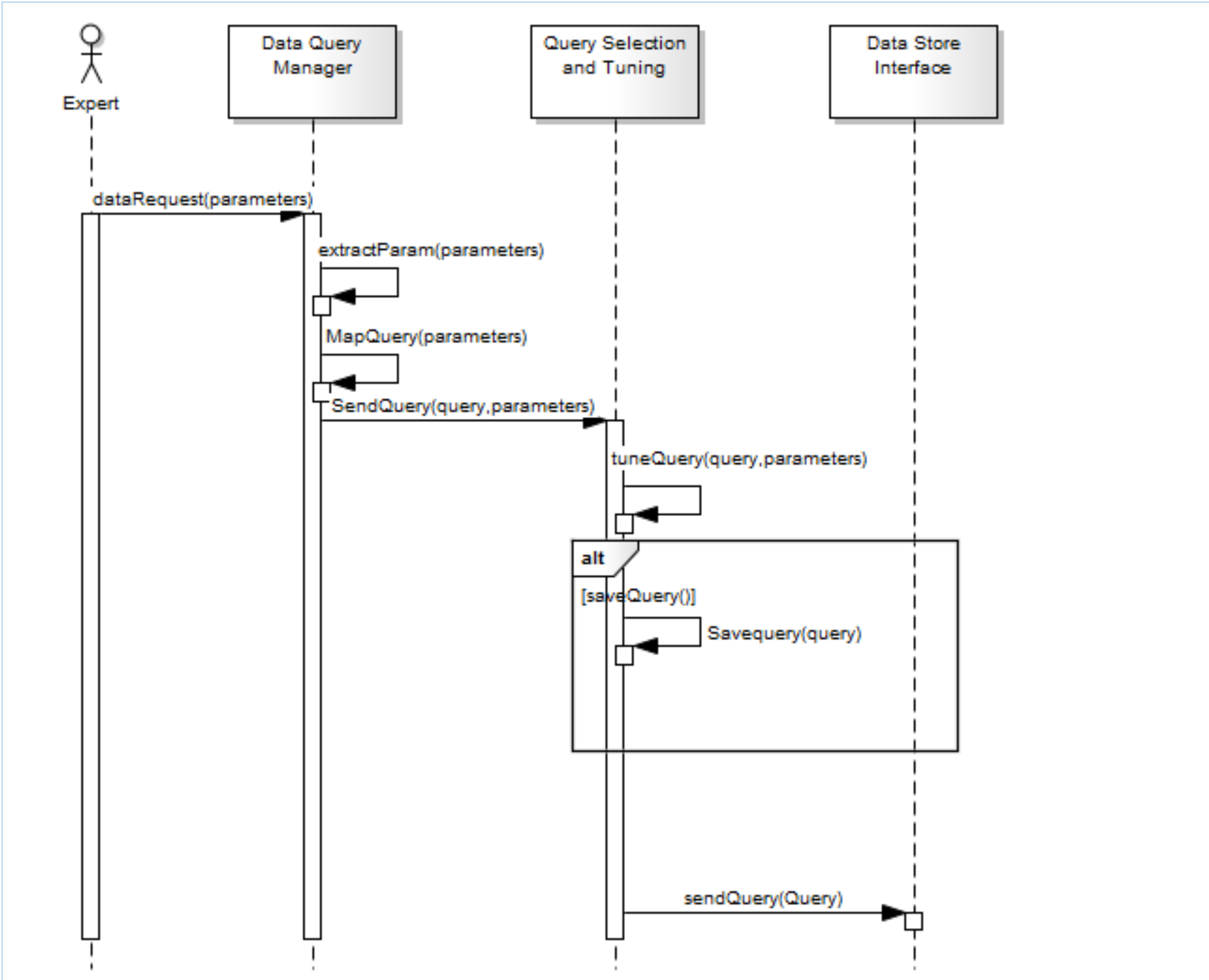
Use Case ID:	*SL2-UCS-08*		
Use Case Name:	Acquire Recommendations for displaying to end user		
Created By:	Jamil Husain	**Last Updated By:**	Wajahat Ali Khan
Date Created:	14 July 2015	**Last Revision Date:**	15 July 2015
Actors:	Primary: End-user Secondary: SCL 2, DCL 2		
Description:	This use case collects the recommendations generated by SCL 2 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.		
Trigger:	SCL 2 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 generate the recommendations and provide it to user interface 2. The SCL 2 recommendations are acquired by the SL 3. SL investigates the user capabilities, context of use, and device characteristics by obtaining from DCL 2 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 recommendations are acquired by the SL 2 1. user request for recommendations (pull method) 2b. In step 2. The SCL 2 recommendations are acquired by the SL 2 1. SL 2 push recommendations to App based on situations
Exceptions:	N/A
Includes:	N/A
Frequency of Use:	Whenever the recommendations are generated by SCL [Medium]
Special Requirements:	N/A
Assumptions:	The user profile data and context information should exist in the DCL 2
Notes and Issues:	N/A
Sequence Diagram:	



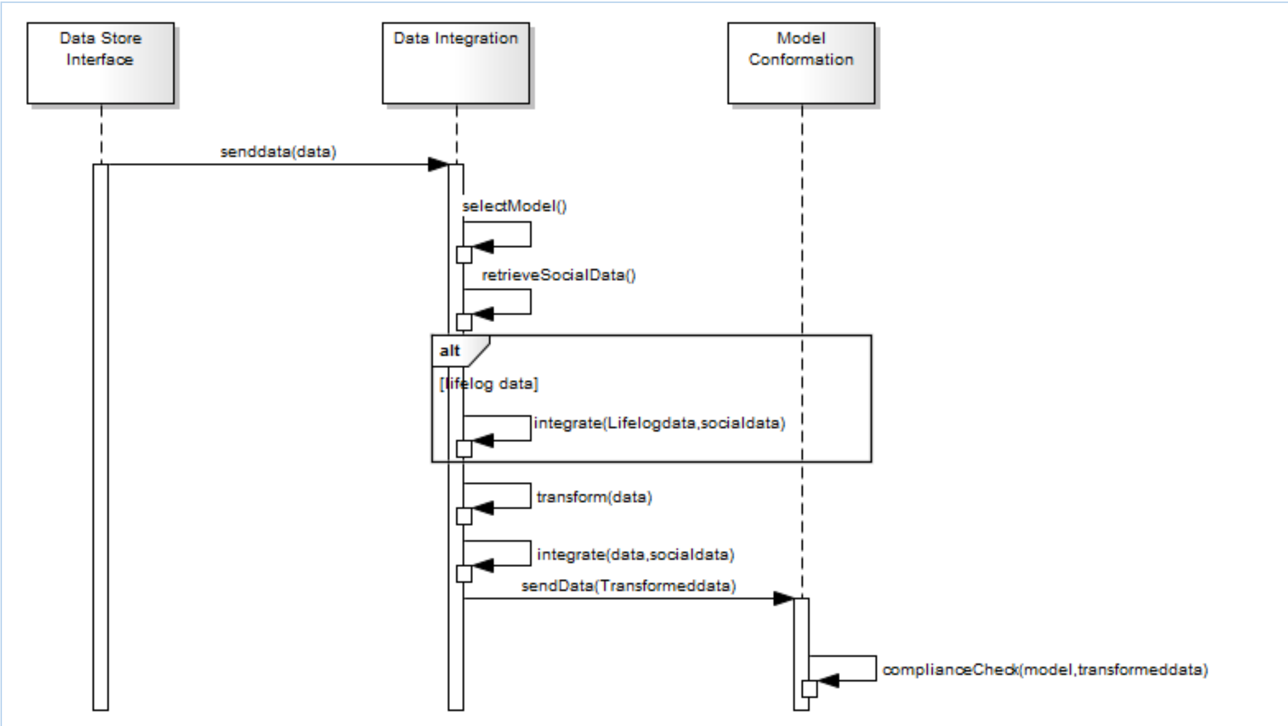
Use Case ID:	SL2-UCS-09		
Use Case Name:	Map Request to Query		
Created By:	Shujaat Hussain	**Last Updated By:**	Wajahat Ali Khan
Date Created:	14 July 2015	**Last Revision Date:**	15 July 2015
Actors:	Primary: Expert		
Description:	This use case focuses on mapping the expert request to the query library 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:	



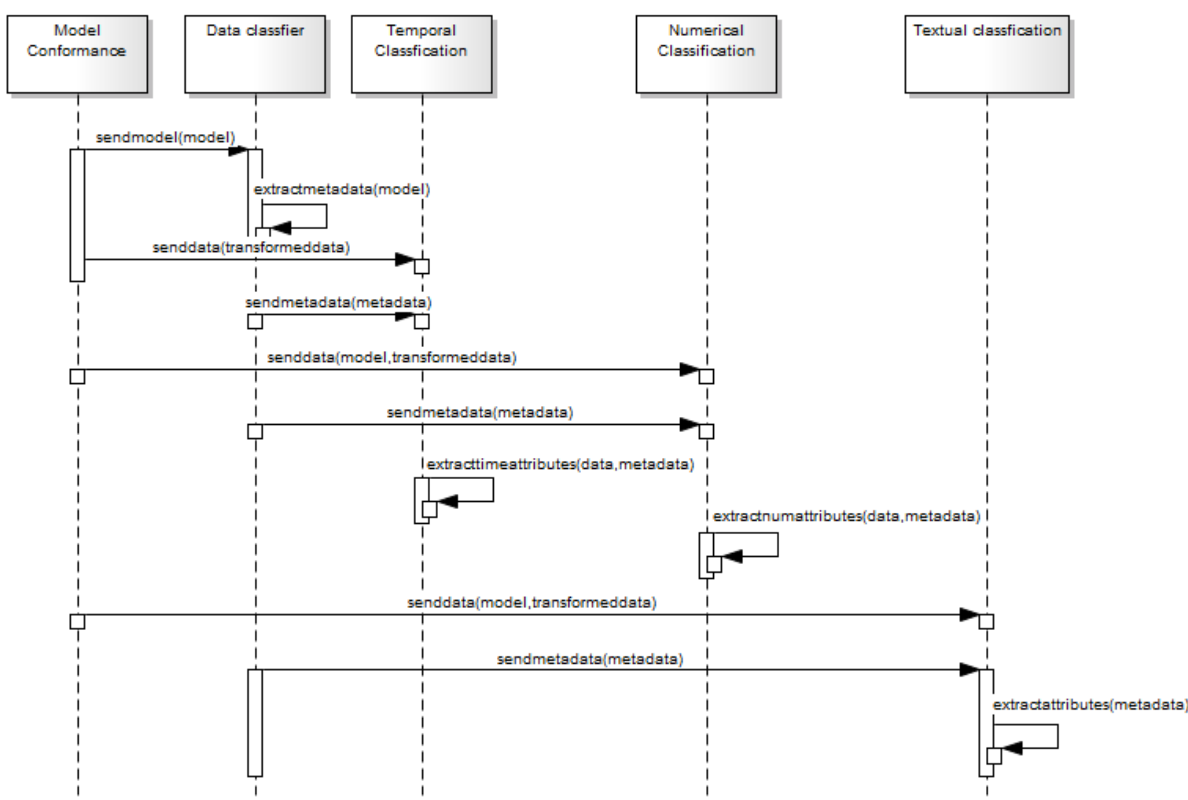
Use Case ID:	*SL2-UCS-10*		
Use Case Name:	Transform Data		
Created By:	Shujaat Hussain	**Last Updated By:**	Wajahat Ali Khan
Date Created:	14 July 2015	**Last Revision Date:**	15 July 2015
Actors:	Primary: Expert		
Description:	The mapping query is transformed to specific model structure for trend analysis.		
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:	



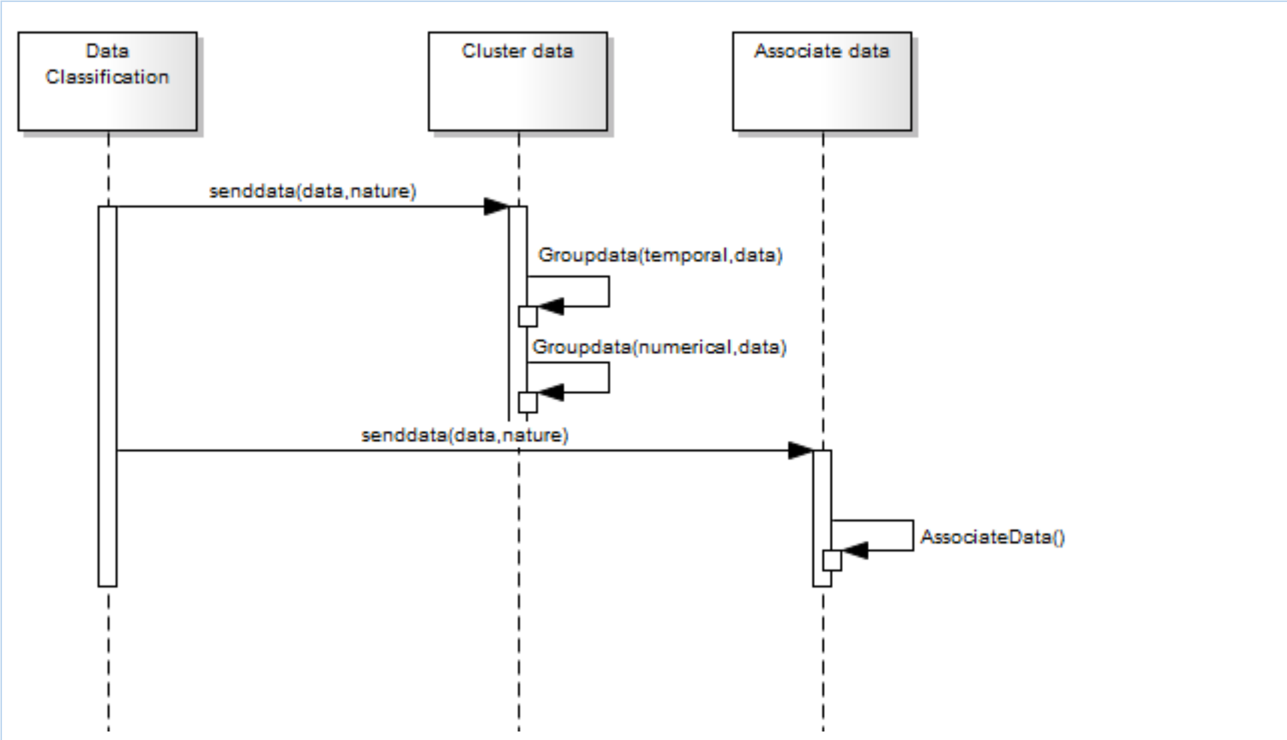
Use Case ID:	*SL2-UCS-11*		
Use Case Name:	Classify Data		
Created By:	Shujaat Hussain	**Last Updated By:**	Wajahat Ali Khan
Date Created:	14 July 2015	**Last Revision Date:**	15 July 2015
Actors:	Primary: Expert		
Description:	The transformed data is further classified and clustered to identify and 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:	



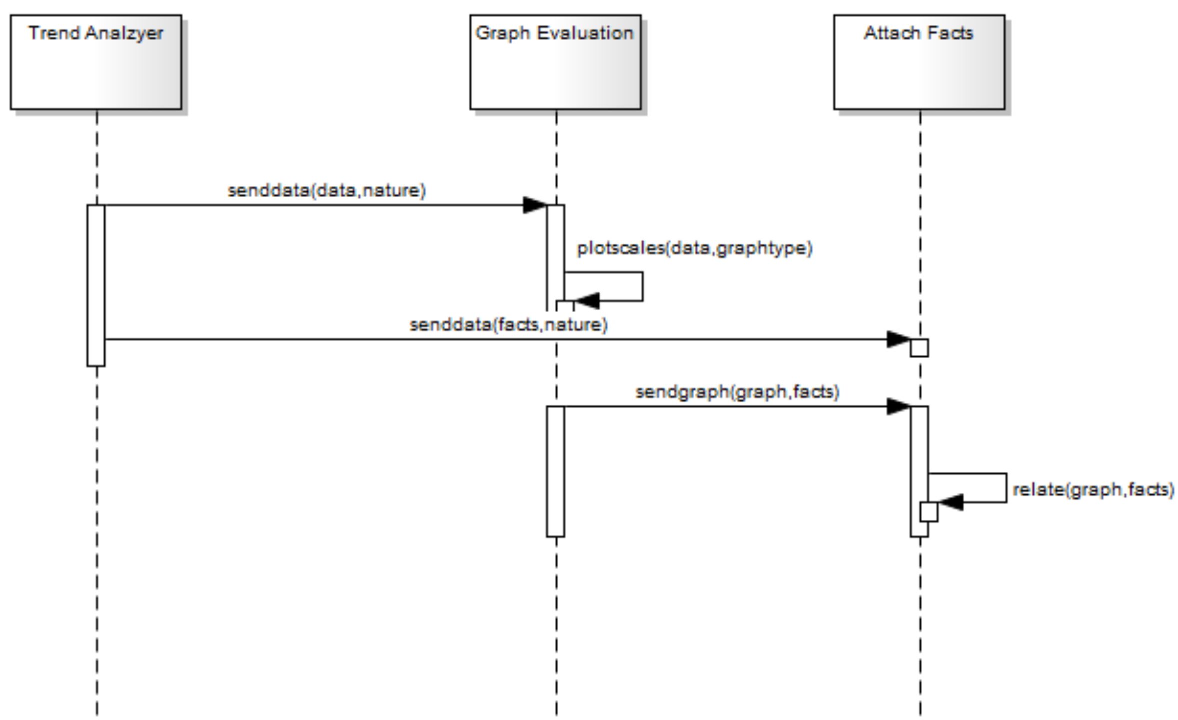
Use Case ID:	*SL2-UCS-12*		
Use Case Name:	Analyze Data		
Created By:	Shujaat	**Last Updated By:**	Wajahat Ali Khan

	Hussain		
Date Created:	14 July 2015	**Last Revision Date:**	15 July 2015
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:			

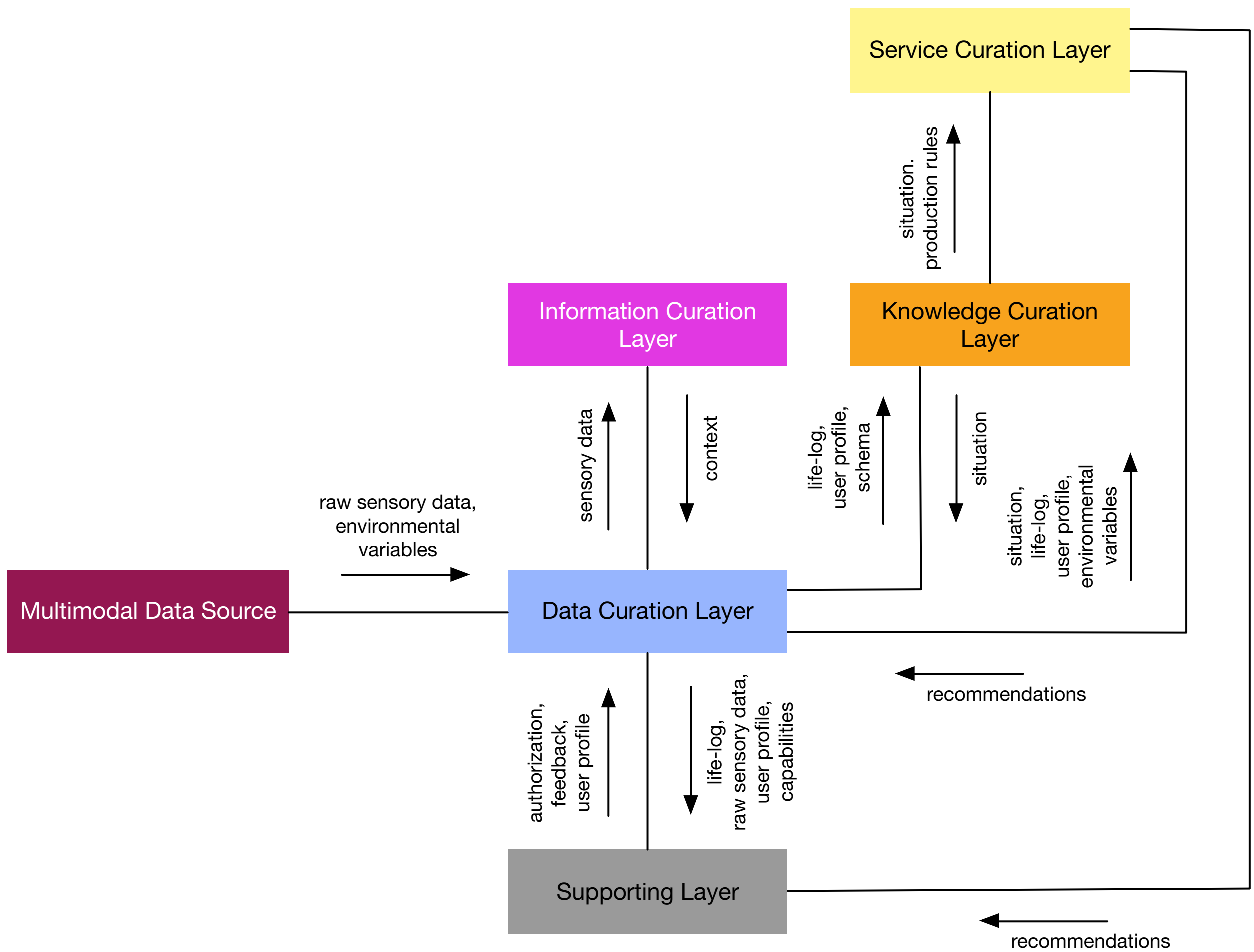


Use Case ID:	*SL2-UCS-13*		
Use Case Name:	Display Analytics		
Created By:	Shujaat Hussain	**Last Updated By:**	Wajahat Ali Khan
Date Created:	14 July 2015	**Last Revision Date:**	15 July 2015
Actors:	Primary: Expert		
Description:	The grouped data and relevant analytics is passed to visualization enabler 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:	



Collaboration Diagram



Section 4-C

System Specifications

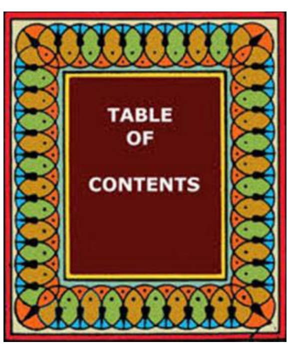
Mining Minds Platform

Principal Investigator:
Professor Sungyoung Lee

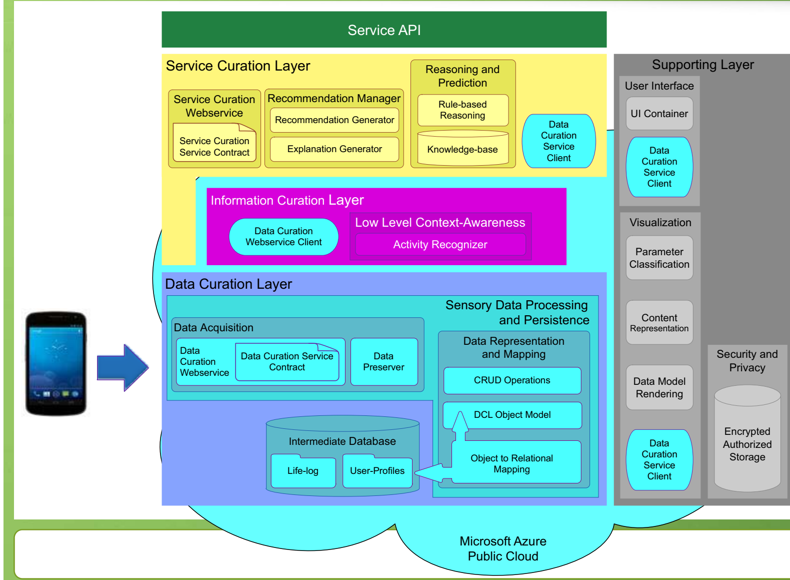
Mining Minds Version 2.0
Comprehensive Specifications

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

Contents

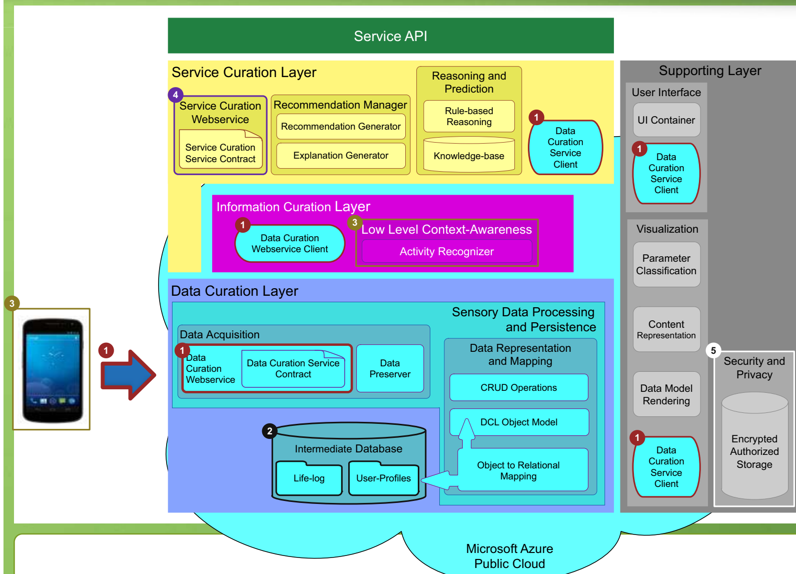


MM Ver. 1.0 (Release Jan 2015)



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)



Technological Limitations

1 Poor Communication Performance due to SOAP-based Web services

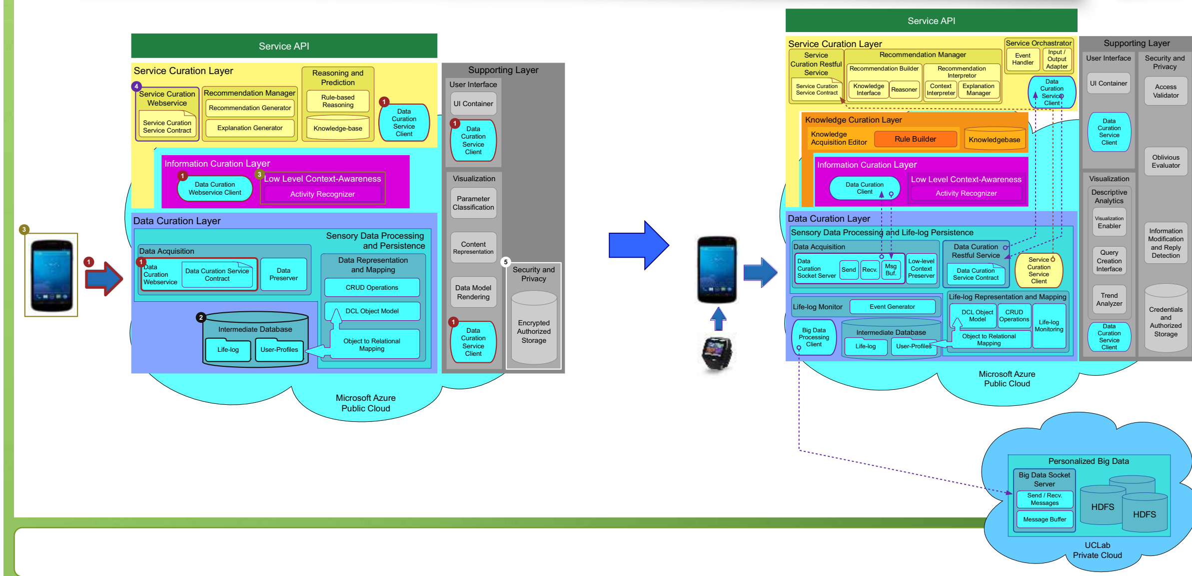
2 Stress on RDBMS due to excess of data storage and persistence

3 Primitive amount of Activities were identified (5 only)

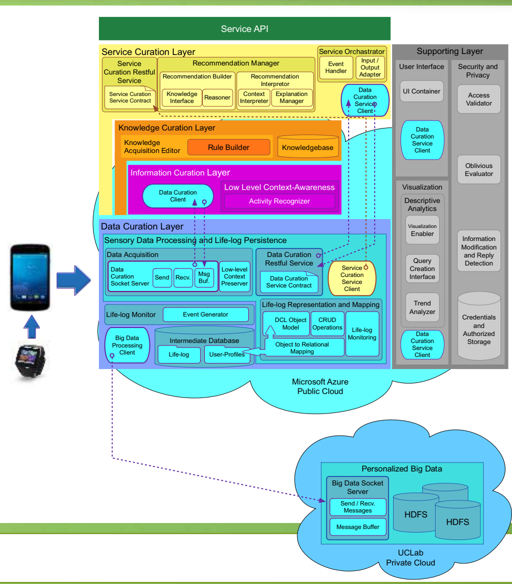
4 Request-based Recommendation generation

5 Message integrity can be compromised during communication

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

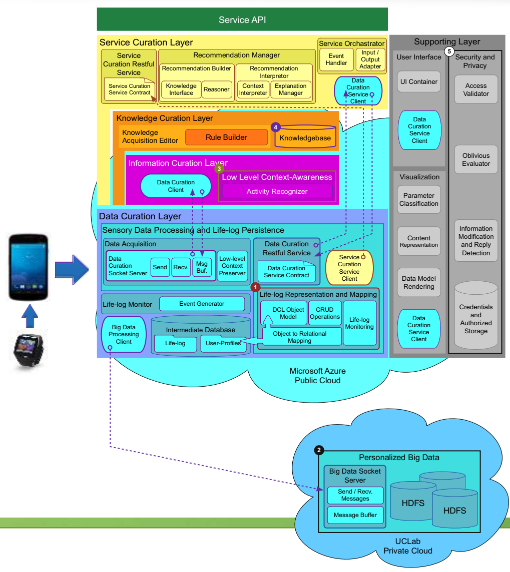


MM Ver. 1.5 (Release Apr 2015)



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

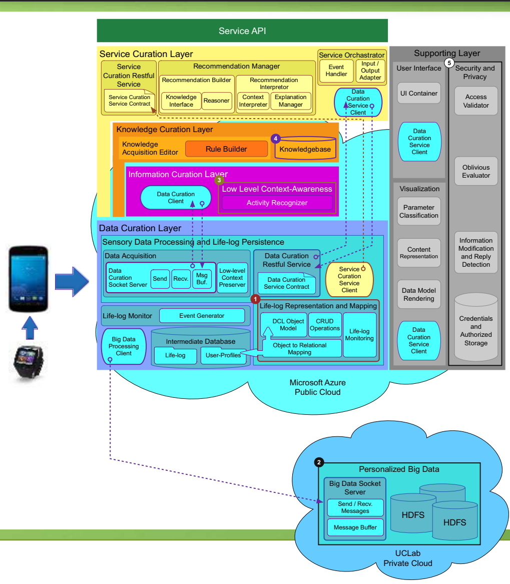
MM Ver. 1.5 (Release Apr 2015)

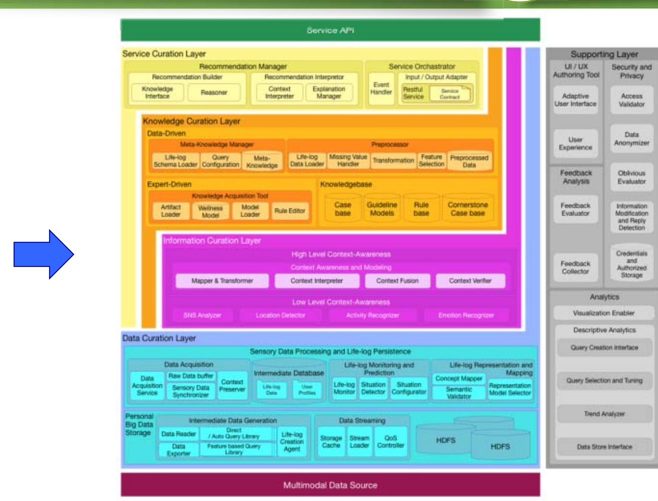


Limitations

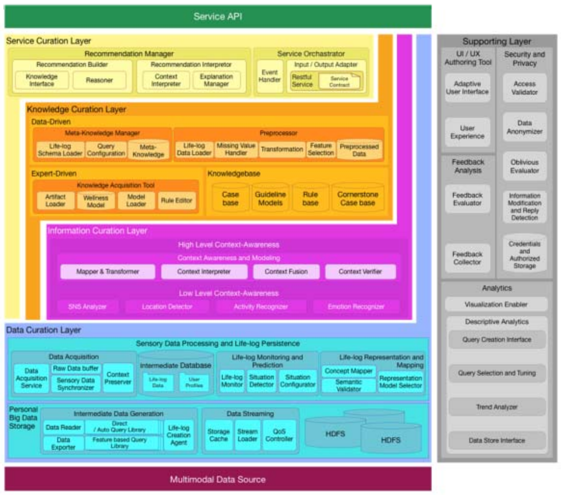
1. Object Oriented and Relational life-log models
2. Big data is only used for persistence
3. Lack of High-level context determination
4. Only static rules as knowledge
5. Anonymized Access

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





MM Ver. 2.0 (Release Dec 2015)



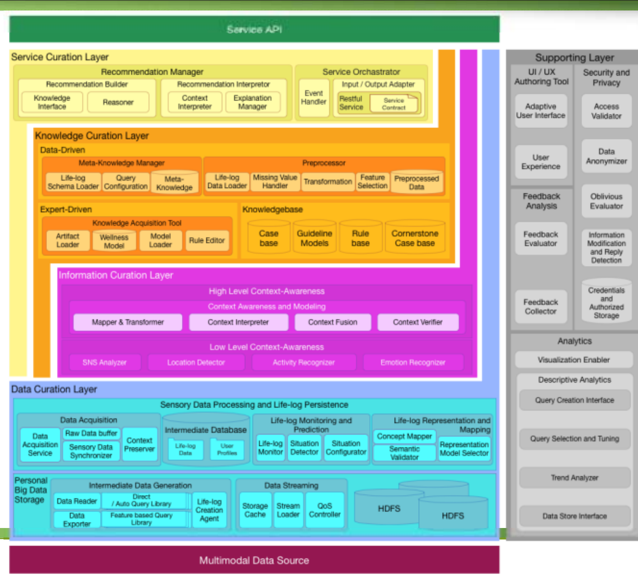
1. Input Data : Sensors (Smartphone-based Accelerometer & GPS, Wearable device) + SNS

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. Feedback Analysis
    4. Personalized Predictions
    5. Descriptive Analytics
    6. User Experience (UX)

Architecture of Mining Minds Version 2.0



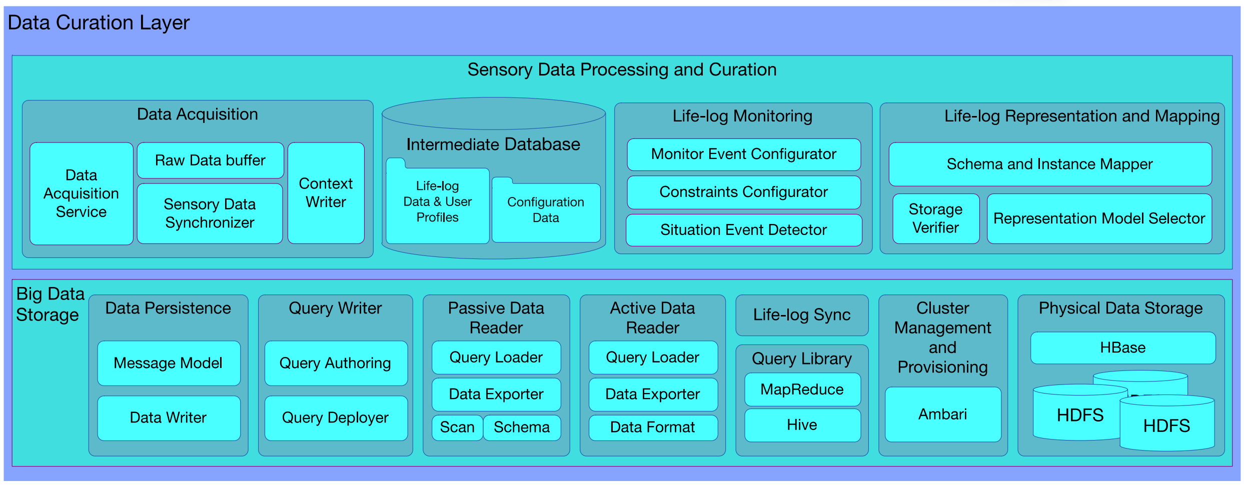
Data Curation Layer (DCL 2.0)

Principal Investigator:
Professor Sungyoung Lee

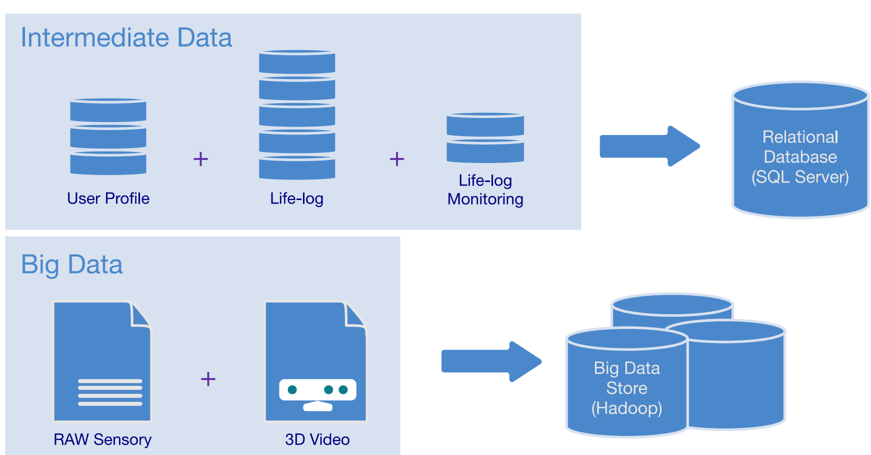
DCL 2.0 Responsibilities

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

DCL V2.0 Architecture

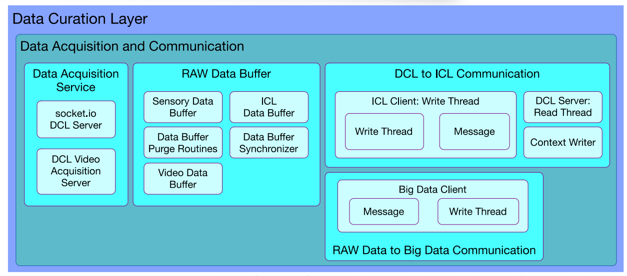


DCL 2.0 High Level Architecture

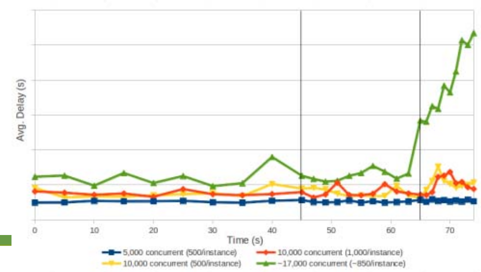


Data Acquisition and Communication Overview

- 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



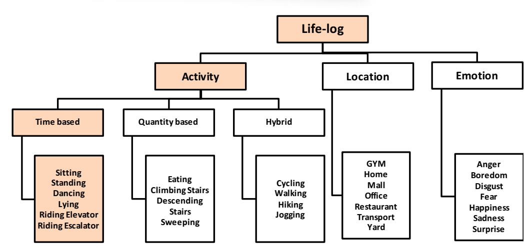
- How is it different from Ver. 1.5 ?
  - More scalable and Concurrent with state-of-the-art Non-blocking Communication IO
  - Substantially less communication overhead with improved resource consumption
  - Data Synchronization and Buffer for heterogeneous data streams (audio, video, activities data,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



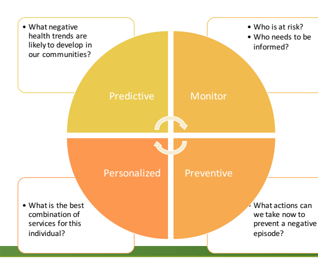
경희대학교
KYUNG HEE UNIVERSITY

Life-log Monitoring and Prediction Overview

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



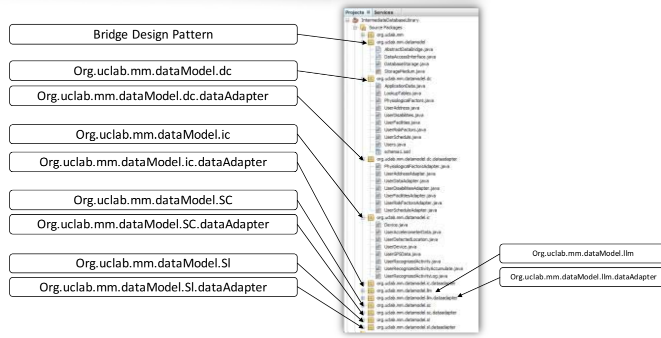
- How is it different from Ver. 1.5 ?
  - Agile Static Situation: Static situations are not hard-coded, they are represented, hosted and executed
  - Dynamic Situations can now be incorporated in the LLM Execution



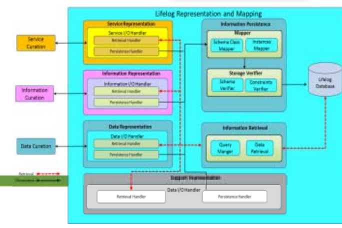
경희대학교
KYUNG HEE UNIVERSITY

Life-log Representation and Mapping Overview

- 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 ?
  - **"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

Data Curation Layer (DCL 2.0)

M. Bilal Amin

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

Introduction

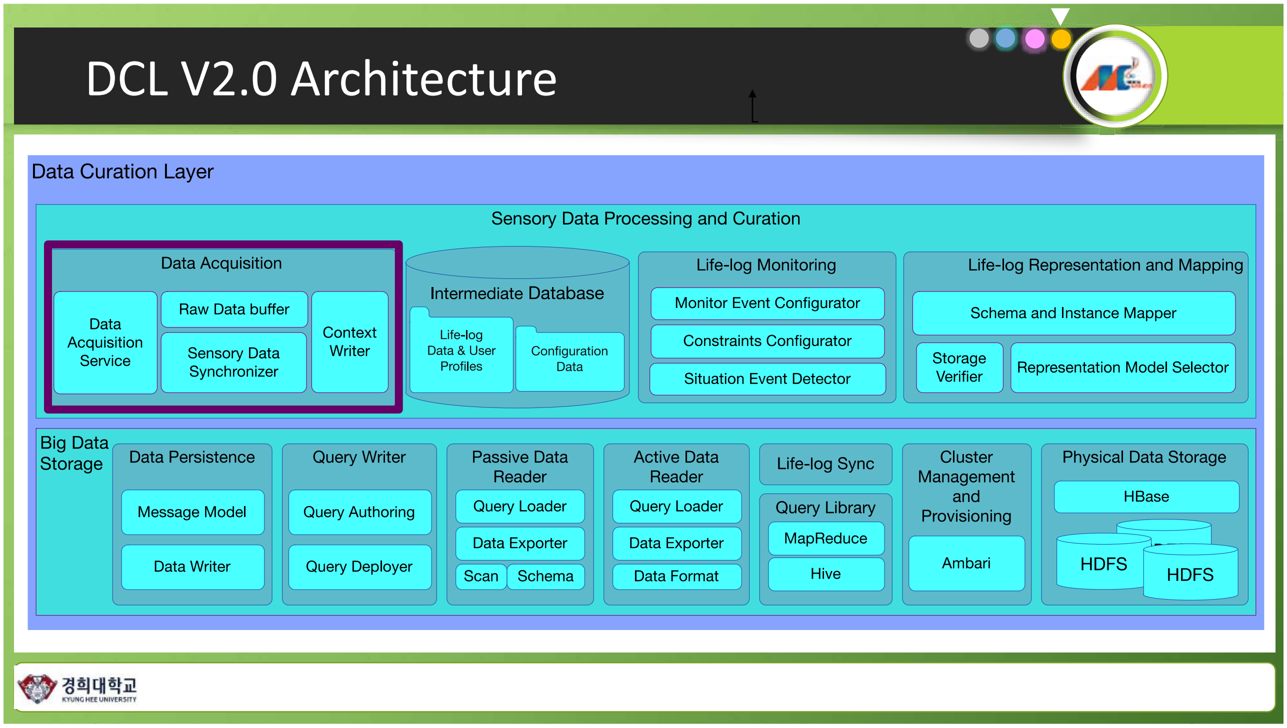
- Acquisition of heterogeneous data from Multi-modal data sources in *Real-time* is a must for data curation layer
- This acquisition of data must be *dynamic, parallel* and of *high performance* to support the influx of multimodal data at real-time
- For reliable acquisition of data at real-time, *optimal 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

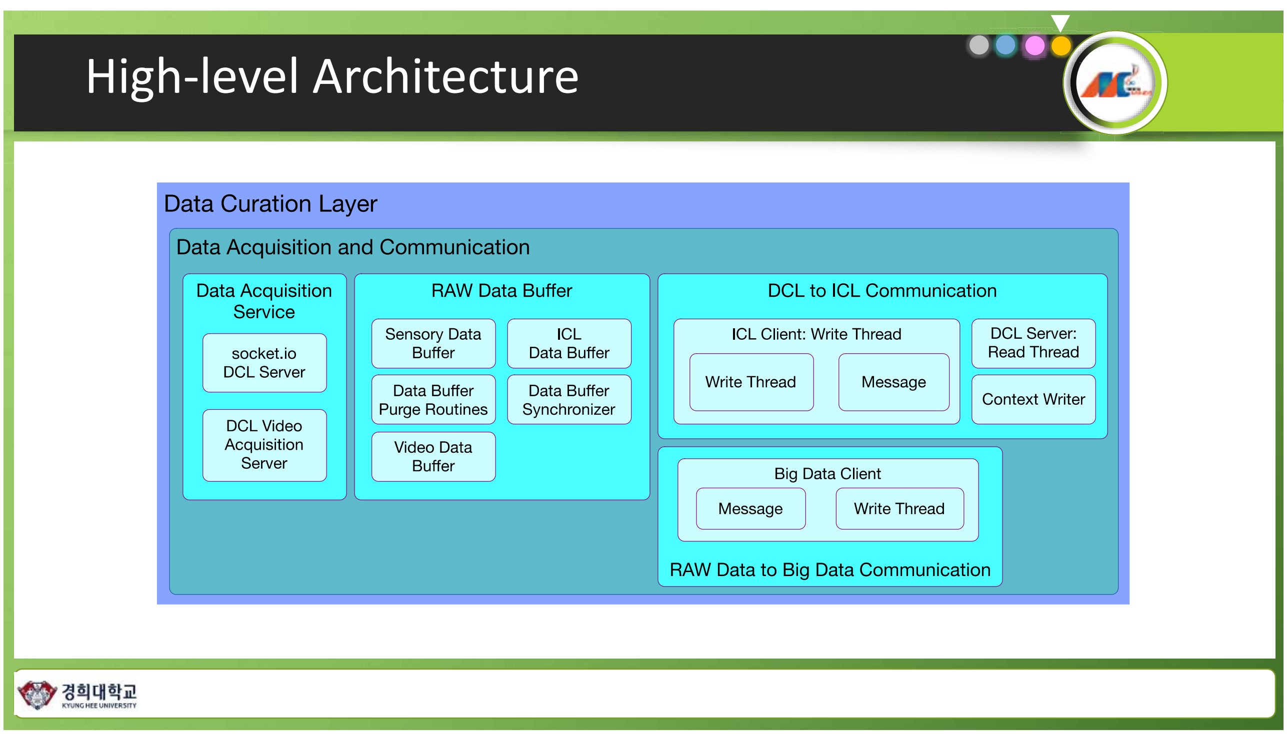
Motivations

- Sensory data is generated after every 3 seconds from data sources
  - *Real-time data acquisition*
- MMV 2.0 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 2.0 is a distributed layer with parallel execution
  - *Optimal resource utilization with no communication bottlenecks*

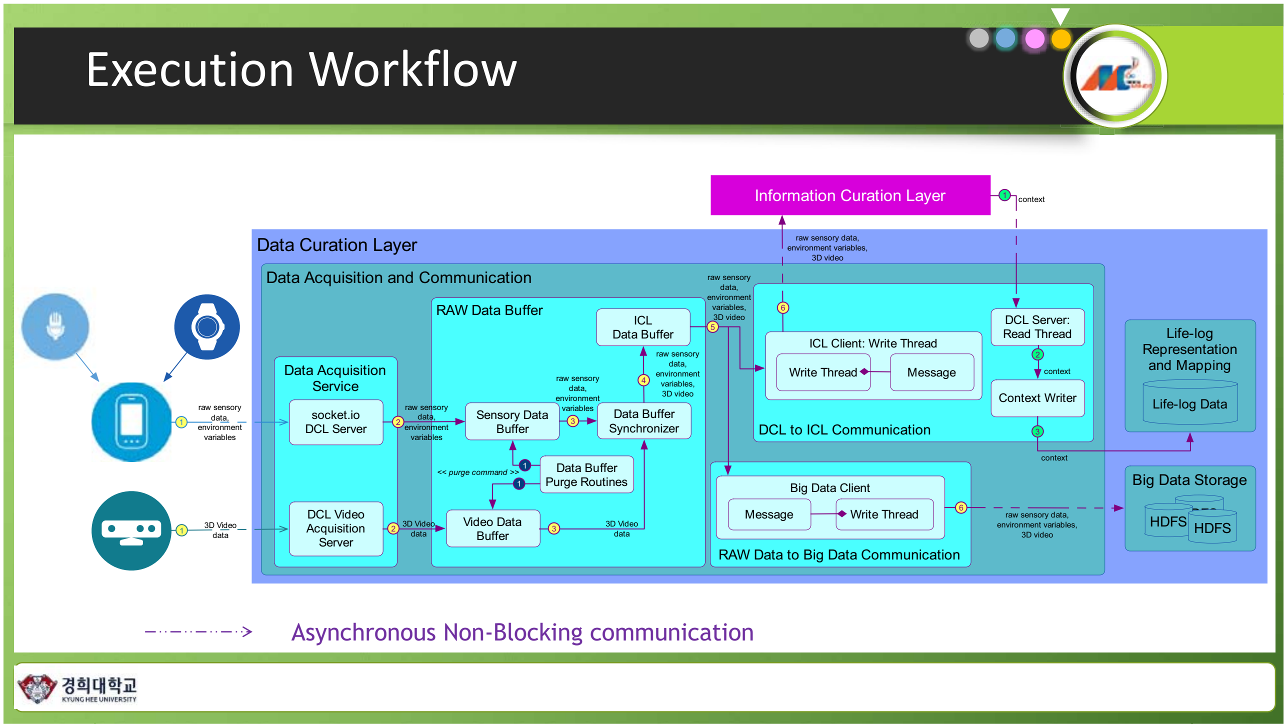
경희대학교
KYUNG HEE UNIVERSITY



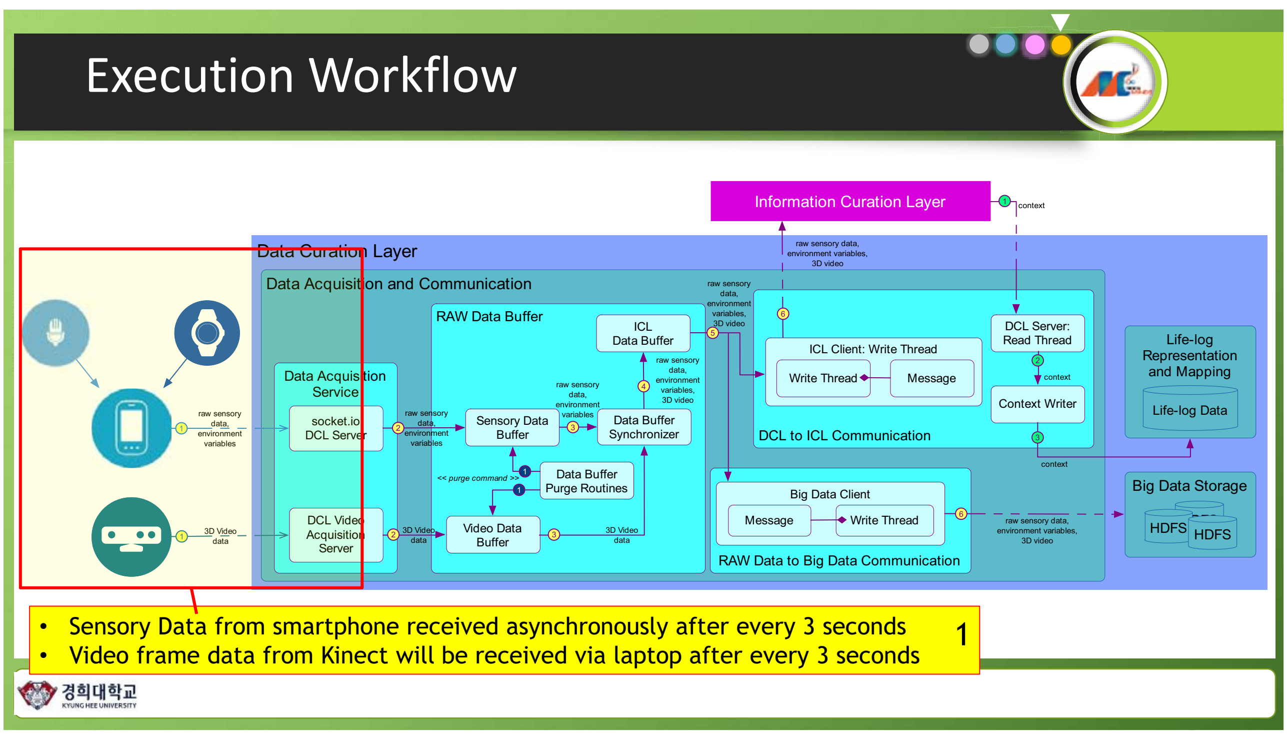
DCL V2.0 Architecture
Data Curation Layer
Sensory Data Processing and Curation
Data Acquisition
Data Acquisition Service
Raw Data buffer
Sensory Data Synchronizer
Context Writer
Intermediate Database
Life-log Data & User Profiles
Configuration Data
Life-log Monitoring
Monitor Event Configurator
Constraints Configurator
Situation Event Detector
Life-log Representation and Mapping
Schema and Instance Mapper
Storage Verifier
Representation Model Selector
Big Data Storage
Data Persistence
Message Model
Data Writer
Query Writer
Query Authoring
Query Deployer
Passive Data Reader
Query Loader
Data Exporter
Scan
Schema
Active Data Reader
Query Loader
Data Exporter
Data Format
Life-log Sync
Query Library
MapReduce
Hive
Cluster Management and Provisioning
Ambari
Physical Data Storage
HBase
HDFS
HDFS
경희대학교
KYUNG HEE UNIVERSITY



High-level Architecture
Data Curation Layer
Data Acquisition and Communication
Data Acquisition Service
socket.io DCL Server
DCL Video Acquisition Server
RAW Data Buffer
Sensory Data Buffer
ICL Data Buffer
Data Buffer Purge Routines
Data Buffer Synchronizer
Video Data Buffer
DCL to ICL Communication
ICL Client: Write Thread
Write Thread
Message
DCL Server: Read Thread
Context Writer
Big Data Client
Message
Write Thread
RAW Data to Big Data Communication
경희대학교
KYUNG HEE UNIVERSITY

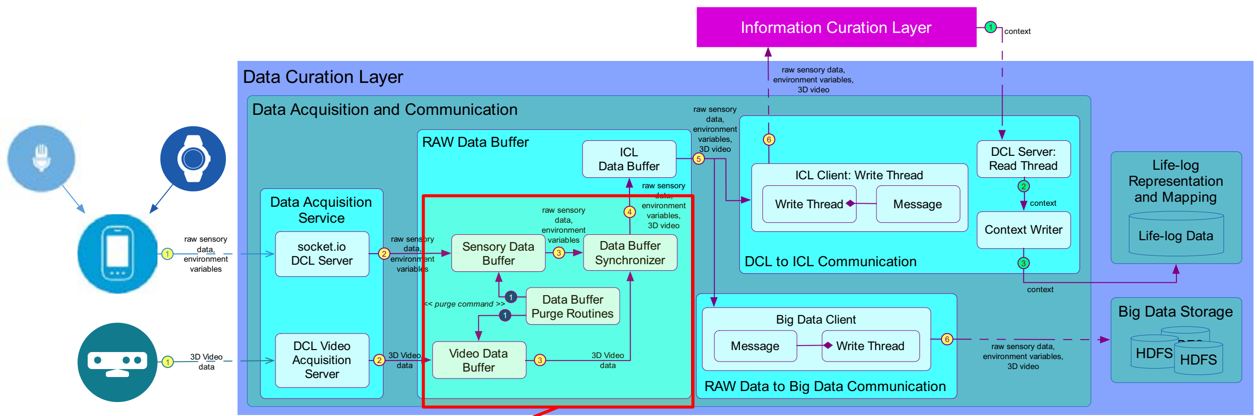


Execution Workflow
Information Curation Layer
context
Data Curation Layer
raw sensory data, environment variables, 3D video
Data Acquisition and Communication
RAW Data Buffer
ICL Data Buffer
Data Acquisition Service
socket.io DCL Server
Sensory Data Buffer
Data Buffer Synchronizer
Data Buffer Purge Routines
<< purge command >>
DCL Video Acquisition Server
Video Data Buffer
3D Video data
raw sensory data, environment variables
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
HDFS
HDFS
Asynchronous Non-Blocking communication
경희대학교
KYUNG HEE UNIVERSITY



Execution Workflow
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
raw sensory data, environment variables
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
HDFS
HDFS
• 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
KYUNG HEE UNIVERSITY

Execution Workflow

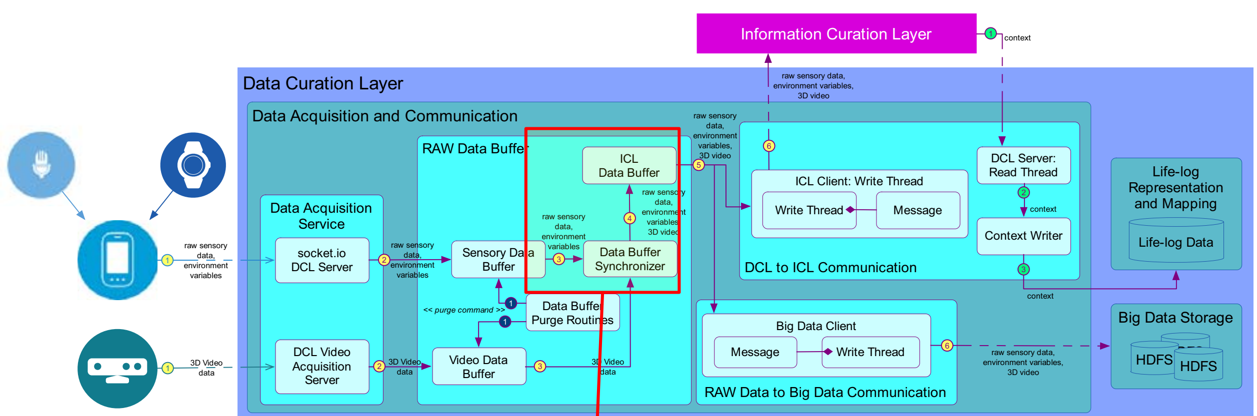


- Sensory data buffer for smartphone-based sensory data
- Video data buffer for for Kinect-based video frames data
- Buffers are synchronized by time-stamp and user id

2

경희대학교 KYUNG HEE UNIVERSITY

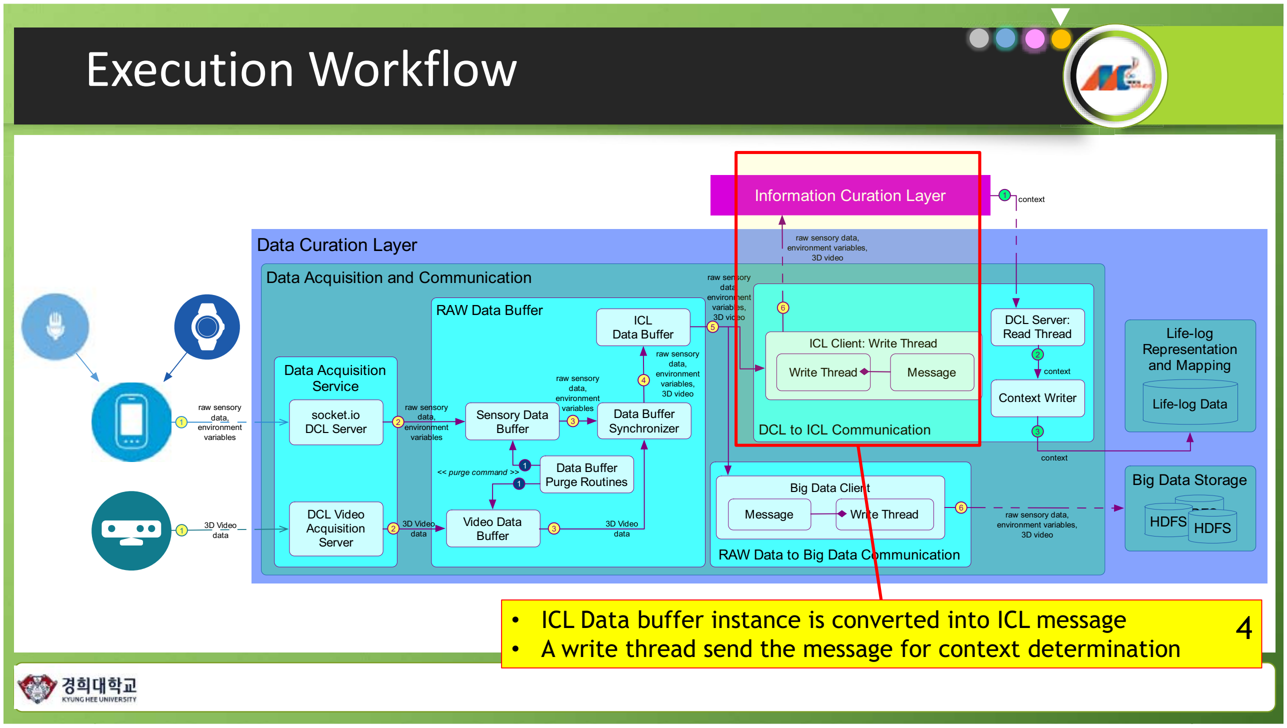
Execution Workflow

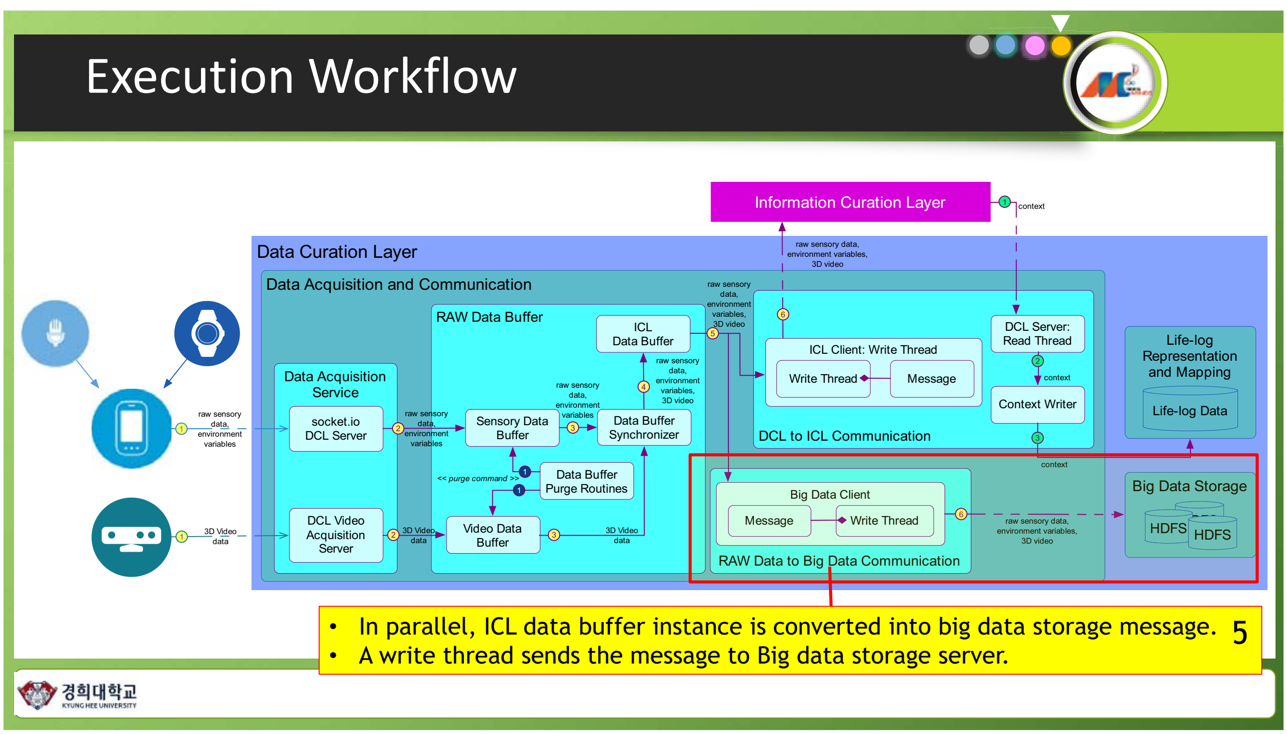


- Data synchronized by time-stamp is moved to ICL buffer
- Data items are purged from Sensory and Video Data buffers

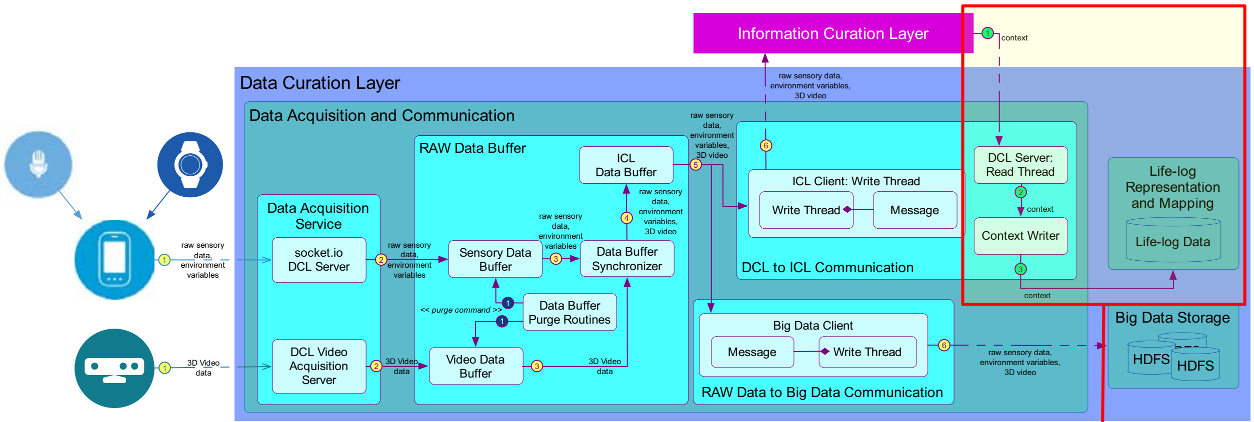
3

경희대학교 KYUNG HEE UNIVERSITY


Execution Workflow
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
<< purge command >>
Data Buffer Purge Routines
DCL Video Acquisition Server
Video Data Buffer
3D Video data
ICL Client: Write Thread
Write Thread
Message
DCL to ICL Communication
DCL Server: Read Thread
Context Writer
Big Data Client
RAW Data to Big Data Communication
Life-log Representation and Mapping
Life-log Data
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
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
<< purge command >>
Data Buffer Purge Routines
DCL Video Acquisition Server
Video Data Buffer
3D Video data
ICL Client: Write Thread
Write Thread
Message
DCL to ICL Communication
DCL Server: Read Thread
Context Writer
Big Data Client
RAW Data to Big Data Communication
Life-log Representation and Mapping
Life-log Data
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



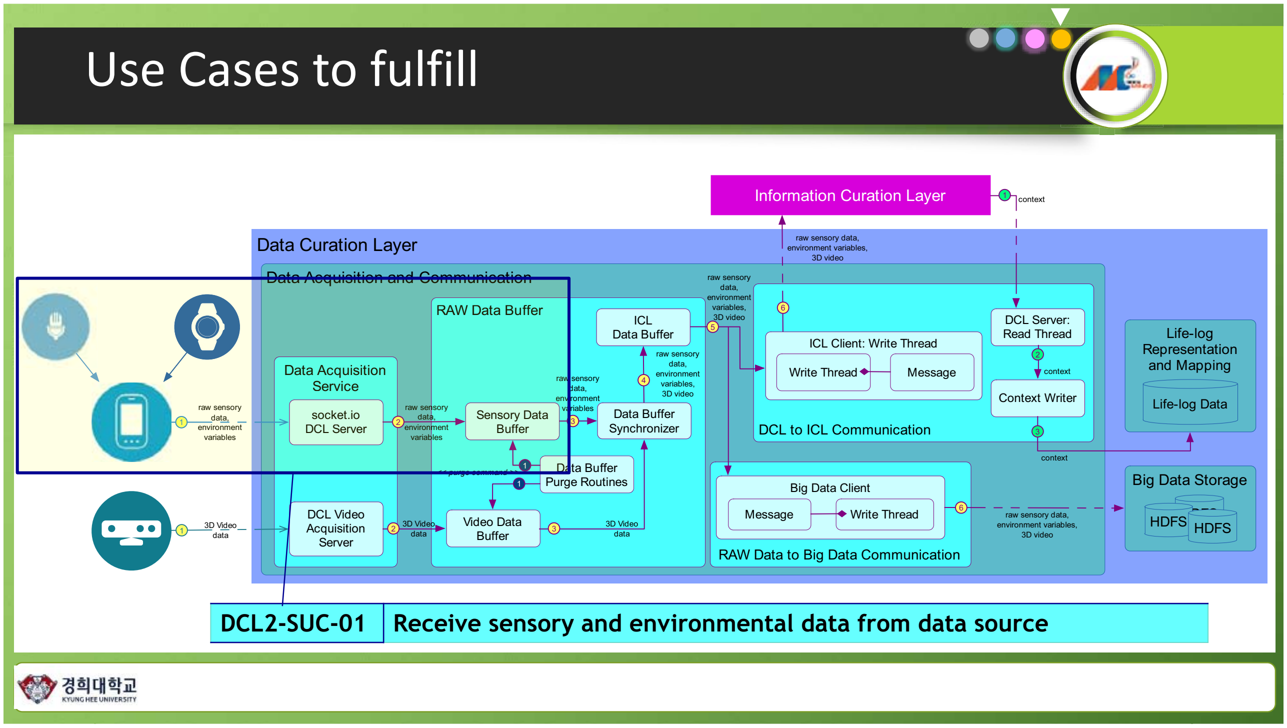
- With change in context, Read thread will receive the new context from ICL 6
- Context is mapped to life-log object model for life-log persistence

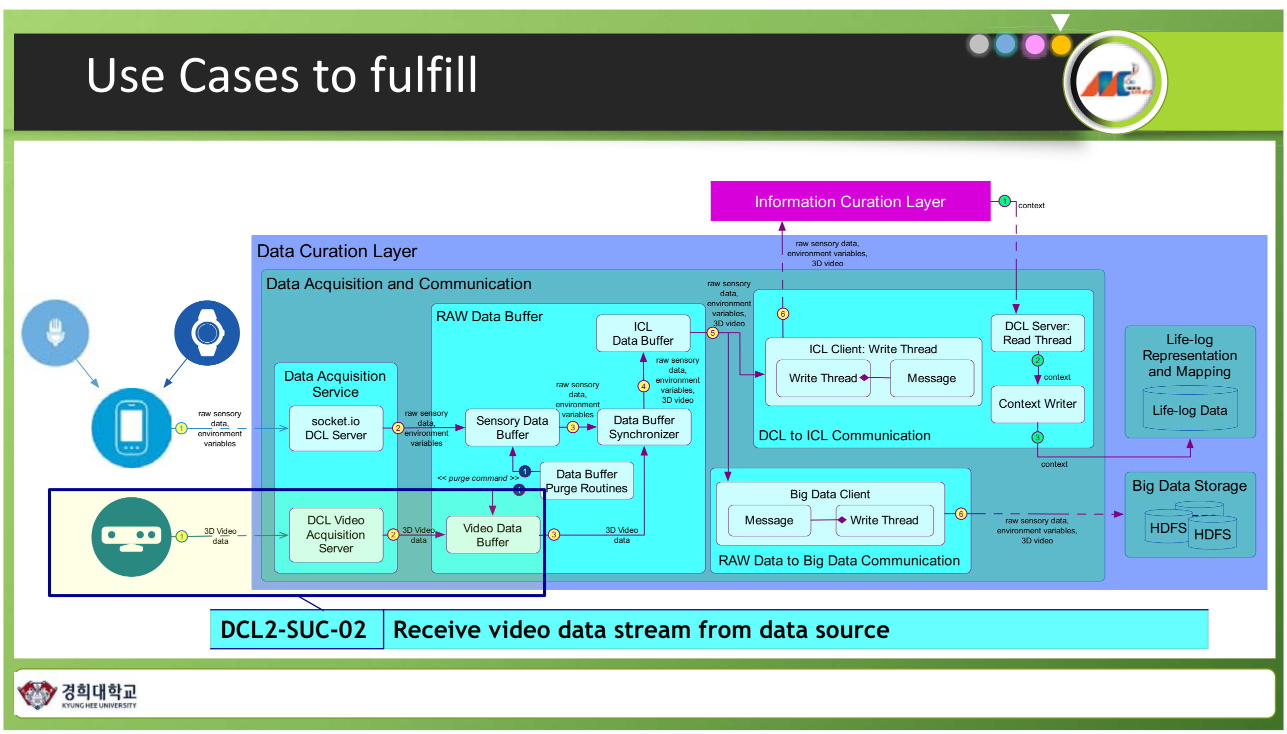
경희대학교 KYUNG HEE UNIVERSITY

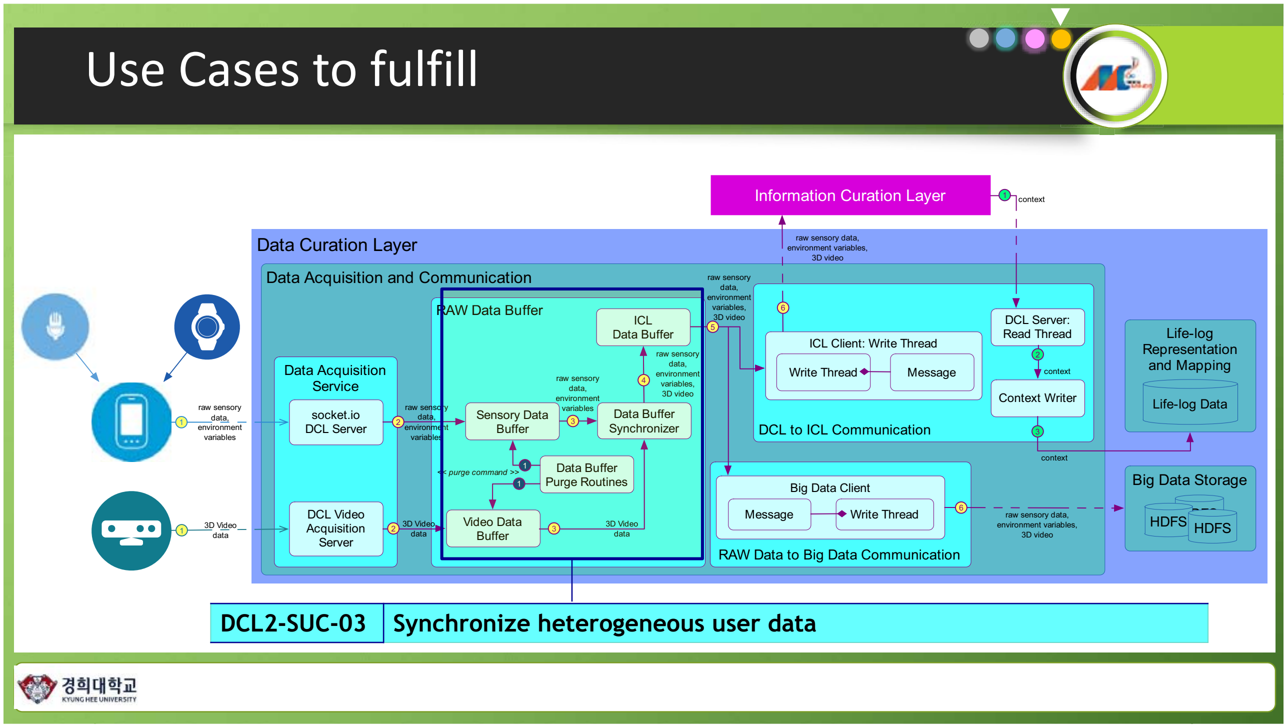
DCL V2.0 Use cases

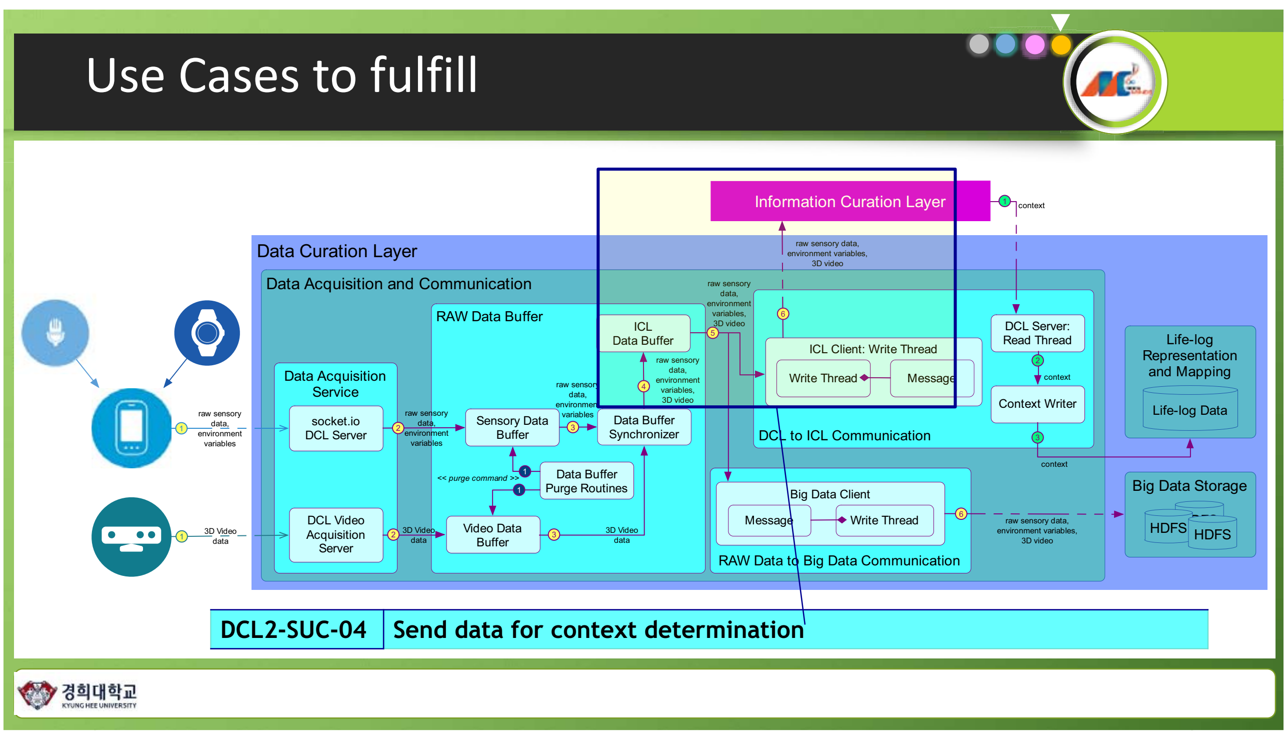
Use case ID	Use Case Description
DCL2-SUC-01	Receive sensory and environmental data from data source
DCL2-SUC-02	Receive video data stream from data source
DCL2-SUC-03	Synchronize heterogeneous user data
DCL2-SUC-04	Send data for context determination
DCL2-SUC-05	Receive context data
DCL2-SUC-15	Persist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

경희대학교 KYUNG HEE UNIVERSITY

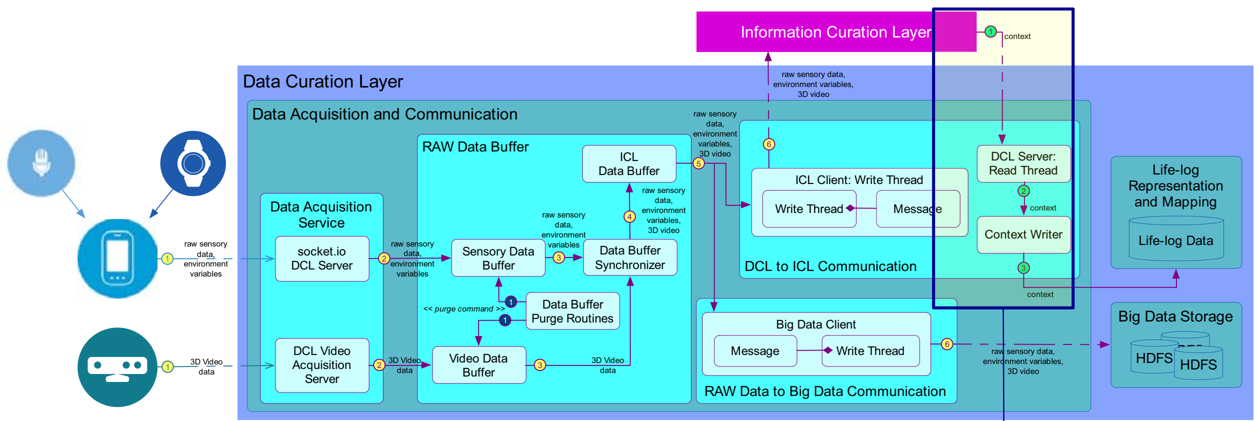

Use Cases to fulfill
Information Curation Layer
context
raw sensory data, environment variables, 3D video
Data Curation Layer
Data Acquisition and Communication
RAW Data Buffer
ICL Data Buffer
Data Acquisition Service
socket.io DCL Server
Sensory Data Buffer
Data Buffer Synchronizer
Data Buffer Purge 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
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-01
Receive sensory and environmental data from data source
경희대학교
KYUNG HEE UNIVERSITY


Use Cases to fulfill
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
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-02
Receive video data stream from data source
경희대학교
KYUNG HEE UNIVERSITY


Use Cases to fulfill
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
<< purge command >>
Data Buffer Purge Routines
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
RAW Data to Big Data Communication
Life-log Representation and Mapping
Life-log Data
Big Data Storage
HDFS
DCL2-SUC-03
Synchronize heterogeneous user data
경희대학교
KYUNG HEE UNIVERSITY

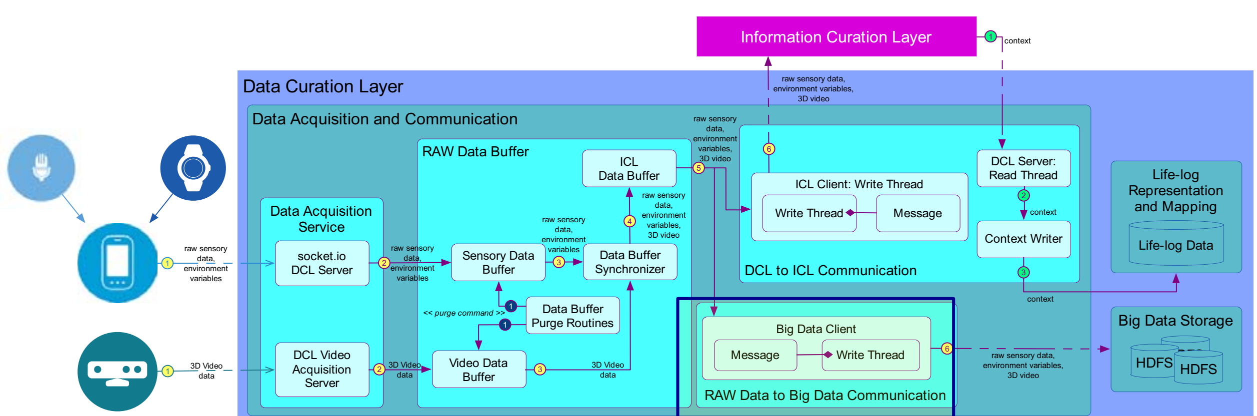

Use Cases to fulfill
Information Curation Layer
context
Data Curation Layer
raw sensory data, environment variables, 3D video
Data Acquisition and Communication
RAW Data Buffer
ICL Data Buffer
Data Acquisition Service
socket.io DCL Server
Sensory Data Buffer
Data Buffer Synchronizer
<< purge command >>
Data Buffer Purge Routines
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
RAW Data to Big Data Communication
Life-log Representation and Mapping
Life-log Data
Big Data Storage
HDFS
DCL2-SUC-04
Send data for context determination
경희대학교
KYUNG HEE UNIVERSITY

Use Cases to fulfill



DCL2-SUC-05	Receive context data

Use Cases to fulfill



DCL2-SUC-15	Persist sensory data in non-volatile storage

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

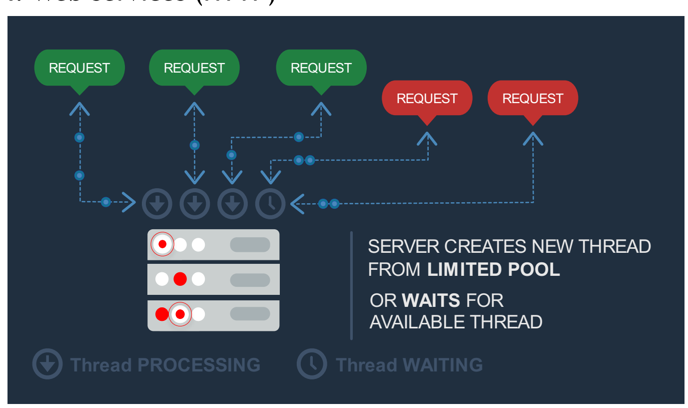
✕ 3. Scalable

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

Requirements

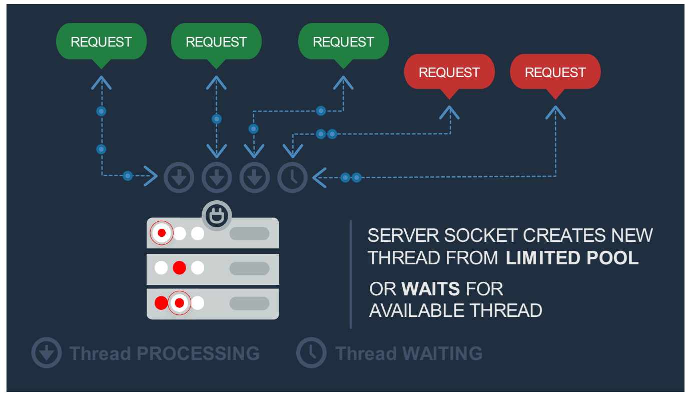
- ✗ 1. Non-Blocking Async Recv.
- ✓ 2. Real-time Communication
- ✗ 3. Scalable

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

Requirements

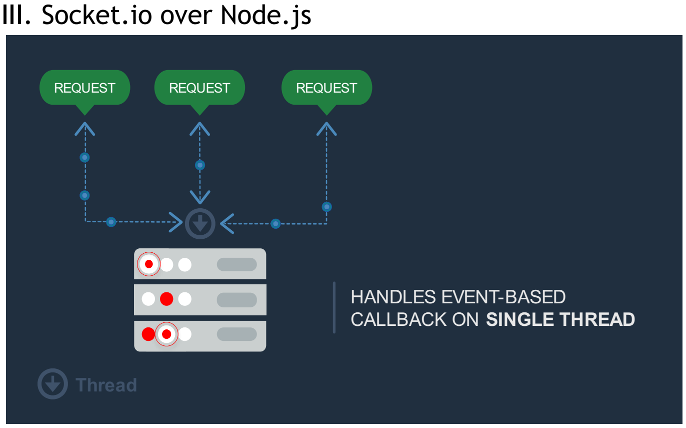
- ✓ 1. Non-Blocking Async Recv.
- ✓ 2. Real-time Communication
- ✓ 3. Scalable

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. Socket.io over Node.js

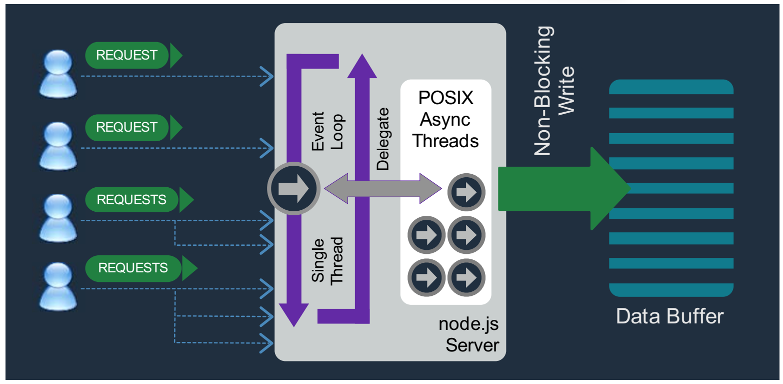


경희대학교 KYUNG HEE UNIVERSITY

1. Application to DCL Real-time Communication

Socket.io over 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!

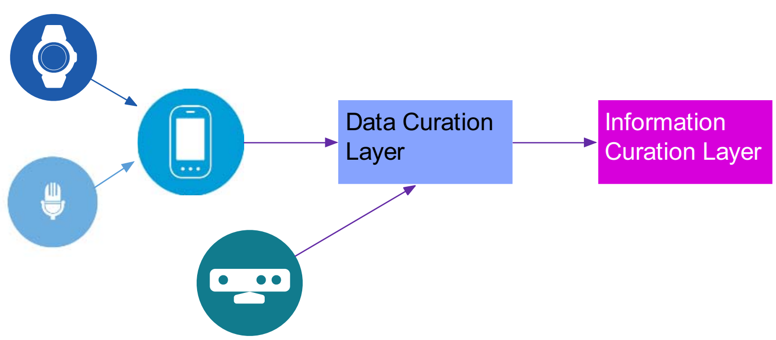
경희대학교

2. Regulation of Data Influx from DCL to ICL

DCL 1.0

DCL 1.5

DCL 2.0



경희대학교

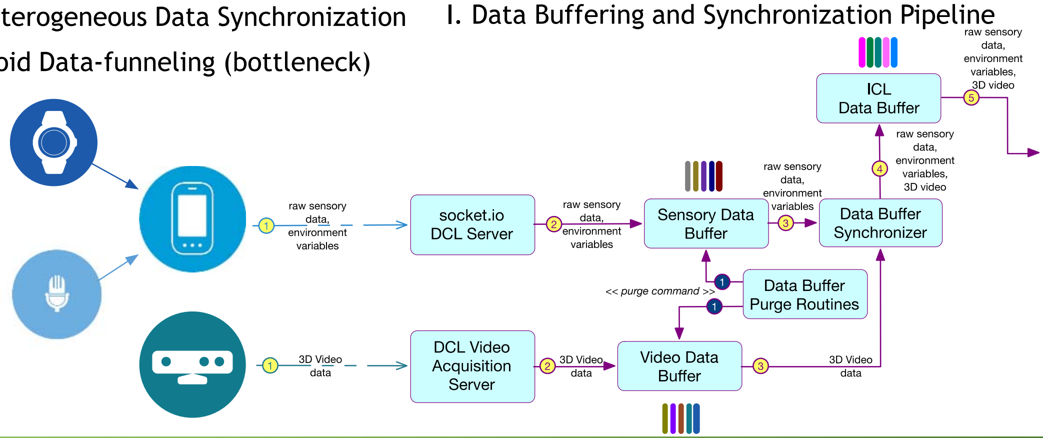
2. Regulation of Data Influx from DCL to ICL

Requirements

1. Heterogeneous Data Synchronization
2. Avoid Data-funneling (bottleneck)

Candidate Solution

I. Data Buffering and Synchronization Pipeline



경희대학교 KYUNG HEE UNIVERSITY

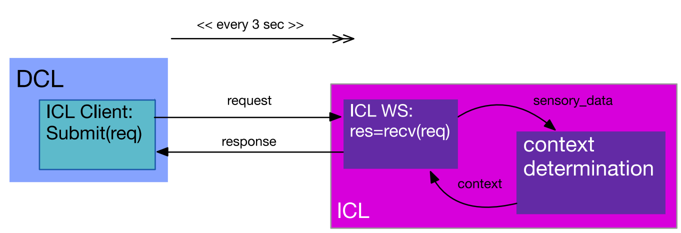
3. DCL to ICL Communication

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

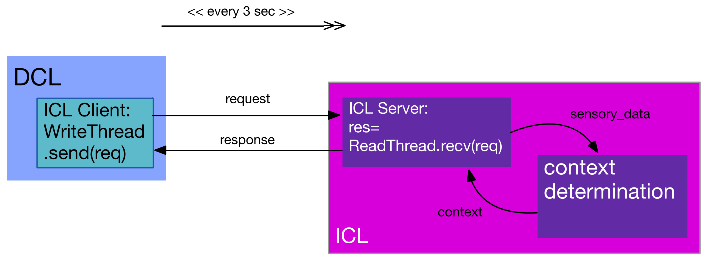
Requirements

✗ 1. Non-Blocking Communication

✗ 2. Reduced Communication Overhead

Candidate Solutions

II. Java sockets (Request-Response)



Limitations

1. Blocking IO
2. Large Communication Overhead

3. DCL to ICL Communication

Requirements

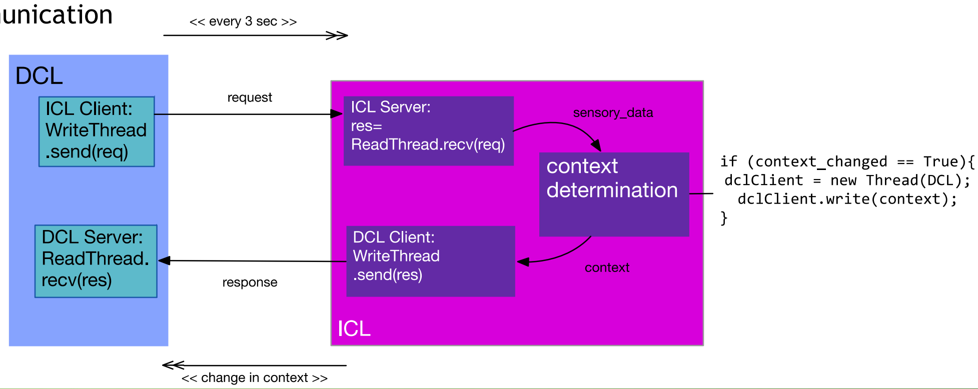
✓ 1. Non-Blocking Communication

✓ 2. Reduced Communication Overhead

Candidate Solutions

III. Java sockets (Non-Blocking)

Java



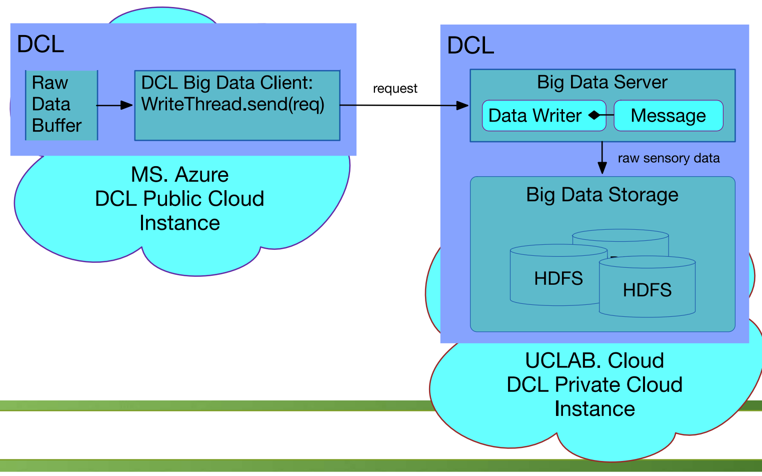
4. Actionable Item : Non-volatile Data Persistence

Requirements

✓ 1. Non-Blocking Communication

Candidate Solutions

I. Java sockets (Non-Blocking)



Contributions

- *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*

Present & Future Work

- Current Progress
  - Use-case model and sequence diagrams completed
  - Review of candidate technologies completed
- Near Future Plans
  - Conceptual model and class diagrams
  - App to DCL Communication Implementation (Next 2 Weeks)
  - Data Buffering and Synchronization Implementation

경희대학교
KYUNG HEE UNIVERSITY

Data Curation Layer (DCL 2.0)

Lifelog Representation and Mapping

Taqdir Ali

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

Introduction

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

Related Work

Graphical Representation [1], [2]

Features

- Diagrammatic Representation at design time
- Focus on issues of formality and generality
- Cover diversity, quality and complex relationships

Limitations

- Needs to translate into usable format
- Limitation of privacy and distribution of data

Logic Based Representation [3]

Features

- Adding data as facts and using expressions or rules
- Sensed data represent in First Order Predicate Logic
- Also represent the meta-propositional properties

Limitations

- Model does not support design and documentation of design process
- The sensed data cannot be used by software tool

Ontological Representation [4], [5], [6], [7], [16], [17]

Features

- 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

Limitations

- 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

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

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

Object Oriented Representation

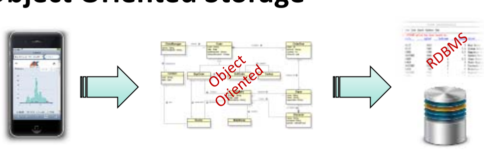
- Pros
  - Data stores as object oriented principals
  - Allows "Real World" to be modeled more closely
  - High extensibility and scalability
  - Support for schema evolution
  - High Performance (Time)
- Cons
  - Lack of universal data model
  - Lack of standards
  - Highly complex to manage
  - Less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

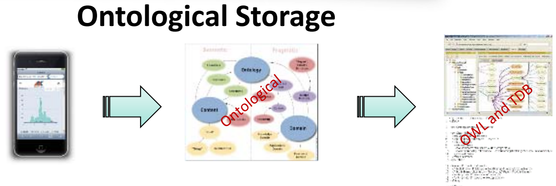
Comparison b/w Object Oriented and Ontological Storage

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

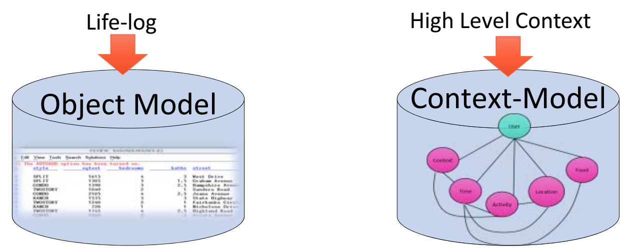


- Consists of axioms (Structure of model) and facts (Concrete Situation)
- Open world assumption (OWA)
  - Missing information treated as unknown
- No-Unique name assumption (No-UNA)
  - Individuals may have more than one name
- Ontology axioms behave like implications (inference rules)

Discussion: Trade off between Object model and ontological model

- 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.

Motivation

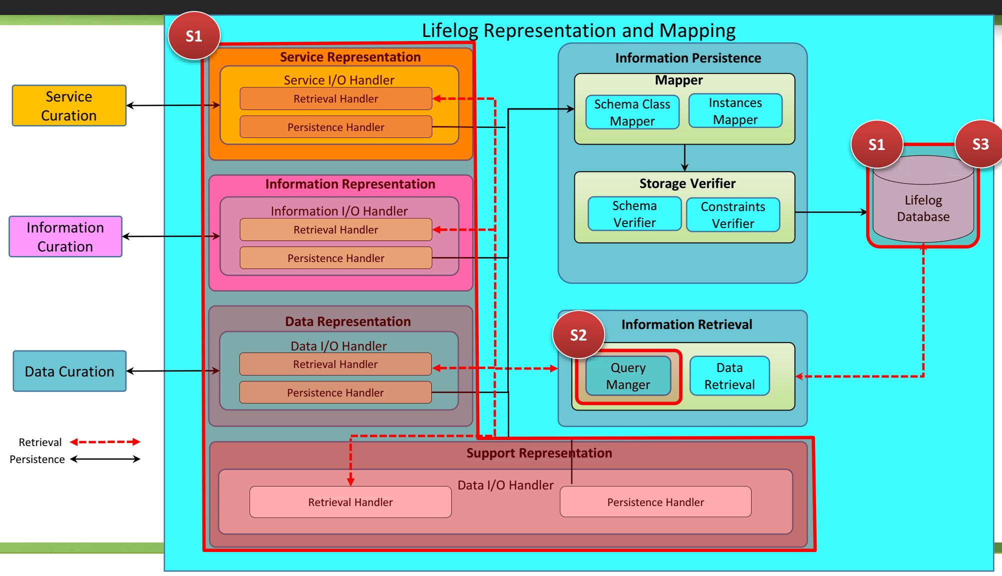
- **Heterogeneous data Processing Streams**: Providing extensible and understandable structured format to variety of data.
- **Fine-grained Information access**: Processable and procedural format to all layers and modules of MM Platform
- **Extensible Data Model**: Integratable new models based on the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

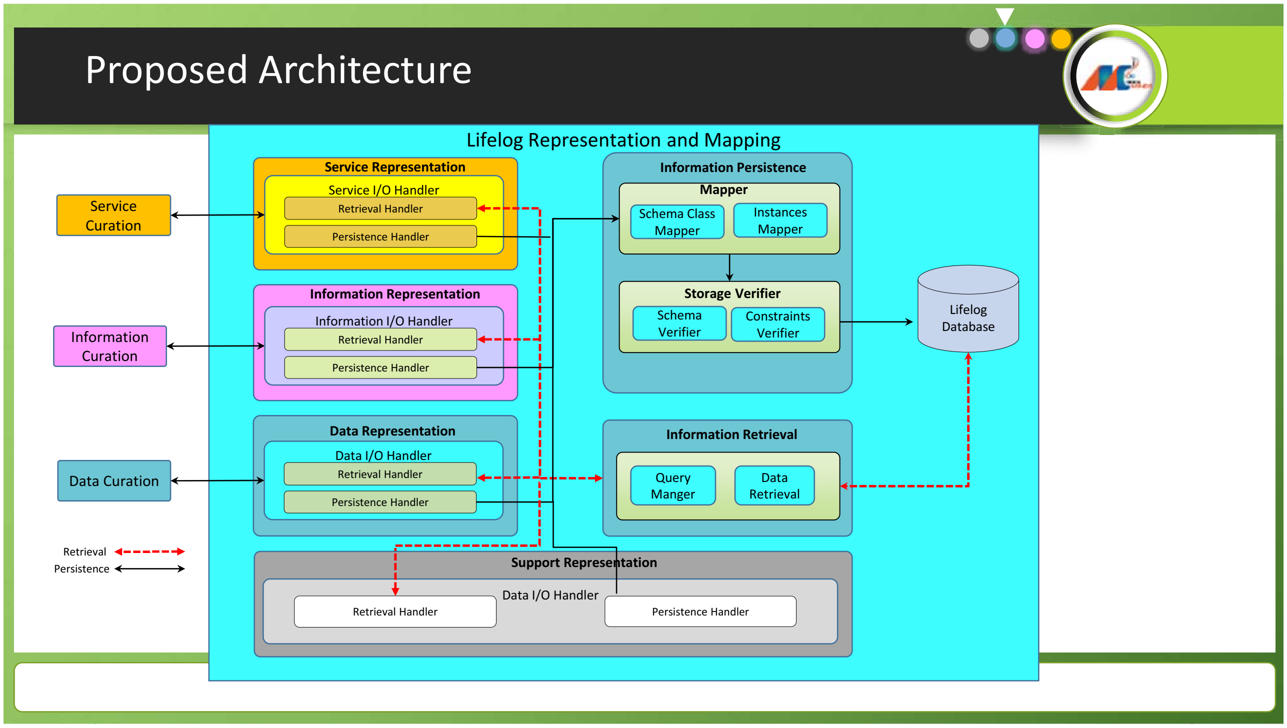
Challenges and Solutions

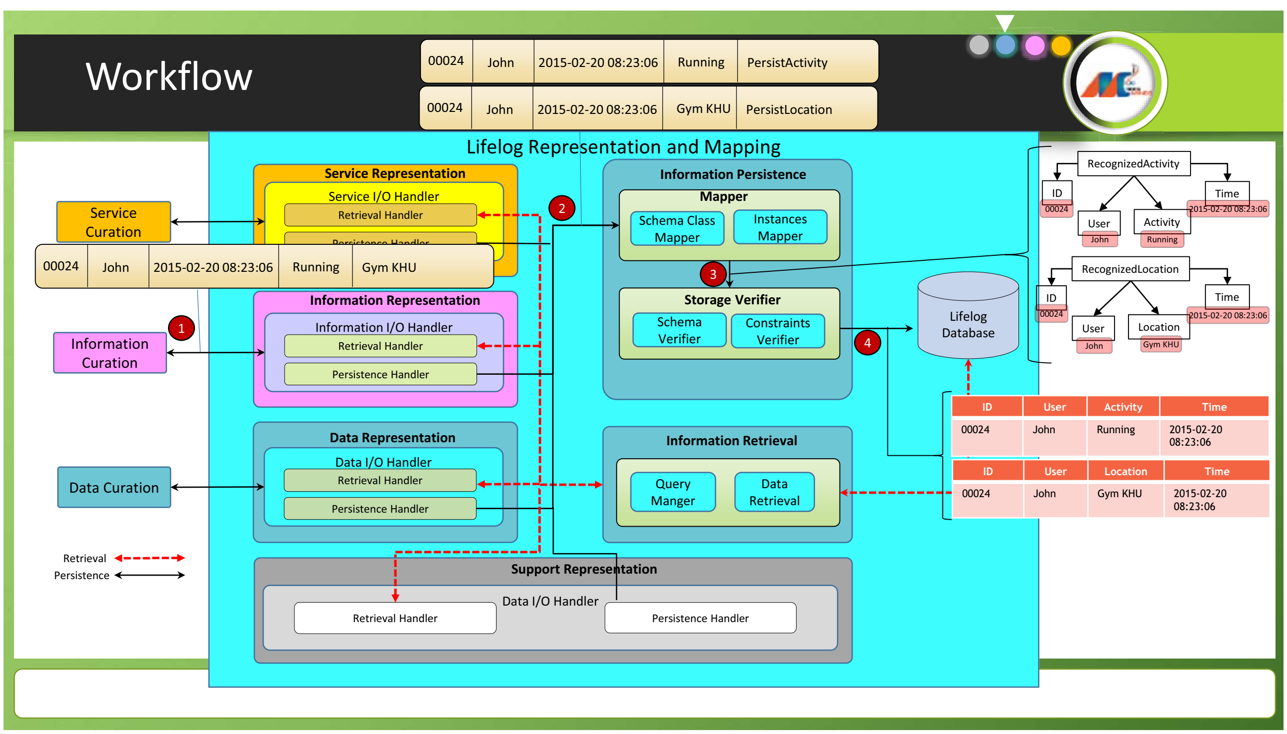
Motivations	Challenges	Solutions	
Heterogeneous data Processing Streams	• Continuous data identification and storage	Representation Models, Lifelog Database	S1
Fine-grained Information access	• Dynamic query generations for different layers	Query Manager	S2
Extensible Data Model	• Design and development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

경희대학교
KYUNG HEE UNIVERSITY

Proposed Architecture and solutions

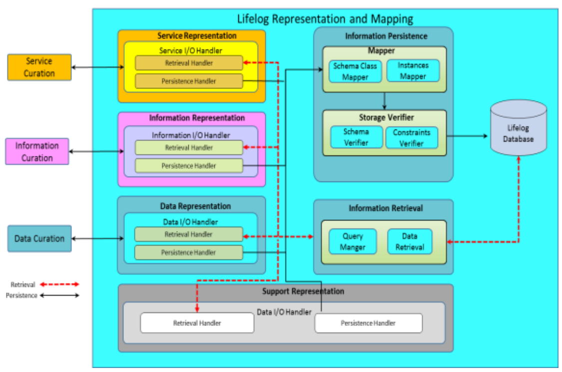



Proposed Architecture
Lifelog Representation and Mapping
Service Curation
Information Curation
Data Curation
Service Representation
Service I/O Handler
Retrieval Handler
Persistence Handler
Information Representation
Information I/O Handler
Retrieval Handler
Persistence Handler
Data Representation
Data 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
Retrieval
Persistence

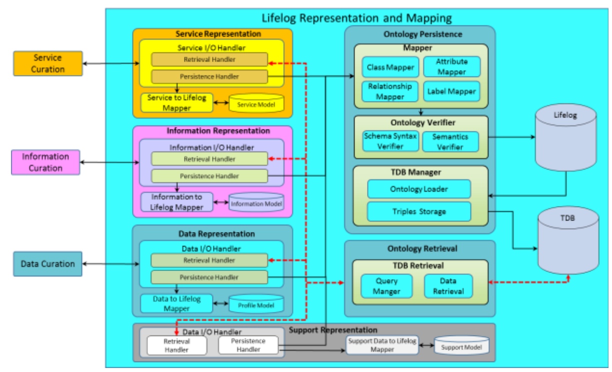

Workflow
00024 John 2015-02-20 08:23:06 Running PersistActivity
00024 John 2015-02-20 08:23:06 Gym KHU PersistLocation
Lifelog Representation and Mapping
Service Curation
00024 John 2015-02-20 08:23:06 Running Gym KHU
Information Curation
Data Curation
Service Representation
Service I/O Handler
Retrieval 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
User
John
Activity
Running
Time
2015-02-20 08:23:06
RecognizedLocation
ID
00024
User
John
Location
Gym KHU
Time
2015-02-20 08:23:06
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
Retrieval
Persistence

Objected Oriented vs Ontological Representation

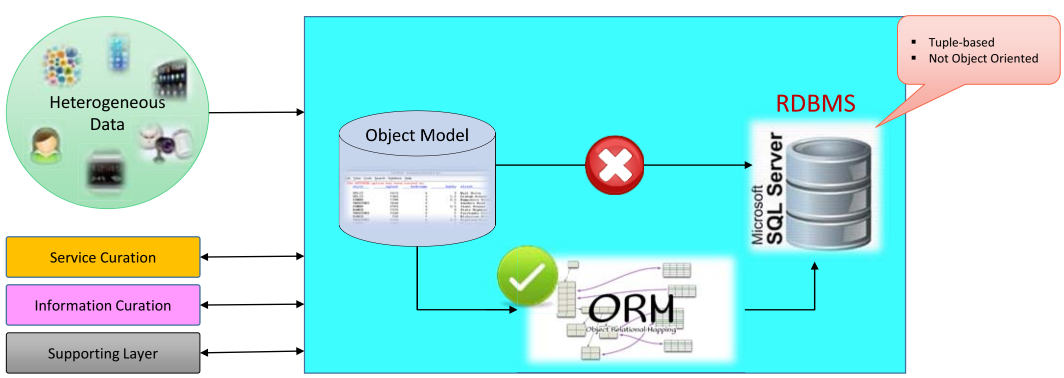
Object Oriented Representation



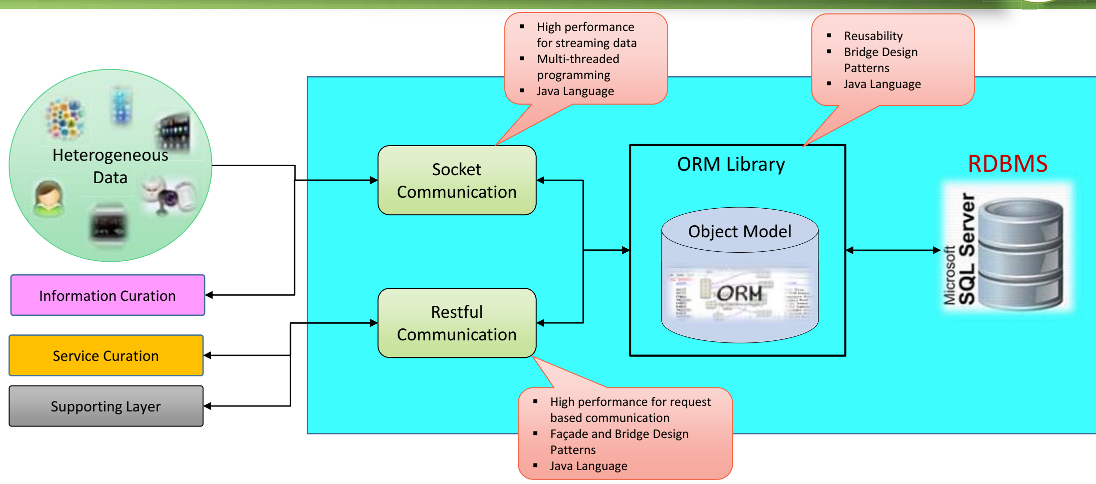
Ontological Representation



Implementation Details



Use of ORM (Object-Relational Mapping)



Object Oriented Design Patterns

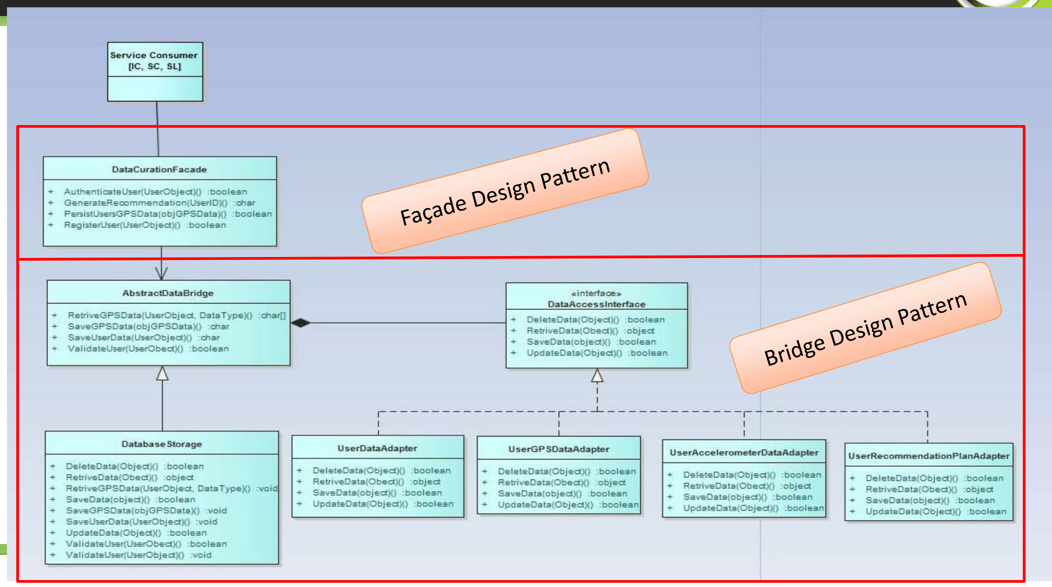
Façade Design Pattern

- 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

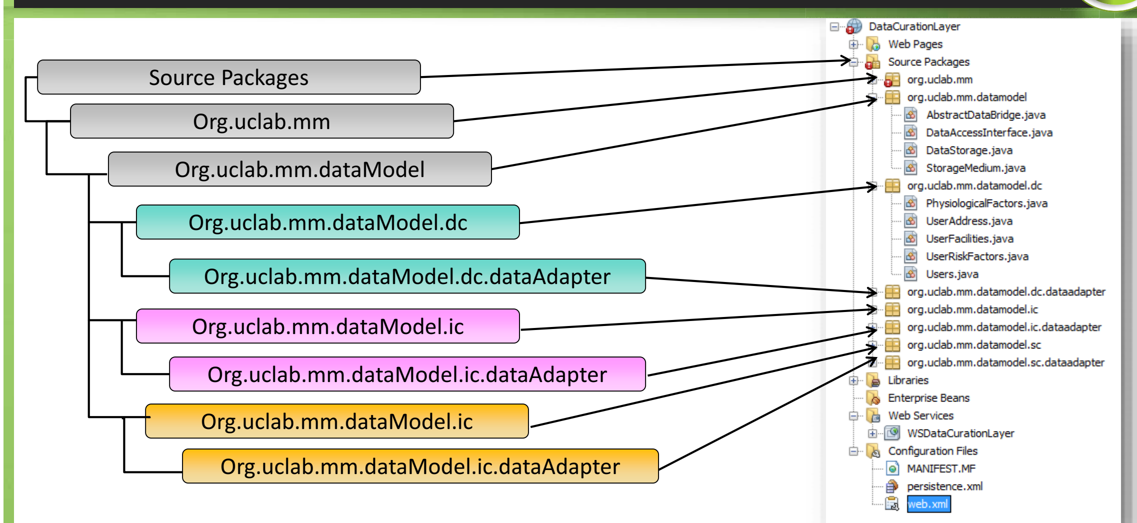
Bridge Design Pattern

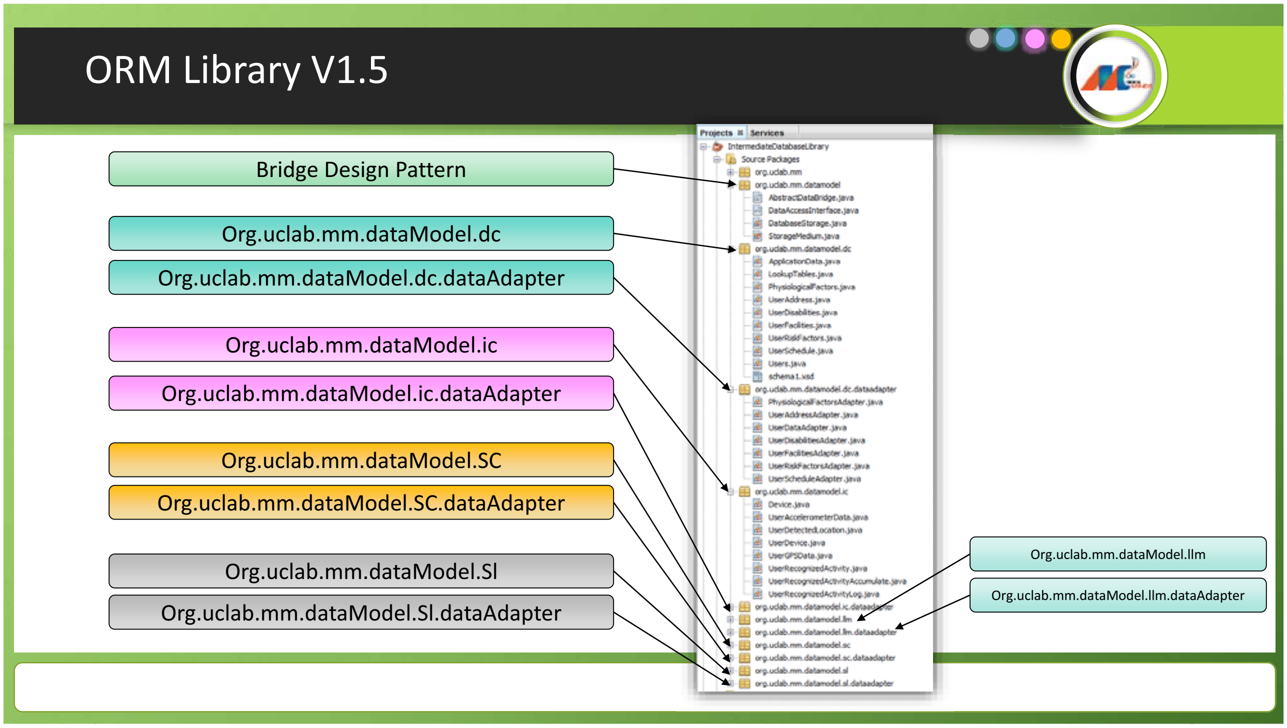
- Scalability for data storage
- Decouple the abstraction from its implementations
- Divide the responsibilities among different classes
- Hide details from client

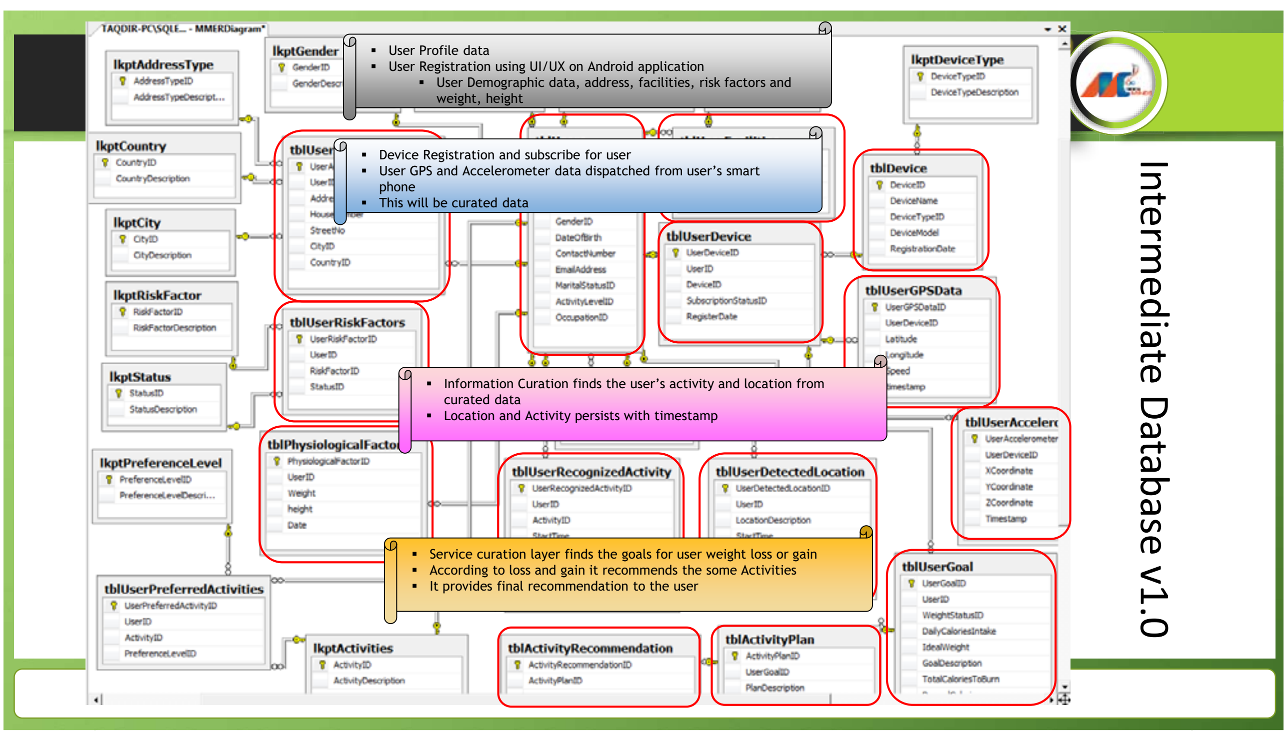
Object Oriented Design Patterns



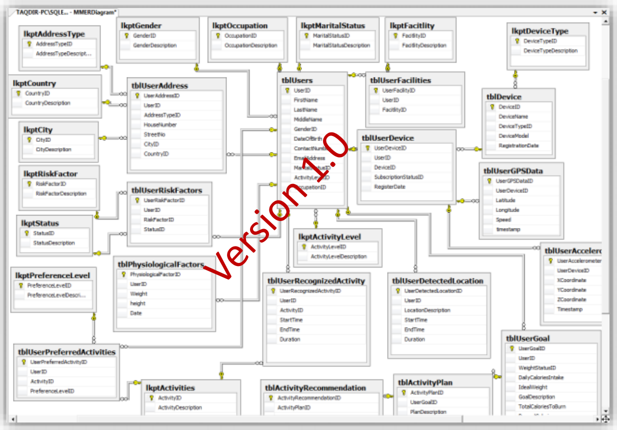
ORM Library V1.0

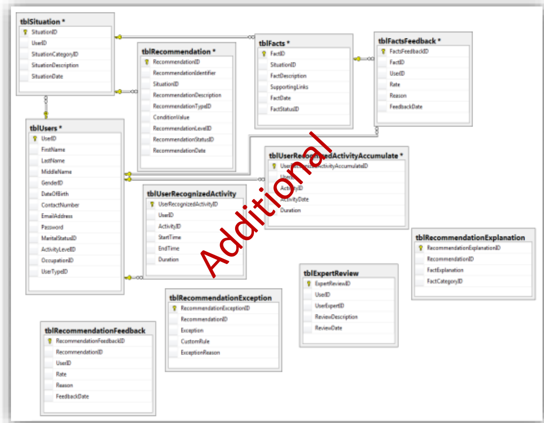



ORM Library V1.5
Bridge Design Pattern
Org.uclab.mm.dataModel.dc
Org.uclab.mm.dataModel.dc.dataAdapter
Org.uclab.mm.dataModel.ic
Org.uclab.mm.dataModel.ic.dataAdapter
Org.uclab.mm.dataModel.SC
Org.uclab.mm.dataModel.SC.dataAdapter
Org.uclab.mm.dataModel.Sl
Org.uclab.mm.dataModel.Sl.dataAdapter
Org.uclab.mm.dataModel.llm
Org.uclab.mm.dataModel.llm.dataAdapter


Intermediate Database v1.0
User Profile data
User Registration using UI/UX on Android application
User Demographic data, address, facilities, risk factors and weight, height
Device Registration and subscribe for user
User GPS and Accelerometer data dispatched from user's smart phone
This will be curated data
Information Curation finds the user's activity and location from curated data
Location and Activity persists with timestamp
Service curation layer finds the goals for user weight loss or gain
According to loss and gain it recommends the some Activities
It provides final recommendation to the user
lkptAddressType
lkptGender
lkptDeviceType
lkptCountry
lkptCity
lkptRiskFactor
lkptStatus
lkptPreferenceLevel
tblUserPreferredActivities
lkptActivities
tblDevice
tblUserDevice
tblUserGPSData
tblUserRiskFactors
tblPhysiologicalFacto
tblUserRecognizedActivity
tblUserDetectedLocation
tblUserAccelero
tblUserGoal
tblActivityRecommendation
tblActivityPlan

Intermediate Database v1.5





Mining Minds Version 2.0

- We will use Object Oriented Representation
- Same communication (Socket and Restful)
- Object-Relational Mapping (ORM)
- Relational Database (MS SQL Server)
- Java Language
- Maintenance of Existing work
  - Changes in DB
  - Changes in ORM
  - Changes in Restful services
  - Changes in SQL Stored Procedures

These changes will be implemented according requirements

Data Curation Layer (DCL 2.0)

Bilal Ali

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

Introduction

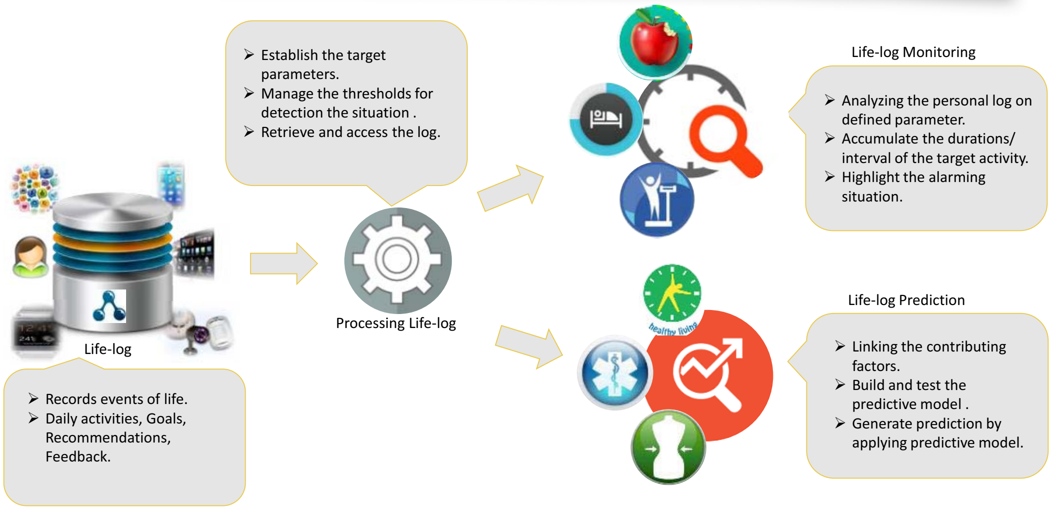
- **Self Quantification**
  - Measure *Life numerically* and perform mathematical manipulation of the results
  - Provide common *standards*, when comparing different institutions and countries
  - Provides *objectivity* to goals, target and achievements
  - Provides a intellect of *certainty*
  - lends *credibility* to the research

The Quantified Self
self knowledge through numbers

**http://www.limitsofebm.org/the-flaw-in-ebm/why-quantify-everything/

경희대학교 KYUNG HEE UNIVERSITY

Introduction



Related Work

Study	Domain	Service	Methodology	Techniques
(Zini 2015) Life-logs Aggregation for Quality of Life Monitoring	Health Care	Visualization of patient log	An architecture for the acquisition of life-logs, their fusion, and their storage. A prototype GUI for the visualization of 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	Visualization 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)

Limitations of Exiting Work

- Life-log is only used for *identification* of activities.
- *Limited visualization* of recorded activities on basis of hours and days.
- *Small* set of activities are accommodated.
- *Manual* entries of activities for logging.

Motivations

2
Dynamically accommodate the critical situation on the basis of personalized activities.

1
Monitor the Life-log in terms of **attractiv** **attributes** to highlig the **atypical** conditions.

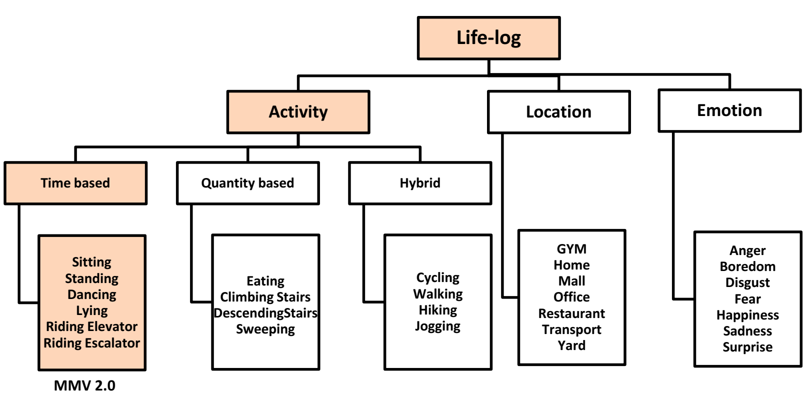
3
Evaluate the **alarming** condition in **real time** and generate alarm for producing recommendation.

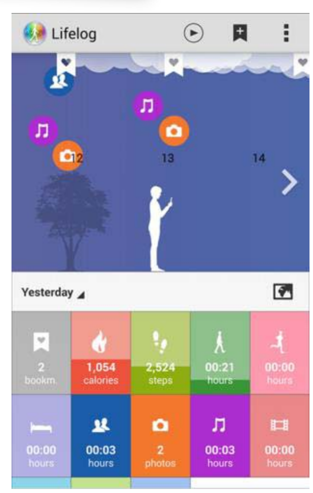
Challenges, Solutions, and Methods

Motivation	Challenges	Solution	Methods
Life-log Monitoring	Selection of Targeted contributing attributes	Monitoring Configuration	Querying target variables based data from Life-log data , preprocess to transform for modeling, assign correlated weight and select essential attributes.
Dynamic Situation	Implementation of experts opinion for the identification of rules related to situation.	Situation 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.

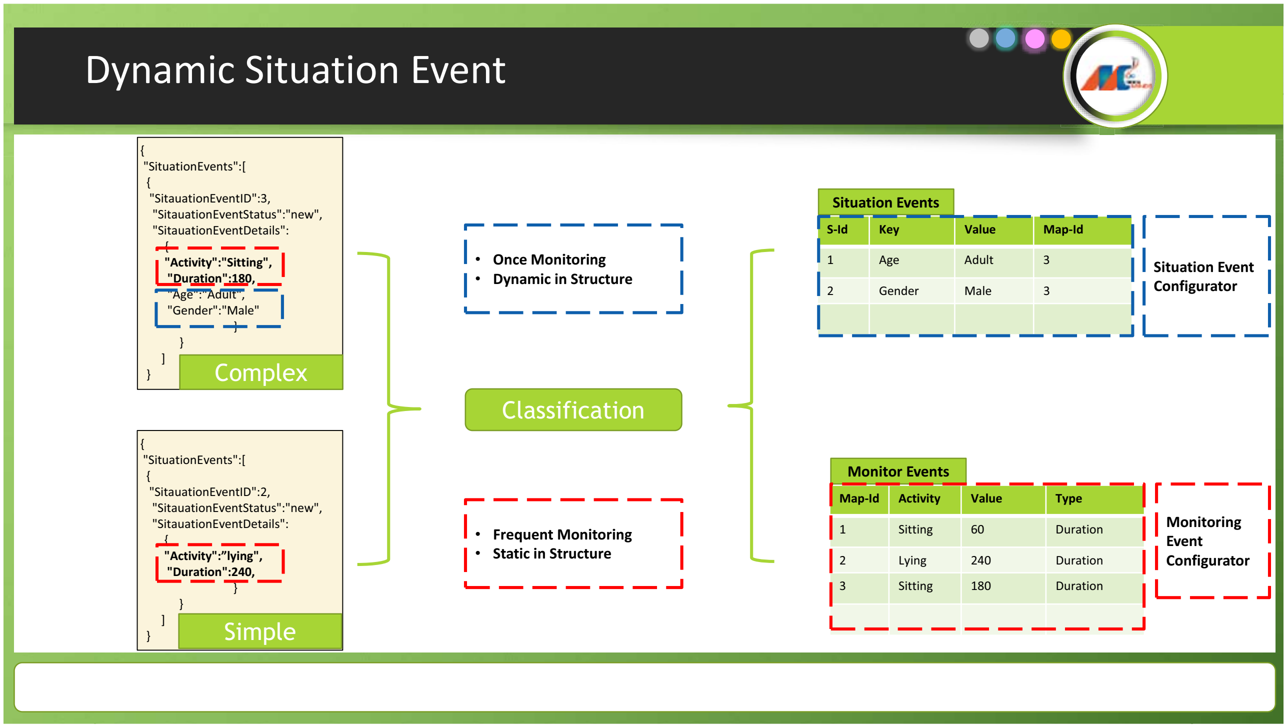
경희대학교 KYUNG HEE UNIVERSITY

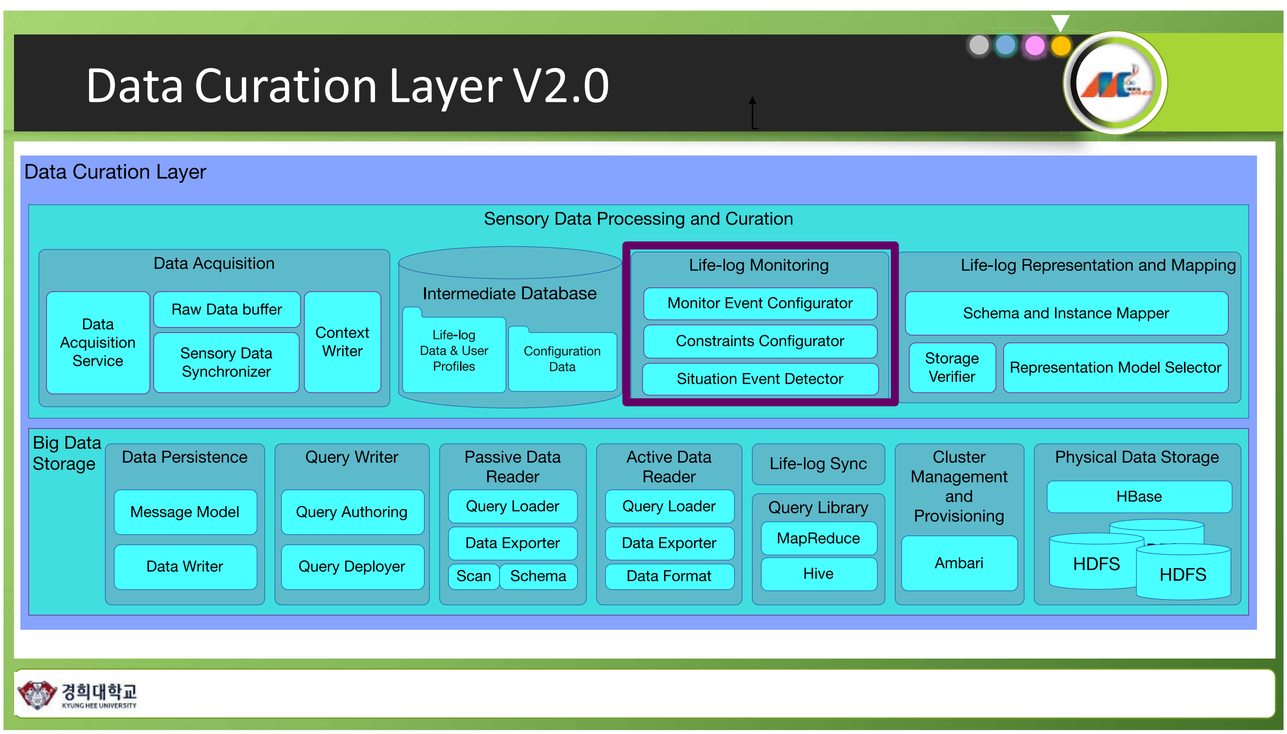
Taxonomy of Life-log Events



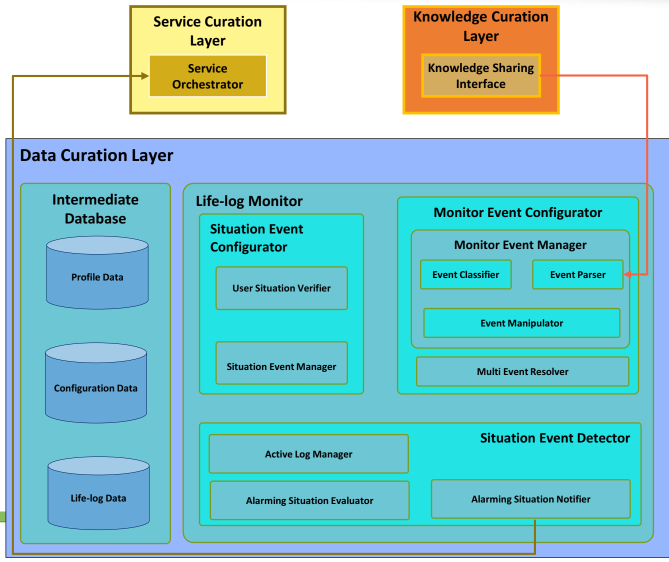


경희대학교 KYUNG HEE UNIVERSITY


Dynamic Situation Event
{
"SituationEvents":[
{
"SitauationEventID":3,
"SitauationEventStatus":"new",
"SitauationEventDetails":
{
"Activity":"Sitting",
"Duration":180,
"Age":"Adult",
"Gender":"Male"
}
}
]
}
Complex
{
"SituationEvents":[
{
"SitauationEventID":2,
"SitauationEventStatus":"new",
"SitauationEventDetails":
{
"Activity":"lying",
"Duration":240,
}
}
]
}
Simple
Once Monitoring
Dynamic in Structure
Classification
Frequent Monitoring
Static in Structure
Situation Events
S-Id | Key | Value | Map-Id
1 | Age | Adult | 3
2 | Gender | Male | 3
Situation Event Configurator
Monitor Events
Map-Id | Activity | Value | Type
1 | Sitting | 60 | Duration
2 | Lying | 240 | Duration
3 | Sitting | 180 | Duration
Monitoring Event Configurator

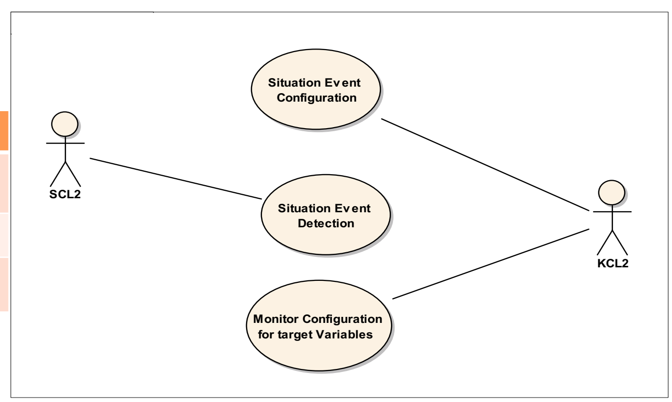

Data Curation Layer V2.0
Data Curation Layer
Sensory Data Processing and Curation
Data Acquisition
Data Acquisition Service
Raw Data buffer
Sensory Data Synchronizer
Context Writer
Intermediate Database
Life-log Data & User Profiles
Configuration Data
Life-log Monitoring
Monitor Event Configurator
Constraints Configurator
Situation Event Detector
Life-log Representation and Mapping
Schema and Instance Mapper
Storage Verifier
Representation Model Selector
Big Data Storage
Data Persistence
Message Model
Data Writer
Query Writer
Query Authoring
Query Deployer
Passive Data Reader
Query Loader
Data Exporter
Scan
Schema
Active Data Reader
Query Loader
Data Exporter
Data Format
Life-log Sync
Query Library
MapReduce
Hive
Cluster Management and Provisioning
Ambari
Physical Data Storage
HBase
HDFS
HDFS
경희대학교
KYUNG HEE UNIVERSITY

Life-log Monitor – Architecture



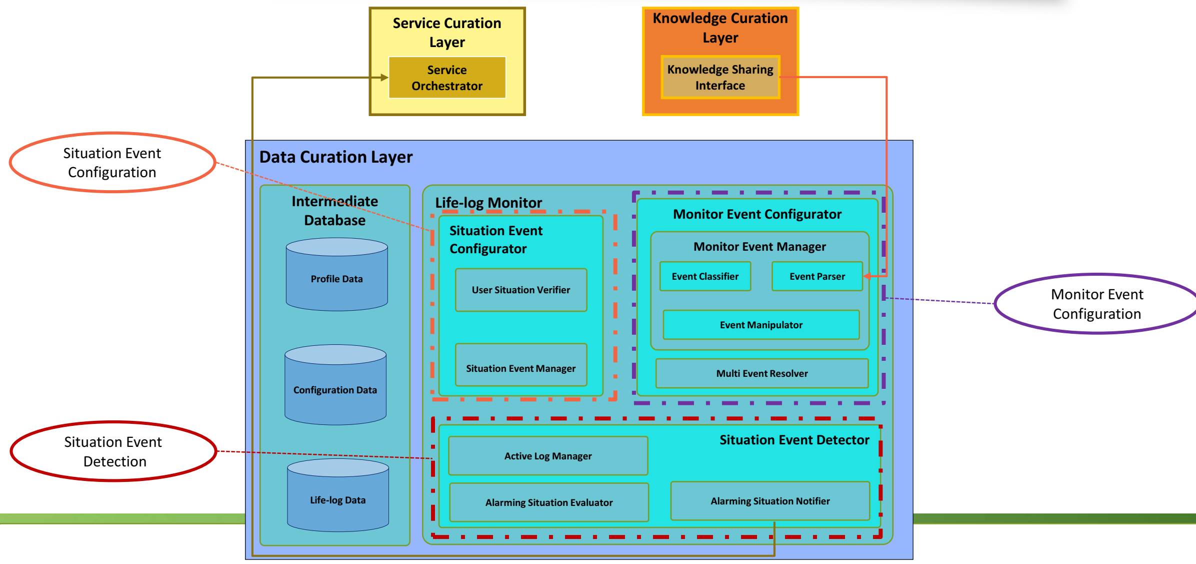
Use Case Model

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



경희대학교
KYUNG HEE UNIVERSITY

Life-log Monitor - Architecture

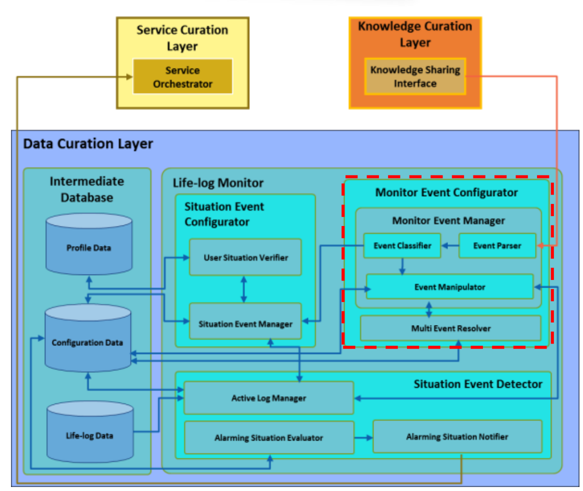


DCL2-SUC-10: Monitor Configuration for target Variables

Methodology Description

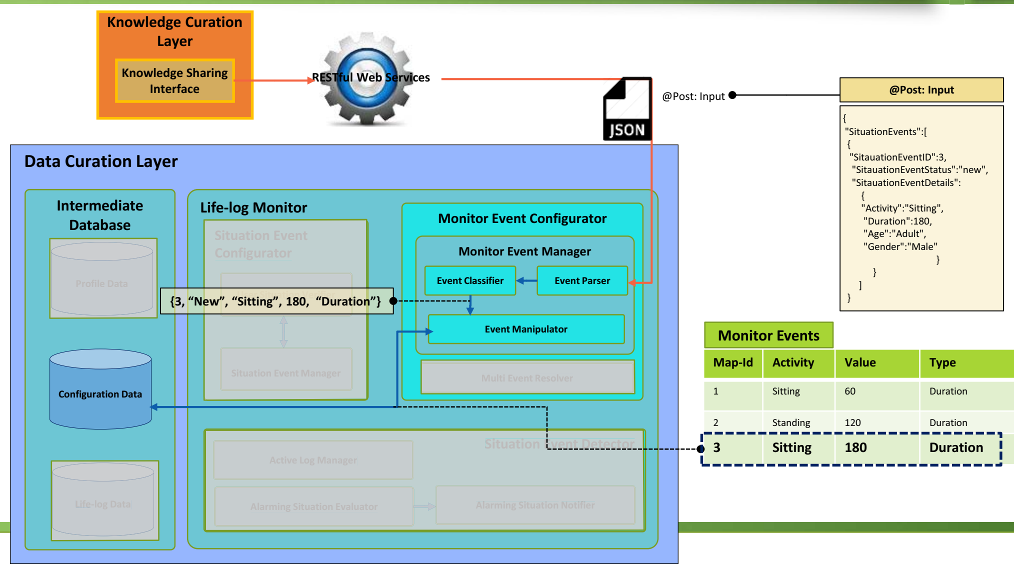
Use-case (DCL2-UCS-10): *Monitor Configuration for target variables.*

- **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 of common configuration format.
  - Life-log monitor is configured on the basis of the shared target variable.
  - Life-log monitor retrieve log data from Life-log against the target variables.
- **Output**
  - Target variable based Life-log data.



DCL2-SUC-10: Monitor Configuration for target Variables

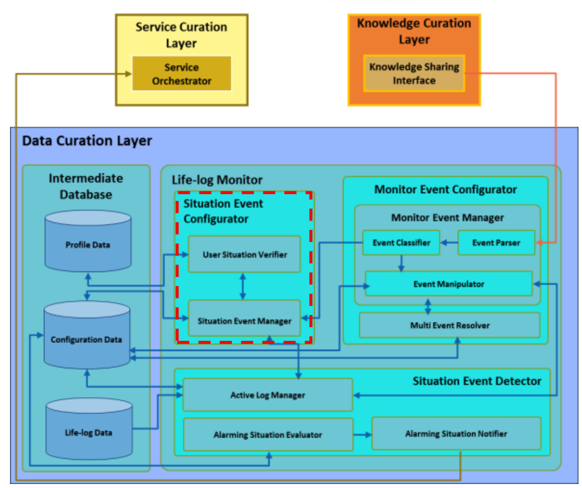
Implementation



DCL2-SUC-11: Situation Event Configuration

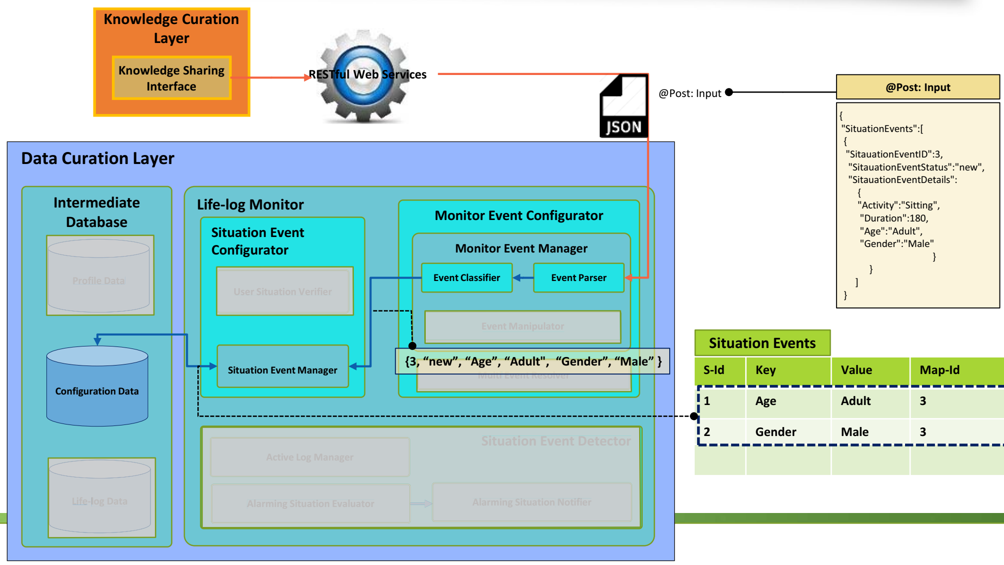
Methodology Description

- DCL2-SUC-11: *Situation Event Configuration*
- **Description**
  - Situation is determined by experts and is communicated to DCL 2 for monitoring the Life-log
- **Input**
  - Situation event in common configuration format
- **Process**
  - KCL 2 connects to DCL 2 and send the newly created situation in common configuration format.
  - DCL 2 evaluates the format of received situation configuration.
  - DCL 2 responds with acknowledgement message.
  - Situation will be parsed into components.
  - Parsed components are updated in persistent storage as per categories.
- **Output**
  - *Situation events.*



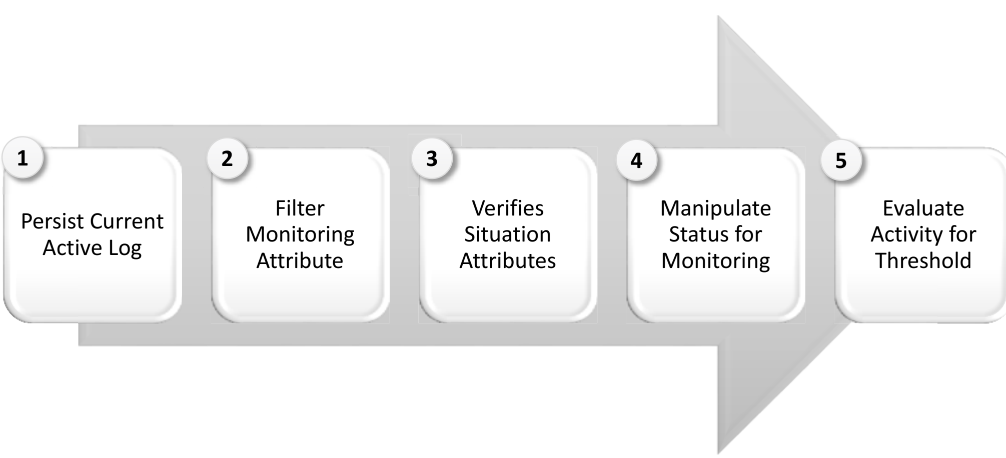
DCL2-SUC-11: Situation Event Configuration

Implementation



DCL2-SUC-12: Situation Detection

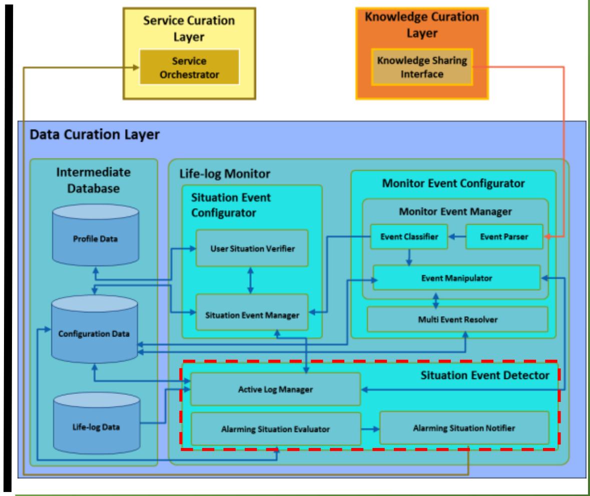
Step-1 | **Step-2** | **Step-3** | **Step-4** | **Step-5**



DCL2-SUC-12: Situation Detection

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

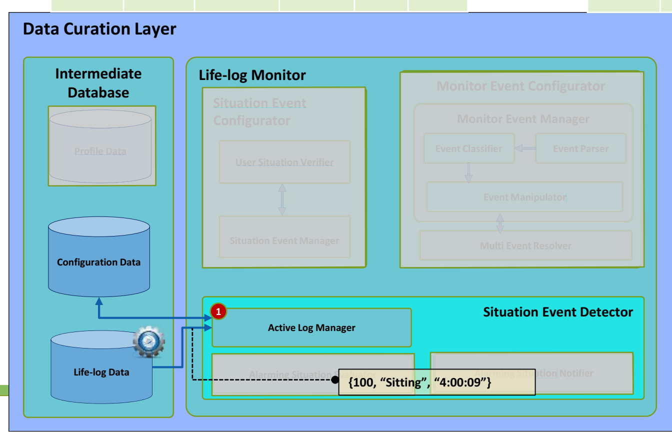
Active Lifelog — After

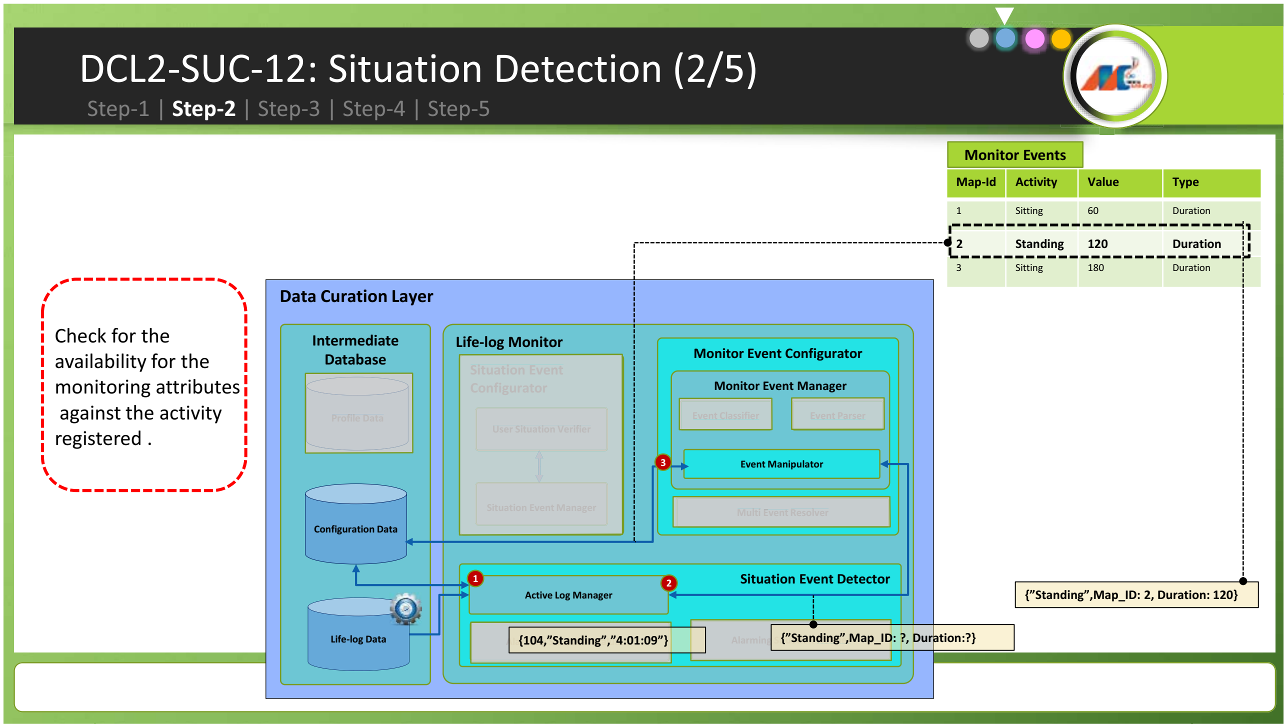
User_ID	Activity	Starttime	Duration	Map_ ID	Status
100	Sitting	4:00:00	60	1	CM
101	Standing	4:00:01	120	2	CM

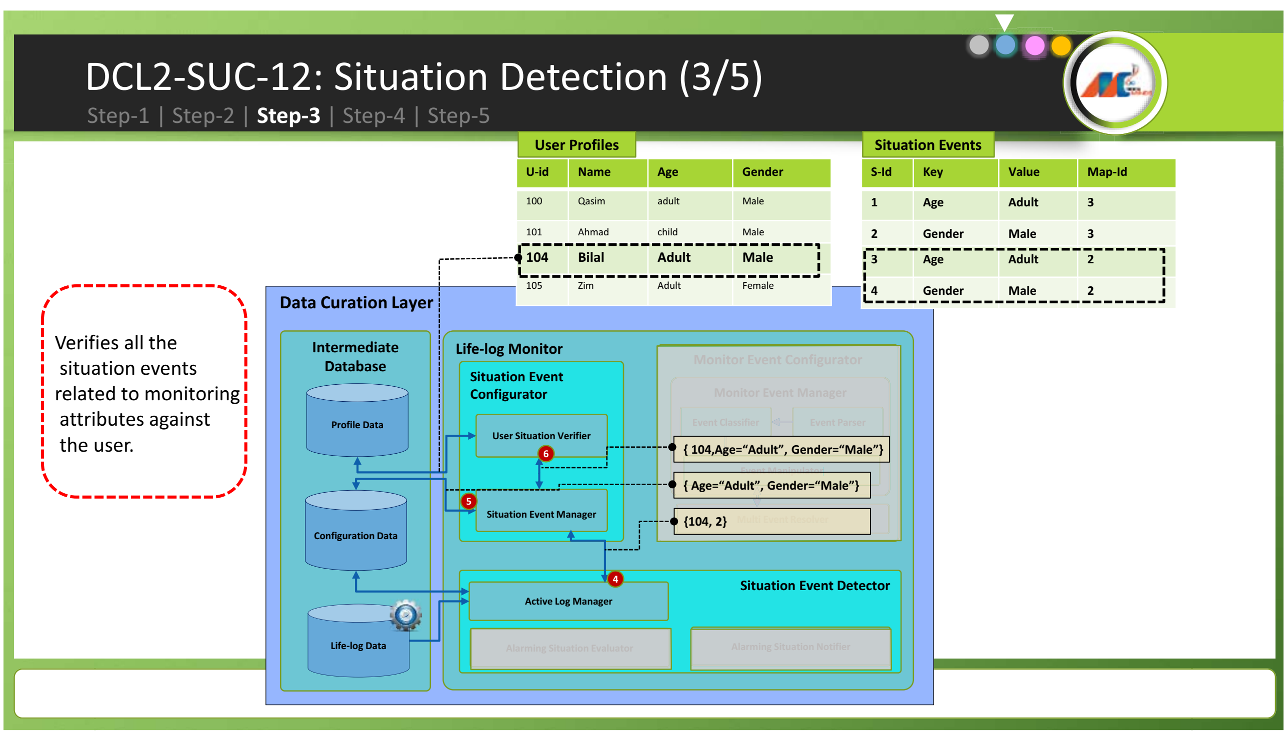
Active Lifelog — Before

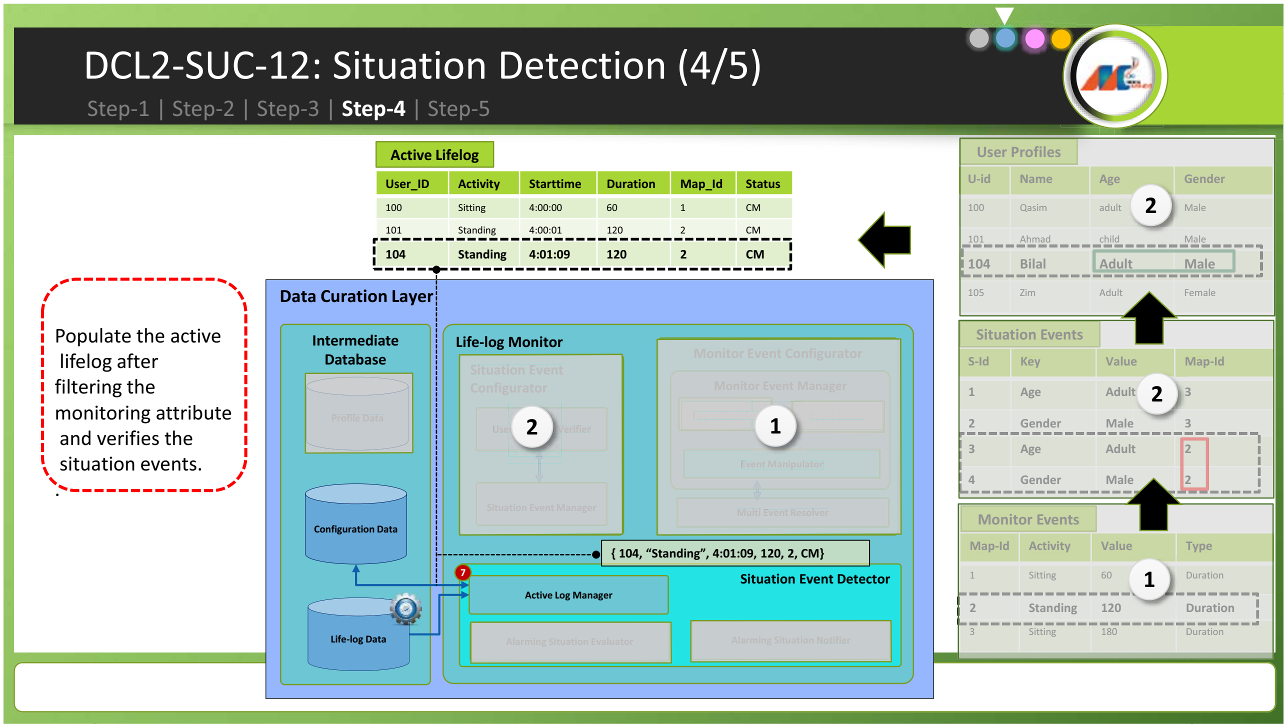
User_ID	Activity	Starttime	Duration	Map_ ID	Status
100	Sitting	4:00:00	60	1	CM
101	Standing	4:00:01	120	2	CM

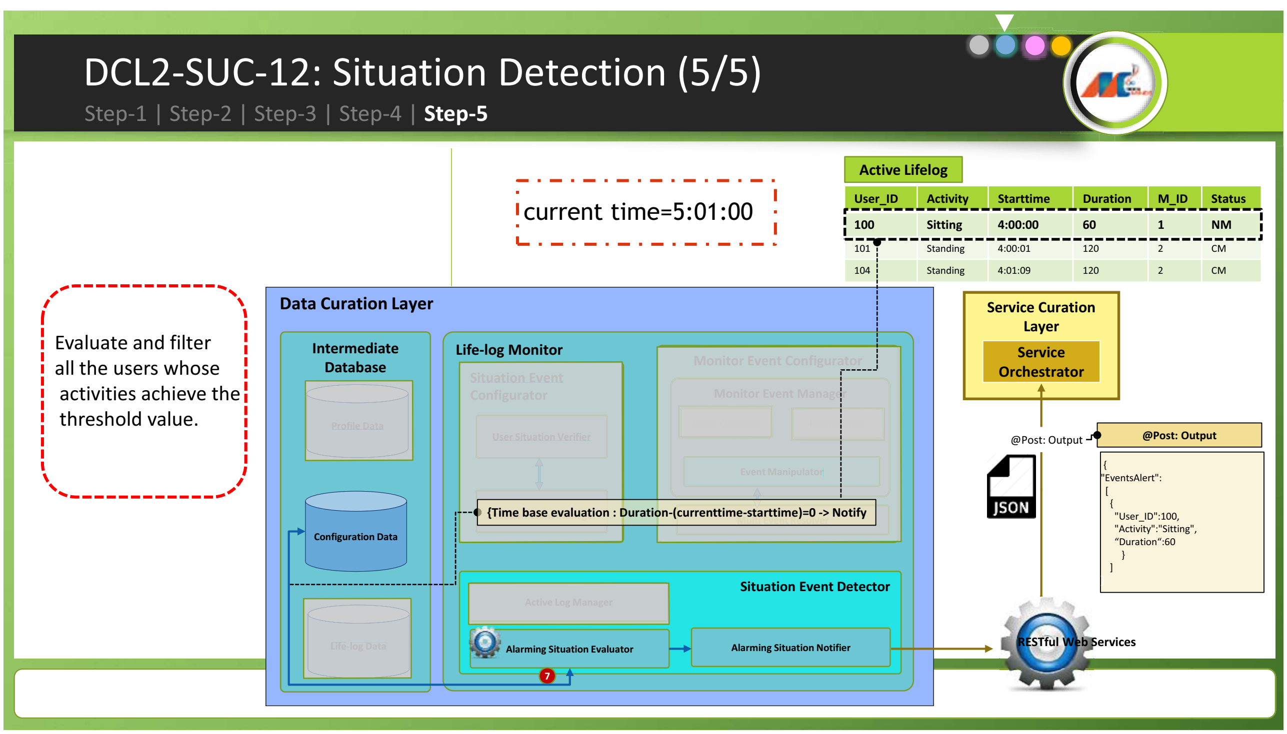
Avoid duplication of Activities, ignore all the reoccurring activities with respect to user.

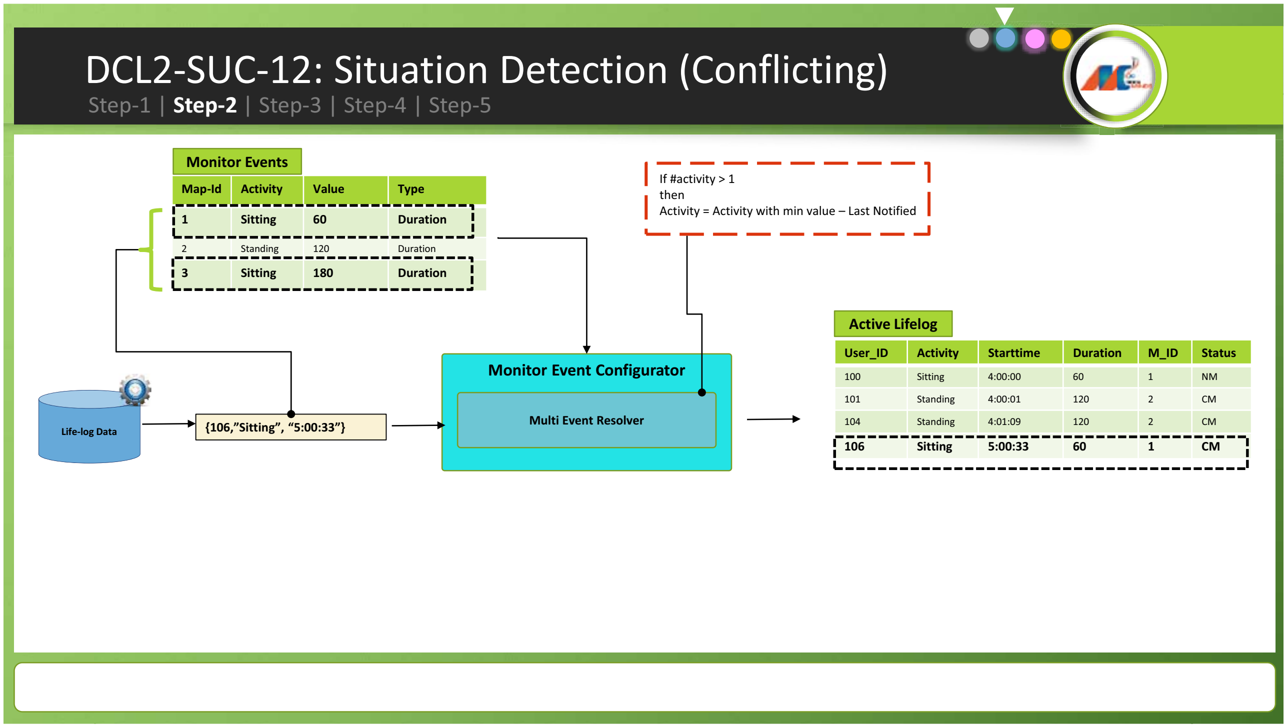



DCL2-SUC-12: Situation Detection (2/5)
Step-1 | Step-2 | Step-3 | Step-4 | Step-5
Check for the availability for the monitoring attributes against the activity registered .
Monitor Events
Map-Id
Activity
Value
Type
1
Sitting
60
Duration
2
Standing
120
Duration
3
Sitting
180
Duration
Data Curation Layer
Intermediate Database
Profile Data
Configuration Data
Life-log Data
Life-log Monitor
Situation Event Configurator
User Situation Verifier
Situation Event 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
Verifies all the situation events related to monitoring attributes against the user.
User Profiles
U-id
Name
Age
Gender
100
Qasim
adult
Male
101
Ahmad
child
Male
104
Bilal
Adult
Male
105
Zim
Adult
Female
Situation Events
S-Id
Key
Value
Map-Id
1
Age
Adult
3
2
Gender
Male
3
3
Age
Adult
2
4
Gender
Male
2
Data Curation Layer
Intermediate Database
Profile Data
Configuration Data
Life-log Data
Life-log Monitor
Situation Event Configurator
User Situation Verifier
Situation Event Manager
Monitor Event Configurator
Monitor Event Manager
Event Classifier
Event Parser
{ 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
Populate the active lifelog after filtering the monitoring attribute and verifies the situation events.
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
Data Curation Layer
Intermediate Database
Configuration Data
Life-log Data
Life-log Monitor
Monitor Event Configurator
Situation Event Configurator
{ 104, "Standing", 4:01:09, 120, 2, CM}
Situation Event Detector
Active Log Manager
User Profiles
U-id Name Age Gender
100 Qasim adult Male
101 Ahmad child Male
104 Bilal Adult Male
105 Zim Adult Female
Situation Events
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


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

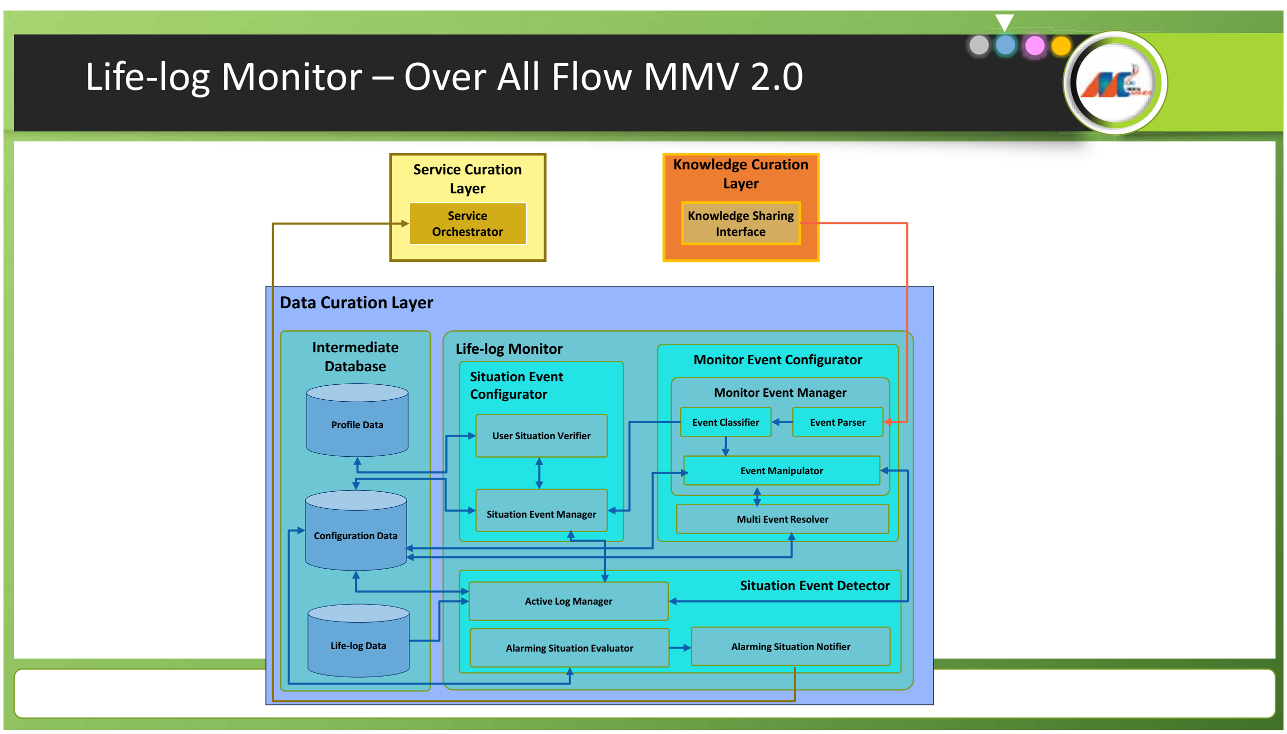


Monitor Events

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

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



Issues

- Static situation
  - Identification of activity fluctuation
  - Threshold of activity interrupts
- Situation Detection
  - Time based activity aggregation per person
  - Generalize routine to detect the situation
  - Scalability for multiple activities
- Dynamic Situation
  - Real time integration of situation
  - Real time situation communication format

Selected Tools and Techniques

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

Present & Future Work

- Current Progress
  - Surveying techniques to monitor the life-log
  - Access data from life-log
  - Accumulating of log for situation detection
  - Evaluation of situation as per expert's defined rules
  - Identification and designing of components
  - Identify the common configuration format
- Future Plan
  - Threshold determination and scheduling for monitoring Life-log
  - Identify the common configuration format
  - Generalization in defining monitoring attributes
  - Dynamically accommodation of expert's situation configuration
  - Scalability for multiple users at a time.

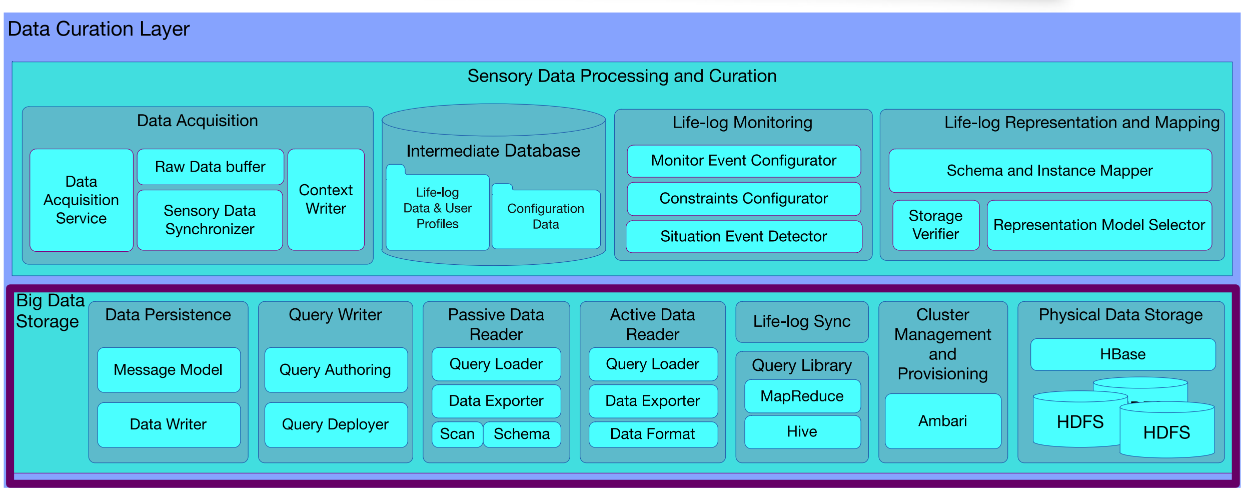
Contribution

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

Data Curation Layer (DCL 2.0)

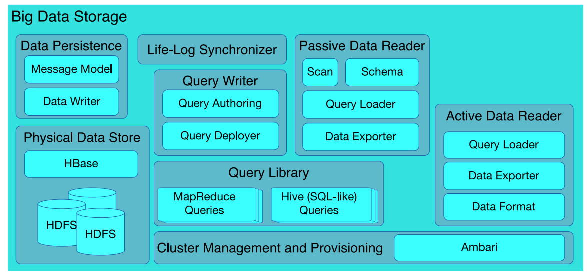
Usman Akhtar
Chosoungho

DCL V2.0 Architecture



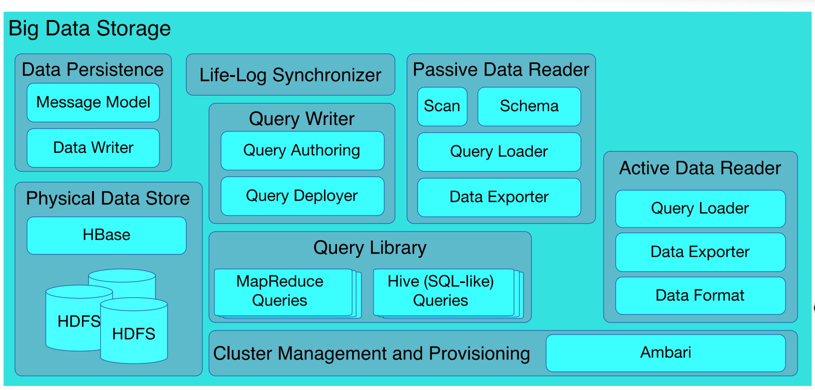
Role of Big Data Storage in Mining Mind

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

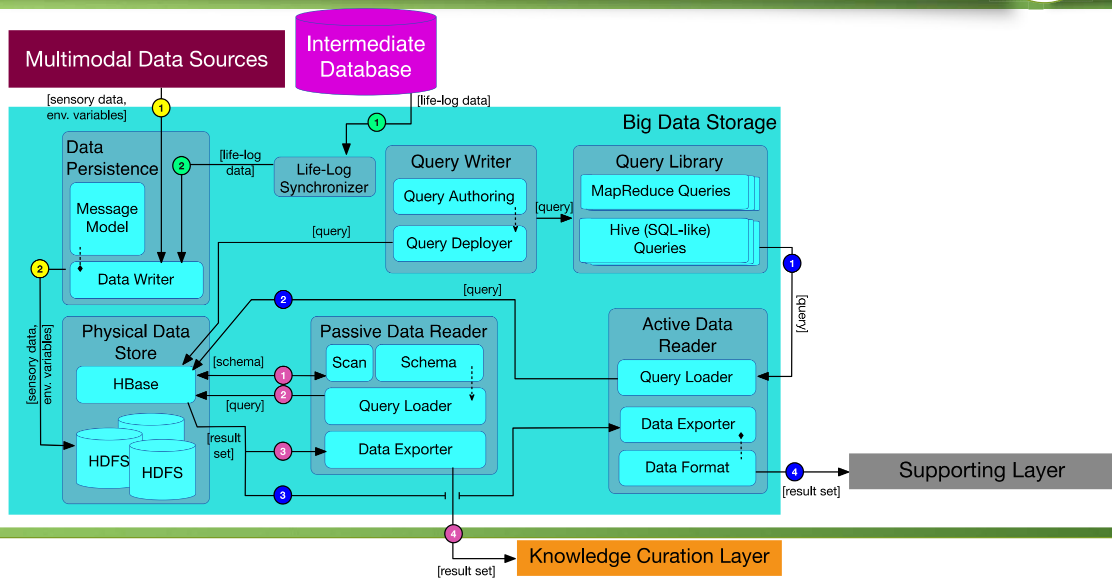


- How is it different from Ver. 1.5 ?
  - Data Raw sensory and environmental data was only written in Big Data Storage
  - Read access for data was not available

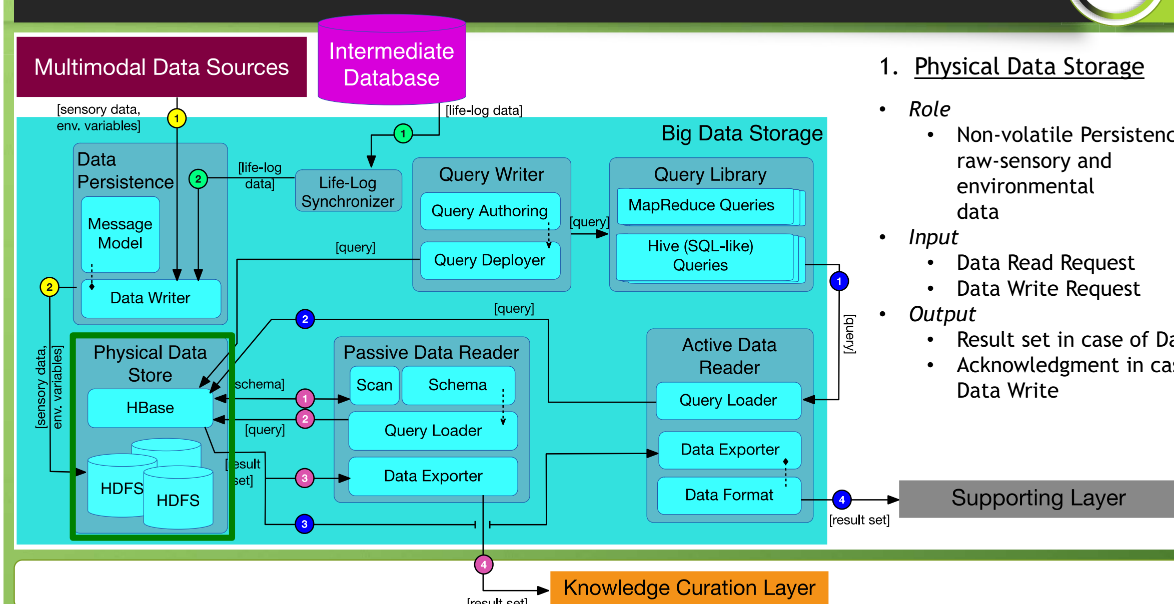
Big Data Storage Architecture and Technologies



Functional Diagram



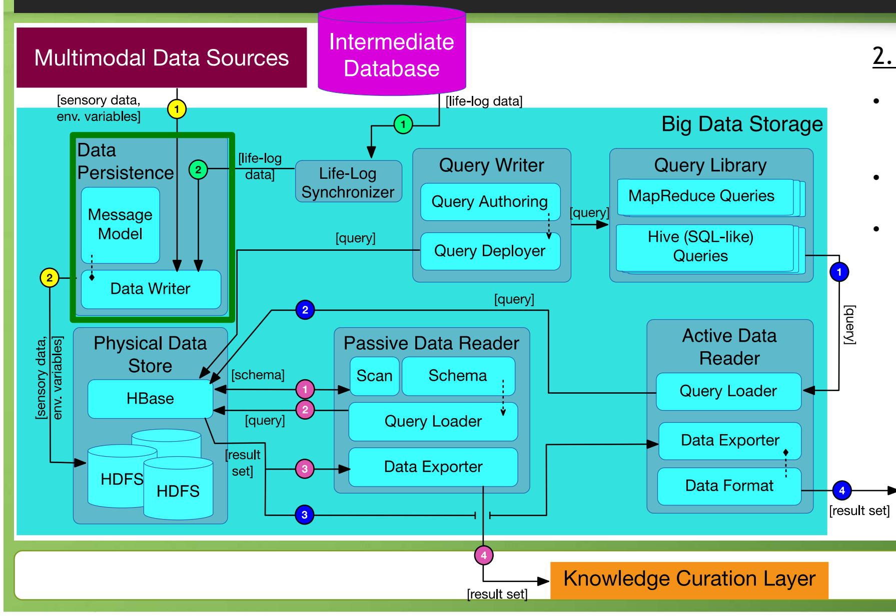
Big Data Storage Components



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

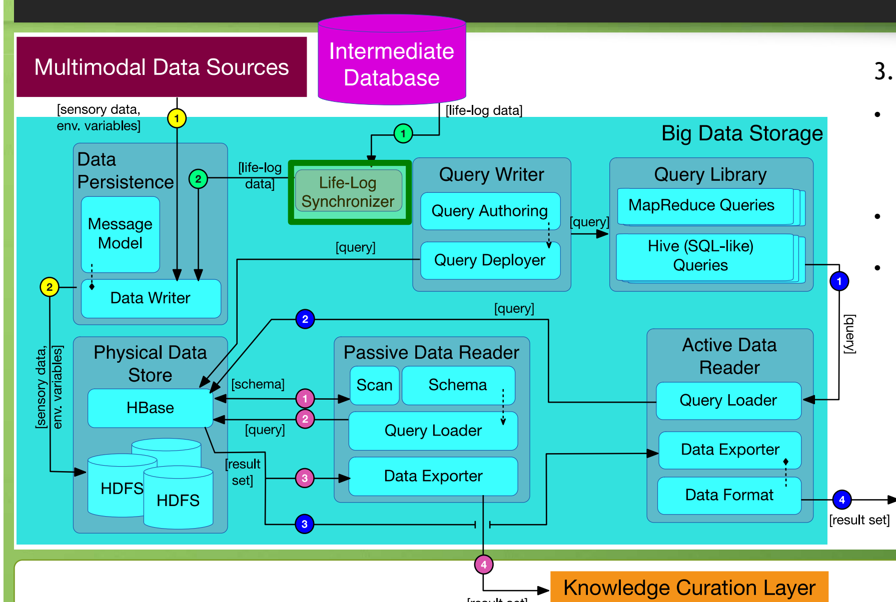
Big Data Storage Components



2. Data Persistence

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

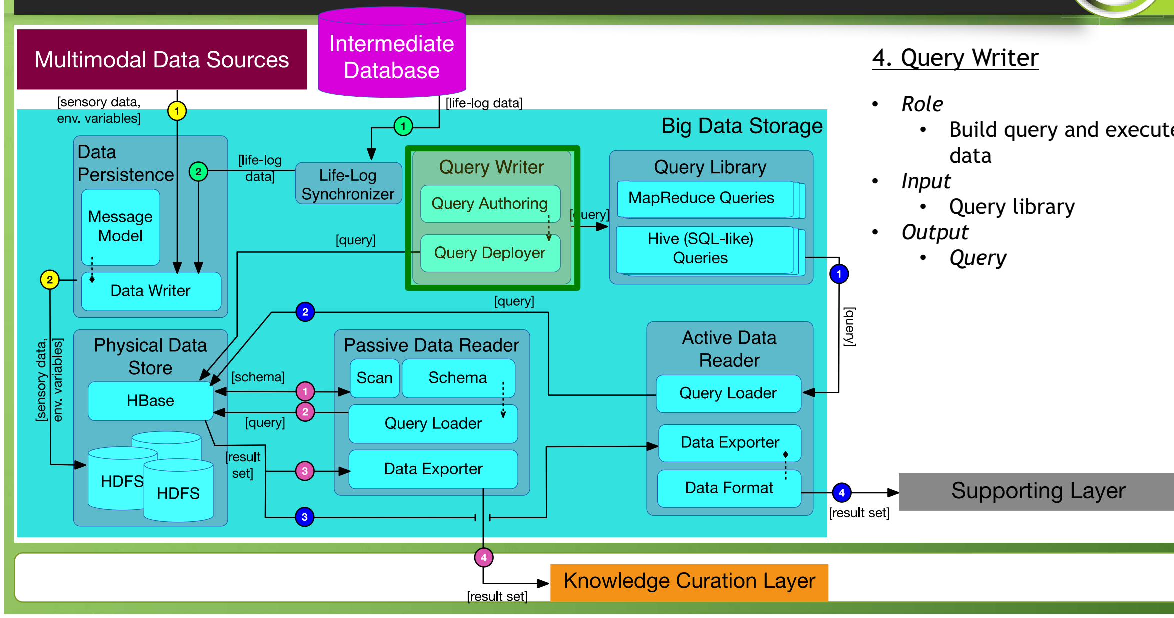
Big Data Storage Components



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

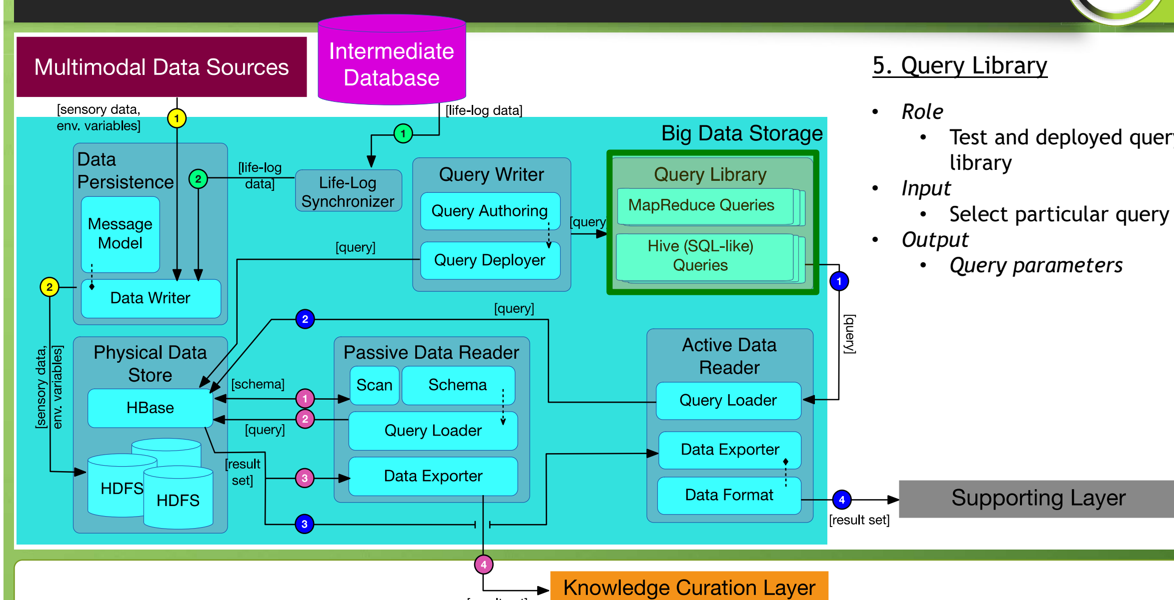
Big Data Storage Components



4. Query Writer

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

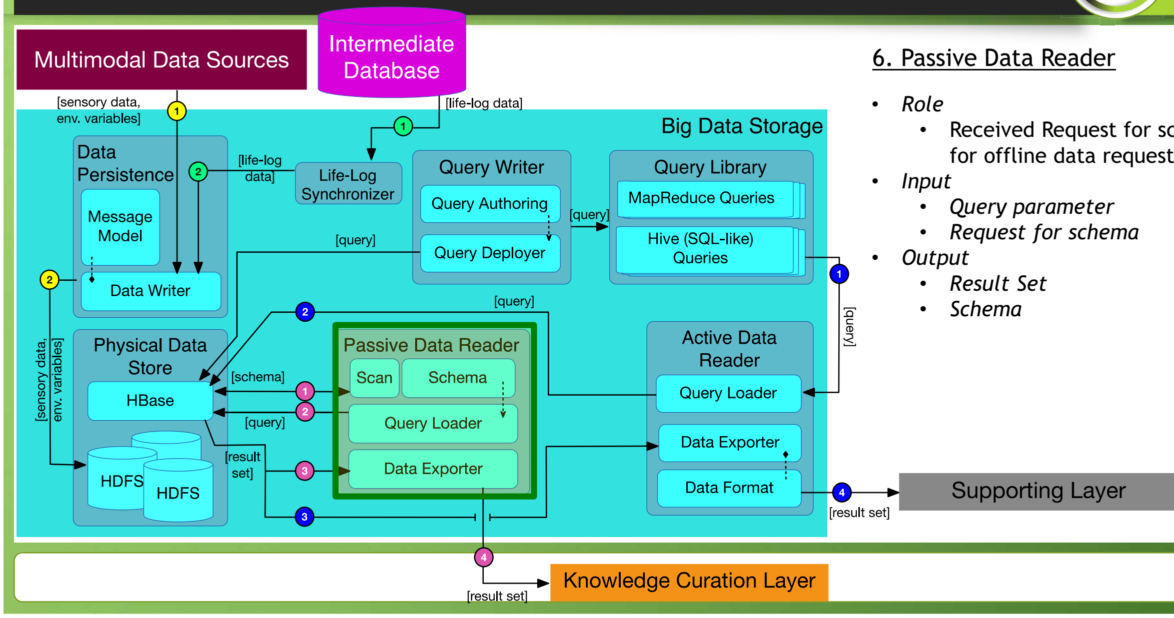
Week 6: Big Data Storage Components



5. Query Library

- *Role*
  - Test and deployed query library
- *Input*
  - Select particular query
- *Output*
  - *Query parameters*

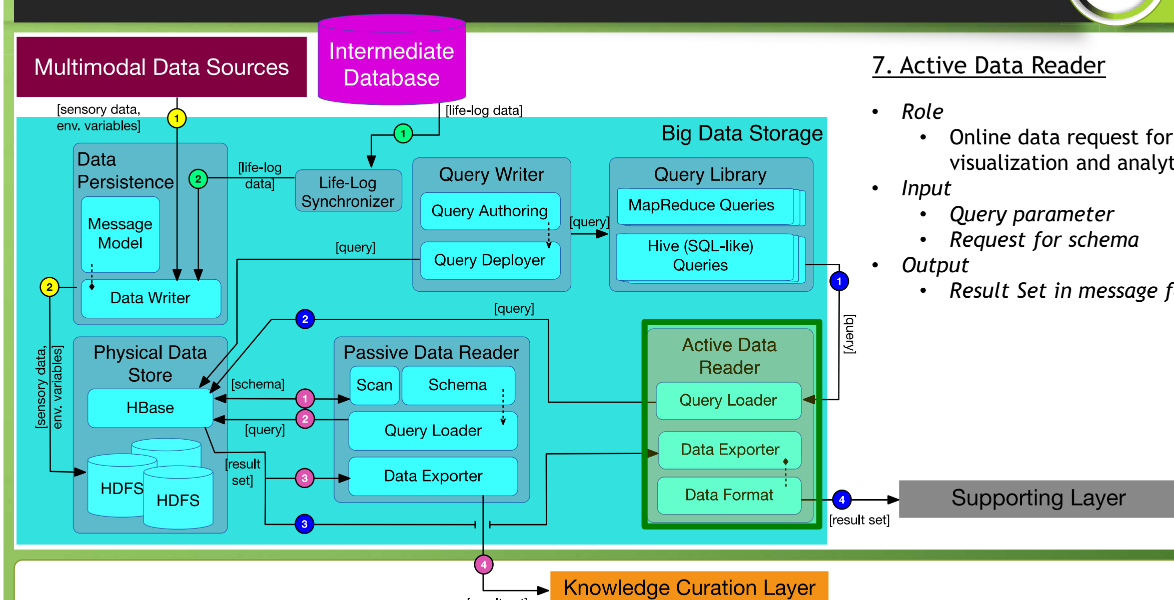
Big Data Storage Components



6. Passive Data Reader

- *Role*
  - Received Request for schema for offline data request
- *Input*
  - *Query parameter*
  - *Request for schema*
- *Output*
  - *Result Set*
  - *Schema*

Big Data Storage Components



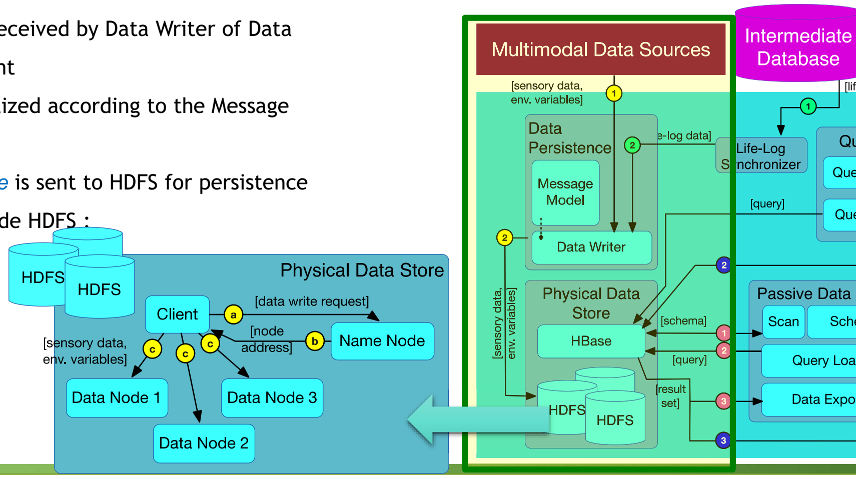
7. Active Data Reader

- *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

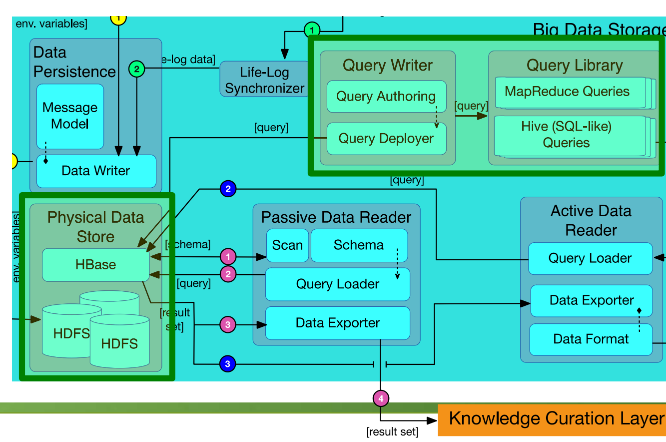
1. Raw Sensory data is received by Data Writer of Data Persistence Component
   i. Data is de-serialized according to the Message Model
2. *De-serialized message* is sent to HDFS for persistence
3. Data gets written inside HDFS :



Execution Scenario #2

Offline Query Authoring

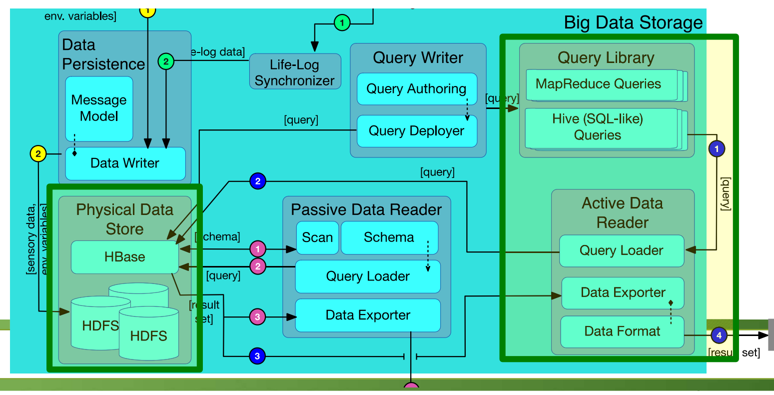
1. Big Data Storage Requirements are converted into MapReduce or Hive Queries
2. Queries are written using Eclipse as an IDE
3. Query is tested and deployed as part of Query Library in Big Data Storage



Execution Scenario #3

Online Data Request for Data Visualization and Analytics

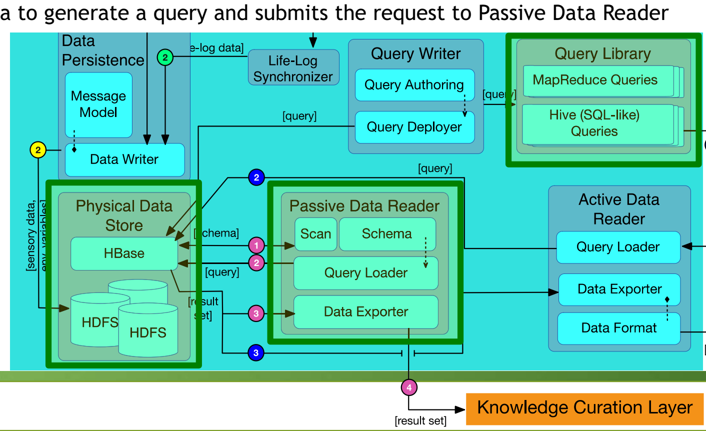
1. Data read request is generated by Visualization and Analytics Components of SL
    I. Query parameters are sent for query selection and execution
2. *Active data reader* selects the particular query depending upon the query parameters
3. Selected Query is sent to Hbase for execution
4. Required data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



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

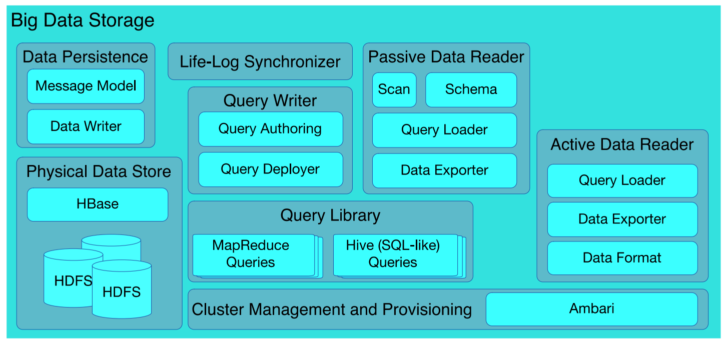


Week 7: Plan to Upgrade Private Cloud

- We want to shift the new architecture implementation on the private cloud.

- We will incorporate the Following major Components:

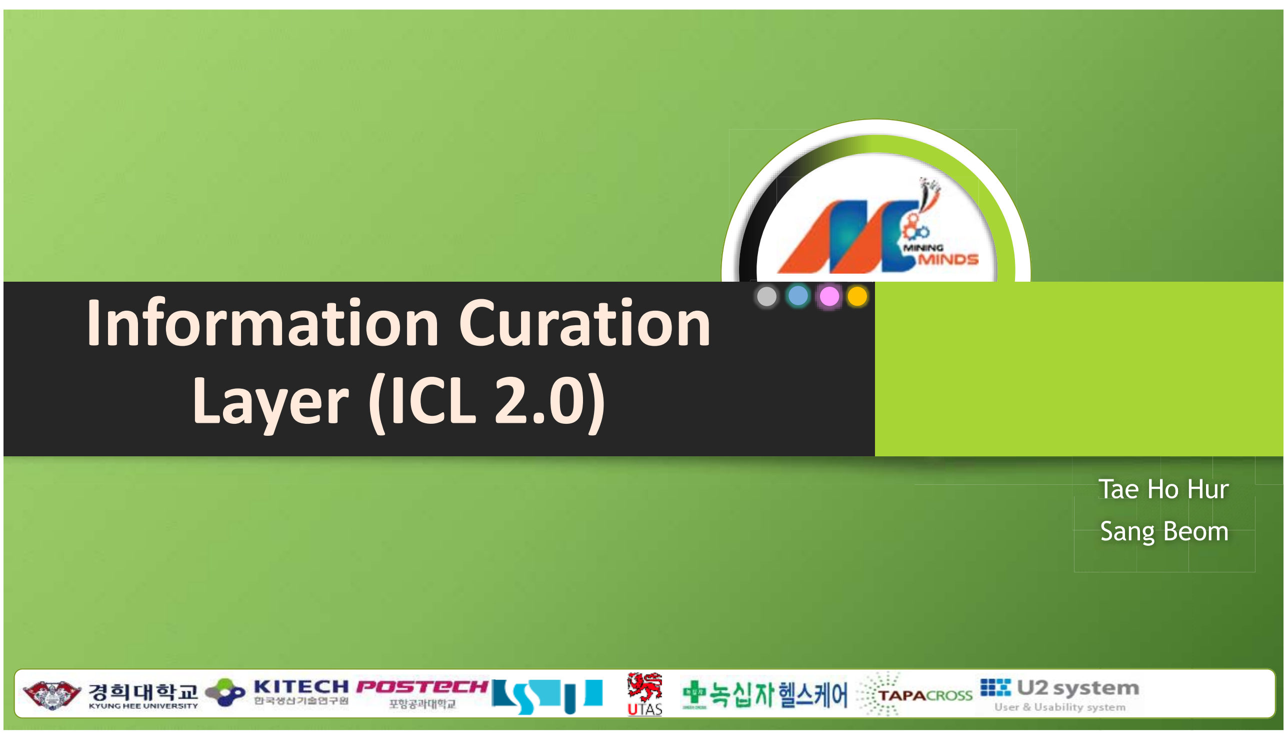
1. HBase
2. Hive
3. Apachi Ambari
4. Eclipse Integration



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.
- Overall Big Data *Cluster Monitoring*


Information Curation Layer
(ICL 2.0)
Principal Investigator:
Professor Sungyoung Lee
MINING MINDS
경희대학교
KYUNG HEE UNIVERSITY
KITECH
POSTECH
UTAS
녹십자헬스케어
TAPACROSS
U2 system
User & Usability system


Information Curation
Layer (ICL 2.0)
Tae Ho Hur
Sang Beom
MINING MINDS
경희대학교
KYUNG HEE UNIVERSITY
KITECH
POSTECH
UTAS
녹십자헬스케어
TAPACROSS
U2 system
User & Usability system

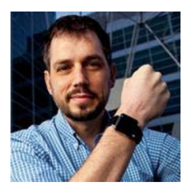
Introduction

- 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

- 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

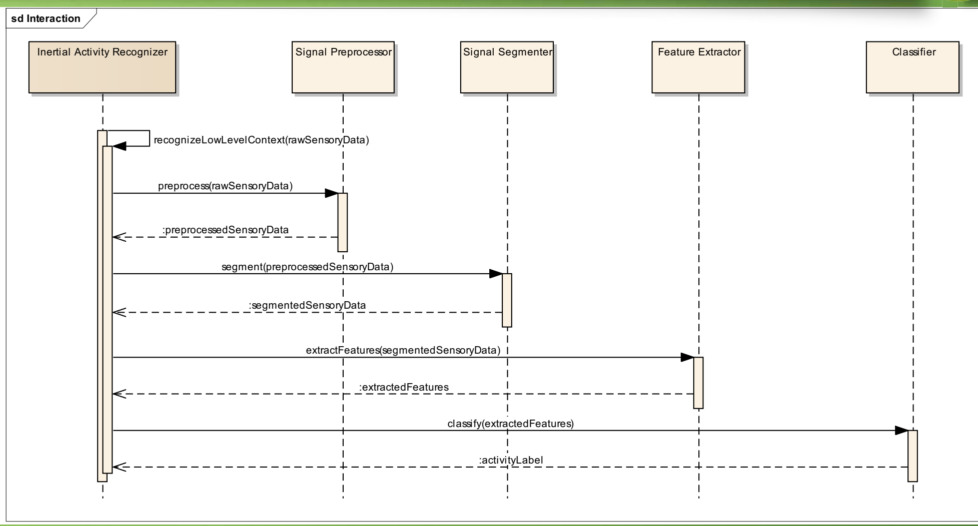
Authors	Published year	Sensor placement	Sensor type	Techniques	Limitation
Chun zhu et al. [1]	2009	Waist, Foot	ACC, Gyro	• Two feed-forward neural networks fusion • Heuristic segmentation	• Few basic activities • Offline evaluation
Jun-Ki Min et al. [3]	2011	Head, two arms and two wrists, fingers	ACC, 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

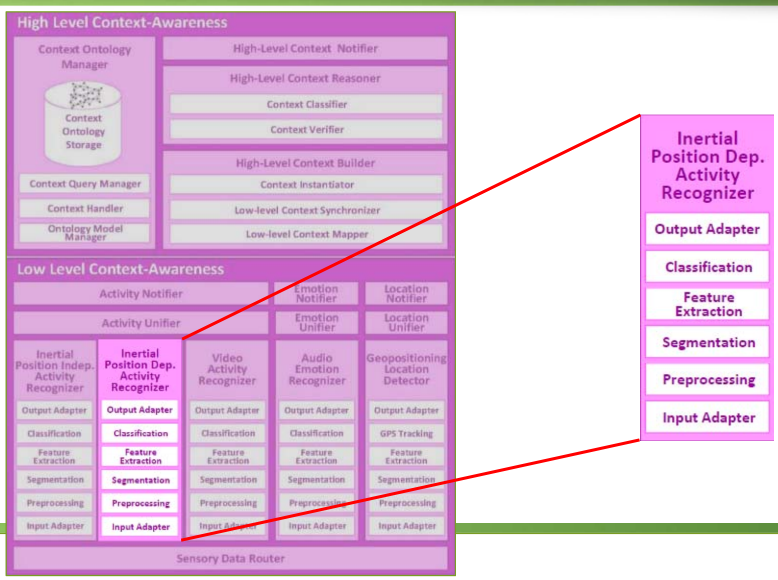
- Position dependency
  - Some of the sensor devices are attached in fixed position while some doesn't such as a smartphone inside a pocket.
  - Satisfying these two different characteristics into one AR model is hard to accomplish

- 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

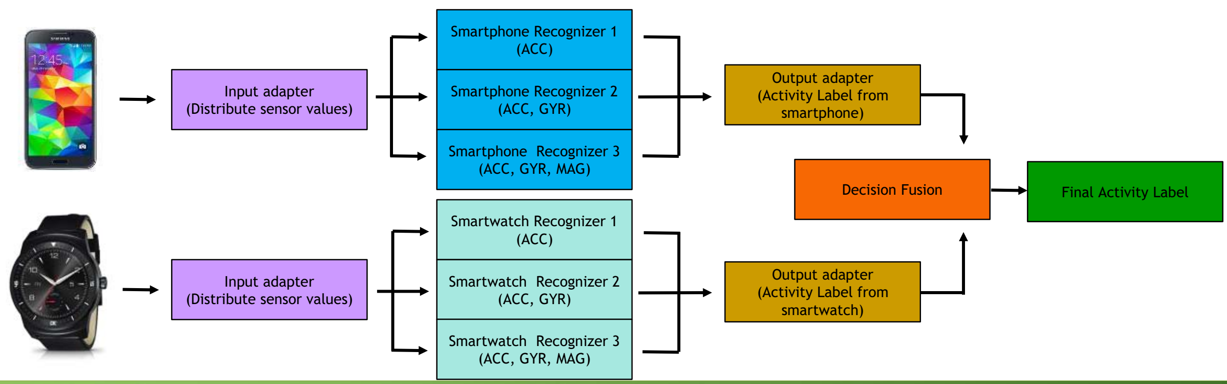


Architecture in Mining Minds

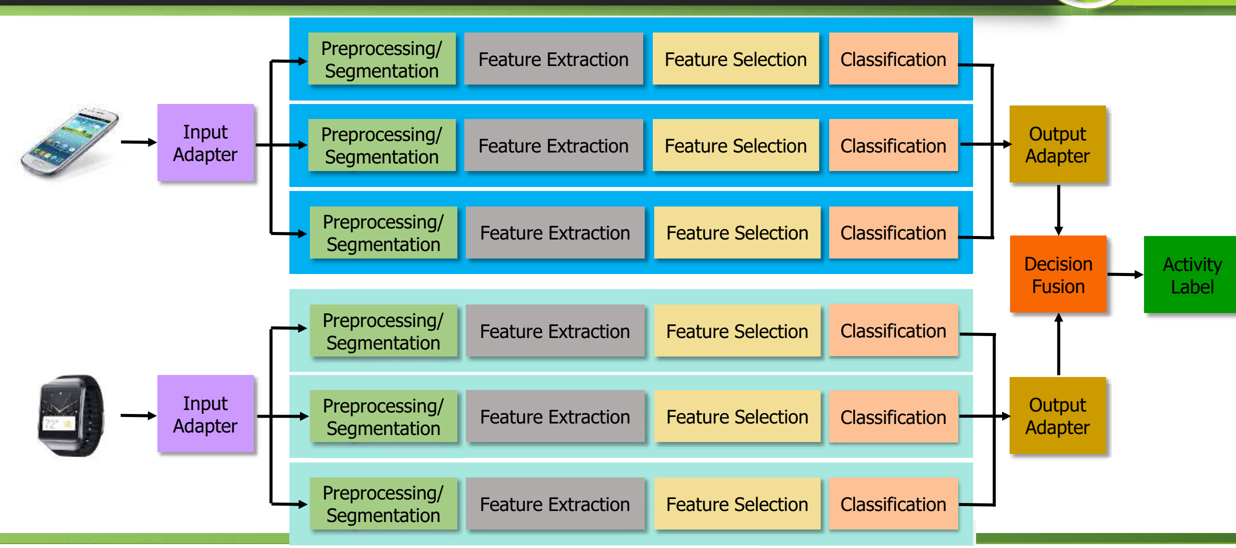


Proposed method

- 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

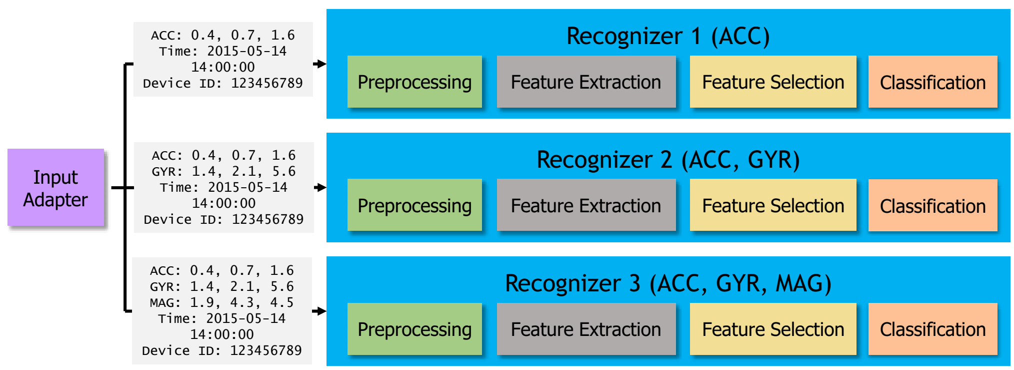


Flow chart (Abstract)



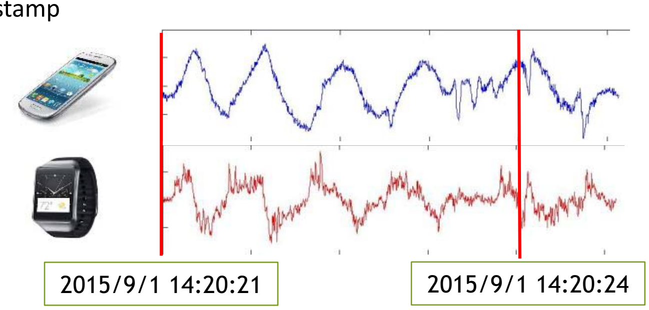
Input Adapter

- Distribute sensor values to each recognizer



Preprocessing & Segmentation

- Data synchronization
  - Based on timestamp



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

Feature Extraction & Selection

- Use both Time-domain features and Wavelet analysis
  - Time domain features
    - mean, median, variance, skewness, kurtosis, Zero crossing
  - Wavelet analysis
    - Discrete wavelet analysis
- Forward–backward search
  - Features are sequentially added and removed
  - Optimal features can be identified depending on the resulting classification accuracies for each feature subset

	A1	A2	.....	An
E1	0.1	0.3		0.2
E2	1.2	0.2		0.7
....				
Em	0.2	1.8		0.3

↓

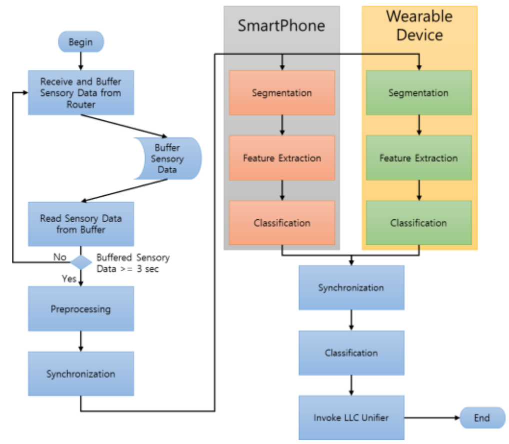
	A1	A2
E1	0.1	0.3
E2	1.2	0.2
....		
Em	0.2	1.8

Classification

- Combine different classifiers
  - To improve the performance of individual classifiers by combining their output using different techniques
- AdaBoost
  - AdaBoost is a type of adaptive boosting that incrementally trains classifiers by suitably increasing the pattern weights
  - It combines multiple weak classifiers to create a single more powerful one

	Stand	Sit	Lie	Walk	Run
Stand	0.51	0.04	0.01	0.45	0.00
Sit	0.04	0.49	0.00	0.44	0.04
Lie	0.02	0.00	0.90	0.06	0.01
Walk	0.24	0.17	0.04	0.55	0.00
Run	0.00	0.00	0.00	0.02	0.98

Flowchart



Devices and Tools

- 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

- Experiment #1 : Using Acc (Integrated Model)
  - Algorithm : SVM
  - Feature : Min, Max, Average, Deviation for each Axis

```
=== Confusion Matrix ===

    a    b    c    d    e    f    g    h    i    j    k    l   <-- classified as
  165    0    1    0    0    0    0   94    0    0    0    0 |    a = Climbing
    0  280    2    0    0    1    3   13    3    0    1    0 |    b = Dancing
    6    7  307    0    0    0    0   70    0    0    0    0 |    c = Descending
    0    0    0  110    0  283    0    0    0    0    0    1 |    d = Eating
    0    0    0    0  269    0    0    2    6    0    0    0 |    e = Running
    0   12    6    7    0 2665   69   25    0    0    0   10 |    f = Sitting
    0    9    0    0    0  151 1355    6    0   18    9    0 |    g = Standing
    6    9   21    0    0   51   12 3071    0    1    3    0 |    h = Walking
    0    1    0    0    4    0    0   11  354    0    0    0 |    i = Jogging
    1    3    0    0    0    1   14    0    0  747    5    0 |    j = Stretching
    0    2    2    0    0    0    5    0    0    6  760    0 |    k = Sweeping
    0    0    0    0    0  207    0    0    0    0    0  192 |    l = Lying
```

Experiment Result(Cont)

- Experiment #1 : Using Acc/Gyro (Integrated Model)
  - Algorithm : SVM
  - Feature : Min, Max, Average, Deviation for each Axis

```
=== Confusion Matrix ===

    a    b    c    d    e    f    g    h    i    j    k    l   <-- classified as
   18    1    2    0    0    0    0  239    0    0    0    0 |    a = Climbing
    1  260    4    0    0    1    7   17    5    7    1    0 |    b = Dancing
    0    8  234    0    0    0    0  148    0    0    0    0 |    c = Descending
    0    0    0   12    0  381    0    0    0    0    0    1 |    d = Eating
    0    0    0    0  267    0    0    1    9    0    0    0 |    e = Running
    0   12    6    3    0 2674   61   28    0    1    1    8 |    f = Sitting
    0   13    0    0    0  150 1343    8    0   21   13    0 |    g = Standing
    0   15   25    0    0   51   11 3063    0    3    6    0 |    h = Walking
    0    2    0    0    9    0    0   13  346    0    0    0 |    i = Jogging
    1    2    0    0    0    0   15    2    0  744    7    0 |    j = Stretching
    0    4    0    0    0    0    8    5    0   20  738    0 |    k = Sweeping
    0    0    0    0    0  212    0    0    0    0    0  187 |    l = Lying
```

Future work

- Determine feature list
  - September 21st ~ October 2nd
- Implementation
  - September 21st ~ October 16th

Conclusion

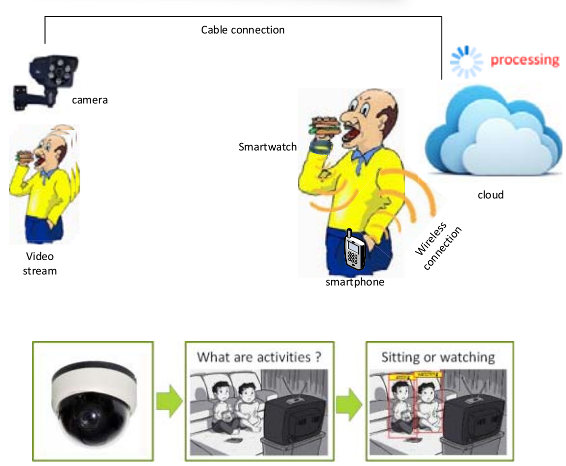
- MM v1.0 recognized 5 basic activities with smartphone
  - High recognition rate but few activities recognized
- MM v1.5 recognized 8 daily activities with smartphone & smartwatch
  - Increased types of activities with additional sensor
  - Weak on robustness & reliability
- MM v2.0 will recognize more activities with higher accuracy
  - Use of Multiple Activity Recognizer will guarantee robustness and reliability
  - Incremental learning classifier will increase the accuracy

Information Curation Layer (ICL 2.0)

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

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

- **Goal**: Design and implement a method that is able to recognize human activities from video.
- **Objectives**
  - Design a video-based activity recognizer
  - Detect and recognize indoor activities with overall accuracy over 80%
  - Implement and integrate to ICL 2.0

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…

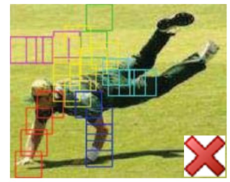
Related Works

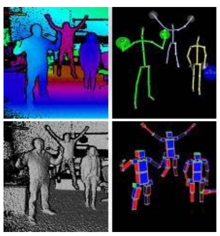
Authors	Year	Dimension	No. Activity	Activity types	Accuracy	Key points	Limitation
Gu	2010	3D	8	Single action	94%	Hidden Markov Mode	High complexity
Ofli	2013	3D	12	Single action	80%	Annotated joint feature	Low recognition rate
Vantigodi	2013	3D	12	Single action	96%	Temporal joint distance feature	
Kruthiventi	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 …

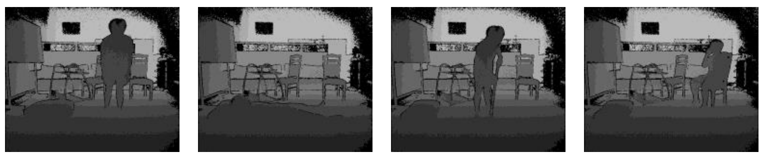
Limitation of Existing Works

- 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

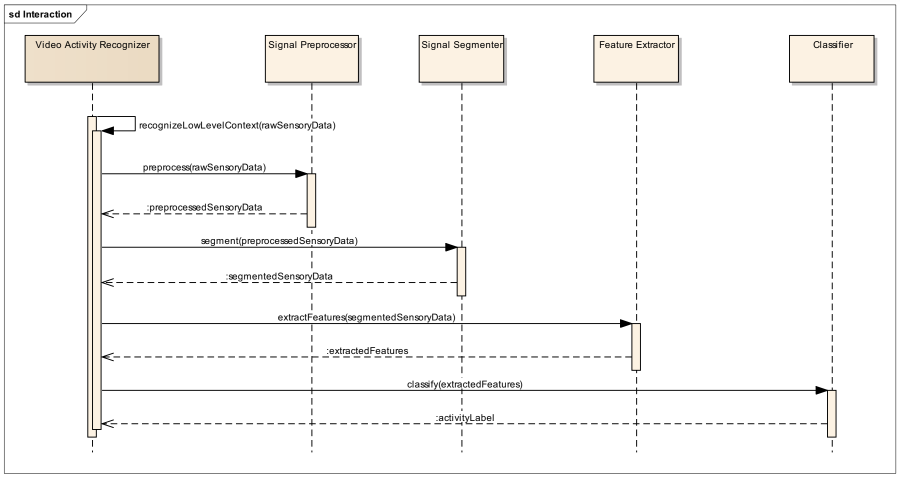
Why Kinect ?

- 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

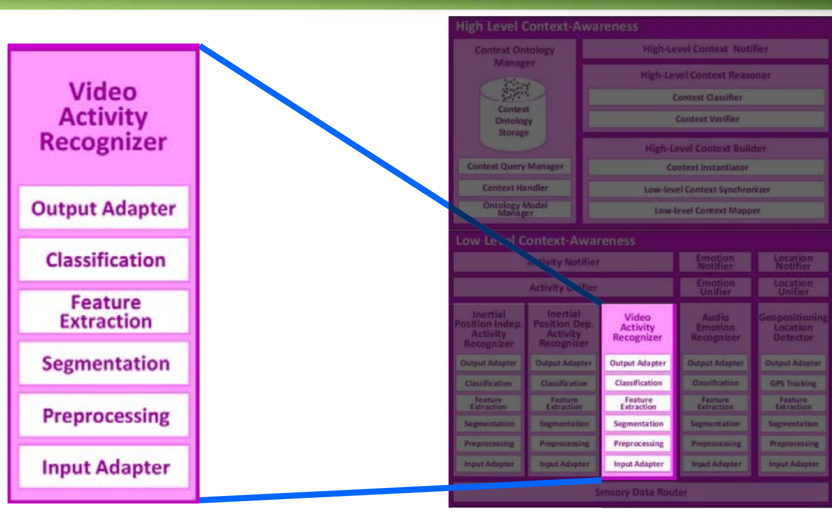
Limitation

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

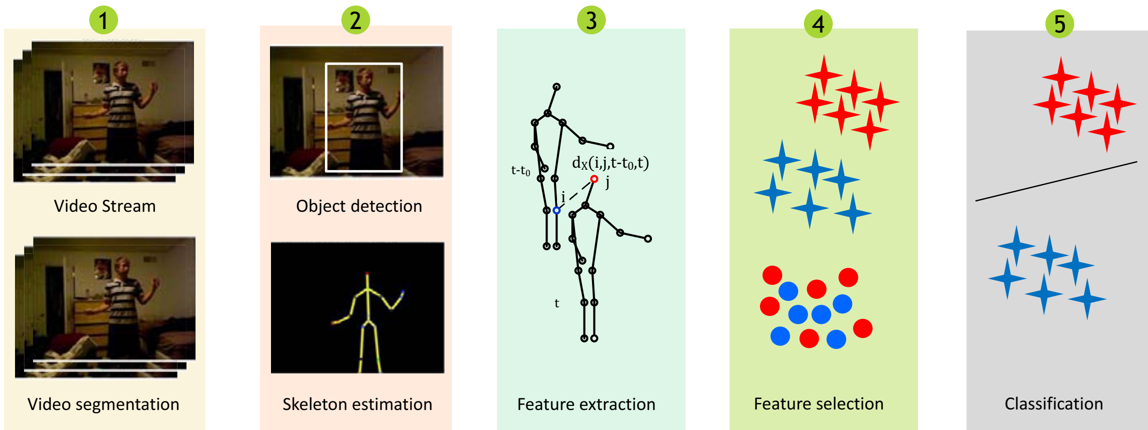
Sequence diagram for VAR



Architecture –ICL 2.0

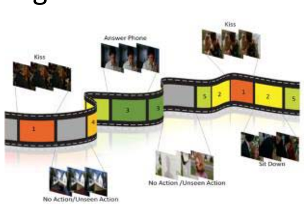


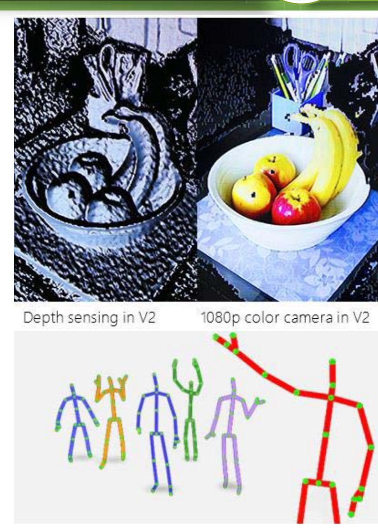
Work Flow



Video segmentation

- Video specification provided by Kinect device:
  - RGB stream 1080p, 30 fps.
  - Depth stream 512x424, 30 fps.
  - Body stream 25 joints, 30 fps
- Segment the video stream from DCL into short windows with length of 1s
- Recognize activities for each window





Preprocessing

- The output from Microsoft Kinect device to DCL will be
  - Body joint coordinate

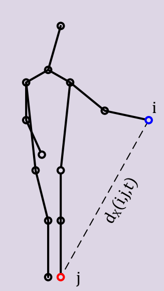
→ Do not use color stream because of privacy issue

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

KINECT FOR DEVELOPERS

Feature extraction

Spatial Joint Distance Feature



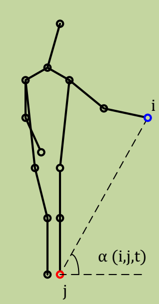
Spatial Joint Distance Feature is measure Euclidean distance between 2 joints of an object in the current frame.

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

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

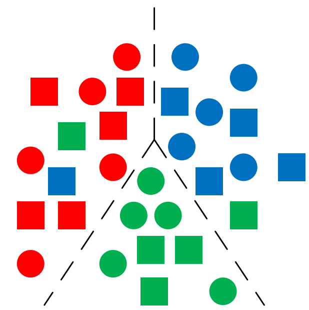
Why we need feature selection ?

- 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

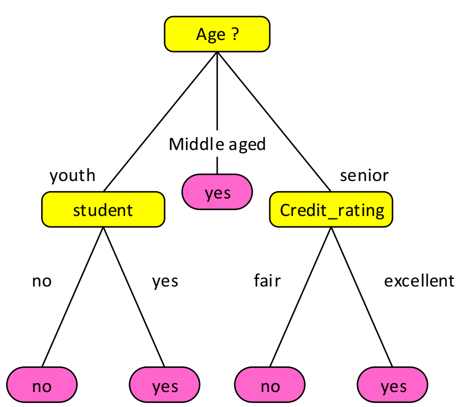
Feature ranking

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



Square or circle is better ?

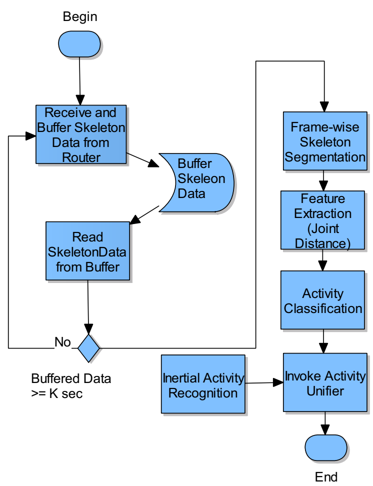
Decision tree



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



경희대학교
KYUNG HEE UNIVERSITY

Tools/Technologies

Windows 8

Java

ORACLE

MATLAB & SIMULINK

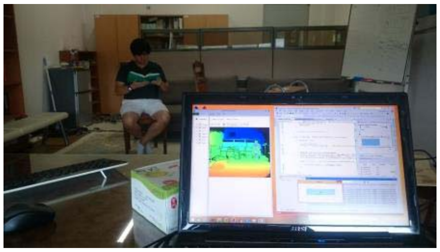
Dataset collection

Scenario:

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

Device:

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

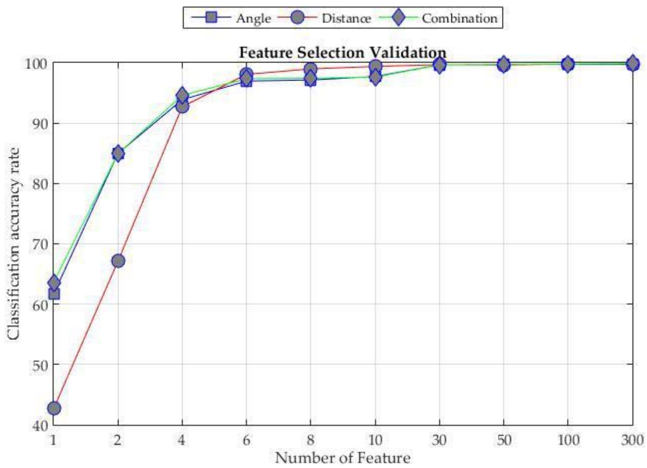


Output data:

- Body frame: 30 frames/second

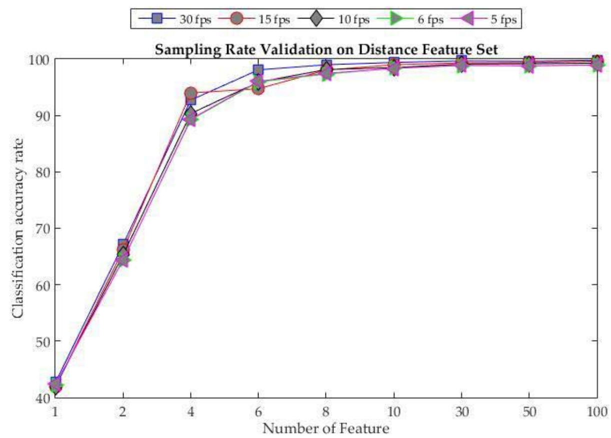
Evaluation results

- Benchmark the algorithm with different types and number of feature
  - Feature types: joint distance, angle, and combination
  - Number of used features
- Less than 6 features, angle is better than distance.
- Achieve 99% in accuracy at 10-features using distance metric
- Higher accuracy, higher computational cost



Evaluation results

- To preserve the smooth transmission from the Kinect to cloud, the sampling rate should be reduced.
- Benchmark the algorithm with various values of sampling rate, for example as 30fps (default), 15 fps, 10 fps, 6 fps, and 5 fps.
- Overall accuracy is generally reduced on the sampling rate factor.
- To get the good tradeoff between the accuracy, reality for data transmission, and computational cost, algorithm is employed with 5fps of sampling rate, 30-distance selected features.



Contribution

- 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

- Apps for data collection (Kinect+Laptop) → [Aug 11th]
- Experimental setup design for dataset collection [Aug 11th]
- ICL Meeting [Aug 12th]
- Collection of multimodal dataset [Aug 13th-15th]
- Models definition and evaluation (on real data) [Aug 17th-31st]
- Report partial results to Dr Banos (by email) [Aug 25th]
- ICL Meeting (final results) [Sep 1st]
- Presentation of final results in regular meeting [Sept 5th]

Information Curation Layer (ICL 2.0)

Le Ba Vui

Introduction

- 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

- 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

- **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

Related Works – Datasets

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

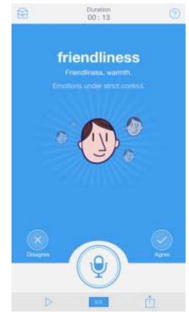
Related Works – Methodologies

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

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



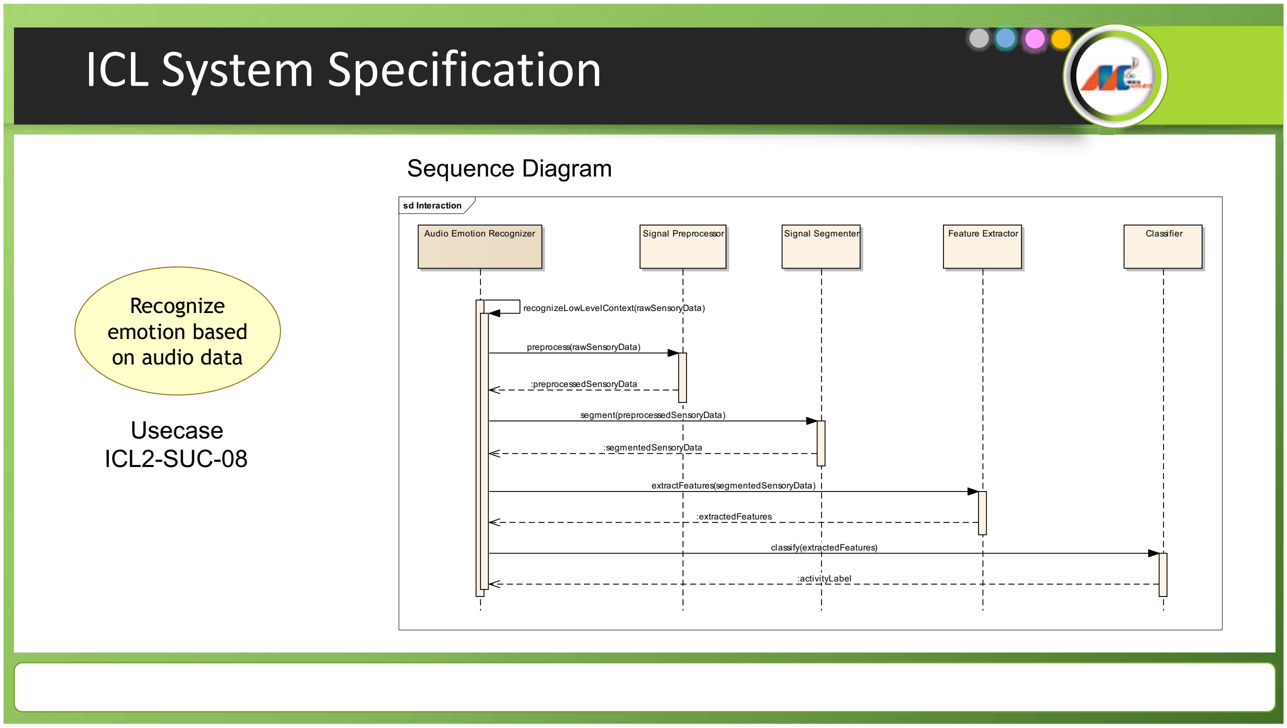
emospeech
Speech Emotion Recognition

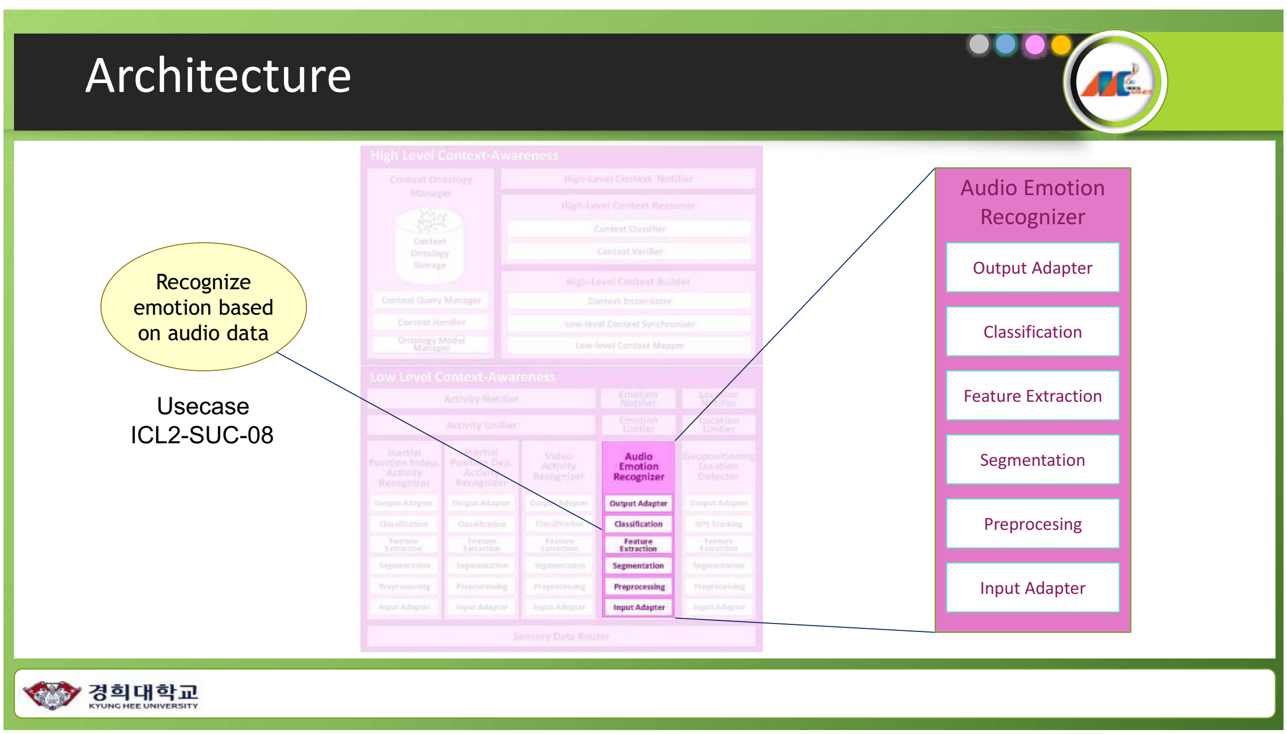
경희대학교
KYUNG HEE UNIVERSITY

Limitations of existing works

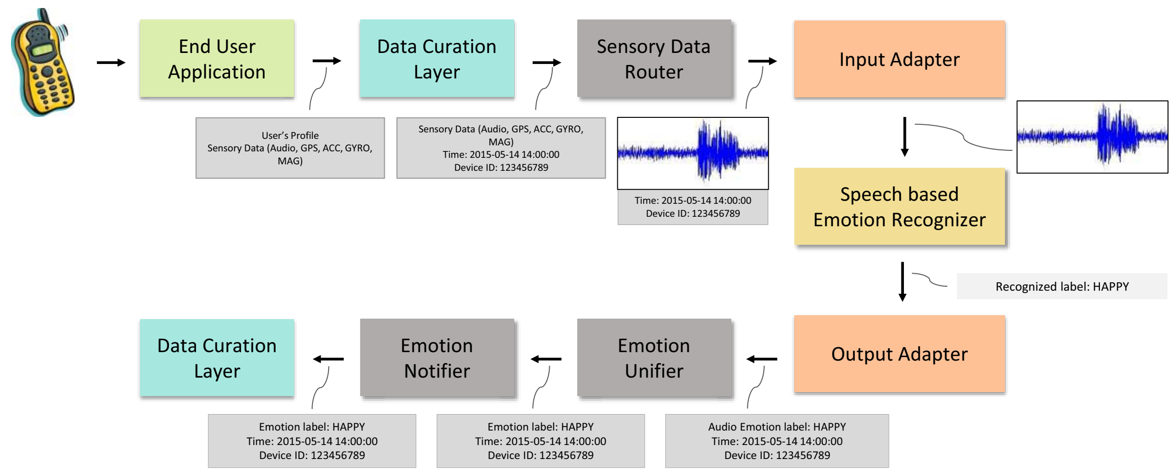
- Just focus on a particular problem
- Lack of preprocessing: noise reduction, voice activation detection, segmentation
- No consideration of personality
- There is no services utilizing emotion information: context modeling, recommendation, etc.

경희대학교
KYUNG HEE UNIVERSITY


ICL System Specification
Sequence Diagram
sd Interaction
Audio Emotion Recognizer
Signal Preprocessor
Signal Segmenter
Feature Extractor
Classifier
recognizeLowLevelContext(rawSensoryData)
preprocess(rawSensoryData)
:preprocessedSensoryData
segment(preprocessedSensoryData)
:segmentedSensoryData
extractFeatures(segmentedSensoryData)
:extractedFeatures
classify(extractedFeatures)
:activityLabel
Recognize emotion based on audio data
Usecase
ICL2-SUC-08

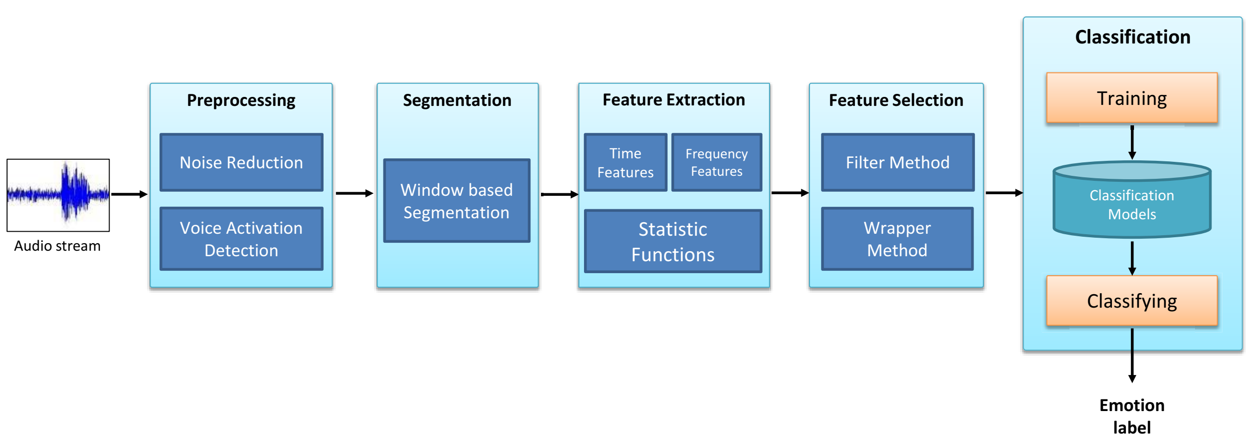

Architecture
Recognize emotion based on audio data
Usecase
ICL2-SUC-08
Audio Emotion Recognizer
Output Adapter
Classification
Feature Extraction
Segmentation
Preprocesing
Input Adapter
경희대학교
KYUNG HEE UNIVERSITY

Workflow

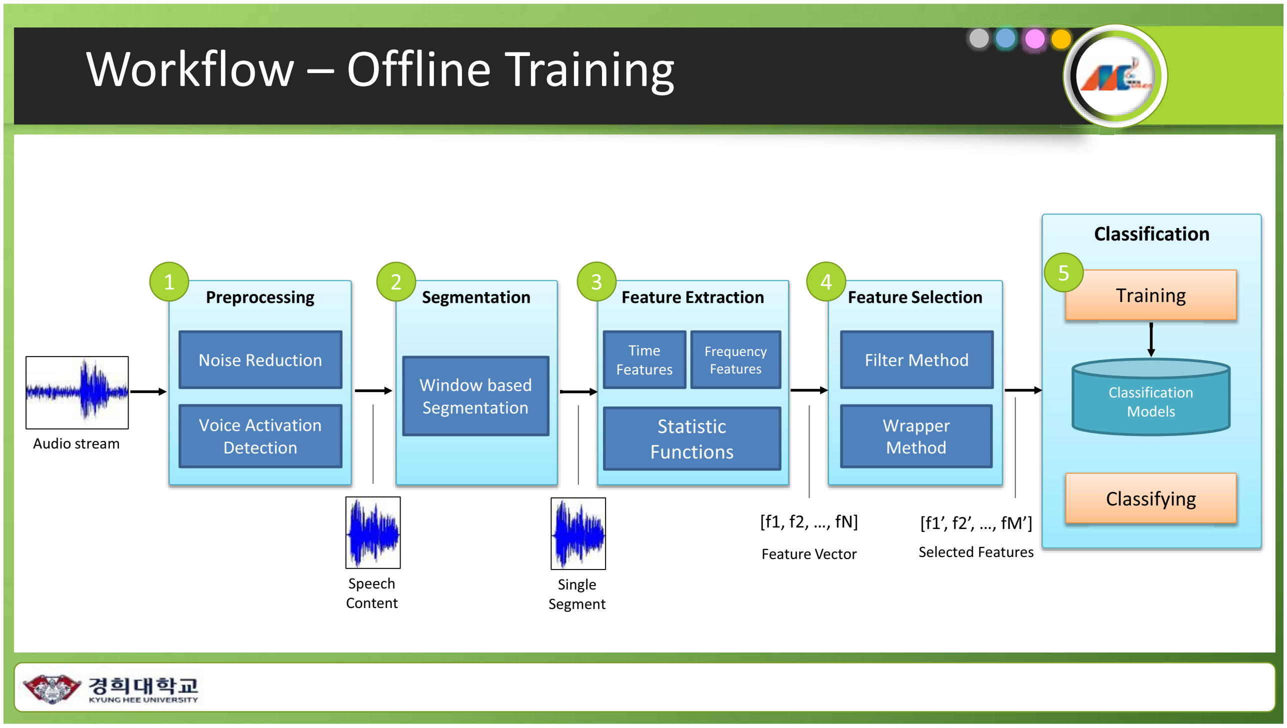


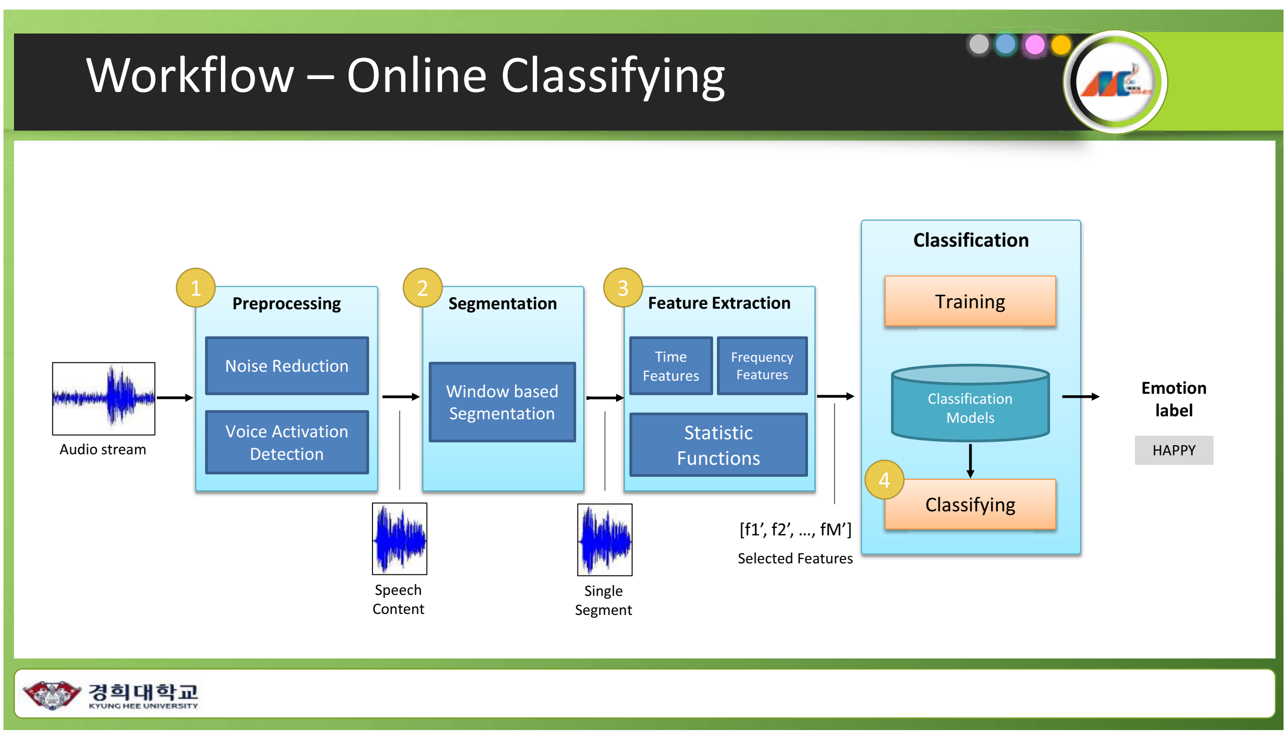
경희대학교 KYUNG HEE UNIVERSITY

Workflow – Emotion Recognizer



경희대학교 KYUNG HEE UNIVERSITY


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


Workflow – Online Classifying
Audio stream
1
Preprocessing
Noise Reduction
Voice Activation Detection
Speech Content
2
Segmentation
Window based Segmentation
Single Segment
3
Feature Extraction
Time Features
Frequency Features
Statistic Functions
[f1', f2', …, fM']
Selected Features
Classification
Training
Classification Models
4
Classifying
Emotion label
HAPPY
경희대학교
KYUNG HEE UNIVERSITY

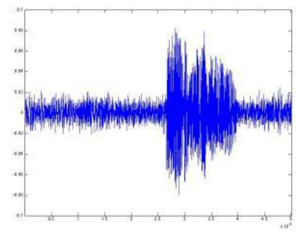
Noise Reduction using Spectral Subtraction

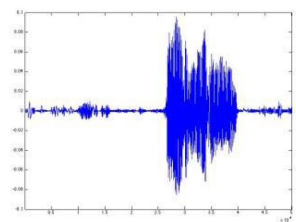
- Reduce the background noise in recording signal using spectral subtraction method

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)





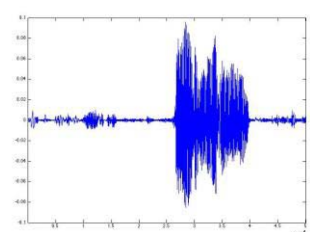
경희대학교 KYUNG HEE UNIVERSITY

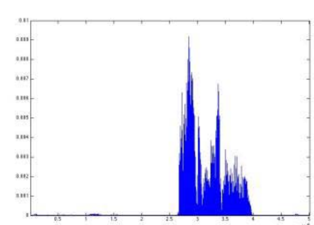
Voice Activation Detection

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

$$E = \sum x^2$$

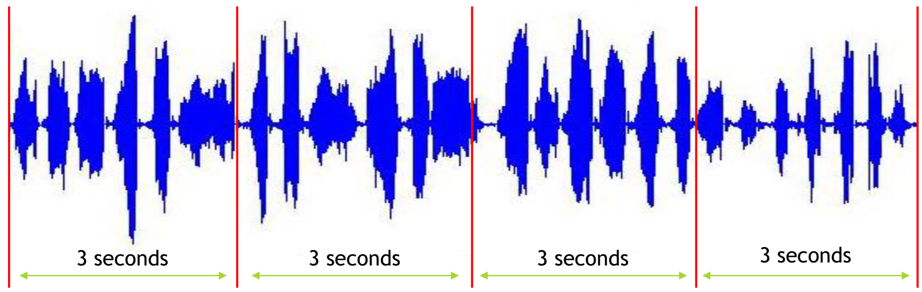
- The average energy of sliding window is compared with a predefined threshold to consider the voice signal





경희대학교 KYUNG HEE UNIVERSITY

Segmentation

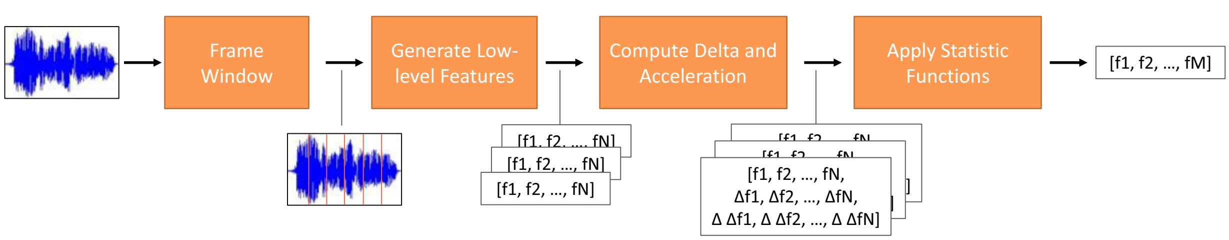


- Sliding window with 3-second length

경희대학교 KYUNG HEE UNIVERSITY

Feature Extraction

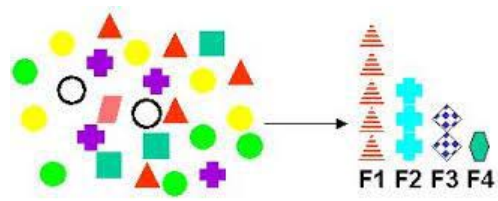
- Generate low-level descriptors for each frame: pitch, root mean square energy, zero crossing rate, harmonic to noise ratio, mel-frequency cepstral coefficients
- Compute delta and acceleration coefficients of each descriptor
- Apply statistic functions: mean, standard deviation, kurtosis, skewness,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



경희대학교 KYUNG HEE UNIVERSITY

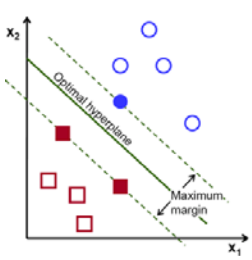
Feature Selection

- 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

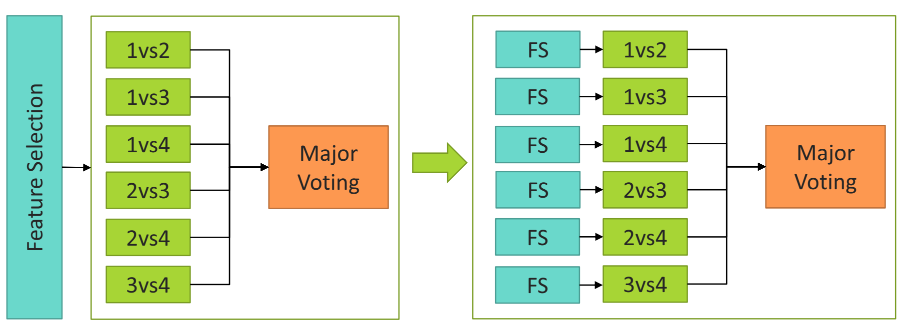


Classification

- Support Vector Machines



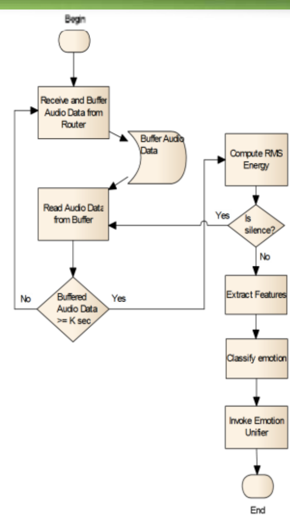
Find the optimal separating hyperplane that has the largest margin.



Original SVM for multiclass

Modified SVM for multiclass

Flowchart



Tools and Technologies

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

경희대학교 KYUNG HEE UNIVERSITY

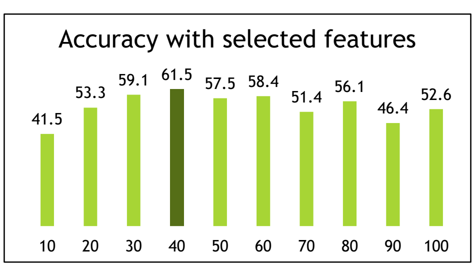
Dataset Collection

- 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

- 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**

경희대학교 KYUNG HEE UNIVERSITY

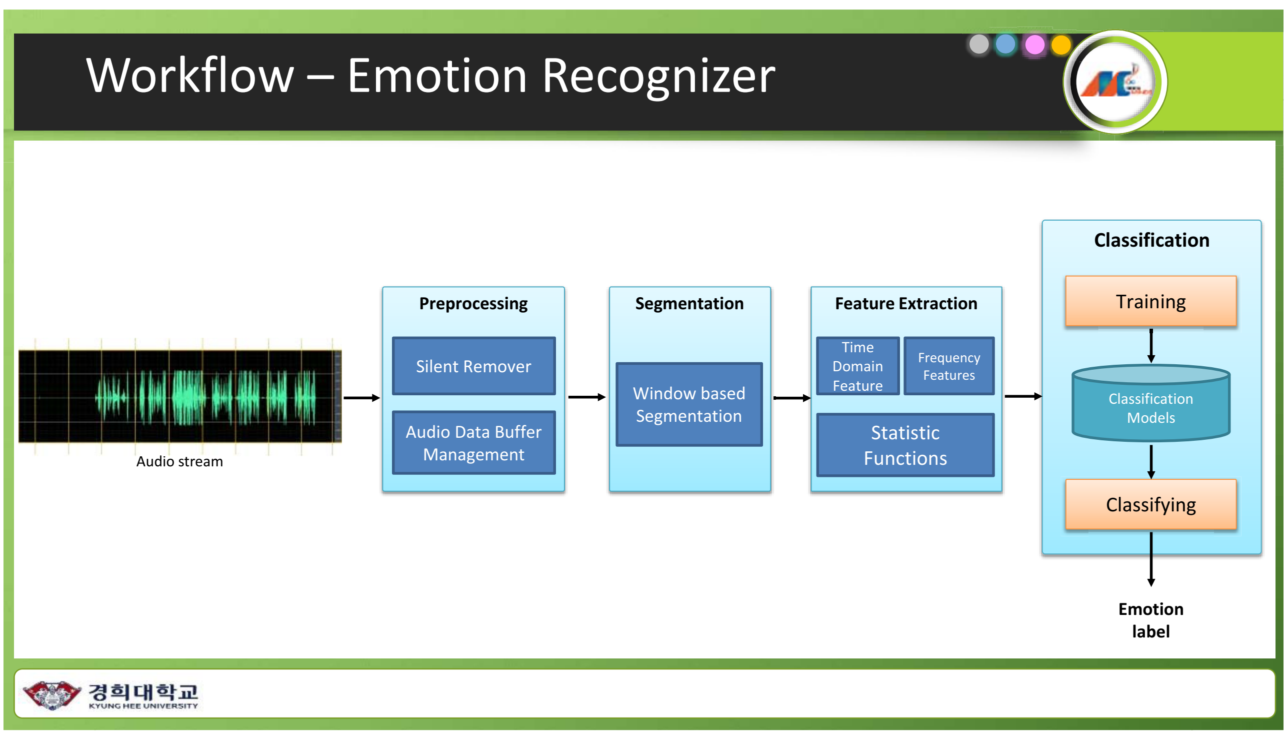
Implementation Plan

- Collection of multimodal real-world dataset (audio, video, GPS, inertial sensors) using smartphone with predefined scenarios (September 11th)
- Evaluation of the proposed methodology on the collected dataset (September 12th)
- Implementation and Evaluation for ICLv2 (October)
  - Designing class, sequence diagrams
  - Writing source code
  - Offline evaluating with collected data
  - Online evaluating with real-time data

Contributions

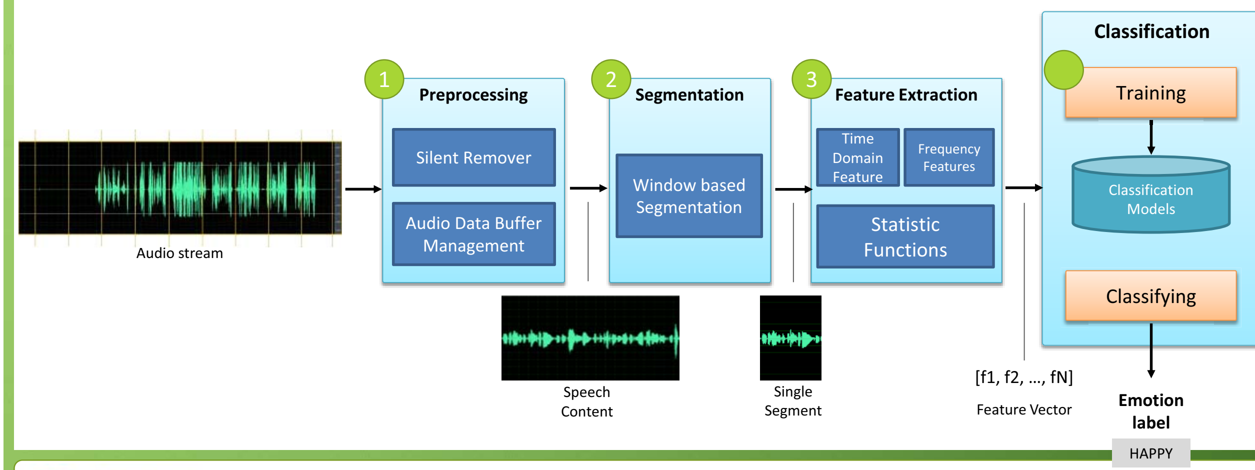
- Proposing a methodology for speech based emotion recognition
- Creation of an emotion set based on requirements of services to be delivered
- Collection of a real-world emotional speech dataset
- Offline evaluation and validation of various emotion recognition models
- Implementation for ICL version 2
- Online validation

Information Curation Layer (ICL 2.0)
Jaehun Bang
MINING MINDS
경희대학교 KYUNG HEE UNIVERSITY
KITECH
POSTECH
UTAS
녹십자헬스케어
TAPACROSS
U2 system
User & Usability system



Workflow – Emotion Recognizer
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

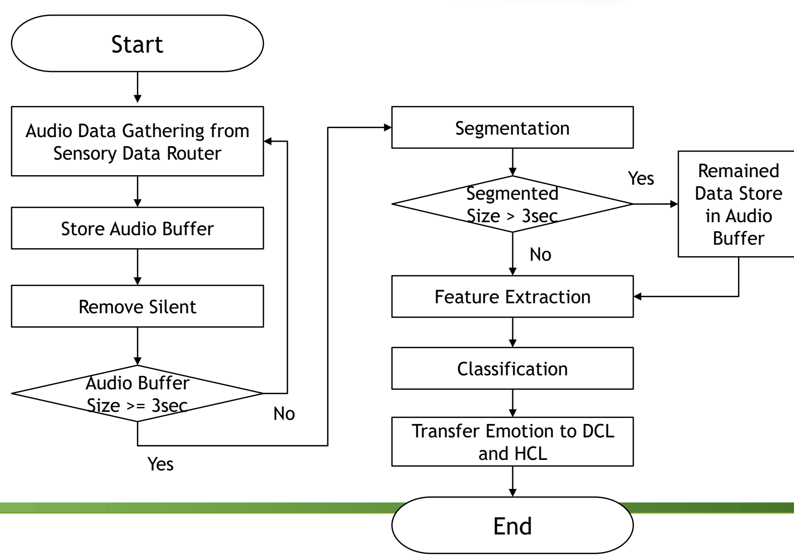
Workflow – Online Classifying



경희대학교
KYUNG HEE UNIVERSITY

Flow Chart

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

- 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

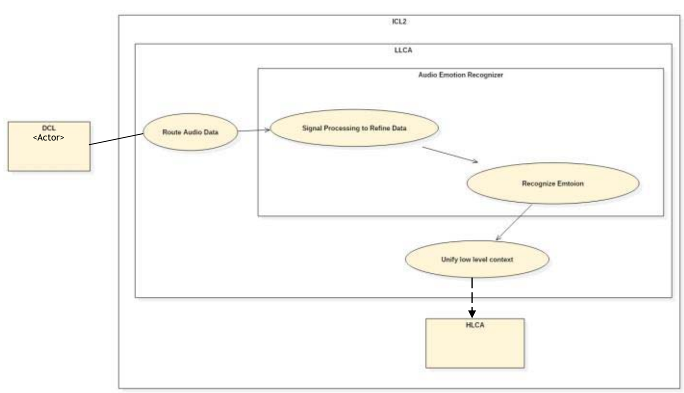
4 emotions – 70.0935%

	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

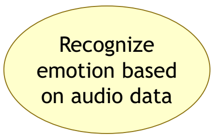
Recognize emotion based on audio data

Usecase
ICL2-SUC-08

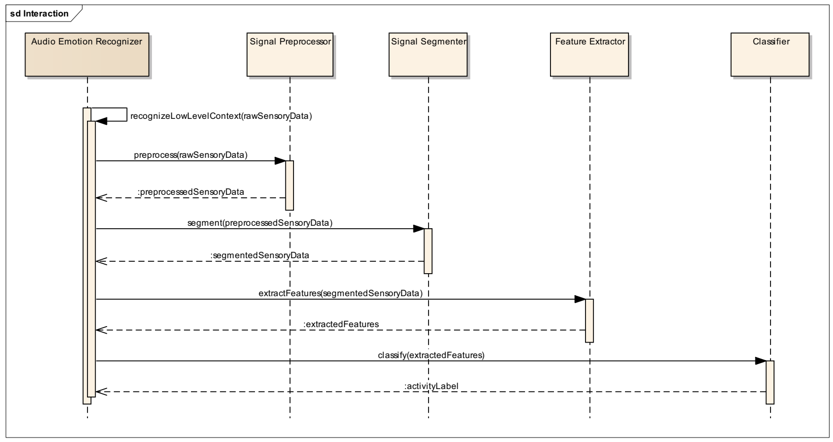


ICL System Specification

Sequence Diagram

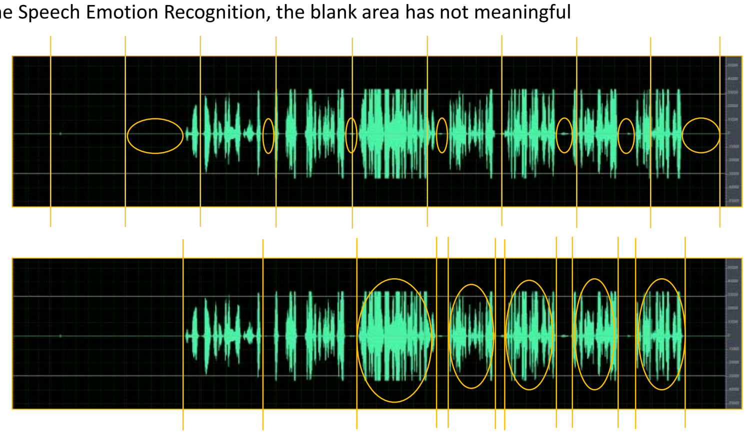


Usecase
ICL2-SUC-08



Future works

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



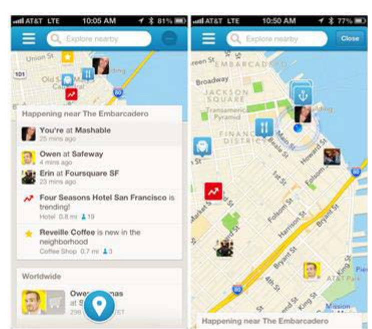
Information Curation Layer (ICL 2.0)

San Kim

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

Introduction

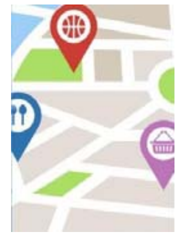
- The identification of people location is of much interest to support diverse type of services
- Location-based services have been widely explored during the last years, mainly exploited in social networks such as Foursquare, TripAdvisor or Twitter among many others
- Location detection techniques can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



경희대학교 KYUNG HEE UNIVERSITY

Problem Statement

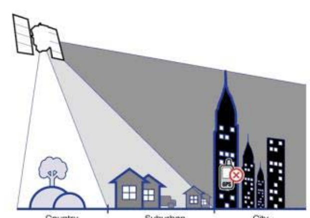
- Motivation
  - Users location is a primary source of information to infer context
  - Tracking of people locations can also serve to identify behaviors and routines
- Goal
  - Define mechanisms to seamlessly identify and register the user location by using technologies readily available to every user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



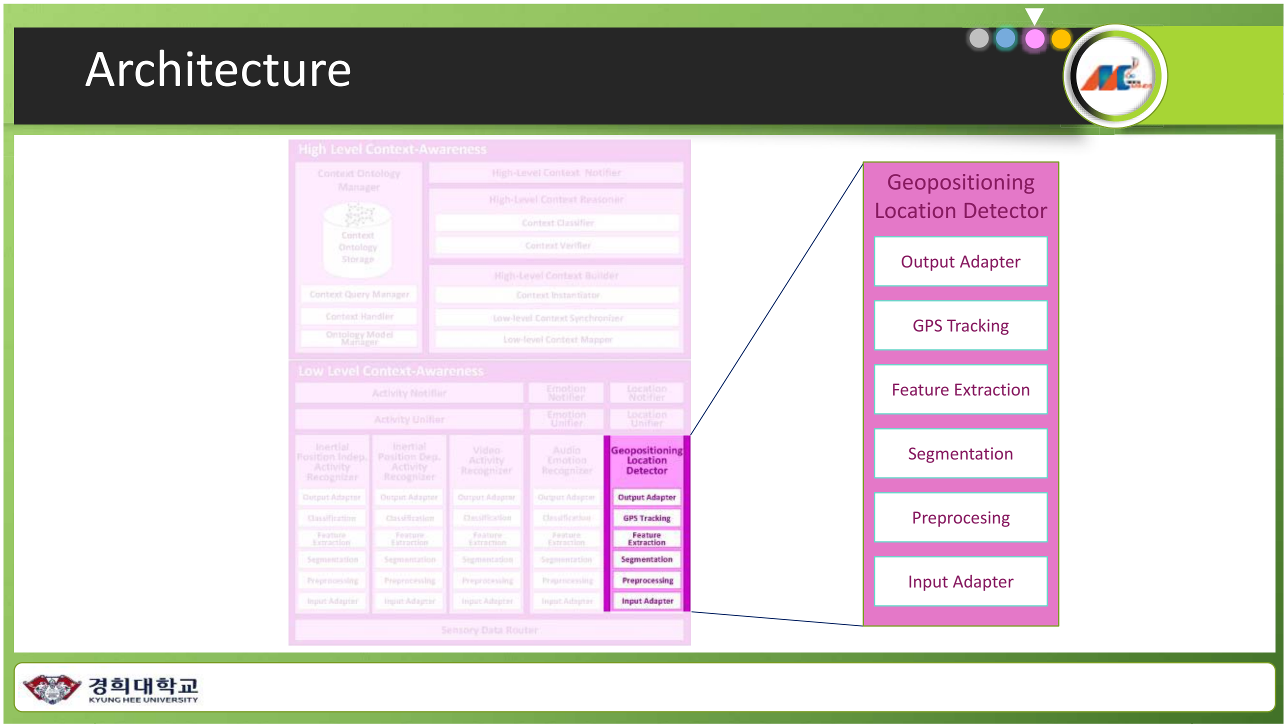
경희대학교
KYUNG HEE UNIVERSITY

Challenges

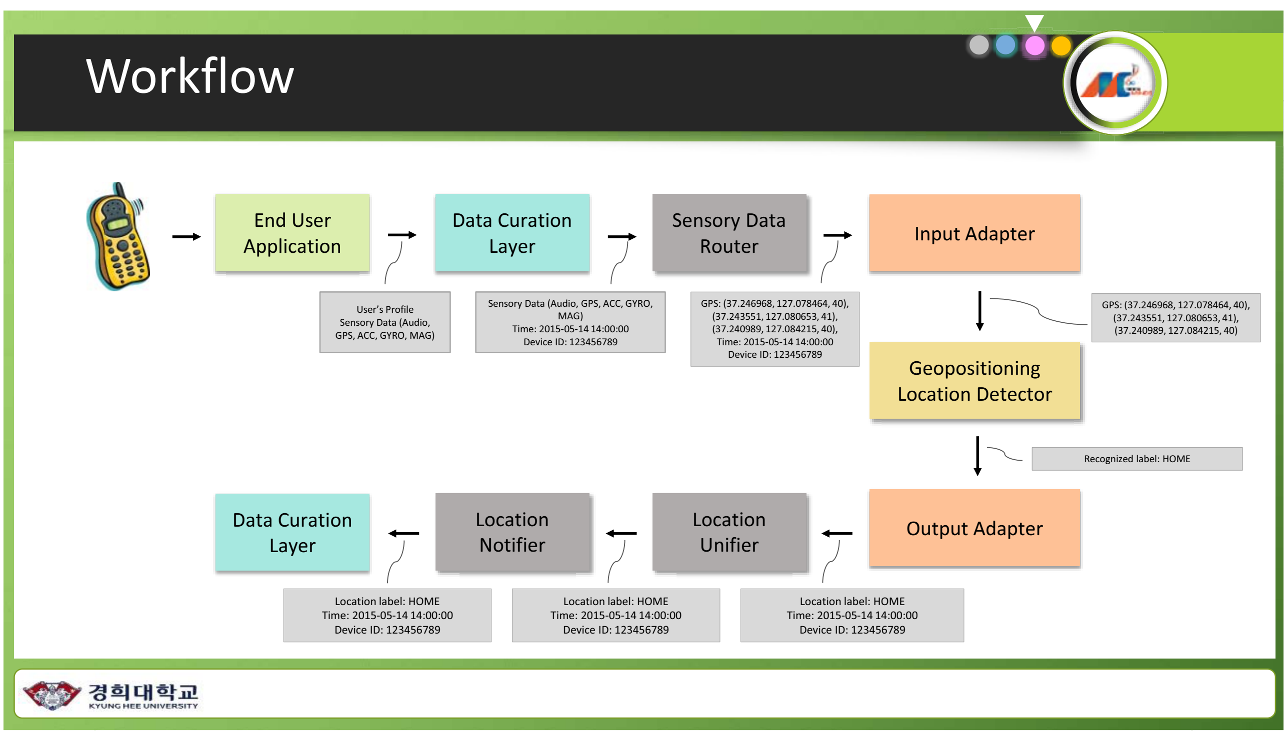
- Geolocation mechanisms are subject to GPS signal availability; thus, no detection is possible in places without coverage (e.g., indoor spaces)
- There exist diverse sort of map APIs; however, they present limitations in their use for some particular countries (e.g., Google Maps in Korea)
- A very location 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



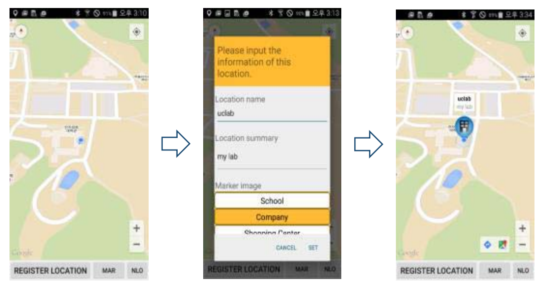
Architecture
High Level Context-Awareness
Low Level Context-Awareness
Geopositioning Location Detector
Output Adapter
GPS Tracking
Feature Extraction
Segmentation
Preprocesing
Input Adapter
경희대학교
KYUNG HEE UNIVERSITY



Workflow
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
GPS: (37.246968, 127.078464, 40), (37.243551, 127.080653, 41), (37.240989, 127.084215, 40),
Time: 2015-05-14 14:00:00
Device ID: 123456789
GPS: (37.246968, 127.078464, 40), (37.243551, 127.080653, 41), (37.240989, 127.084215, 40)
Geopositioning Location Detector
Recognized label: HOME
Data Curation Layer
Location Notifier
Location Unifier
Output Adapter
Location label: HOME
Time: 2015-05-14 14:00:00
Device ID: 123456789
Location label: HOME
Time: 2015-05-14 14:00:00
Device ID: 123456789
Location label: HOME
Time: 2015-05-14 14:00:00
Device ID: 123456789
경희대학교
KYUNG HEE UNIVERSITY

Location Detection – Progress

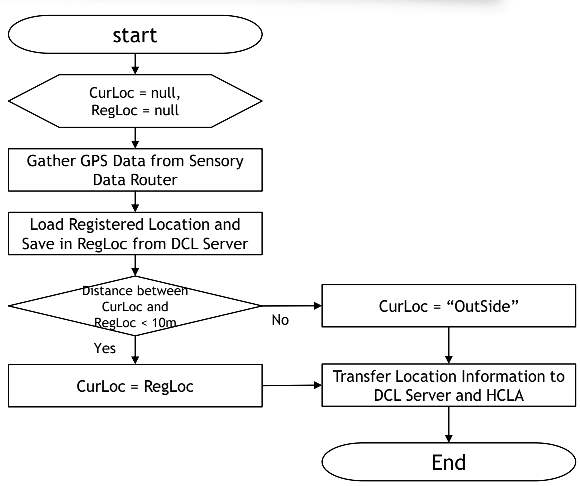
- Implementation of user-centered location registation module
- Implementation of classfying general POI (e.g., “City Hall”), user-centered POI (e.g., “home”) and Frequency POI (e.g., “Restuarants”)
- Visualization of registered location.



경희대학교
KYUNG HEE UNIVERSITY

Designed Location Detector Flowchart

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



Location Detection – Plan

- Usage of Google Maps to overcome some of the limitations observed for Google Maps (e.g., no pedestrian tracking)
- Investigation on techniques for alternative when GPS provider is not work
- Implementation for ICL version 2
- Online validation

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
- Both user-centric and generic maps are considered to personalize the information acquired from the user location

경희대학교
KYUNG HEE UNIVERSITY

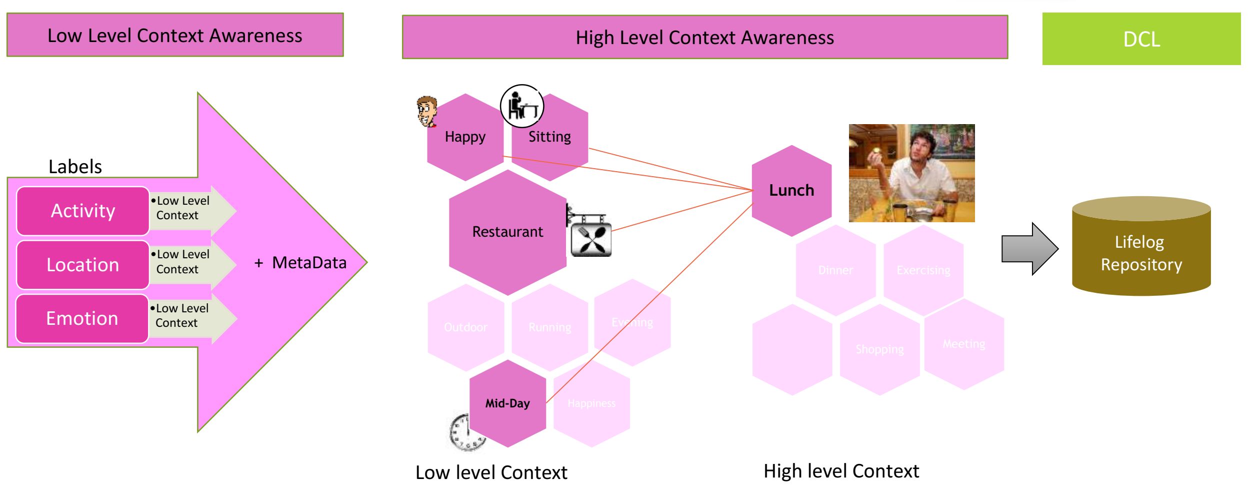
Information Curation Layer (ICL 2.0)

Muhammad Asif Razzaq
Claudia Villalonga

Introduction

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

Introduction – HLCA Example



경희대학교 KYUNG HEE UNIVERSITY

Motivation

- 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

- **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

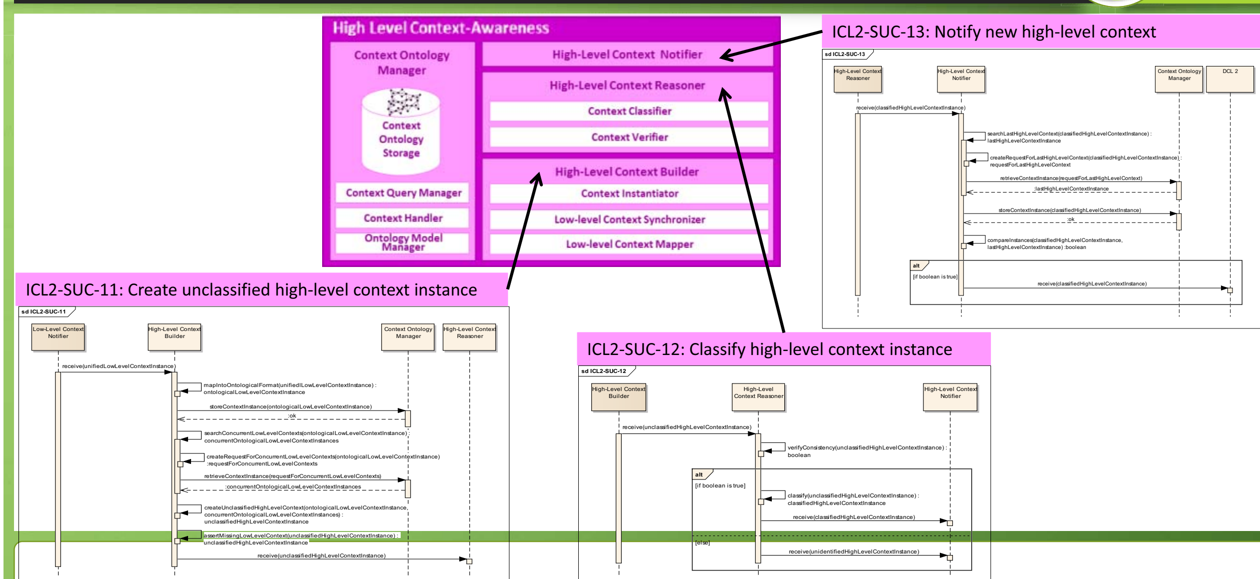
Related Work

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

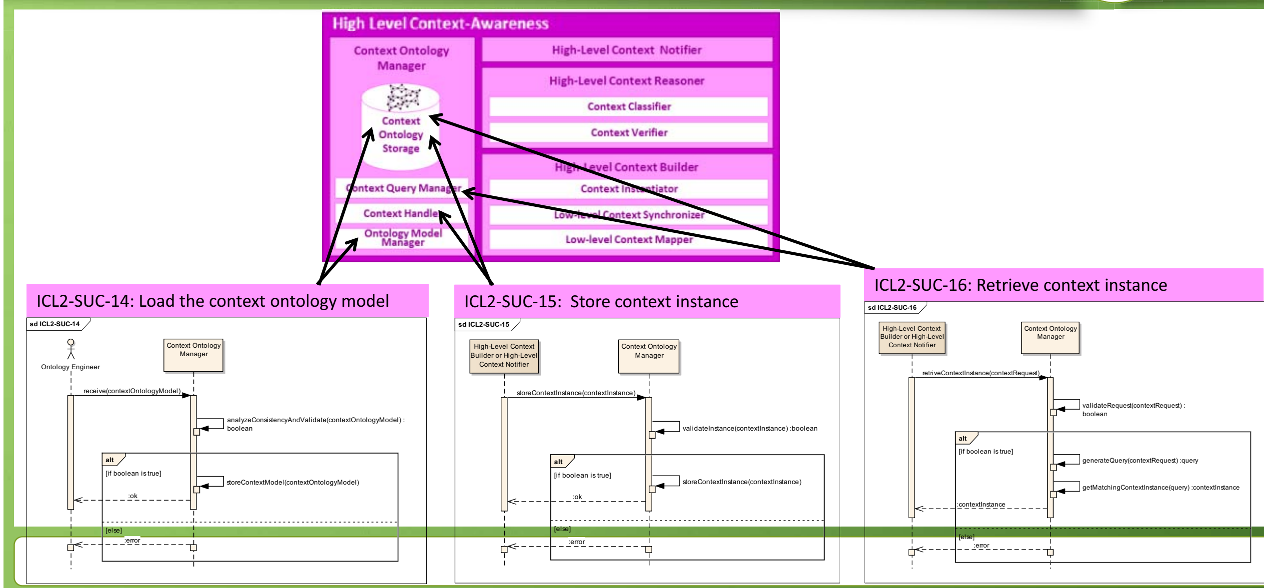
Limitations of existing Work

- 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

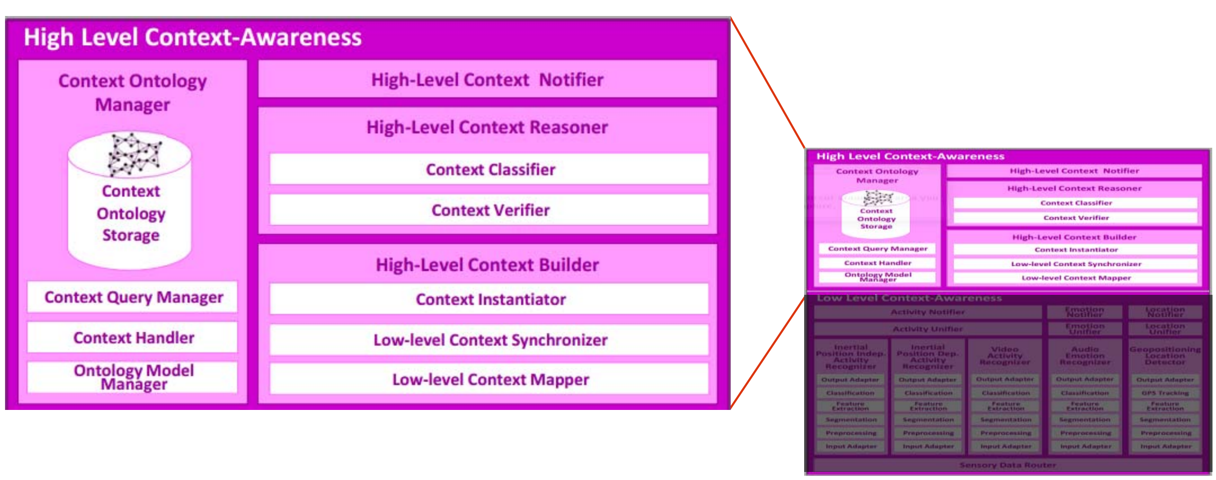
HLCA Use Cases (1)



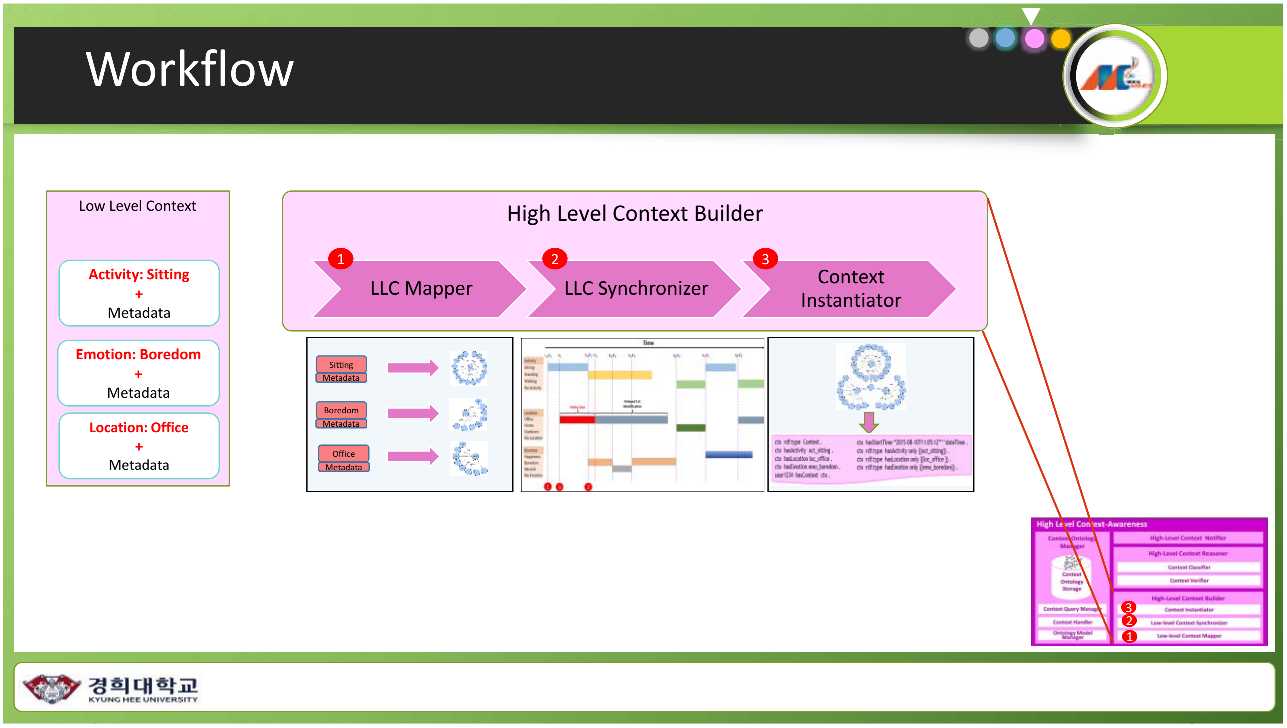
HLCA Use Cases (2)

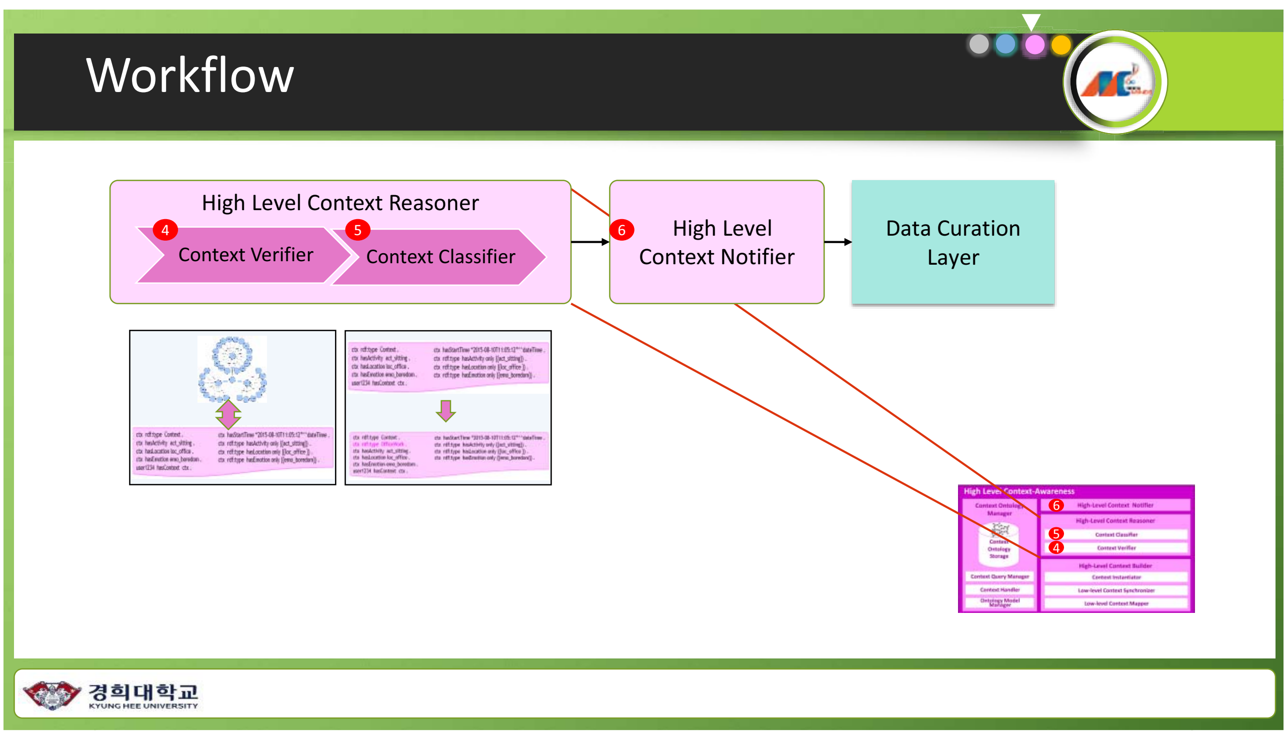


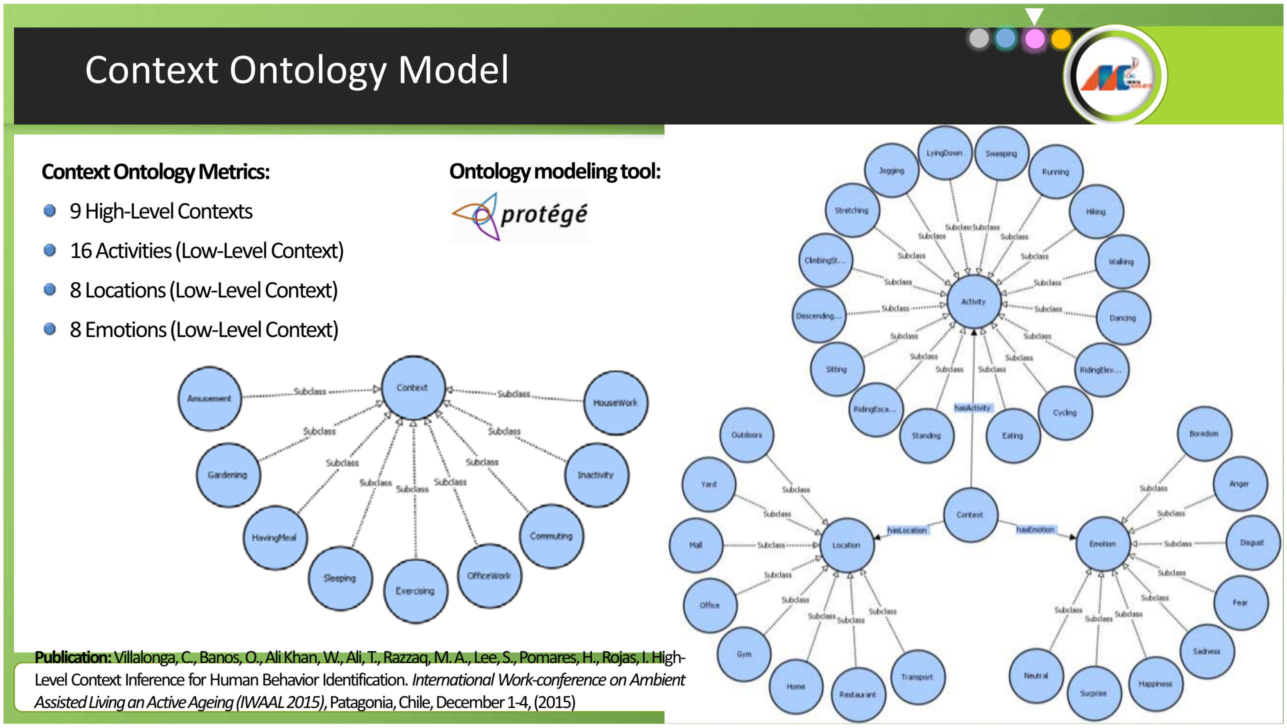
HLCA Architecture



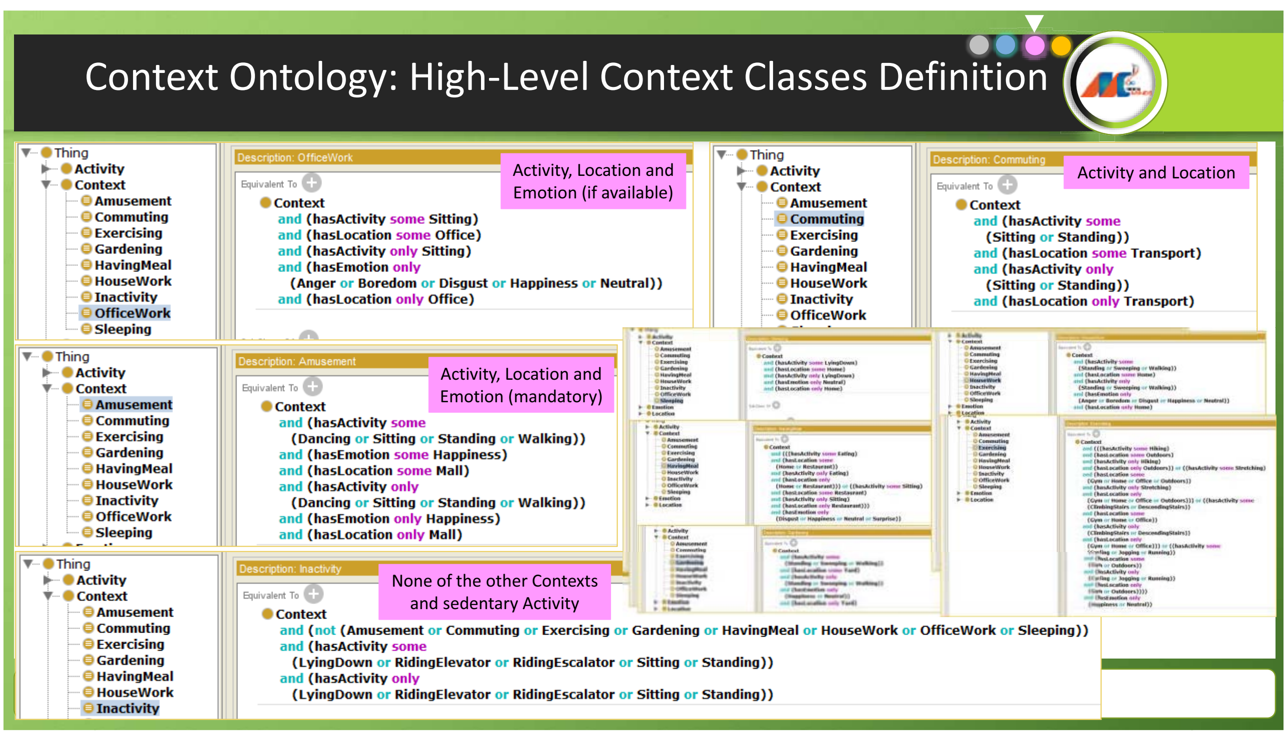
경희대학교
KYUNG HEE UNIVERSITY


Workflow
Low Level Context
Activity: Sitting
+
Metadata
Emotion: Boredom
+
Metadata
Location: Office
+
Metadata
High Level Context Builder
1
LLC Mapper
2
LLC Synchronizer
3
Context Instantiator
Sitting
Metadata
Boredom
Metadata
Office
Metadata
High Level Context-Awareness
3
2
1
경희대학교
KYUNG HEE UNIVERSITY

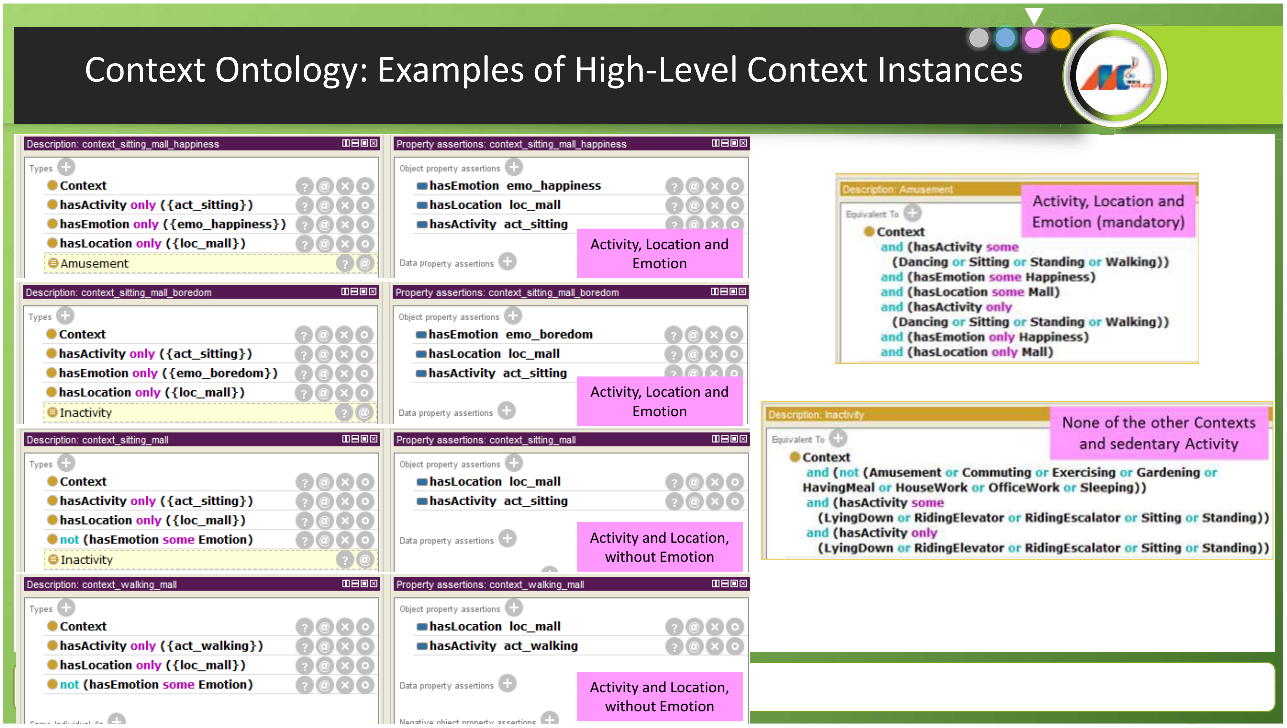

Workflow
High Level Context Reasoner
4
Context Verifier
5
Context Classifier
6
High Level Context Notifier
Data Curation Layer
High Level Context-Awareness
6
5
4
경희대학교
KYUNG HEE UNIVERSITY



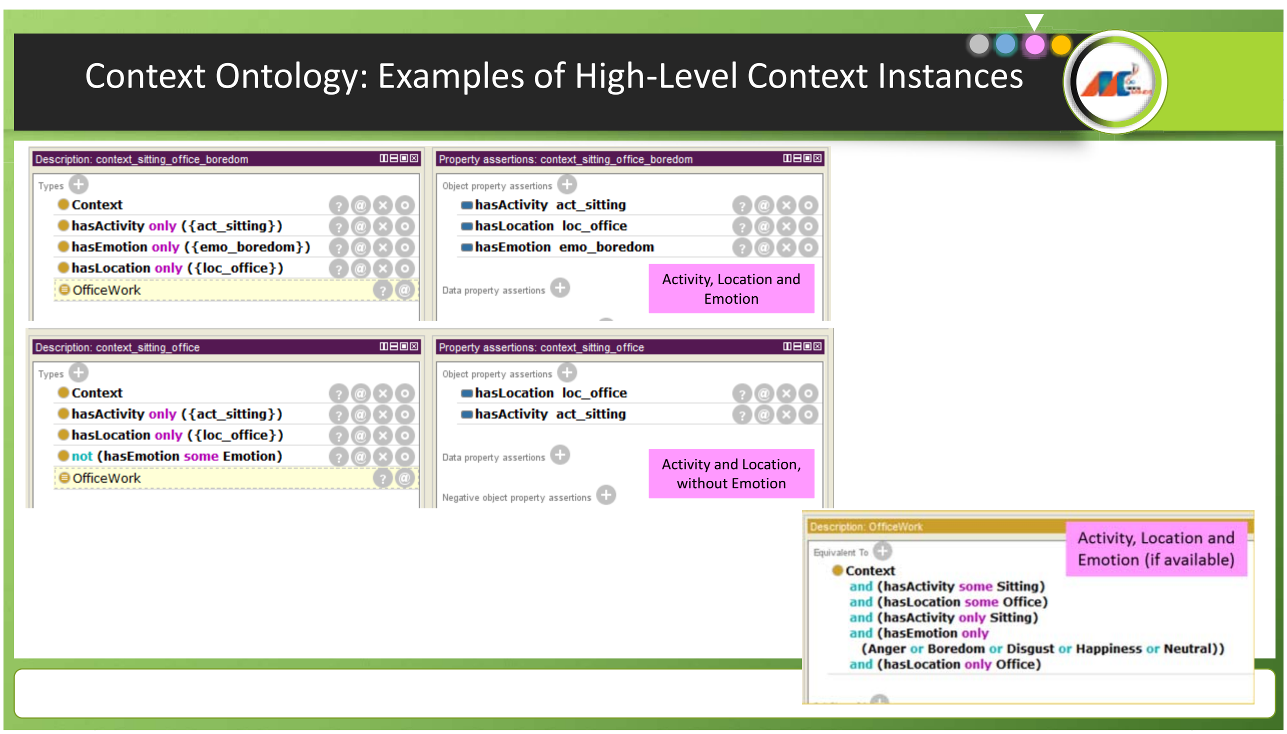
Context Ontology Model
Context Ontology Metrics:
9 High-Level Contexts
16 Activities (Low-Level Context)
8 Locations (Low-Level Context)
8 Emotions (Low-Level Context)
Ontology modeling tool:
protégé
Amusement
Context
HouseWork
Gardening
Inactivity
HavingMeal
Commuting
Sleeping
Exercising
OfficeWork
Subclass
Activity
LyingDown
Sweeping
Jogging
Running
Stretching
Hiking
Walking
Dancing
Descending...
Sitting
RidingElev...
RidingEsca...
Standing
Eating
Cycling
hasActivity
Outdoors
Boredom
Yard
Anger
Mall
Location
hasLocation
hasEmotion
Emotion
Disgust
Office
Fear
Gym
Sadness
Home
Restaurant
Transport
Neutral
Surprise
Happiness
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 Context Classes Definition
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)
Activity, Location and Emotion (if available)
Description: Commuting
Context
and (hasActivity some
(Sitting or Standing))
and (hasLocation some Transport)
and (hasActivity only
(Sitting or Standing))
and (hasLocation only Transport)
Activity and Location
Description: Amusement
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)
Activity, Location and Emotion (mandatory)
Description: Inactivity
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))
None of the other Contexts and sedentary Activity

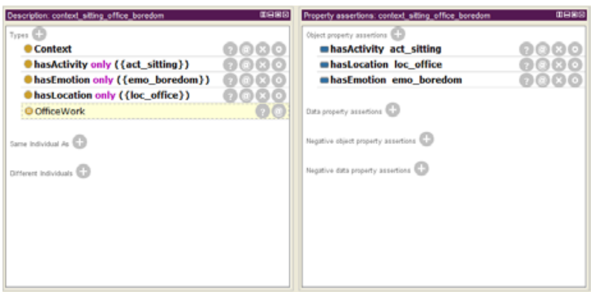


Context Ontology: Examples of High-Level Context Instances
Description: context_sitting_mall_happiness
Types
Context
hasActivity only ({act_sitting})
hasEmotion only ({emo_happiness})
hasLocation only ({loc_mall})
Amusement
Property assertions: context_sitting_mall_happiness
Object property assertions
hasEmotion emo_happiness
hasLocation loc_mall
hasActivity act_sitting
Data property assertions
Activity, Location and Emotion
Description: context_sitting_mall_boredom
Types
Context
hasActivity only ({act_sitting})
hasEmotion only ({emo_boredom})
hasLocation only ({loc_mall})
Inactivity
Property assertions: context_sitting_mall_boredom
Object property assertions
hasEmotion emo_boredom
hasLocation loc_mall
hasActivity act_sitting
Data property assertions
Activity, Location and Emotion
Description: context_sitting_mall
Types
Context
hasActivity only ({act_sitting})
hasLocation only ({loc_mall})
not (hasEmotion some Emotion)
Inactivity
Property assertions: context_sitting_mall
Object property assertions
hasLocation loc_mall
hasActivity act_sitting
Data property assertions
Activity and Location, without Emotion
Description: context_walking_mall
Types
Context
hasActivity only ({act_walking})
hasLocation only ({loc_mall})
not (hasEmotion some Emotion)
Property assertions: context_walking_mall
Object property assertions
hasLocation loc_mall
hasActivity act_walking
Data property assertions
Activity and Location, without Emotion
Description: Amusement
Equivalent To
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)
Activity, Location and Emotion (mandatory)
Description: Inactivity
Equivalent To
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))
None of the other Contexts and sedentary Activity



Context Ontology: Examples of High-Level Context Instances
Description: context_sitting_office_boredom
Types
Context
hasActivity only ({act_sitting})
hasEmotion only ({emo_boredom})
hasLocation only ({loc_office})
OfficeWork
Property assertions: context_sitting_office_boredom
Object property assertions
hasActivity act_sitting
hasLocation loc_office
hasEmotion emo_boredom
Data property assertions
Activity, Location and Emotion
Description: context_sitting_office
Types
Context
hasActivity only ({act_sitting})
hasLocation only ({loc_office})
not (hasEmotion some Emotion)
OfficeWork
Property assertions: context_sitting_office
Object property assertions
hasLocation loc_office
hasActivity act_sitting
Data property assertions
Negative object property assertions
Activity and Location, without Emotion
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)
Activity, Location and Emotion (if available)

Context Ontology: Low-level Context Instances, Unclassified and Classified High-Level Context Instances



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}) .

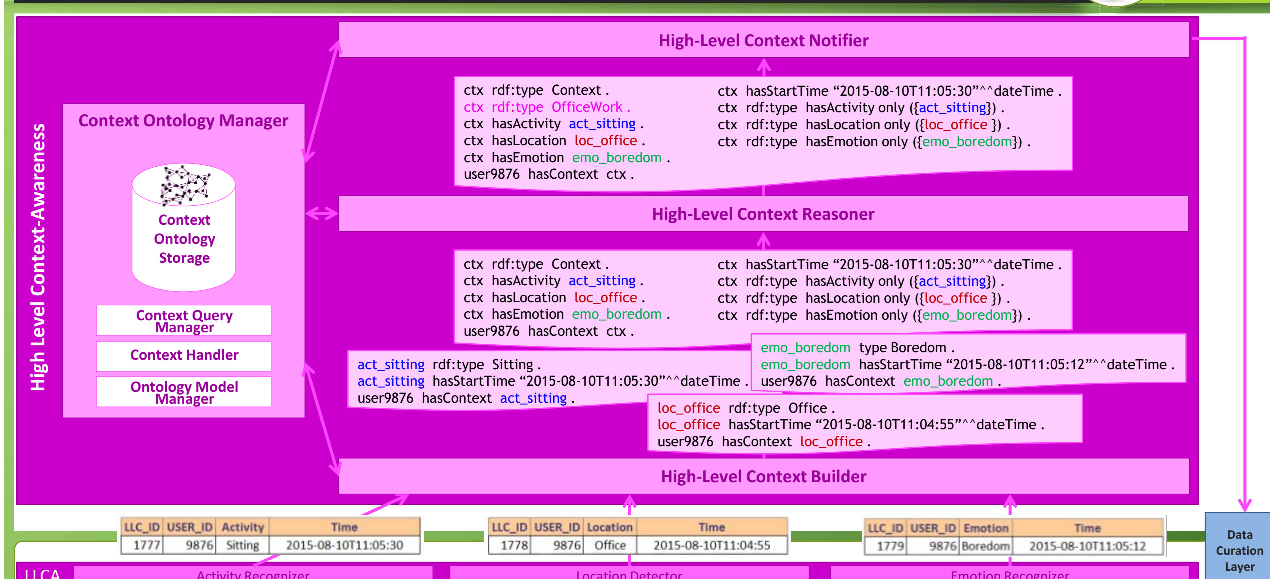
Low-Level Context Instances

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

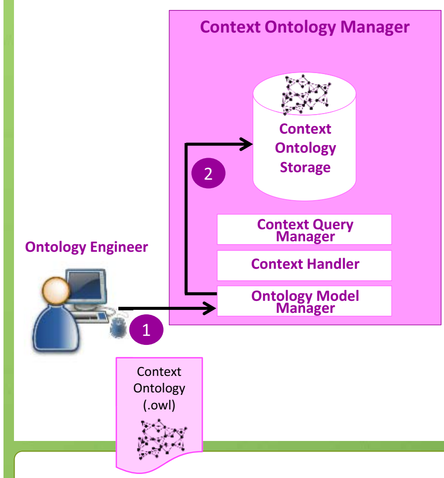
loc_office rdf:type Office .
loc_office hasStartTime "2015-08-10T11:04:55"^^dateTime .
user9876 hasContext loc_office .

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

Context Ontology in HLCA

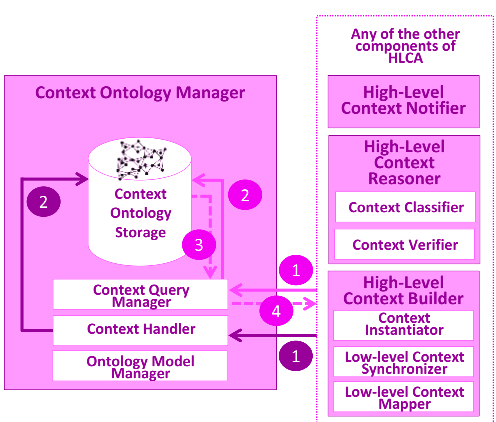


Context Ontology Manager



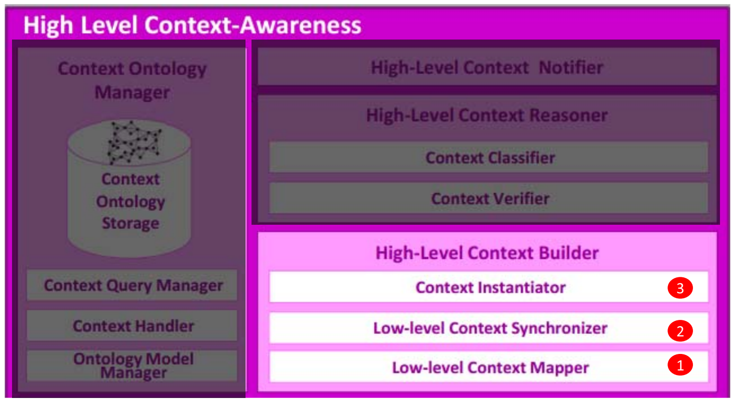
- **Context Ontology Storage**
  - Persist the Context Ontology Model and Context Instances
  - Technology: Triplestore
  - Tool (to be confirmed) : TDB of Jena
- **Ontology Model Manager**
  - Load into MM the Context Ontology Model
  - Workflow:
    - 1 Receive Context Ontology Model (in OWL format) 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



- **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



Low Level Context Mapper

Low Level Context Instances

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

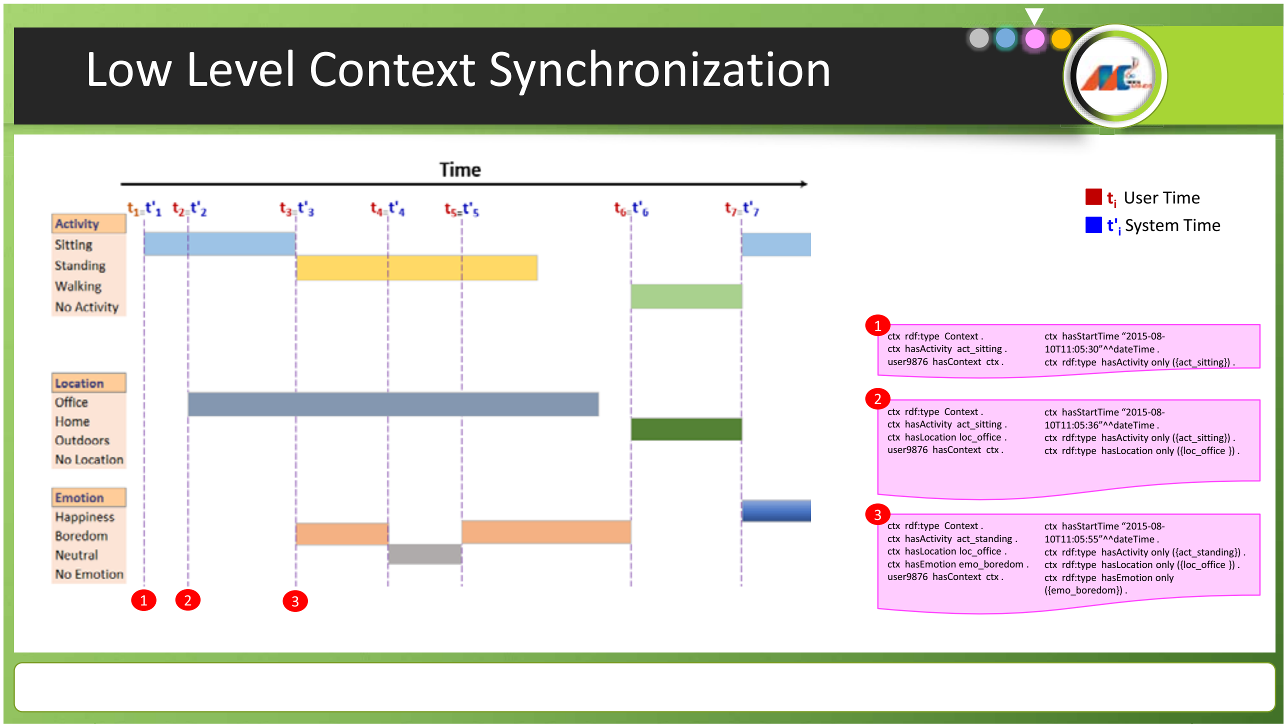
LLC_ID	USER_ID	Location	Time
1778	9876	Office	2015-08-10T11:04:55

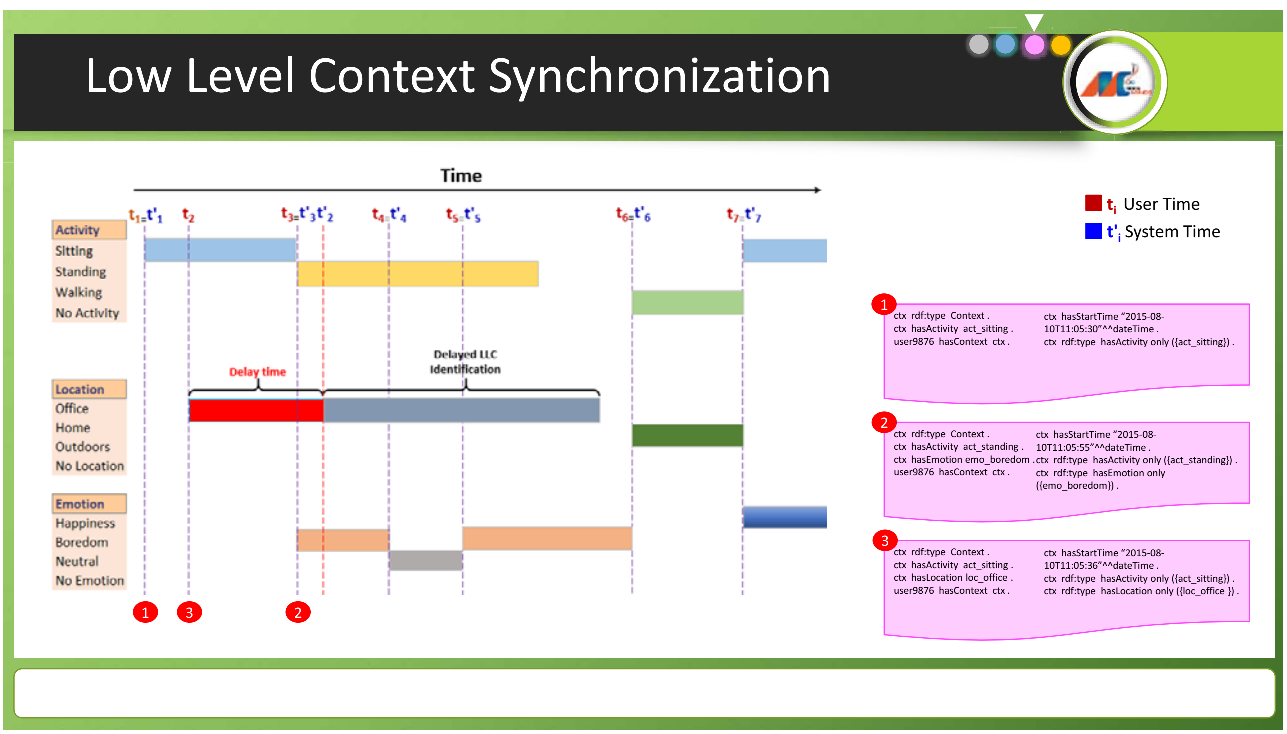
LLC_ID	USER_ID	Emotion	Time
1779	9876	Boredom	2015-08-10T11:05:12

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

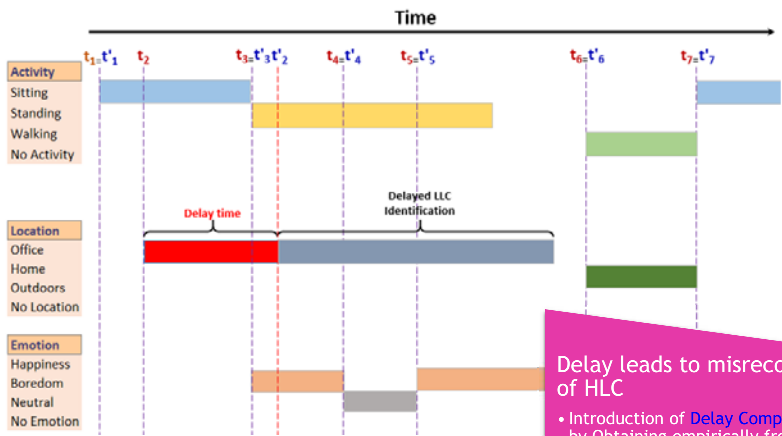
loc_office rdf:type Office .
loc_office hasStartTime “2015-08-10T11:04:55”^^dateTime .
user9876 hasContext loc_office .

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


Low Level Context Synchronization
Time
t1=t'1
t2=t'2
t3=t'3
t4=t'4
t5=t'5
t6=t'6
t7=t'7
Activity
Sitting
Standing
Walking
No Activity
Location
Office
Home
Outdoors
No Location
Emotion
Happiness
Boredom
Neutral
No Emotion
1
2
3
ti User Time
t'i System Time
1
ctx rdf:type Context .
ctx hasActivity act_sitting .
user9876 hasContext ctx .
ctx hasStartTime "2015-08-10T11:05:30"^^dateTime .
ctx rdf:type hasActivity only ({act_sitting}) .
2
ctx rdf:type Context .
ctx hasActivity act_sitting .
ctx hasLocation loc_office .
user9876 hasContext ctx .
ctx hasStartTime "2015-08-10T11:05:36"^^dateTime .
ctx rdf:type hasActivity only ({act_sitting}) .
ctx rdf:type hasLocation only ({loc_office }) .
3
ctx rdf:type Context .
ctx hasActivity act_standing .
ctx hasLocation loc_office .
ctx hasEmotion emo_boredom .
user9876 hasContext ctx .
ctx hasStartTime "2015-08-10T11:05:55"^^dateTime .
ctx rdf:type hasActivity only ({act_standing}) .
ctx rdf:type hasLocation only ({loc_office }) .
ctx rdf:type hasEmotion only ({emo_boredom}) .


Low Level Context Synchronization
Time
t1=t'1
t2
t3=t'3t'2
t4=t'4
t5=t'5
t6=t'6
t7=t'7
Activity
Sitting
Standing
Walking
No Activity
Delay time
Delayed LLC Identification
Location
Office
Home
Outdoors
No Location
Emotion
Happiness
Boredom
Neutral
No Emotion
1
3
2
ti User Time
t'i System Time
1
ctx rdf:type Context .
ctx hasActivity act_sitting .
user9876 hasContext ctx .
ctx hasStartTime "2015-08-10T11:05:30"^^dateTime .
ctx rdf:type hasActivity only ({act_sitting}) .
2
ctx rdf:type Context .
ctx hasActivity act_standing .
ctx hasEmotion emo_boredom .
user9876 hasContext ctx .
ctx hasStartTime "2015-08-10T11:05:55"^^dateTime .
ctx rdf:type hasActivity only ({act_standing}) .
ctx rdf:type hasEmotion only ({emo_boredom}) .
3
ctx rdf:type Context .
ctx hasActivity act_sitting .
ctx hasLocation loc_office .
user9876 hasContext ctx .
ctx hasStartTime "2015-08-10T11:05:36"^^dateTime .
ctx rdf:type hasActivity only ({act_sitting}) .
ctx rdf:type hasLocation only ({loc_office }) .

Low Level Context Synchronization



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

Low Level Context Instances

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

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

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

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

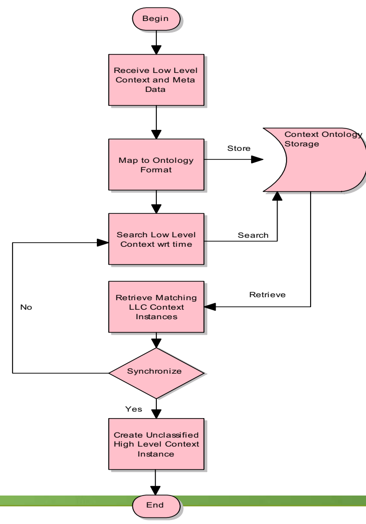
LLC_ID	USER_ID	Emotion	Time
1779	9876	Boredom	2015-08-10T11:05:12

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

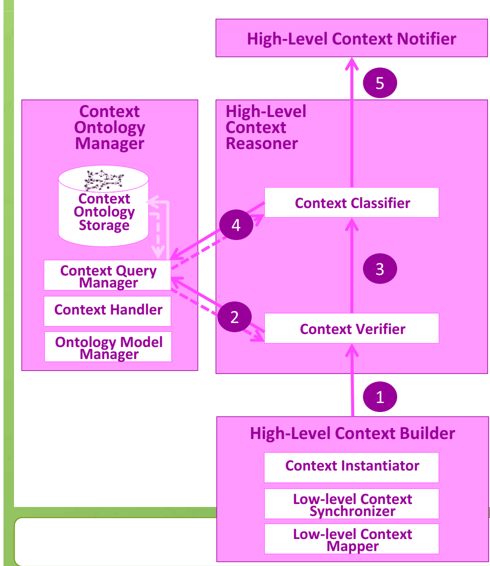
Unclassified High Level Context Instance

```
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



High-level Context Reasoner



- **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

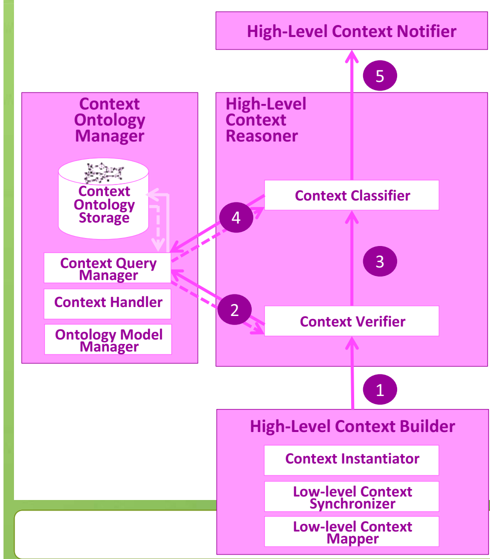
```
ctx rdf:type Context .
ctx hasActivity act_sitting .
ctx hasLocation loc_office .
ctx hasEmotion emo_boredom .
user9876 hasContext ctx .
ctx hasStartTime "2015-08-10T11:05:30"^^dateTime .
ctx rdf:type hasActivity only ({act_sitting}) .
ctx rdf:type hasLocation only ({loc_office }) .
ctx rdf:type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



- **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

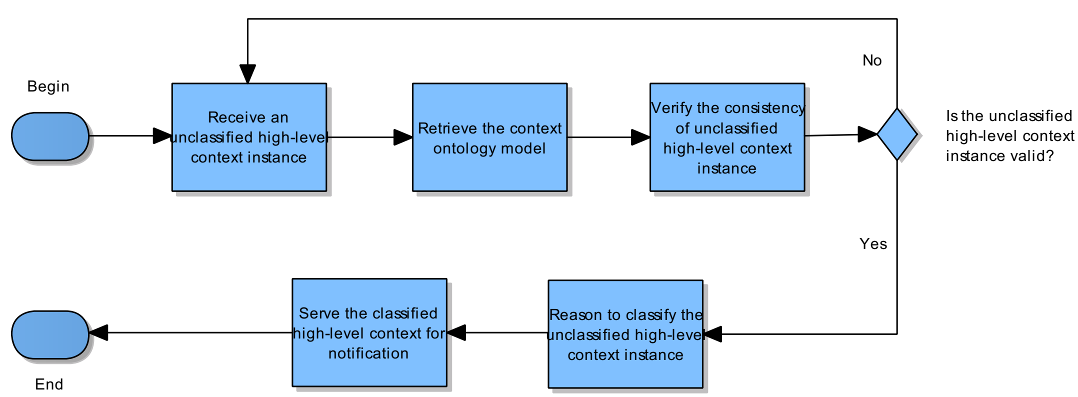
```
ctx rdf:type Context .
ctx hasActivity act_sitting .
ctx hasLocation loc_office .
ctx hasEmotion emo_boredom .
user9876 hasContext ctx .
ctx hasStartTime "2015-08-10T11:05:30"^^dateTime .
ctx rdf:type hasActivity only ({act_sitting}) .
ctx rdf:type hasLocation only ({loc_office }) .
ctx rdf:type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

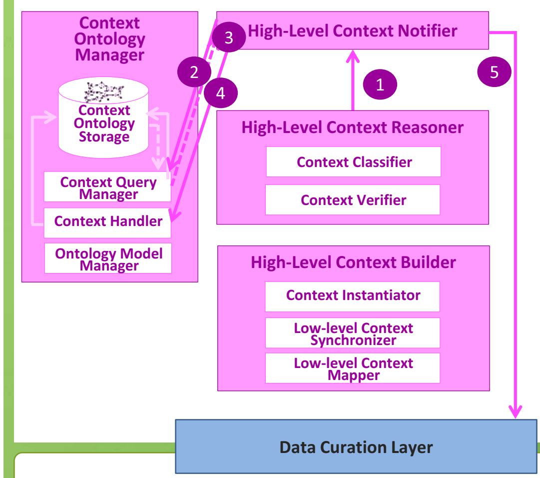
```
ctx rdf:type Context .
ctx rdf:type OfficeWork .
ctx hasActivity act_sitting .
ctx hasLocation loc_office .
ctx hasEmotion emo_boredom .
user9876 hasContext ctx .
ctx hasStartTime "2015-08-10T11:05:30"^^dateTime .
ctx rdf:type hasActivity only ({act_sitting}) .
ctx rdf:type hasLocation only ({loc_office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

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

High-level Context Reasoner Flowchart



High-Level Context Notifier



- **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}) .

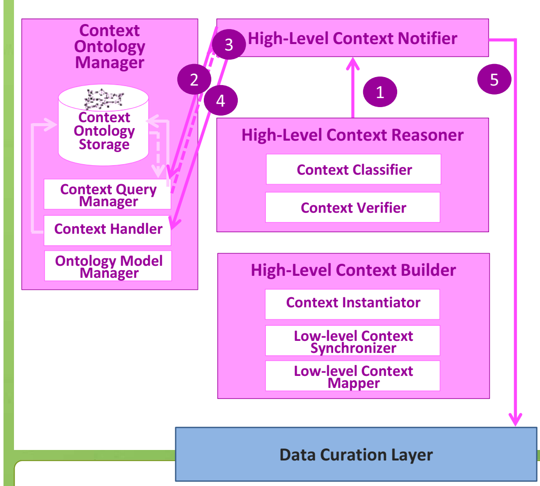
    2. Retrieve previous high-level context instance

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



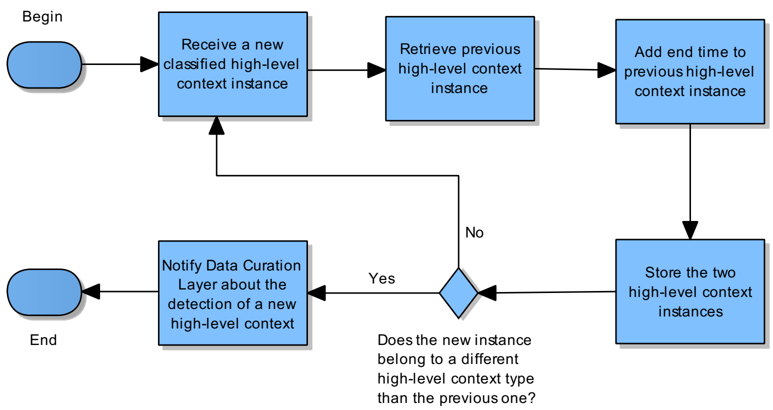
- **Context Notifier**
  - Communicate the newly recognized high-level context to Data Curation Layer for storage into the LifeLog
  - Workflow:
    4. Store the two high-level context instances

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



Technologies for Data Representation

- 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

W3C OWL2 W3C SPARQL

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

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

Jena

Tools for Ontological Reasoning

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

- 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

- 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

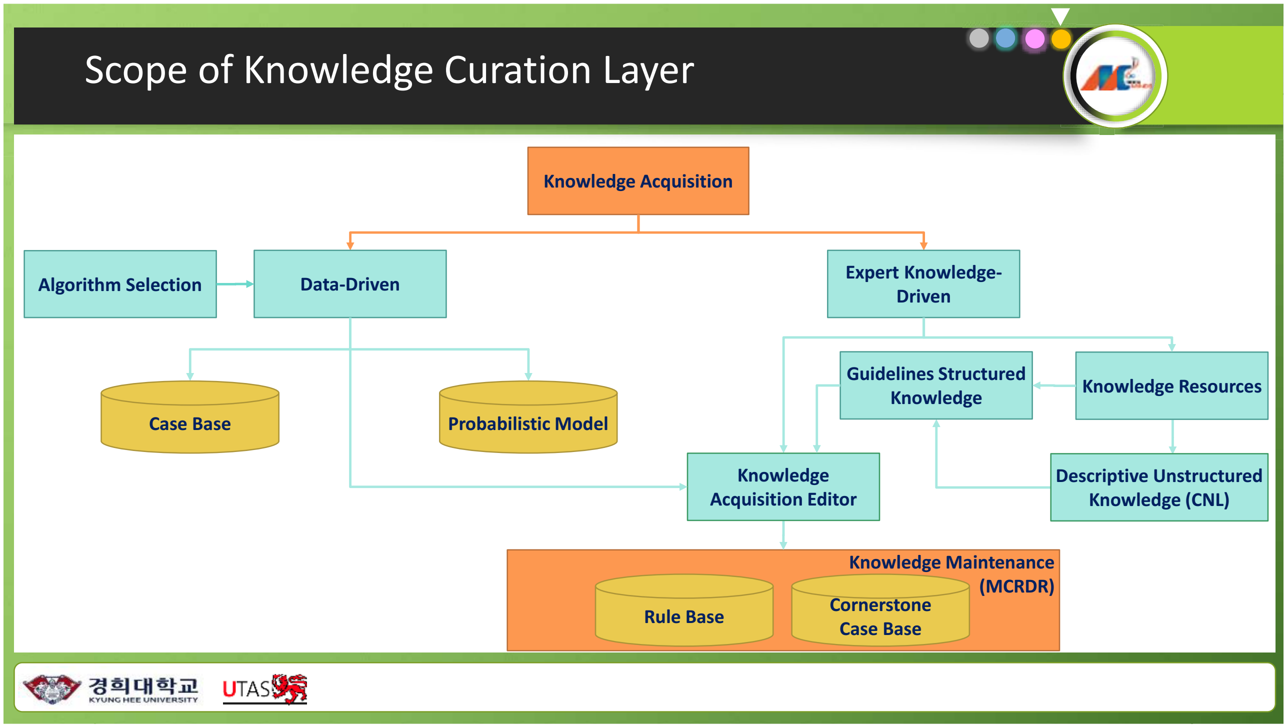
Knowledge Curation Layer (KCL 2.0)

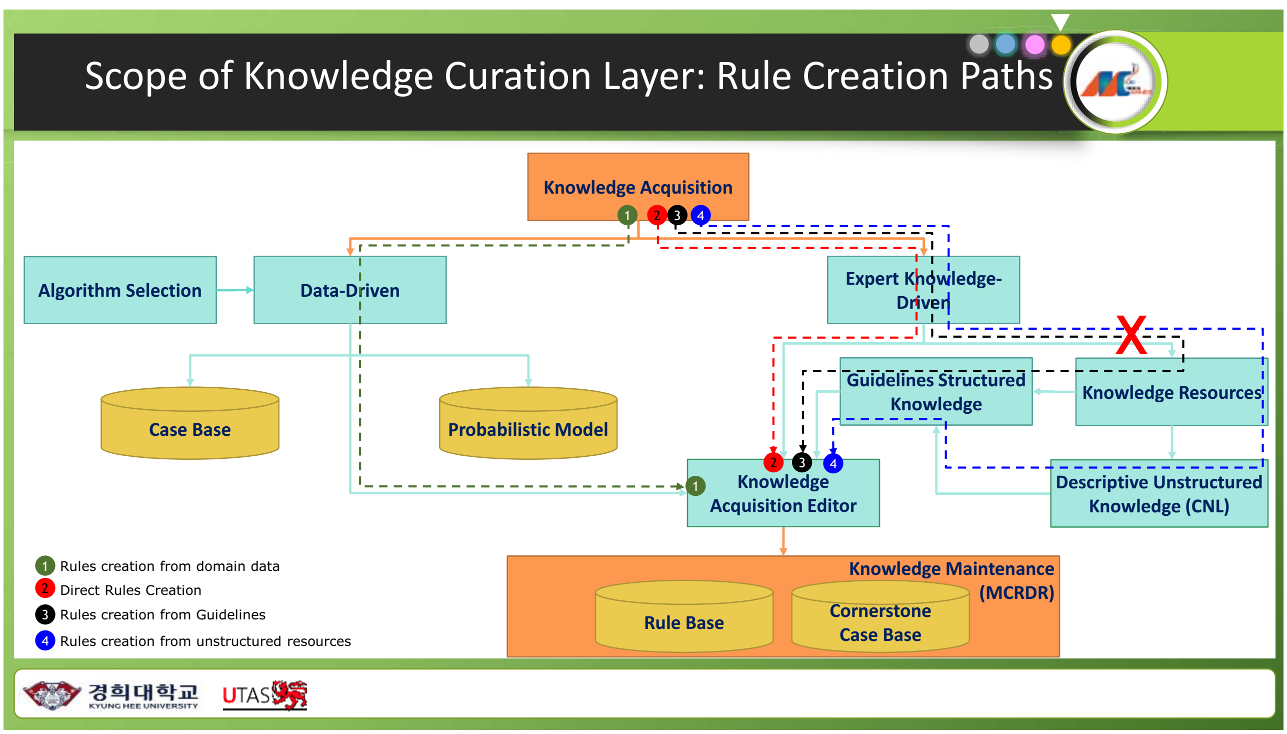
Principal Investigator:
Professor Sungyoung Lee

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

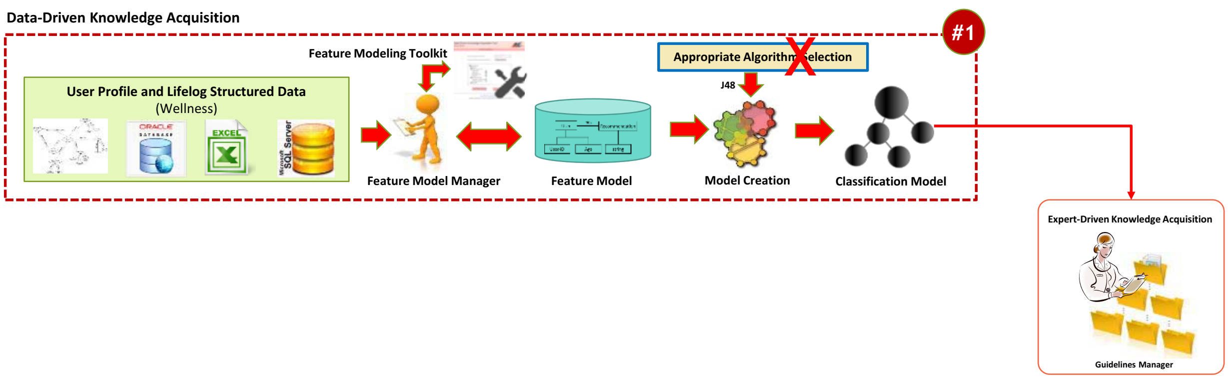
Knowledge Curation Layer (KCL 2.0)

경희대학교 KYUNG HEE UNIVERSITY UTAS

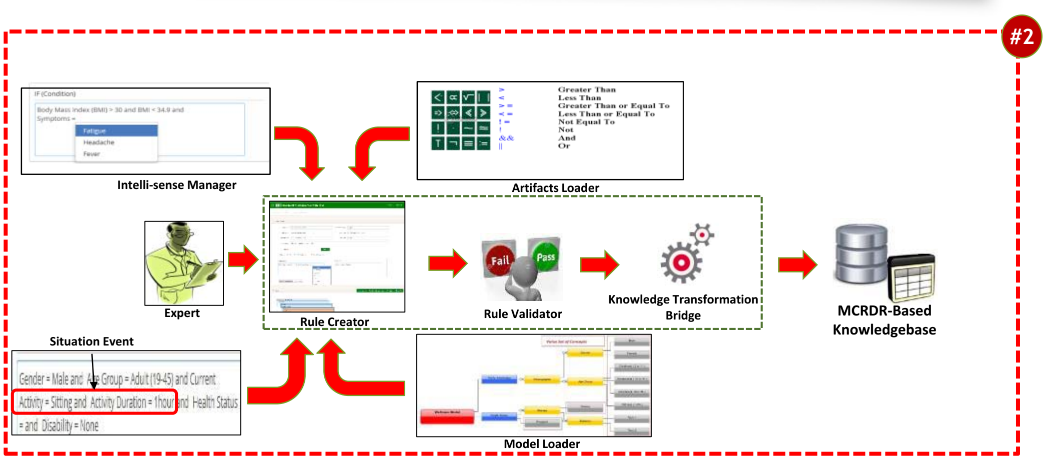

Scope of Knowledge Curation Layer
Knowledge Acquisition
Algorithm Selection
Data-Driven
Expert Knowledge-Driven
Case Base
Probabilistic Model
Guidelines Structured Knowledge
Knowledge Resources
Knowledge Acquisition Editor
Descriptive Unstructured Knowledge (CNL)
Knowledge Maintenance (MCRDR)
Rule Base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

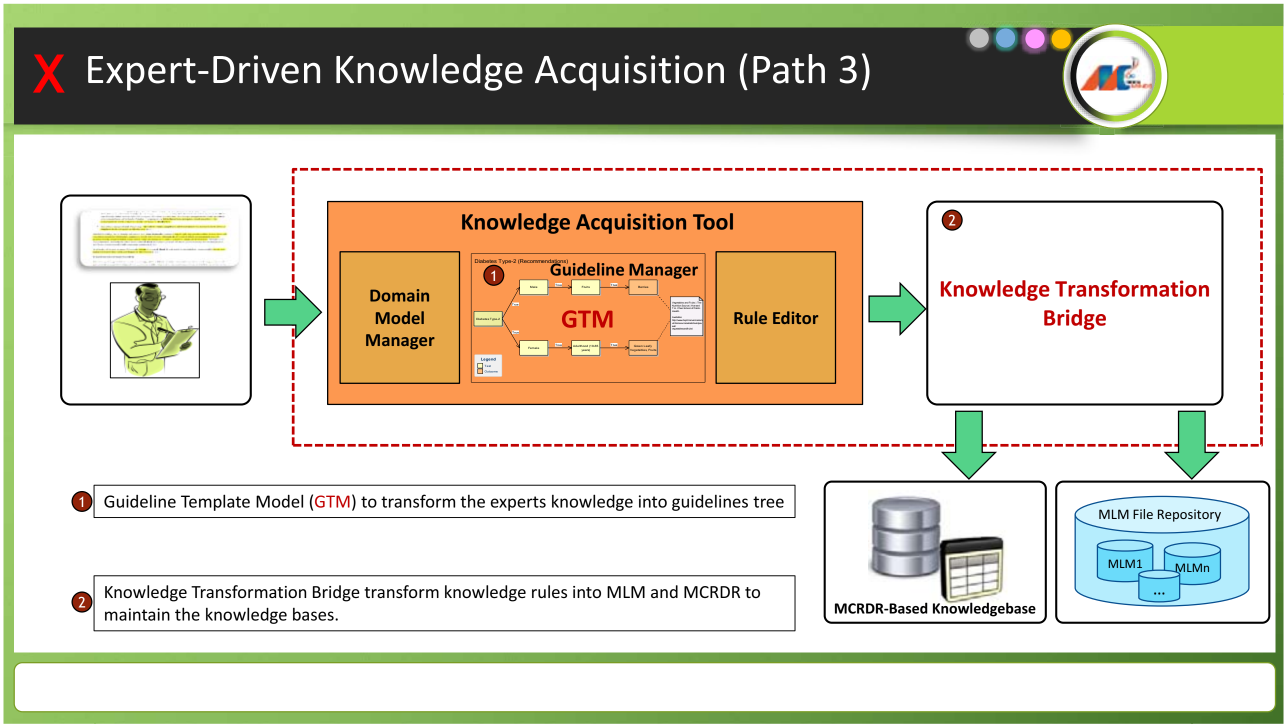

Scope of Knowledge Curation Layer: Rule Creation Paths
Knowledge Acquisition
1
2
3
4
Algorithm Selection
Data-Driven
Expert Knowledge-Driven
X
Case Base
Probabilistic Model
Guidelines Structured Knowledge
Knowledge Resources
Knowledge Acquisition Editor
Descriptive Unstructured Knowledge (CNL)
1 Rules 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

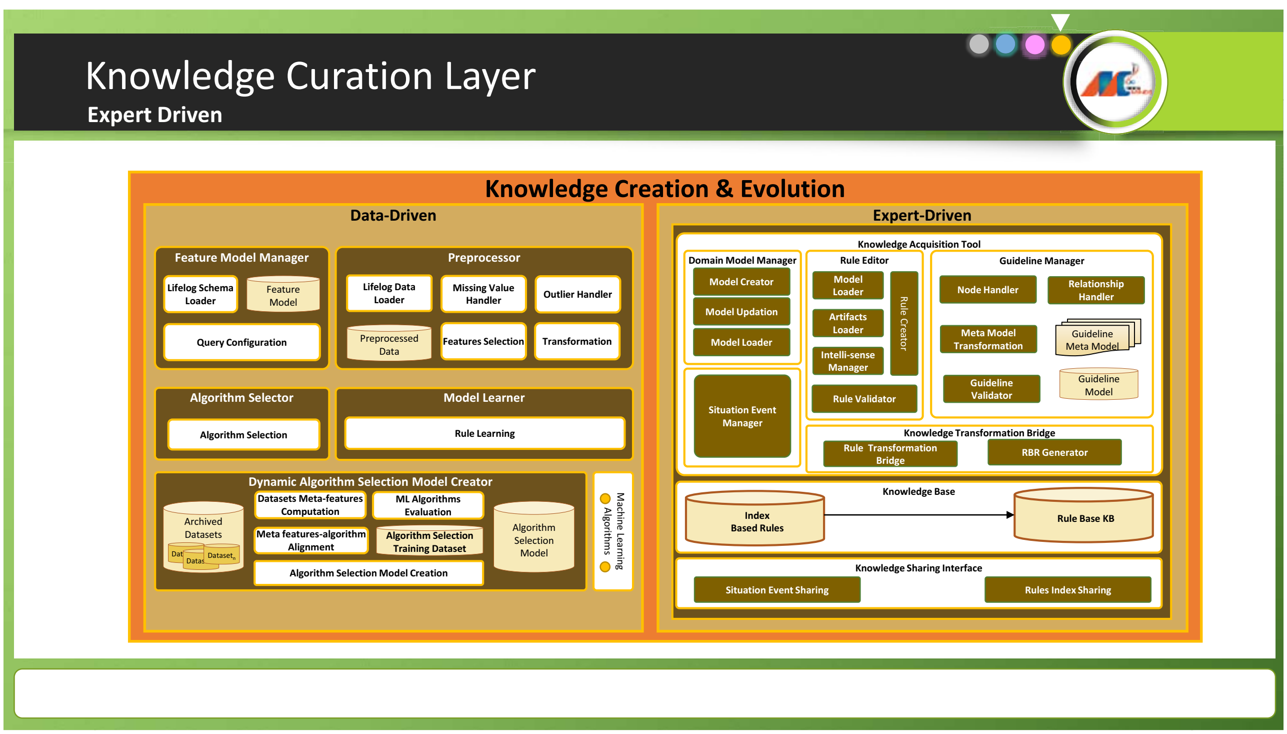
Data-Driven Knowledge Acquisition (Path 1)



Expert-Driven Knowledge Acquisition (Path 2)

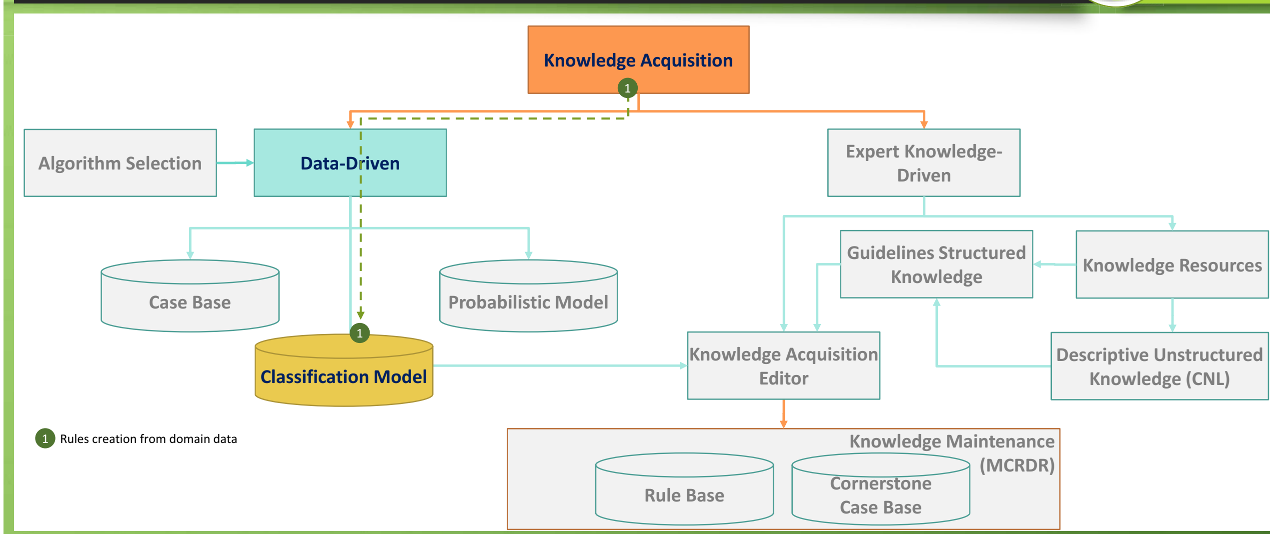



X Expert-Driven Knowledge Acquisition (Path 3)
Knowledge Acquisition Tool
Domain Model Manager
Guideline Manager
GTM
Rule Editor
2
Knowledge Transformation Bridge
1 Guideline Template Model (GTM) to transform the experts knowledge into guidelines tree
2 Knowledge Transformation Bridge transform knowledge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


Knowledge Curation Layer
Expert Driven
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
Algorithm Selection
Model Learner
Rule Learning
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
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

Knowledge Curation Layer (KCL 2.0)

Scope of Data Driven Knowledge Acquisition for MM Ver. 2.0



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

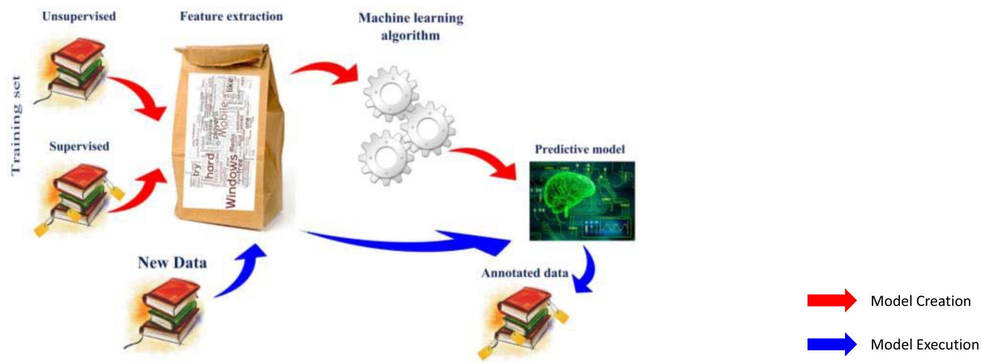
What is Data Driven?

- 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

- 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

- 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

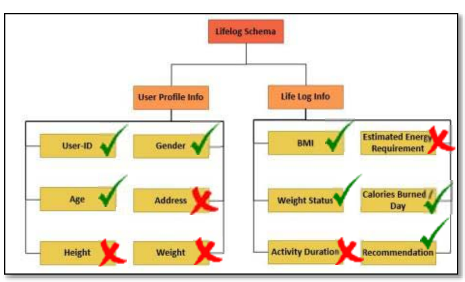
- Real world data is usually in raw structured format
  - incomplete: lacking attribute values, containing only aggregate data
    - e.g., occupation=“ ”
  - noisy: containing errors or outliers
    - e.g., Salary=“-10”
  - inconsistent: containing discrepancies in codes or names
    - e.g., Age=“42” 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.

What is Feature Modeling?

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



Related Works

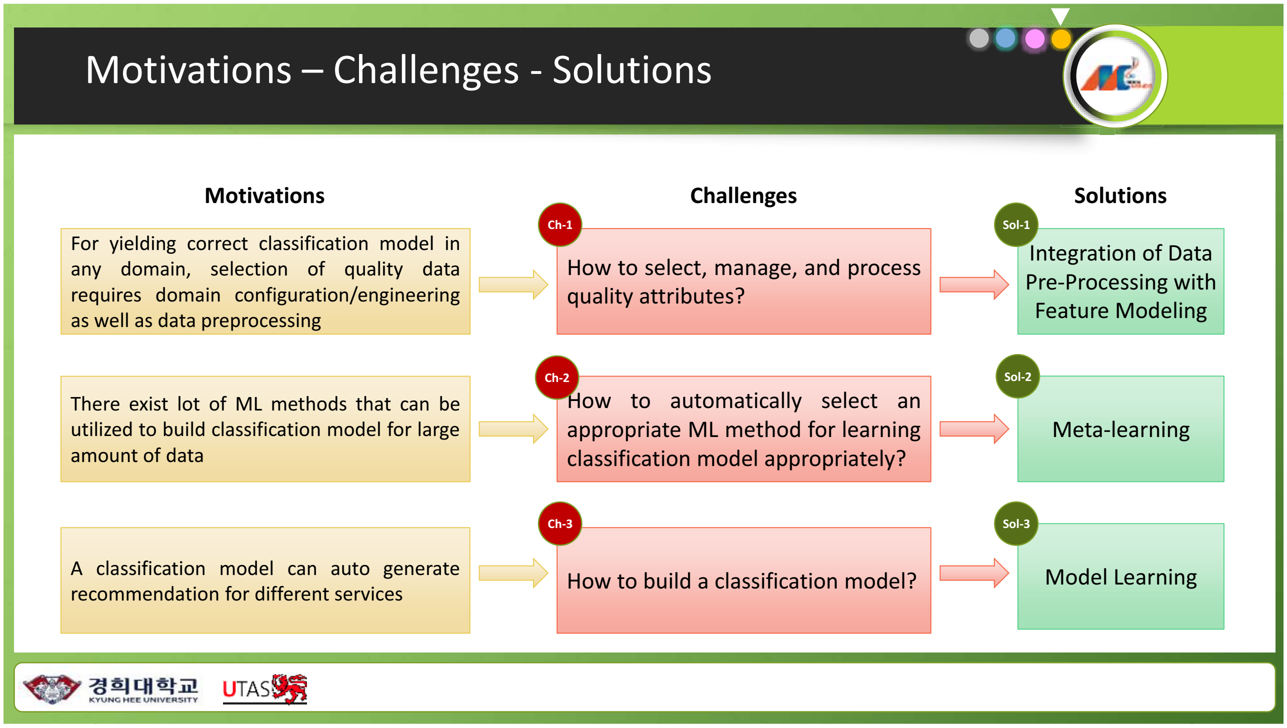
Category	Study	Features	Limitations
Feature Modeling	Sánchez [1], Vranić [2]	• Addresses quality attributes at run time by means of feature models • Context based feature selection • Used in the field of Software Product Lines (SPL)	• Requires extra engineering efforts at development time for designing 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

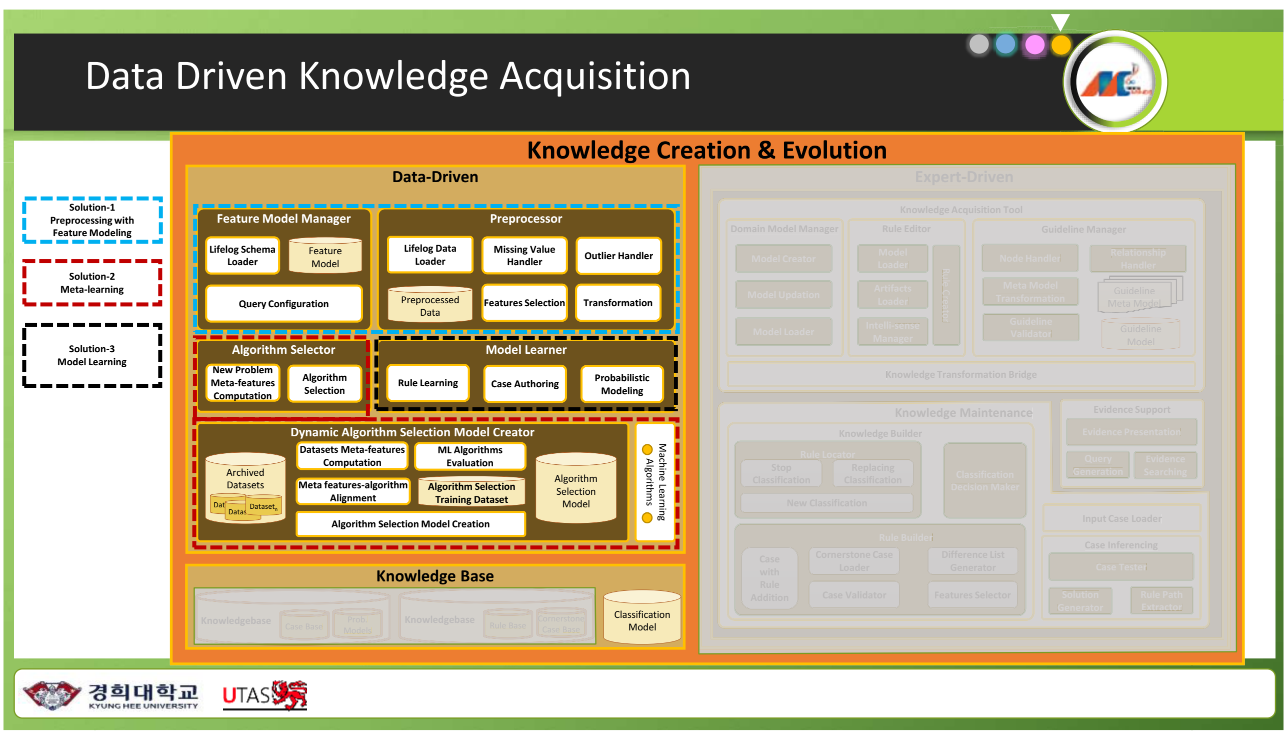
경희대학교 KYUNG HEE UNIVERSITY UTAS

Limitations

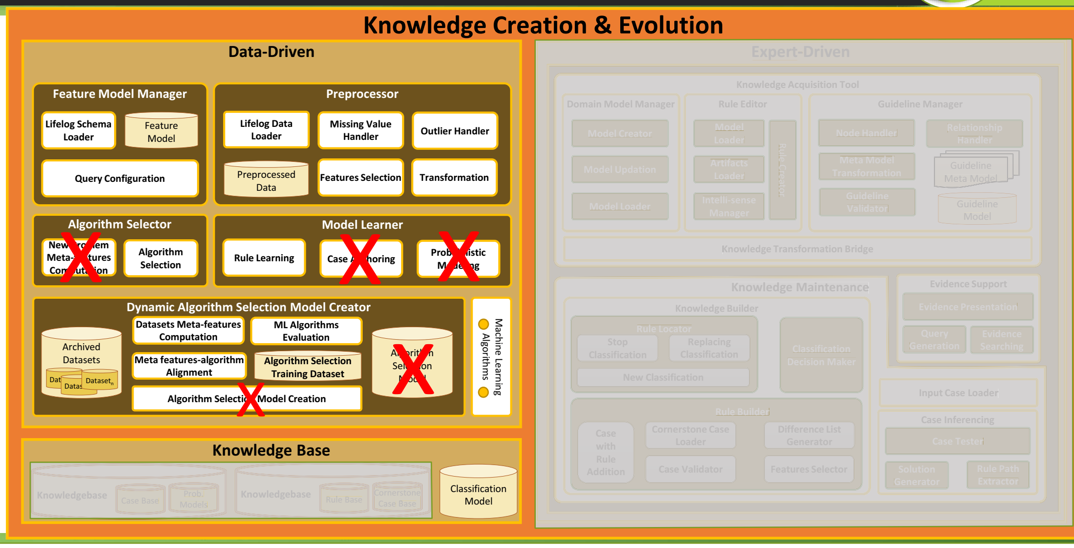
- Expert manually select machine learning algorithm for building classification model
- Use a sub-set of meta-features
- Lack of feature modeling for preprocessing of data

경희대학교 KYUNG HEE UNIVERSITY UTAS


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
경희대학교
KYUNG HEE UNIVERSITY
UTAS


Data Driven Knowledge Acquisition
Knowledge Creation & Evolution
Data-Driven
Expert-Driven
Solution-1
Preprocessing with Feature Modeling
Solution-2
Meta-learning
Solution-3
Model Learning
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
Knowledge Base
Classification Model
경희대학교
KYUNG HEE UNIVERSITY
UTAS

Data Driven Knowledge Acquisition Ver. 2.0

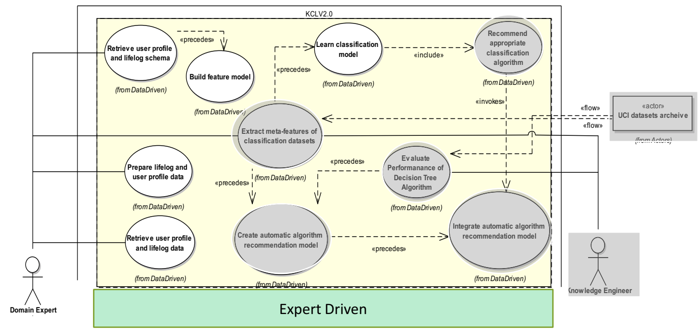


Goals of Data Driven Knowledge Acquisition Ver. 2.0

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

Data Driven Knowledge Acquisition Ver. 2.0 – Use cases

Use Case #ID	Use Case Name
KCL2-SUC-05	Retrieve user profile and lifelog schema
KCL2-SUC-01	Build feature model
KCL2-SUC-06	Retrieve user profile and lifelog data
KCL2-SUC-02	Prepare lifelog and user profile data
KCL2-SUC-13	Learn classification model



경희대학교 KYUNG HEE UNIVERSITY UTAS

Data Driven Knowledge Acquisition Ver. 2.0 – Use cases

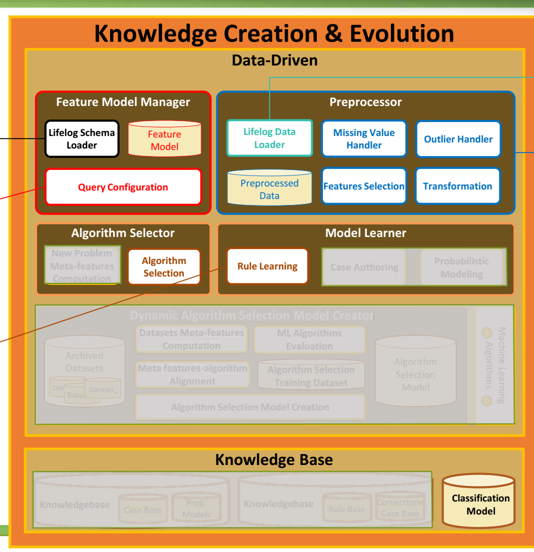
Retrieve user profile and lifelog schema

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

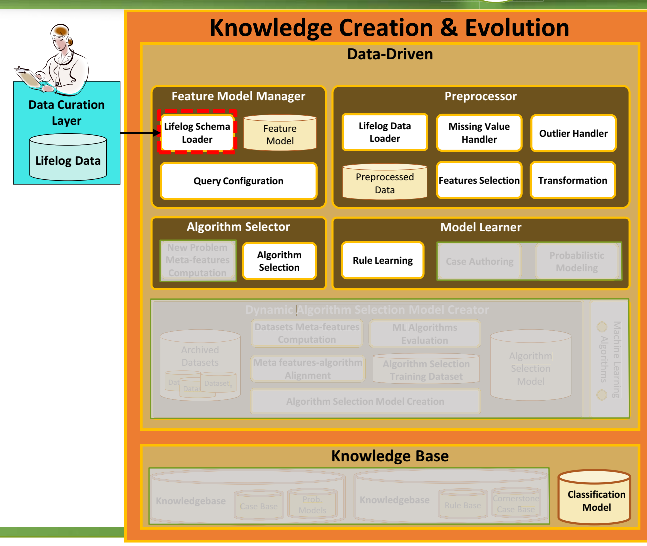
- Data Cleaning
  - Fill in missing values, identify or remove outliers
- Data Transformation
  - Generalization: concept hierarchy climbing
- Data Reduction
  - Obtains reduced representation in volume

경희대학교 KYUNG HEE UNIVERSITY UTAS

KCL2-SUC-05: Retrieve user profile and lifelog schema

Use Case

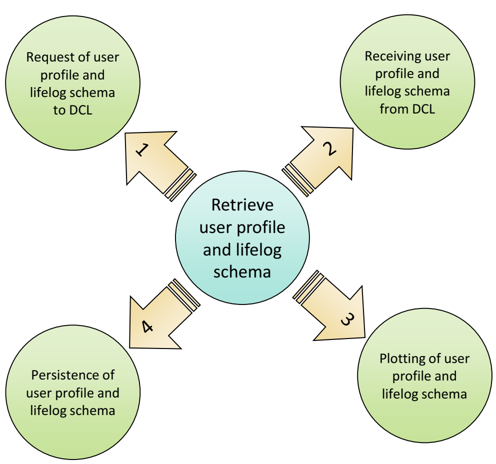
- **Objective**
  - To help domain expert to view all available features for building feature model
- **Methodology**
  - ***Communication***: Restful Service between *Feature Model Manager* and *DCL*
  - ***Lifelog Schema Loader*** *(operation)*
    1. **Input**: Required configuration of domain
    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*

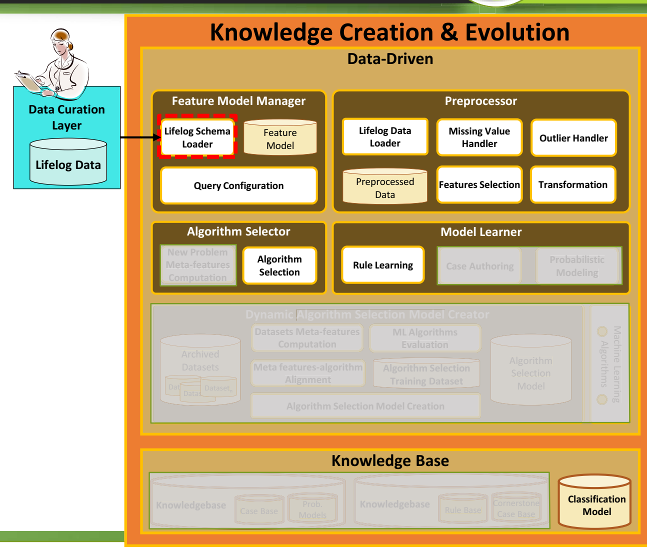


경희대학교 KYUNG HEE UNIVERSITY UTAS

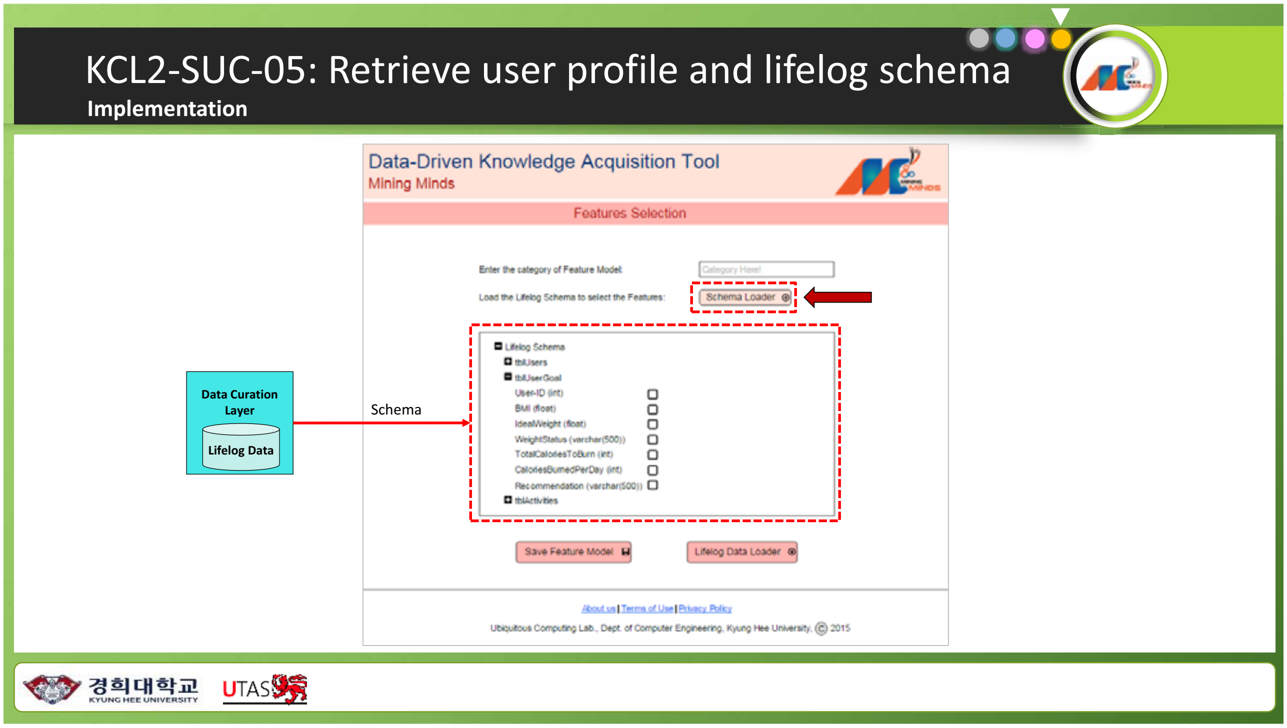
KCL2-SUC-05: Retrieve user profile and lifelog schema

Use Case Functions

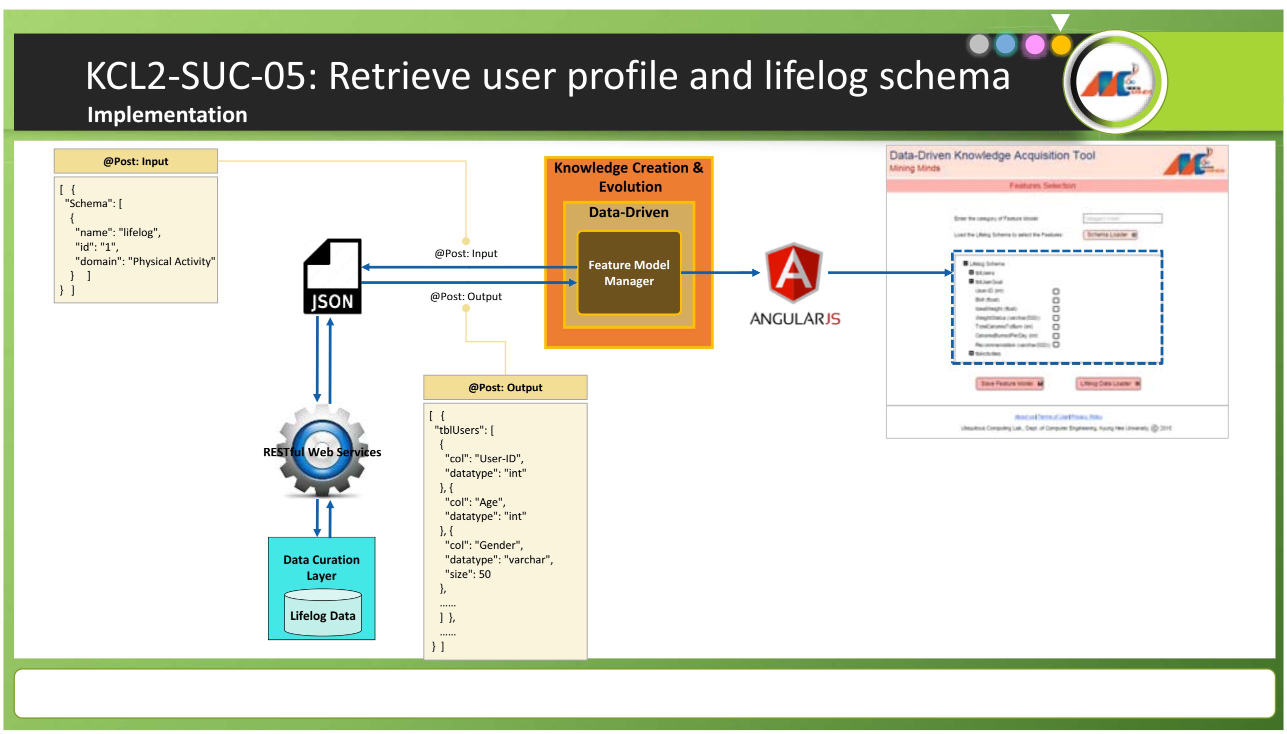




경희대학교 KYUNG HEE UNIVERSITY UTAS



KCL2-SUC-05: Retrieve user profile and lifelog schema
Implementation
Data-Driven Knowledge Acquisition Tool
Mining Minds
Features Selection
Enter the category of Feature Model:
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



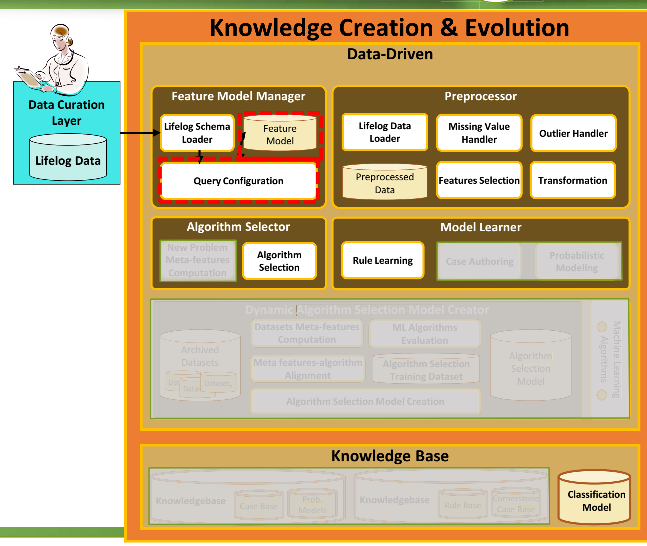
KCL2-SUC-05: Retrieve user profile and lifelog schema
Implementation
@Post: Input
[{
"Schema": [
{
"name": "lifelog",
"id": "1",
"domain": "Physical Activity"
}]
}]
JSON
@Post: Input
@Post: Output
Knowledge Creation & Evolution
Data-Driven
Feature Model Manager
ANGULARJS
Data-Driven Knowledge Acquisition Tool
Mining Minds
Features Selection
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
},
......
] },
......
}]

KCL2-SUC-01: Build feature model

Use Case

- **Objective**
  - ➢ To define the valid combinations of features in a domain for building feature model
- **Methodology**
  - ➢ ***Query Configuration*** *(operation)*

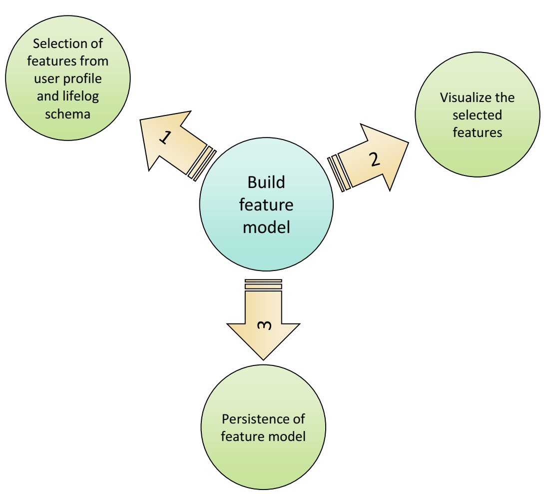
1. **Input**: Lifelog schema
2. **Processing**:
   a. Domain expert retrieves the schema from schema storage.
   b. System loads and plots the schema
   c. Domain expert builds the feature models as follows:
      a. Select the required features for corresponding domain
      b. Verify the consistency 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*

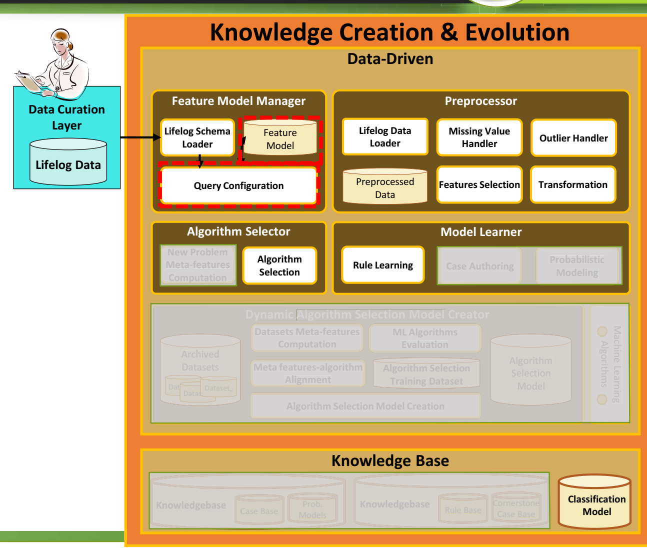


경희대학교 KYUNG HEE UNIVERSITY UTAS

KCL2-SUC-01: Build feature model

Use Case Functions

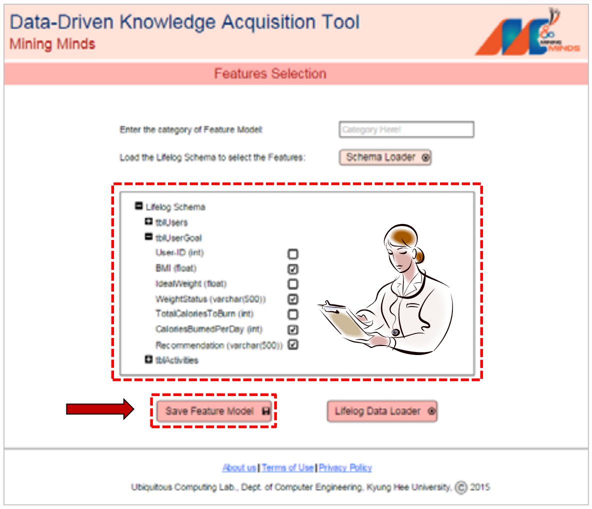




경희대학교 KYUNG HEE UNIVERSITY UTAS

KCL2-SUC-01: Build feature model

Implementation




```
<<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>
```

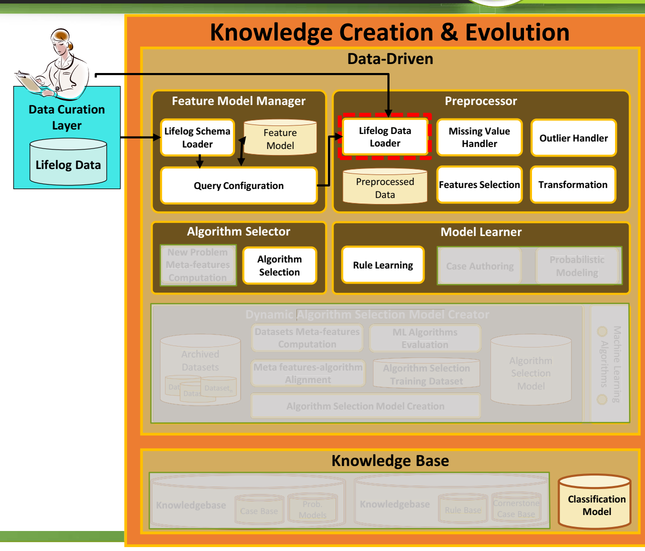

Feature Model

경희대학교 KYUNG HEE UNIVERSITY UTAS

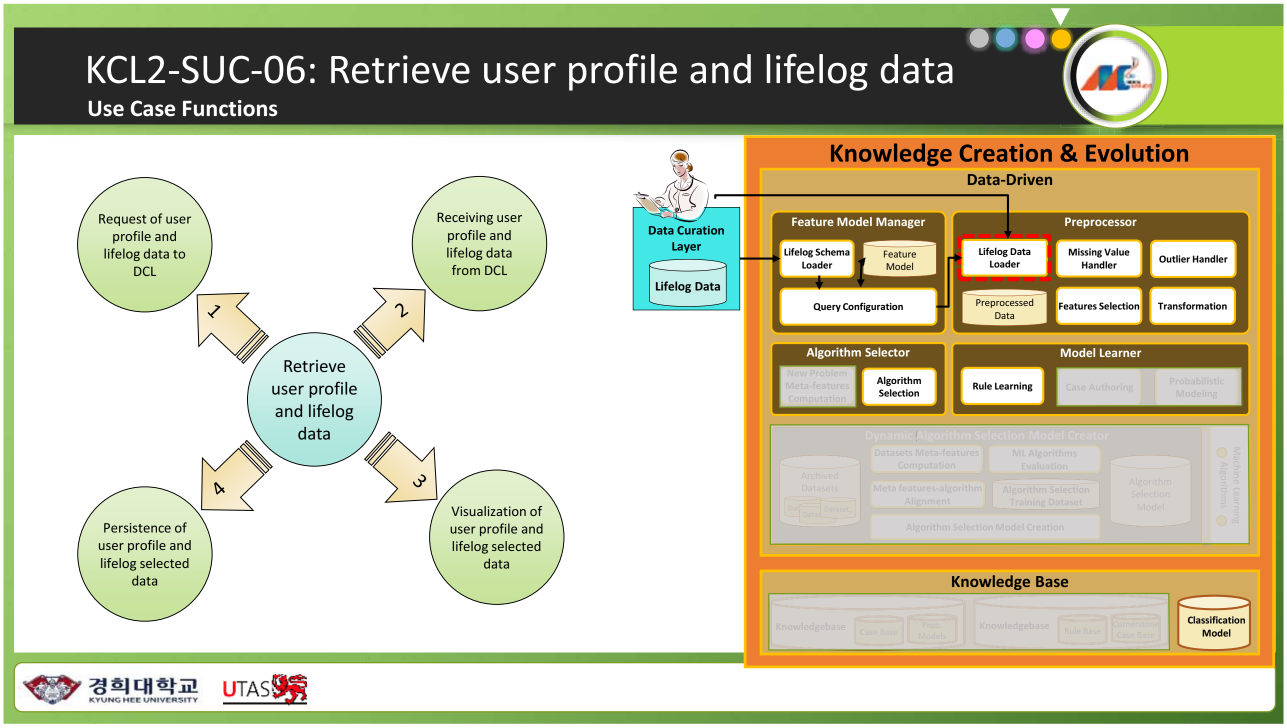
KCL2-SUC-06: Retrieve user profile and lifelog data

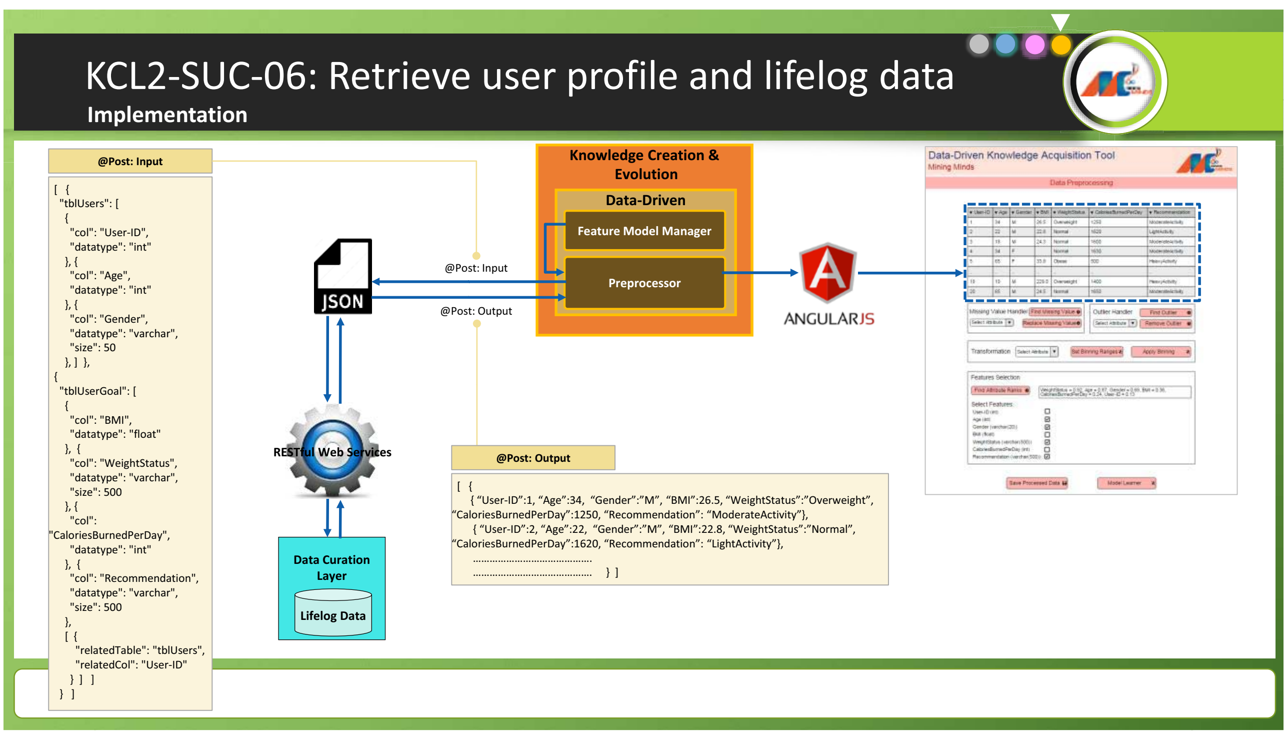
Use Case

- **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*



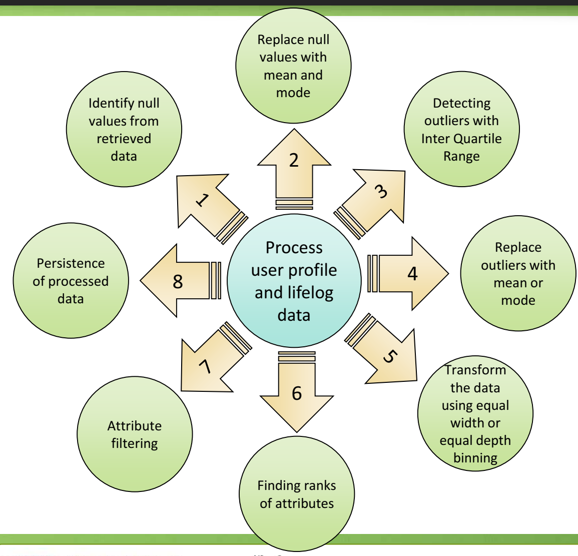
경희대학교 KYUNG HEE UNIVERSITY UTAS

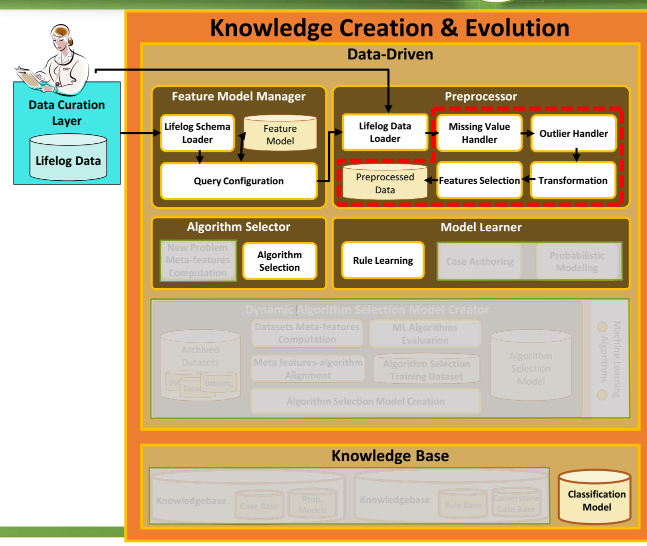

KCL2-SUC-06: Retrieve user profile and lifelog data
Use Case Functions
Request of user profile and lifelog data to DCL
Receiving user profile and lifelog data from DCL
Retrieve user profile and lifelog data
Persistence of user profile and lifelog selected data
Visualization of user profile and lifelog selected data
1
2
3
4
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
Knowledge Base
Classification Model
경희대학교
KYUNG HEE UNIVERSITY
UTAS


KCL2-SUC-06: Retrieve user profile and lifelog data
Implementation
@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-SUC-02: Prepare lifelog and user profile data

Use Case Functions

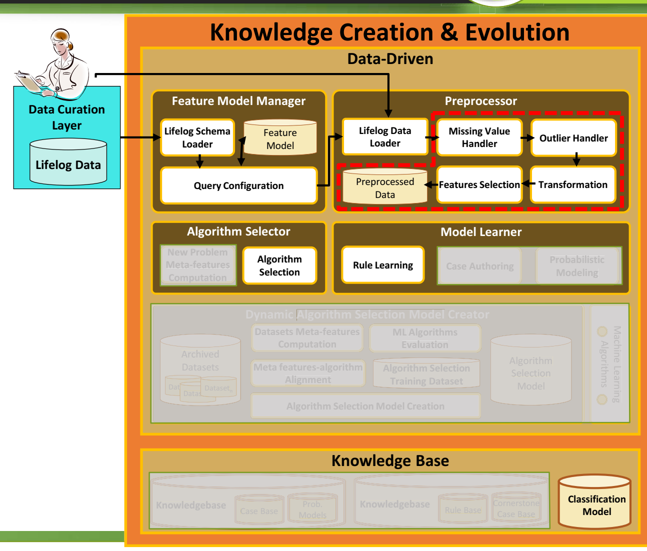




KCL2-SUC-02: Prepare lifelog and user profile data (1/4)

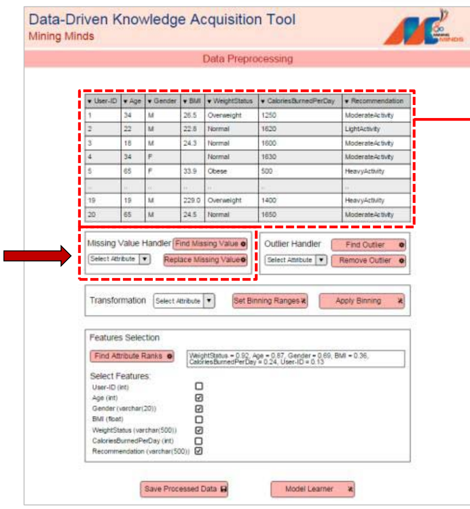
Missing Value Handler | Outlier Handler | Transformation | Features Selection

- **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-SUC-02: Prepare lifelog and user profile data (1/4)

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

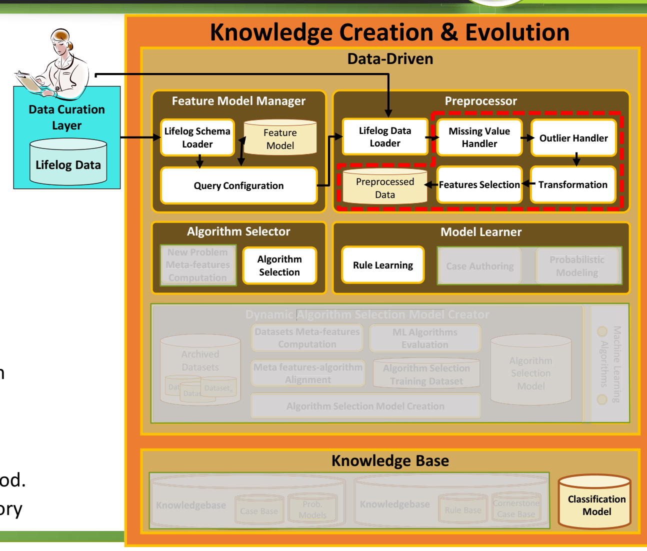
- **Mean/Mode substitution**
  - Replace missing value with sample mean or mode, which are two kinds of "averages".
  - The "**mean**" is the "average" you're used to, where you add up all the numbers and then divide by the number of numbers.
  - The "**mode**" is the value that occurs most often. If 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.

경희대학교 KYUNG HEE UNIVERSITY UTAS

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

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

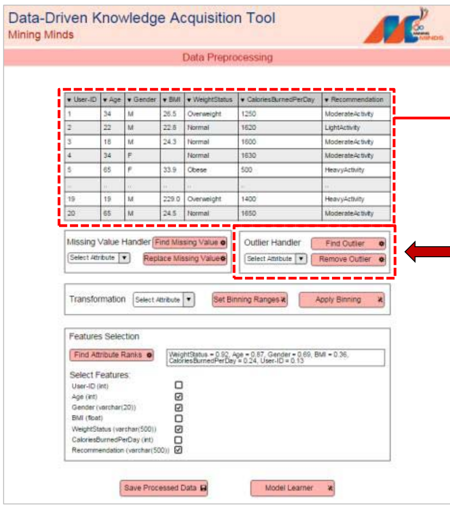
- **Objective**
  - To preprocess the data to generate models with high accuracy.
- **Methodology**
  - ***Outlier Handler*** *(operation)*
    1. **Input**: processed data
    2. **Processing**:
       a. Domain expert loads the processed data
       b. System displays the retrieved data
       c. For 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*



경희대학교 KYUNG HEE UNIVERSITY UTAS

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

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



Unprocessed Data

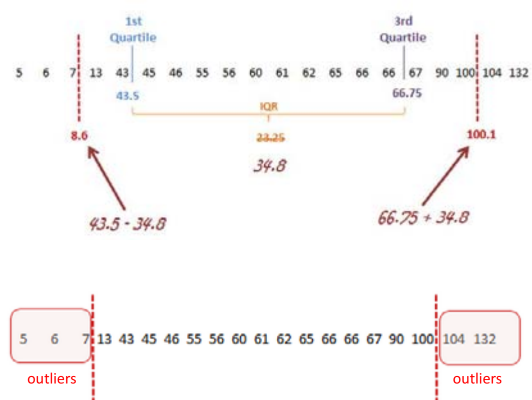
User-ID	Age	Gender	BMI	WeightStatus	CaloriesBurnedPerDay	Recommendation
1	34	M	26.5	Overweight	1250	ModerateActivity
2	22	M	22.8	Normal	1620	LightActivity
3	18	M	24.3	Normal	1600	ModerateActivity
4	34	F		Normal	1630	ModerateActivity
5	65	F	33.9	Obese	500	HeavyActivity
..	..	..	..		...	..
19	19	M	229.0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

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

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

- **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)

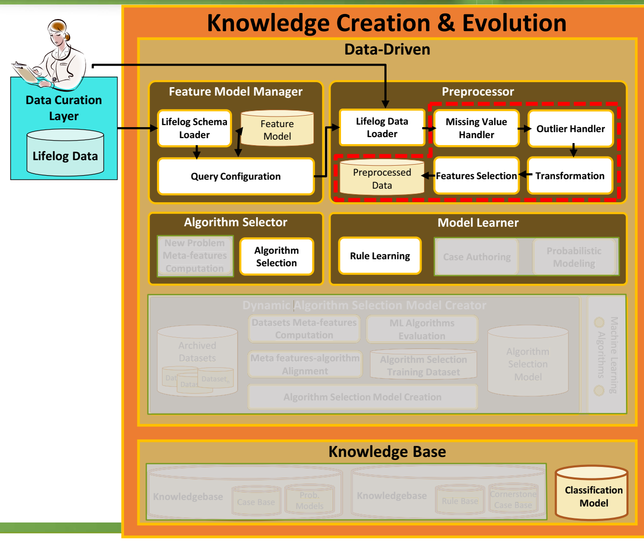


경희대학교 KYUNG HEE UNIVERSITY UTAS

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

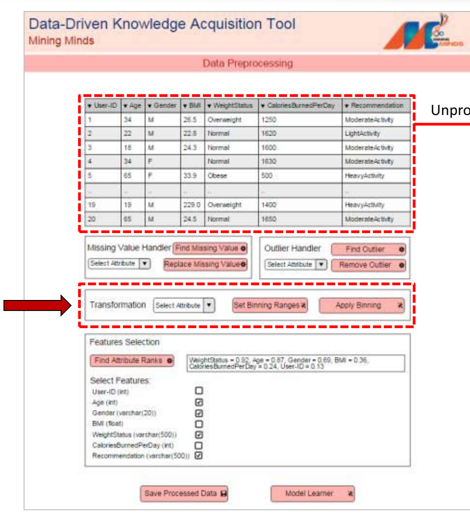
Missing Value Handler | Outlier Handler | **Transformation** | Features Selection

- **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-SUC-02: Prepare lifelog and user profile data (3/4)

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

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

CaloriesBurnedPerDay
Low
Normal
Low
Low
Low
...
Low
Normal

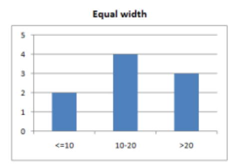
- Transformation
  - A function that maps the entire set of values of a given attribute to a new set of replacement values such that each old value can be identified with one of the new values
- Methods
  - **Discretization: Concept hierarchy climbing**
  - Smoothing: Remove noise from data
  - Attribute/feature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

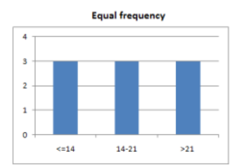
KCL2-SUC-02: Prepare lifelog and user profile data (3/4)

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

- **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,+)

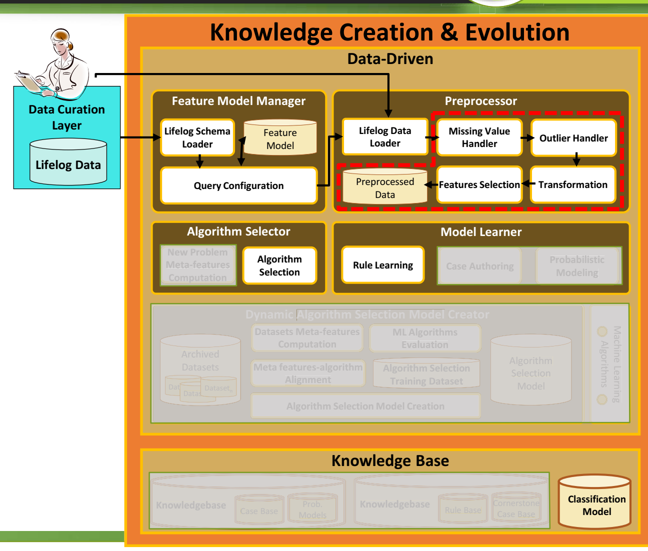




KCL2-SUC-02: Prepare lifelog and user profile data (4/4)

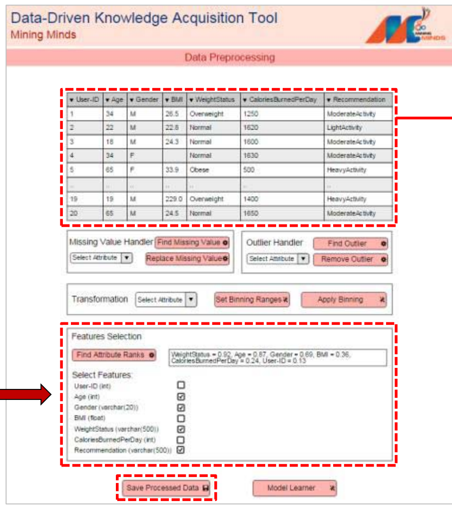
Missing Value Handler | Outlier Handler | Transformation | **Features Selection**

- **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



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

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

- 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

$$Information\ Gain(S,F_i) = Entropy(S) - \sum_{v_i \epsilon V_{Fj}} \frac{|S_{vi}|}{|S|} * Entropy(S_{vi})$$

$$Entropy(S) = \sum_{i=1}^{c} -p_i * log_2(p_i)$$

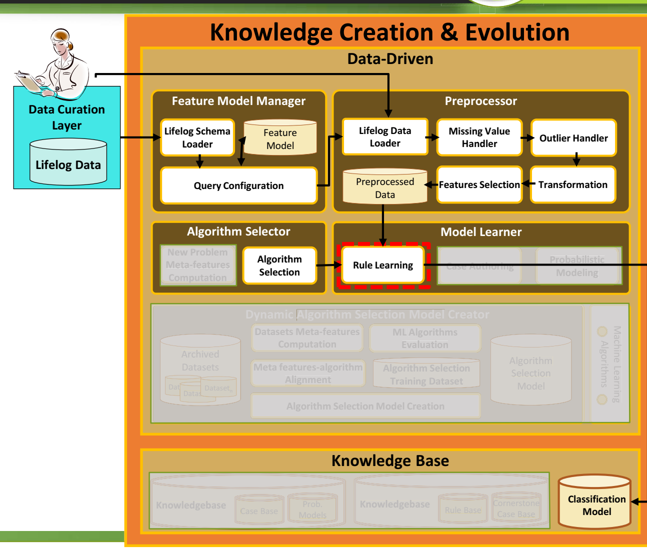
Where: p_i - proportion of the examples belonging to the i-th class.
S_{vi} – subset of S, for which feature F_j has value vi
V_{Fj} – set of all possible values of feature F_j

경희대학교 KYUNG HEE UNIVERSITY UTAS

KCL2-SUC-13: Learn classification model

Use Case

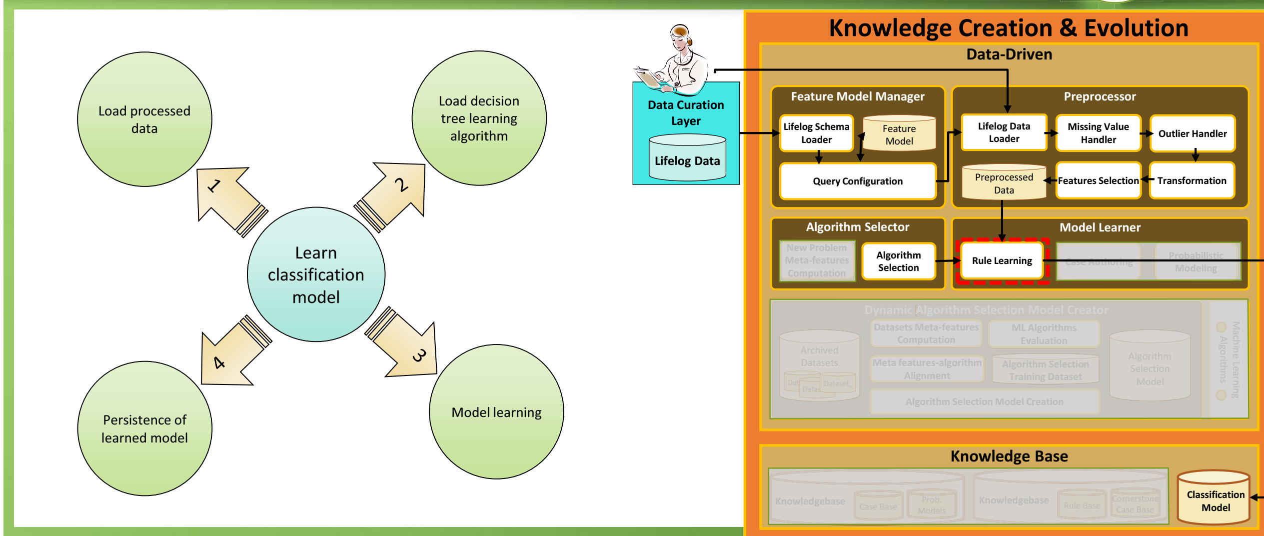
- **Objective**
  - To generate the classification model.
- **Methodology**
  - ***Rule Learning*** *(operation)*
    1. **Input**: processed data
    2. **Processing**:
       a. Domain expert loads the user profile lifelog processed data for selected domain
       b. System loads the corresponding 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



경희대학교 KYUNG HEE UNIVERSITY UTAS

KCL2-SUC-13: Learn classification model

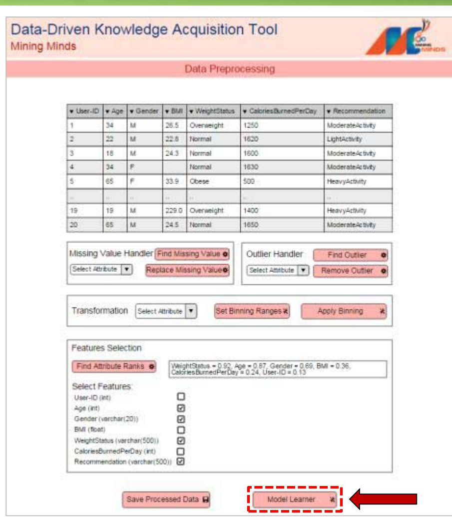
Use Case Functions

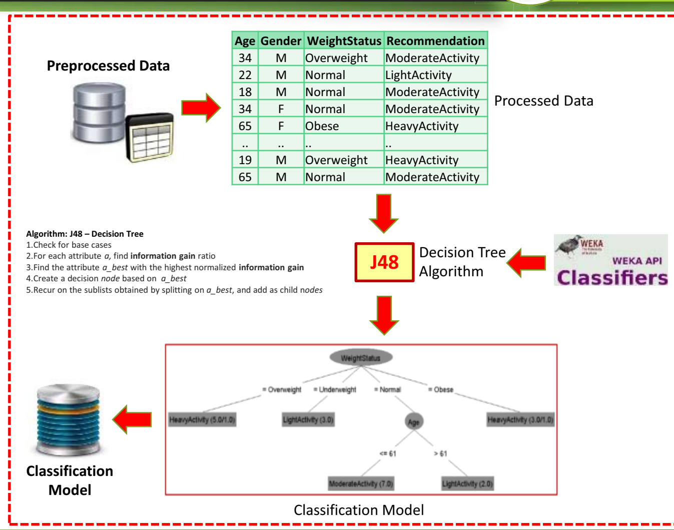


경희대학교 KYUNG HEE UNIVERSITY UTAS

KCL2-SUC-13: Learn classification model

Implementation

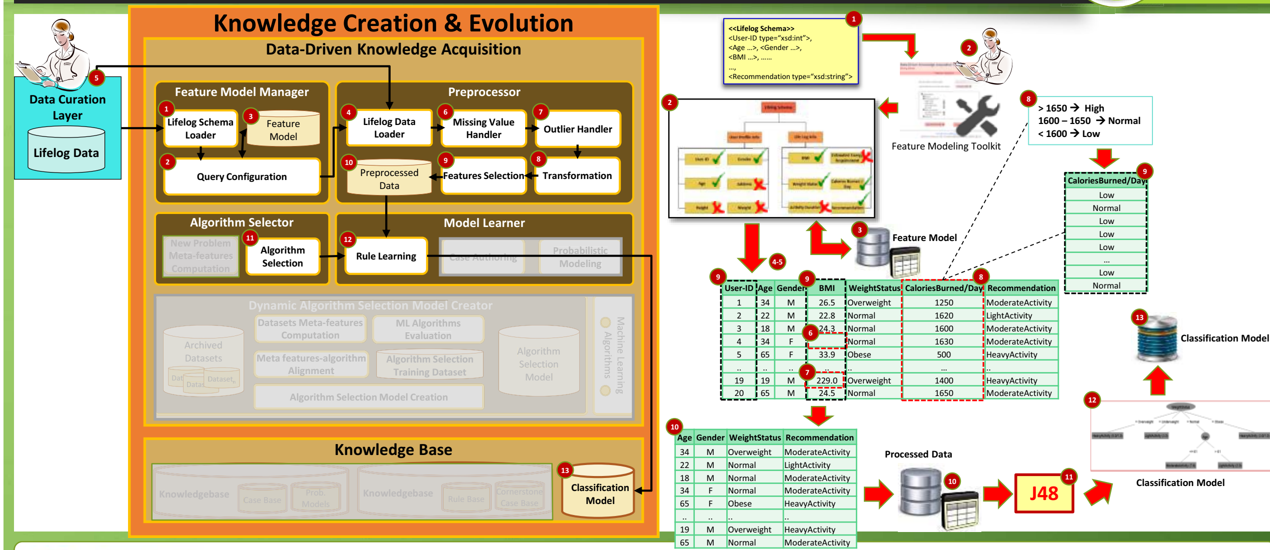




Age	Gender	WeightStatus	Recommendation
34	M	Overweight	ModerateActivity
22	M	Normal	LightActivity
18	M	Normal	ModerateActivity
34	F	Normal	ModerateActivity
65	F	Obese	HeavyActivity
..	..	..	..
19	M	Overweight	HeavyActivity
65	M	Normal	ModerateActivity

경희대학교 KYUNG HEE UNIVERSITY UTAS

Workflow



Dependencies, Issues and Challenges

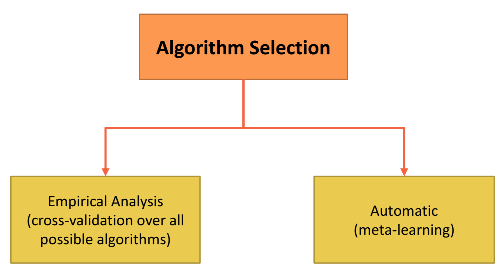
- The user profile and life log schema and data retrieval from DCL
- Feature model development
- Data preprocessing
- Model learning after configuration with available machine learning algorithms API's

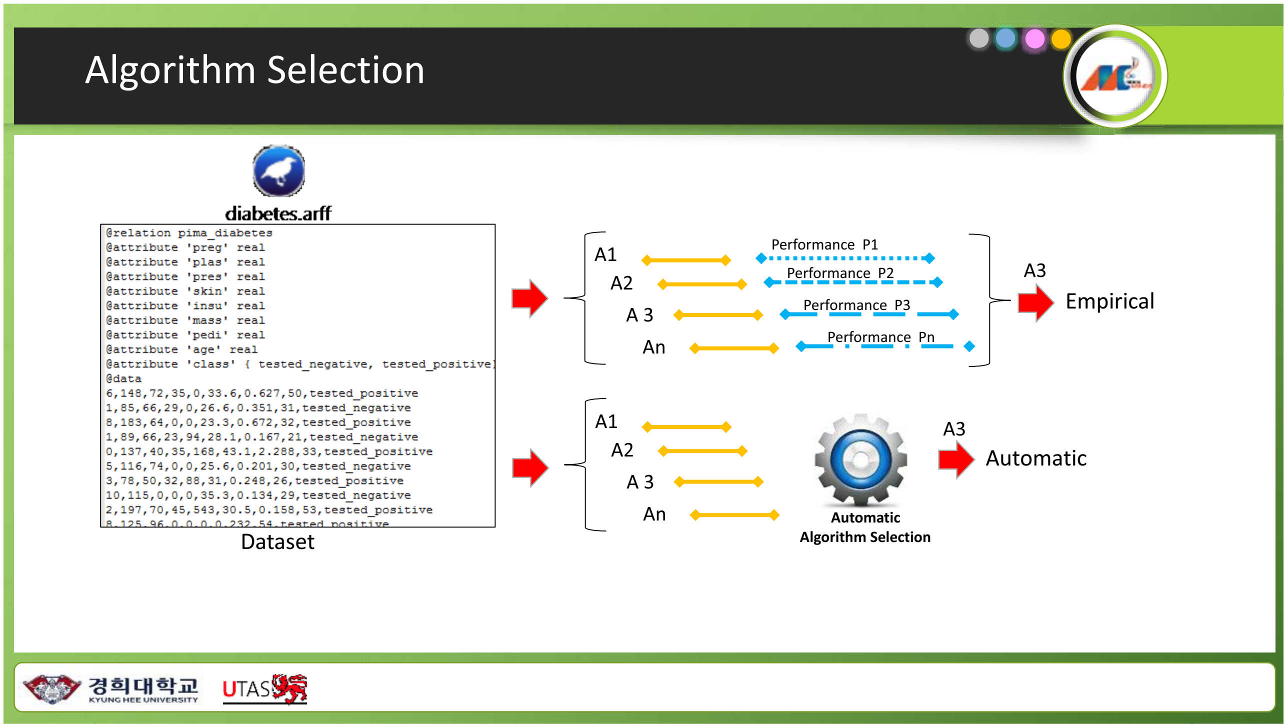
Selected Tools and Techniques

- Language
  - Java
- Frameworks:
  - Spring MVC
  - AngularJS
  - Hibernate
- APIs:
  - Weka
  - Java-ML
- Database:
  - MS SQL Server
- IDEs:
  - Eclipse

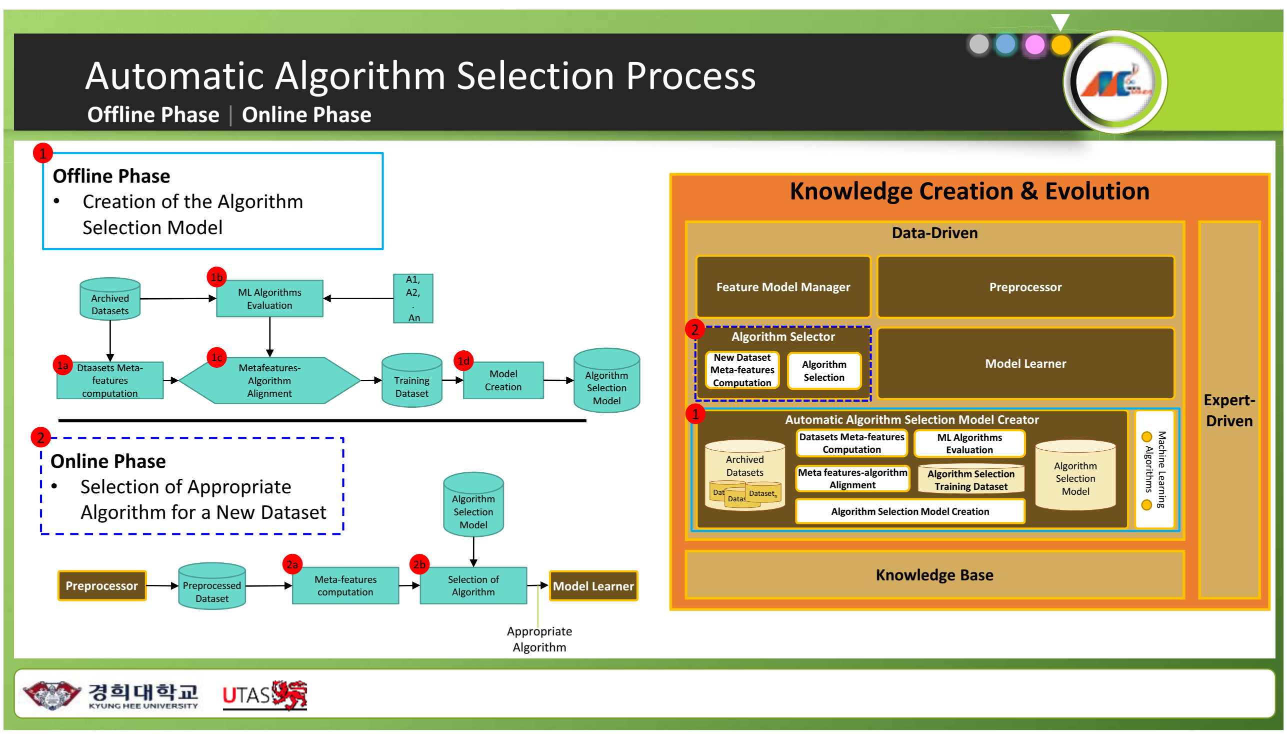
Algorithm Selection

- The process of selecting an appropriate algorithm for learning a dataset
- Types of selecting appropriate algorithms
  - Empirical analysis
  - Automatic recommendation





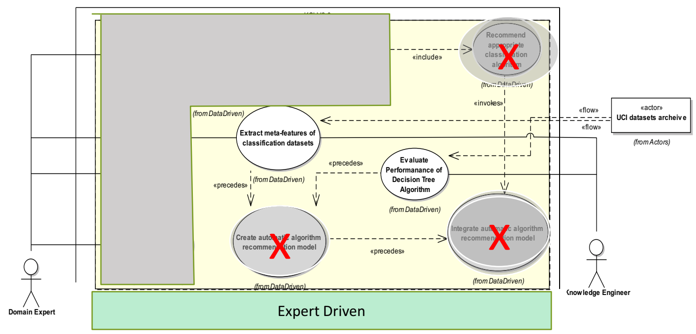
Algorithm Selection
diabetes.arff
@relation pima_diabetes
@attribute 'preg' real
@attribute 'plas' real
@attribute 'pres' real
@attribute 'skin' real
@attribute 'insu' real
@attribute 'mass' real
@attribute 'pedi' real
@attribute 'age' real
@attribute 'class' { tested_negative, tested_positive}
@data
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
1
Offline Phase
• Creation of the Algorithm Selection Model
Archived Datasets
1b
ML Algorithms Evaluation
A1, A2, . An
1a
Dtaasets Meta-features computation
1c
Metafeatures-Algorithm Alignment
Training Dataset
1d
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
Algorithm Selector
New Dataset Meta-features Computation
Algorithm Selection
Model Learner
Automatic Algorithm Selection Model Creator
Archived Datasets
Datasets Meta-features Computation
ML Algorithms Evaluation
Meta features-algorithm Alignment
Algorithm Selection Training Dataset
Algorithm Selection Model Creation
Algorithm Selection Model
Machine Learning Algorithms
Knowledge Base
Expert-Driven
경희대학교
KYUNG HEE UNIVERSITY
UTAS

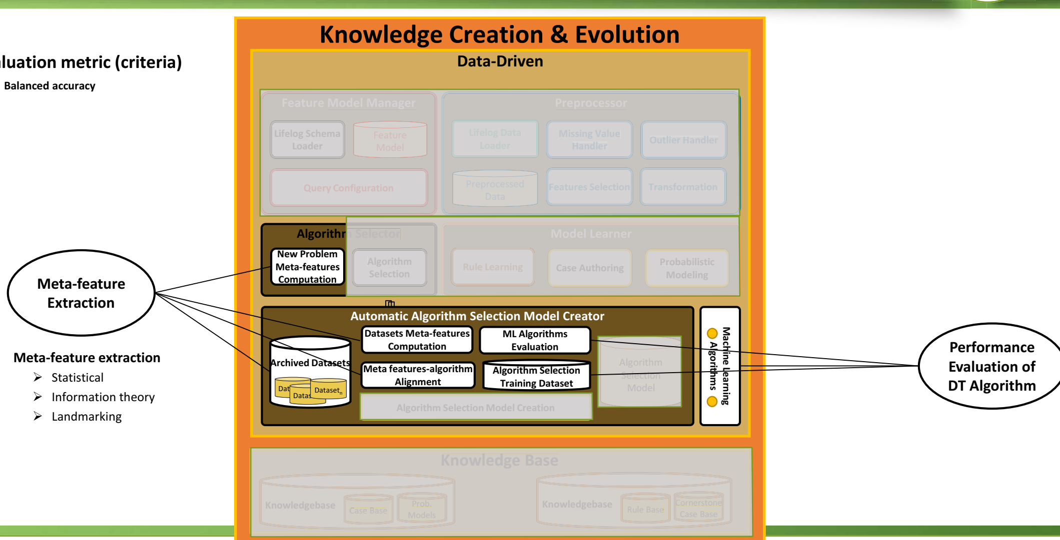
Use-case diagram of automatic algorithm selection

Use Case #ID	Use Case Name
KCL2-SUC-07	Extract meta-features of classification datasets
KCL2-SUC-08	Evaluate performance of classification algorithms



Use-case diagram of automatic algorithm selection

- **Evaluation metric (criteria)**
  - **Balanced accuracy**



- **Meta-feature extraction**
  - Statistical
  - Information theory
  - Landmarking

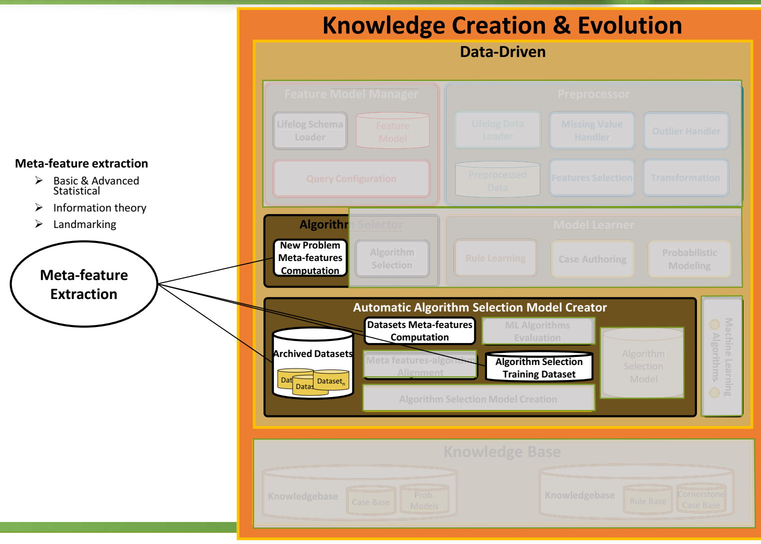
Meta-Features Extraction

Use Case

Use-case (KCL2-SUC-07): *Extract meta-features of classification dataset*

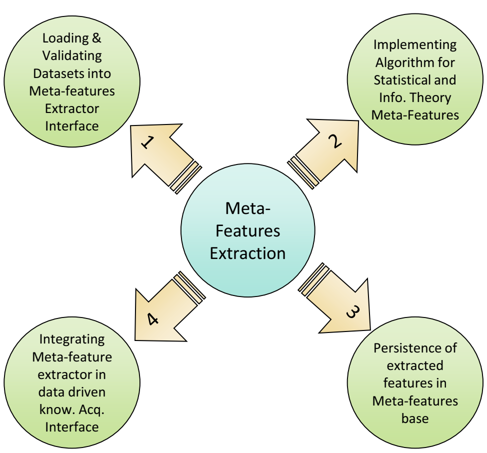
- **Objective**
  - Extraction of meta-characteristics
- **Methodology**
  - *Library used*: OpenML dataset characterization library is used
  - ***Dataset Meta-features computation*** *(operation)*
    1. **Input**: UCI and OpenMl classification datasets
    2. **Processing**:
       a. Takes each dataset 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

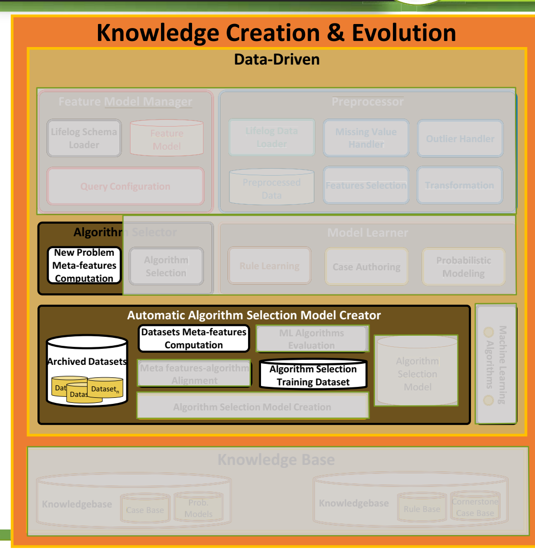
- **Meta-feature extraction**
  - Basic & Advanced Statistical
  - Information theory
  - Landmarking



Meta-Features Extraction

Use Case Features Functions





List of Meta-features

No.	Simple 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

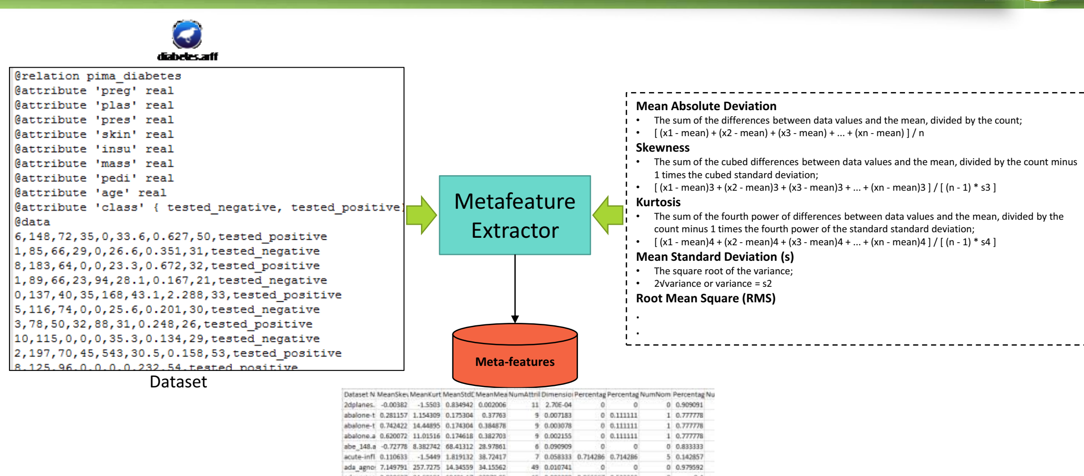
No.	Advanced Statistical Features
Advanced Statistic 1	MeanStdDevOfNumericAtts
Advanced Statistic 2	MeanMeansOfNumericAtts
Advanced Statistic 3	NegativePercentage
Advanced Statistic 4	PositivePercentage
Advanced Statistic 5	DefaultAccuracy
Advanced Statistic 6	IncompleteInstanceCount
Advanced Statistic 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)



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 (80-100)
  - ➢ Clustering

OpenML Dataset Characterization Library

UCI and OpenML Classification Datasets

	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

Meta-features

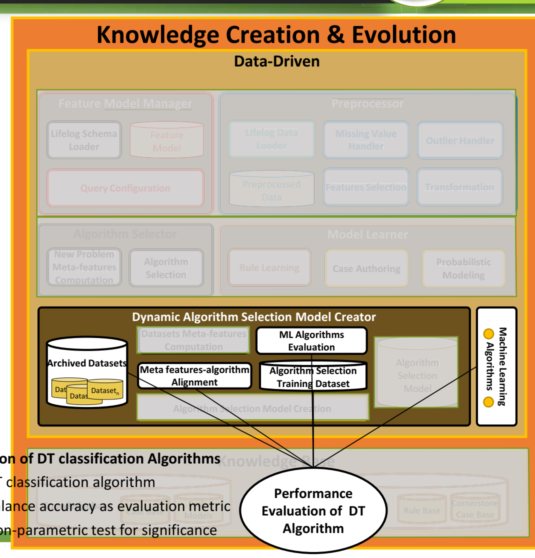
$\# \, Exp. \, for \, meta - features \, computation$
$= n \, (no \, datasets) \, * m(no \, metafetures)$

경희대학교 KYUNG HEE UNIVERSITY UTAS

Performance Evaluation of Classification Algorithm

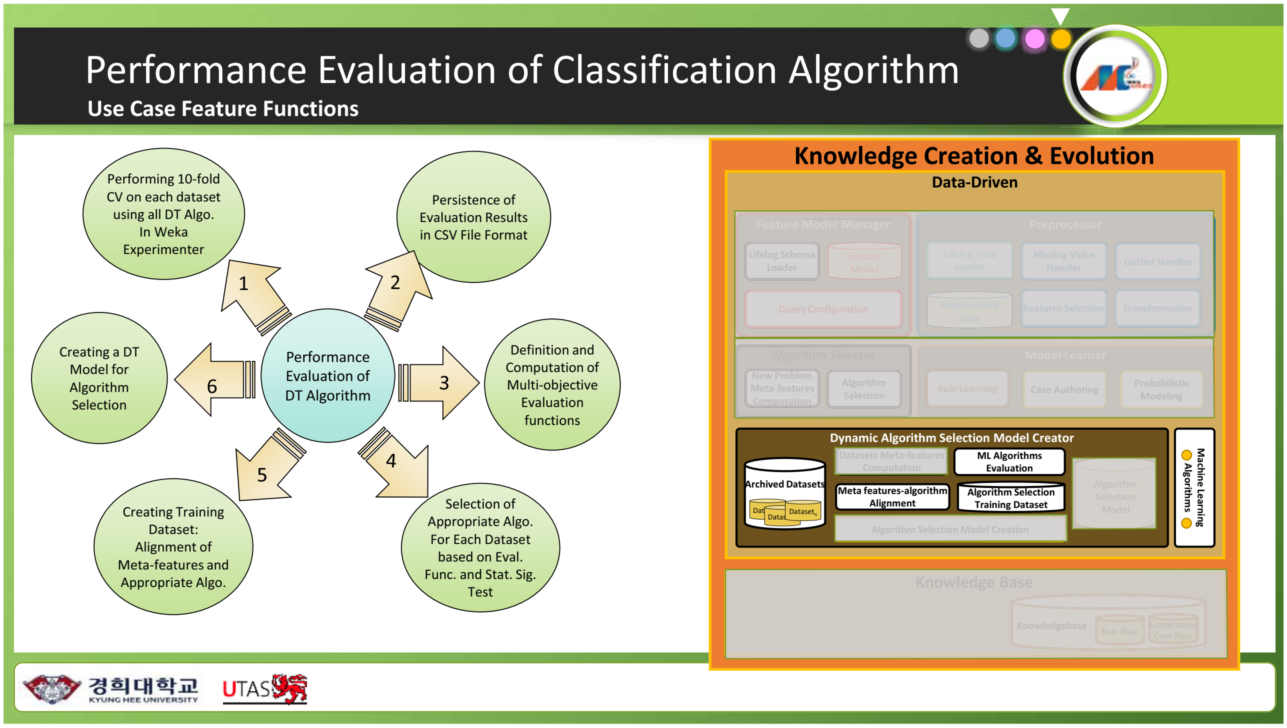
Use-case (KCL2-SUC-08): *Evaluate decision tree classification algorithms*

- **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 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 on-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. Create a DT-based Model for Algorithm Selection
    3. **Output**: Appropriate algorithm

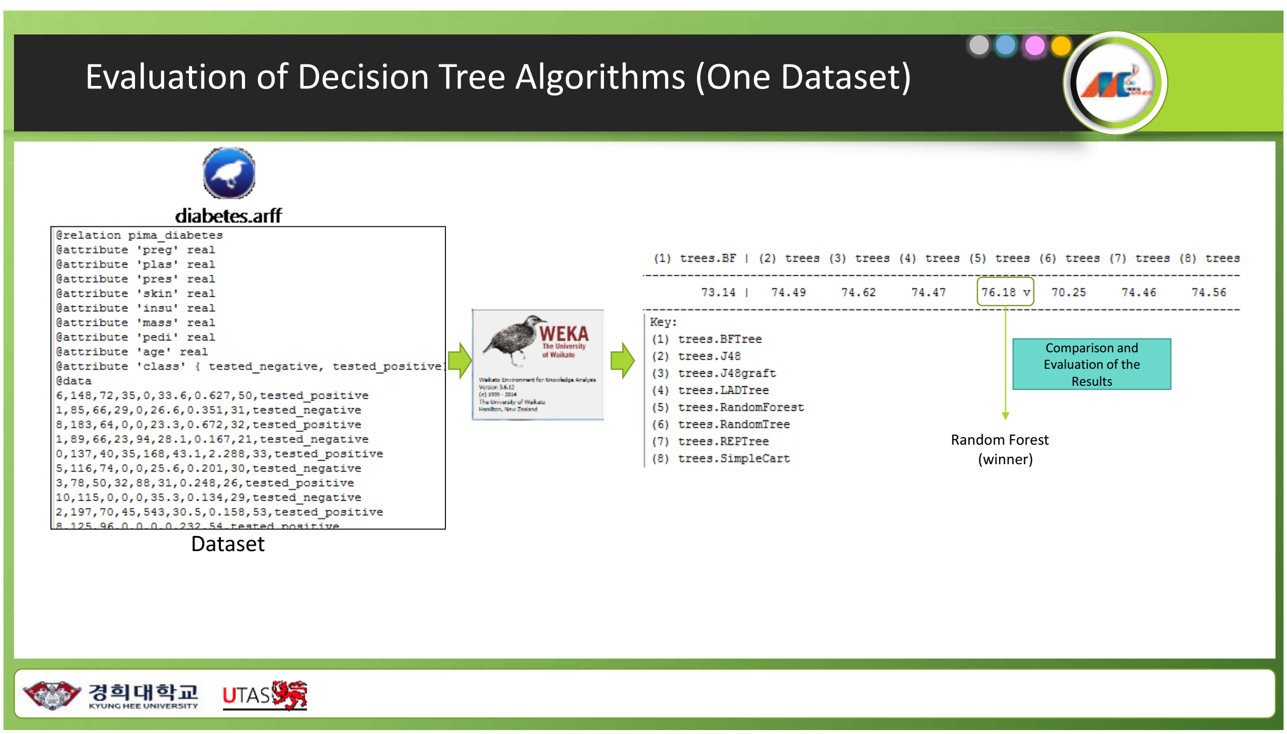


- **Evaluation of DT classification Algorithms**
  - ➢ DT classification algorithm
  - ➢ Balance accuracy as evaluation metric
  - ➢ Non-parametric test for significance

경희대학교 KYUNG HEE UNIVERSITY UTAS



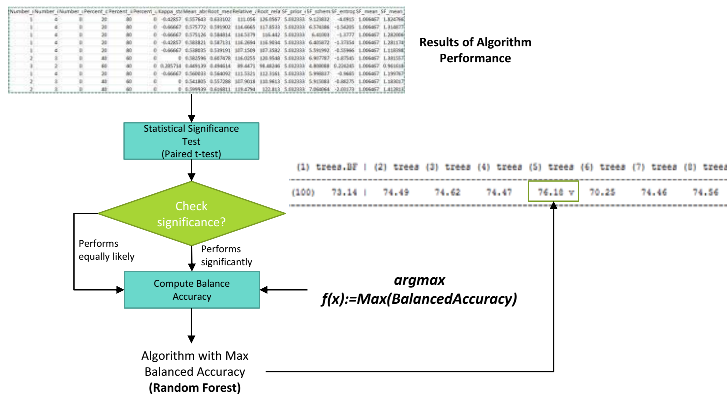
Performance Evaluation of Classification Algorithm
Use Case Feature Functions
Performing 10-fold CV on each dataset using all DT Algo. In Weka Experimenter
Persistence of Evaluation Results in CSV File Format
Creating a DT Model for Algorithm Selection
Performance Evaluation of DT Algorithm
Definition and Computation of Multi-objective Evaluation functions
Creating Training Dataset: Alignment of Meta-features and Appropriate Algo.
Selection of Appropriate Algo. For Each Dataset based on Eval. Func. and Stat. Sig. Test
1
2
3
4
5
6
Knowledge Creation & Evolution
Data-Driven
Dynamic Algorithm Selection Model Creator
Archived Datasets
ML Algorithms Evaluation
Meta features-algorithm Alignment
Algorithm Selection Training Dataset
Machine Learning Algorithms
Knowledge Base
경희대학교
KYUNG HEE UNIVERSITY
UTAS



Evaluation of Decision Tree Algorithms (One Dataset)
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)

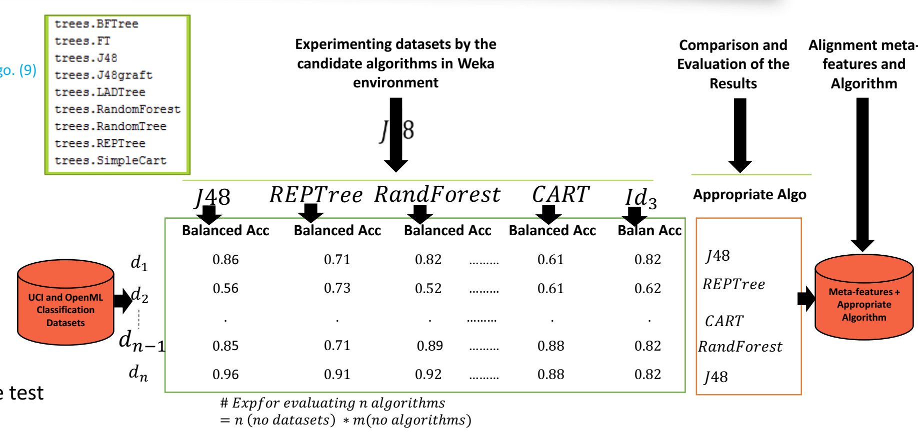
- Perform statistical significance test (Paired t-test)
  1. Perform significantly better
  2. Perform equally likely, cannot decide
  3. Perform significantly poor
- Drops algorithms that performs significantly poor
- Perform evaluation for balanced accuracy
  - $Balanced\ Accuracy = \frac{Sensitativity+Specificity}{2} = \frac{0.5*TP}{TP+FN} + \frac{0.5*TN}{TN+FP}$
- Select the algorithm with highest balanced accuracy value



경희대학교 KYUNG HEE UNIVERSITY UTAS

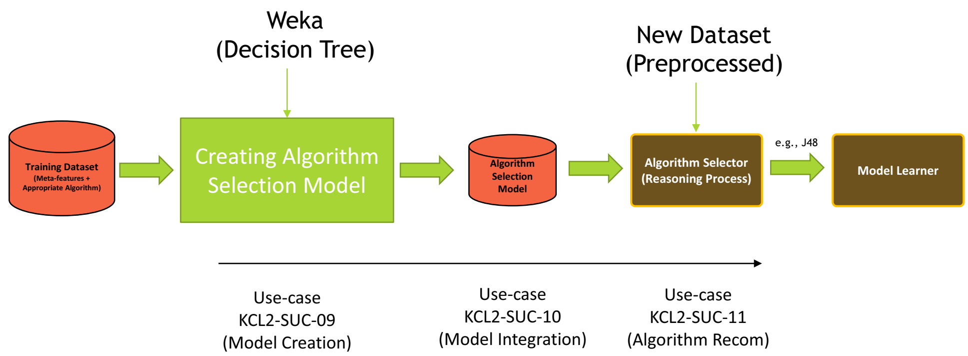
Comparison and Evaluation of Results (All Datasets)

- Types algorithms
  - Decision tree-based algo. (9)
  - Rule-based algo.
  - Probabilistic algo.
  - Distance-based algo.
  - Meta-learning algo.
  - Function-based algo.
- Evaluation criteria
  - Accuracy
  - F-score
  - Balanced accuracy
  - Precision
  - Error rate
  - Time-complexity
  - .
  - .
- Statistical significance test
  - Parametric
  - Non-parametric



경희대학교 KYUNG HEE UNIVERSITY UTAS

Algorithm Selection Model Creation and Algorithm Recommendation (MMV2.5)



Selected Tools and Techniques

- APIs:
  - Weka
  - OpenML
- Tools:
  - Weka
  - RapidMiner

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 appropriate algorithm for a new dataset at run time

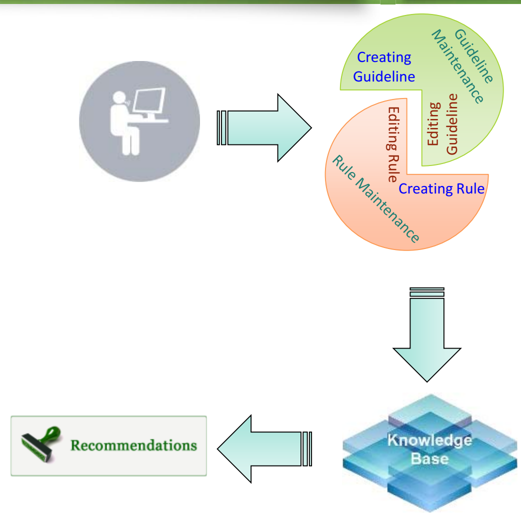
경희대학교 KYUNG HEE UNIVERSITY UTAS

Knowledge Curation Layer (KCL 2.0)

경희대학교 KYUNG HEE UNIVERSITY

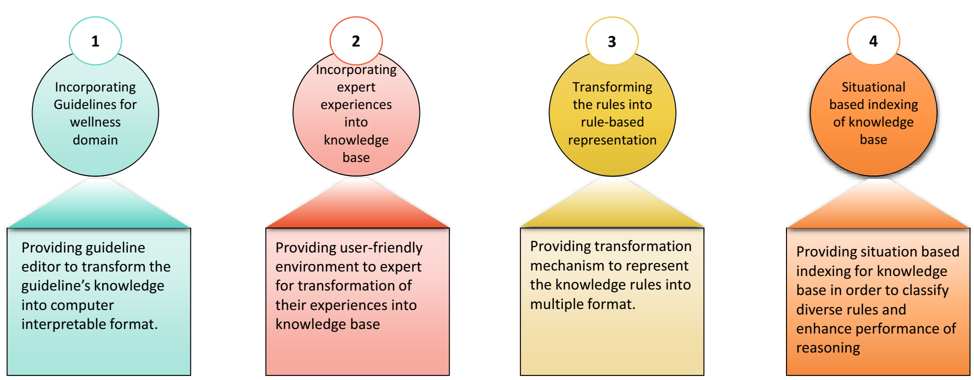
Goal and Objectives

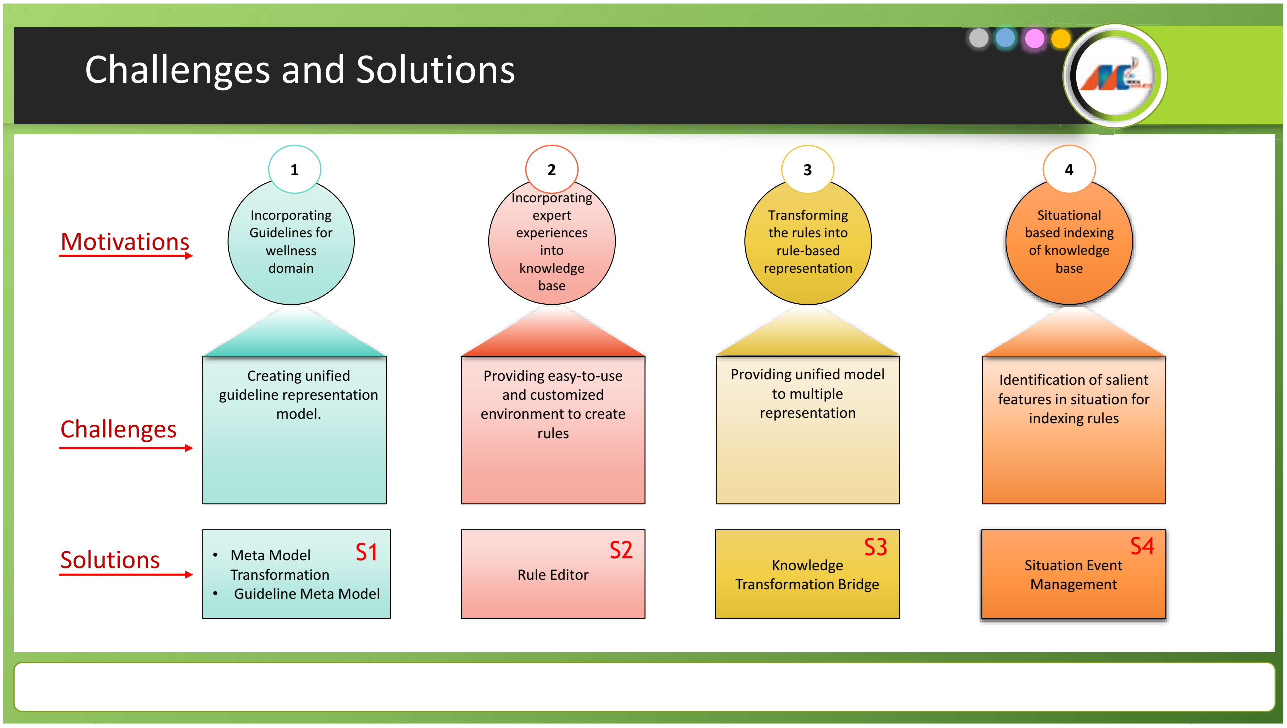
- Goal
  - Provide user-friendly environment to create recommendation's and alert's guidelines to transform experts knowledge into knowledge base
- Objectives
  - ❖ Create easy-to-use Rule Editor to facilitate the domain experts to create knowledge rules using contextual selection of concepts from Intelli-sense 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



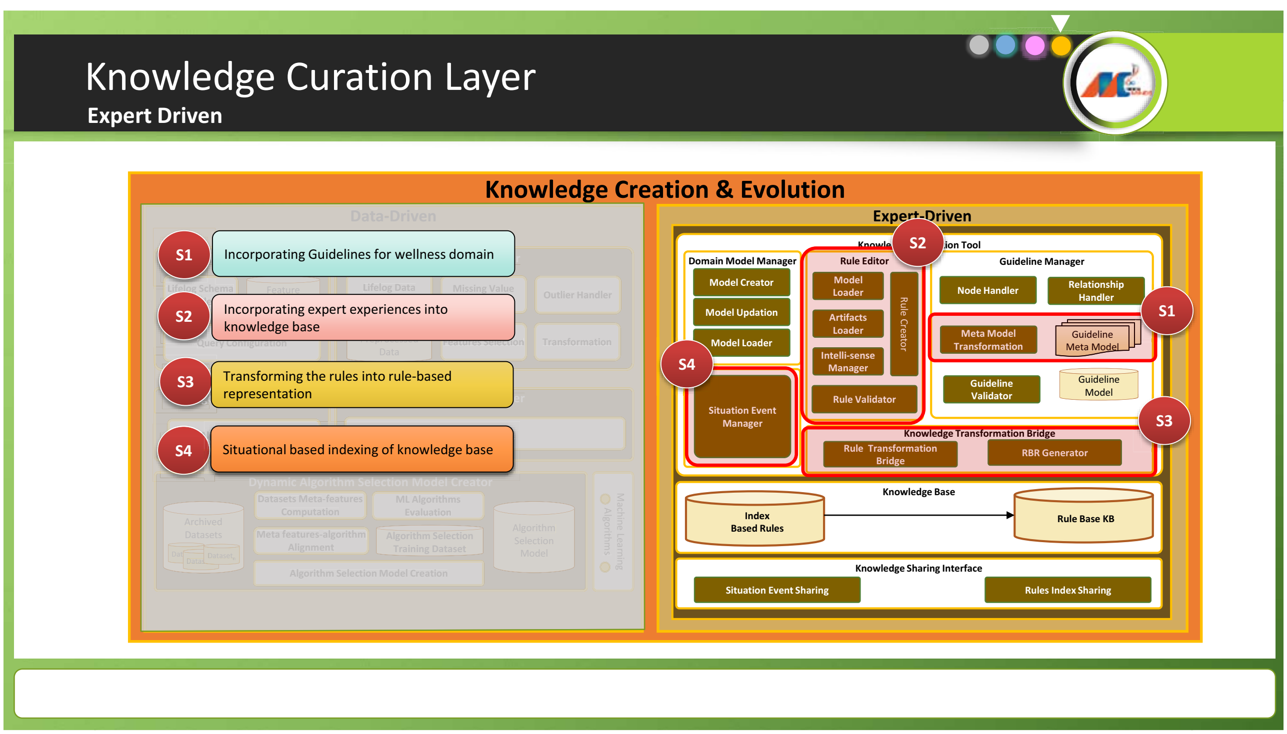
경희대학교 KYUNG HEE UNIVERSITY

Motivations





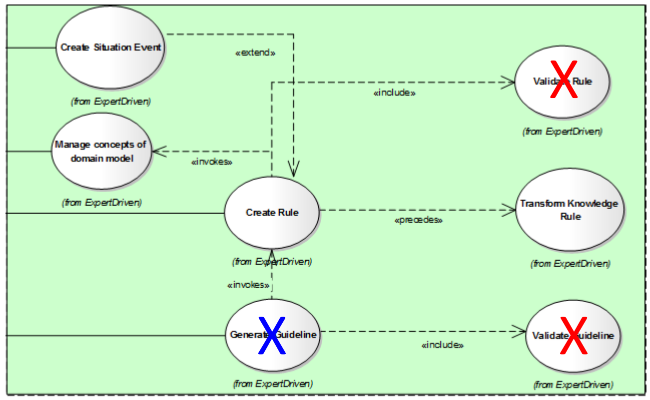
Challenges and Solutions
Motivations
Challenges
Solutions
1
Incorporating Guidelines for wellness domain
2
Incorporating expert experiences into knowledge base
3
Transforming the rules into rule-based representation
4
Situational based indexing of knowledge base
Creating unified guideline representation 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
• Meta Model Transformation
• Guideline Meta Model
S1
Rule Editor
S2
Knowledge Transformation Bridge
S3
Situation Event Management
S4



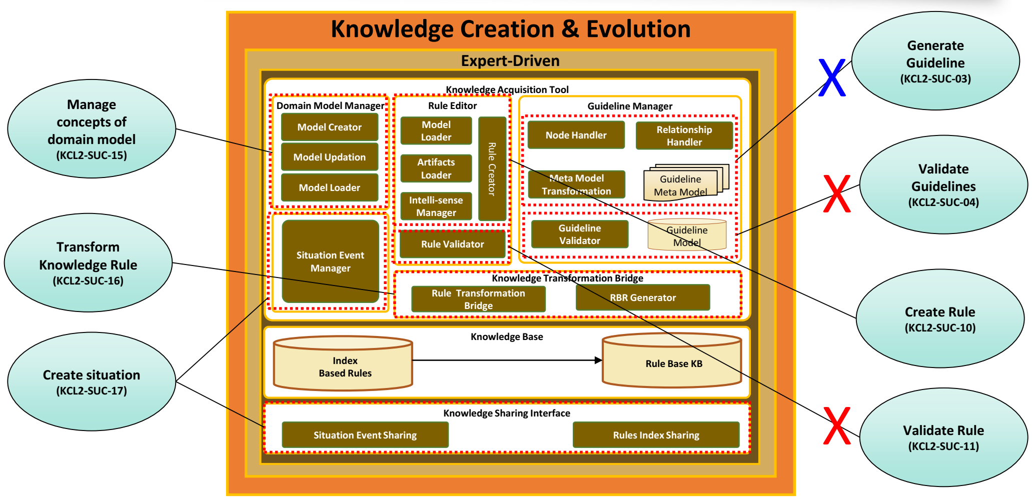
Knowledge Curation Layer
Expert Driven
Knowledge Creation & Evolution
Data-Driven
Expert-Driven
S1
Incorporating Guidelines for wellness domain
S2
Incorporating expert experiences into knowledge base
S3
Transforming the rules into rule-based representation
S4
Situational based indexing of knowledge base
Domain Model Manager
Model Creator
Model Updation
Model Loader
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 Knowledge Acquisition-Use cases

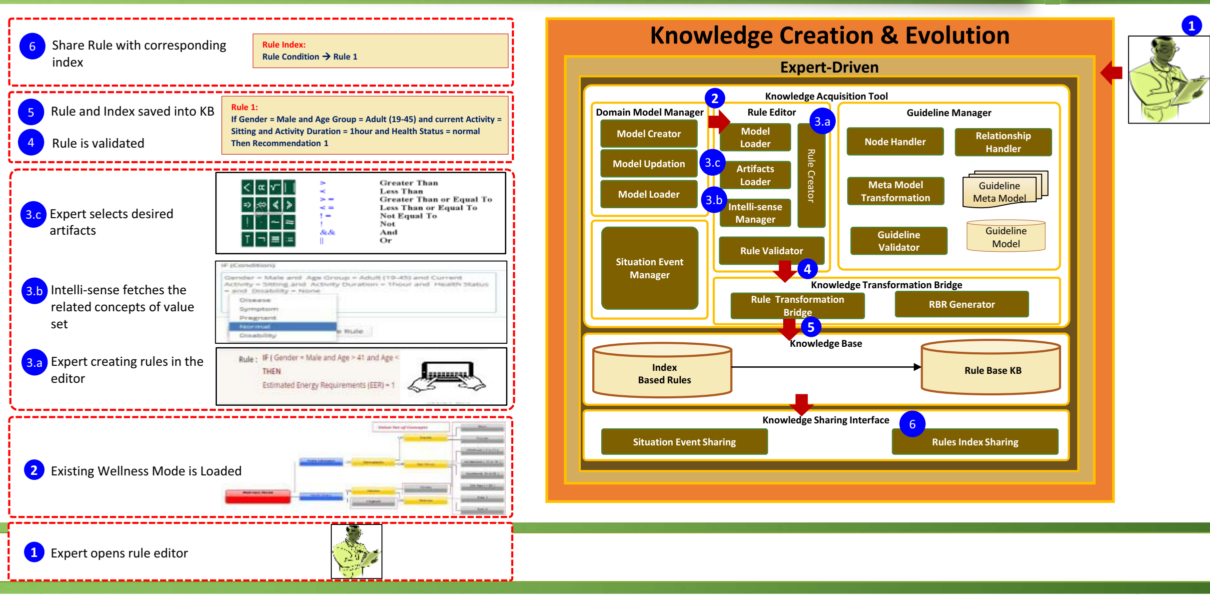
Use Case #ID	Description
KCL2-SUC-03	Generate Guideline
KCL2-SUC-04	Validate Guideline
KCL2-SUC-10	Create Rule
KCL2-SUC-11	Validate Rule
KCL2-SUC-15	Manage Concepts of Domain Model
KCL2-SUC-16	Transform Knowledge Rule
KCL2-SUC-17	Create Situation Event



Expert Driven Knowledge Acquisition-Use cases



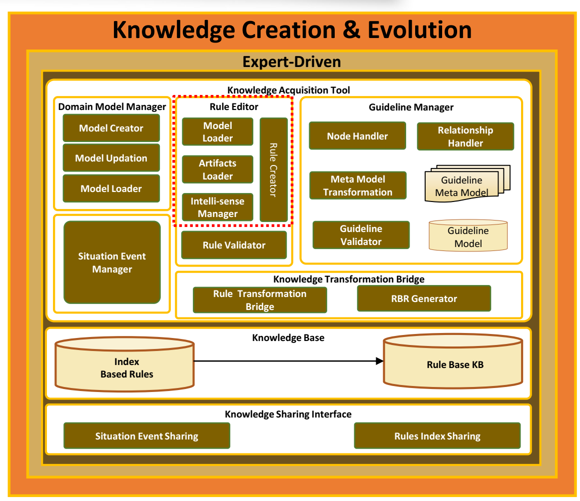
Expert Driven Knowledge Scenario for #2 path



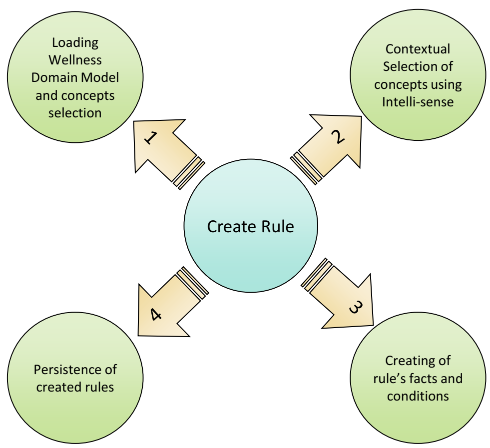
Methodology: Rule Editor

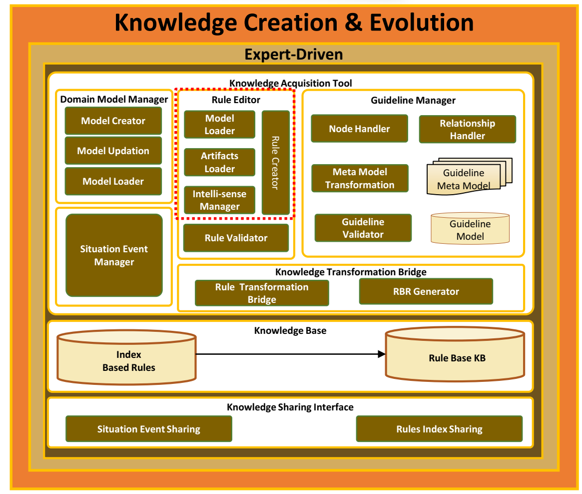
Use-case (KCL2-SUC-10): *Create Rule*

- **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



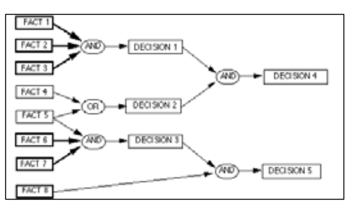
Use Case Functions: Rule Editor





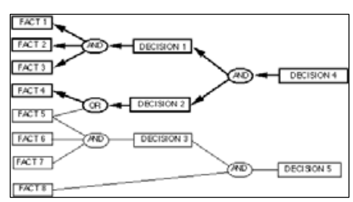
Rule Reasoning Strategies

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



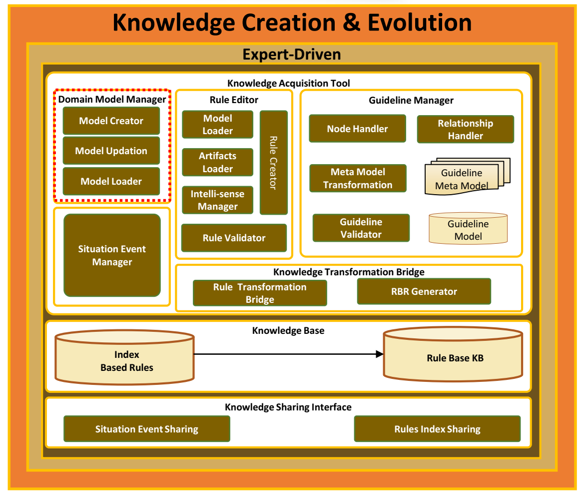
Features
Works from the conclusion to facts (Goal Driven)
Appropriate when the goal is given in the problem statement
Aims for finding necessary data
Appropriate for Diagnostic, prescription and debugging applications
Depth-first search

We will use Forward Chaining

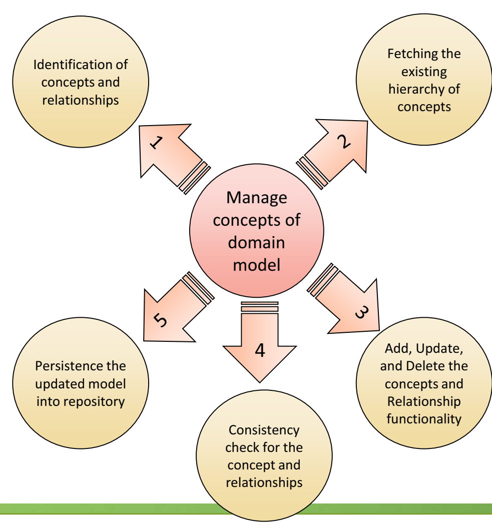
Methodology: Domain Model Manager

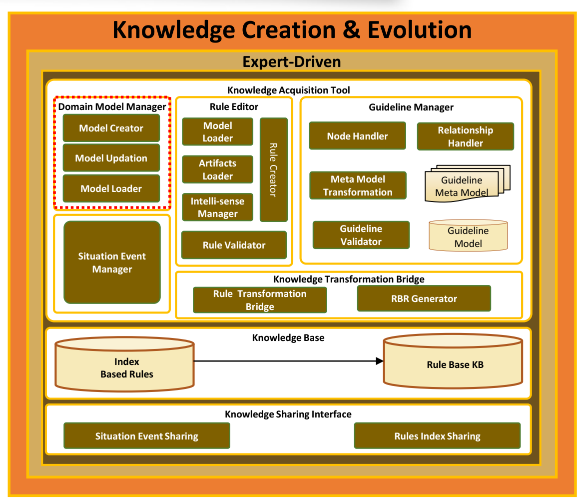
Use-case (KCL2-SUC-15): *Manage concepts of domain model*

- **Objective**
  - ➢ Provides user interface to manage wellness domain model with easy manner
- **Methodology**
  - ➢ *Approach*: Model View Controller (MVC)
    - ➢ *Client side script: AngularJS*
    - ➢ *Server side script: Spring MVC*
  - ➢ ***Domain Model Manager*** *(operations)*
    1. **Input**: Concepts and relationships
    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

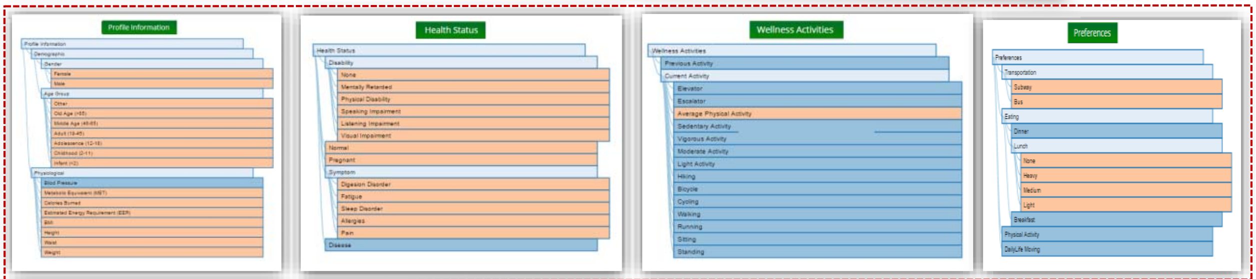


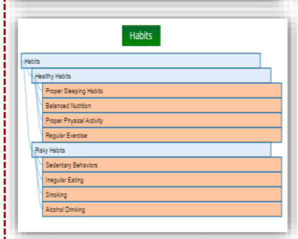
Use case functions: Domain Model Manager

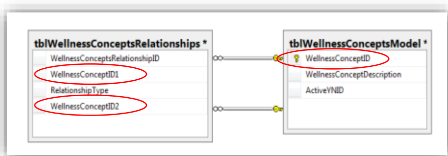




Existing Wellness Model and Schema





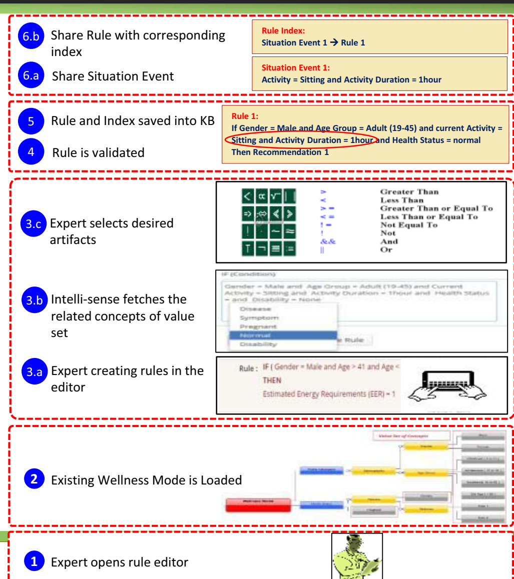


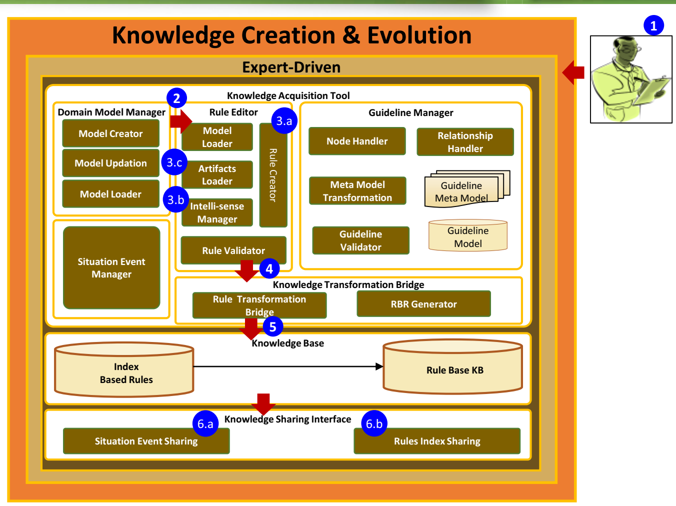
WellnessConceptID	WellnessConceptDescription	ActiveYNID
54	Risky Habits	Yest
55	Alcohol Drinking	Yes
56	Smoking	Yes

WellnessConcept RelationshipID	WellnessConceptID 1	Relationship Type	WellnessConceptID 2
60	55 (Alcohol Drinking)	IsA	54 (Risky Habits)
61	56 (Smoking)	IsA	54 (Risky Habits)

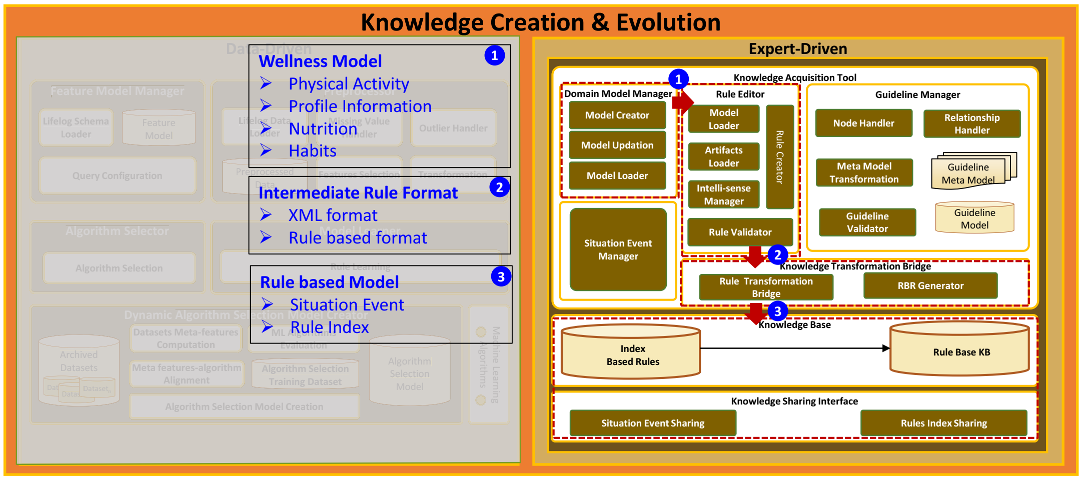
Expert Driven Knowledge Scenario for #2 path

Situation Event Base





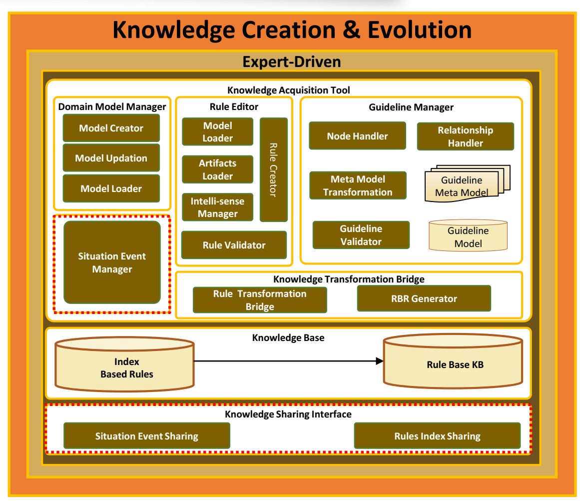
Expert Driven Rule Creation Workflow for #2 path



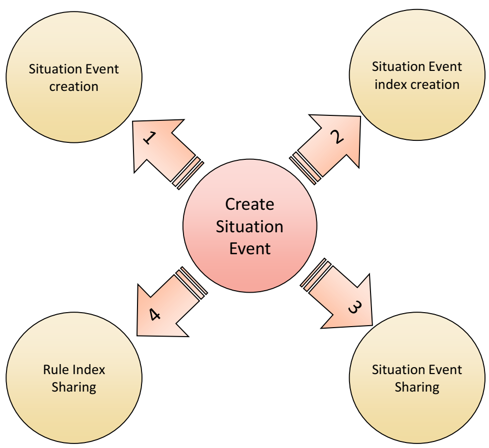
Methodology: Situation Event Manager

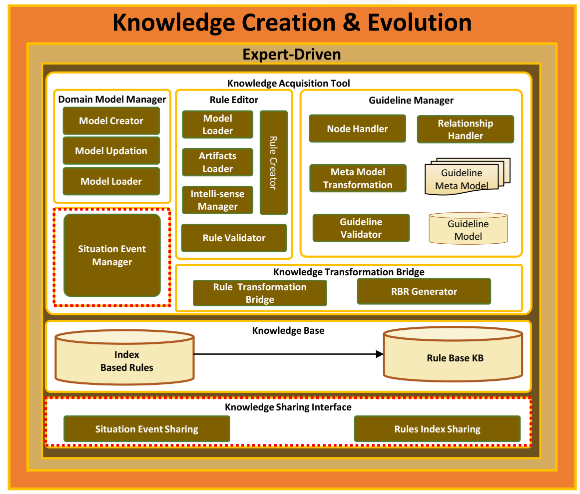
Use-case (KCL2-SUC-15): *Create Situation Event*

- **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

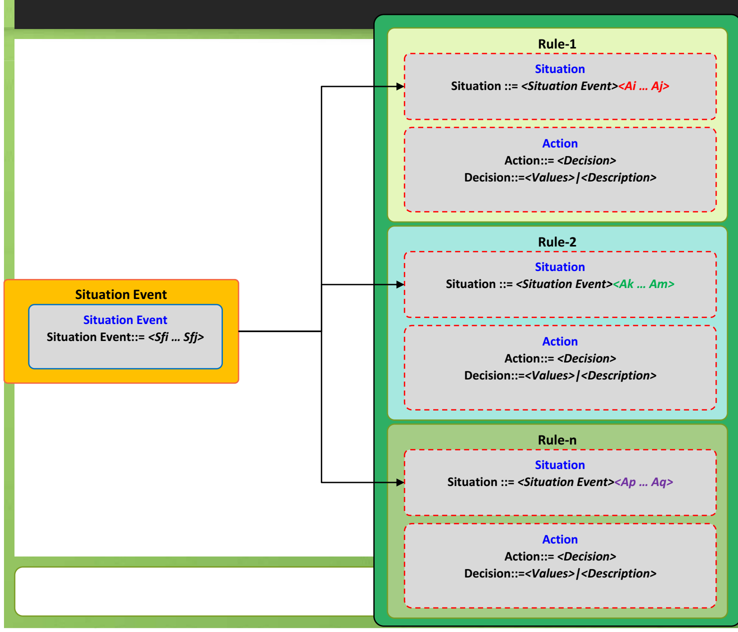


Use case functions: Situation Event Manager and Knowledge Sharing Interface

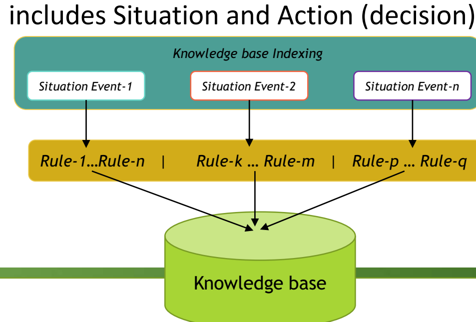


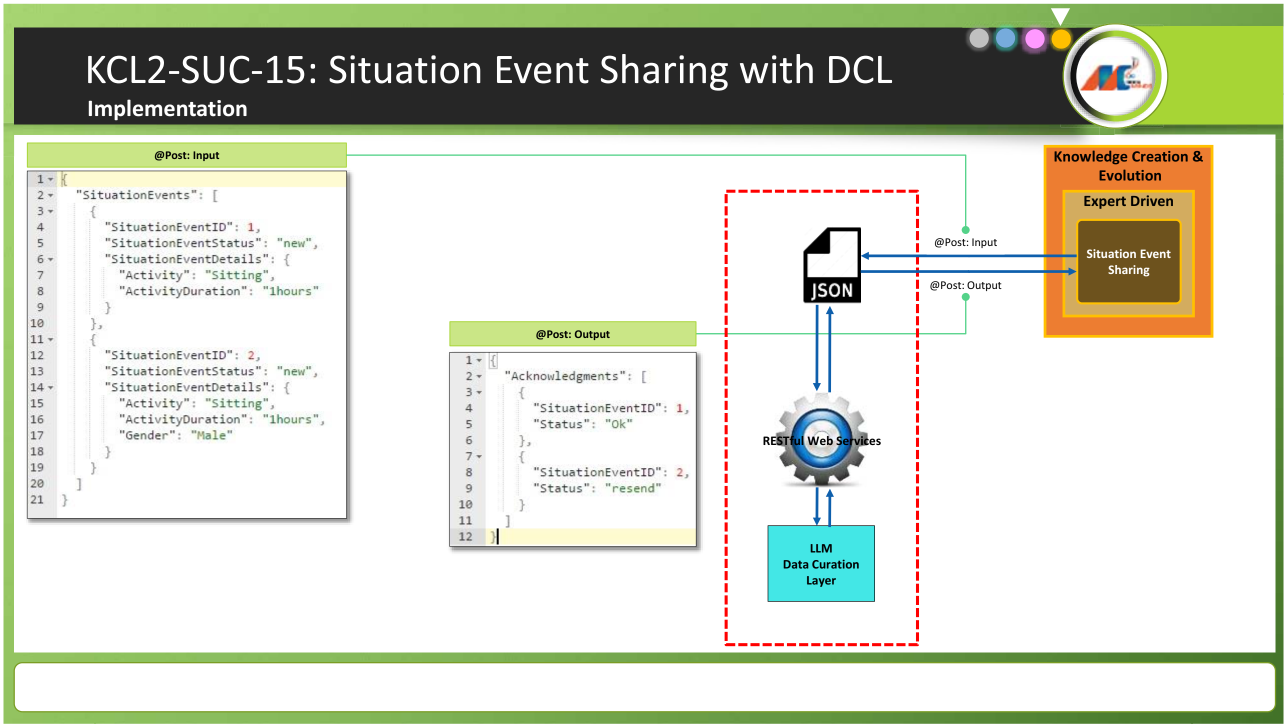


Situation, Situation Event and Rule

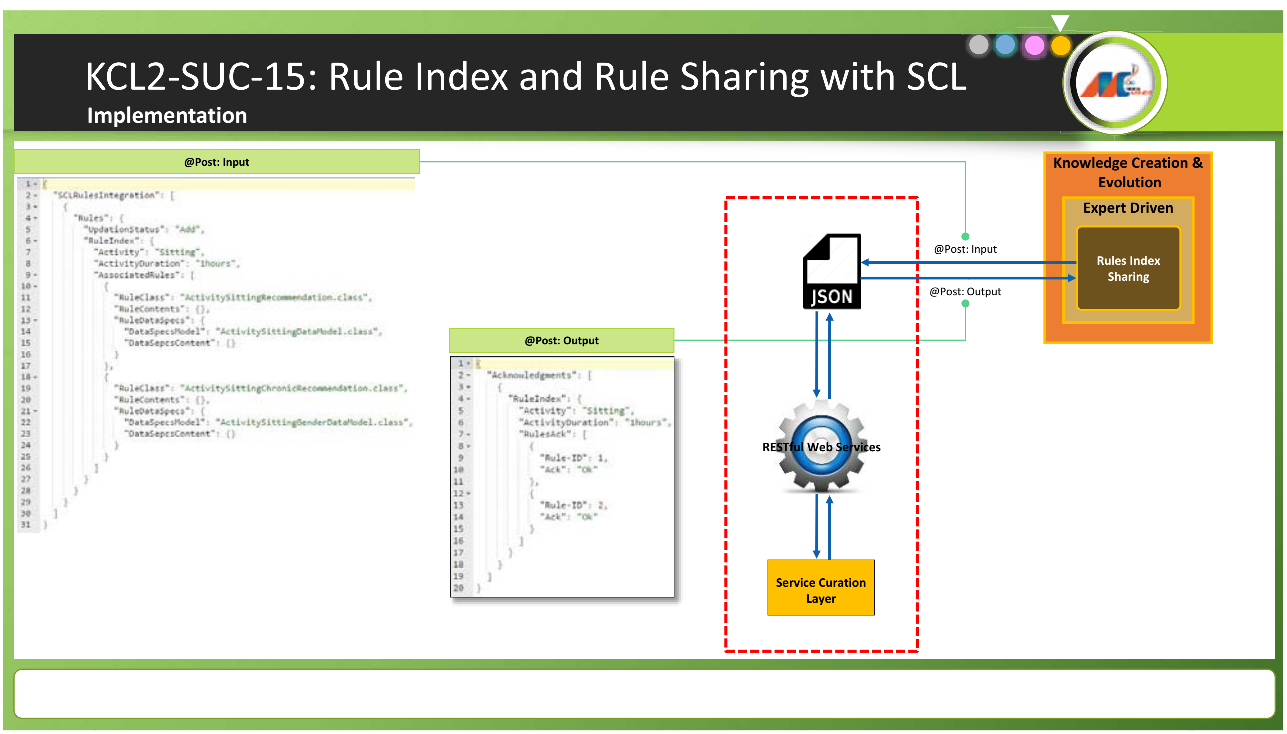


- *Situation* constitute part of rule which includes data elements(condition) required to process rule.
- *Situation Event* is part of *Situation* which includes *salient features* of data elements(condition).
- *Rule* is part of knowledge base which includes Situation and Action (decision)





KCL2-SUC-15: Situation Event Sharing with DCL
Implementation
@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

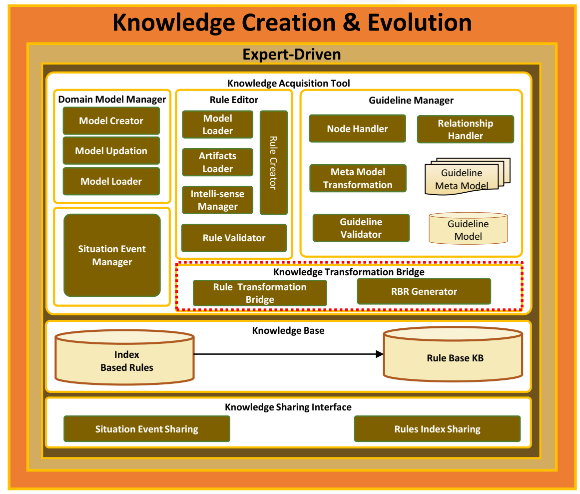


KCL2-SUC-15: Rule Index and Rule Sharing with SCL
Implementation
@Post: Input
{
"SCLRulesIntegration": [
{
"Rules": {
"UpdationStatus": "Add",
"RuleIndex": {
"Activity": "Sitting",
"ActivityDuration": "1hours",
"AssociatedRules": [
{
"RuleClass": "ActivitySittingRecommendation.class",
"RuleContents": {},
"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

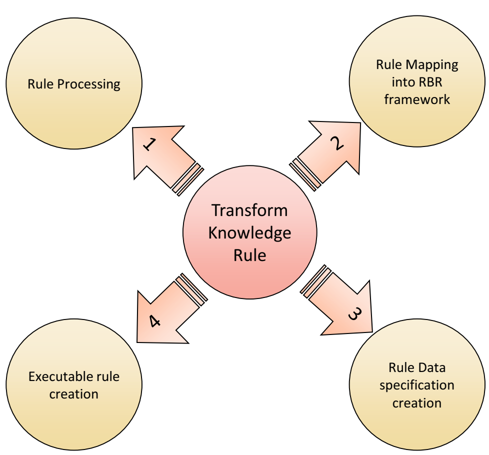
Methodology: Knowledge Transformation Bridge

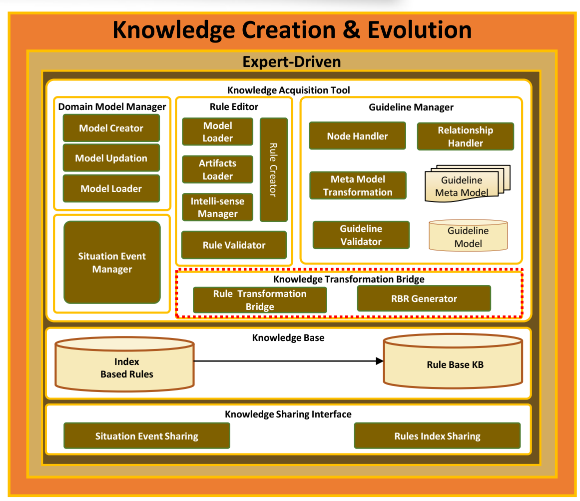
Use-case (KCL2-SUC-15): *Transform knowledge Rule*

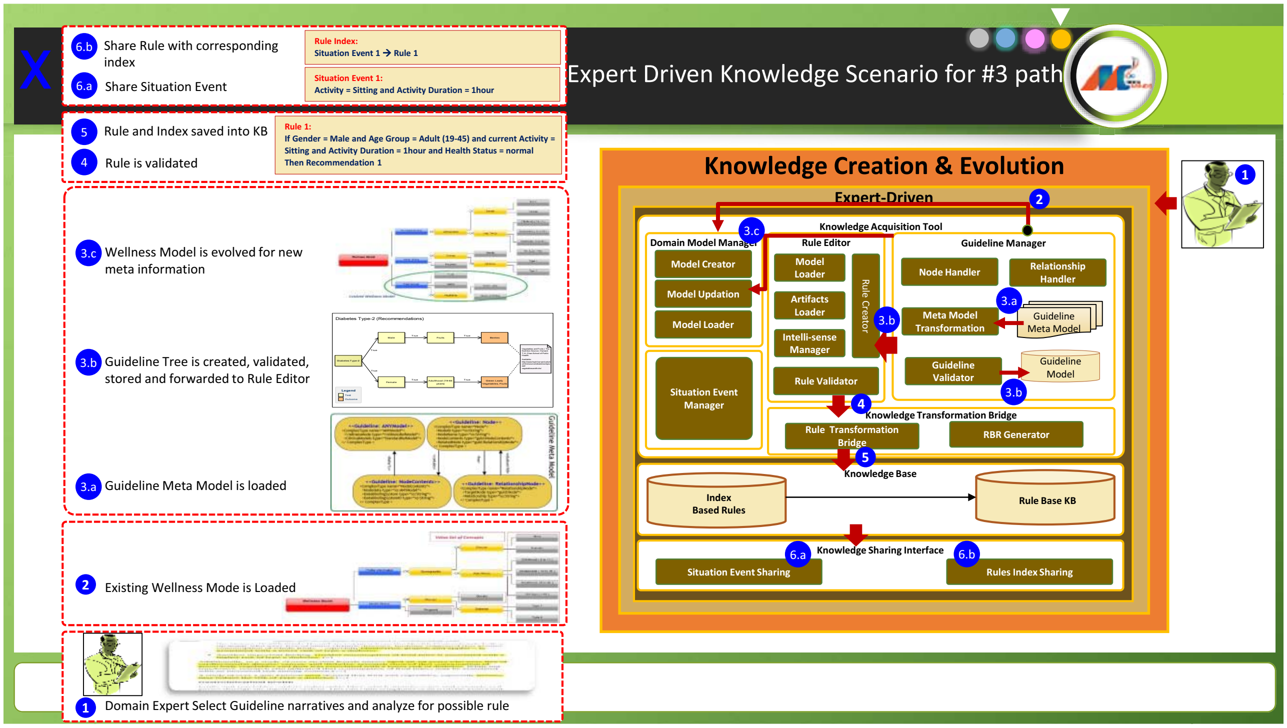
- **Objective**
  - Transform the created rules and guidelines into executable format to generate recommendations
- **Methodology**
  - *Approach*: Template based code generations
    - *Server side script: Core Java*
  - ***Knowledge Transformation Bridge** (operations)*
    1. **Input**: Plain rule 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

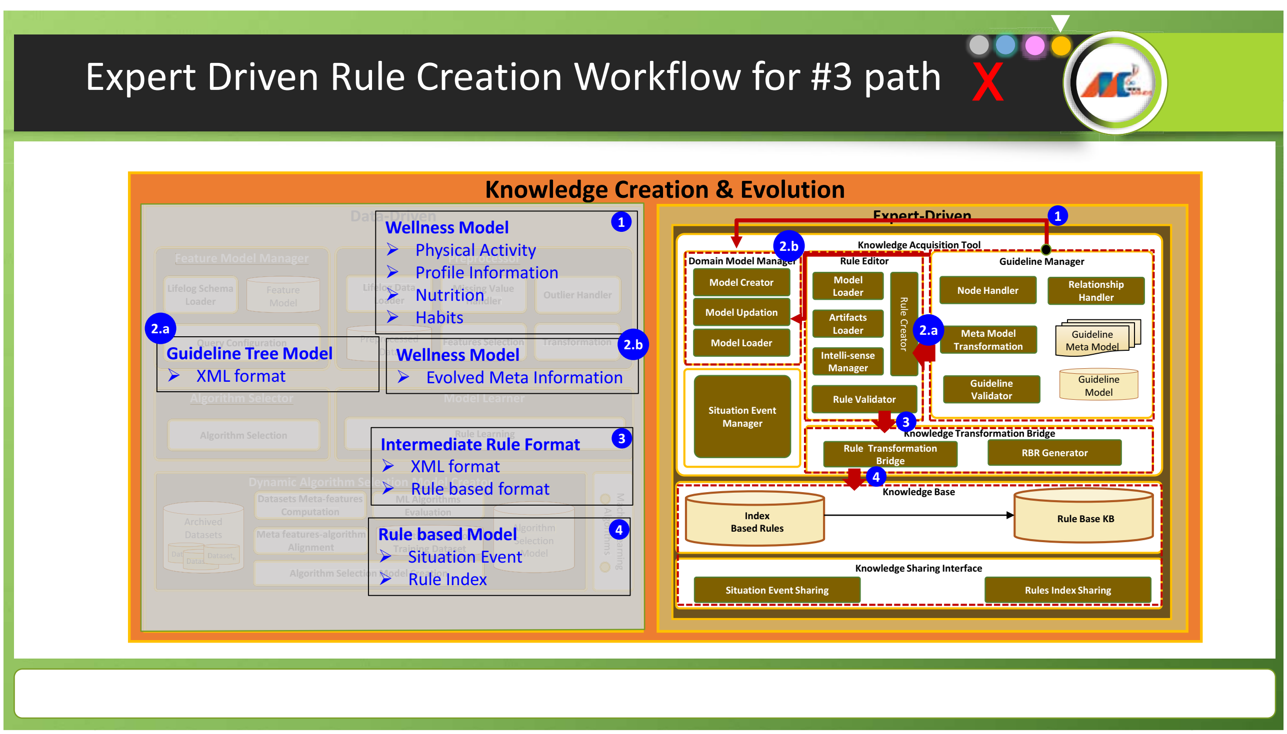


Use case functions: Knowledge Transformation Bridge






Expert Driven Knowledge Scenario for #3 path
6.b Share Rule with corresponding index
Rule Index:
Situation Event 1 → Rule 1
6.a Share Situation Event
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
3.a Guideline Meta Model is loaded
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

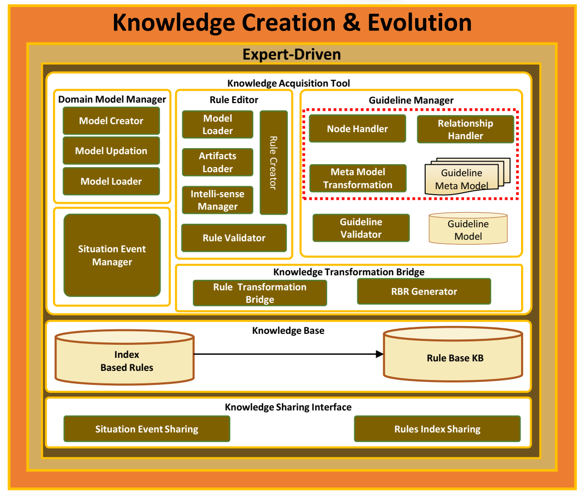

Expert Driven Rule Creation Workflow for #3 path
Knowledge Creation & Evolution
Data-Driven
Wellness Model
Physical Activity
Profile Information
Nutrition
Habits
Guideline Tree Model
XML format
Wellness Model
Evolved Meta Information
Intermediate Rule Format
XML format
Rule based format
Rule based Model
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

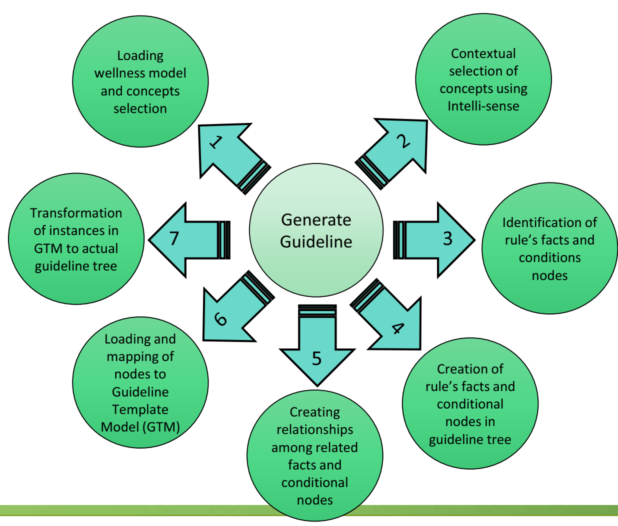
Use-case (KCL2-SUC-03): *Generate Guideline*

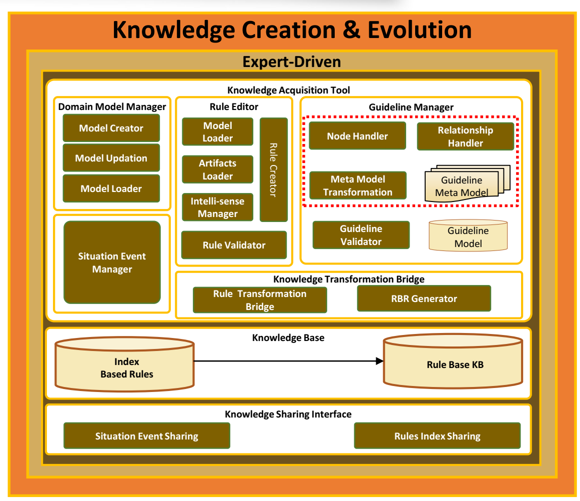
- **Objective**
  - Generate domain guidelines in tree format with easy manner
- **Methodology**
  - *Approach*: Model View Controller (MVC*)*
    - *Client side script: AngularJS*
    - *Server side script: Spring MVC*
  - ***Guideline Manager*** *(operations)*

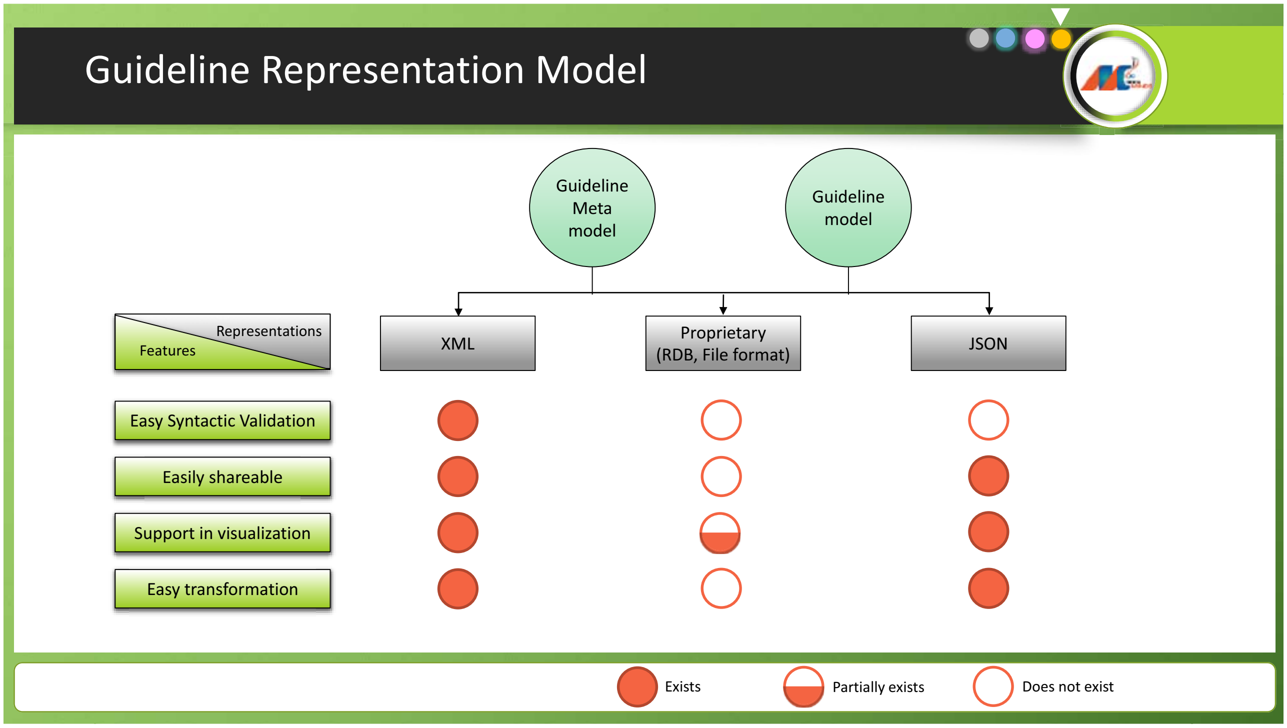
1. **Input**: Concepts from wellness model, Expert's/Guideline knowledge
2. **Processing**:
   a. Domain expert generate tree nodes using Intelli-sense window and wellness model
   b. Make relationships among the nodes according to expert's knowledge
   c. Load guideline meta model and map the created 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)

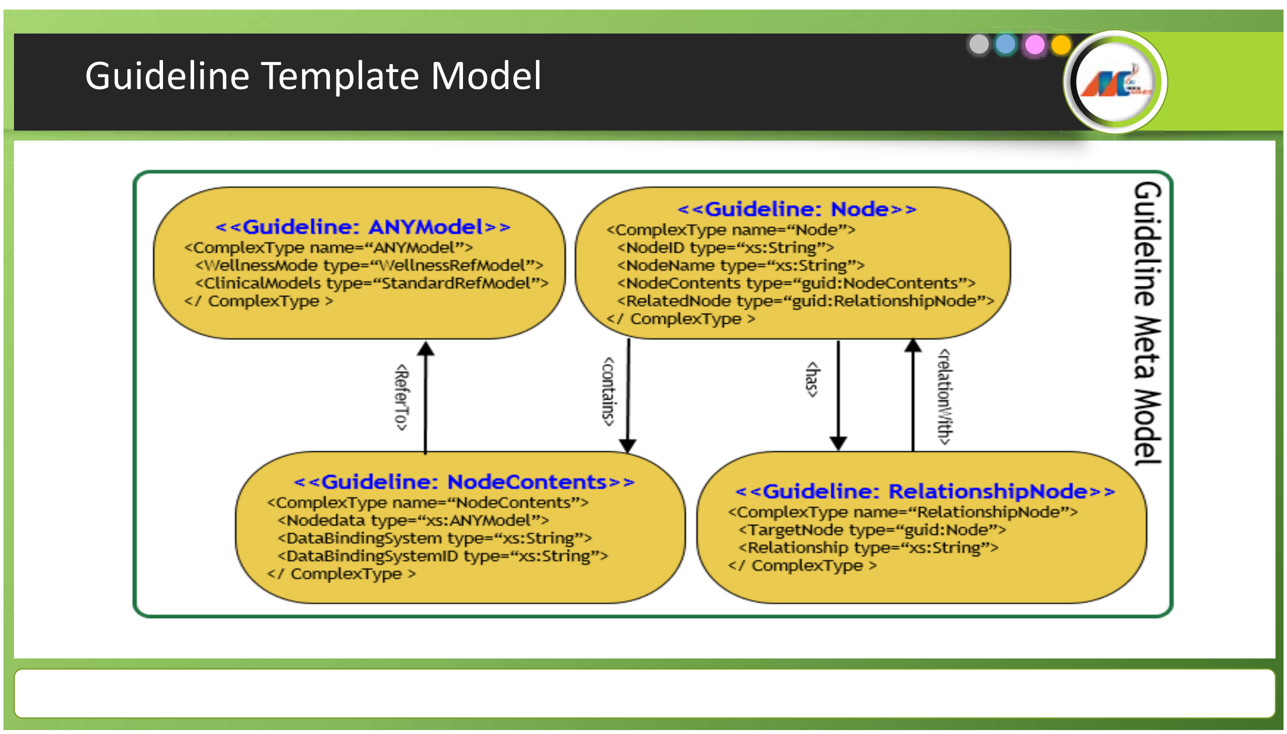


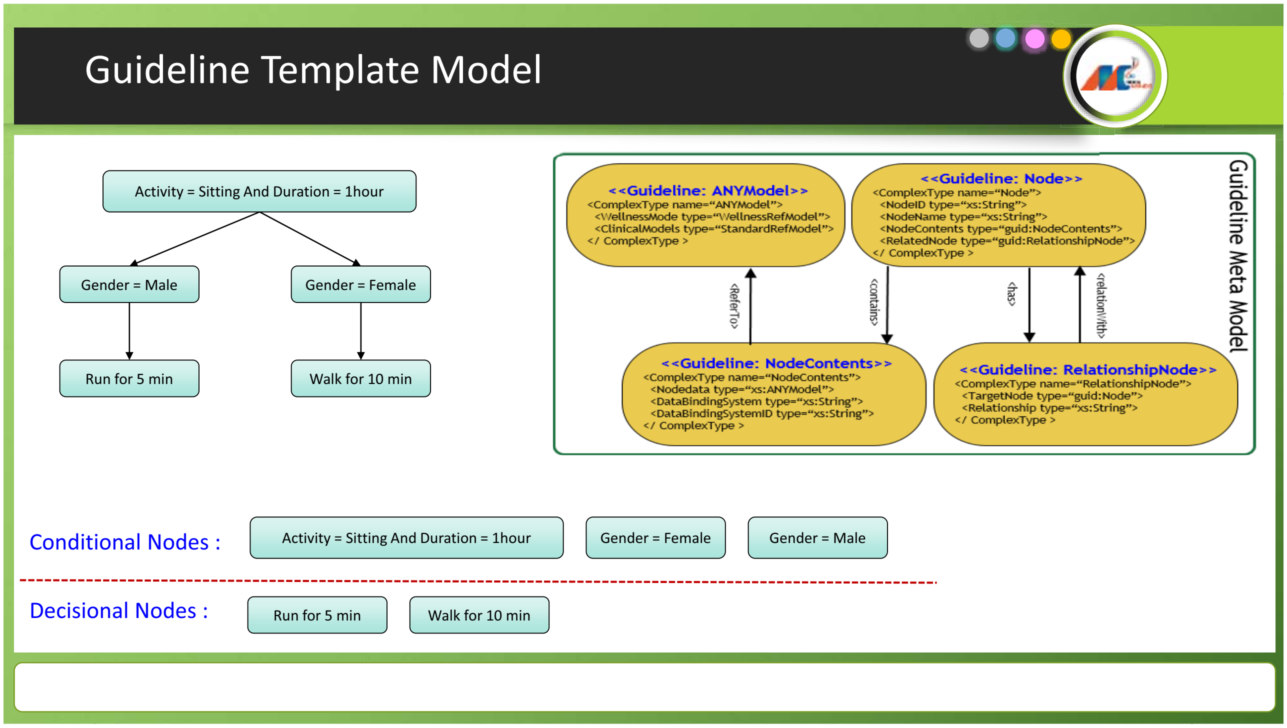
Use case functions: Guideline Manager

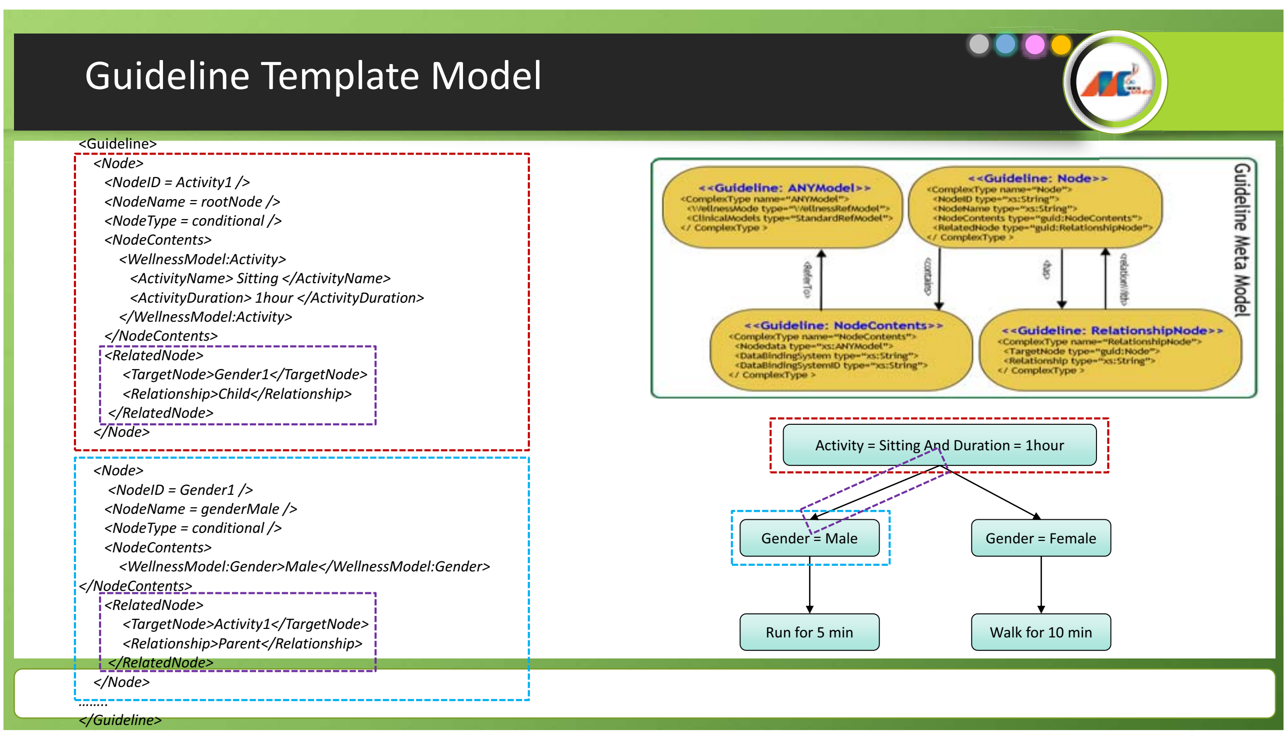





Guideline Representation Model
Guideline Meta model
Guideline model
Representations
Features
XML
Proprietary (RDB, File format)
JSON
Easy Syntactic Validation
Easily shareable
Support in visualization
Easy transformation
Exists
Partially exists
Does not exist


Guideline Template Model
<<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
Activity = Sitting And Duration = 1hour
Gender = Male
Gender = Female
Run for 5 min
Walk for 10 min
<<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
Conditional Nodes :
Activity = Sitting And Duration = 1hour
Gender = Female
Gender = Male
Decisional Nodes :
Run for 5 min
Walk for 10 min


Guideline Template Model
<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>
Guideline Meta Model
Activity = Sitting And Duration = 1hour
Gender = Male
Gender = Female
Run for 5 min
Walk for 10 min

Contributions

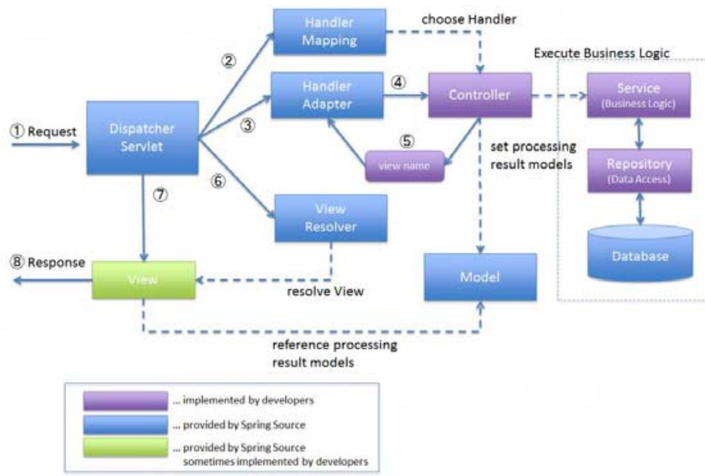
- Provide user-friendly Rule Editor to domain experts
- Provide easy-to-use Guideline Editor to transform the guidelines from text to computer executable guidelines and rules
- Provide uniform Guideline Template Model (GTM)
- Provide wellness model manager to create wellness model by experts
- Index based rules (Situation enabled) generation and sharing for different services

Selected Tools and Techniques

- 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



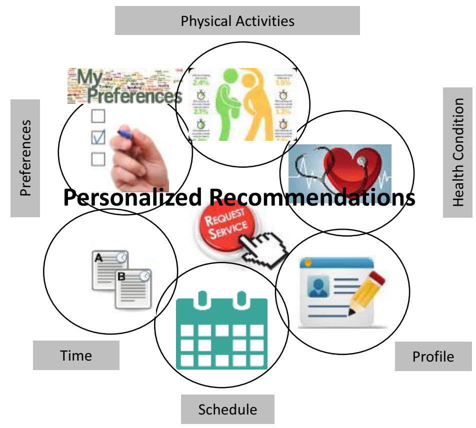
http://terasolunaorg.github.io/

Service Curation Layer (SCL 2.0)

Principal Investigator:
Professor Sungyoung Lee

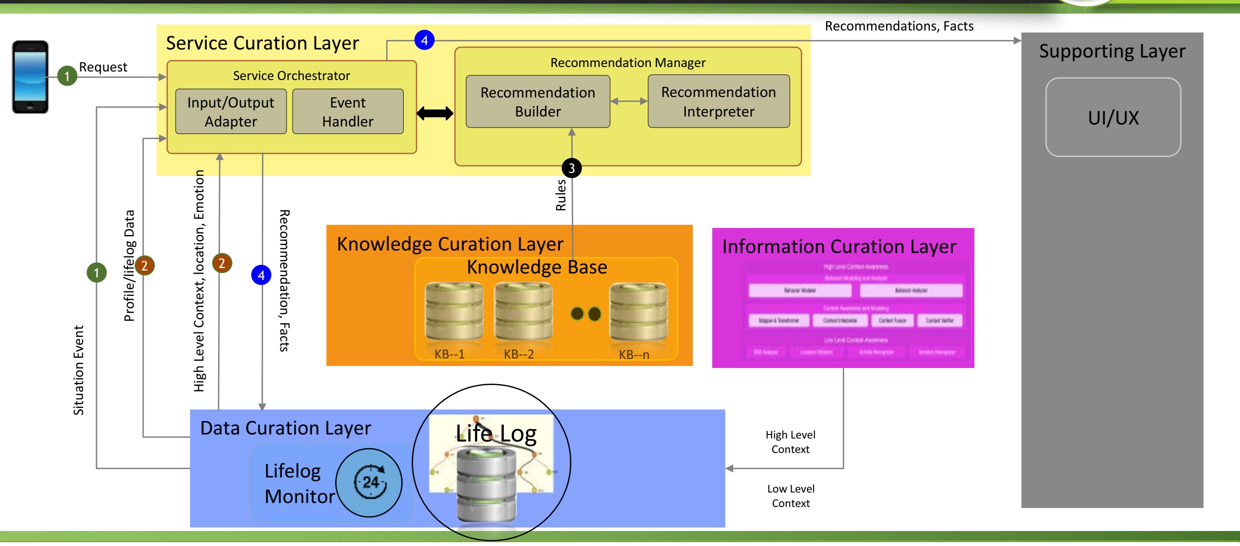
Goal and Objectives

- 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.



경희대학교 KYUNG HEE UNIVERSITY

SCL v2.0 Functional Diagram

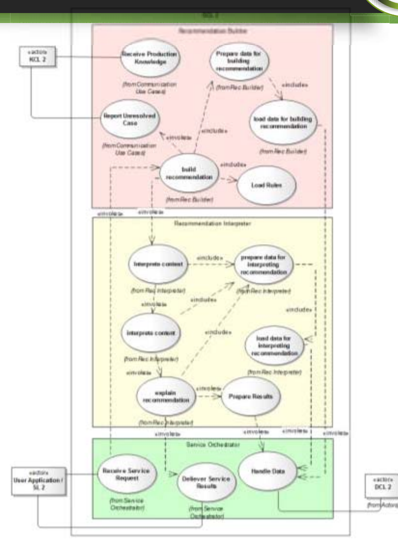


SCL Components and Req Specifications

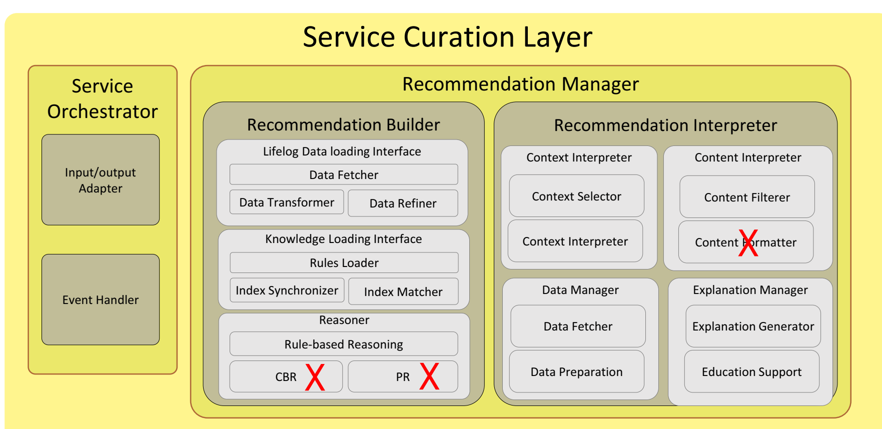
Recommendation Builder — Rahman Ali

Recommendation Interpreter — Muhammad Afzal

Service Orchestrator — Maqbool Hussain



Service Curation Layer - Architecture

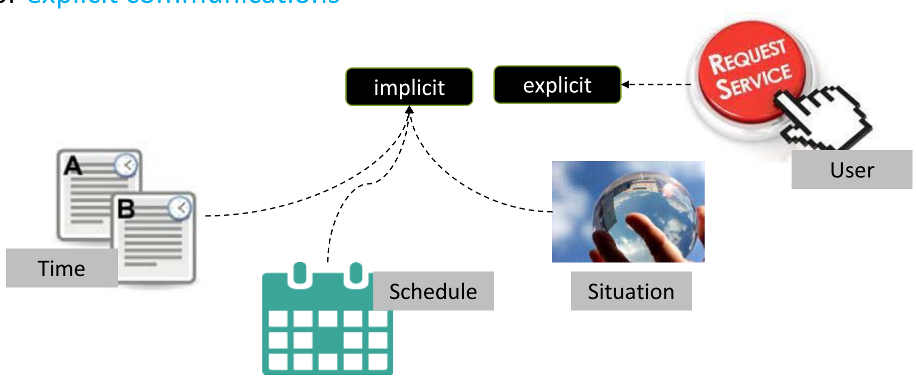


Service Curation Layer (SCL 2.0)

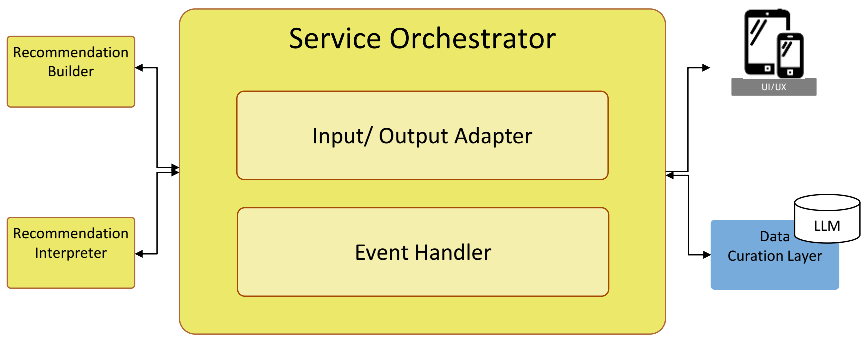
Service Orchestrator

Introduction

- Orchestrator enable communication of SCL with other layers and user application
- It allows implicit or explicit communications

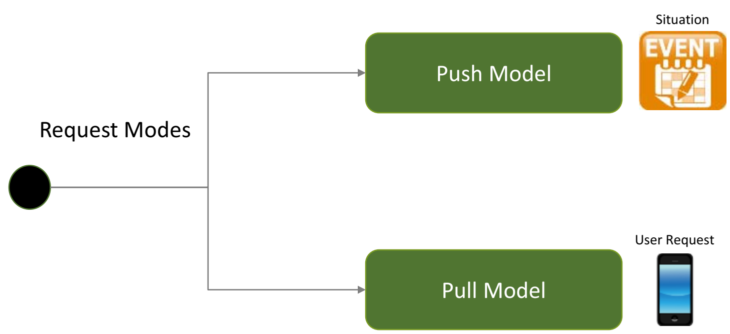


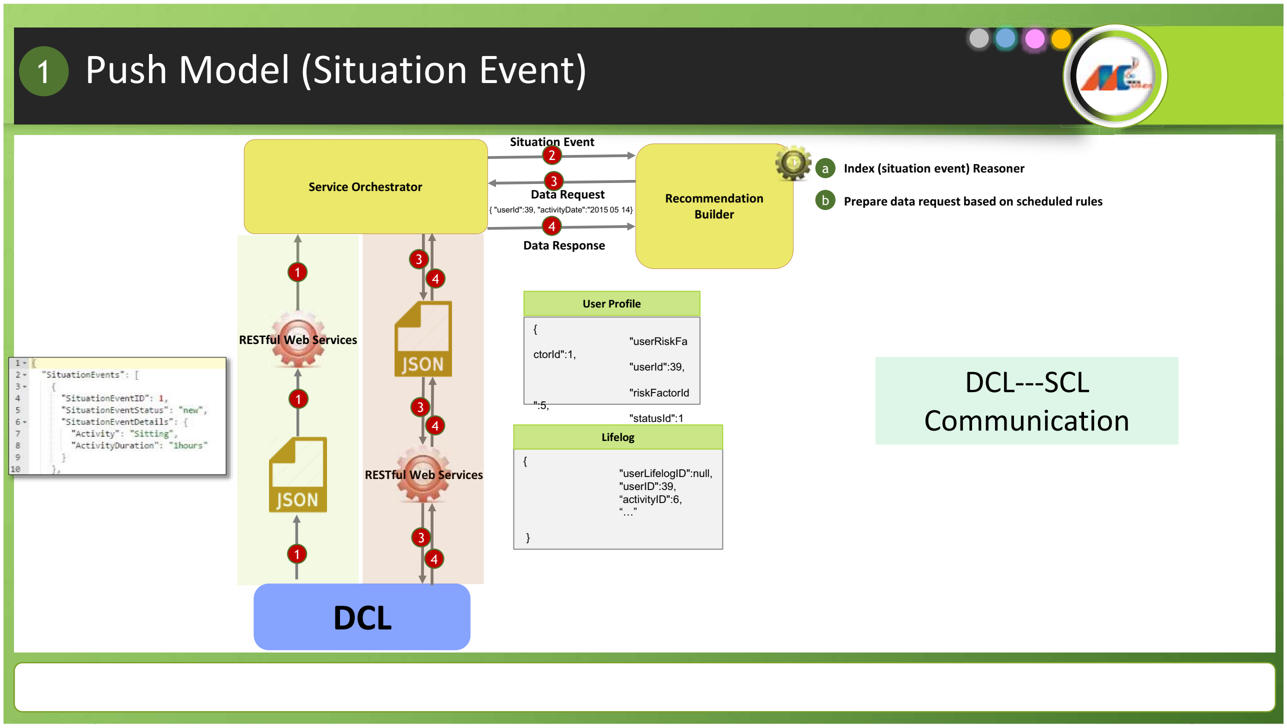
Service Orchestrator - Architecture

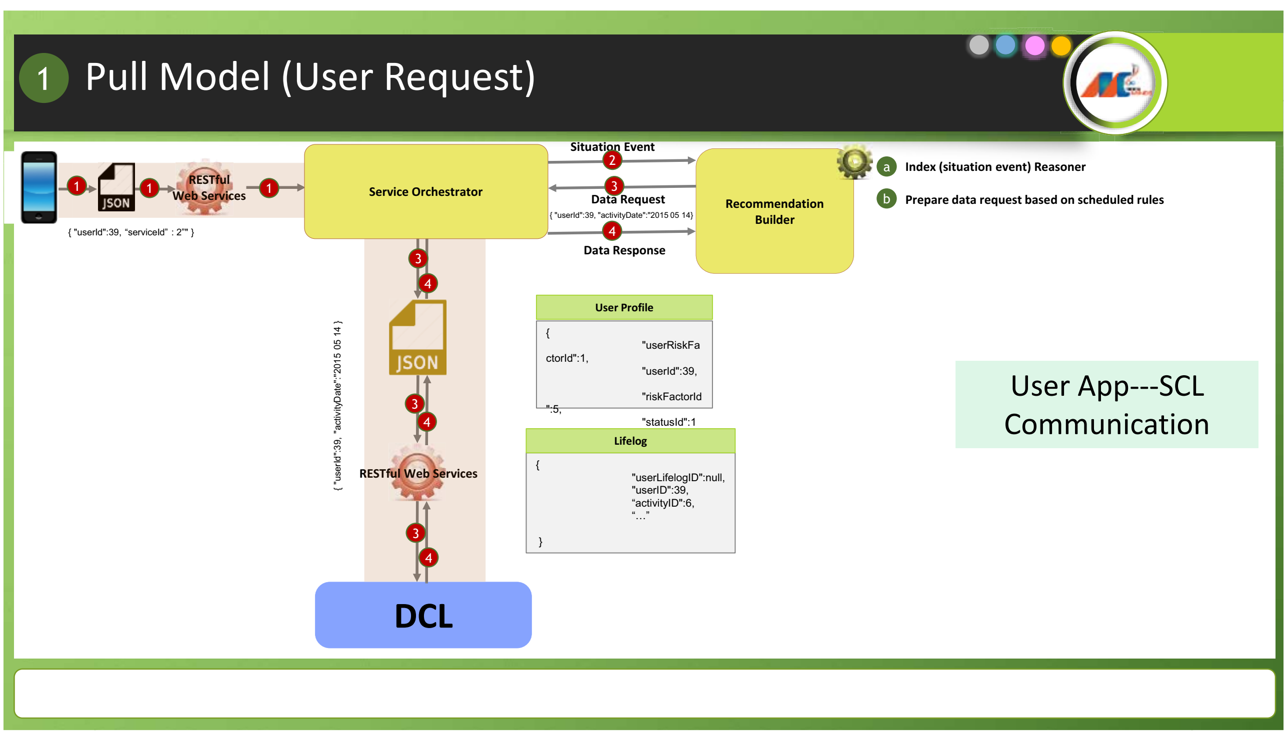


경희대학교
KYUNG HEE UNIVERSITY

Request for Recommendation




1
Push Model (Situation Event)
Service Orchestrator
Situation Event
2
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
1
3
4
User Profile
{ "userRiskFactorId":1, "userId":39, "riskFactorId":5, "statusId":1
Lifelog
{ "userLifelogID":null, "userID":39, "activityID":6, "..." }
DCL---SCL Communication
DCL
"SituationEvents": [
"SituationEventID": 1,
"SituationEventStatus": "new",
"SituationEventDetails": {
"Activity": "Sitting",
"ActivityDuration": "1hours"


1
Pull Model (User Request)
JSON
RESTful Web Services
{ "userId":39, "serviceId" : 2" }
Service Orchestrator
Situation Event
2
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
{ "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---SCL Communication
DCL

Contributions

- 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

Service Curation Layer (SCL 2.0)

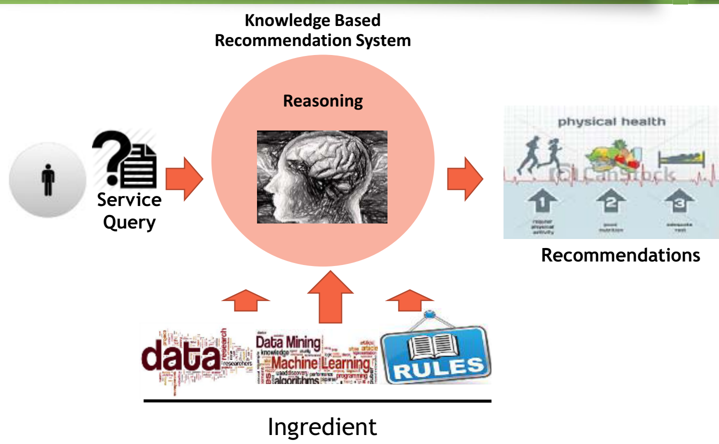
Recommendation Builder

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

Introduction (1/3)

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].

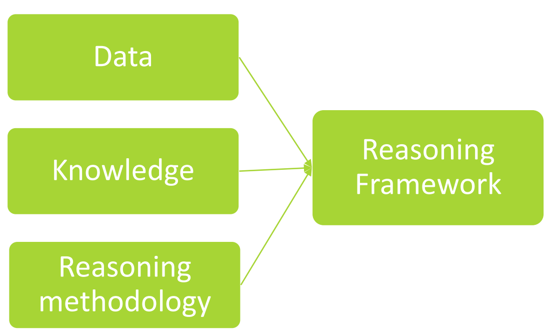


[1] Moses, Yoram; Vardi, Moshe Y; Fagin, Ronald; Halpern, Joseph Y (2003). Reasoning About Knowledge. MIT Press. ISBN 978-0-262-56200-3.

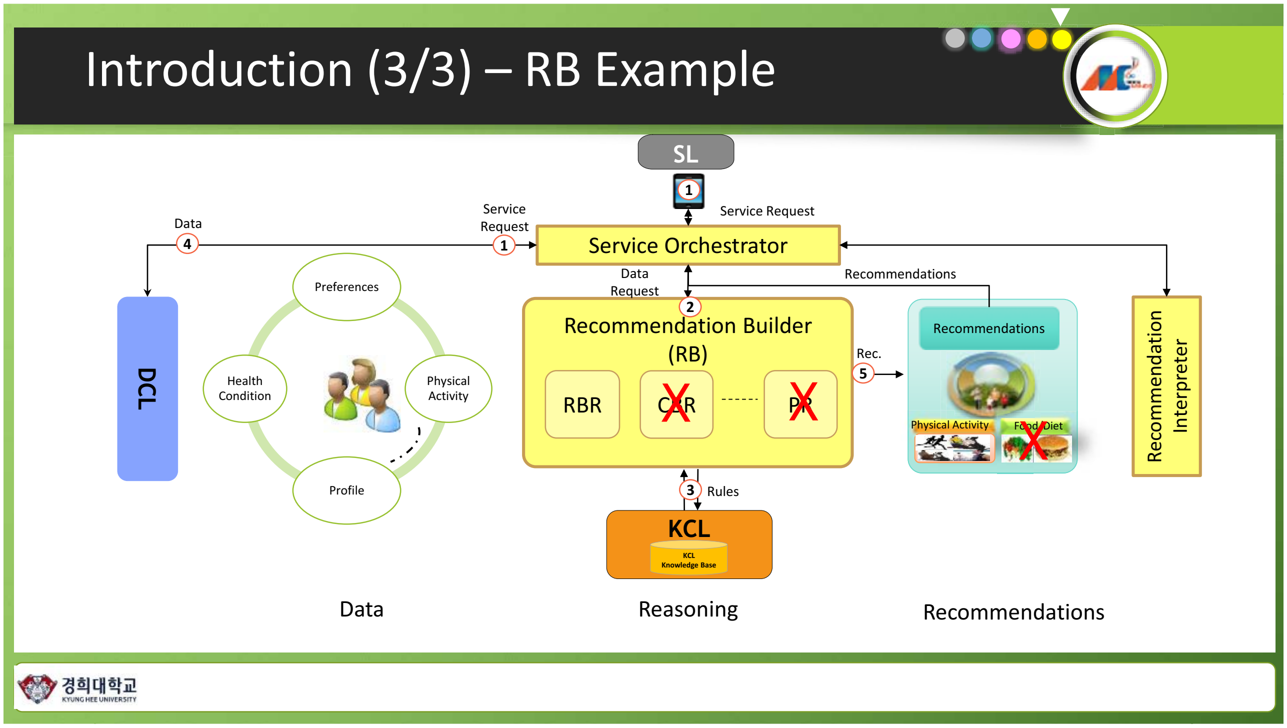
경희대학교 KYUNG HEE UNIVERSITY

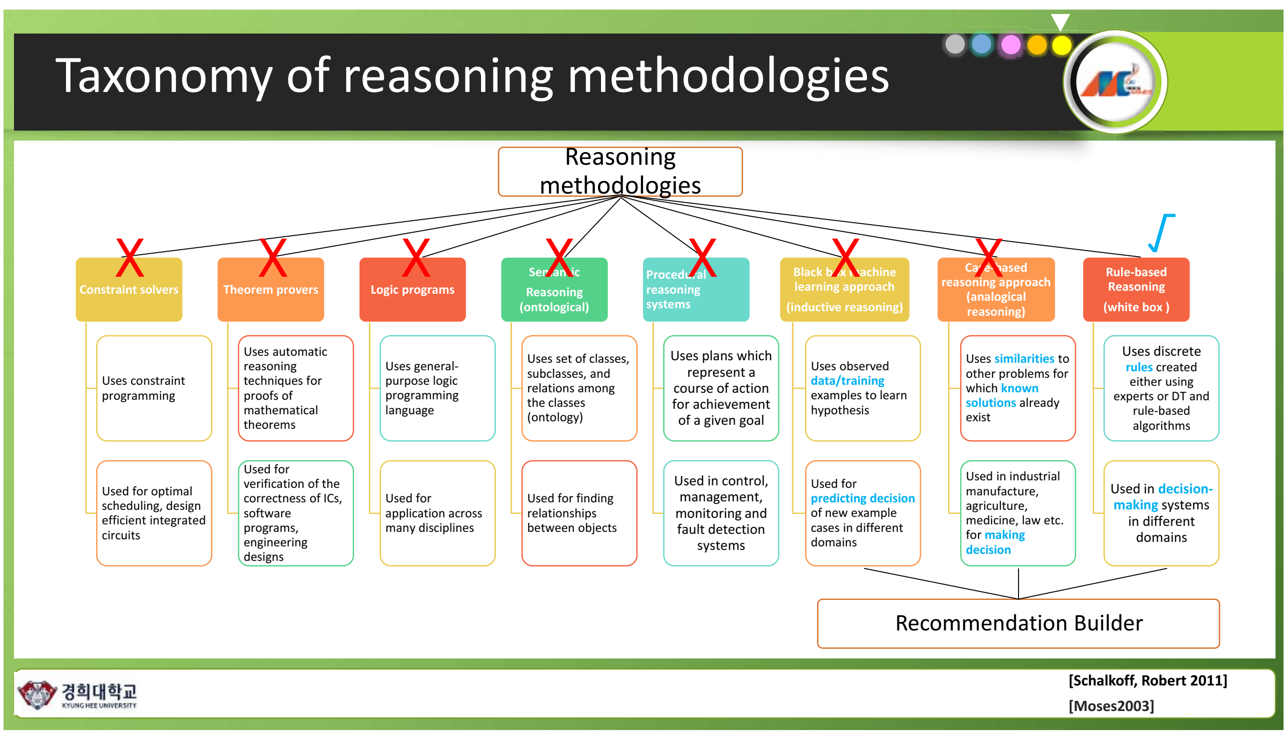
Introduction (2/3)

Ingredients of a Reasoning Framework



경희대학교 KYUNG HEE UNIVERSITY


Introduction (3/3) – RB Example
SL
Service Request
Service Request
Service Orchestrator
Data
Data Request
Recommendations
DCL
Preferences
Health Condition
Physical Activity
Profile
Recommendation Builder (RB)
RBR
CBR
PR
Rec.
Recommendations
Physical Activity
Food Diet
Recommendation Interpreter
Rules
KCL
KCL Knowledge Base
Data
Reasoning
Recommendations
경희대학교
KYUNG HEE UNIVERSITY


Taxonomy of reasoning methodologies
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
경희대학교
KYUNG HEE UNIVERSITY
[Schalkoff, Robert 2011]
[Moses2003]

Related Work

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

- 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

Motivations

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

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)

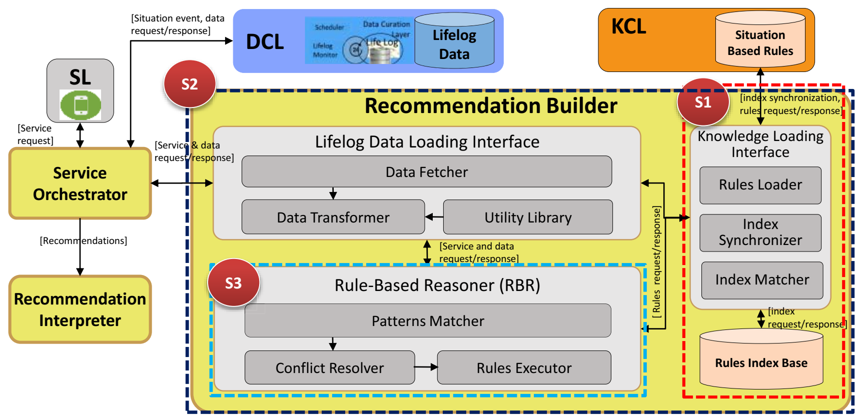
경희대학교 KYUNG HEE UNIVERSITY

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

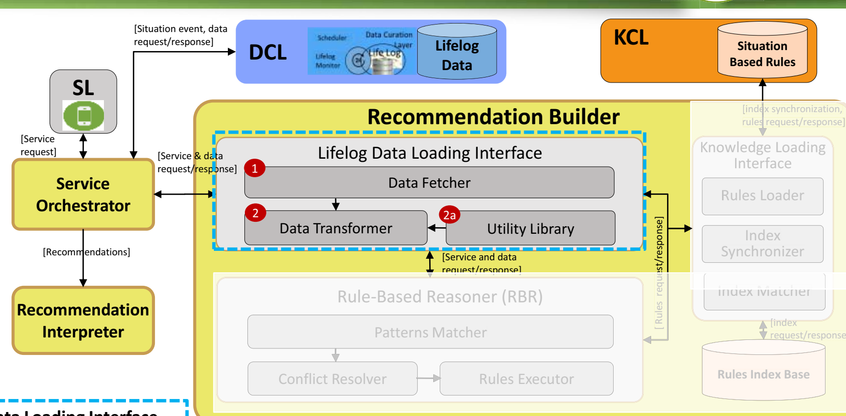
Data Fetcher

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

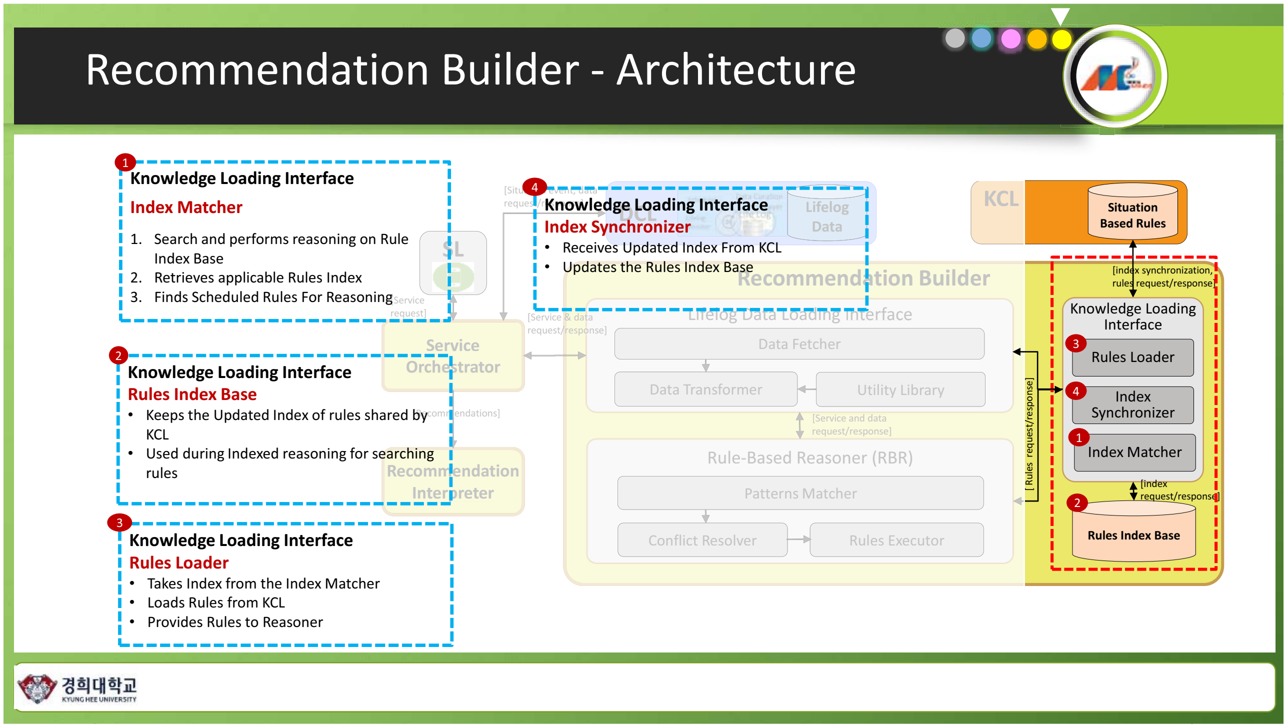
- Parse rules conditions part
- Conformance check of condition attributes and data
- Transformation of data to the condition formats

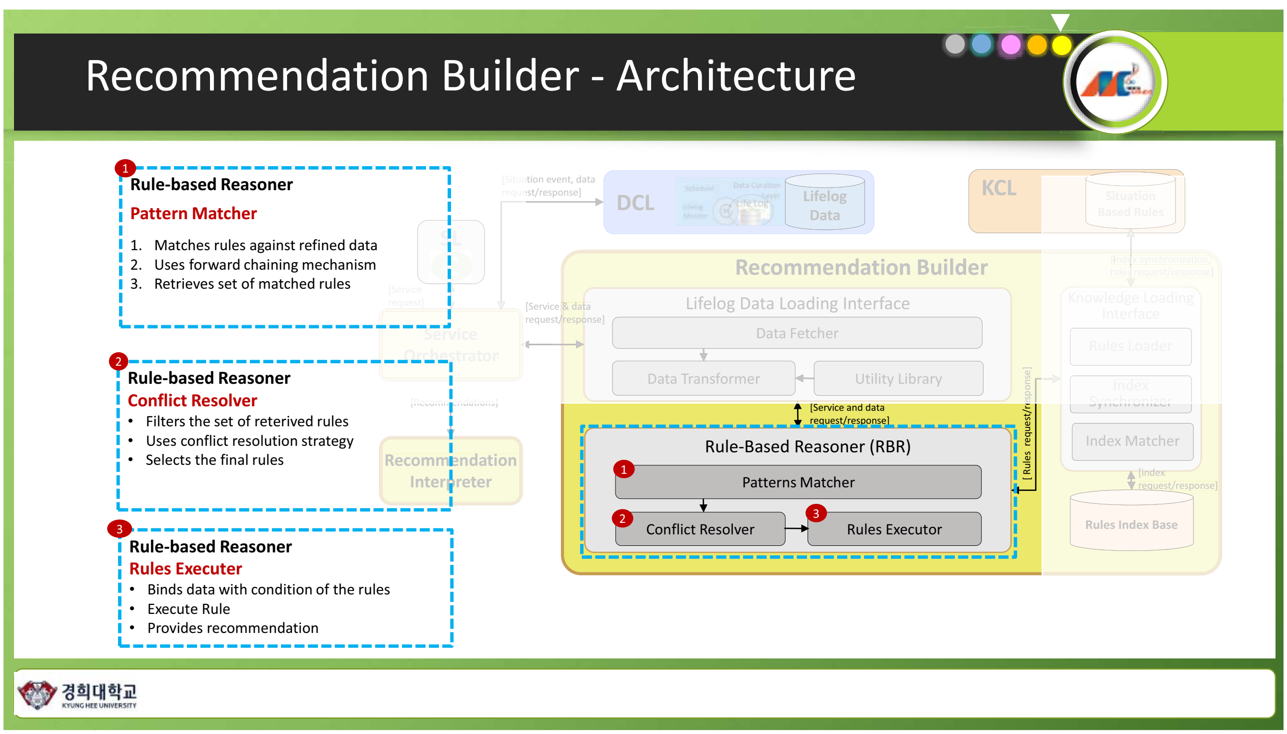


2a Lifelog Data Loading Interface

Utility Library

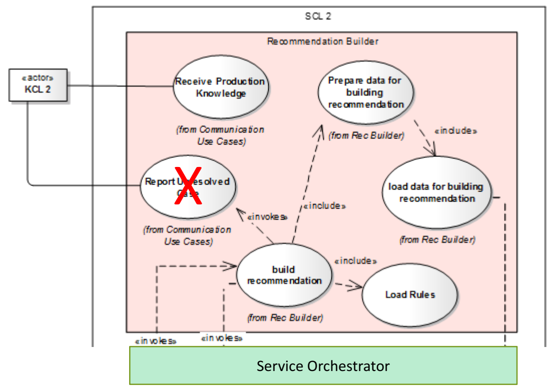
- Utilities functions are defined
- Computations for rules condition attributes are done


Recommendation Builder - Architecture
1 Knowledge Loading Interface
Index Matcher
1. Search and performs reasoning on Rule Index Base
2. Retrieves applicable Rules Index
3. Finds Scheduled Rules For Reasoning
2 Knowledge Loading Interface
Rules Index Base
• Keeps the Updated Index of rules shared by KCL
• Used during Indexed reasoning for searching rules
3 Knowledge Loading Interface
Rules Loader
• Takes Index from the Index Matcher
• Loads Rules from KCL
• Provides Rules to Reasoner
4 Knowledge Loading Interface
Index Synchronizer
• Receives Updated Index From KCL
• Updates the Rules Index Base
KCL
Situation Based Rules
[index synchronization, rules request/response]
Knowledge Loading Interface
Rules Loader
Index Synchronizer
Index Matcher
[index request/response]
Rules Index Base
Recommendation Builder
Lifelog Data Loading Interface
Data Fetcher
Data Transformer
Utility Library
Rule-Based Reasoner (RBR)
Patterns Matcher
Conflict Resolver
Rules Executor
Service Orchestrator
Recommendation Interpreter
Lifelog Data
[Rules request/response]
[Service and data request/response]
경희대학교
KYUNG HEE UNIVERSITY


Recommendation Builder - Architecture
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
DCL
Lifelog Data
KCL
Recommendation Builder
Lifelog Data Loading Interface
Data Fetcher
Data Transformer
Utility Library
[Service and data request/response]
Rule-Based Reasoner (RBR)
Patterns Matcher
Conflict Resolver
Rules Executor
[Service & data request/response]
Index Matcher
Rules Index Base
Recommendation Interpreter
경희대학교
KYUNG HEE UNIVERSITY

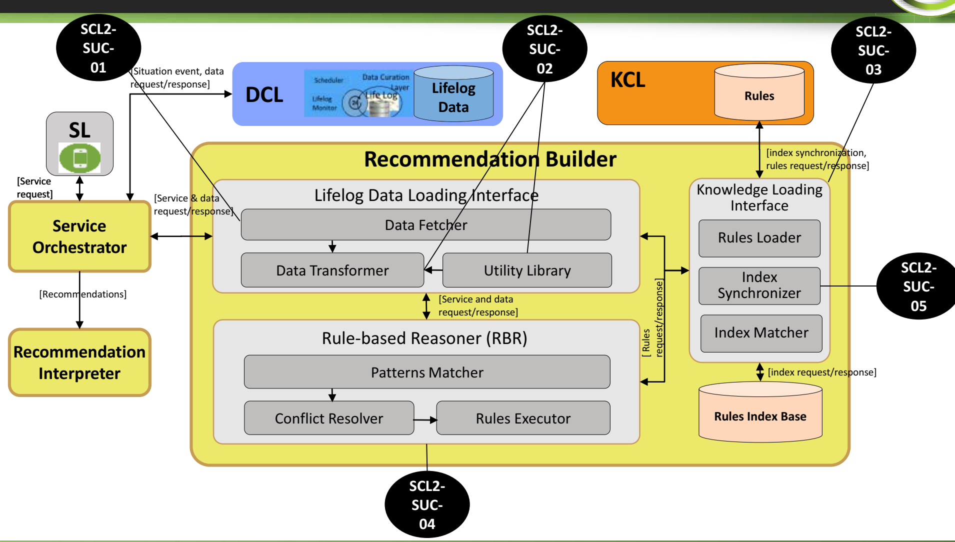
Recommendation Builder – Use cases

Use Case #ID	Description
SCL2-SUC-01	Load data for building recommendation
SCL2-SUC-02	Prepare data for building recommendation
SCL2-SUC-03	Load Rules
SCL2-SUC-04	Build Recommendation
SCL2-SUC-05	Receive Production Knowledge
SCL2-SUC-06	Report Unresolved Case



경희대학교
KYUNG HEE UNIVERSITY

Recommendation Builder - Architecture



경희대학교
KYUNG HEE UNIVERSITY

Use-case description

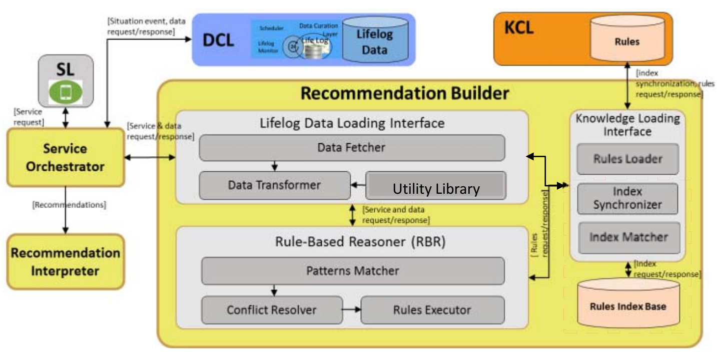
SCL2-SUC-05

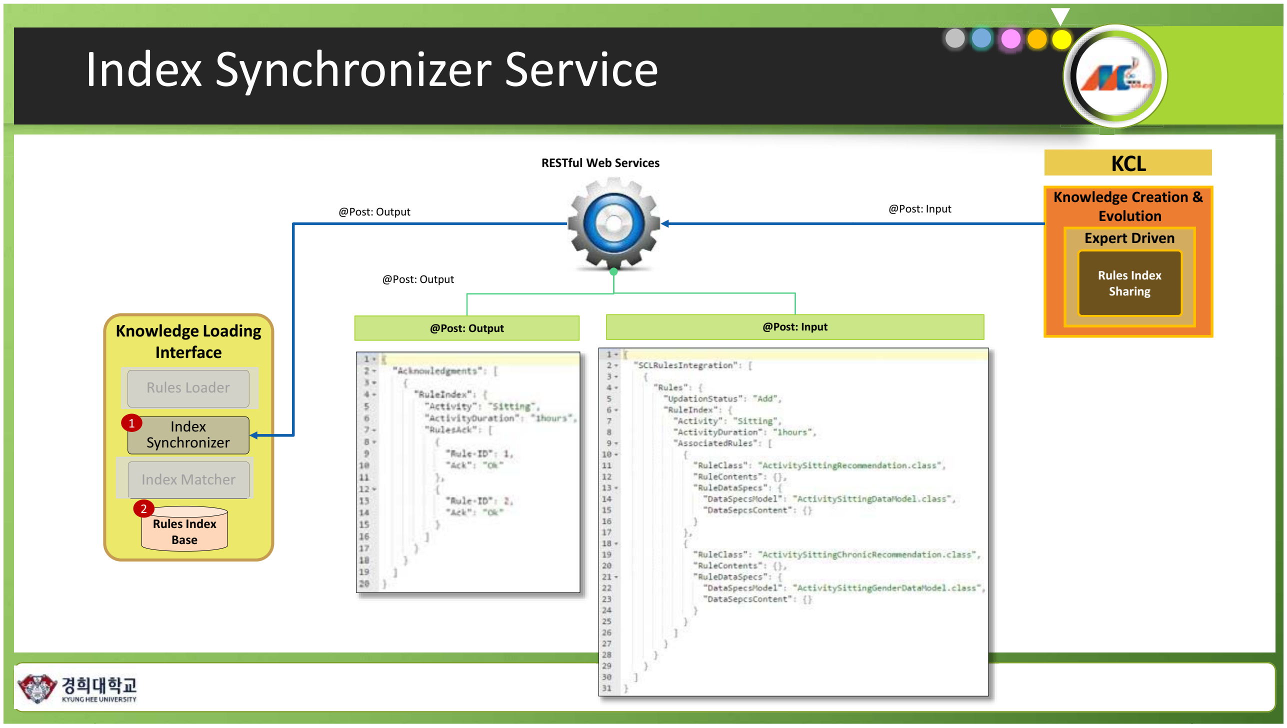
Index Synchronizer

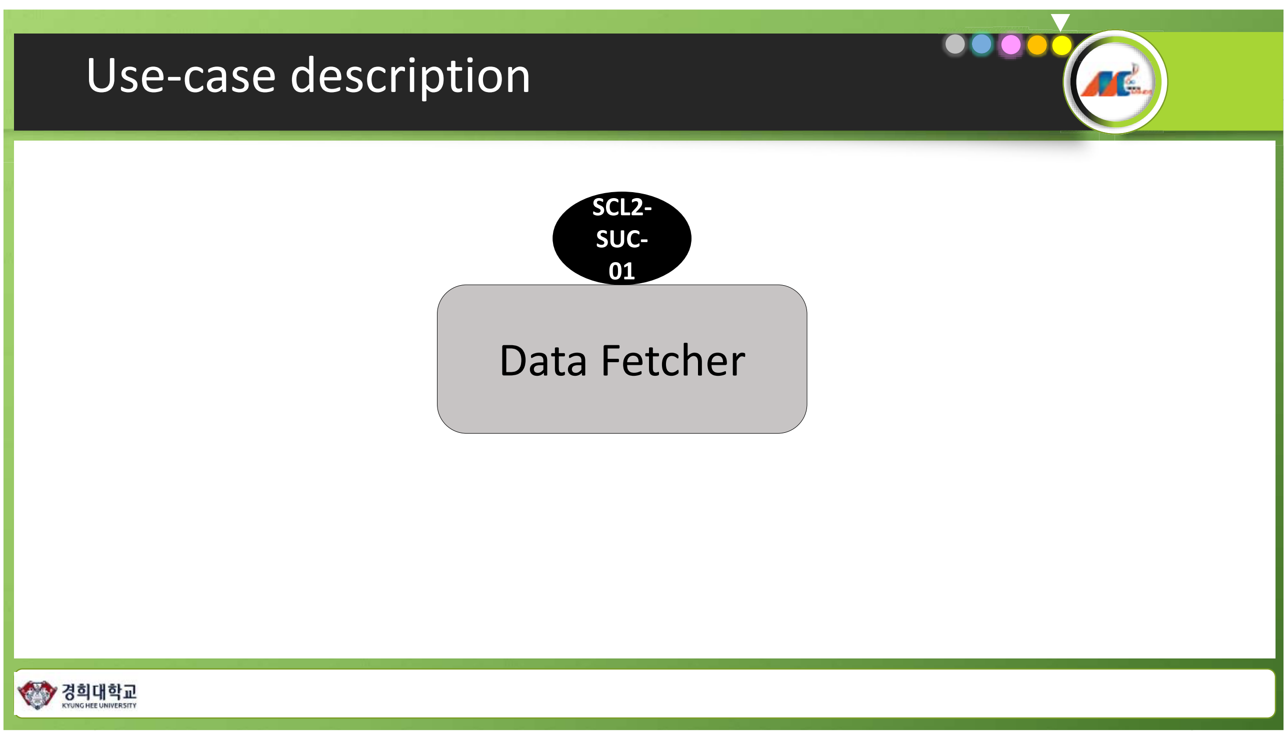
Methodology: Index Synchronizer

Use-case (SCL2-SUC-04): *Receive Production Knowledge*

- **Objective**
  - To update local *Rules Index Base*
- **Methodology**
  - *Restful Service* is defined for Commination between *RB* and *KCL* to update rules index
  - ***Index Synchronizer*** *(operation)*
    1. **Input**: New Rules Index
    2. **Processing**:
       a. receives new Index of new rules from *KCL*
       b. updates the local copy of *Rules Index Base*
    3. **Output**: Updated *Rules Index Base*



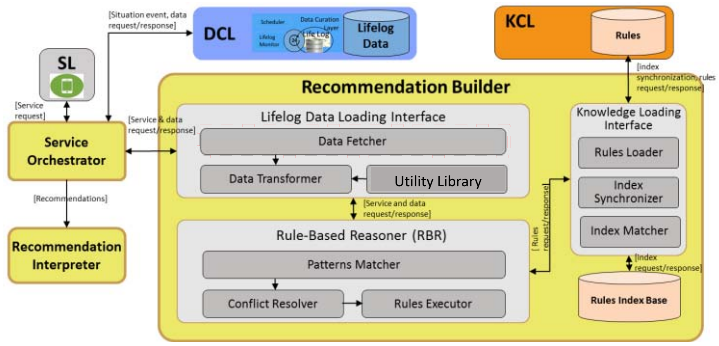

Index Synchronizer Service
RESTful Web Services
@Post: Output
@Post: Input
@Post: Output
KCL
Knowledge Creation & Evolution
Expert Driven
Rules Index Sharing
Knowledge Loading Interface
Rules Loader
1
Index Synchronizer
Index Matcher
2
Rules Index Base
@Post: Output
{
"Acknowledgments": [
{
"RuleIndex": {
"Activity": "Sitting",
"ActivityDuration": "1hours",
"RulesAck": [
{
"Rule-ID": 1,
"Ack": "Ok"
},
{
"Rule-ID": 2,
"Ack": "Ok"
}
]
}
}
]
}
@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": {}
}
}
]
}
}
}
]
}
경희대학교
KYUNG HEE UNIVERSITY


Use-case description
SCL2-SUC-01
Data Fetcher
경희대학교
KYUNG HEE UNIVERSITY

Methodology: Data Fetcher

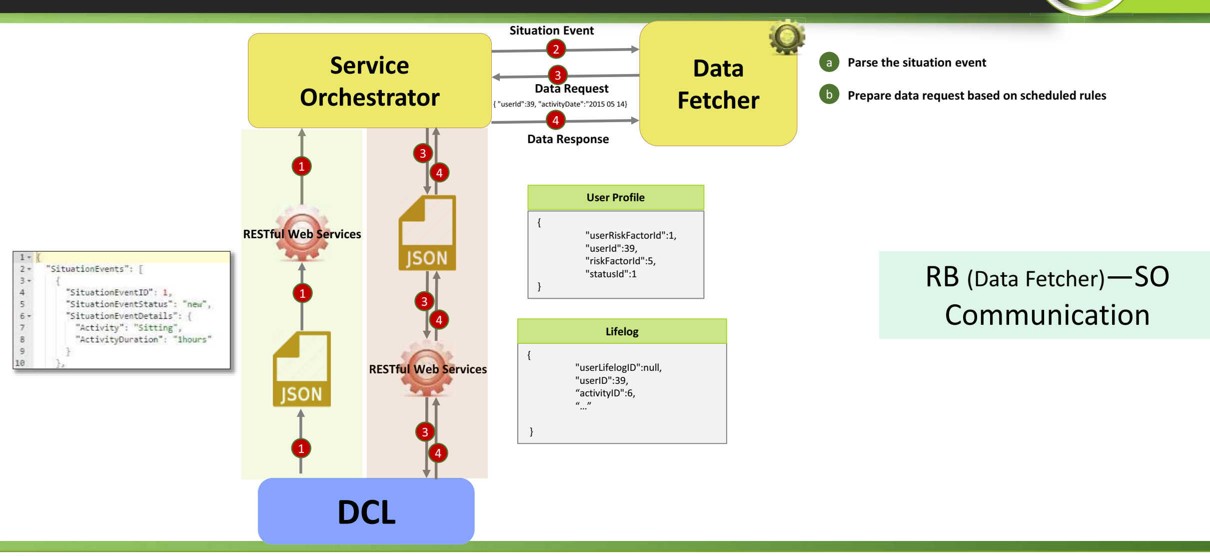
Use-case (SCL2-SUC-01): *Load Data for Building Recommendation*

- **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*

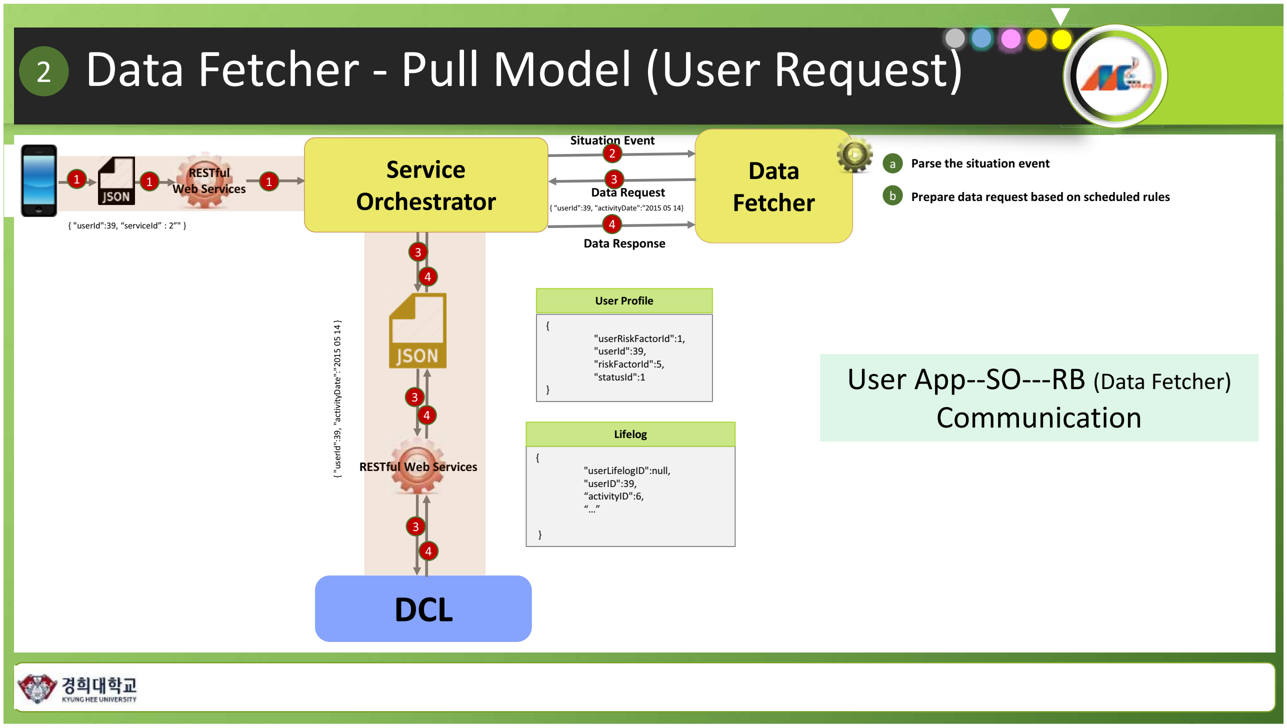


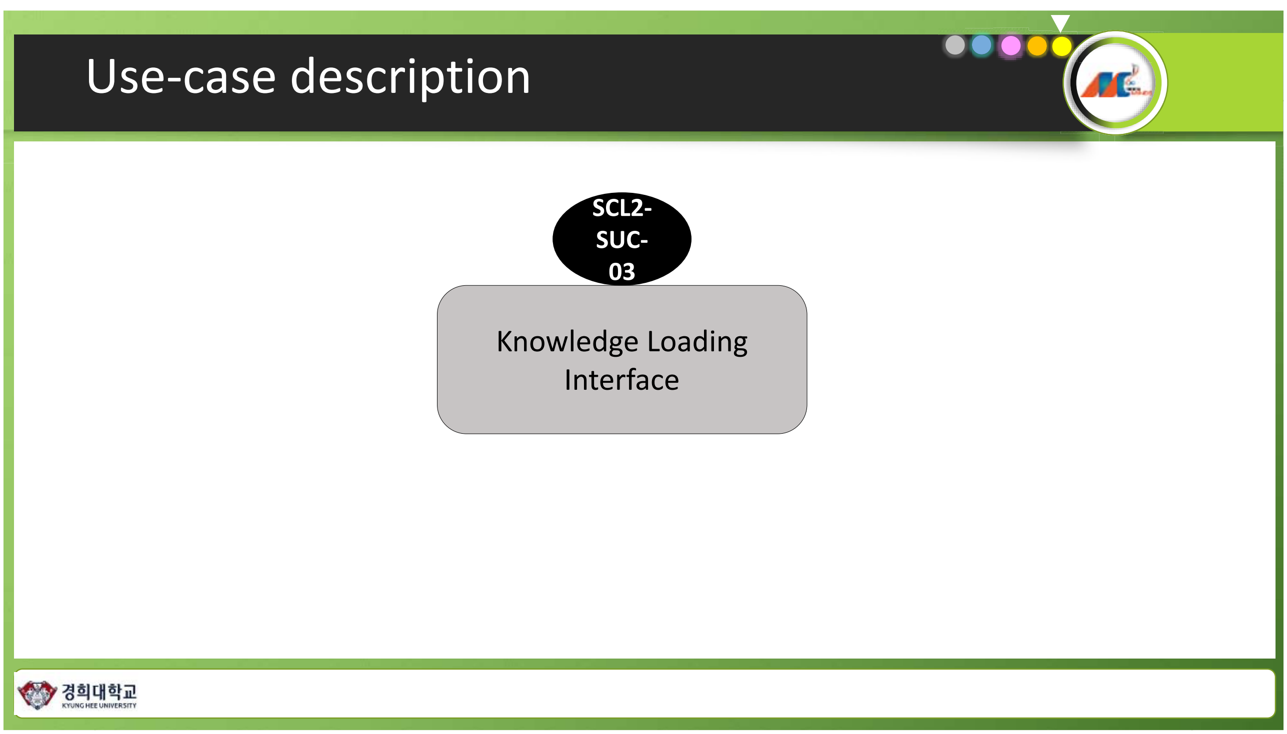
경희대학교 KYUNG HEE UNIVERSITY

1 Data Fetcher - Push Model (Situation Event)



경희대학교 KYUNG HEE UNIVERSITY

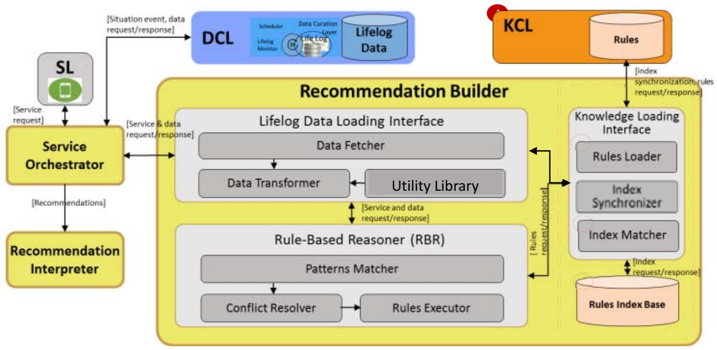

2
Data Fetcher - Pull Model (User Request)
RESTful Web Services
JSON
{ "userId":39, “serviceId” : 2”" }
Service Orchestrator
Situation Event
Data Request
{ "userId":39, "activityDate":"2015 05 14}
Data Response
Data Fetcher
a Parse the situation event
b Prepare data request based on scheduled rules
{ "userId":39, "activityDate":"2015 05 14 }
JSON
RESTful Web Services
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
경희대학교
KYUNG HEE UNIVERSITY


Use-case description
SCL2-SUC-03
Knowledge Loading Interface
경희대학교
KYUNG HEE UNIVERSITY

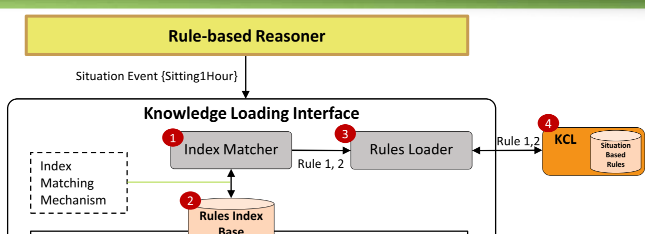
Methodology: Knowledge Loading Interface

Use-case (SCL2-SUC-03): *Load Rules*

- **Objective**
  - Enables Rules Integration
- **Methodology**
  - ***Communication***: *Restful Service* between *KCL* and *RB*
  - ***Index Matcher*** *(operation)*
    1. **Input**: Service Request/Situation Event
    2. **Processing**:
       a. Search and performs reasoning on ***Rule Index Base***
       b. Retrieves appropriate *Rules Index*
    3. **Output**:
       a. Appropriate *Rules Index*
  - ***Rule Loader*** *(operation)*
    1. **Input**: *Rules Index*
    2. **Processing**:
       a. requests KCL for the rules under the specified Index
       b. receives retrieved rules from KCL
    3. **Output**:
       a. A subset of rules, which is forwarded to reasoner



Index matching and rules loading



Index

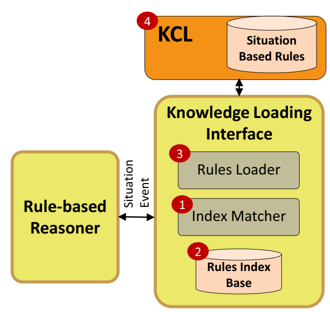
Index Id	Index Name
1	Sitting1Hour
2	Sitting1HourGender
.	
n	

Rules Index Attributes

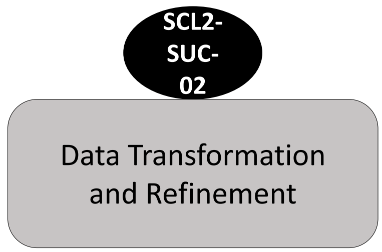
Index Id	Attribute Key	Value
1	Activity	Sitting
1	Activity duration	1Hour
2	Activity	Sitting
2	Activity duration	1Hour
2	Gender	Male
.		

Indexed Rules

Rule Id	Rule Name	Index Id
1	Sitting1HoursRecommendation.cls	1
2	Sitting1HoursGenderRecommendation.cls	1
.		2
n		.



Use-case description

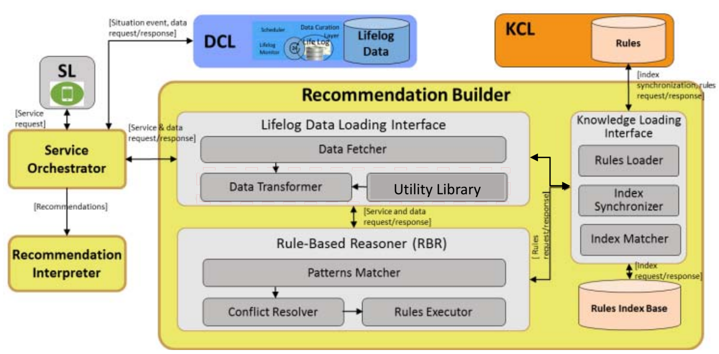


경희대학교 KYUNG HEE UNIVERSITY

Methodology: Data Transformer & Refiner

Use-case (SCL2-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



경희대학교 KYUNG HEE UNIVERSITY

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**

<u>**Lifelog loaded Data**</u>
- Gender: Male
- DoB: 1985
- Height: 6ft
- Weight: 89Kg
- Physical activities
  - Walking: 20m
  - Running: 30m
  - .
  - .
  - Jogging: 10m

Use-case description

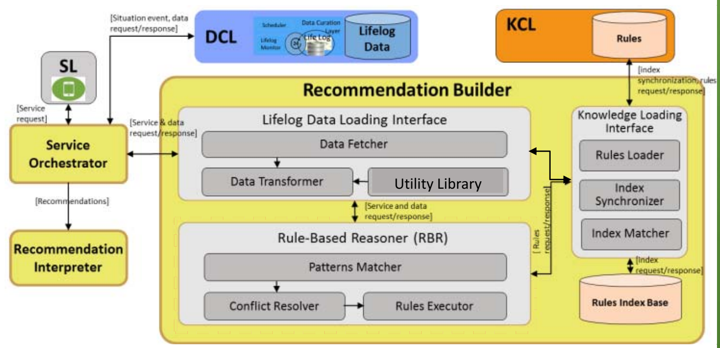
SCL2-SUC-04

Rule-based Reasoner (RBR)

Methodology: Rule-based Reasoner

Use-case (SCL2-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

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

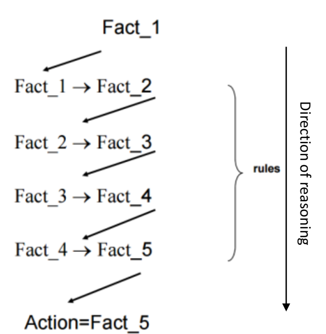
Forward chaining	Backward chaining
• Appropriate when all the facts are provided with the problem statement	• Appropriate when the goal is given in the problem statement
• Appropriate when there are many possible goals or there isn't any sensible way to guess what the goal is at the beginning of the 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



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)

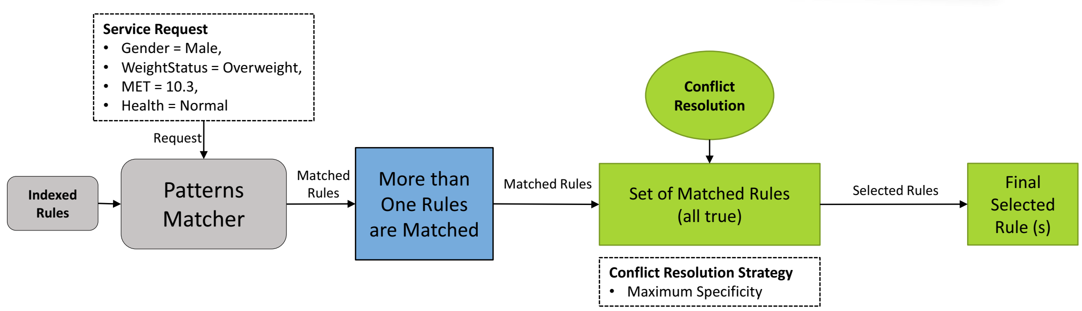
R#2: If BMI=X → Weight Status (UW, OW, N, OB)

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

경희대학교 KYUNG HEE UNIVERSITY

Challenge in forward chaining (pattern matching)



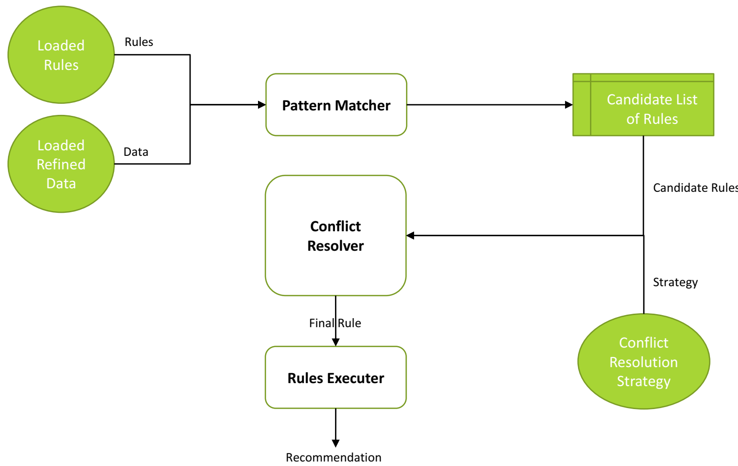
Rule 1:
If Gender = Male, WeightStatus = Overweight, MET = 10.3 → **Recommendation** (Brisk walk for 30 minutes)
Rule 2:
If Gender = Male, WeightStatus = Overweight, MET = 10.3, Health = Normal →
Recommendation (Running for 20 minutes)

Rule 2:
If Gender = Male, WeightStatus = Overweight, MET = 10.3, Health = Normal
→
Recommendation (Running for 20 minutes)

경희대학교 KYUNG HEE UNIVERSITY

RBR with Forward Channing - Flowchart

- Reasoning with forward chaining and conflict resolution



Possible conflict resolution strategies

- 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

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

Tools and Technologies

- Language: Java
- Java Framework: Spring
- Web Server: Glassfish
- NetBean IDE/Eclipse
- Communications
  - Orchestrator-Recommendation builder (*Restful Services*)
  - Recommendation Builder - KCL (*Restful Services*)

Contributions

- 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

- 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

경희대학교
KYUNG HEE UNIVERSITY

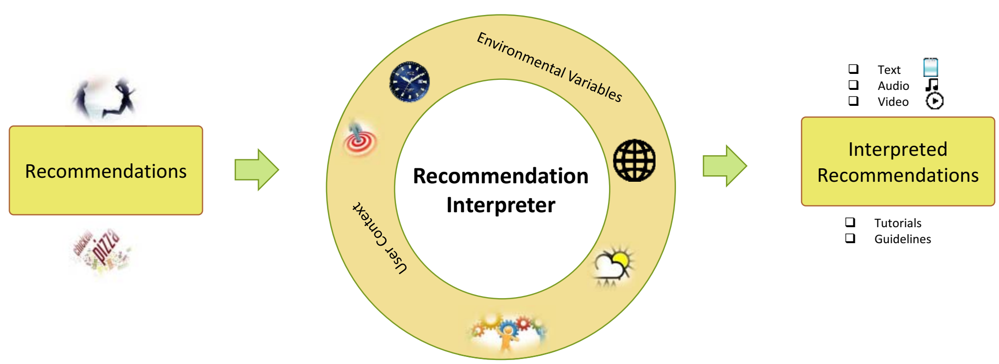
Service Curation Layer (SCL 2.0)

Recommendation Interpreter

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

Introduction: Interpretations

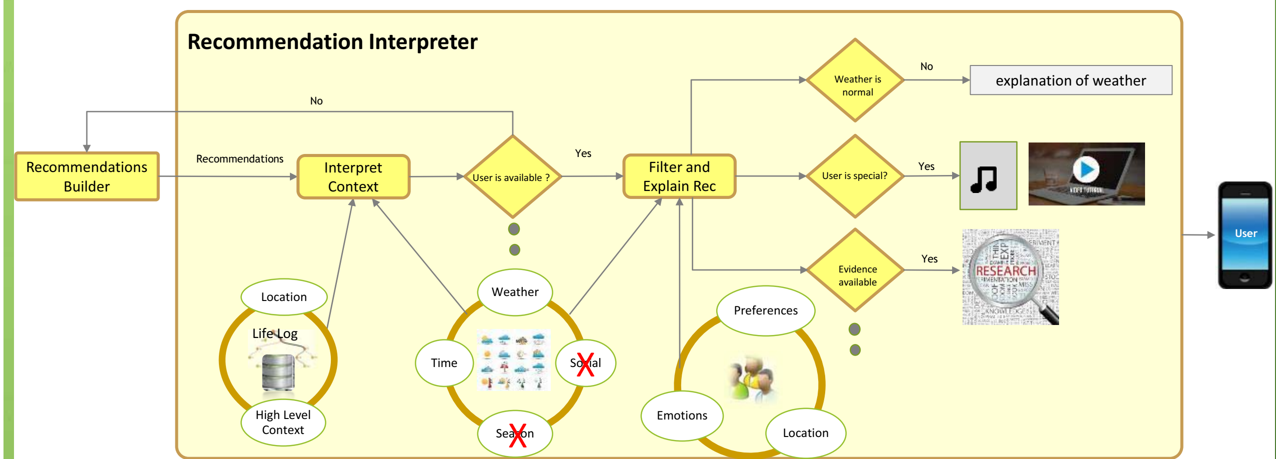
Interpret the recommendations according to the context of the user and environmental variables to deliver the appropriate recommendation at right time



경희대학교 KYUNG HEE UNIVERSITY

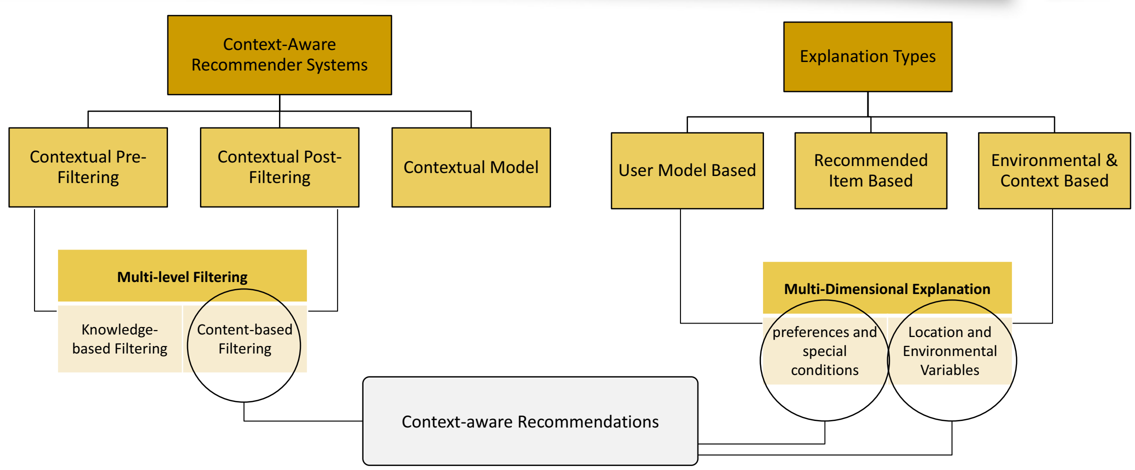
Introduction

Deliver the recommendation according to current context and situation



경희대학교 KYUNG HEE UNIVERSITY

Taxonomy of the work



경희대학교 KYUNG HEE UNIVERSITY

Related Work

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

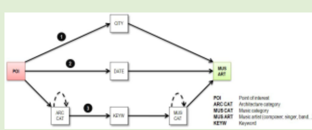
- Lack of incorporating special condition
- Lack of explanations

[Yuan 2014]

경희대학교
KYUNG HEE UNIVERSITY

Related Work

A Generic Semantic-based Framework for Cross-domain Recommendation



Features

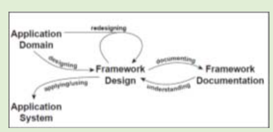
- Generate a generic-base description framework and provide cross-domain recommendation
- Generate open structure knowledge sources from multiple domains

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

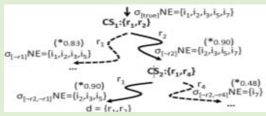
Limitations

- Research based study…not real implementation
- Knowledge Acquisition process bottleneck

[AZAK 2010]

경희대학교
KYUNG HEE UNIVERSITY

Related Work

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 [Felfernig et al. 2009]	• Generate explanations only when user interest matches [Zanker and Ninaus 2010]

경희대학교
KYUNG HEE UNIVERSITY

Limitations of existing work

- 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

경희대학교
KYUNG HEE UNIVERSITY

Motivation

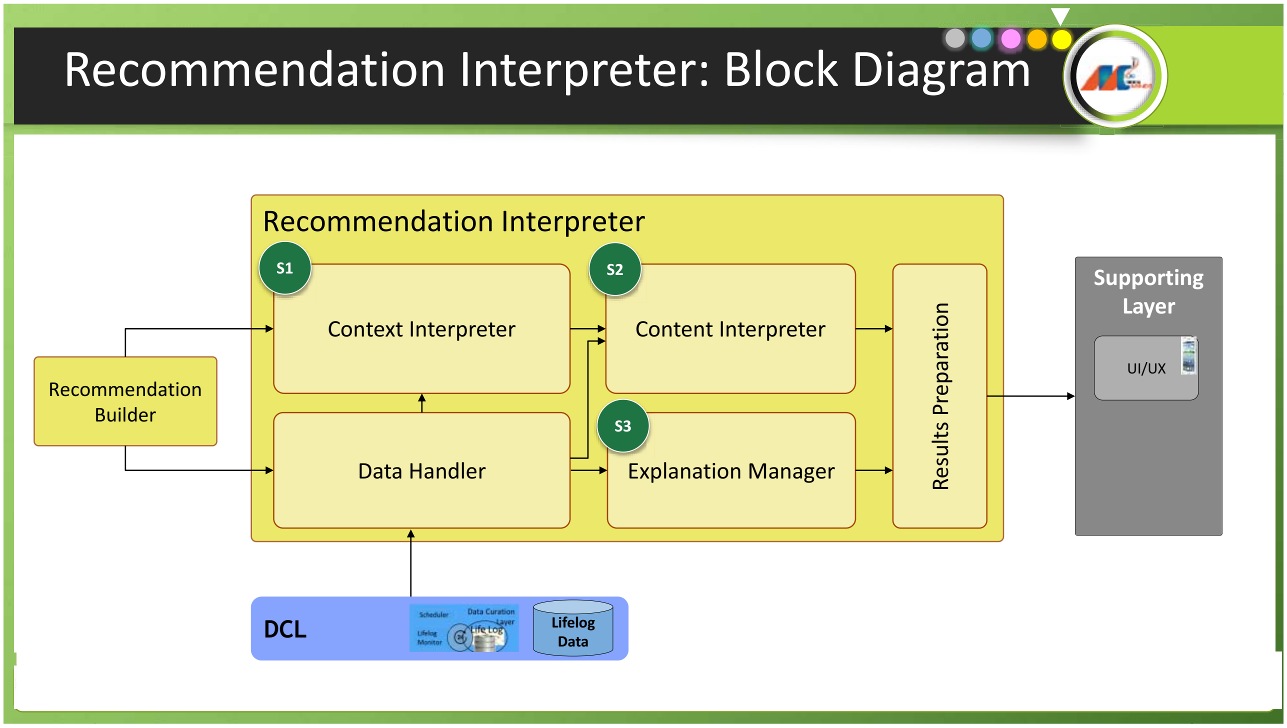
To deliver recommendation at appropriate time based on user current context

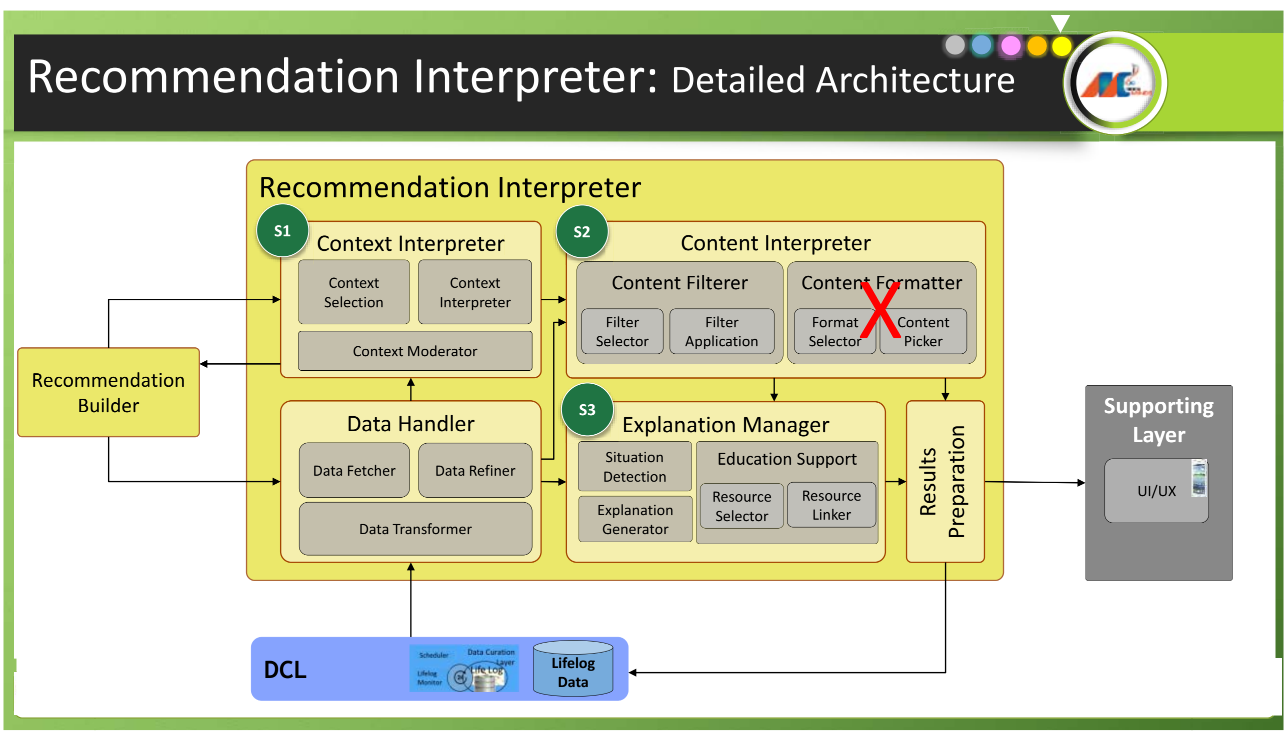
To filter out unnecessary recommendations based on user preferences

To explain recommendation according to situation for better understandability

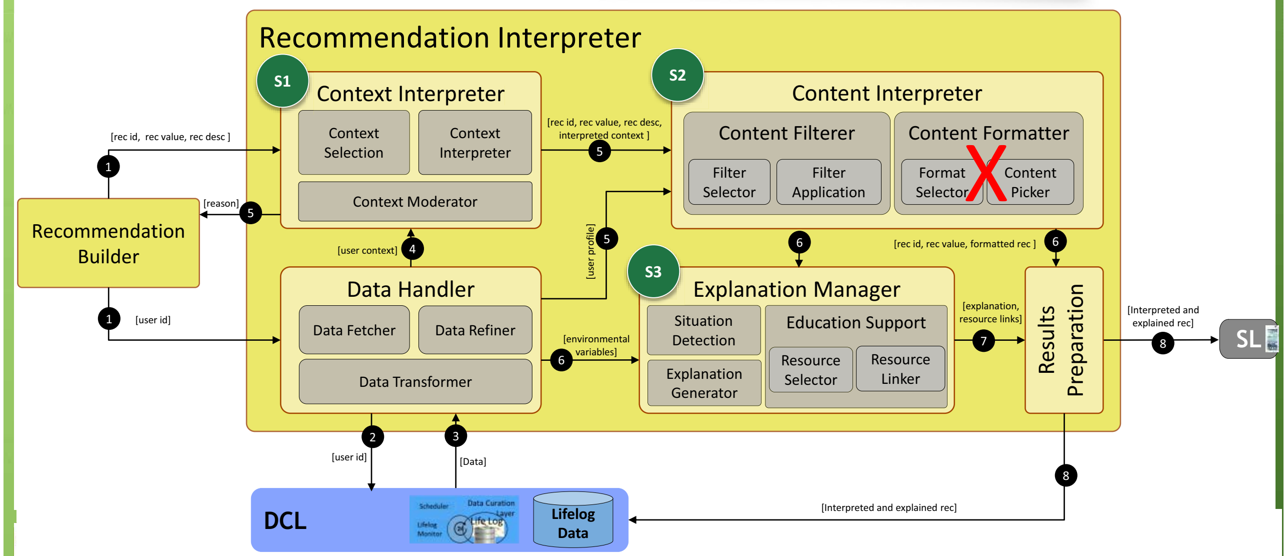
Challenges and Solutions

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**


Recommendation Interpreter: Block Diagram
Recommendation Interpreter
S1
Context Interpreter
S2
Content Interpreter
S3
Explanation Manager
Data Handler
Results Preparation
Recommendation Builder
Supporting Layer
UI/UX
DCL
Lifelog Data

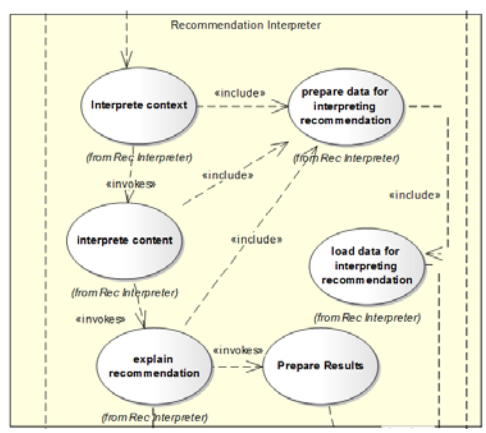

Recommendation Interpreter: Detailed Architecture
Recommendation Interpreter
S1
Context Interpreter
Context Selection
Context Interpreter
Context Moderator
S2
Content Interpreter
Content Filterer
Filter Selector
Filter Application
Content Formatter
Format Selector
Content Picker
S3
Explanation Manager
Situation Detection
Explanation Generator
Education Support
Resource Selector
Resource Linker
Data Handler
Data Fetcher
Data Refiner
Data Transformer
Results Preparation
Recommendation Builder
Supporting Layer
UI/UX
DCL
Lifelog Data

Recommendation Interpreter: Data Flow



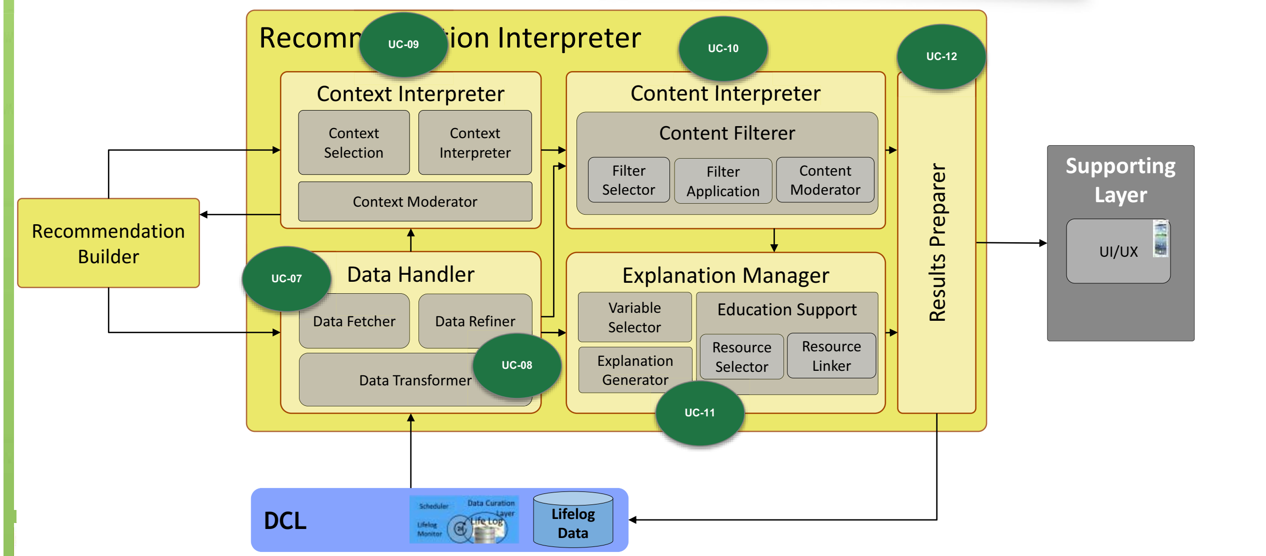
MM V 2.0 Implementation Scope

SCL2-SUC-07	Load data for interpreting recommendation
SCL2-SUC-08	Prepare data for interpreting recommendation
SCL2-SUC-09	Interpret Context
SCL2-SUC-10	Interpret Content
SCL2-SUC-11	Explain recommendations
SCL2-SUC-12	Prepare Results



경희대학교
KYUNG HEE UNIVERSITY

Recommendation Interpreter: Refined Architecture



SCL2-SUC-09

Interpret Context

경희대학교
KYUNG HEE UNIVERSITY

Interpret Context (Use Case 09): Methodology

Methodology Description

Use Case 09: *Interpret Context*

Objective: *To ensure the time is appropriate to deliver the recommendations.*

Input:

1. Recommendation 2. Context

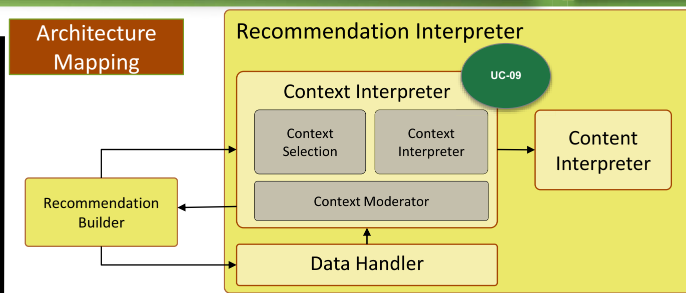
Process:

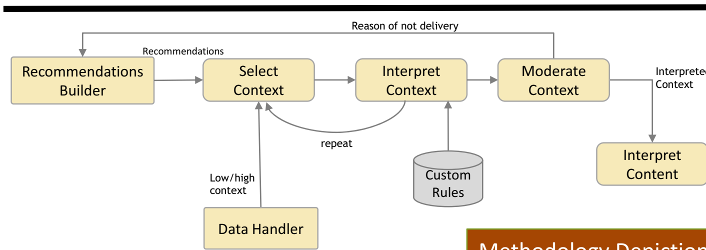
1. *Get the recommendation generated by RB*
2. *Selects context from fetched by Data Handler*
3. *Interpret the selected context*
   a. *Fetch the rules from custom rules repository*
   b. *Parse the rule on selected context*
4. *Moderate the context*
   a. *If time is appropriate → forward the rec and context to content interpreter*
   b. *Else inform the rec builder with reason*

Output:

Interpreted context

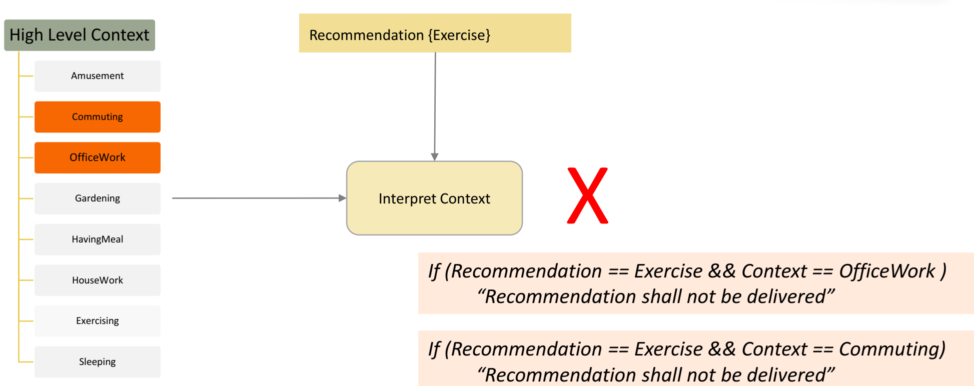
Architecture Mapping



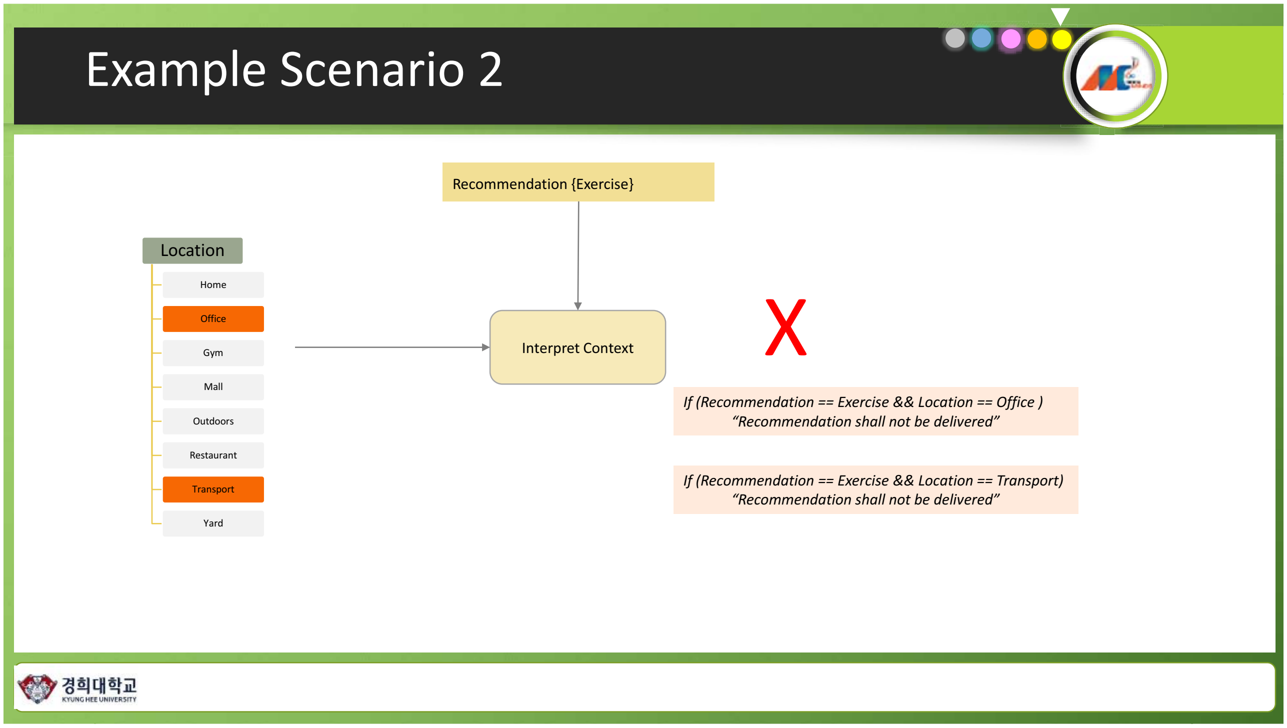


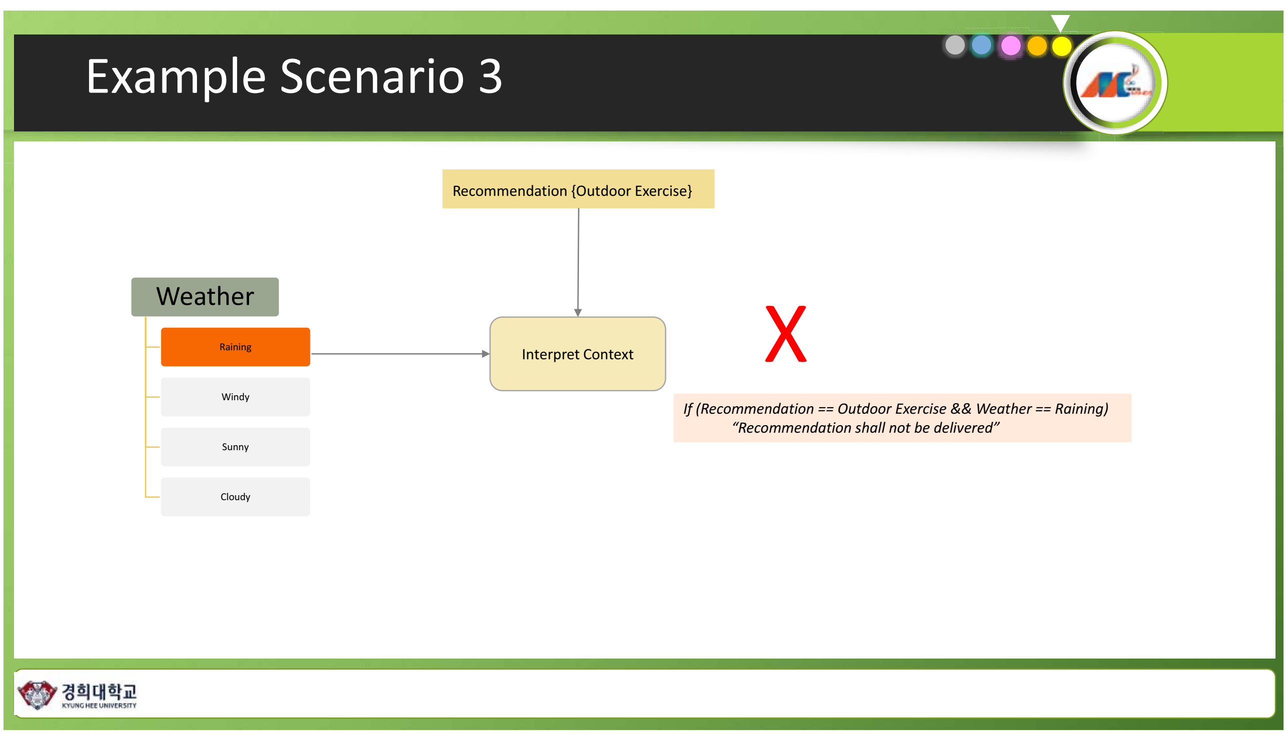
Methodology Depiction

Example Scenario 1

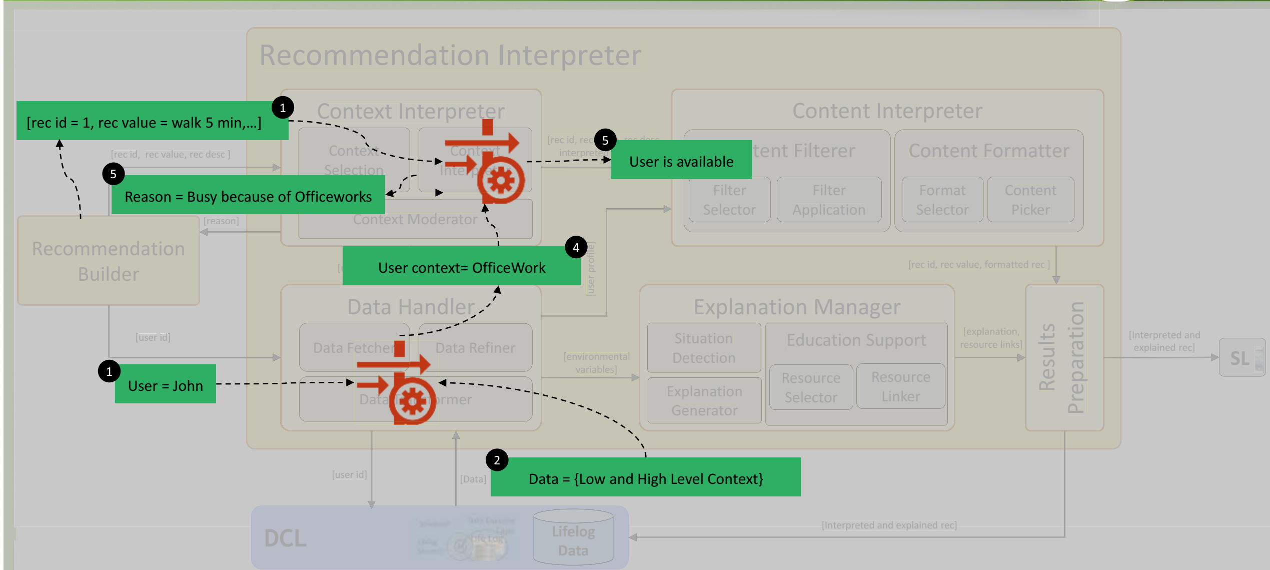


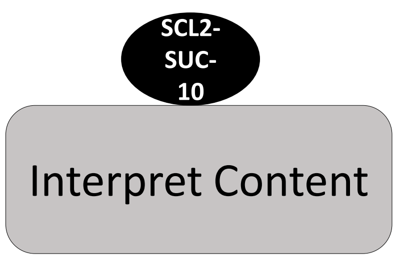
경희대학교
KYUNG HEE UNIVERSITY


Example Scenario 2
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"
경희대학교
KYUNG HEE UNIVERSITY


Example Scenario 3
Recommendation {Outdoor Exercise}
Weather
Raining
Windy
Sunny
Cloudy
Interpret Context
X
If (Recommendation == Outdoor Exercise && Weather == Raining)
"Recommendation shall not be delivered"
경희대학교
KYUNG HEE UNIVERSITY

Interpret Context (Use Case 09): Example Scenario





경희대학교 KYUNG HEE UNIVERSITY

Interpret Content (Use Case 10): Methodology

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

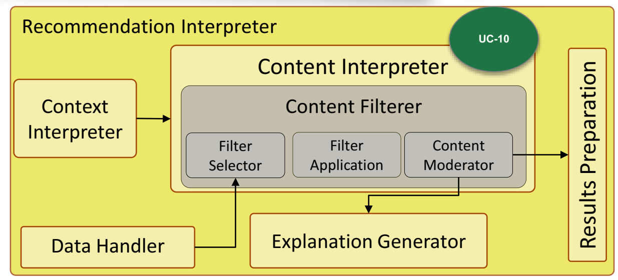
Process:

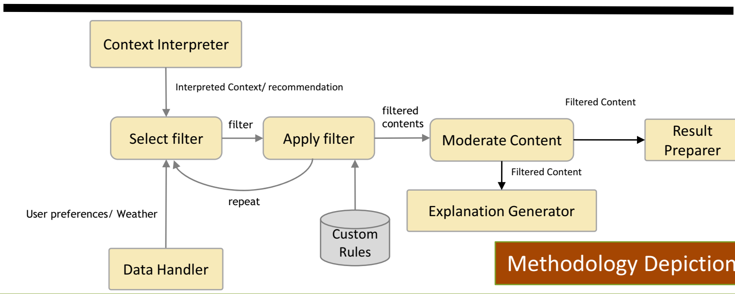
1. *Selects the context filter from context interpreter and (user preferences) filters from data handler*
2. *Content Filtration*
   a. *Get the data (recommendation and filter)*
   b. *Fetch the rules from custom rules repository*
   c. *Parse the rule on selected filter*
3. *Moderate the context*
   a. *If contents are understandable → forward the filtered contents to Result Preparer*
   b. *Else forward to the explanation manager*

Output:

Filtered content

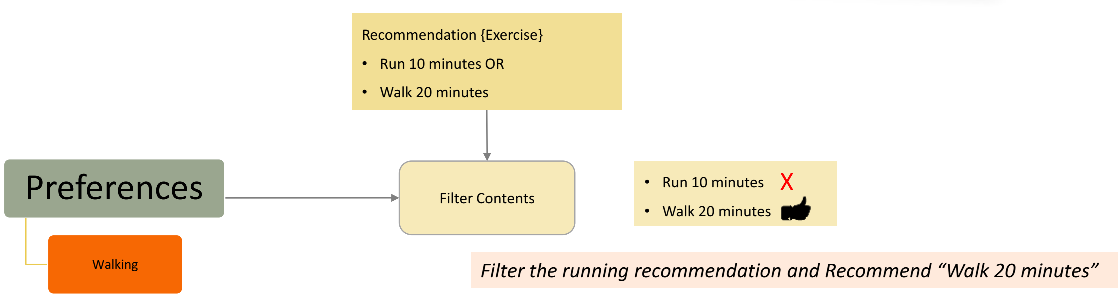
Architecture Mapping





Methodology Depiction

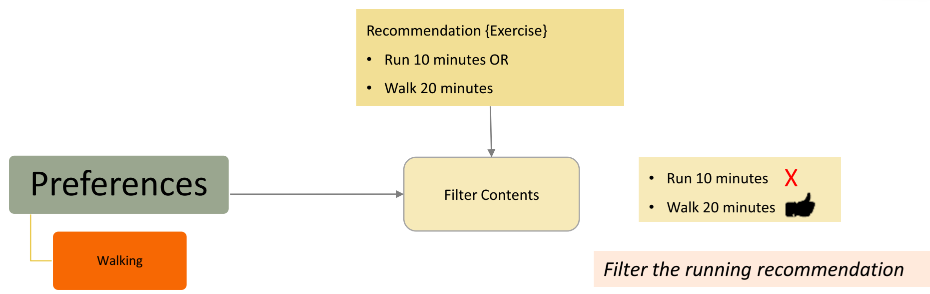
Example Scenario 1



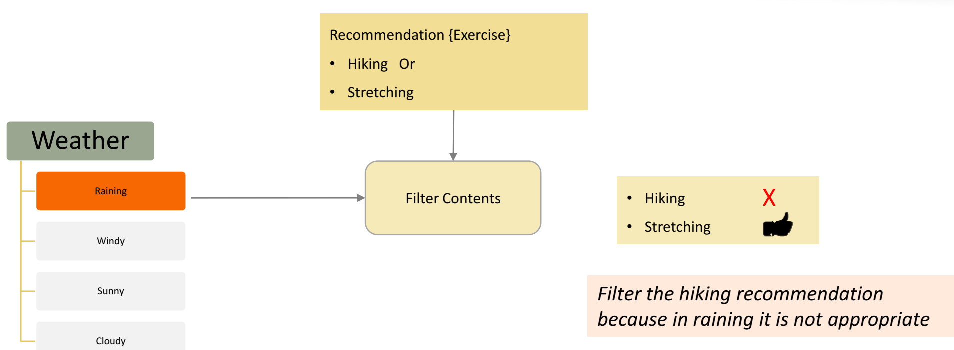
Filter the running recommendation and Recommend "Walk 20 minutes"

경희대학교
KYUNG HEE UNIVERSITY

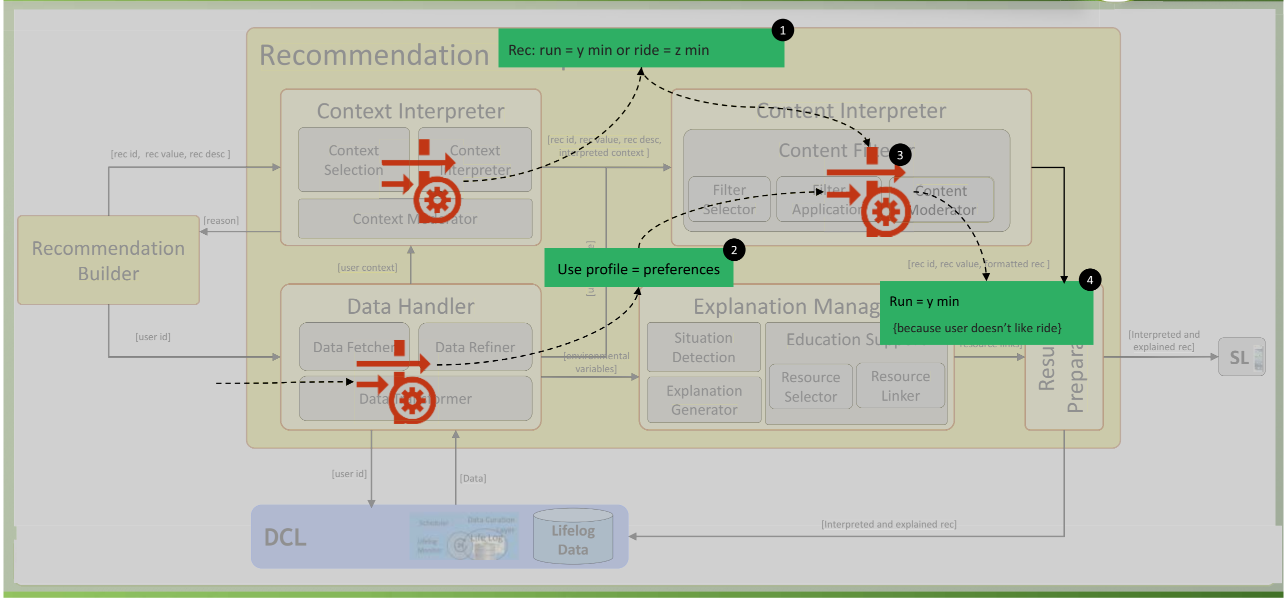
Example Scenario 1

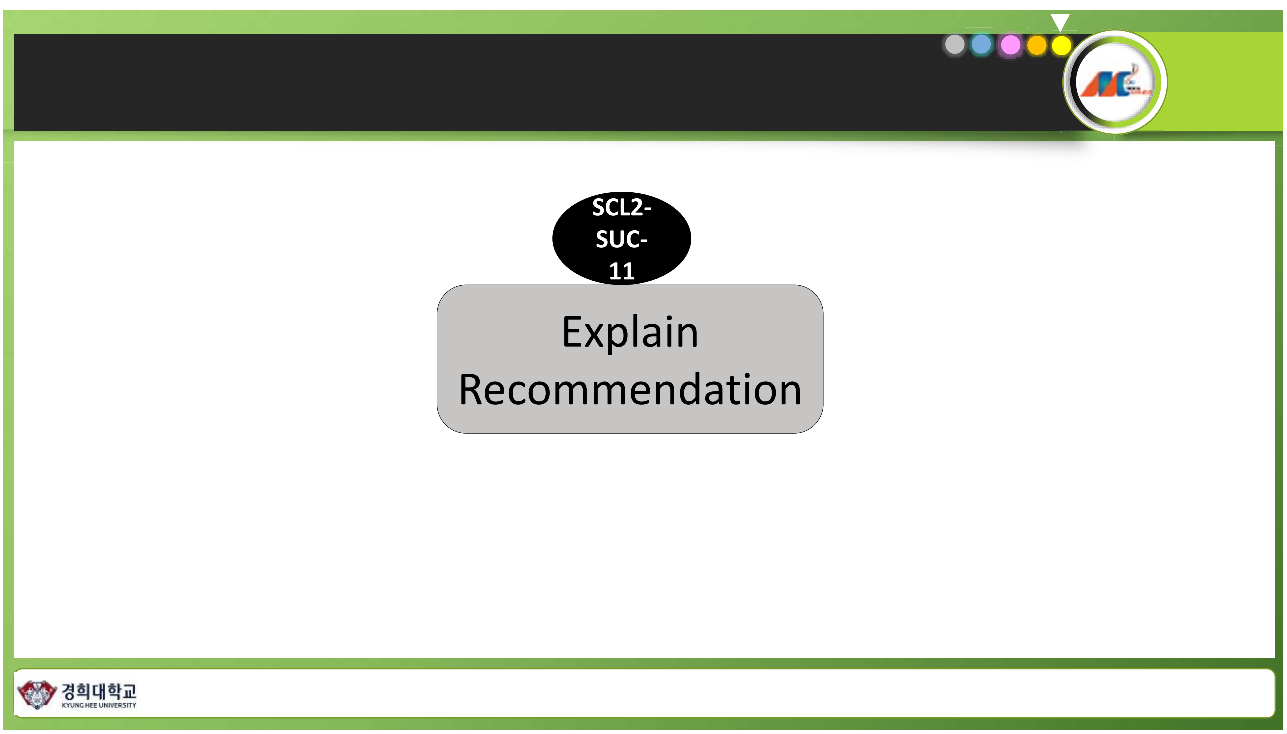


Example Scenario 2



Interpret Content (Use Case 10): Example Scenario





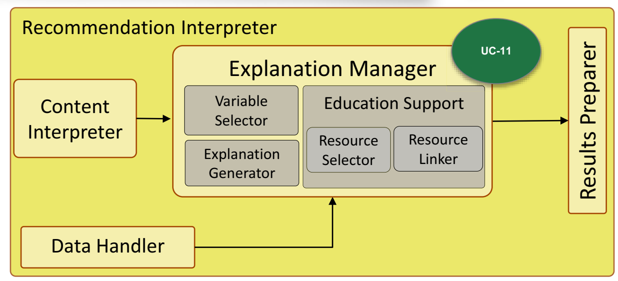
Explain Recommendation (Use Case 11): Methodology

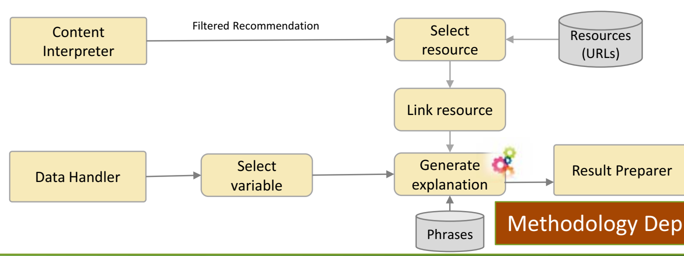
Methodology Description

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

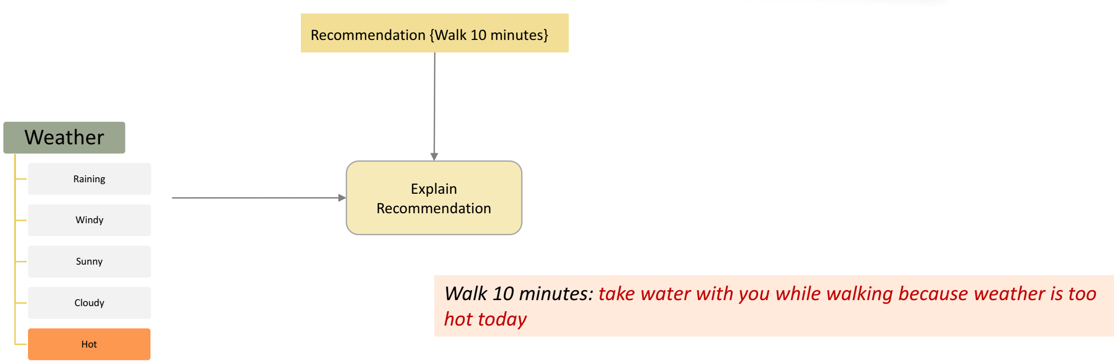
Architecture Mapping





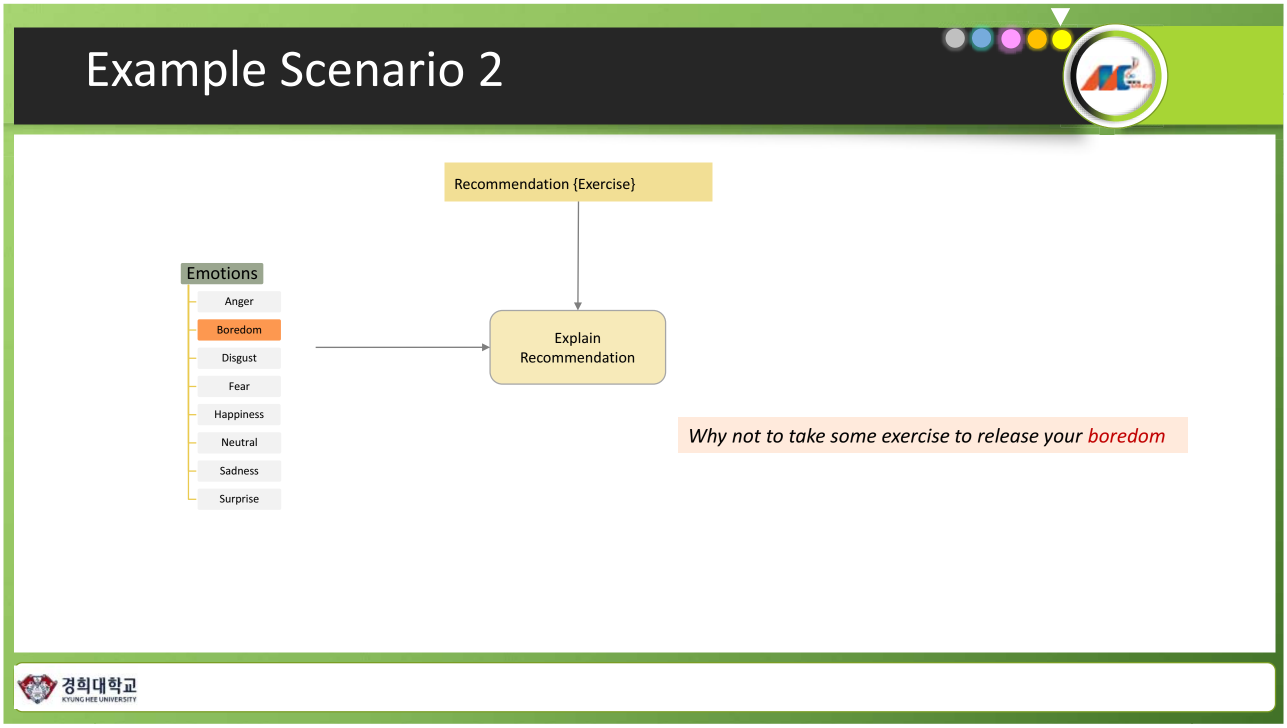
Methodology Depiction

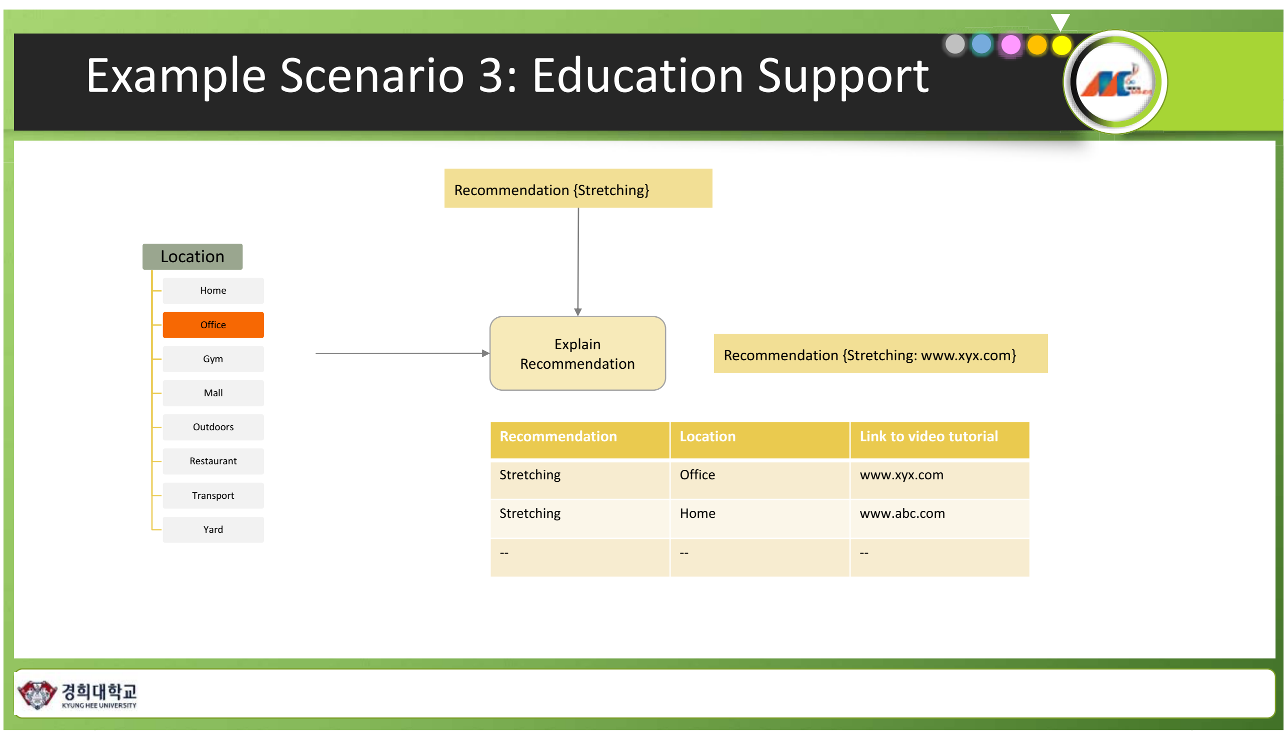
Example Scenario 1



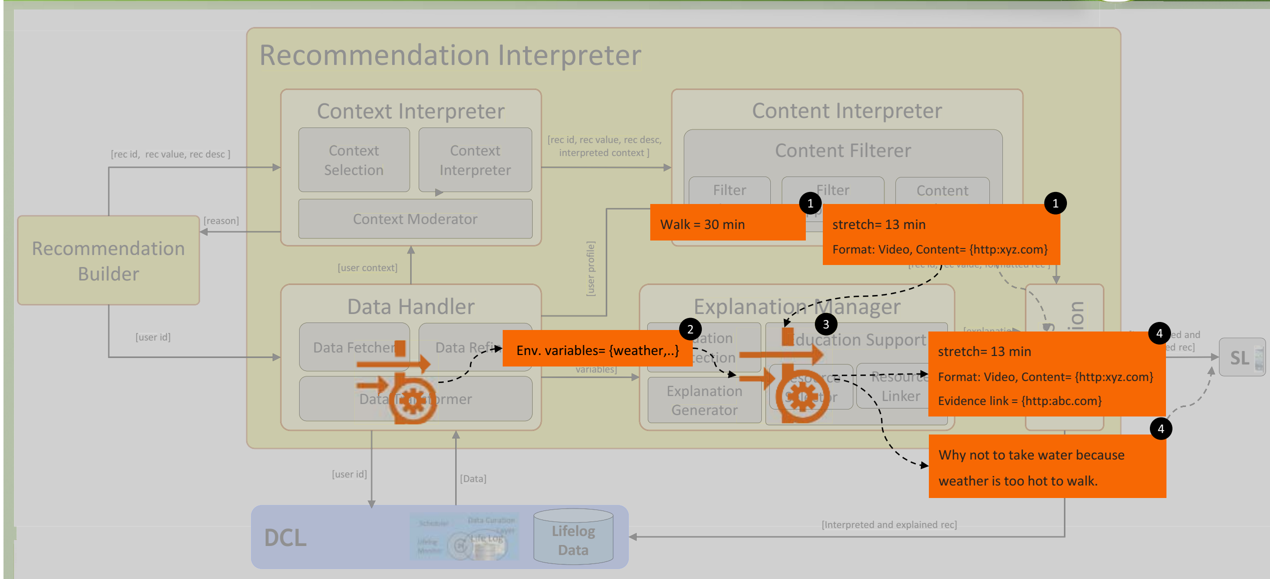
Walk 10 minutes: take water with you while walking because weather is too hot today

경희대학교
KYUNG HEE UNIVERSITY


Example Scenario 2
Recommendation {Exercise}
Emotions
Anger
Boredom
Disgust
Fear
Happiness
Neutral
Sadness
Surprise
Explain Recommendation
Why not to take some exercise to release your boredom
경희대학교
KYUNG HEE UNIVERSITY


Example Scenario 3: Education Support
Recommendation {Stretching}
Location
Home
Office
Gym
Mall
Outdoors
Restaurant
Transport
Yard
Explain Recommendation
Recommendation {Stretching: www.xyx.com}
Recommendation | Location | Link to video tutorial
Stretching | Office | www.xyx.com
Stretching	Home	www.abc.com
경희대학교
KYUNG HEE UNIVERSITY

Explain Recommendation (Use Case 11): Example Scenario



SCL2-SUC-12

Prepare and Deliver Results

경희대학교
KYUNG HEE UNIVERSITY

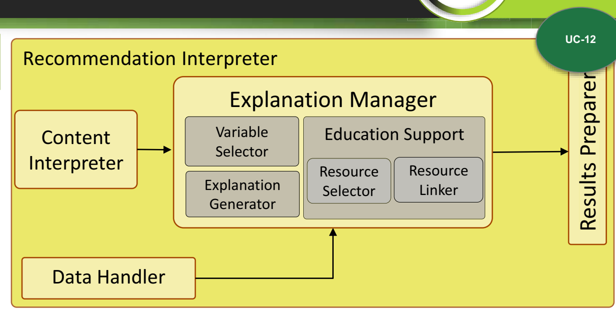
Prepare and Deliver Results (Use Case 12): Methodology

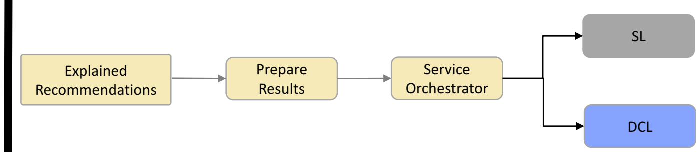
Methodology Description

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





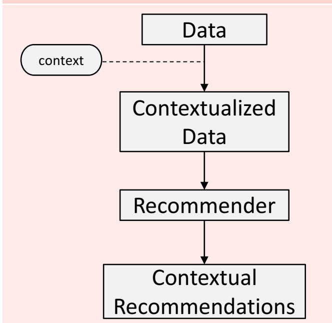
Methodology Depiction

Strategies

Filtering Strategies

Pre-contextual Filtering

Use context prior to generate recommendations

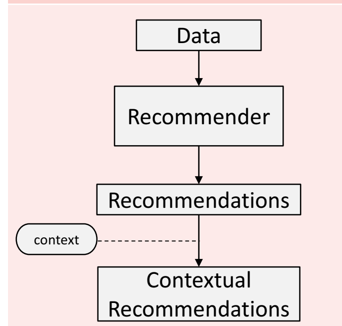


Preferable when recommendation are required to generate only contextualized data

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

경희대학교 KYUNG HEE UNIVERSITY

Explanation Strategies

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.

경희대학교 KYUNG HEE UNIVERSITY

Tools and Technologies

- Language: Java
- Java Framework: J2EE
- Web Server: Glassfish
- IDE: NetBeans/Eclipse
- Communications
  - Rec Builder-Rec Interpreter (*Method Call*)
  - Rec Interpreter – SL (Restful Service)

Contributions

- 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 education by providing links to external knowledge resources

Supporting Layer (SL 2.0)

Principal Investigator:
Professor Sungyoung Lee

Supporting Layer (SL 2.0)

경희대학교 KYUNG HEE UNIVERSITY

Background

V 1.0 Traditional Analytics

- Internally Sourced and relatively small structured data
- Teams of Analysts
- Internal Decision support

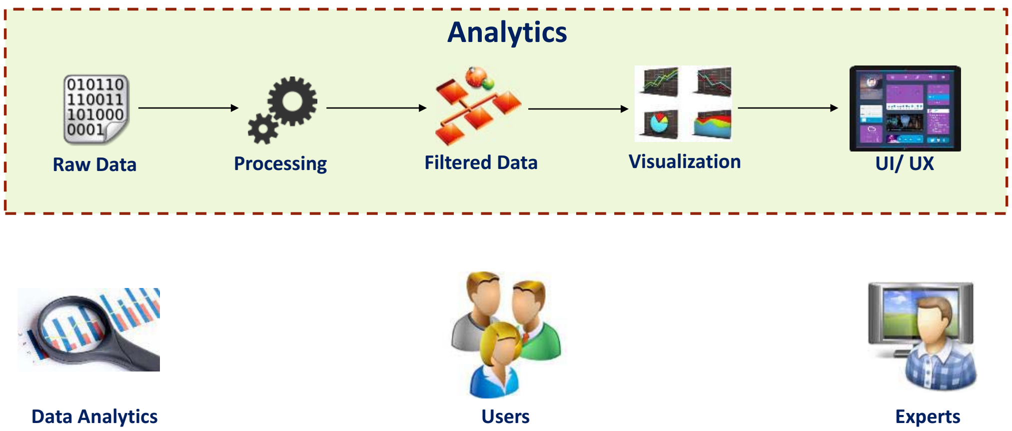
V 2.0 Descriptive Analytics

- Complex Large unstructured data sources
- New Analytical and computation capabilities
- Data based Product and Services

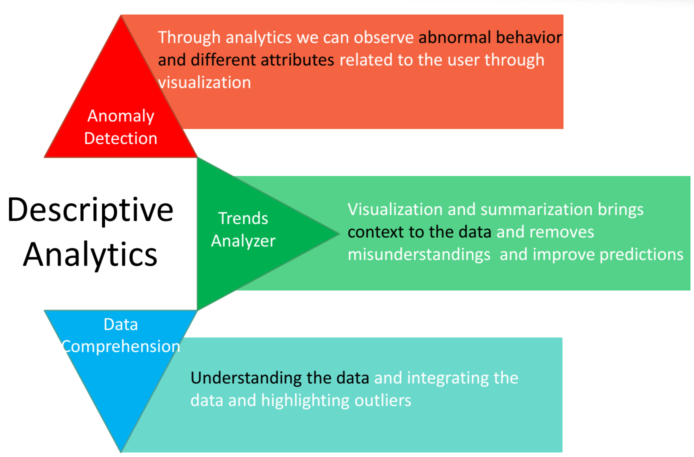
- Provide trending information
- Quantitative summary
- Data visualization

경희대학교 KYUNG HEE UNIVERSITY

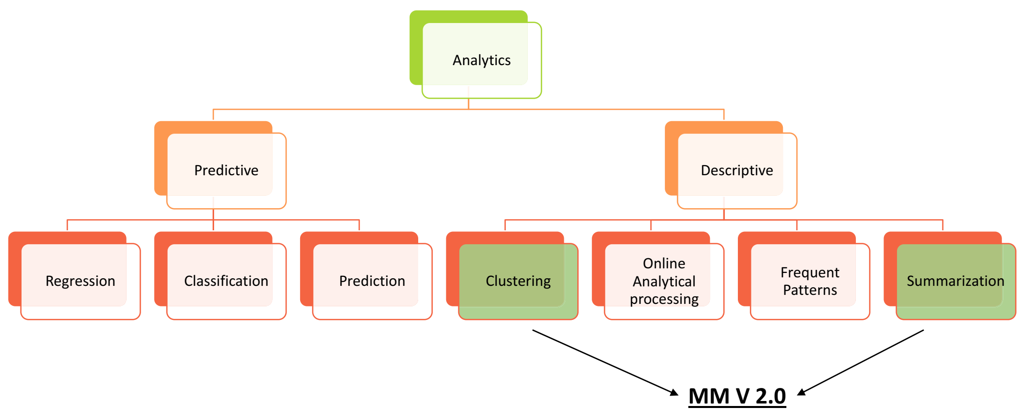
Introduction



Motivation



Taxonomy



A Survey of Predictive Analytics in Data Mining with Big Data ATHABASCA UNIVERSITY July 2014

Related Work

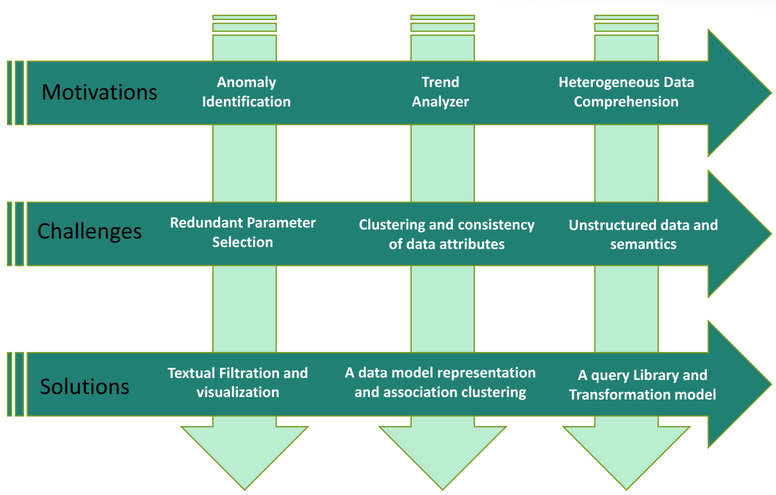
	Systems	Limitations
Clustering	• The automated extraction of interrelated data objects from ERP systems is discussed but without using a graph model and for the single analytical goal of process mining	• No parameter type classification • No distribution of data ranges and rendering information
	• Gradoop (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

- 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

MM V2 Challenges and Solutions

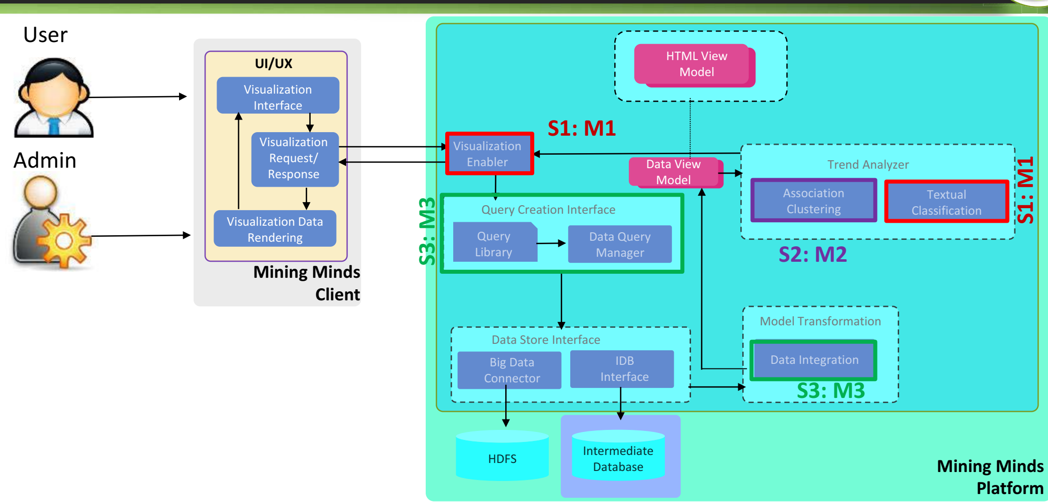


Solutions and Methods

	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

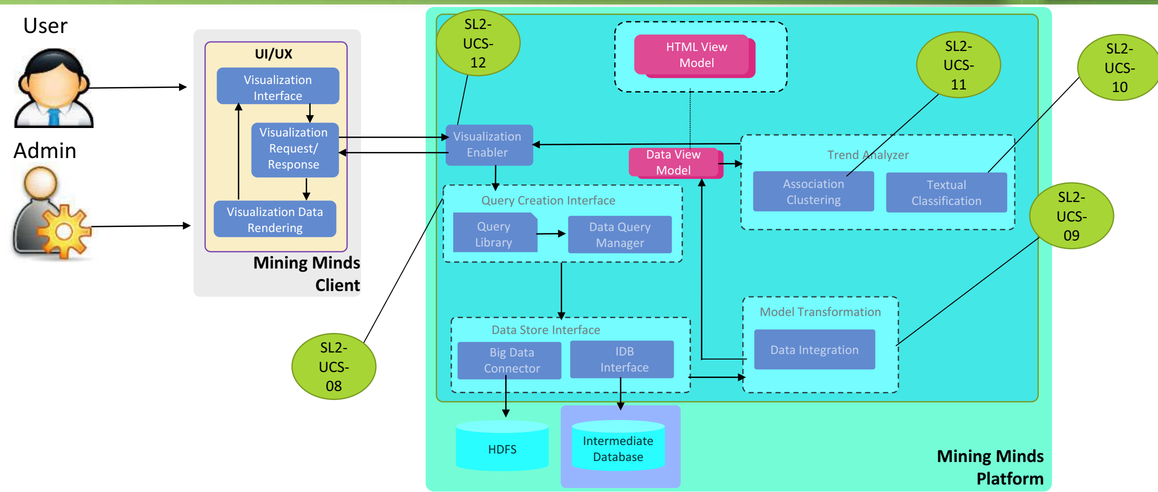
경희대학교
KYUNG HEE UNIVERSITY

Descriptive Analytics in MMV2



경희대학교
KYUNG HEE UNIVERSITY

Descriptive Analytics in MMV2

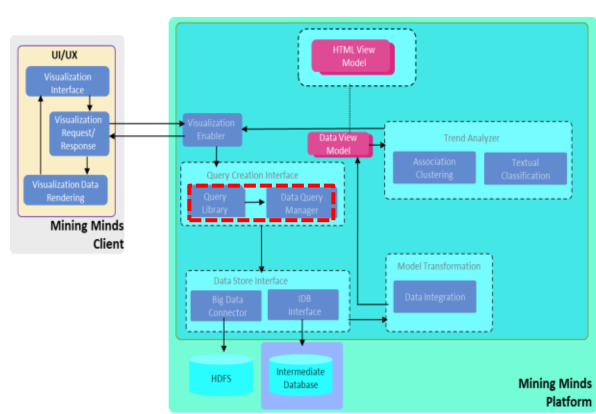


경희대학교 KYUNG HEE UNIVERSITY

Methodology : Query Creation Interface

Use-case (SL2-UCS-08): *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.

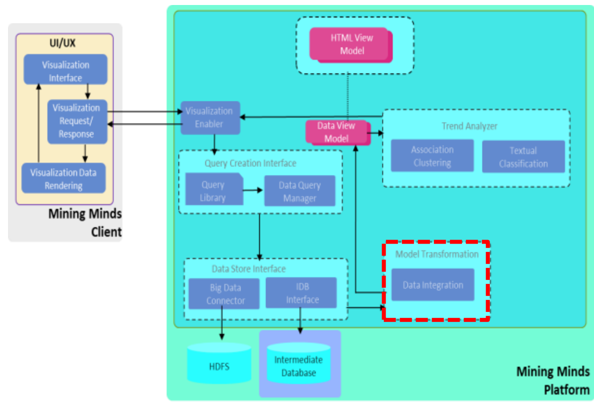


경희대학교 KYUNG HEE UNIVERSITY

Methodology : Model Transformation

Use-case (SL2-UCS-09)**:** *Transform Data*

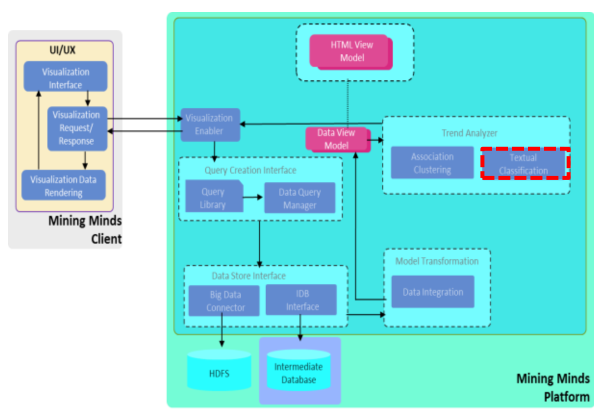
- **Objective**
  - ➢ The mapping query is to be transformed to specific model structure for trend analysis.
- **Methodology**
  - ➢ *Communication*: *Data is passed to model transformation from data store interface*
  - ➢ ***Data Integration****(operation)*
    1. **Input**: receives *request from the 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

Use-case (SL2-UCS-10)**:** *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.

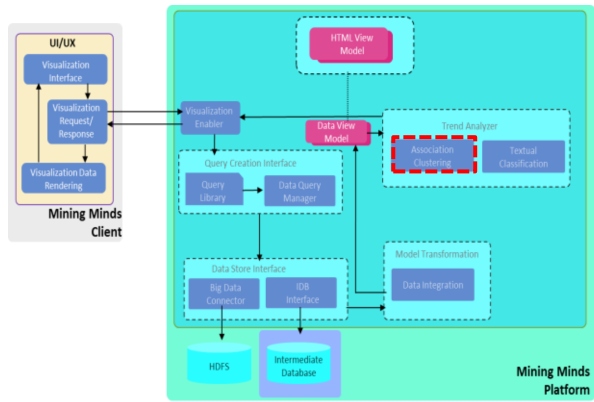


경희대학교 KYUNG HEE UNIVERSITY

Methodology : Association Clustering

Use-case (SL2-UCS-11)**:** *Analyze data*

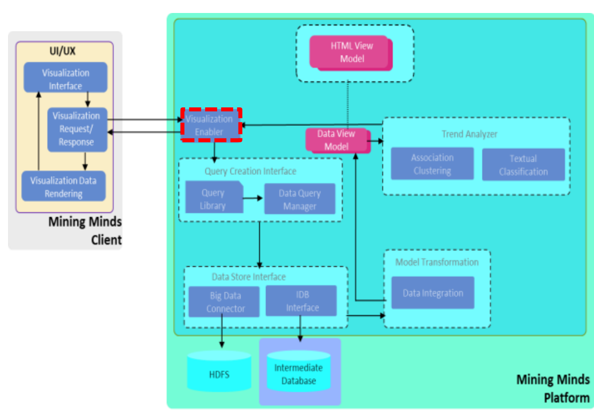
- **Objective**
  - ➢ To prepare data for visualization and analytics
- **Methodology**
  - ➢ *Trend Analysis (operation)*
    1. **Input**: receives *data from model transformation*
    2. **Processing**:
       - The data classifier passes the data for association clustering.
       - The temporal 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

Use-case (SL2-UCS-12)**:** *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



Workflow of Descriptive Analytics

Graph Data

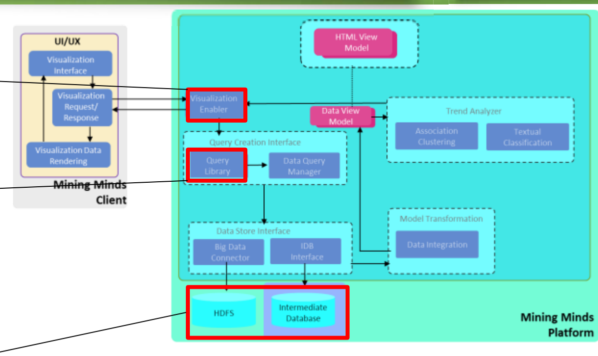
Id	X-axis	Y-axis	Multiple series	Graph
1	Yes	No	No	Pie
2	Yes	Yes	Yes	Line
3	Yes	Yes	Yes	Bar
4	Yes	Yes	No	Bubble

Query Library

Return Attribute	Conditional Attributes	Temporal Attribute
Food	Age	Days
Activity	Duration	DateTime
*.activity	uid	None
Activity	GPS/Location	Days

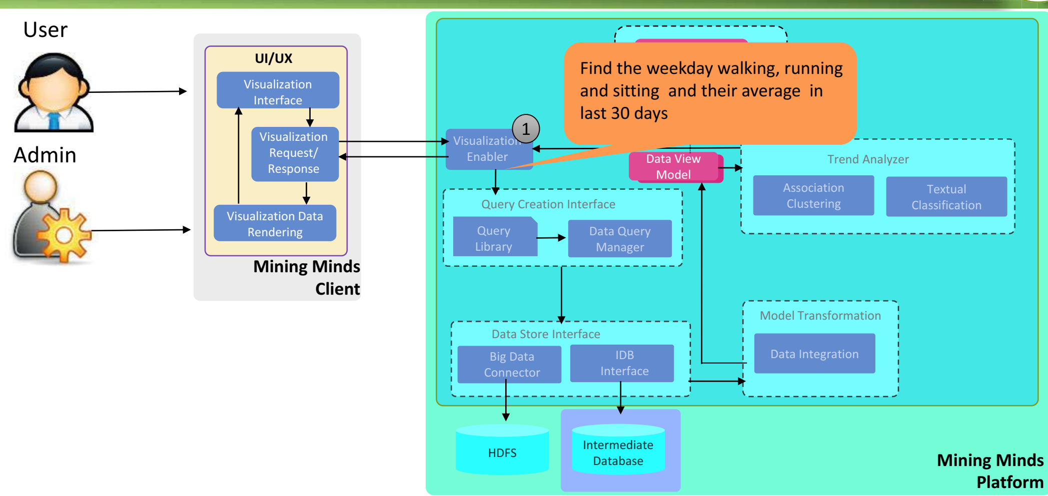
Data Store

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

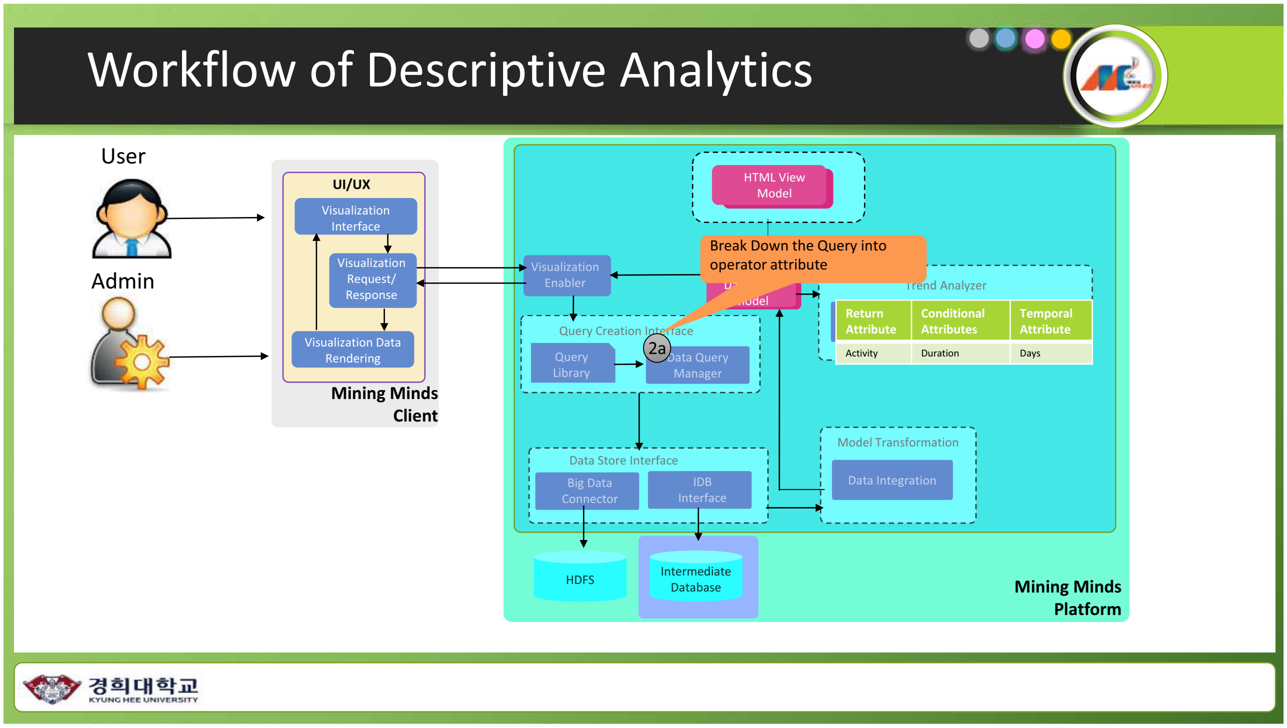


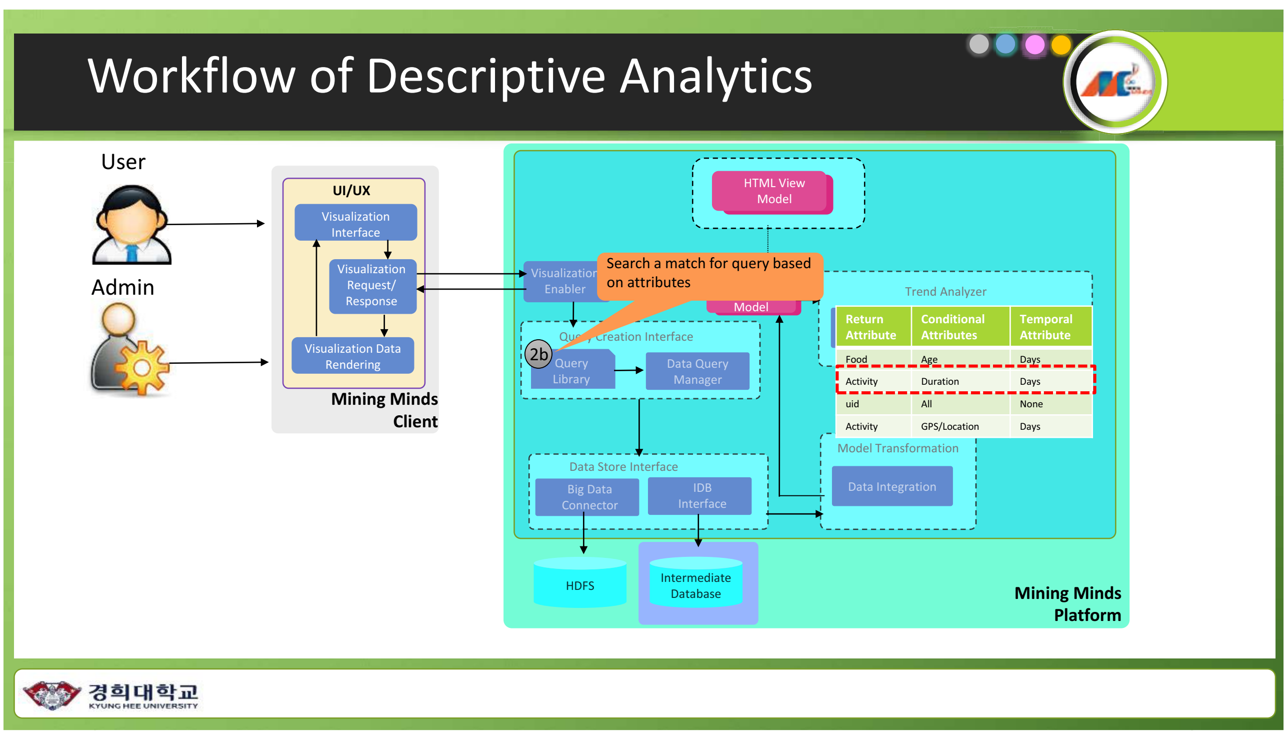
경희대학교
KYUNG HEE UNIVERSITY

Workflow of Descriptive Analytics

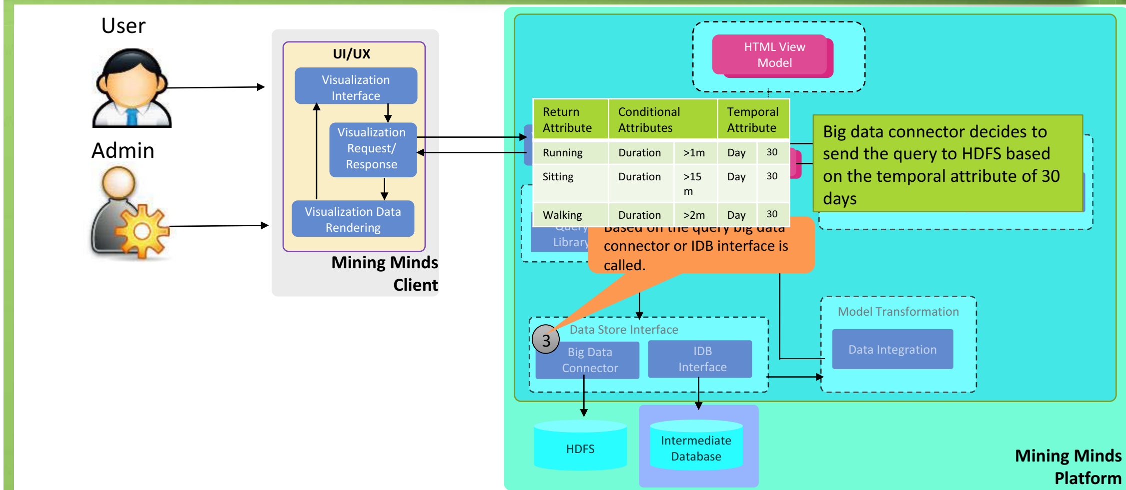


경희대학교
KYUNG HEE UNIVERSITY


Workflow of Descriptive Analytics
User
Admin
UI/UX
Visualization Interface
Visualization Request/ Response
Visualization Data Rendering
Mining Minds Client
HTML View Model
Break Down the Query into operator attribute
Visualization Enabler
Trend Analyzer
Return Attribute
Conditional Attributes
Temporal Attribute
Activity
Duration
Days
Query Creation Interface
Query Library
2a
Data Query Manager
Data Store Interface
Big Data Connector
IDB Interface
Model Transformation
Data Integration
HDFS
Intermediate Database
Mining Minds Platform
경희대학교
KYUNG HEE UNIVERSITY

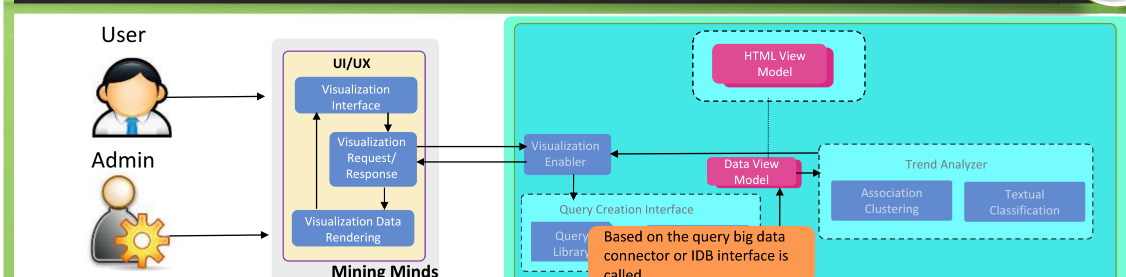

Workflow of Descriptive Analytics
User
Admin
UI/UX
Visualization Interface
Visualization Request/ Response
Visualization Data Rendering
Mining Minds Client
HTML View Model
Search a match for query based on attributes
Visualization Enabler
Model
Trend Analyzer
Return Attribute
Conditional Attributes
Temporal Attribute
Food
Age
Days
Activity
Duration
Days
uid
All
None
Activity
GPS/Location
Days
2b
Query Library
Data Query Manager
Data Store Interface
Big Data Connector
IDB Interface
Model Transformation
Data Integration
HDFS
Intermediate Database
Mining Minds Platform
경희대학교
KYUNG HEE UNIVERSITY

Workflow of Descriptive Analytics



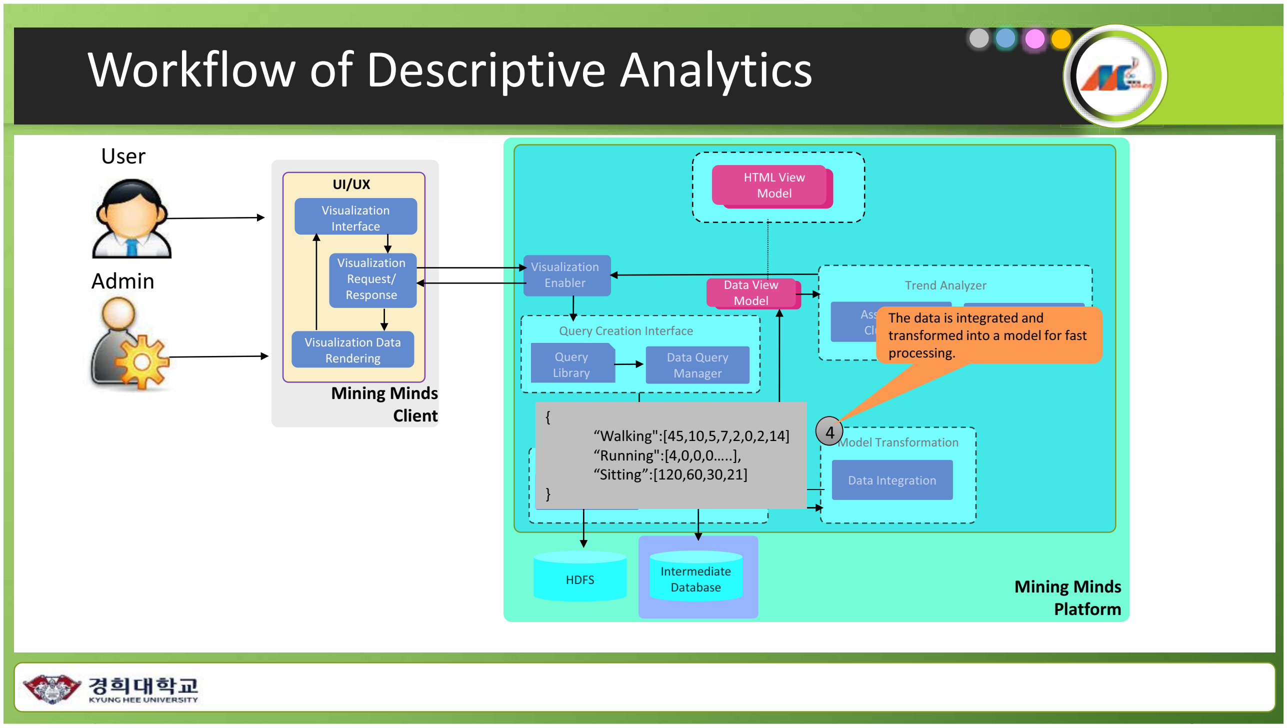
Return Attribute	Conditional Attributes		Temporal Attribute	
Running	Duration	>1m	Day	30
Sitting	Duration	>15 m	Day	30
Walking	Duration	>2m	Day	30

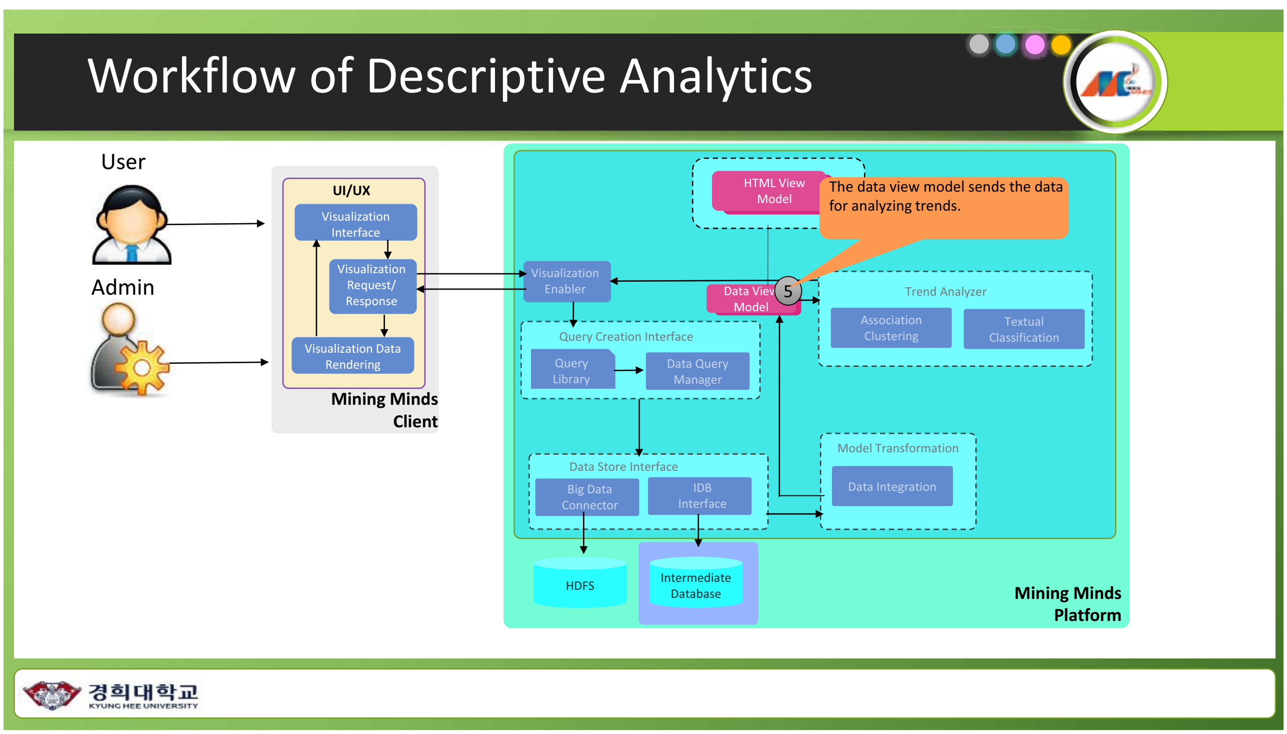
Workflow of Descriptive Analytics



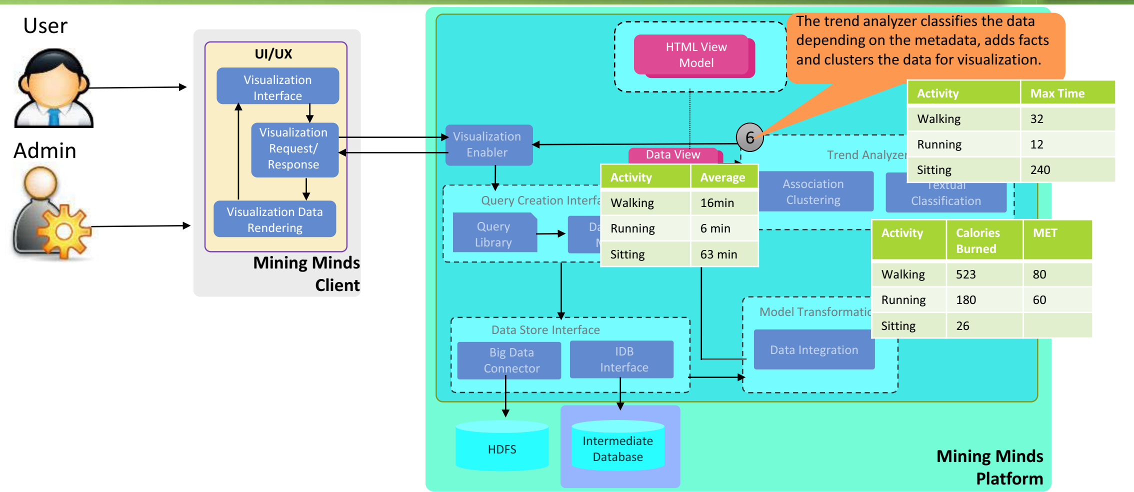
uid	ActId	DateTime	Activity	Duration	accX	accY	accZ
2	2	20/08/1513: 00:03	Walking	5Min	1213.23 1123	1213.23 1123	1213.23 1123
29	2	20/08/1510: 00:03	Walking	45m24s	14334.1 23	14334.1 23	14334.1 23
30	4	20/08/1511: 00:03	Sitting	2hr	1232	1232	1232
23	2	21/08/1518: 00:03	Walking	10min	1213.23 1123	1213.23 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

경희대학교
KYUNG HEE UNIVERSITY


Workflow of Descriptive Analytics
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
경희대학교
KYUNG HEE UNIVERSITY


Workflow of Descriptive Analytics
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
Big Data Connector
IDB Interface
Model Transformation
Data Integration
HDFS
Intermediate Database
Mining Minds Platform
경희대학교
KYUNG HEE UNIVERSITY

Workflow of Descriptive Analytics



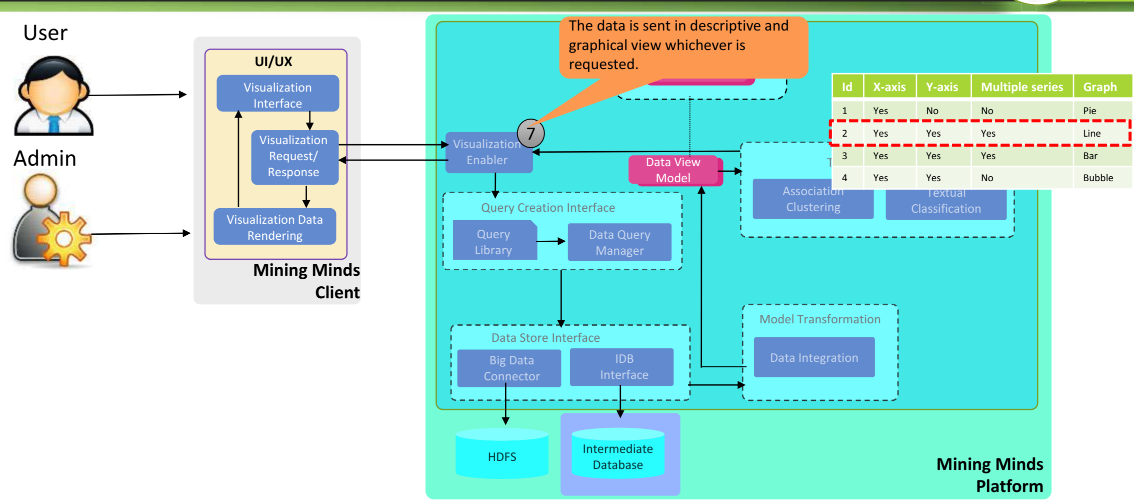
Activity	Max Time
Walking	32
Running	12
Sitting	240

Activity	Average
Walking	16min
Running	6 min
Sitting	63 min

Activity	Calories Burned	MET
Walking	523	80
Running	180	60
Sitting	26	

경희대학교 KYUNG HEE UNIVERSITY

Workflow of Descriptive Analytics



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

경희대학교 KYUNG HEE UNIVERSITY

Workflow of Descriptive Analytics

Tools /Technologies- Data Store Interface and Query Creation Interface

Options	Selected Tool

Options

- Apache Hive
- Pig Latin
- **MapReduce**
- MongoDB

Selected Tool

hadoop MapReduce

- Much Easier to Scale out.
- Compressed data could be queried
- Faster than most alternatives and mature

Tools /Technologies- Visualization, Model Transformation and Trend Analyzer

Options
- Java
- **PHP**

Hosting and Styling
- CSS
- Apache Webserver

Options
- **AmCharts Graph API**
- Anychart
- D3

php

- Fast, flexible, and free.
- Much faster to develop and PHP tends to have fewer issues when using shared hosting.

AMCHARTS

- Stable and been longer on marketing
- Theme Support

경희대학교
KYUNG HEE UNIVERSITY

Contributions

- 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

경희대학교
KYUNG HEE UNIVERSITY

MMv1.5 Implementation Challenges

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.

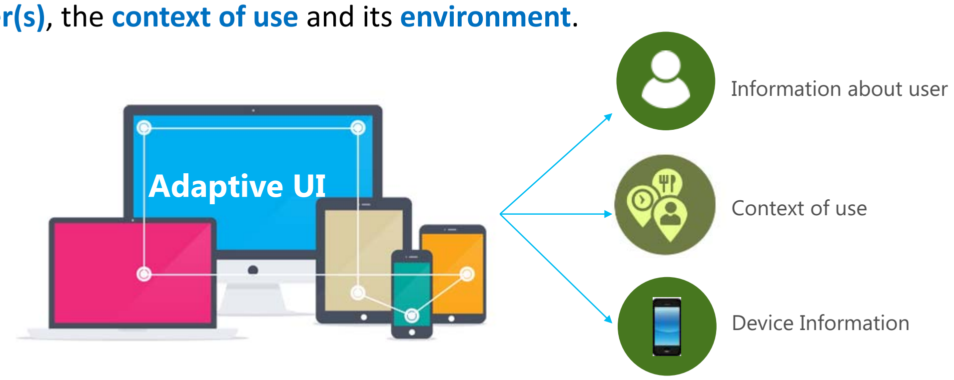
Conclusion

- 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.

Supporting Layer (SL 2.0)

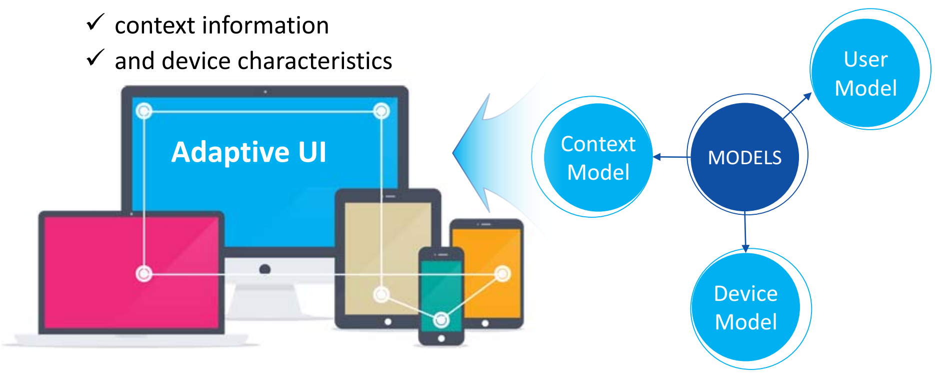
Introduction [1/2]

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**.



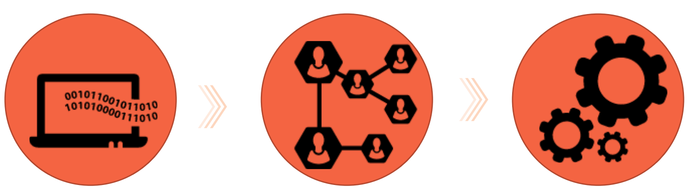
Introduction [2/2]

- In order to achieve a **satisfactory adaptation**, it is necessary to have several inputs such as
  - ✓ user characteristics model
  - ✓ context information
  - ✓ and device characteristics



Composition of adaptive systems

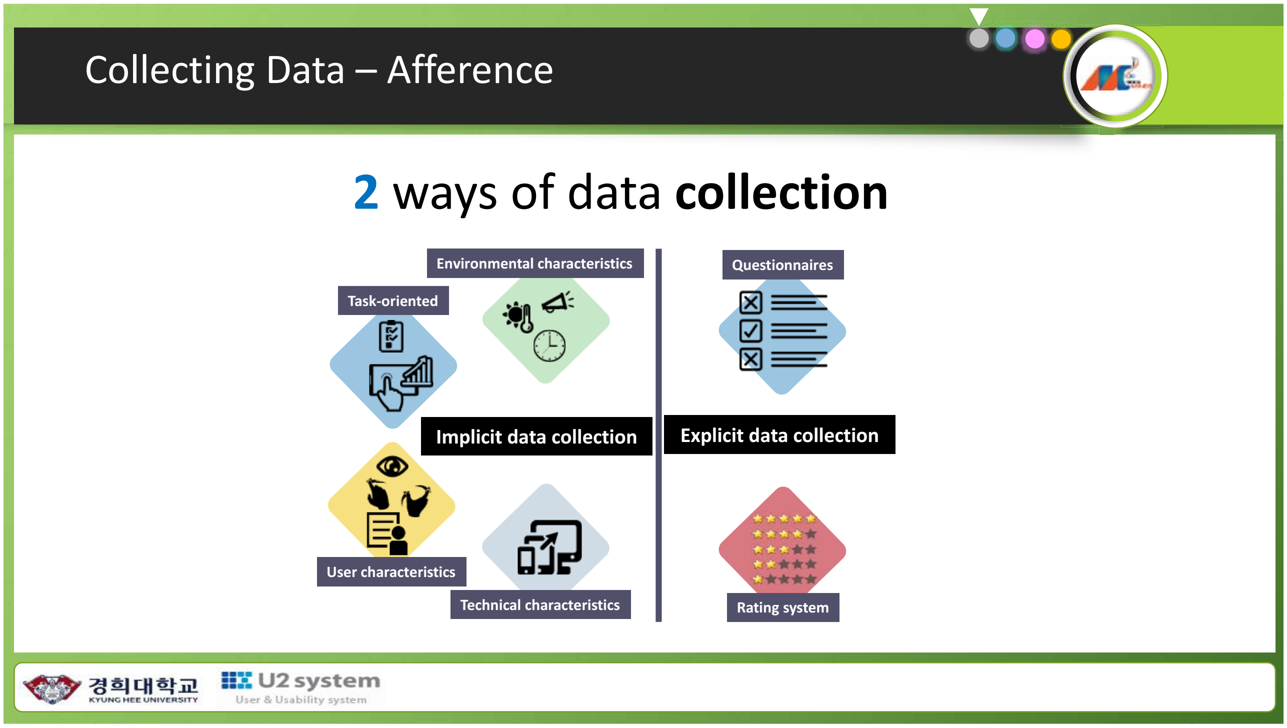
3 stages for adaptive UI

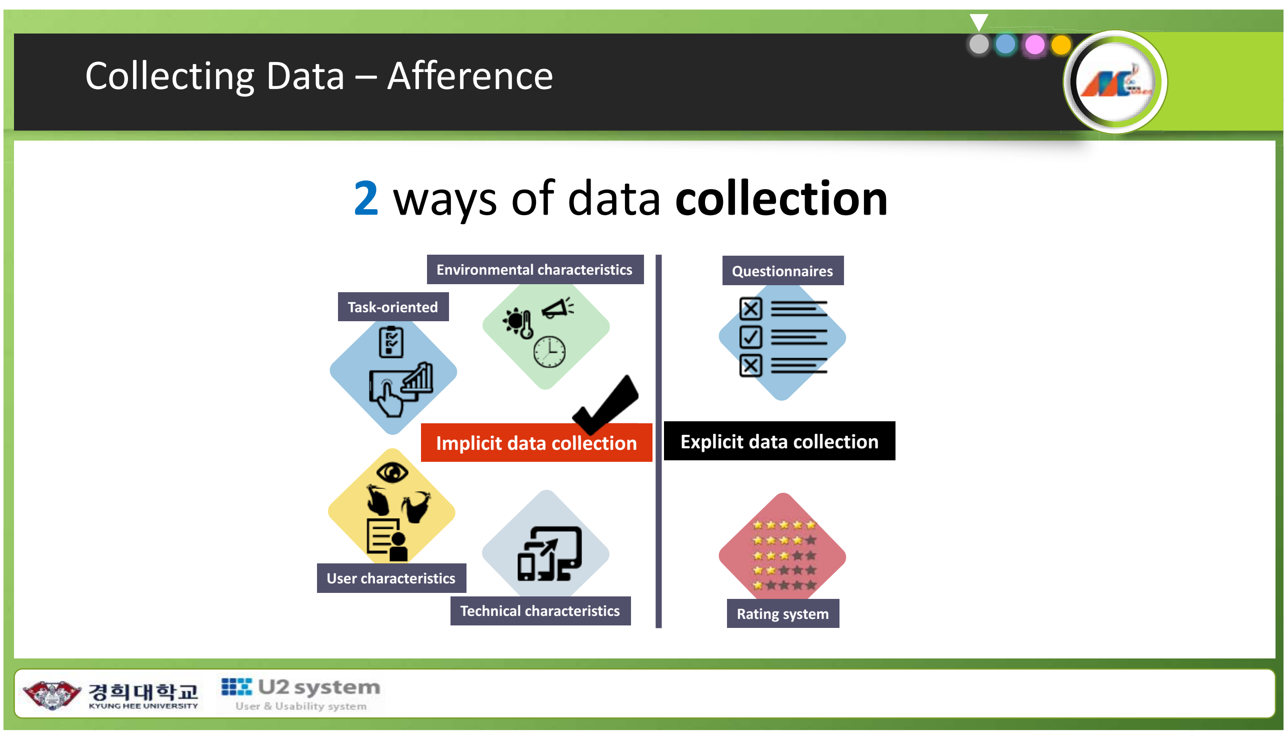


Afference
Collection of observational data about the user

Inference
Creating or **updating** a user model based on that data.

Efference
Deciding how to adapt the system behavior


Collecting Data – Afference
2 ways of data collection
Environmental characteristics
Task-oriented
Implicit data collection
User characteristics
Technical characteristics
Questionnaires
Explicit data collection
Rating system
경희대학교
KYUNG HEE UNIVERSITY
U2 system
User & Usability system

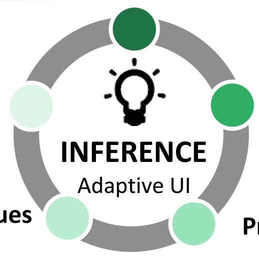

Collecting Data – Afference
2 ways of data collection
Environmental characteristics
Task-oriented
Implicit data collection
User characteristics
Technical characteristics
Questionnaires
Explicit data collection
Rating system
경희대학교
KYUNG HEE UNIVERSITY
U2 system
User & Usability system

Evaluating Data – Inference

Rule-based ✔

```
If ( Age==32 & Visual Acuity == 20/100)
  then font size=60dp
      :
```

Collaborative filtering



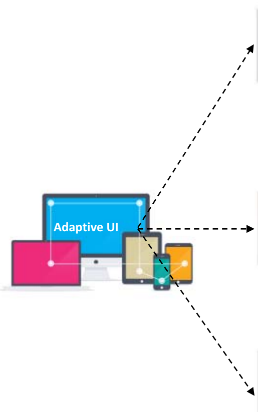
Algorithms ✔

```
• e.g. such as Approri algorithms,
LOSTNESS for measure of efficiency,
learning curves etc.
```

Machine learning techniques

Probabilistic methods

Implementation – Efference



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

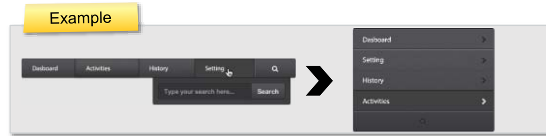
Presentation ✔

- ❑ It concern the design of the interface, e.g. font size or color, layout of user interface elements

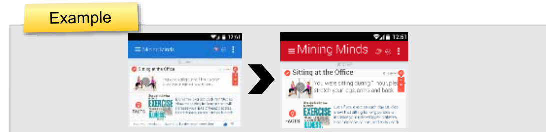
Content

- ❑ Adaptations that deal with the context of the situation occurred.

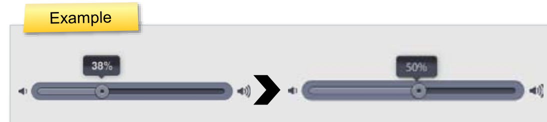
Example



Example



Example



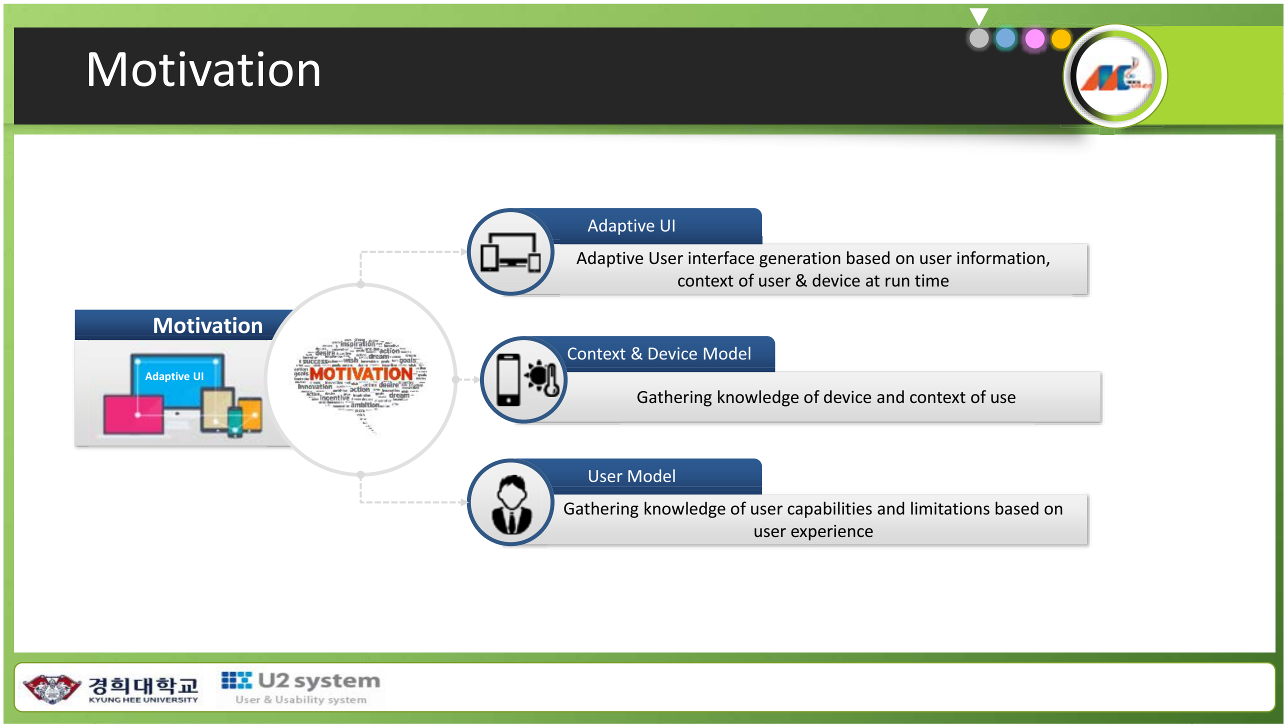
경희대학교 KYUNG HEE UNIVERSITY U2 system User & Usability system

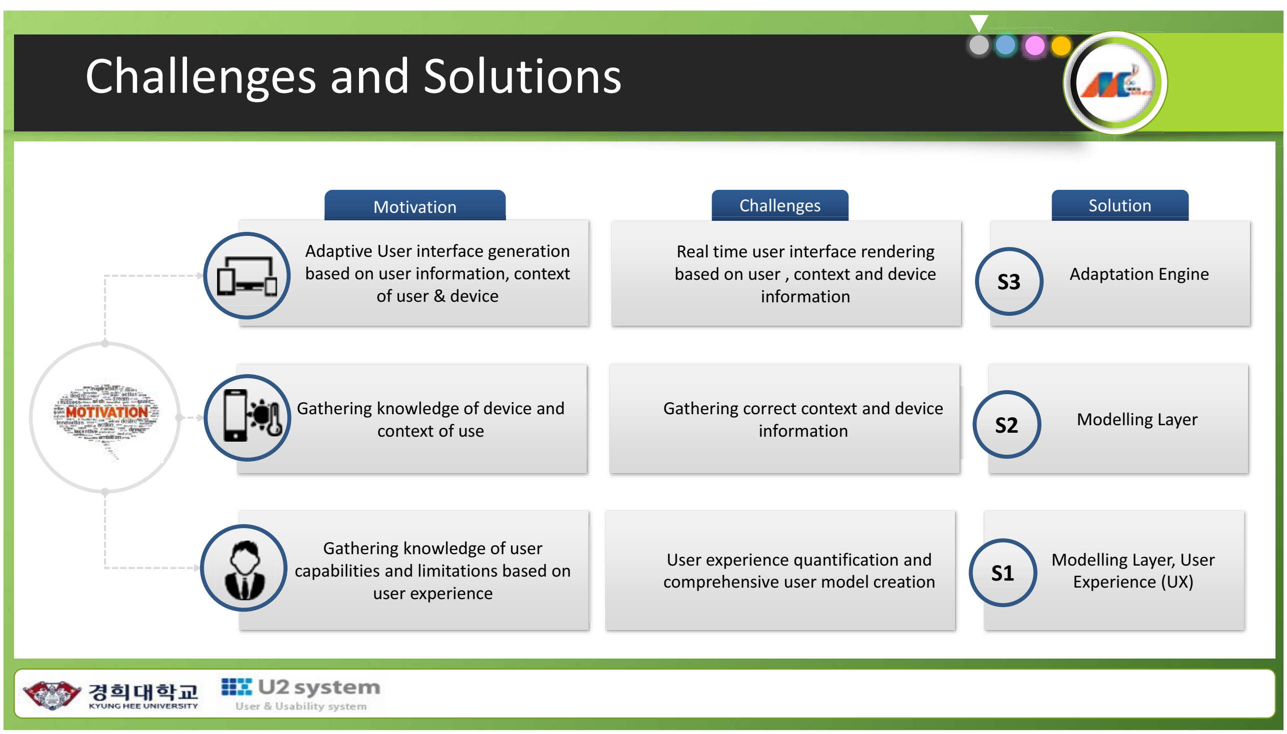
Overview of existing adaptive systems

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

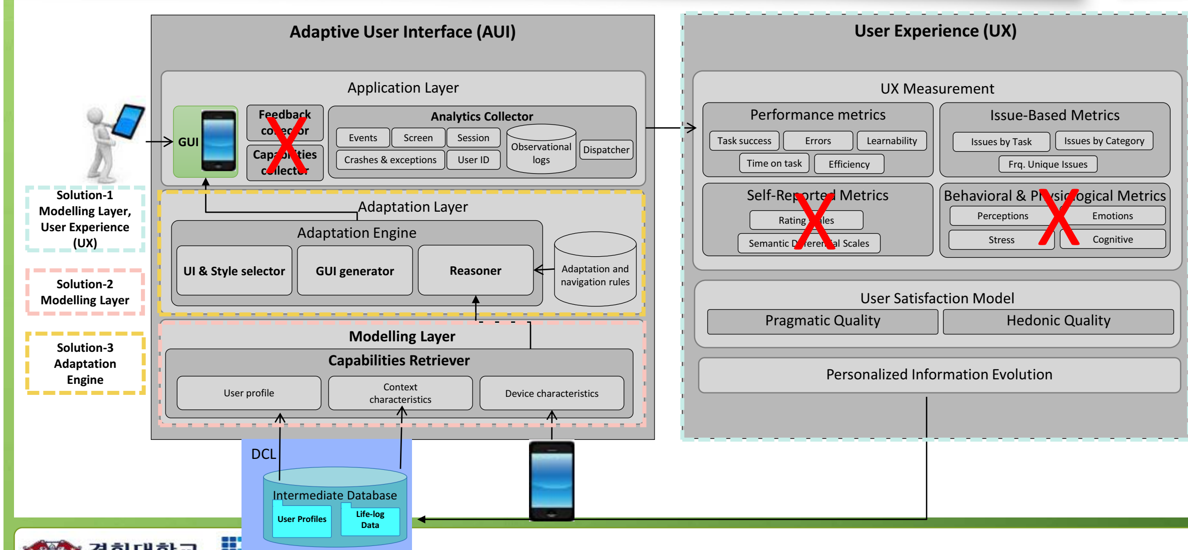
Limitations of existing Work

- 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
Adaptive UI
Adaptive User interface generation based on user information, context of user & device at run time
Motivation
Adaptive UI
MOTIVATION
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
U2 system
User & Usability system


Challenges and Solutions
Motivation
Challenges
Solution
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
MOTIVATION
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
U2 system
User & Usability system

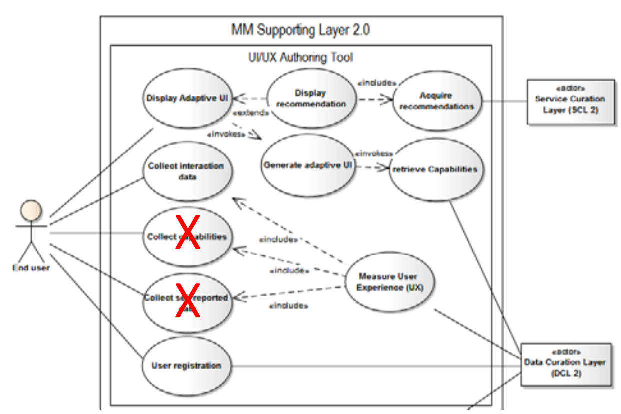
UI/UX Authoring Tool : Proposed Architecture



경희대학교 KYUNG HEE UNIVERSITY
User & Usability system

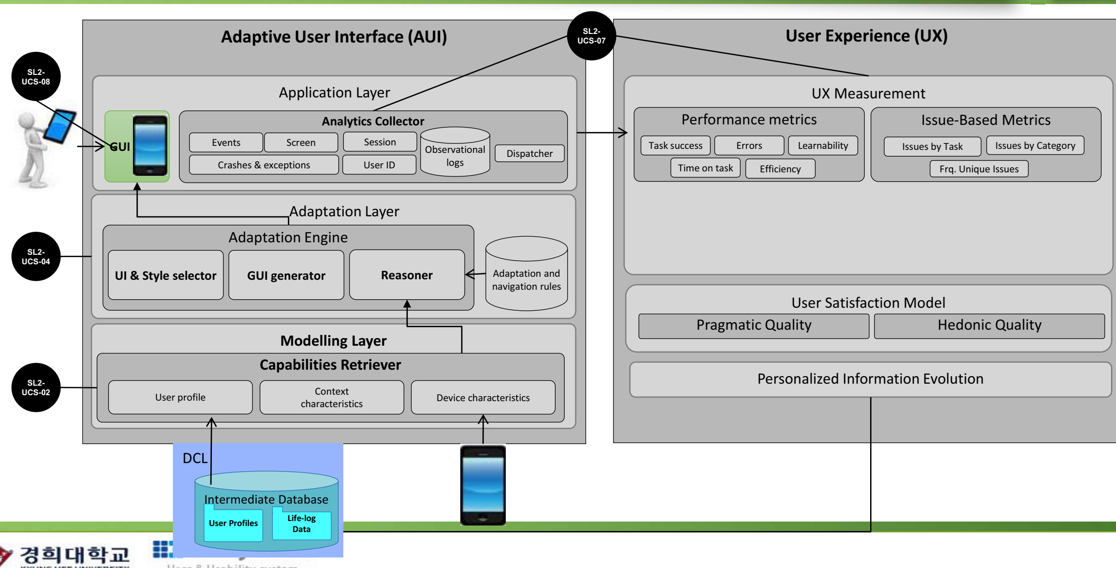
UI/UX Authoring Tool Implementation Scope – Use cases

Use Case #ID	Description
SL2-UCS-01	User registration
SL2-UCS-02	Retrieve capabilities for user interface adaptation
SL2-UCS-03	Mapping the user capability information into model
SL2-UCS-04	Adapt user interface based on user profile, context and device
SL2-UCS-05	User capabilities collection
SL2-UCS-06	Self-reporting user experience measurement
SL2-UCS-07	Collect and analyze observational data
SL2-UCS-08	Acquire Recommendations for displaying to end user



경희대학교 KYUNG HEE UNIVERSITY
U2 system
User & Usability system

UI/UX Authoring Tool : Refined Architecture



Adaptation Strategies

ADAPTATION STRATEGIES

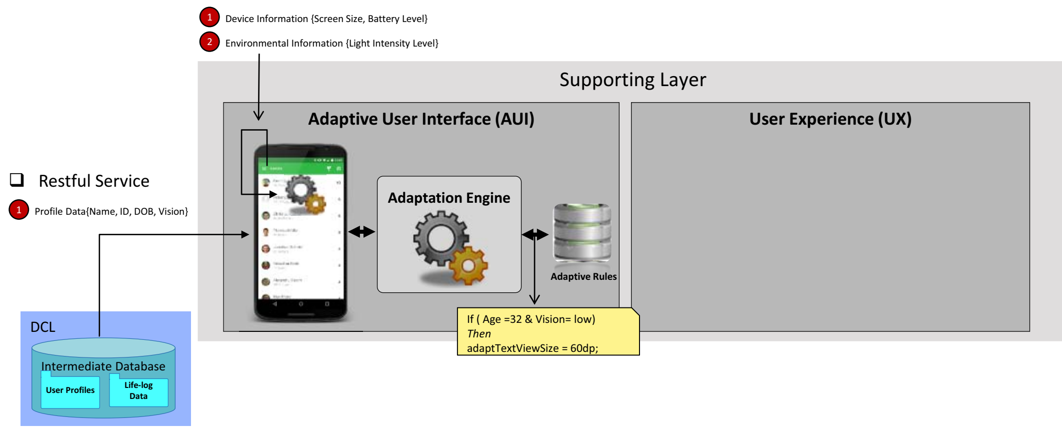
Frequent (on app load)

Based on user capabilities , limitations , environment information , and device information, the UI adapt accordingly

Non-frequent (Weekly/ monthly)

Based on user interaction data with app, to identify areas of improvement and maximize the user interaction by analyzing the user interaction with app

Strategy I: Frequent



경희대학교 KYUNG HEE UNIVERSITY U2 system User & Usability system

Methodology : Capabilities Retriever

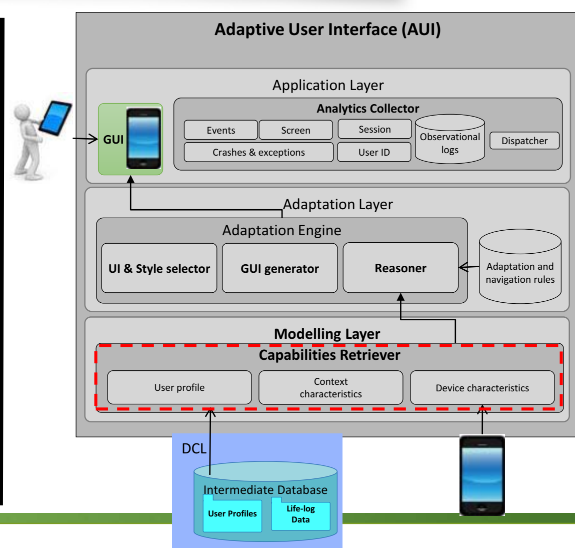
Methodology Description

Use-case (SL2-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)

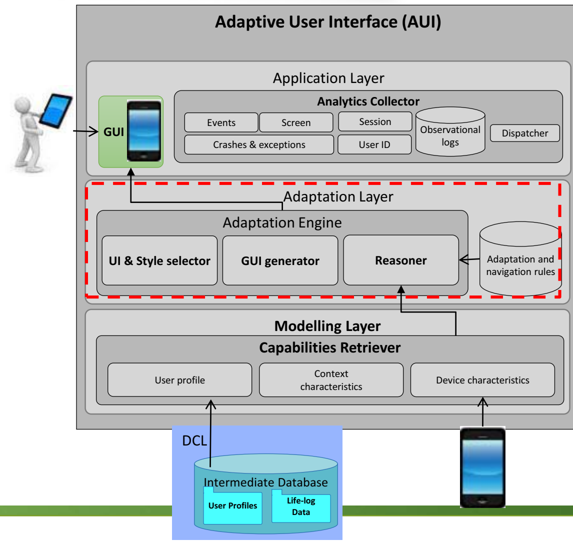


Methodology : Adaptation Layer

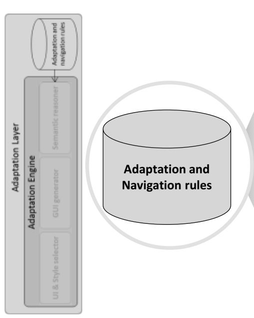
Methodology Description

DCL2-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



Adaptation & navigation Rules



Pre-adaptation Rules

These rules are designed to check the values that are needed for final adaptive UI

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 }
```

The usability Rules

These rules determine if the interaction with the adapted user inter- face might be considered enough.

Example

```
checkTaskEfficiency

If(TaskCompletionRate = 65 && TaskTime==1.5 )
{ efficiency = 71; }
```

Adaptation Rules

Depending on the pre-adaptation and usability checked rules, different rules are triggered; which bringing the final adaptation to the device

Example

```
TextView

If ( Age =32 & Vision= low) {
adaptTextViewSize = 60dp;
}
```

경희대학교 KYUNG HEE UNIVERSITY U2 system User & Usability system

Adaptation Rule Format

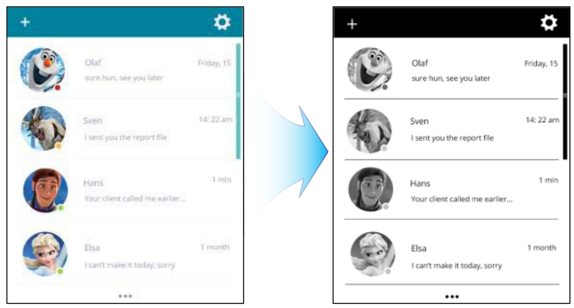
Rule Format: Event / Condition / Action

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 U2 system User & Usability system

Pre-adaptation Rules

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 U2 system User & Usability system

Usability Rules

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 U2 system User & Usability system

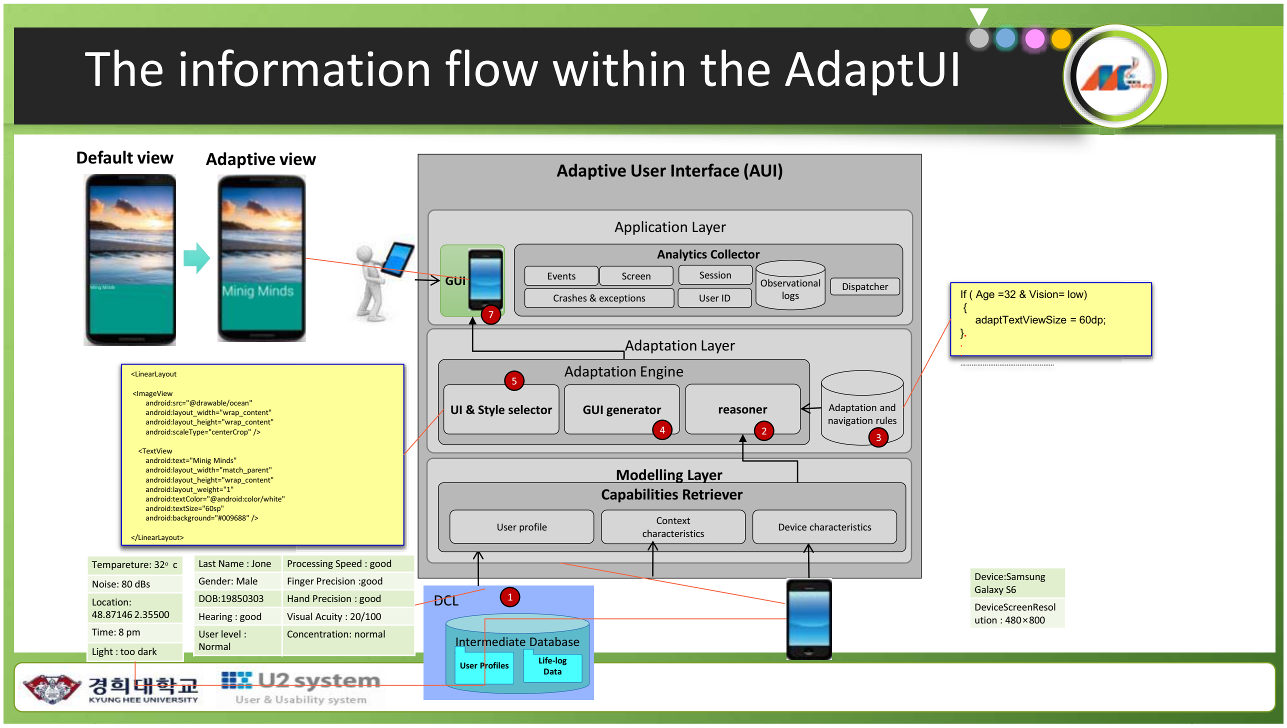
Adaptation Rules

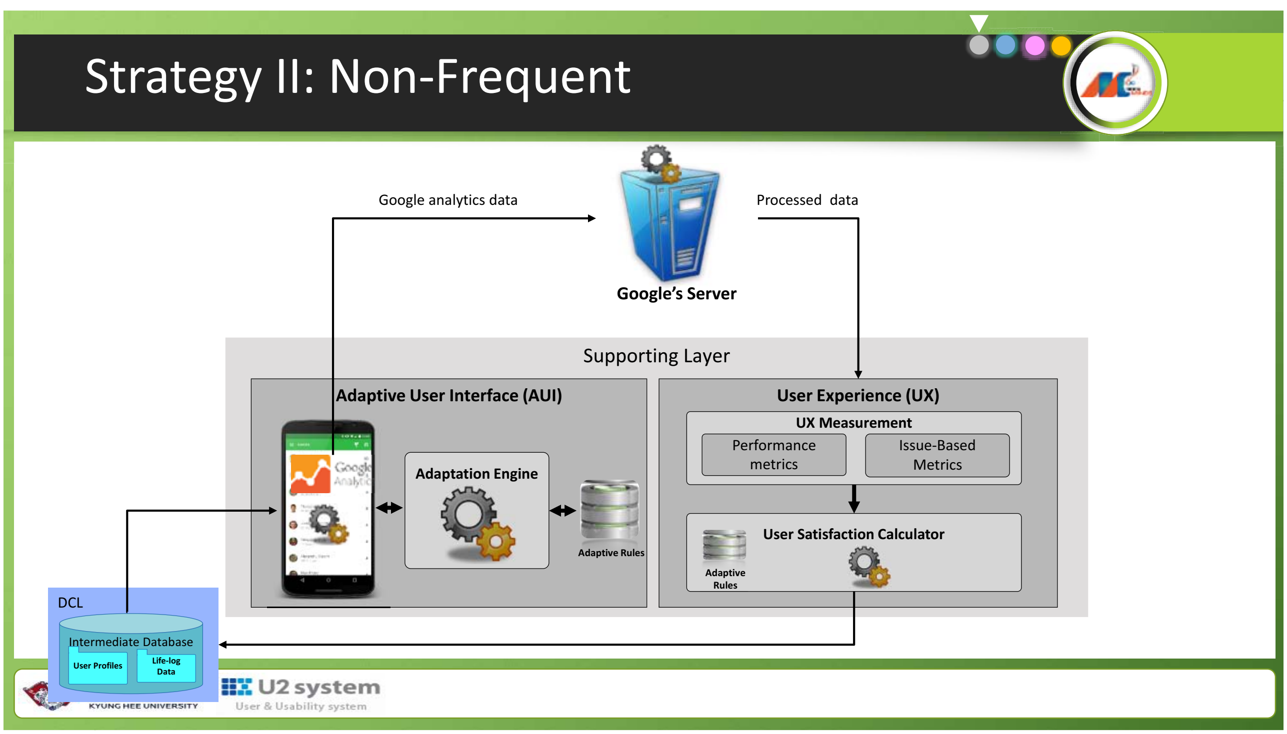
Adaptation Rules

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 U2 system User & Usability system


The information flow within the AdaptUI
Default view
Adaptive view
Minig Minds
Adaptive User Interface (AUI)
Application Layer
Analytics Collector
Events
Screen
Session
Observational logs
Dispatcher
Crashes & exceptions
User ID
GUI
Adaptation Layer
Adaptation Engine
UI & Style selector
GUI generator
reasoner
Adaptation and navigation rules
Modelling Layer
Capabilities Retriever
User profile
Context characteristics
Device characteristics
If (Age =32 & Vision= low)
{
adaptTextViewSize = 60dp;
}.
Temparature: 32° c
Noise: 80 dBs
Location: 48.87146 2.35500
Time: 8 pm
Light : too dark
Last Name : Jone
Gender: Male
DOB:19850303
Hearing : good
User level : Normal
Processing Speed : good
Finger Precision :good
Hand Precision : good
Visual Acuity : 20/100
Concentration: normal
DCL
Intermediate Database
User Profiles
Life-log Data
Device:Samsung Galaxy S6
DeviceScreenResolution : 480×800
경희대학교
KYUNG HEE UNIVERSITY
U2 system
User & Usability system

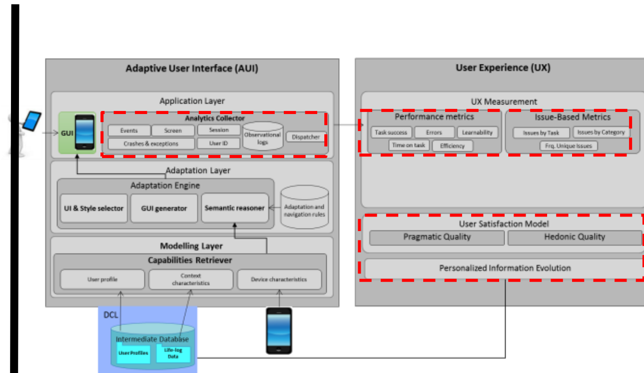

Strategy II: Non-Frequent
Google analytics data
Processed data
Google's Server
Supporting Layer
Adaptive User Interface (AUI)
Adaptation Engine
Adaptive Rules
User Experience (UX)
UX Measurement
Performance metrics
Issue-Based Metrics
User Satisfaction Calculator
Adaptive Rules
DCL
Intermediate Database
User Profiles
Life-log Data
KYUNG HEE UNIVERSITY
U2 system
User & Usability system

Methodology : Analytics collector and UX measurement

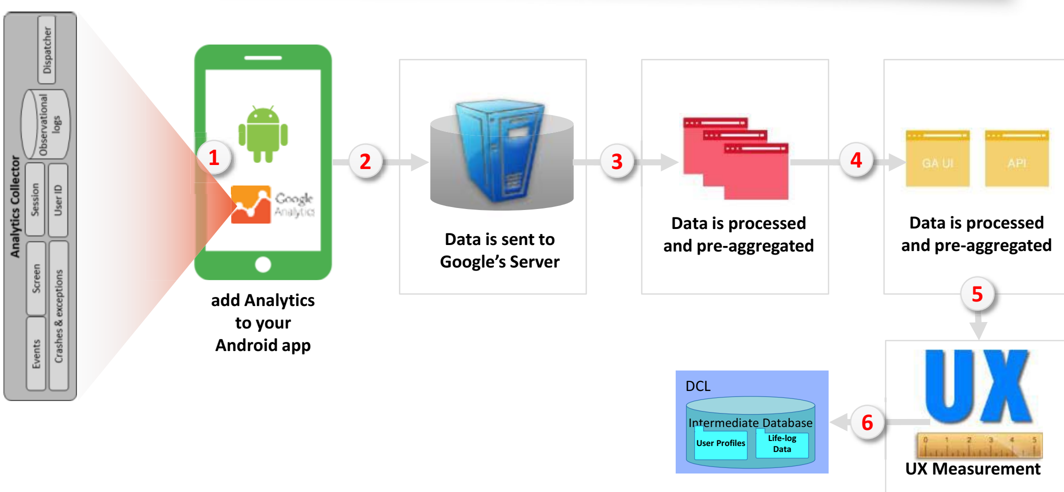
Methodology Description

Use-case (SL2-UCS-05): *Collect and analyze observational data*

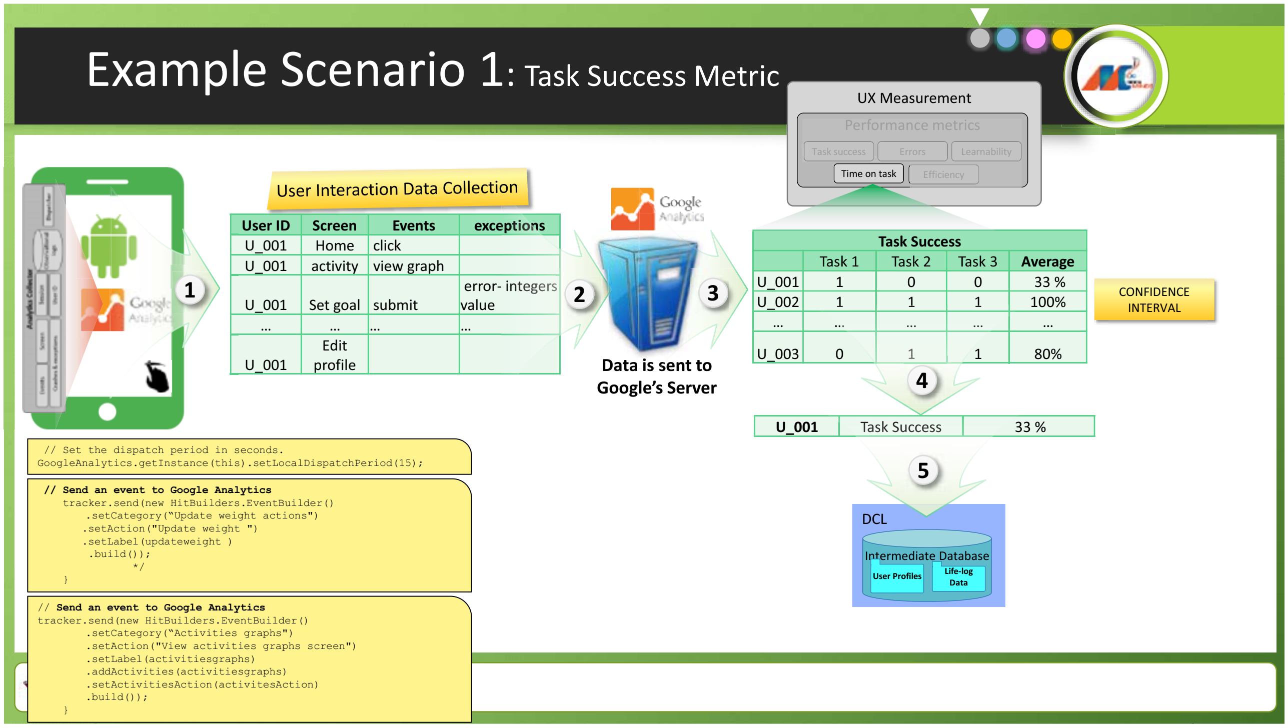
- **Objective**
  - ➢ To identify areas of improvement and maximize the user interaction by analyzing the user interaction with app.
- **Methodology**
  - ➢ *Communication*: end user interaction with MM app
  - ➢ **Analytics collector and UX measurment**
    1. **Input**: receives *request from end user*
    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.

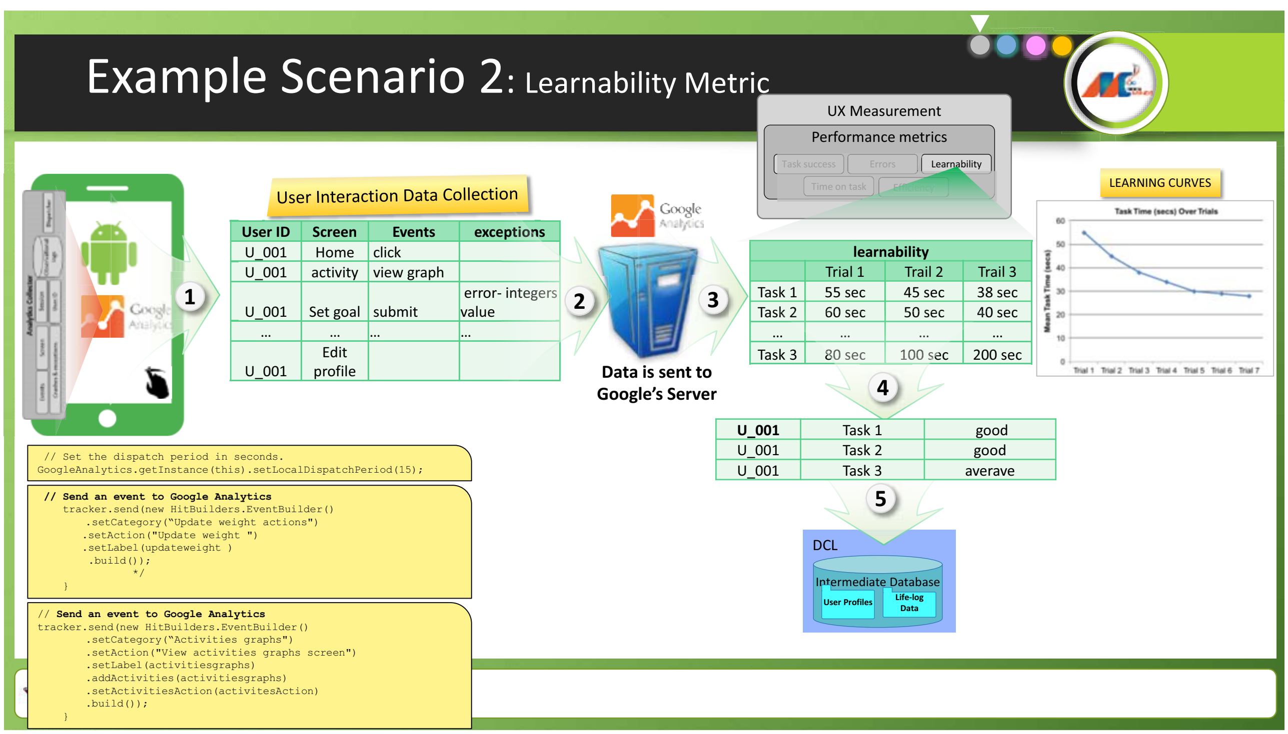


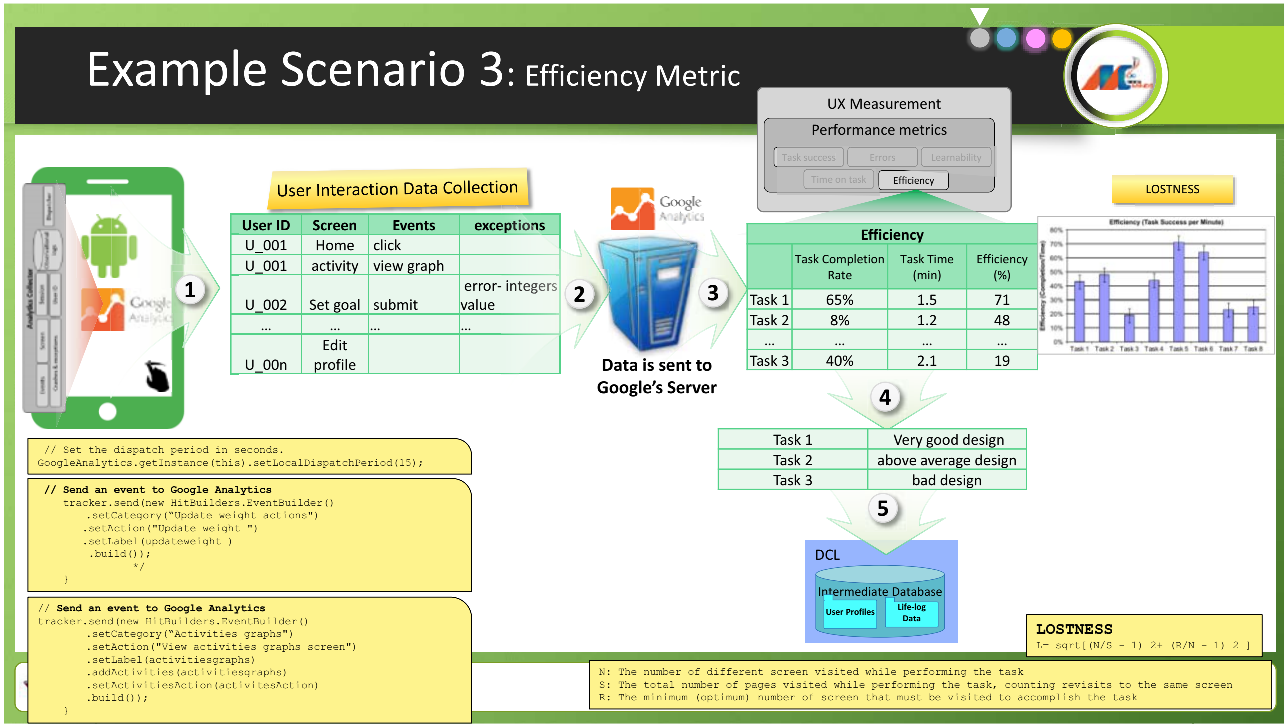
Analytics Collector Work flow

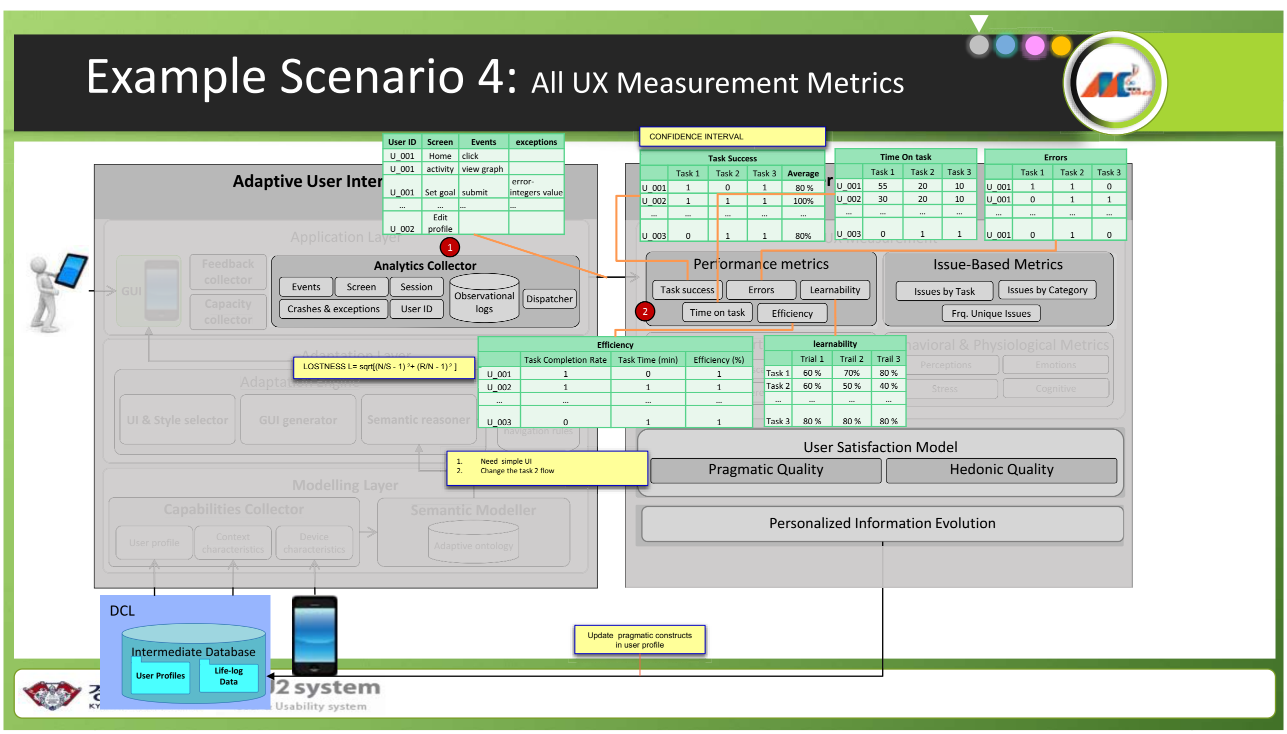


경희대학교 KYUNG HEE UNIVERSITY U2 system User & Usability system


Example Scenario 1: Task Success Metric
UX Measurement
Performance metrics
Task success
Errors
Learnability
Time on task
Efficiency
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
Google Analytics
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
UX Measurement
Performance metrics
Task success
Errors
Learnability
Time on task
Efficiency
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
Google Analytics
Data is sent to Google's Server
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
LEARNING CURVES
Task Time (secs) Over Trials
Mean Task Time (secs)
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
// 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
UX Measurement
Performance metrics
Task success
Errors
Learnability
Time on task
Efficiency
LOSTNESS
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
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
// 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());
}
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


Example Scenario 4: All UX Measurement Metrics
Adaptive User Inter
Application Layer
Analytics Collector
Events
Screen
Session
Crashes & exceptions
User ID
Observational logs
Dispatcher
CONFIDENCE INTERVAL
Task Success
Time On task
Errors
Performance metrics
Task success
Errors
Learnability
Time on task
Efficiency
Issue-Based Metrics
Issues by Task
Issues by Category
Frq. Unique Issues
LOSTNESS L= sqrt[(N/S - 1) 2+ (R/N - 1) 2]
Efficiency
learnability
1. Need simple UI
2. Change the task 2 flow
User Satisfaction Model
Pragmatic Quality
Hedonic Quality
Personalized Information Evolution
Modelling Layer
DCL
Intermediate Database
User Profiles
Life-log Data
Update pragmatic constructs in user profile

Tools and Technologies

- **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

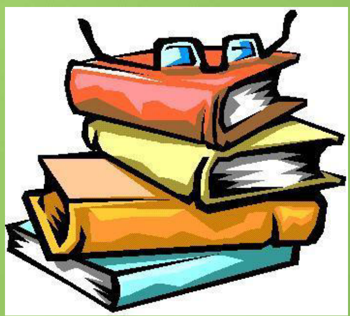
- Comprehensive adaptive UI ontology development
- Rules for adaptive UI
- User experience measurement toolkit development

경희대학교 KYUNG HEE UNIVERSITY U2 system User & Usability system

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

경희대학교 KYUNG HEE UNIVERSITY U2 system User & Usability system



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

경희대학교
KYUNG HEE UNIVERSITY

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.

경희대학교
KYUNG HEE UNIVERSITY

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

경희대학교 KYUNG HEE UNIVERSITY POSTECH 포항공과대학교

References- Lifelog Monitoring and Prediction

- **[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

- **[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

[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

- 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

- 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

[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

[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

- **[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

- [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]

- **[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]

- [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

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

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

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/

Section 4-D

System Implementation

Layers Implementation

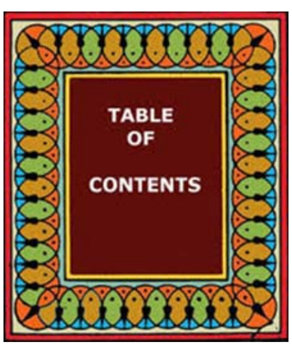
Mining Minds Platform

Principal Investigator:
Professor Sungyoung Lee

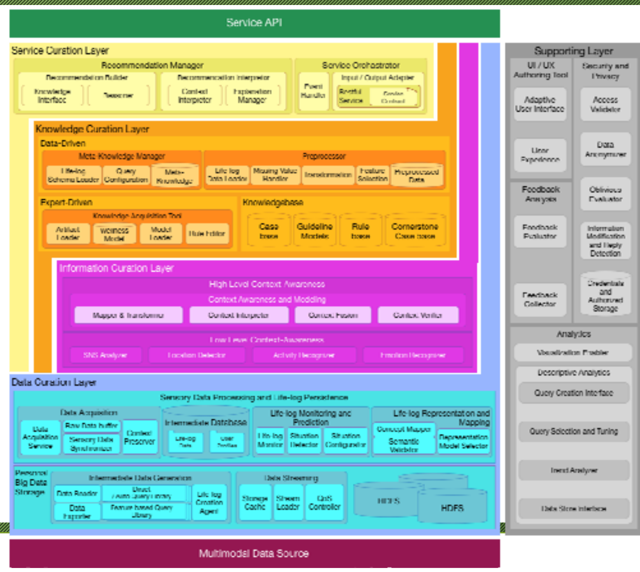
December 18th, 2015
Kyung Hee University, Korea

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

Contents



Architecture of Mining Minds Version 2.0



Data Curation Layer (DCL 2.0)

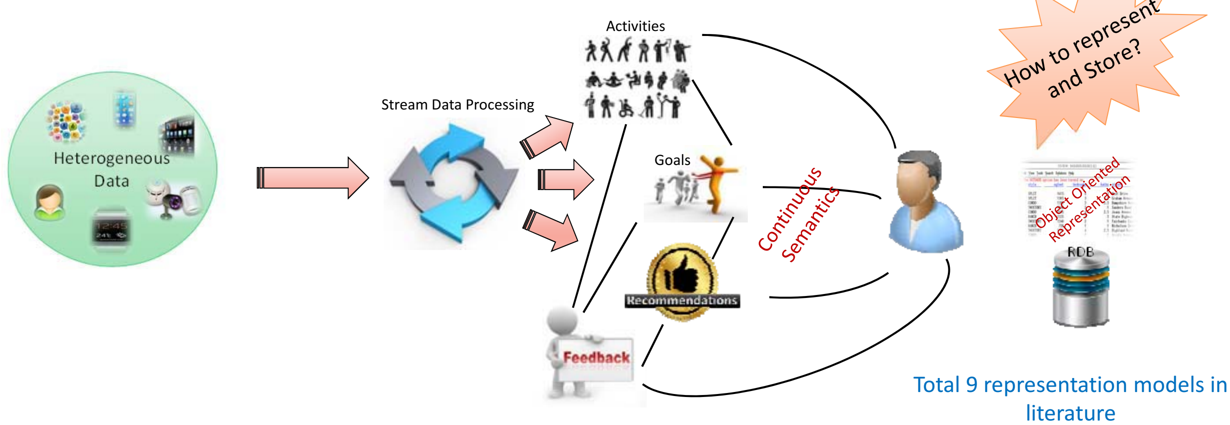
Principal Investigator:
Professor Sungyoung Lee

Life-log Mapping and Representation

Taqdir Ali

Purpose of Life-log Mapping and Representation

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



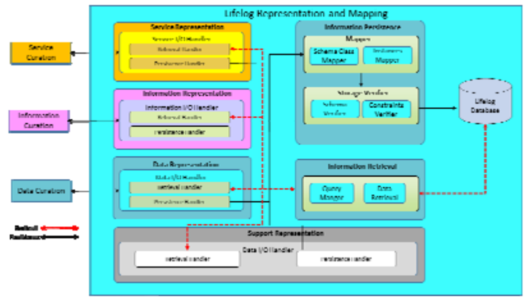
Total 9 representation models in literature

Completed and Future tasks for MM V1.0 and V1.5

MM V2.0_DCL_T5_V0.1_20151003_Taqdir Ali

- Well designed Database schema
- Restful Web services for each layer
- ORM (Object-Relational Mapping) Library

- Future Tasks as maintenance
  - Changes in Database schema
  - Changes in ORM
  - Changes in Restful services
  - Changes in SQL Stored Procedures



- **These changes will be implemented according to requirements**
- **Team members need to discuss new requirements**
- **Kindly provide new requirements in next week**

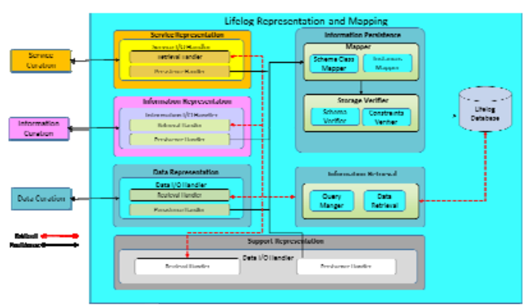
경희대학교
KYUNG HEE UNIVERSITY

Completed and Future tasks for MM V1.0 and V1.5

MM V2.0_DCL_T5_V0.2_20151010_Taqdir Ali

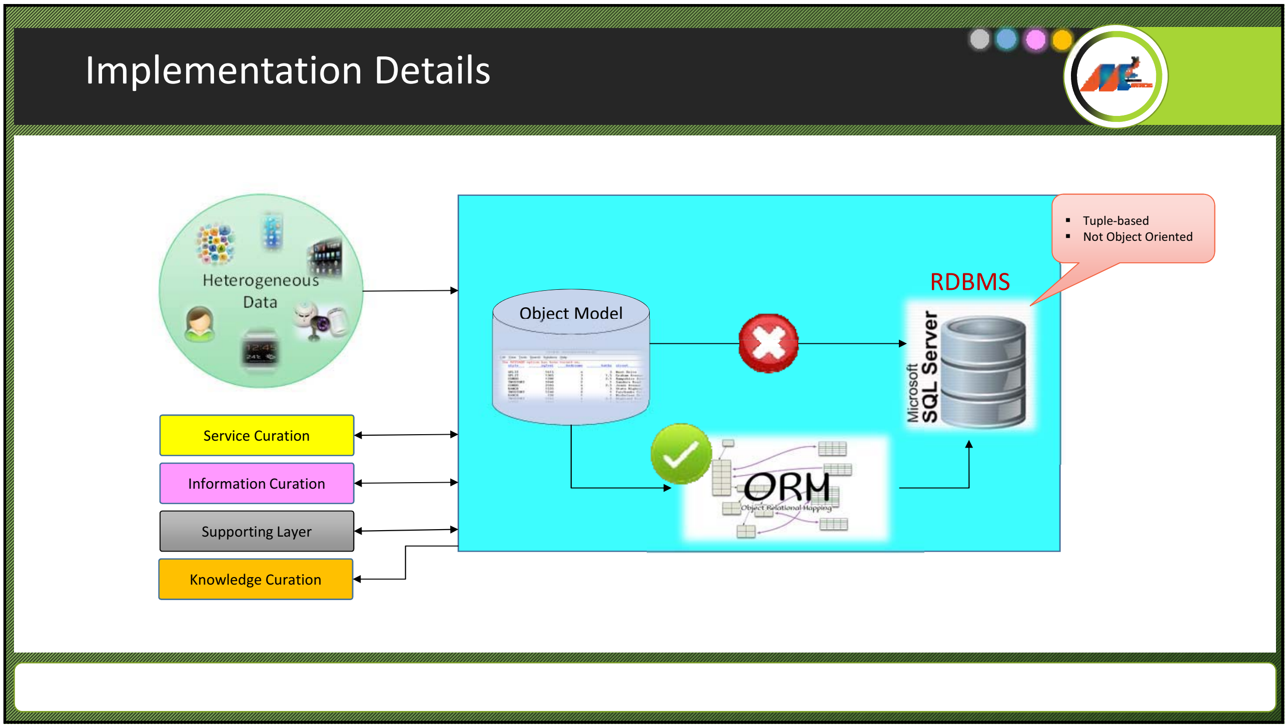
- Requirements meeting with SL for Individual and group Analytics
  - Analytics of total number of User Recommendations
  - Analytics of total number of users with all profile information
  - Analytics of all weekly and monthly user's activities
  - Analytics of user's location
  - Analytics of accumulative user's activities

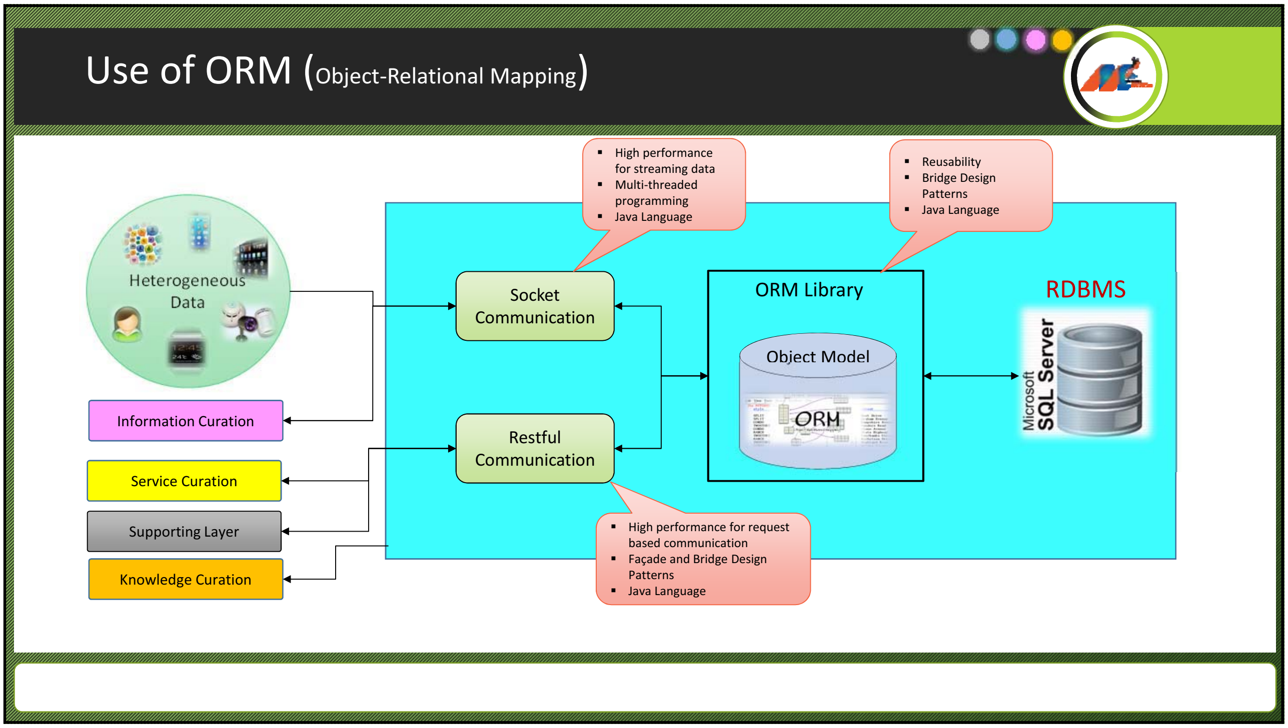
- Future Tasks as maintenance
  - Changes in Database schema
  - Changes in ORM
  - Changes in Restful services
  - Changes in SQL Stored Procedures



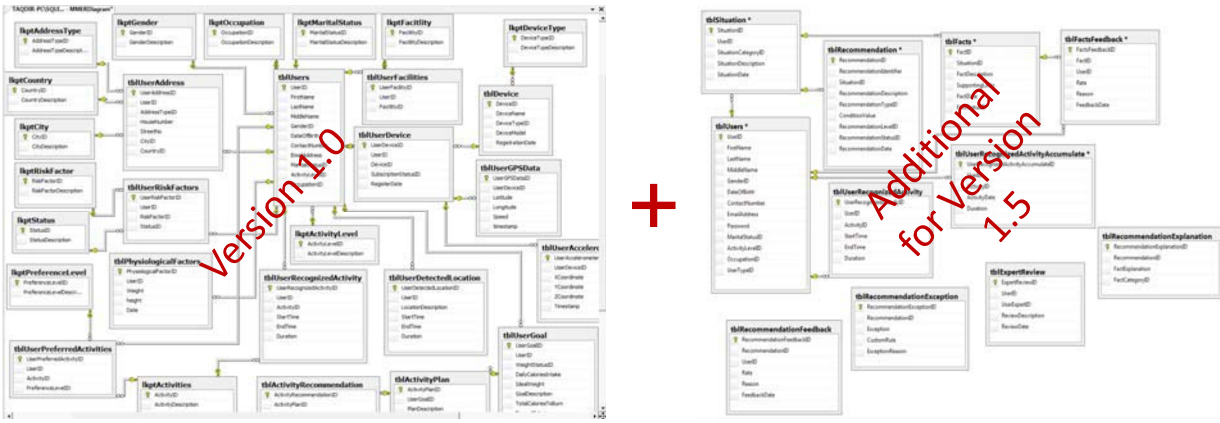
- **Requirements of Supporting Layers are collected**
- **Other team members need to discuss new requirements**
- **Kindly provide new requirements in next week**

경희대학교
KYUNG HEE UNIVERSITY


Implementation Details
Heterogeneous Data
Service Curation
Information Curation
Supporting Layer
Knowledge Curation
Object Model
RDBMS
Tuple-based
Not Object Oriented
Microsoft SQL Server
ORM

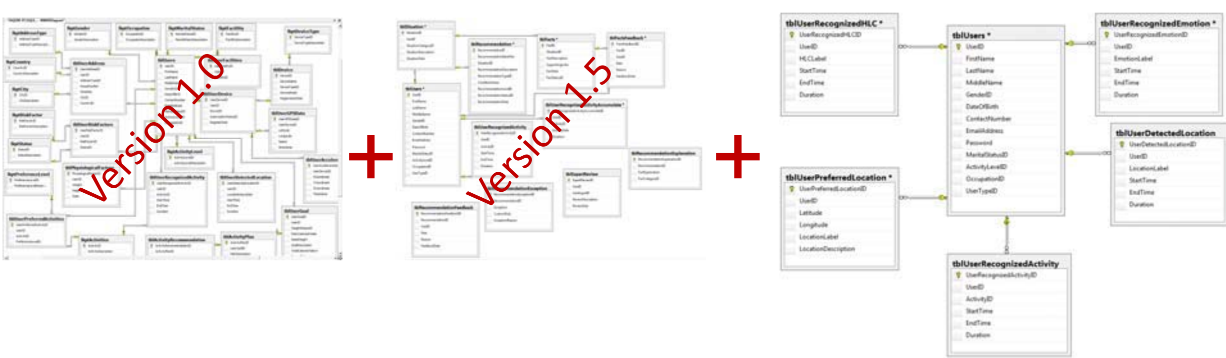

Use of ORM (Object-Relational Mapping)
Heterogeneous Data
Information Curation
Service Curation
Supporting Layer
Knowledge Curation
High performance for streaming data
Multi-threaded programming
Java Language
Reusability
Bridge Design Patterns
Java Language
Socket Communication
Restful Communication
ORM Library
Object Model
RDBMS
Microsoft SQL Server
High performance for request based communication
Façade and Bridge Design Patterns
Java Language

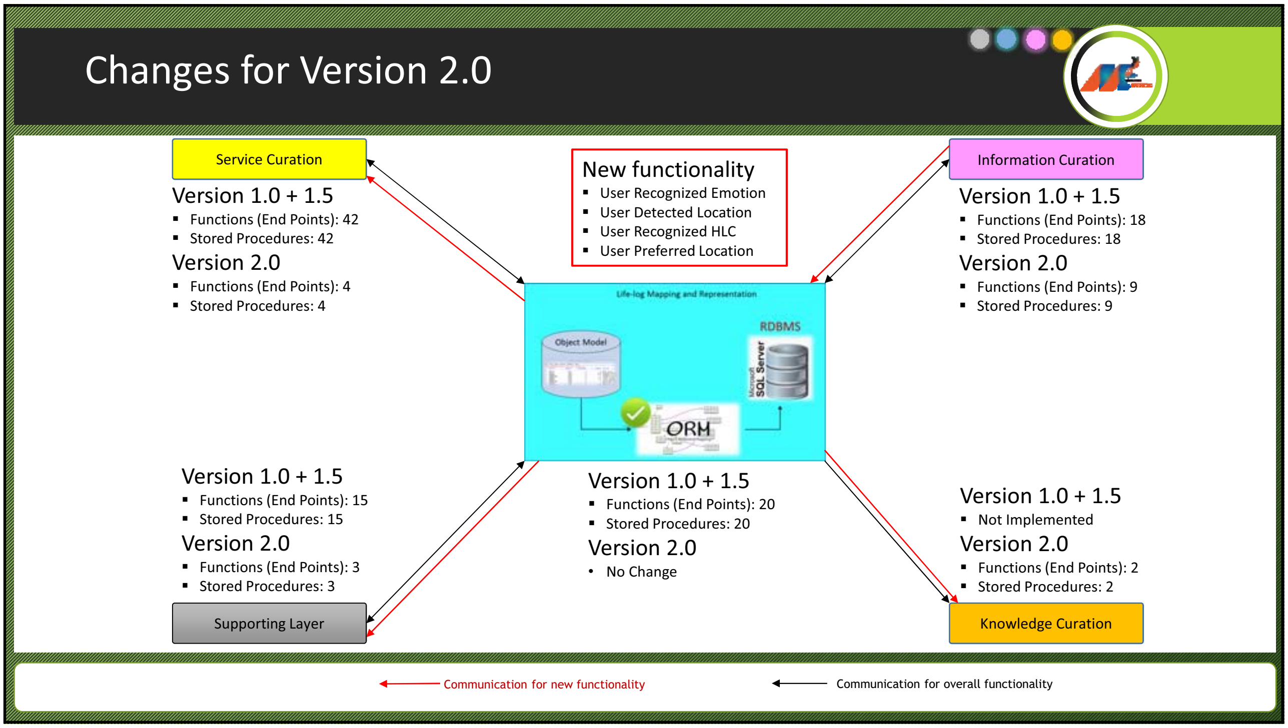
Intermediate Database v1.5

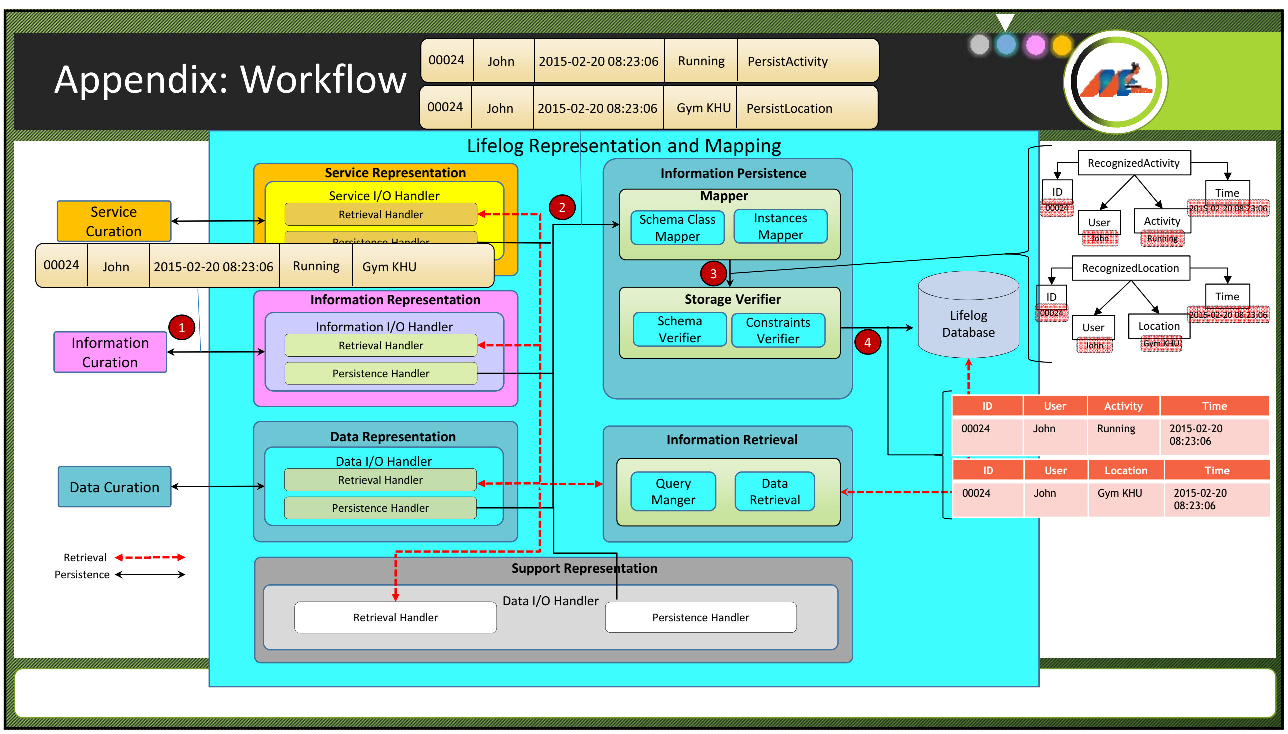


Intermediate Database v2.0

Additional for Version 2.0




Changes for Version 2.0
Service Curation
Version 1.0 + 1.5
Functions (End Points): 42
Stored Procedures: 42
Version 2.0
Functions (End Points): 4
Stored Procedures: 4
New functionality
User Recognized Emotion
User Detected Location
User Recognized HLC
User Preferred Location
Information Curation
Version 1.0 + 1.5
Functions (End Points): 18
Stored Procedures: 18
Version 2.0
Functions (End Points): 9
Stored Procedures: 9
RDBMS
ORM
Version 1.0 + 1.5
Functions (End Points): 15
Stored Procedures: 15
Version 2.0
Functions (End Points): 3
Stored Procedures: 3
Supporting Layer
Version 1.0 + 1.5
Functions (End Points): 20
Stored Procedures: 20
Version 2.0
No Change
Version 1.0 + 1.5
Not Implemented
Version 2.0
Functions (End Points): 2
Stored Procedures: 2
Knowledge Curation
Communication for new functionality
Communication for overall functionality


Appendix: Workflow
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
Information Retrieval
Query Manger
Data Retrieval
Support Representation
Data I/O Handler
Retrieval Handler
Persistence Handler
Lifelog Database
RecognizedActivity
ID
User
Activity
Time
RecognizedLocation
ID
User
Location
Time
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

Big Data Storage

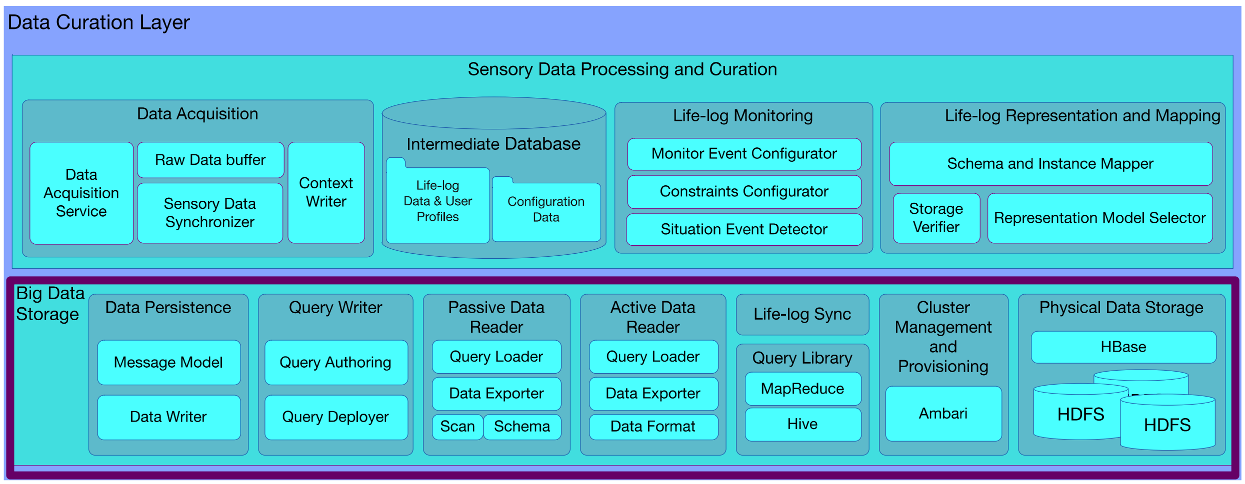
Usman Akhtar

Outline

In Today's presentation we will focus on the following:

1. Weekly Progress
2. Discussion on New Big Data Storage Architecture
3. Functional diagram & Scenarios
4. Future Plan

DCL V2.0 Architecture



Week 1: Progress

MM V2.0_DCL_T5_V0.1_20151003_Usman Akhtar

1. **Problem:** How to extract the specific data from HDFS?
2. **Solution:** Proposed a candidate solution

APACHE HBASE

3. **Implementation: Linux (Ubuntu), Single Cluster.**
4. **Main Challenges:** Version Compatibility issues between HDFS and Hbase

Week 1: Future Plan

MM V2.0_DCL_T5_V0.1_20151003_Usman Akhtar

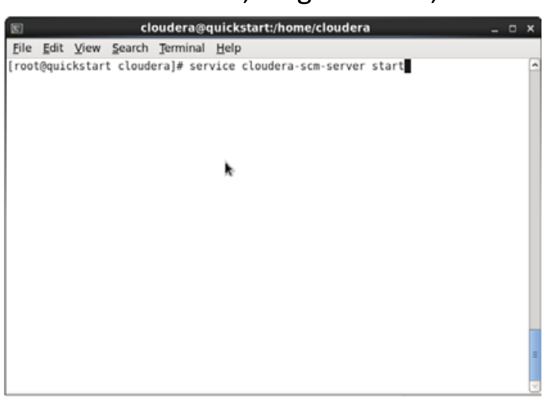
- Integrate the Cloudera Manager 5.4.7

Fig. Showing overall working plan.

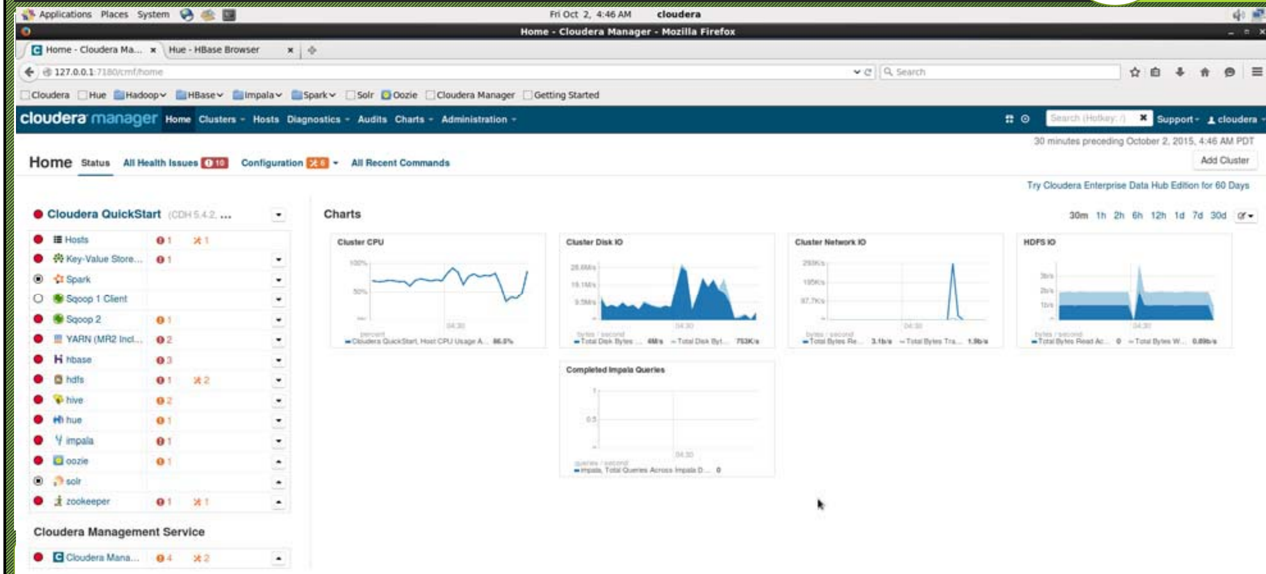
Week 2: Progress

MM V2.0_DCL_T5_V0.1_20151003_Usman Akhtar

- Installation Environment: Linux, Single Cluster, Cloudera Manager



Week 2: Progress



Week 2: Progress

MM V2.0_LayerName_T5_V0.2_151008_MemberName

- Challenges while working with Cloudera Manager

1. Consume lot of resources and very slow response time
2. The free version does not support other packages.

Week 2: Future Plan

MM V2.0_LayerName_T5_V0.2_151008_MemberName

- Two Main Tasks

1. Data extraction from HBASE
2. Simulate Development environment

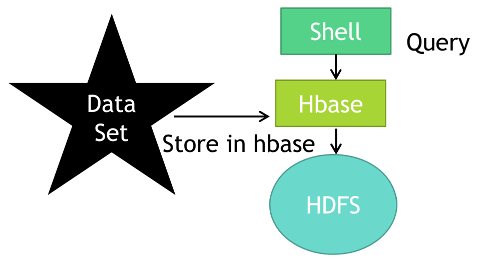


Fig. Single Cluster Implementation

Week 3: Progress

MM V2.0_LayerName_T5_V0.2_151008_MemberName

- **Handling Big Data with HBase**
  - Finally we are able to use Hbase v1.2 on HDFS v2.5.

- **During shell session we have performed the following operation:**
  - Create table
  - List the available tables in Hbase
  - Add blog entry and retrieve that entry
  - Scan the table

Week 3: Progress

MM V2.0_LayerName_T5_V0.2_151008_MemberName

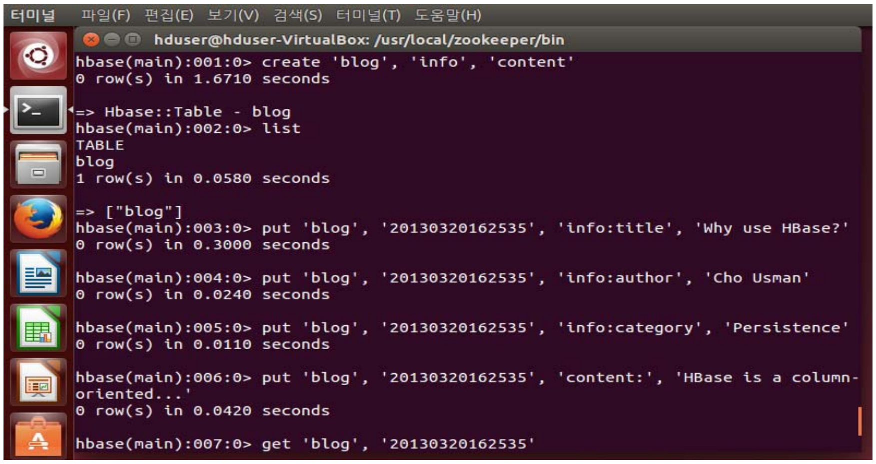
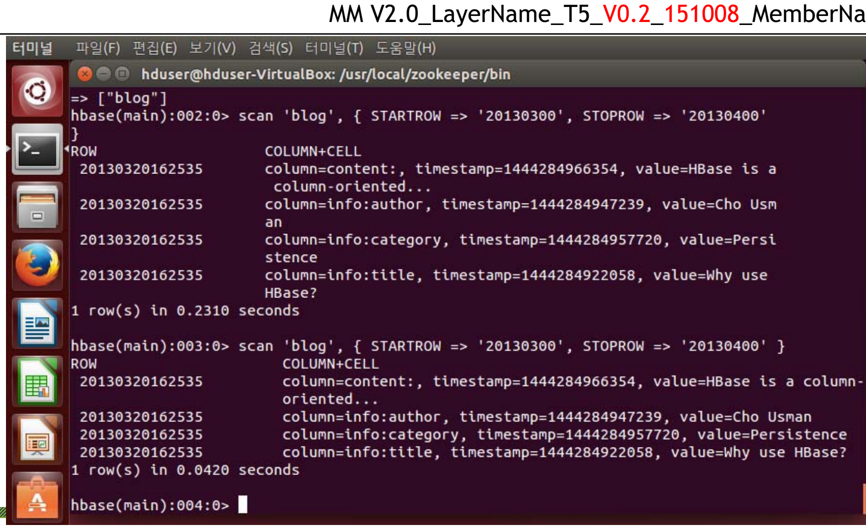


Fig. showing how to create a table in hbase shell

Week 3: Progress

MM V2.0_LayerName_T5_V0.2_151008_MemberName



- Clients interact with HBase via one of several available APIs,
- Native Java API as well as a REST-based interface
- and several RPC interfaces (Apache Thrift, Apache Avro).

Week 3: Future Plan

MM V2.0_LayerName_T5_V0.2_151008_MemberName

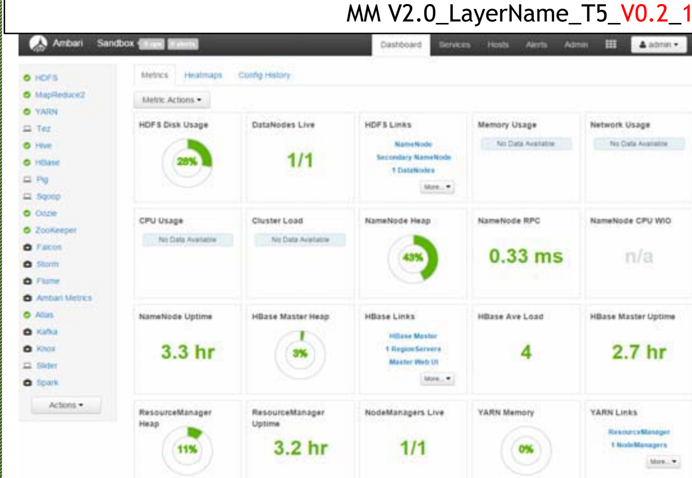
Cluster Management and Provisioning

- Candidate solution: Apache Ambari

Apache Ambari

Week 4: Progress

MM V2.0_LayerName_T5_V0.2_151008_MemberName



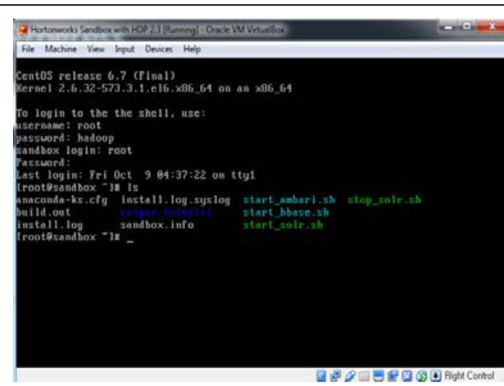


Fig. 2. VM Cluster runs on Centos

Fig. 1. Ambari Hadoop monitoring

Week 4: Future Task

MM V2.0_DCL_T5_V0.4_20151003_Usman Akhtar

- Enable the request to read and write operations on HDFS and HBASE
- Implementing client to send request.

- **Candidate Solution:** Hadoop Eclipse Plug-in

Week 4: Progress

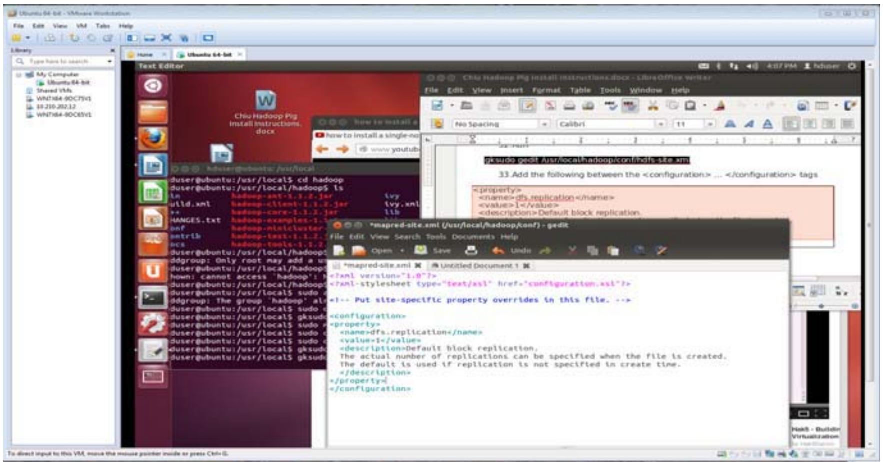
MM V2.0_DCL_T5_V0.4_20151010_Usman Akhtar

- **Integrate the eclipse plugin**
  - Launching Map-Reduces programs on a Hadoop cluster.

- **Following operation:**
  - Browsing/inspecting HDFS nodes
  - Browsing/inspecting Zookeeper nodes

Week 4: Results

MM V2.0_DCL_T5_V0.4_20151010_Usman Akhtar

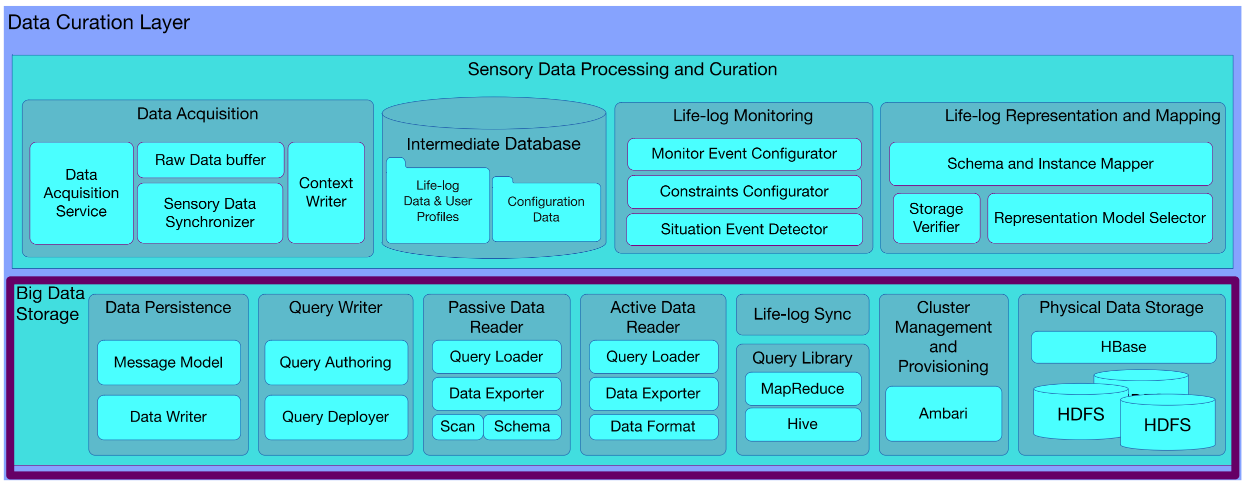


Week 5: Future Plan

MM V2.0_DCL_T5_V0.5_20151017_Usman Akhtar

- With the development of the previous week progress we are interested in building the new Architectural view of Big Data Storage.
- How different layers will interact with big data storage with no scalability issues.

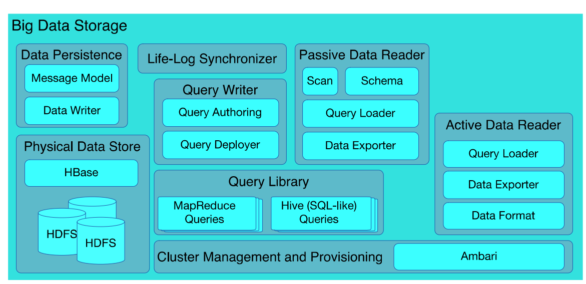
Week 6: DCL V2.0 Architecture



경희대학교

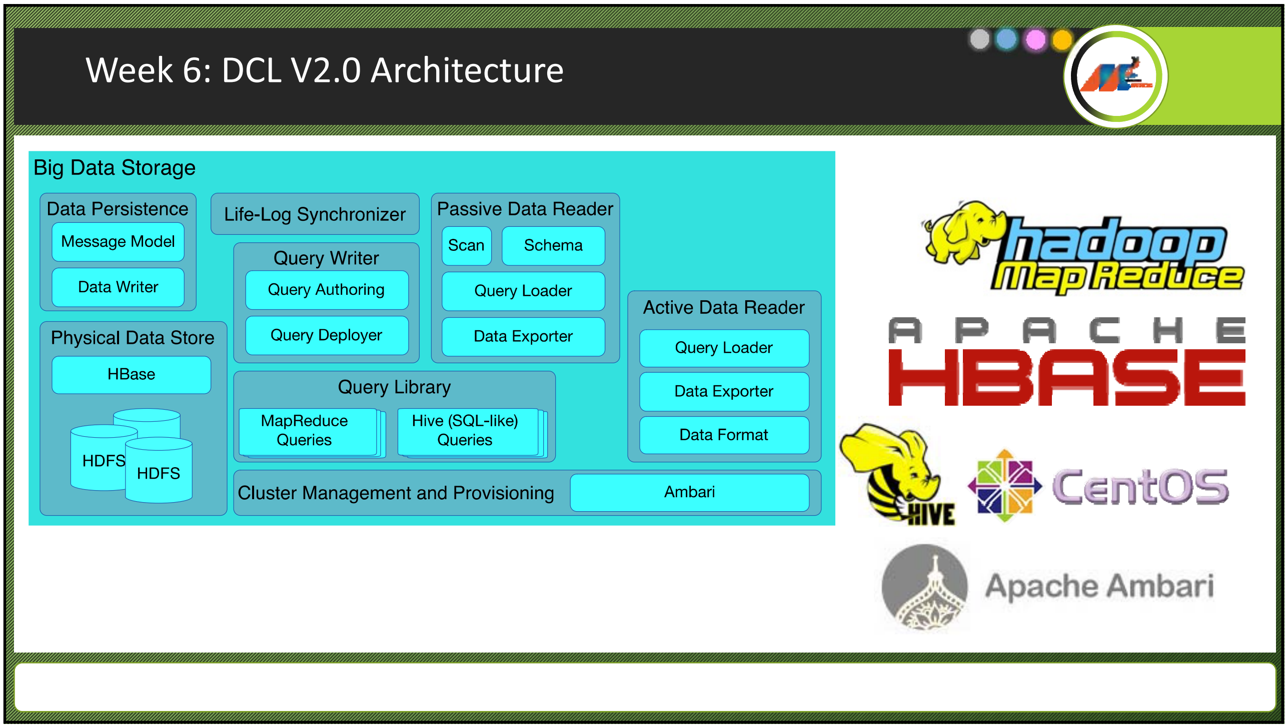
Week 6: Role of Big Data Storage in Mining Mind

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

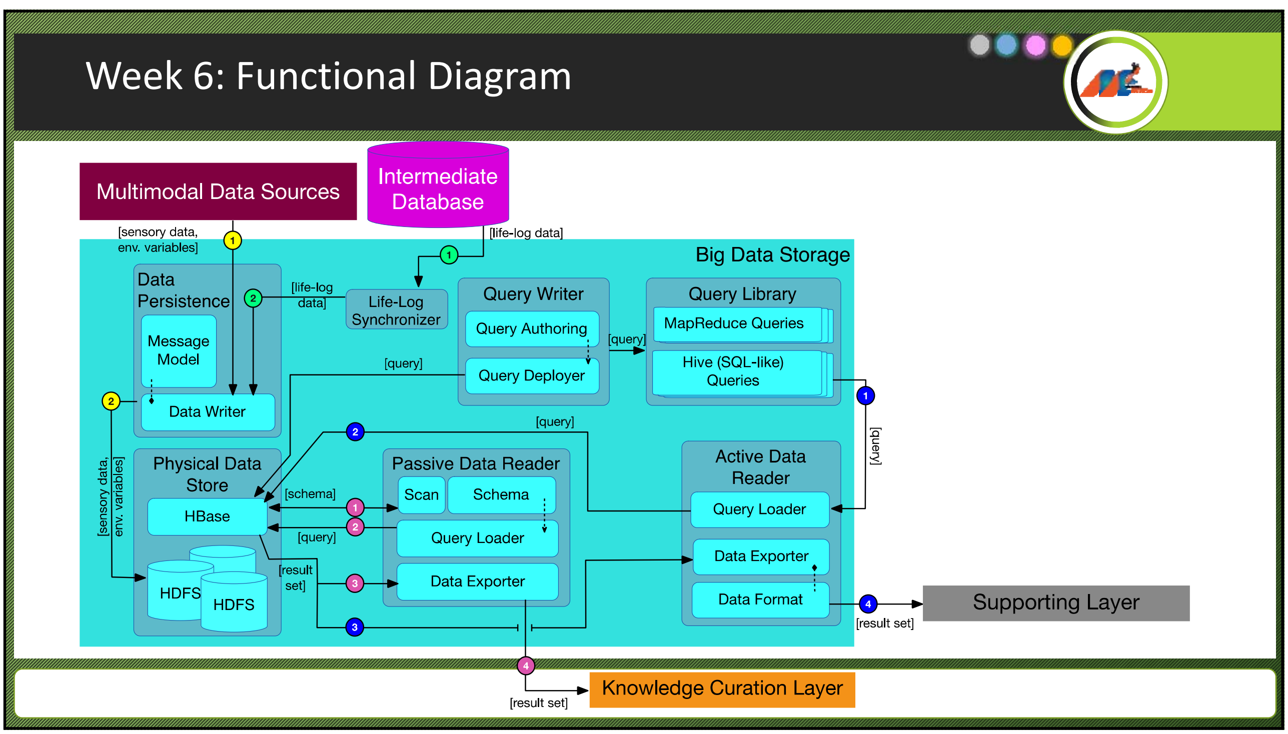


- How is it different from Ver. 1.5 ?
  - Data Raw sensory and environmental data was only written in Big Data Storage
  - Read access for data was not available

경희대학교

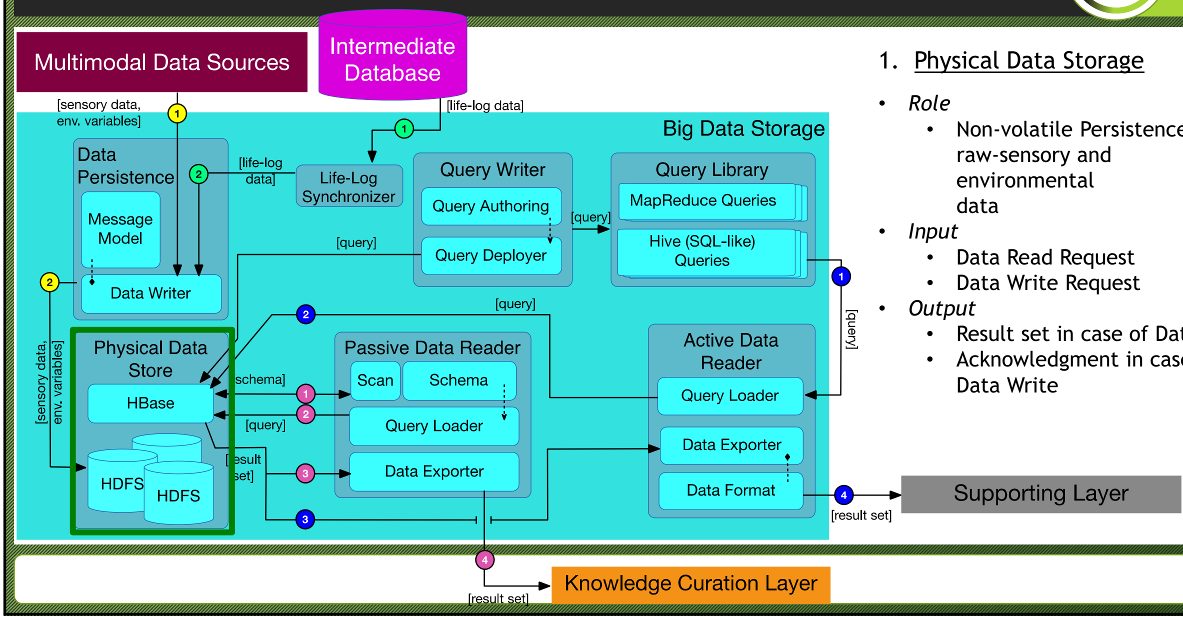


Week 6: DCL V2.0 Architecture
Big Data Storage
Data Persistence
Message Model
Data Writer
Physical Data Store
HBase
HDFS
HDFS
Life-Log Synchronizer
Query Writer
Query Authoring
Query Deployer
Query Library
MapReduce Queries
Hive (SQL-like) Queries
Passive Data Reader
Scan
Schema
Query Loader
Data Exporter
Active Data Reader
Query Loader
Data Exporter
Data Format
Cluster Management and Provisioning
Ambari
hadoop MapReduce
APACHE HBASE
HIVE
CentOS
Apache Ambari



Week 6: Functional Diagram
Multimodal Data Sources
Intermediate Database
[sensory data, env. variables]
[life-log data]
Big Data Storage
Data Persistence
Message Model
Data Writer
[life-log data]
Life-Log Synchronizer
Query Writer
Query Authoring
Query Deployer
[query]
Query Library
MapReduce Queries
Hive (SQL-like) Queries
[query]
[query]
[query]
[sensory data, env. variables]
Physical Data Store
HBase
HDFS
HDFS
[schema]
[query]
[result set]
Passive Data Reader
Scan
Schema
Query Loader
Data Exporter
Active Data Reader
Query Loader
Data Exporter
Data Format
[result set]
Supporting Layer
[result set]
Knowledge Curation Layer

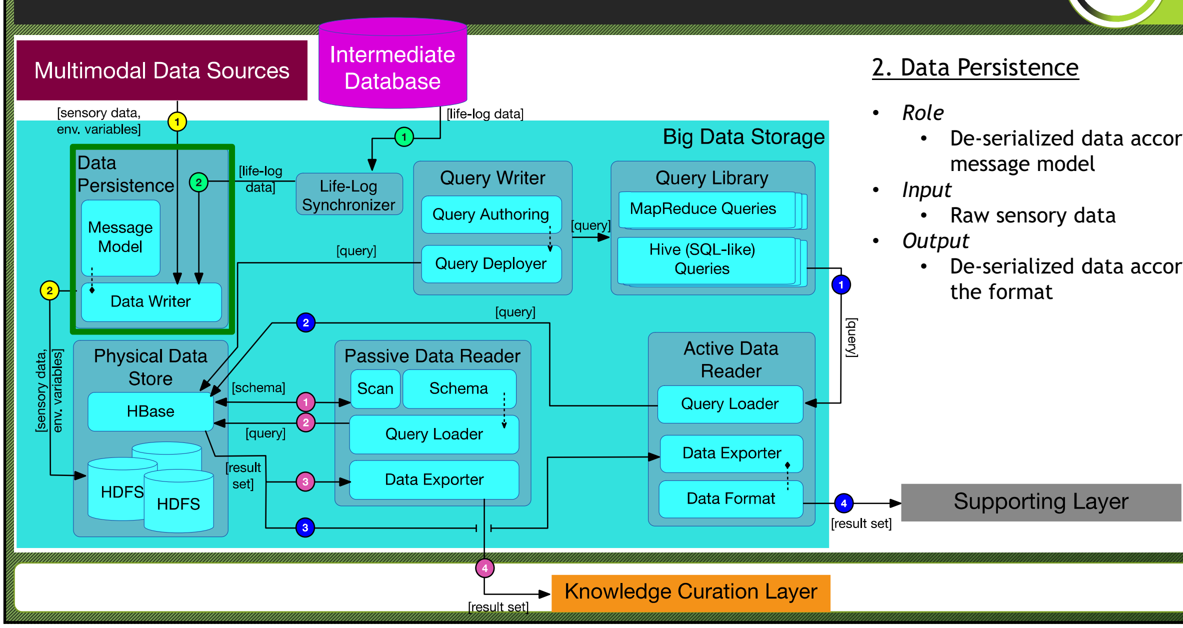
Week 6: Big Data Storage Components



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

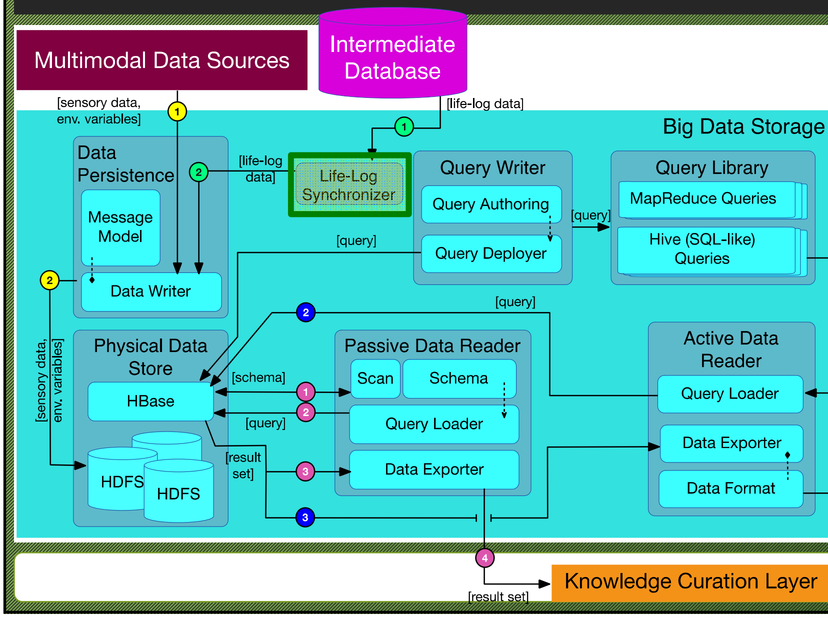
Week 6: Big Data Storage Components



2. Data Persistence

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

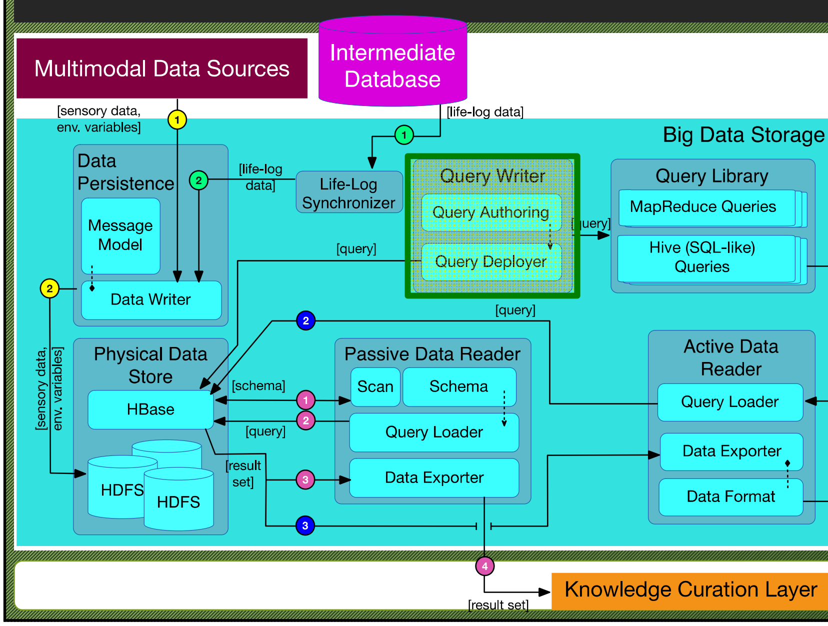
Week 6: Big Data Storage Components



3. <u>Life-log Synchronizer</u>

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

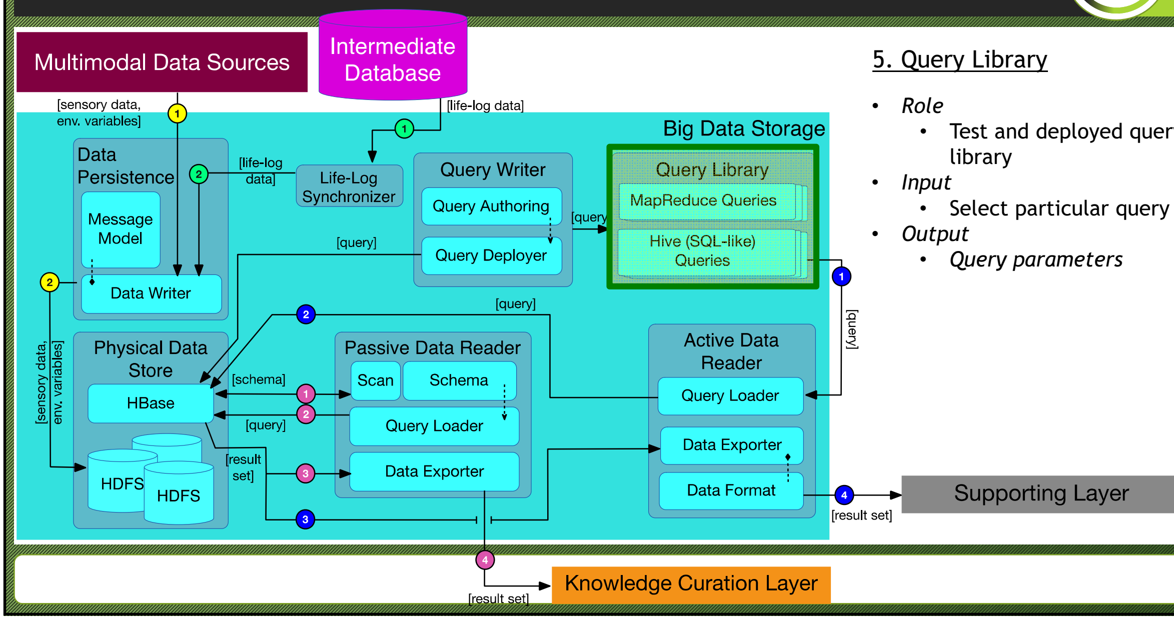
Week 6: Big Data Storage Components



4. <u>Query Writer</u>

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

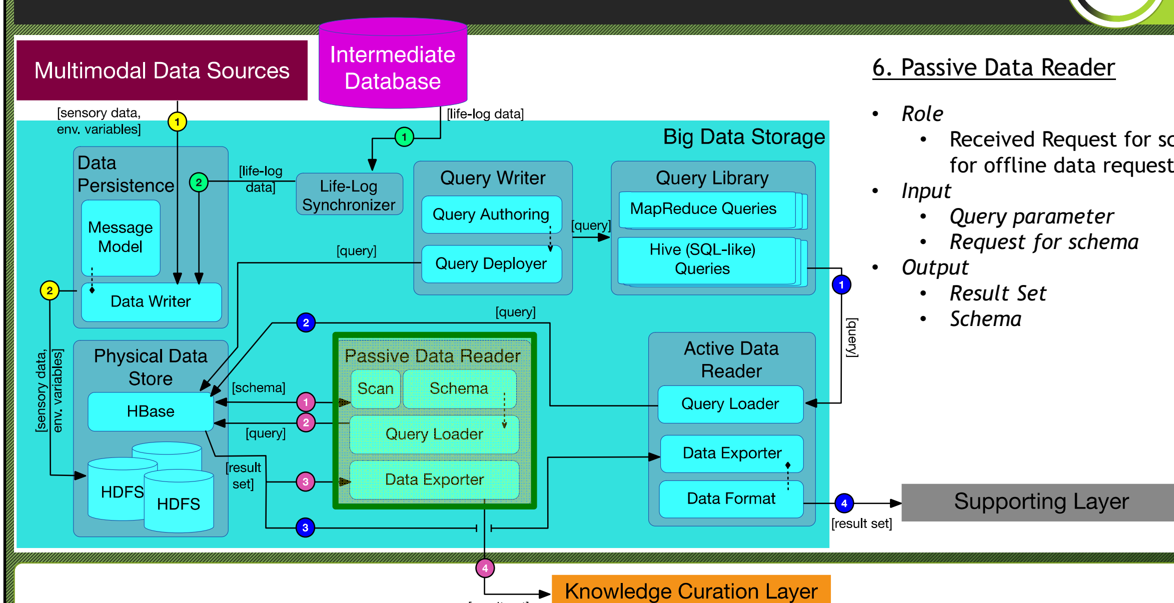
Week 6: Big Data Storage Components



5. Query Library

- *Role*
  - Test and deployed query library
- *Input*
  - Select particular query
- *Output*
  - *Query parameters*

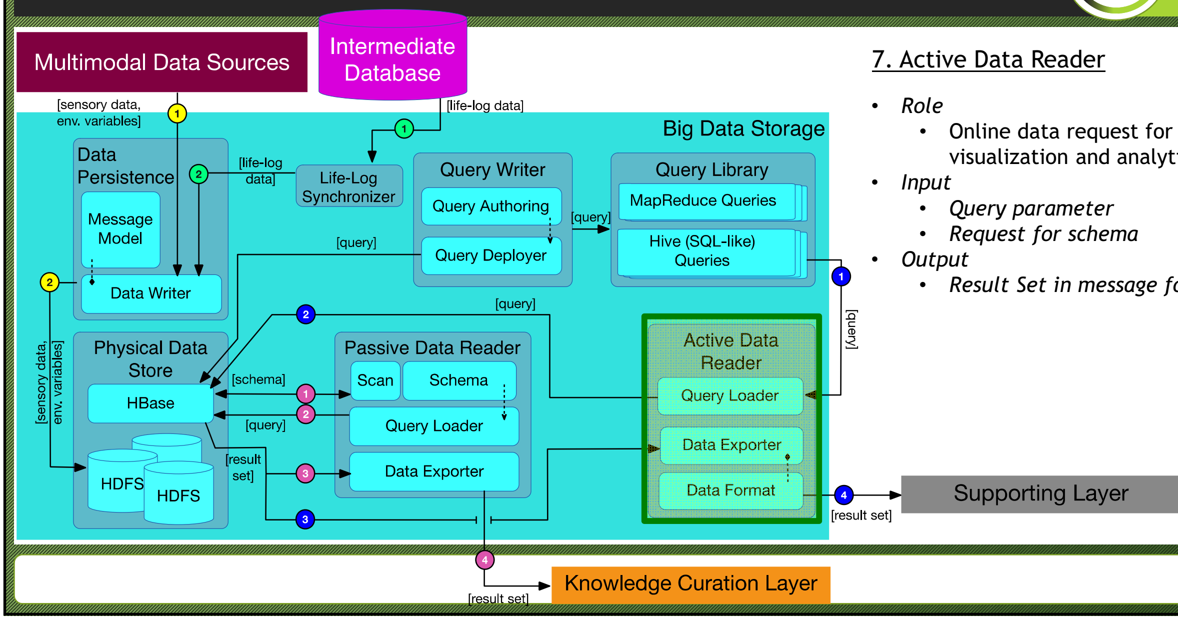
Big Data Storage Components



6. Passive Data Reader

- *Role*
  - Received Request for schema for offline data request
- *Input*
  - *Query parameter*
  - *Request for schema*
- *Output*
  - *Result Set*
  - *Schema*

Big Data Storage Components



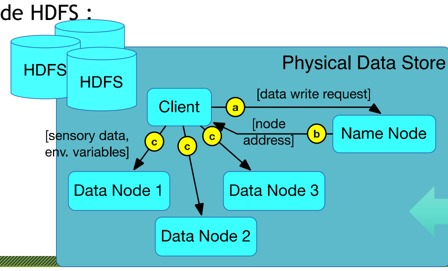
7. Active Data Reader

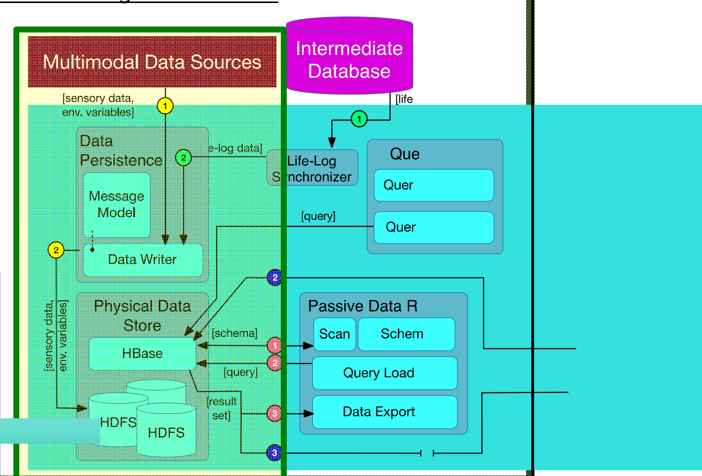
- *Role*
  - Online data request for data visualization and analytics
- *Input*
  - *Query parameter*
  - *Request for schema*
- *Output*
  - *Result Set in message format*

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 :

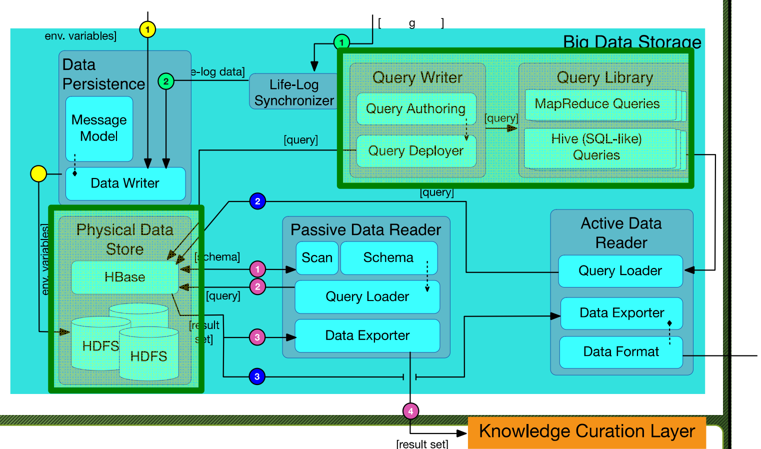




Execution Scenario #2

Offline Query Authoring

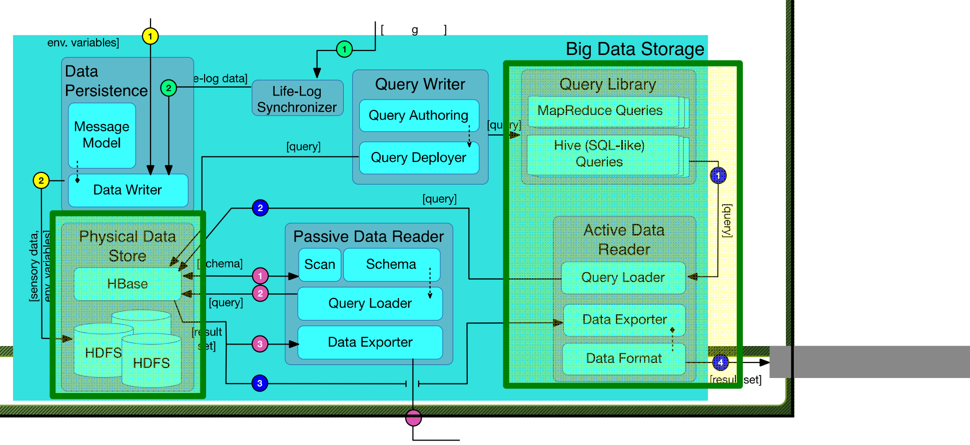
1. Big Data Storage Requirements are converted into MapReduce or Hive Queries
2. Queries are written using Eclipse as an IDE
3. Query is tested and deployed as part of Query Library in Big Data Storage



Execution Scenario #3

Online Data Request for Data Visualization and Analytics

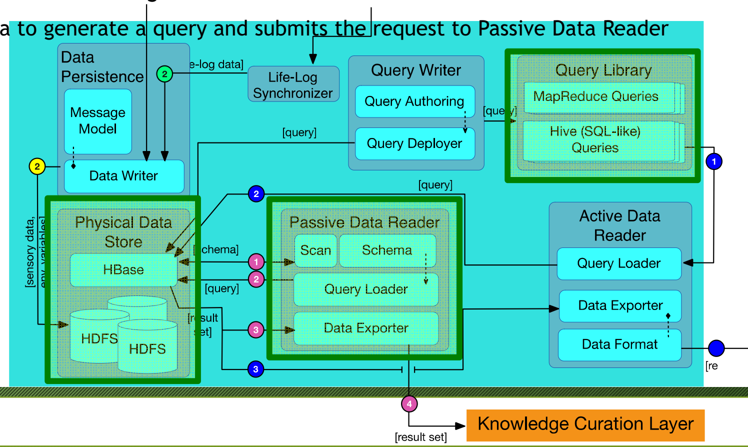
1. Data read request is generated by Visualization and Analytics Components of SL
   I. Query parameters are sent for query selection and execution
2. *Active data reader* selects the particular query depending upon the query parameters
3. Selected Query is sent to Hbase for execution
4. Required data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



Execution Scenario #4

<u>Offline Data Request by KCL for Model Training and Rules Generation</u>

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



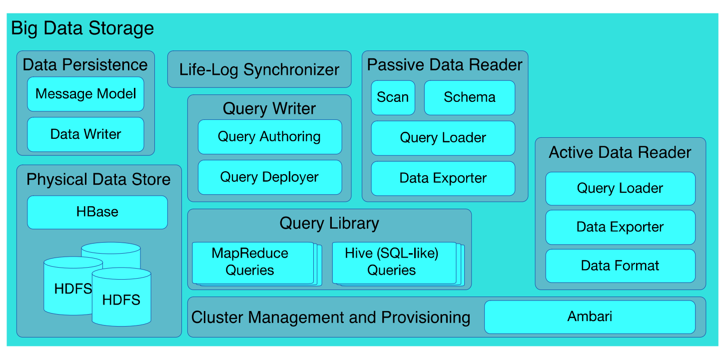
경희대학교

Week 7: Plan to Upgrade Private Cloud

- We want to shift the new architecture implementation on the private cloud.

- We will incorporate the Following major Components:

1. HBase
2. Hive
3. Apachi Ambari
4. Eclipse Integration



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.
- Overall Big Data *Cluster Monitoring*

Information Curation Layer (ICL 2.0)

Principal Investigator:

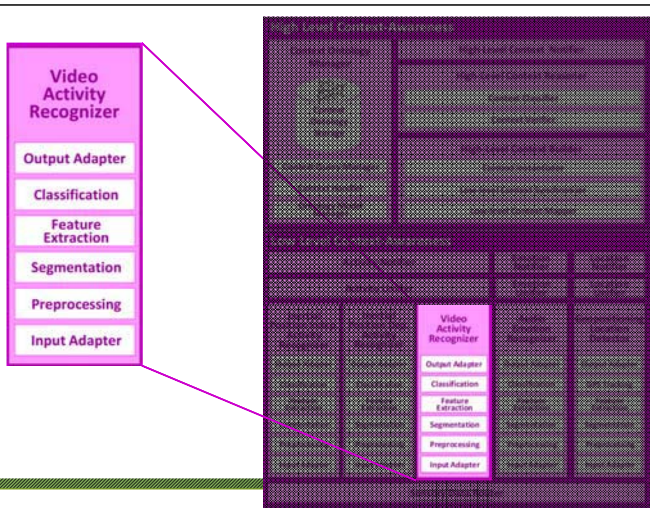
Professor Sungyoung Lee

Video-based Activity Recognition (VAR)

Huynh The Thien
DongUk Kang

Video-based Activity Recognition

MM V2.0_ICL_LLC_VAR_T5_V0.1_150926_THIEN



Compc	Task	Responsible
Video-based AR	Finalize research prototypes	Thien
	Create preprocessing algorithms	Thien, Dong Uk
	Create segmentation algorithms	Thien, Dong Uk
	Create features computation algorithms	Thien, Dong Uk
	Create classification algorithms	Thien, Dong Uk
	Testing	Thien

VAR – Week 1 Progress

MM V2.0_ICL_LLC_VAR_T5_V0.1_150926_THIEN

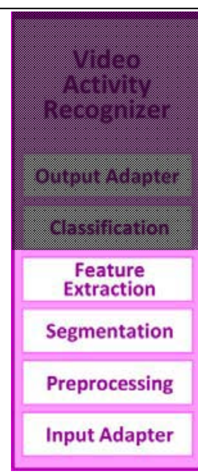
- **Prototype for VAR algorithm**
  - Evaluate the algorithm for VAR component on Matlab with several parameter configuration
  - Finalize the algorithm with the good tradeoff of computational cost and accuracy and guarantee the performance in data transmission
    - Only collect the body frame containing skeleton data
    - Reduce the sampling rate from 30 fps to 5 fps
    - Use only the distance feature
    - Reduce the number of feature from 300 to 30

```
Correctly Classified Instances       47721               99.2162 %
Incorrectly Classified Instances       377                0.7838 %
Kappa statistic                          0.9906
Mean absolute error                      0.0025
Root mean squared error                  0.0464
Relative absolute error                  1.0465 %
Root relative squared error             13.4598 %
Total Number of Instances            48098

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.977     0.002      0.975     0.977     0.976      0.99     stretching
                 0.999     0          1         0.999     0.999      1        lying
                 0.982     0.002      0.988     0.982     0.985      0.993    sweeping
                 0.996     0.001      0.996     0.996     0.996      0.998    sittingcall
                 0.995     0.001      0.993     0.995     0.994      0.998    settingread
                 0.996     0.001      0.993     0.996     0.994      0.998    sittingeat
                 0.987     0          0.992     0.987     0.989      0.996    standingwatch
Weighted Avg.    0.992     0.001      0.992     0.992     0.992      0.997

=== Confusion Matrix ===

     a     b     c     d     e     f     g   <-- classified as
  4062     3    69     4     4    13     1 |     a = stretching
     0  7333     0     5     1     0     0 |     b = lying
    84     0  7161     5     0    33     7 |     c = sweeping
     0     0     1 11966    49     0     3 |     d = sittingcall
     2     0     0    30  7502     0     2 |     e = settingread
    14     0    14     0     0  7582     5 |     f = sittingeat
     5     0     6     7     0    10  2115 |     g = standingwatch
```

VAR– Week 2 Plan

MM V2.0_ICL_LLC_VAR_T5_V0.1_150926_THIEN

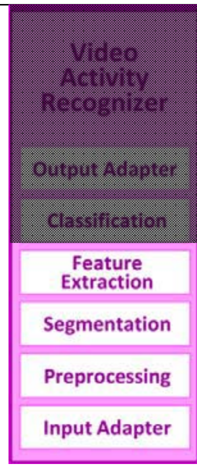
- **Implementation stage**
  - Collaborate with DongUk to finalize the implementation for VAR component based on the prototype in Matlab
  - Preprocessing algorithm for data acquisition from Kinect to DCL and from DCL to ICL
  - Segmentation process
  - Feature extraction



VAR – Week 2 Progress

MM V2.0_ICL_LLC_VAR_T5_V0.2_151003_THIEN_DONGUK

- **Complete the implementation**
  - Preprocessing: collect the skeleton from Kinect by a computer
  - Segmentation process
  - Feature extraction
    - Feature type: joint distance
    - No. feature: 30

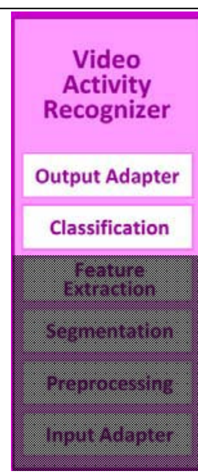



```
private double angle3d(int i,int j){
    double deltax=x[j]-x[i];
    double deltay=y[j]-y[i];
    double a = Math.atan(deltay/deltax);
    if (deltay == 0)
        a=0;
    else if (deltay > 0 && deltax > 0)
        a=a;
    else if (deltay > 0 && deltax < 0)
        a=a+Math.PI;
    else if (deltay <0 && deltax < 0)
        a=a+Math.PI;
    else if (deltay <0 && deltax > 0)
        a=a+2*Math.PI;
    return a;
}
 public double[] extractDistanceFeature(int[] si, int[] sj){
     int numFeatures = si.length;
     double[] features = new double[numFeatures];
     for (int i = 0; i < numFeatures; i++){
         features[i]=distance3d(si[i],sj[i]);
     }
     return features;
 }
 public double[] extractAngleFeature(int[] si, int[] sj){
     int numFeatures = si.length;
     double[] features = new double[numFeatures];
     for (int i = 0; i < numFeatures; i++){
         features[i]=angle3d(si[i],sj[i]);
     }
     return features;
```

VAR– Week 3 Plan

MM V2.0_ICL_LLC_VAR_T5_V0.2_151003_THIEN_DONGUK

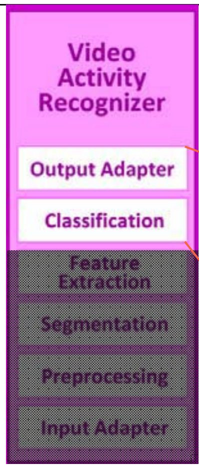
- **VAR implementation (cont)**
  - Feature selection using the feature raking algorithm
  - Classification using the decision tree algorithm with the model extracted from Weka



Task	Responsibl	19-25Sep	26Sep-2Oct	3-9Oct
Finalize research prototypes	Thien			
Create preprocessing algorithms	Thien, Dong Uk			
Create segmentation algorithms	Thien, Dong Uk			
Create features computation algorithms	Thien, Dong Uk			
Create classification algorithms	Thien, Dong Uk			
Testing	Thien			

VAR – Week 3 Progress

MM V2.0_ICL_LLC_VAR_T5_V0.3_151010_THIEN_DONGUK



- Implemented some VAR component
  - Feature selection
  - Classification

```
System.out.println("Distance Feature: ");
for (int i=0;i<distanceFeatures.length; i++)
    System.out.println(distanceFeatures[i]);
System.out.println("Angle Feature: ");
for (int i=0;i<angleFeatures.length; i++)
    System.out.println(angleFeatures[i]);

double[] features = new double[distanceFeatures.length + angleFeatures.length];
System.arraycopy(distanceFeatures, 0, features, 0, distanceFeatures.length);
System.arraycopy(angleFeatures, 0, features, distanceFeatures.length, angleFeatures.length);

Classification j48 = new Classification();
j48.loadModel("D:\\J48VAR.model");
double labelIndex = j48.classify(features);
```

VAR– Week 4 Plan

MM V2.0_ICL_LLC_VAR_T5_V0.3_151010_THIEN_DONGUK

- **Testing stage**
  - Evaluate the algorithm with collected data in Java
  - Verify and refer the output at each step in Java and Matlab
    - Feature calculation
    - Feature extraction
    - Classification

Task	*Responsibl*	19-25Sep	26Sep-2Oct	3-9Oct	10-16Oct
Finalize research prototypes	Thien				
Create preprocessing algorithms	Thien, Dong Uk				
Create segmentation algorithms	Thien, Dong Uk				
Create features computation algorithms	Thien, Dong Uk				
Create classification algorithms	Thien, Dong Uk				
Testing	Thien				

VAR – Week 4 Progress - Testing

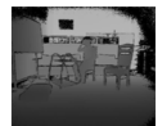
MM V2.0_ICL_LLC_VAR_T5_V0.4_151017_THIEN_DONGUK

- Low accuracy with the different activities having a same posture:
  - Sitting: reading/calling vs eating
  - Standing: watching TV vs calling

```
double [] inputData;
double [] x,y,z;
int[] siD = {2, 3, 3, 3, 3, 3, 8, 17, 2, 2, 2, 2, 2, 2, 3, 3, 4, 8, 8, 8, 9, 10, 13, 14, 18, 20, 1, 2, 2, 2};
int[] sjD = {17, 10, 11, 17, 23, 24, 17, 20, 10, 11, 13, 14, 18, 23, 13, 18, 13, 14, 18, 23, 18, 20, 20, 20, 20, 23, 18, 6, 21,

public void setInputData(double[] inputData){
    this.inputData = inputData;
    separateData();

}
private void separateData(){
    x = new double[25];
    y = new double[25];
    z = new double[25];
    for (int i = 0;  i < 25; i++){
        x[i]=inputData[i*3];
        y[i]=inputData[i*3+1];
        z[i]=inputData[i*3+2];
    }
```


Calling

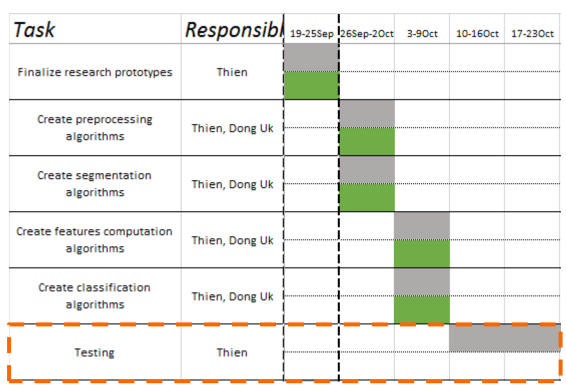

Reading book


Eating

VAR – Week 5 Plan

MM V2.0_ICL_LLC_VAR_T5_V0.4_151017_THIEN_DONGUK

- **Testing stage (cont)**
  - Verify the Java code of VAR model
  - Update the recognition algorithm in Matlab and Java.
    - Extend the feature types: utilize the joint angle feature besides joint distance.
  - Evaluate again with the training data.



VAR – Week 5 Progress

MM V2.0_ICL_LLC_VAR_T5_V0.5_151024_THIEN_DONGUK

- Improve the algorithm by extend the feature type
- Utilize both the joint distance and joint angle
- Update the feature selection algorithm

Joint distance feature index

```
double [] inputData;
double [] x,y,z;
int[] siD = {2, 3, 3, 3, 3, 3, 8, 17, 2, 2, 2, 2, 2, 2, 3, 3, 4, 8, 8, 8, 9, 10, 13, 14, 18, 20, 1, 2, 2, 2};
int[] sjD = {17, 10, 11, 17, 23, 24, 17, 20, 10, 11, 13, 14, 18, 23, 13, 18, 13, 14, 18, 23, 18, 20, 20, 20, 20, 23, 18, 6, 21,
int[] siA = {3, 0, 0, 0, 2, 5, 12, 16, 0, 0, 0, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 5, 8, 9};
int[] sjA = {12, 1, 13, 20, 12, 17, 20, 23, 2, 3, 23, 2, 3, 10, 11, 20, 24, 4, 14, 15, 18, 20, 5, 15, 16, 18, 19, 12, 18, 10};

public void setInputData(double[] inputData){
    this.inputData = inputData;
    separateData();

}
private void separateData(){
    x = new double[25];
    y = new double[25];
    z = new double[25];
    for (int i = 0;  i < 25; i++){
        x[i]=inputData[i*3];
        y[i]=inputData[i*3+1];
```

Joint angle feature index

VAR – Week 6 Plan

MM V2.0_ICL_LLC_VAR_T5_V0.5_151024_THIEN_DONGUK

- **Testing stage (cont)**
  - Do the real-time test for VAR component using the Kinect device
  - Integrate the VAR component to ICL

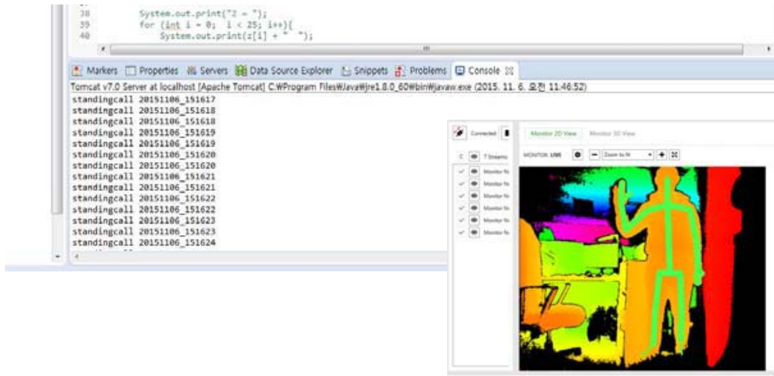
Task	Responsibl	19-25Sep	26Sep-2Oct	3-9Oct	10-16Oct	17-23Oct	24-30Oct
Finalize research prototypes	Thien						
Create preprocessing algorithms	Thien, Dong Uk						
Create segmentation algorithms	Thien, Dong Uk						
Create features computation algorithms	Thien, Dong Uk						
Create classification algorithms	Thien, Dong Uk						
Testing	Thien						

VAR – Week 6 Progress

MM V2.0_ICL_LLC_VAR_T5_V0.6_151031_THIEN_DONGUK

- **Testing stage (cont)**
  - The data from Kinect is collected by a computer and transmitted to ICL
  - The VAR component detects and recognizes the activity in the real-time
  - Do some preliminary tests with Kinect on desk

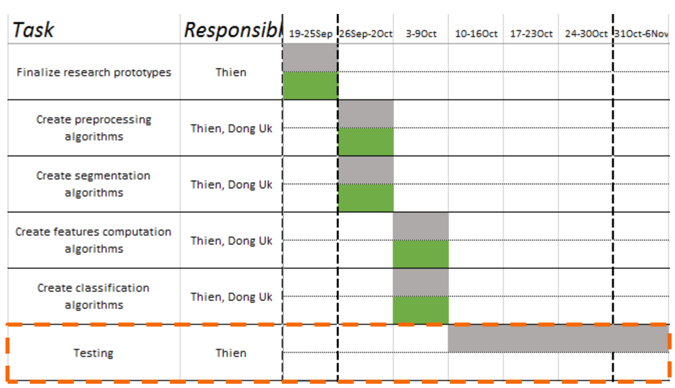
The recognition results by VAR component in ICL



VAR – Week 7 Plan

MM V2.0_ICL_LLC_VAR_T5_V0.6_151031_THIEN_DONGUK

- **Testing stage (cont)**
  - Do the online test with Kinect device in the set up scenario (ITRC room)
  - The test is implemented with new volunteers
  - Evaluate the accuracy in the real-time and refer the detected labels with ground truth



VAR – Week 7 Progress

MM V2.0_ICL_LLC_VAR_T5_V0.7_151107_THIEN_DONGUK

- **Testing stage (cont)**
  - The online testing is implemented with some new volunteers
  - The classification accuracy is reported by the confusion matrix

	Stretching	Lying	Sweeping	Sitting	Eating	Standing
Stretching	**95.5**	0.0	2.0	0.5	0.5	1.5
Lying	0.2	**96.8**	0.0	2.9	0.0	0.1
Sweeping	3.5	0	**95.5**	0.2	0.0	0.8
Sitting	5.3	0.0	0.0	**92.1**	2.6	0.0
Eating	0.0	0.0	0.2	13.7	**86.1**	0.0
Standing	5.5	0.0	5.5	0.0	2.7	**86.3**

Subject	Age	Height (cm)	Weight (kg)
S01	24	165	63
S02	25	180	85
S03	28	173	68
S04	22	173	90
S05	23	170	64

- **Testing scenario**
  - Indoor (ITRC room)
  - Number of subject: 5, number of activity: 6
  - Totally of sample: over 74 000
  - Each activity is presented with 2 minutes of duration

VAR – Week 8 Progress

To evaluate other alternate strategies to improve the performance of the recognizer.

Appendix

Source code – VAR Feature Extraction

```
package com.uclab.icl.var;

public class FeatureExtraction {
double [] inputData;
double [] x,y,z;

public void setInputData(double[] inputData){
this.inputData = inputData;
separateData();
}
private void separateData(){
x = new double[25];
y = new double[25];
z = new double[25];
for (int i = 0;  i < 25; i++){
x[i]=inputData[i*3+1];
y[i]=inputData[i*3+2];
z[i]=inputData[i*3+3];
}
}
private double distance3d(int i, int j){
double d=Math.sqrt(Math.pow(x[i]-x[j],2)+Math.pow(y[i]-
y[j],2)+Math.pow(z[i]-z[j],2));
return d;
}
```

```
private double angle3d(int i,int j){
double deltax=x[j]-x[i];
double deltay=y[j]-y[i];
double a = Math.atan(deltay/deltax);
if (deltay == 0)
a=0;
else if (deltay > 0 && deltax > 0)
a=a;
else if (deltay > 0 && deltax < 0)
a=a+Math.PI;
else if (deltay <0 && deltax < 0)
a=a+Math.PI;
else if (deltay <0 && deltax > 0)
a=a+2*Math.PI;
return a;
}
 public double[] extractDistanceFeature(int[] si, int[] sj){
 int numFeatures = si.length;
 double[] features = new double[numFeatures];
 for (int i = 0; i < numFeatures; i++){
 features[i]=distance3d(si[i],sj[i]);
 }
 return features;
 }
 public double[] extractAngleFeature(int[] si, int[] sj){
 int numFeatures = si.length;
 double[] features = new double[numFeatures];
 for (int i = 0; i < numFeatures; i++){
 features[i]=angle3d(si[i],sj[i]);
 }
 return features;
 }
}
```

Information Curation Layer

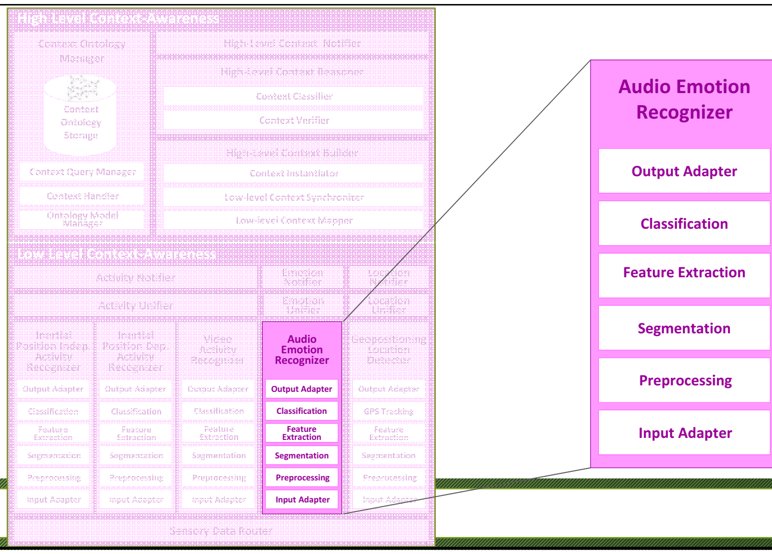
Audio-based Emotion Recognition (AER)

Le Ba Vui

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

Audio-based Emotion Recognition

MM V2.0_ICL_LLC_AER_T5_V0.1_150926_LEBAVUI



AER – Week 1 Progress

MM V2.0_ICL_LLC_AER_T5_V0.1_150926_LEBAVUI

- Evaluation with final dataset of 10 people

	a	b	c	d	e	f	g
a=anger	**53**	6	9	7	4	4	29
b=boredom	0	**50**	17	0	12	35	5
c=fear	13	17	**59**	7	13	4	6
d=happiness	9	5	13	**48**	15	3	9
e=neutral	3	14	12	11	**32**	14	3
f=sadness	1	38	6	2	17	**41**	3
g=surprise	30	6	5	9	8	6	**28**

7 emotions - average accuracy: 42.5%

	a	b	c	d
a=anger	**76**	12	11	13
b=sadness	4	**73**	29	2
c=neutral	5	28	**42**	14
d=happiness	16	6	20	**60**

4 emotions - average accuracy: 61%

- Implementation of the prototype: 80%
  - Preprocessor ✓
  - Feature Extractor ✓
  - Classifier ✓
  - Input, output adapters

AER– Week 2 Plan

MM V2.0_ICL_LLC_AER_T5_V0.1_150926_LEBAVUI

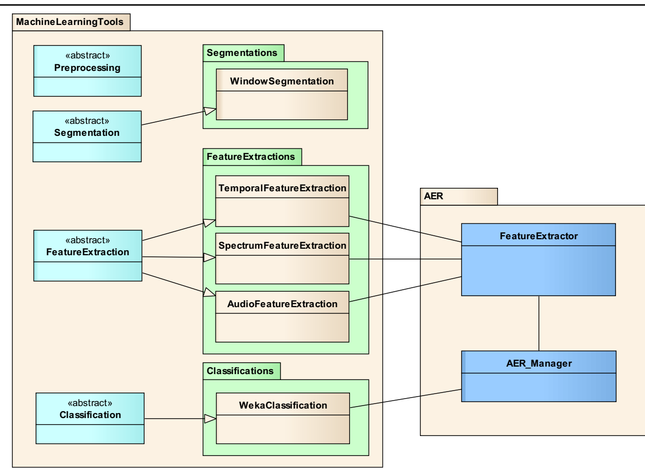
- Migrate ICLv1.5 source code to new package MachineLearningTools
- Implementing (coding and debugging) preprocessing and segmentation algorithms

Component	Task	Responsible	19-25Sep	26Sep-2Oct	3-9Oct
	Migrate MMV1.5 Preprocessing, Segmentation, FeatureExtraction, Classification classes to pkg MachineLearningTools	Vui			
Audio-based ER 1	Finalize research prototypes	Vui			
	Create preprocessing algorithms	Vui			
	Create segmentation algorithms	Vui			
	Create features computation algorithms	Vui			
	Create classification algorithms	Vui			
	Testing	Vui			

AER – Week 2 Progress

MM V2.0_ICL_LLC_AER_T5_V0.2_151003_LEBAVUI

- Implemented initial version of MachineLearningTools package
- Implemented initial version of AER component
  - Segmentation
  - Feature Extraction
  - Classification
- Challenge
  - Make the package more generic for future use



AER – Week 3 Plan

MM V2.0_ICL_LLC_AER_T5_V0.2_151003_LEBAVUI

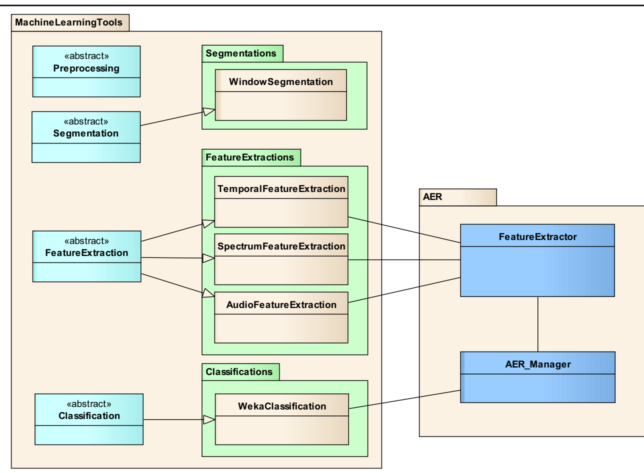
- Debugging and testing AER component with trained model

Component	*Task*	*Responsible*	19-25Sep	26Sep-2Oct	3-9Oct
	Migrate MMV1.5 Preprocessing, Segmentation, FeatureExtraction, Classification classes to pkg MachineLearningTools	Vui			
Audio-based ER 1	Finalize research prototypes	Vui			
	Create preprocessing algorithms	Vui			
	Create segmentation algorithms	Vui			
	Create features computation algorithms	Vui			
	Create classification algorithms	Vui			
	Testing	Vui			

AER – Week 3 Progress

MM V2.0_ICL_LLC_AER_T5_V0.3_151010_LEBAVUI

- Implementing (coding and debugging) all required algorithms for AER component
  - Completed
- Defining configuration file for a particular component (AER)
  - In progress



AER – Week 3 Progress - Configuration File

MM V2.0_ICL_LLC_AER_T5_V0.3_151010_LEBAVUI

An example of configuration file for AER component

```xml
<component name="aer" description="Audio Emotion Recognition">
  <instances>
    <instance name="windowSegmentation" description="Window Segmentation">
      <input>
        audioData
      </input>
      <output>
        audioWindow
      </output>
      <params>
        <param name="windowLength" value="3"/>
      </params>
    </instance>
    <instance name="framingWindow" description="Overlapping Framming Window">
      <input>
        audioWindow
      </input>
      <output>
        frames
      </output>
      <params>
        <param name="frameSize" value="0.1"/>
        <param name="frameStep" value="0.05"/>
      </params>
    </instance>
    <instance name="mfccExtractor" description="MFCC Feature Extraction">
      <input>
        frames
      </input>
      <output>
        mfccFeatures
      </output>
      <params>
        <param name="samplingRate" value="8000"/>
        <param name="extractEnergy" value="true"/>
      </params>
    </instance>
    <instance name="zcrExtractor" description="Zero Crossing Feature Extraction">
      <input>
        frames
      </input>
      <output>
        zcrFeatures
      </output>
      <params>
        <param name="samplingRate" value="8000"/>
      </params>
    </instance>
```

AER – Week 3 Progress - Configuration File

MM V2.0_ICL_LLC_AER_T5_V0.3_151010_LEBAVUI

An example of configuration file for AER component (con't)

```xml
    <instance name="mean2D" description="Compute Mean of Matrix">
      <input>
        mfccFeatures,zcrFeatures
      </input>
      <output>
        meanFeatures
      </output>
      <params>

      </params>
    </instance>
    <instance name="std2D" description="Compute STD of Matrix">
      <input>
        mfccFeatures,zcrFeatures
      </input>
      <output>
        stdFeatures
      </output>
      <params>

      </params>
    </instance>

    <instance name="wekaClassifier" description="WEKA Classification">
      <input>
        meanFeatures,stdFeatures
      </input>
      <output>
        emotionLabel
      </output>
      <params>
        <param name="labelList" value="angry,happiness,sadness,neutral"/>
        <param name="trainedModel" value="models/aer.model"/>
      </params>
    </instance>

  </instances>
</component>
```

AER – Week 4 Plan

MM V2.0_ICL_LLC_AER_T5_V0.3_151010_LEBAVUI

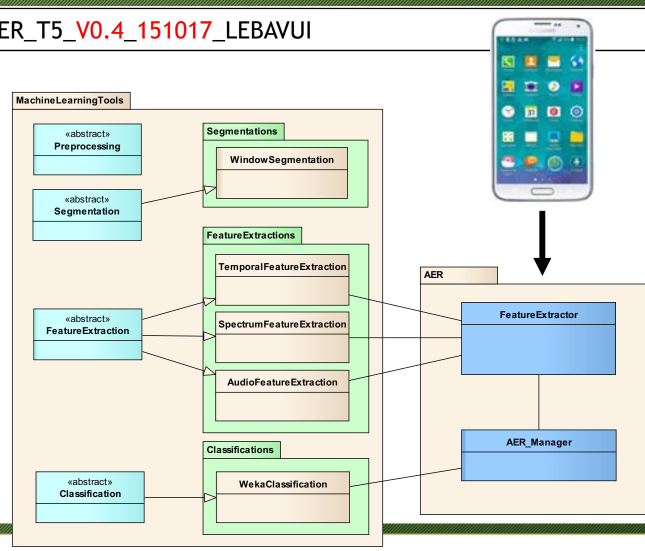
- Testing whole implementation of AER
- Combine all component's machine learning algorithms into MachineLearningTools package
- Modify classes to work with configuration file approach

Component	Task	Responsible	26Sep-2Oct	3-9Oct	10-16Oct
	Migrate MMV1.5 Preprocessing, Segmentation, FeatureExtraction, Classification classes to pkg MachineLearningTools	Vui			
Audio-based ER 1	Finalize research prototypes	Vui			
	Create preprocessing algorithms	Vui			
	Create segmentation algorithms	Vui			
	Create features computation algorithms	Vui			
	Create classification algorithms	Vui			
	Testing	Vui			

AER – Week 4 Progress

MM V2.0_ICL_LLC_AER_T5_V0.4_151017_LEBAVUI

- Testing whole implementation of AER
  - Completed
- Combine all component's machine learning algorithms into MachineLearningTools package
  - In progress



AER – Week 4 Progress - Testing

MM V2.0_ICL_LLC_AER_T5_V0.4_151017_LEBAVUI

```
public static void main(String[] args) throws IOException {
    int n = 0;
    ServerSocket listener = new ServerSocket(9090);
    try {
        while (true) {
            Socket s = listener.accept();
            try {
                DataInputStream in = new DataInputStream(s.getInputStream());
                int len = in.readInt();
                System.out.println("Data length: " + len);
                double[] samples = new double[len];
                for(int i = 0; i < len; i++)
                    samples[i] = in.readDouble();

                AERThread thread = new AERThread("Thread " + ++n, samples);
                thread.start();
            } finally {
                s.close();
            }
        }
    }
    finally {
        listener.close();
    }
}
```

Read data from client via socket

Start new recognition thread

Main testing application

```
public void run() {
    try {
        long startTime = System.nanoTime();

        double rmsEnergy = TemporalFeatureExtraction.computeRMSE(samples);
        if (rmsEnergy < 0.1)
            System.out.println("Silence ...");
        else {
            System.out.println("Extracting features ...");
            AERFeatureExtraction fe = new AERFeatureExtraction();
            double[] features = fe.extractFeature(samples);

            System.out.println("Classifying label ...");
            WekaClassification classifier = new WekaClassification();
            classifier.loadModel("F:\\smo.model");
            int labelIndex = classifier.classify(features);
            System.out.println("Classified label: " + LABELS[labelIndex]);
        }

        long estimatedTime = System.nanoTime() - startTime;
        System.out.println("Elapsed Time - " + estimatedTime / 1000000 + " ms");
    }
    catch (Exception ex) {
        ex.printStackTrace();
    }
}
```

Extract features

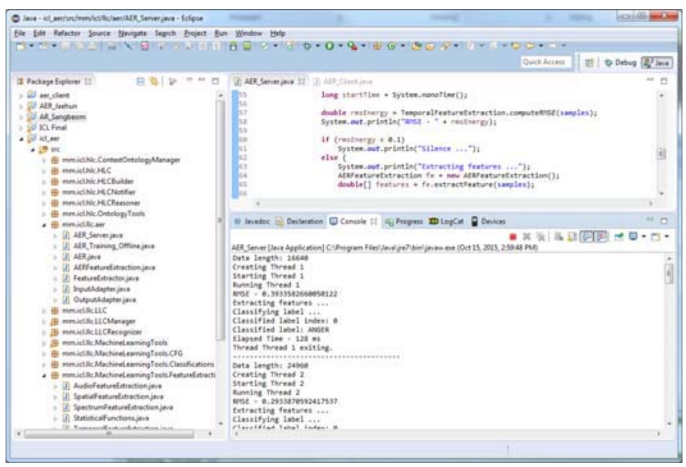
Classify label

AER Thread

AER – Week 4 Progress - Testing

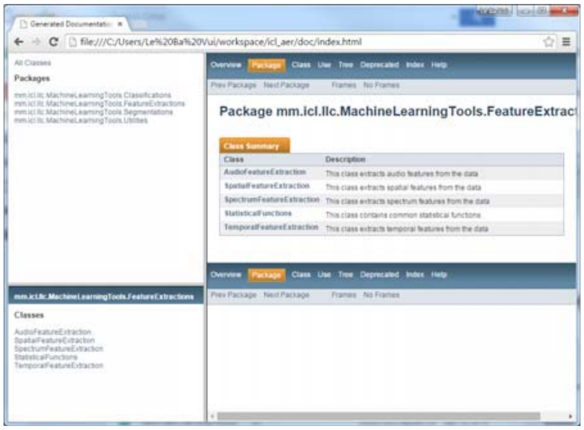
MM V2.0_ICL_LLC_AER_T5_V0.4_151017_LEBAVUI

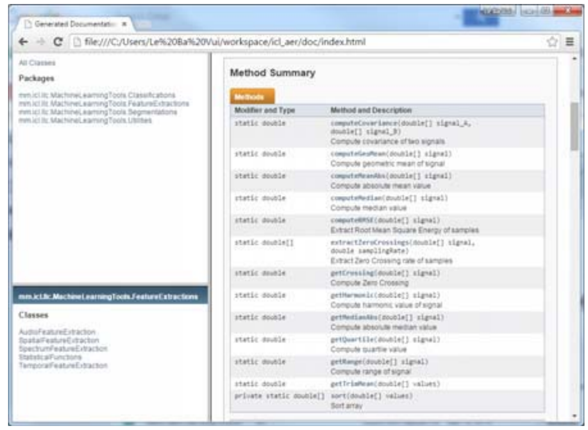
```
Data length: 16640
Creating Thread 1
Starting Thread 1
Running Thread 1
RMSE - 0.3933582660050122
Extracting features ...
Classifying label ...
Classified label index: 0
Classified label: ANGER
Elapsed Time - 128 ms
Thread Thread 1 exiting.
----------------------------------------
Data length: 24960
Creating Thread 2
Starting Thread 2
Running Thread 2
RMSE - 0.2933870592417537
Extracting features ...
Classifying label ...
Classified label index: 0
Classified label: ANGER
Elapsed Time - 39 ms
Thread Thread 2 exiting.
----------------------------------------
```



AER – Week 4 Progress – Machine Learning Tools

MM V2.0_ICL_LLC_AER_T5_V0.4_151017_LEBAVUI





http://lebavui.github.io/mm_icl_llc_mlt/doc/index.html

AER – Week 5 Plan

MM V2.0_ICL_LLC_AER_T5_V0.4_151017_LEBAVUI

- Integrating and testing with other components
  - Implement input and output adapter to interface with notifier and unifier components
  - Integrate and test with other components

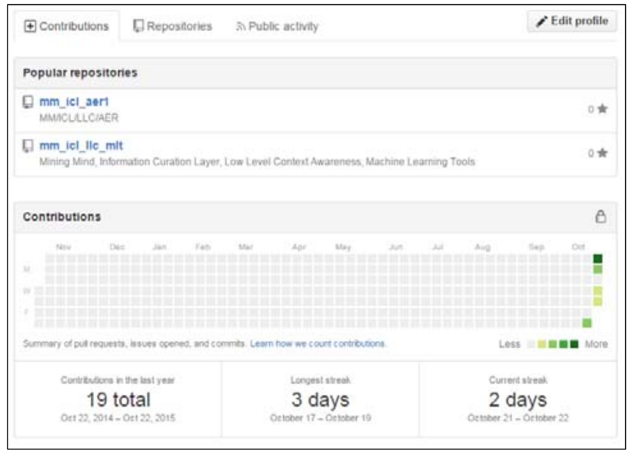
Component	*Task*	*Responsible*	3-9Oct	10-16Oct	17-23Oct
	Migrate MMV1.5 Preprocessing, Segmentation, FeatureExtraction, Classification classes to pkg MachineLearningTools	Vui			
Audio-based ER 1	Finalize research prototypes	Vui			
	Create preprocessing algorithms	Vui			
	Create segmentation algorithms	Vui			
	Create features computation algorithms	Vui			
	Create classification algorithms	Vui			
	Testing	Vui			

AER – Week 5 Progress

MM V2.0_ICL_LLC_AER_T5_V0.5_151024_LEBAVUI

- Finalized Machine Learning Tools package.
  - Shared with all ICL members
- Complete implementation of AER component
  - Shared with Donguk for integration
  - Shared with Jaehun for online evaluation

https://github.com/lebavui/mm_icl_llc_mlt
https://github.com/lebavui/mm_icl_aer1



AER – Week 5 Progress

MM V2.0_ICL_LLC_AER_T5_V0.5_151024_LEBAVUI

- Online evaluation results
  - Overall accuracy: 43.28%

Testing scenario: Recorded phone call is sent directly to ICL server via socket communication for real-time testing.

Recording Application

Socket Comm. →

Audio based Emotion Recognition

ICL Servers

	Age	Gender	Weight (kg)	Height (cm)
User1	23	male	63	165
User2	26	male	85	180
User3	23	male	90	173
User4	23	male	64	170
User5	24	male	105	169

	Happiness	Anger	Sadness	Neutral
Happiness	**40**	7	10	15
Anger	27	**34**	21	17
Sadness	15	4	**32**	2
Neutral	13	7	35	**26**

AER – Week 6 Plan

MM V2.0_ICL_LLC_AER_T5_V0.5_151024_LEBAVUI

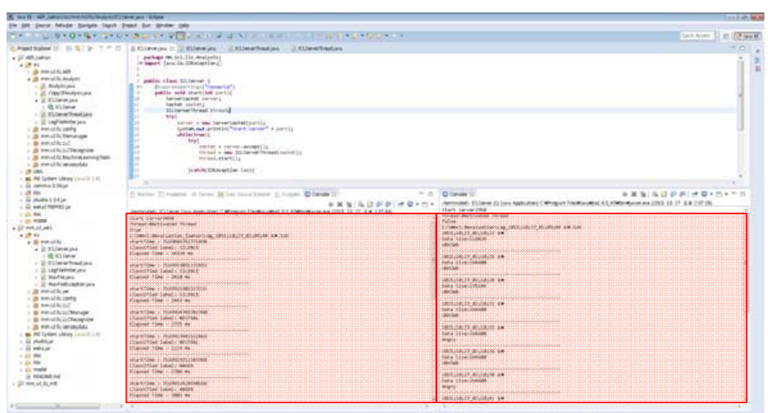
- Implement input and output adapter to interface with notifier and unifier components
- Test and debug if required

Component	*Task*	*Responsible*	10-16Oct	17-23Oct	24-30Oct
	Migrate MMV1.5 Preprocessing, Segmentation, FeatureExtraction, Classification classes to pkg MachineLearningTools	Vui			
Audio-based ER 1	Finalize research prototypes	Vui			
	Create preprocessing algorithms	Vui			
	Create segmentation algorithms	Vui			
	Create features computation algorithms	Vui			
	Create classification algorithms	Vui			
	Testing	Vui			

AER – Week 6 Progress

MM V2.0_ICL_LLC_AER_T5_V0.6_151030_LEBAVUI

- Implement input and output adapter to interface with notifier and unifier components
  - Pending
- Test and debug if required
  - Combined with Jaehun's component to make 2 servers running in the same machine for real-time testing



AER – Week 7 Plan

MM V2.0_ICL_LLC_AER_T5_V0.6_151030_LEBAVUI

- Refinement of the model to improve the performance
- Test and debug if required

Component	*Task*	*Responsible*	17-23Oct	24-30Oct	31Oct-6Nov
	Migrate MMV1.5 Preprocessing, Segmentation, FeatureExtraction, Classification classes to pkg MachineLearningTools	Vui			
Audio-based ER 1	Finalize research prototypes	Vui			
	Create preprocessing algorithms	Vui			
	Create segmentation algorithms	Vui			
	Create features computation algorithms	Vui			
	Create classification algorithms	Vui			
	Testing	Vui			

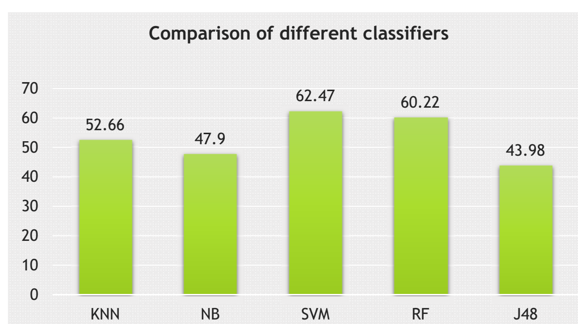
AER – Week 7 Progress

MM V2.0_ICL_LLC_AER_T5_V0.6_151107_LEBAVUI

- Offline re-evaluation
- Feature Extractors
  - Features: MFCC, ZCR, LPC, RMS, Spectral Centroid, Spectral Rolloff Point, Spectral Variability
  - Statistical functions: mean, std
  - Compute delta
  - Total: 84 features
- Classifiers
  - KNN, NB, SVM, RF, J48

	a	b	c	d
a=anger	**65**	15	7	11
b=happiness	18	**57**	5	10
c=sadness	4	2	**61**	16
d=neutral	10	10	26	**40**

SVM - 4 emotions - average accuracy: 62.47%



AER – Week 8 Plan

MM V2.0_ICL_LLC_AER_T5_V0.6_151107_LEBAVUI

- Refinement of the model to improve the performance
- Test and debug if required

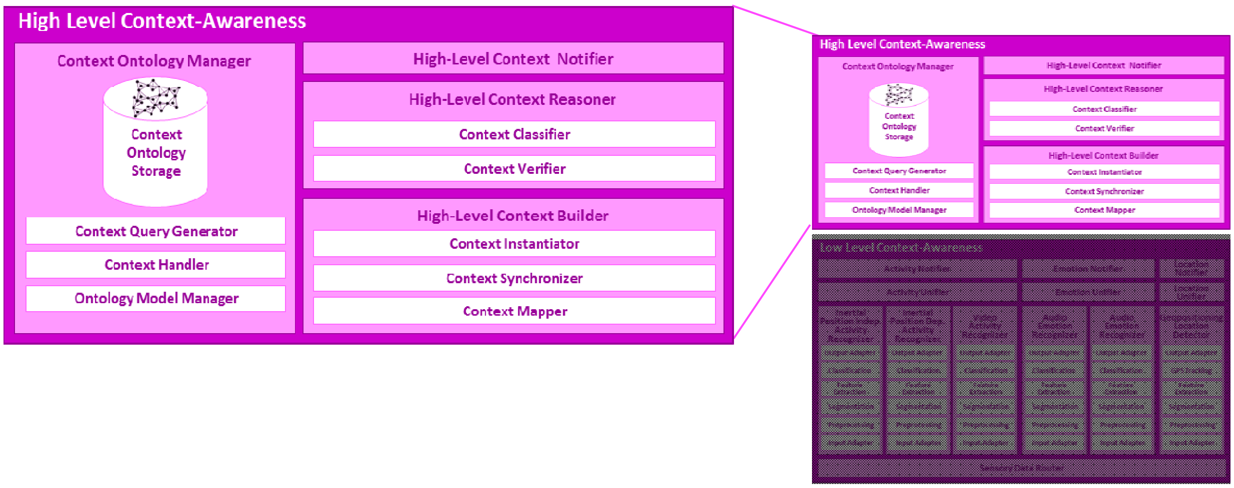
Component	*Task*	*Responsible*	24-30Oct	31Oct-6Nov	7-14Nov
	Migrate MMV1.5 Preprocessing, Segmentation, FeatureExtraction, Classification classes to pkg MachineLearningTools	Vui			
Audio-based ER 1	Finalize research prototypes	Vui			
	Create preprocessing algorithms	Vui			
	Create segmentation algorithms	Vui			
	Create features computation algorithms	Vui			
	Create classification algorithms	Vui			
	Testing	Vui			

High-Level Context Awareness Implementation

Claudia Villalonga

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

HLCA Architecture



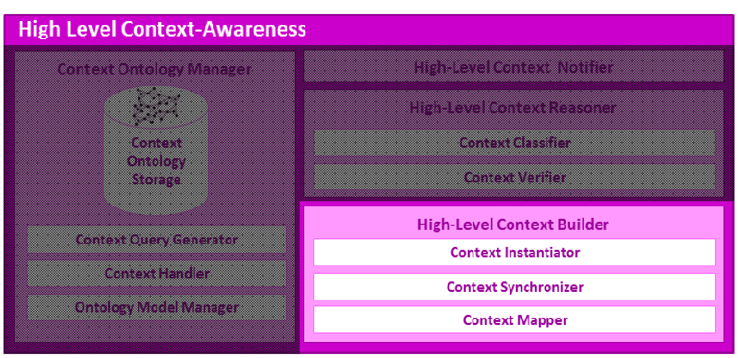
HLCA Tools and Technologies

- Data Representation
  - Ontological representation (OWL2)
- Ontology Query Language
  - SPARQL
- Ontology Engineering
  - Protégé
- Ontological Reasoner
  - Pellet
- Ontological Framework
  - Jena (Semantic Web Framework)
- Ontology Storage (Triplestore)
  - Jena TDB
- Programming Language
  - Java

HLC Builder

Claudia Villalonga

High-Level Context Builder

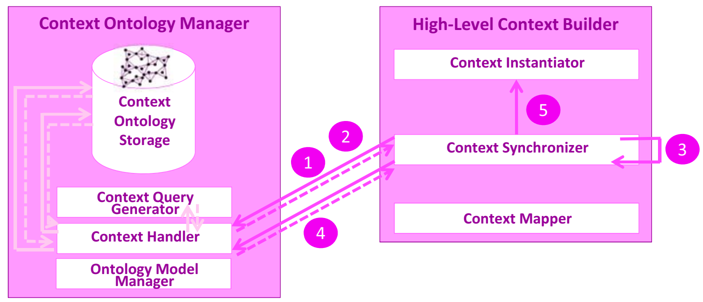


- Objectives:
  - Convert low-level context information into ontological format
  - Align concurrent ontological low-level contexts of the same user
  - Generate unclassified high-level context instances from concurrent low-level contexts
- Subcomponents:
  - Context Mapper
  - Context Synchronizer
  - Context Instantiator

HLC Builder

MM V2.0_ICL_T5_V0.7_151107_ClaudiaVillalonga

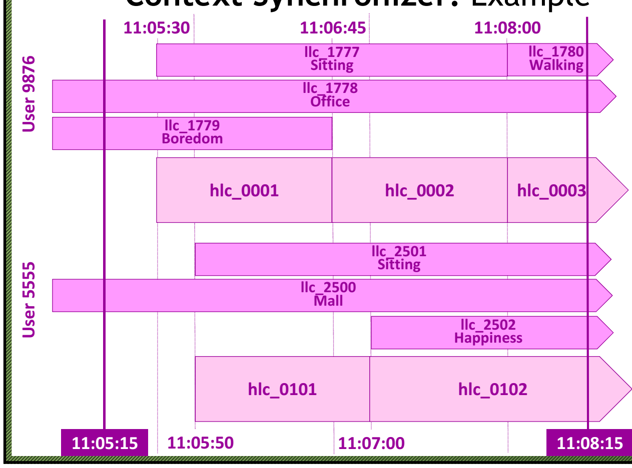
- **Context Synchronizer:** Business Logic:
  1. Retrieve the LLC instances which start within a time window.
  2. Retrieve the LLC instances which end within the time window.
  3. Order chronologically the LLC start and end events
  4. For each LLC instance retrieve the concurrent LLC
  5. Notify the Context Instantiator



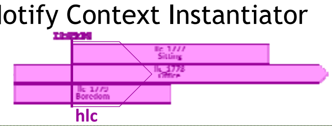
HLC Builder

MM V2.0_ICL_T5_V0.7_151107_ClaudiaVillalonga

- **Context Synchronizer:** Example



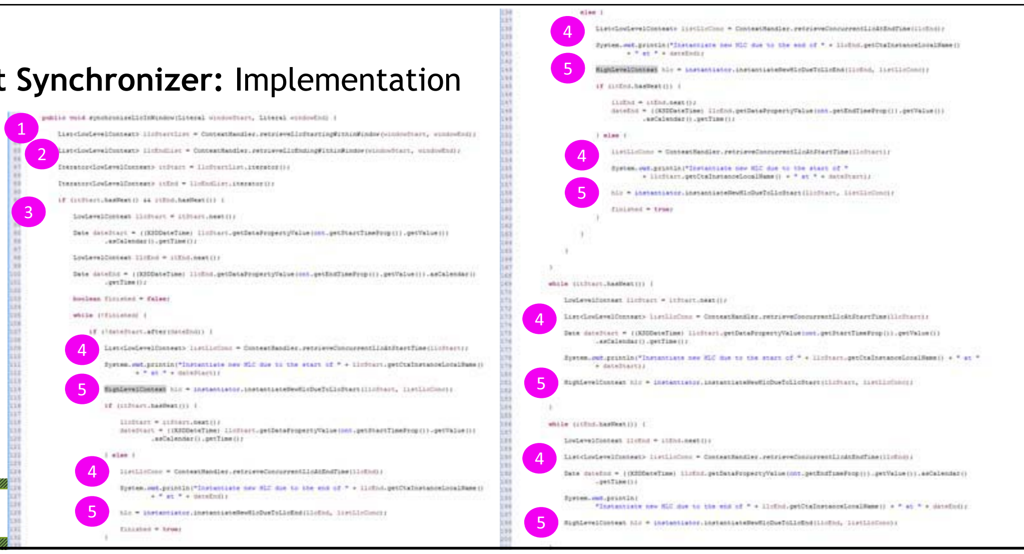
1. LLC instances starting within the window: llc_1777, llc_2501, llc_2502, llc_1780
2. LLC instances ending within the window: llc_1779
3. Order chronologically
4. Concurrent LLC for llc_1777: llc_1778 and llc_1779
5. Notify Context Instantiator



HLC Builder

MM V2.0_ICL_T5_V0.7_151107_ClaudiaVillalonga

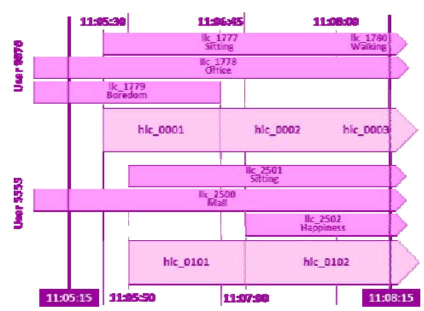
- **Context Synchronizer:** Implementation

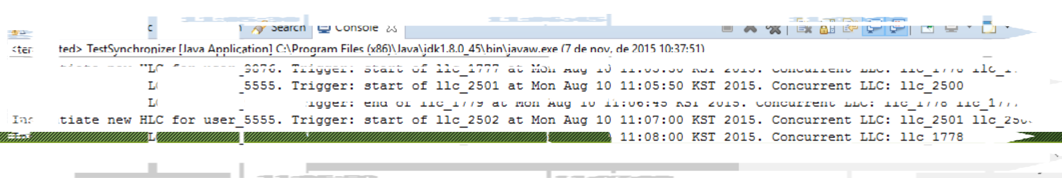


HLC Builder

MM V2.0_ICL_T5_V0.7_151107_ClaudiaVillalonga

- **Context Synchronizer:** Testing

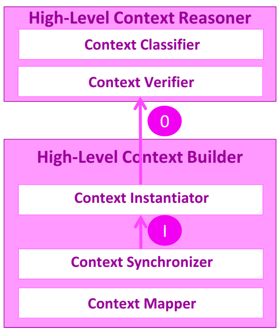




HLC Builder

MM V2.0_ICL_T5_V0.7_151107_ClaudiaVillalonga

- **Context Instantiatior:** Business Logic to create a new unclassifed High Level Context instance whenever the Context Synchronizer has detected a change in a LLC



I **INPUT:**
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 .

Concurrent LLC

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

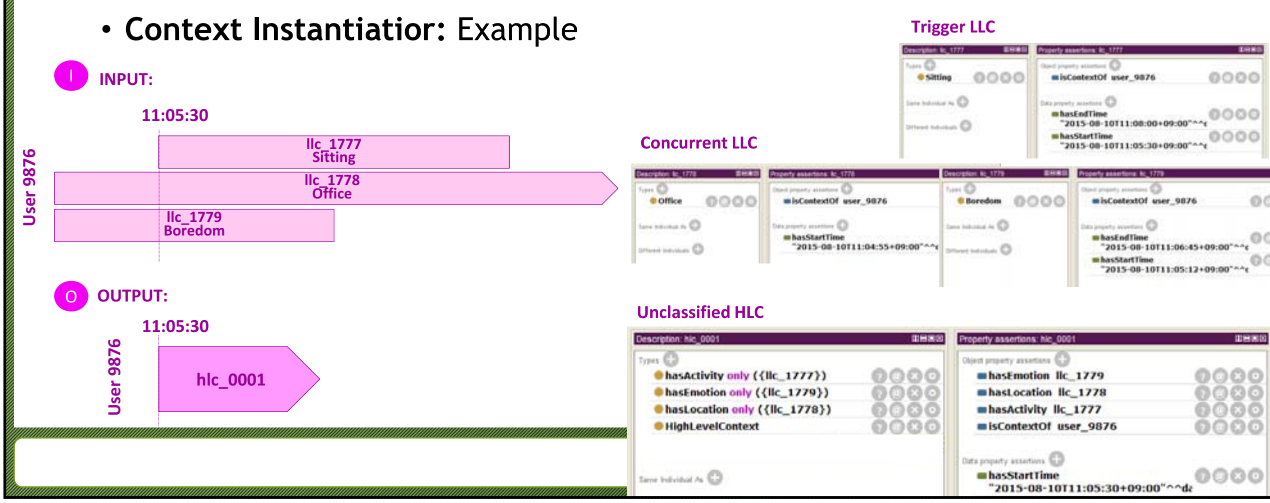
O **OUTPUT:**
Unclassified HLC

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}) .

HLC Builder

MM V2.0_ICL_T5_V0.7_151107_ClaudiaVillalonga

- **Context Instantiatior:** Example



HLC Builder

MM V2.0_ICL_T5_V0.7_151107_ClaudiaVillalonga

- **Context Instantiatior:** Implementation

```java
public HighLevelContext instantiateNewHlcDueToLlcStart(LowLevelContext llc, List<LowLevelContext> listLlcConc){

    OntModel hlcModel = createHlcModel();
    Individual hlcIndiv = createHlcIndiv(hlcModel);
    setUserForHlc(hlcIndiv, llc);
    setStartTimeForHlcDueToLlcStart(hlcIndiv, llc);

    boolean act = false;
    boolean loc = false;
    boolean emo = false;

    switch (llc.getLlcCategoryName()){

        case HLCA.activityClassName:

            setLlcBeingPartOfHlc(hlcModel, hlcIndiv, llc, ont.getActivityProp());
            act = true;
            break;

        case HLCA.locationClassName:

            setLlcBeingPartOfHlc(hlcModel, hlcIndiv, llc, ont.getLocationProp());
            loc = true;
            break;

        case HLCA.emotionClassName:

            setLlcBeingPartOfHlc(hlcModel, hlcIndiv, llc, ont.getEmotionProp());
            emo = true;
            break;

    }

        Iterator<LowLevelContext> itConc = listLlcConc.iterator();

        while(itConc.hasNext()) {

            LowLevelContext llcConc = itConc.next();

            switch (llcConc.getLlcCategoryName()){

            case HLCA.activityClassName:

                setLlcBeingPartOfHlc(hlcModel, hlcIndiv, llcConc, ont.getActivityProp());
                act = true;
                break;

            case HLCA.locationClassName:

                setLlcBeingPartOfHlc(hlcModel, hlcIndiv, llcConc, ont.getLocationProp());
                loc = true;
                break;

            case HLCA.emotionClassName:

                setLlcBeingPartOfHlc(hlcModel, hlcIndiv, llcConc, ont.getEmotionProp());
                emo = true;
                break;

            }

        }

        if(!act)

            setNoLlcBeingPartOfHlc(hlcModel, hlcIndiv, ont.getLlcCategoryClass(HLCA.activityClassName), ont.getActivityProp());

        if(!loc)

            setNoLlcBeingPartOfHlc(hlcModel, hlcIndiv, ont.getLlcCategoryClass(HLCA.locationClassName), ont.getLocationProp());

        if(!emo)

            setNoLlcBeingPartOfHlc(hlcModel, hlcIndiv, ont.getLlcCategoryClass(HLCA.emotionClassName), ont.getEmotionProp());

        this.hlcInstNum++;

        return new HighLevelContext(hlcModel);
    }
```

HLC Builder

MM V2.0_ICL_T5_V0.7_151107_ClaudiaVillalonga

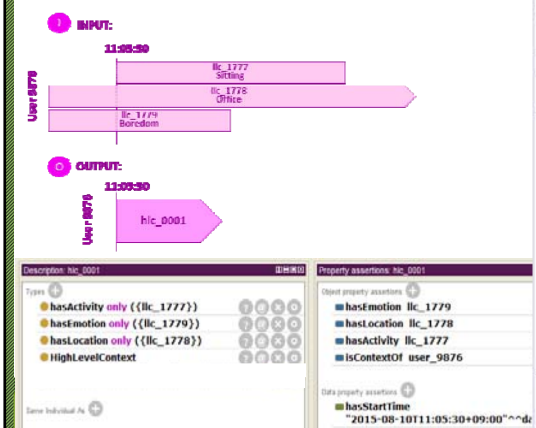
- **Context Instantiatior:** Implementation

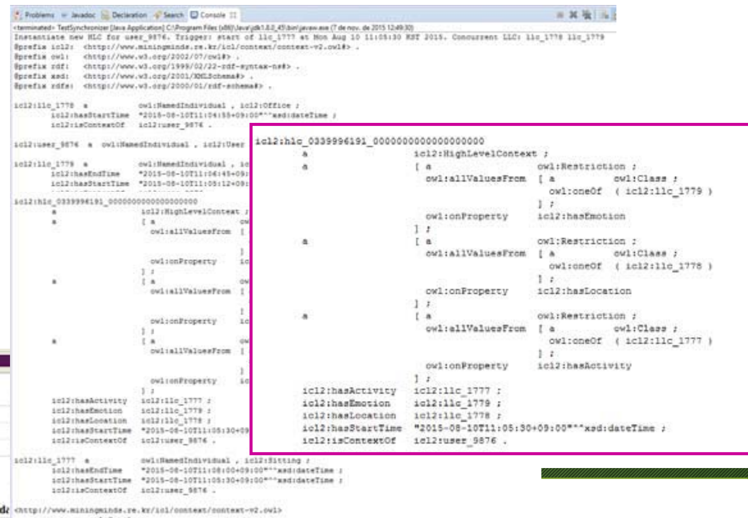
```java
public HighLevelContext instantiateNewHlcDueToLlcEnd(LowLevelContext llc, List<LowLevelContext> listLlcConc) {

    OntModel hlcModel = createHlcModel();
    Individual hlcIndiv = createHlcIndiv(hlcModel);
    setUserForHlc(hlcIndiv, llc);
    setStartTimeForHlcDueToLlcEnd(hlcIndiv, llc);

    boolean act = false;
    boolean loc = false;
    boolean emo = false;

    Iterator<LowLevelContext> itConc = listLlcConc.iterator();

    while (itConc.hasNext()) {

        LowLevelContext llcConc = itConc.next();

        switch (llcConc.getLlcCategoryName()) {

        case HLCA.activityClassName:

            setLlcBeingPartOfHlc(hlcModel, hlcIndiv, llcConc, ont.getActivityProp());
            act = true;
            break;

        case HLCA.locationClassName:

            setLlcBeingPartOfHlc(hlcModel, hlcIndiv, llcConc, ont.getLocationProp());
            loc = true;
            break;

        case HLCA.emotionClassName:

            setLlcBeingPartOfHlc(hlcModel, hlcIndiv, llcConc, ont.getEmotionProp());
            emo = true;
            break;
```

HLC Builder

MM V2.0_ICL_T5_V0.7_151107_ClaudiaVillalonga

- **Context Instantiatior:** Test



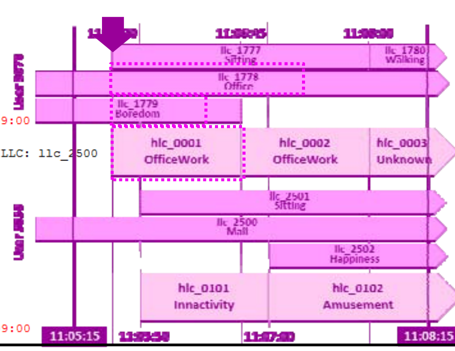


HLC Builder

MM V2.0_ICL_T5_V0.7_151107_ClaudiaVillalonga

- Integration of the **Context Synchronizer** and the **Context Instantiatior** with the **HLC Reasoner and HLC Synchronizer**

```
Instantiate new HLC for user_9876. Trigger: start of llc_1777 at Mon Aug 10 11:05:30 KST 2015. Concurrent LLC: llc_1778 llc_1779

Classifying 54 elements
Classifying:  100% complete in 00:00
Classifying finished in 00:00
Realizing 54 elements
Realizing:  100% complete in 00:00
Realizing finished in 00:00

HLC instance classified as: OfficeWork
[Notifier] new: OfficeWork, previous: none
DCL Notification Message: hlc_0321908006_0000000000000000000, OfficeWork, user_9876, 2015-08-10T11:05:30+09:00

Instantiate new HLC for user_5555. Trigger: start of llc_2501 at Mon Aug 10 11:05:50 KST 2015. Concurrent LLC: llc_2500

Classifying 54 elements
Classifying:  100% complete in 00:00
Classifying finished in 00:00
Realizing 54 elements
Realizing:  100% complete in 00:00
Realizing finished in 00:00

HLC instance classified as: Inactivity
[Notifier] new: Inactivity, previous: none
DCL Notification Message: hlc_0882177789_0000000000000000000, Inactivity, user_5555, 2015-08-10T11:05:50+09:00
```

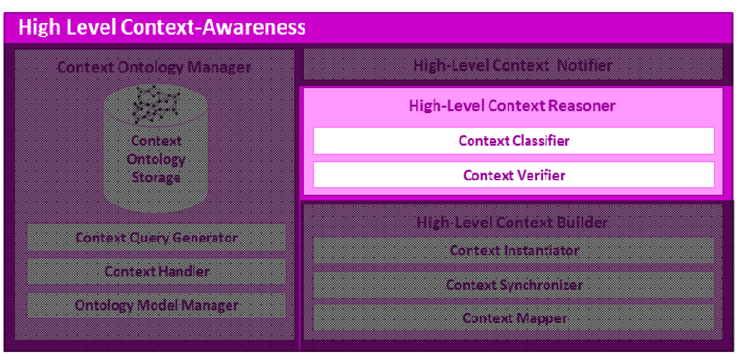


HLC Reasoner

Claudia Villalonga

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

High-level Context Reasoner

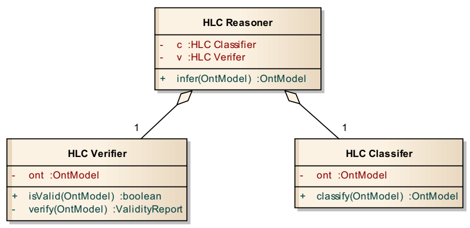


- Objectives:
  - Consistency check (validation and verification) of the unclassified high-level context instance versus the Context Ontology Model
  - Classification or identification of the context type to which the unclassified high-level context instance belongs
- Subcomponents:
  - Context Verifier
  - Context Classifier

HLC Reasoner

MM V2.0_ICL_T5_V0.1_150926_ClaudiaVillalonga

- Analysis of Jena in order to identify the Classes that might be reused in the MM implementation (e.g., org.apache.jena.ontology.OntModel)
- Definition of the java classes for the HLC Reasoner



HLC Reasoner

MM V2.0_ICL_T5_V0.2_151003_ClaudiaVillalonga

- Successful configuration of the **Pellet Reasoner** to work with Jena
- Generation of the **OntModels** structures from ontology OWL files
- Analysis of the **model configurations** and election of the best one

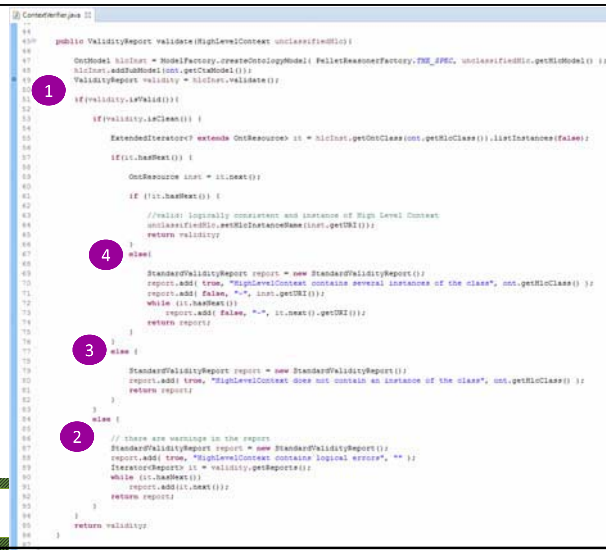
```


        // READ THE CONTEXT ONTOLOGY FROM FILE

        // create an empty model with pellet reasoner
        OntModel ontModel = ModelFactory.createOntologyModel( PelletReasonerFactory.THE_SPEC );

        // read the ontology from file
        ontModel.read(ontologyFile);


        // READ THE HIGH LEVEL CONTEXT INSTANCE FROM FILE

        // create an empty model with settings: OWL DL, in-memory storage, and without inference
        OntModel hlcModel = ModelFactory.createOntologyModel(OntModelSpec.OWL_DL_MEM);

        //ignore import of context ontology when creating the model
        hlcModel.getDocumentManager().setProcessImports(false);

        // read the instance from file
        hlcModel.read("file:"+ testInstance +".owl");

```

HLC Reasoner

MM V2.0_ICL_T5_V0.2_151003_ClaudiaVillalonga

- **HLC Verifier:** Definition of the business logic to validate an unclassified high-level context instance

1. Logical consistency check versus the ontology
2. Handling of warnings in logical consistency check
3. Verification that it is actually an instance of HLC
4. Verification that it only an instance of HLC

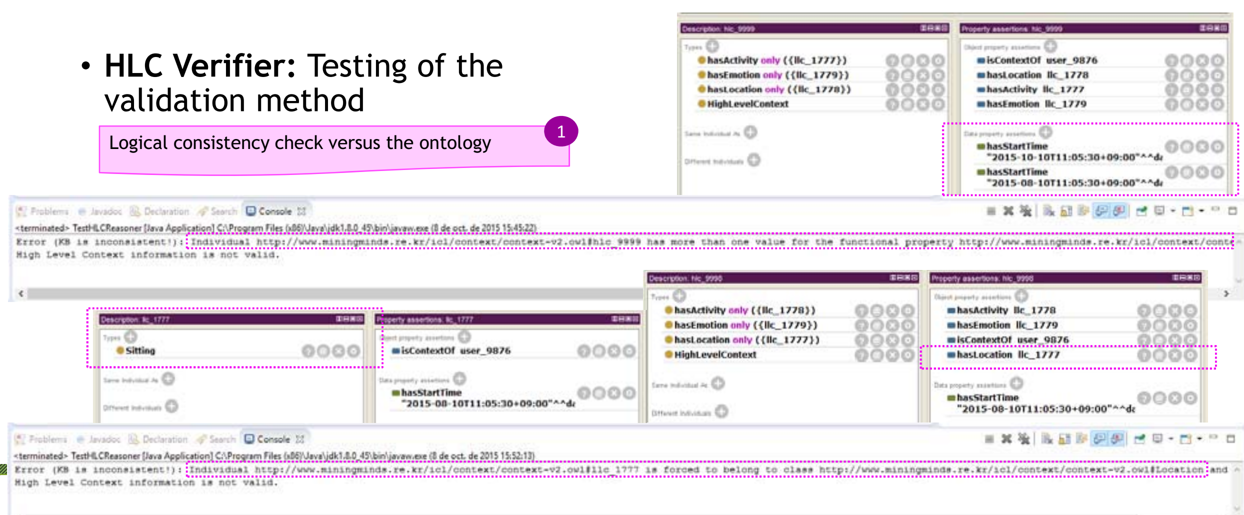


HLC Reasoner

MM V2.0_ICL_T5_V0.2_151003_ClaudiaVillalonga

- **HLC Verifier:** Testing of the validation method

1. Logical consistency check versus the ontology



HLC Reasoner

MM V2.0_ICL_T5_V0.2_151003_ClaudiaVillalonga

- **HLC Verifier:** Testing of the validation method

Verification that it is actually an instance of HLC (3)

```
Problems  Javadoc  Declaration  Search  Console
<terminated> TestHLCReasoner [Java Application] C:\Program Files (x86)\Java\jdk1.8.0_45\bin\javaw.exe (8 de oct. de 2015 15:59:22)
Error (HighLevelContext does not contain an instance of the class): http://www.miningminds.re.kr/icl/context/context-v2.owl#HighLevelContext
High Level Context information is not valid.
```

Verification that it only an instance of HLC (4)

```
Problems  Javadoc  Declaration  Search  Console
<terminated> TestHLCReasoner [Java Application] C:\Program Files (x86)\Java\jdk1.8.0_45\bin\javaw.exe (8 de oct. de 2015 15:57:50)
Error (HighLevelContext contains several instances of the class): http://www.miningminds.re.kr/icl/context/context-v2.owl#HighLevelContext
Warning (-): http://www.miningminds.re.kr/icl/context/context-v2.owl#hlc_0001
Warning (-): http://www.miningminds.re.kr/icl/context/context-v2.owl#hlc_0003
Warning (-): http://www.miningminds.re.kr/icl/context/context-v2.owl#hlc_0002
High Level Context information is not valid.
```

HLC Reasoner

MM V2.0_ICL_T5_V0.3_151010_ClaudiaVillalonga

- **HLC Classifier:** Definition of the business logic for the classification of an unclassified high-level context instance

Classification of the HLC instance using Pellet (1)

Realization of the HLC instance (2)

Validation of belonging to subclasses of HLC (3)

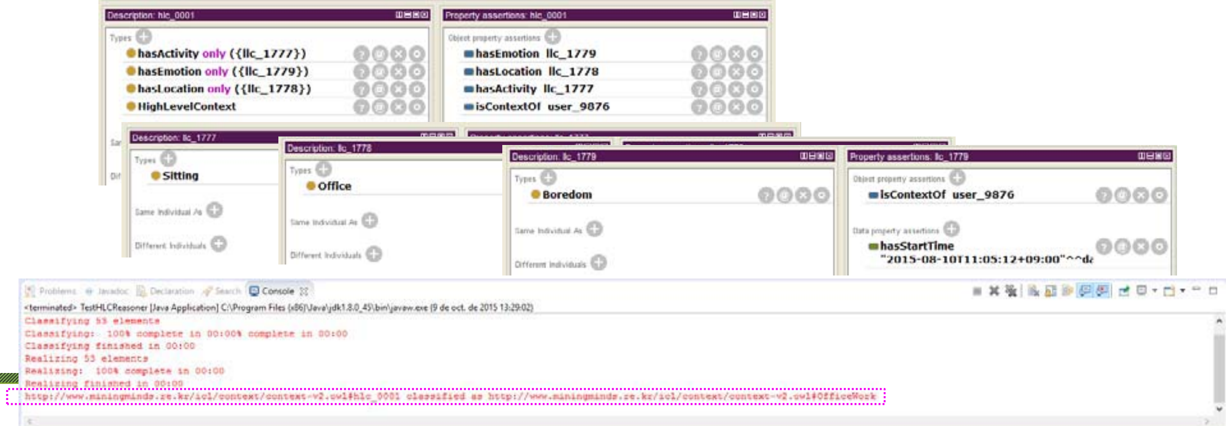
ContextClassifier.java

```
 * @author Claudia Villalonga
 */
public class ContextClassifier {

    private ContextOntology ont;

    /**
     *
     * @param ont
     */
    public ContextClassifier(ContextOntology ont){
        this.ont = ont;
    }


    public HighLevelContext classify(HighLevelContext unclassifiedHlc){


        OntModel hlcInst = ModelFactory.createOntologyModel( PelletReasonerFactory.THE_SPEC, unclassifiedHlc.getHlcModel() );

        hlcInst.addSubModel(ont.getCtxModel());

        ((PelletInfGraph) hlcInst.getGraph()).classify();

        OntClass c = hlcInst.getIndividual(unclassifiedHlc.getHlcInstanceName()).listOntClasses(true).next();

        if(!c.getURI().equals(ont.getHlcClass())){

            // Classification OK and HLC belongs to one of the classes defined in the MM Context Ontology.

            // Add inferred class to HLC Instance
            Individual hlcIndiv = unclassifiedHlc.getHlcModel().getIndividual(unclassifiedHlc.getHlcInstanceName());

            hlcIndiv.addOntClass(c);

            unclassifiedHlc.setHlcType(c.getURI());


        }

        return unclassifiedHlc;
    }
}
```

HLC Reasoner

MM V2.0_ICL_T5_V0.3_151010_ClaudiaVillalonga

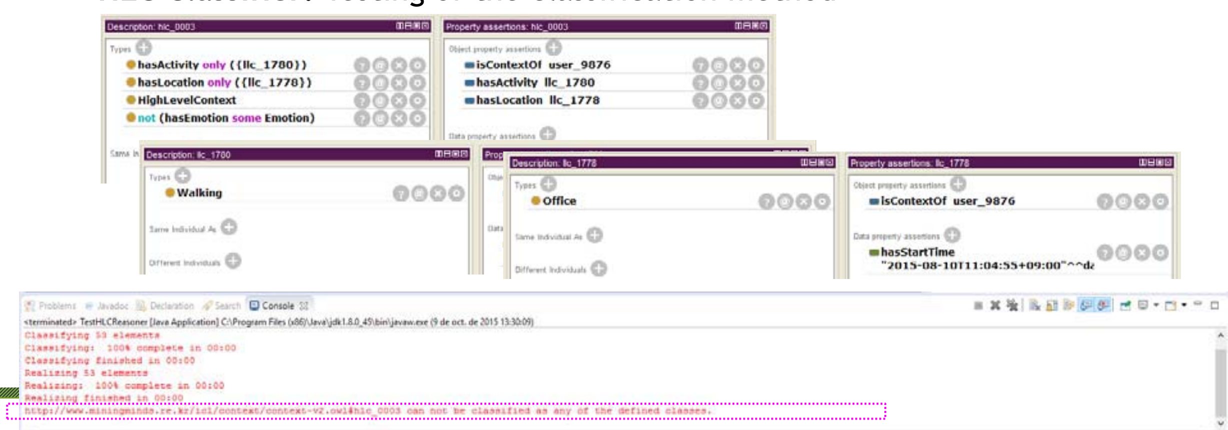
- **HLC Classifier:** Testing of the classification method



HLC Reasoner

MM V2.0_ICL_T5_V0.3_151010_ClaudiaVillalonga

- **HLC Classifier:** Testing of the classification method



HLC Reasoner

MM V2.0_ICL_T5_V0.3_151010_ClaudiaVillalonga

- **HLC Reasoner:**
  - Integration of the HLC Verifier and HLC Classifier in order to complete the process of identification of high level context
  - Test of the HLC Reasoner

```java
// create new ContextOntology class
ContextOntology ont = new ContextOntology(ontModel, NS, hlcClass);

// Validate the ontology
ValidityReport ontReport = ont.validate();

if(ontReport.isValid()) {

        HLCReasoner r = new HLCReasoner(ont);

        // every time a new unclassified high level context (hlc) is built, the infer method should be called
        HighLevelContext classifiedHlc = r.infer(hlc);

        if (!classifiedHlc.isInstanceOfHlc())

                System.out.println("High Level Context information is not valid.");

        else if (!classifiedHlc.isClassified())

                System.out.println(classifiedHlc.getHlcInstanceName() + " can not be classified as any of the defined classes.");

        else

                System.out.println(classifiedHlc.getHlcInstanceName() + " classified as " + classifiedHlc.getHlcType());

}
else

        printIterator(ontReport.getReports(), "Error validating the ontology.");
```

```java
public class HLCReasoner {

    ContextClassifier cc;
    ContextVerifier cv;

    public HLCReasoner(ContextOntology ont){

        this.cc = new ContextClassifier(ont);
        this.cv = new ContextVerifier(ont);

    }



    /**
     *
     * @param unclassified
     * @return
     */
    public HighLevelContext infer(HighLevelContext unclassified){

        if(cv.isValid(unclassified))
            return cc.classify(unclassified);

        return unclassified;
    }
```

HLC Reasoner

MM V2.0_ICL_T5_V0.5_151024_ClaudiaVillalonga

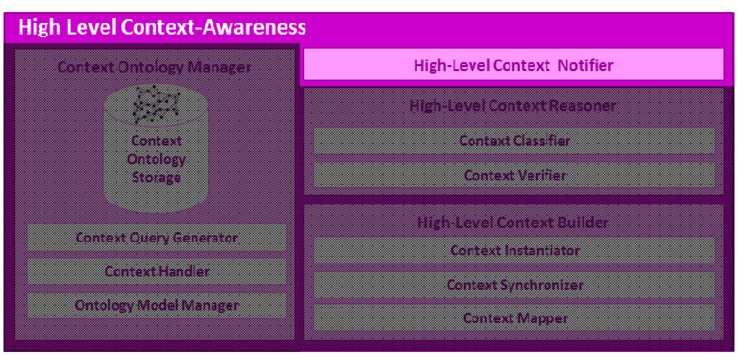
- Integration of the **HLCReasoner** with the **HLCNotifier** and the **Context Ontology Manager**

```java
// CREATE THE CONTEXT ONTOLOGY MANAGER
ContextOntologyManager mng = new ContextOntologyManager(tdbDirectory, ontModel);


if(mng.correctInitialization()) {

    // CREATE THE HLC BUILDER

    // CREATE THE HLC REASONER
    HLCReasoner r = new HLCReasoner(mng.getInfOnt());

    // CREATE THE HLC NOTIFIER
    HLCNotifier n = new HLCNotifier(mng.getOnt());

    // SIMULATE HLC BUILDER: READ THE HIGH LEVEL CONTEXT INSTANCE FROM FILE

    // create an empty model with settings: OWL DL, in-memory storage, and without inference
    OntModel hlcModel = ModelFactory.createOntologyModel(OntModelSpec.OWL_DL_MEM);

    // read the instance from file
    hlcModel.read("file:"+ testInstance +".owl");

    // create new HighLevelContext class
    HighLevelContext unclassifiedHlc = new HighLevelContext(hlcModel);

    // CALL THE REASONER EVERY TIME A NEW UNCLASSIFIED HLC IS CRATED
    HighLevelContext classifiedHlc = r.inferHlc(unclassifiedHlc);

    // CALL THE NOTIFIER EVERY TIME THE REASONER HAS CLASSIFIED A HLC
    n.notify(classifiedHlc);

}
else

    System.out.println("Error initializing the Context Ontology Manager.");
```

HLC Notifier

Claudia Villalonga

High-Level Context Notifier



- Objectives:
  - Communication of the newly recognized high-level context to Data Curation Layer for storage into the LifeLog
  - Storage of high-level context into the Context Ontology Storage

HLC Notifier

MM V2.0_ICL_T5_V0.4_151017_ClaudiaVillalonga

- Business logic to determine whether the newly recognized HLC needs to be notified

1. New HLC is classified and the previous one was unknown OR Both new and previous HLC are classified and different
2. New HLC is unknown and the previous one was classified
3. New HLC is classified and there is no previous context
4. New HLC is unknown and there is no previous context

HLC Notifier

MM V2.0_ICL_T5_V0.4_151017_ClaudiaVillalonga

- Management of the start and end times on the context instance

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"/>
```

```
Literal time = newHlc.getDataPropertyValue(ont.getStartTimeProp());
Resource user = newHlc.getObjectPropertyValue(ont.getContextOfProp());

if(hlcModel != null) {

    // Create a HighLevelContext class with the instance retrieved from the Triplestore
    HighLevelContext lastHlc = new HighLevelContext(hlcModel, ont);

    // Set end time of previous HLC Instance
    lastHlc.setDataPropertyValue(ont.getEndTimeProp(), time);
```

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"/>
```

HLC Notifier

MM V2.0_ICL_T5_V0.5_151024_ClaudiaVillalonga

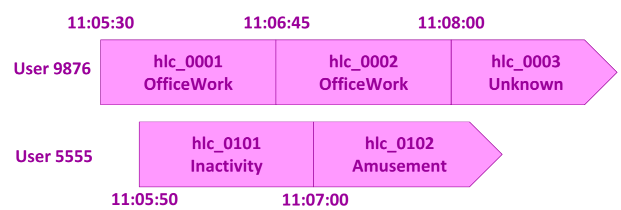
- Retrieval and storage of classified HLC on the Triplestore (integration with Context Ontology Manager)

```java
    public void notify(HighLevelContext newHlc) {

        if (newHlc.isValidInstanceOfHlc()) { // New High Level Context Instance
                                            // is valid.

            // Blocking operation: Retrieve previous HLC Instance from the
            // Triplestore. Set end time of previous HLC Instance. And store the
            // two instances to the Triplestore
            HighLevelContext previousHlc = ContextHandler.retrievePreviousHlcAndStoreNew(newHlc);

            if (previousHlc == null || !newHlc.getCtxTypeName().equals(previousHlc.getCtxTypeName()))

                notifyDCL(newHlc.getCtxInstanceLocalName(), newHlc.getCtxTypeLocalName(),
                        newHlc.getObjectPropertyValue(ont.getContextOfProp()).getLocalName(),
                        newHlc.getDataPropertyValue(ont.getStartTimeProp()).getLexicalForm());

        }
    }
```

HLC Notifier

MM V2.0_ICL_T5_V0.5_151024_ClaudiaVillalonga

- HLC Notifier testing and validation

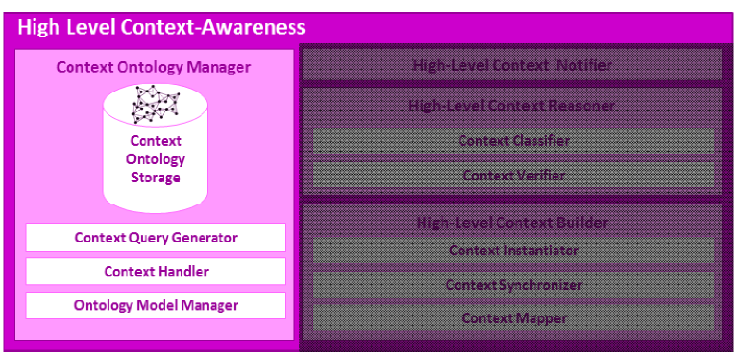



```
Classifying 53 elements
Classifying:  100% complete in 00:00
Classifying finished in 00:00
Realizing 53 elements
Realizing:  100% complete in 00:00
Realizing finished in 00:00
[Notifier] new: OfficeWork, previous: none
DCL Notification Message: hlc_0001, OfficeWork, user_9876, 2015-08-10T11:05:30+09:00

[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
```

Context Ontology Manager

Claudia Villalonga

Context Ontology Manager

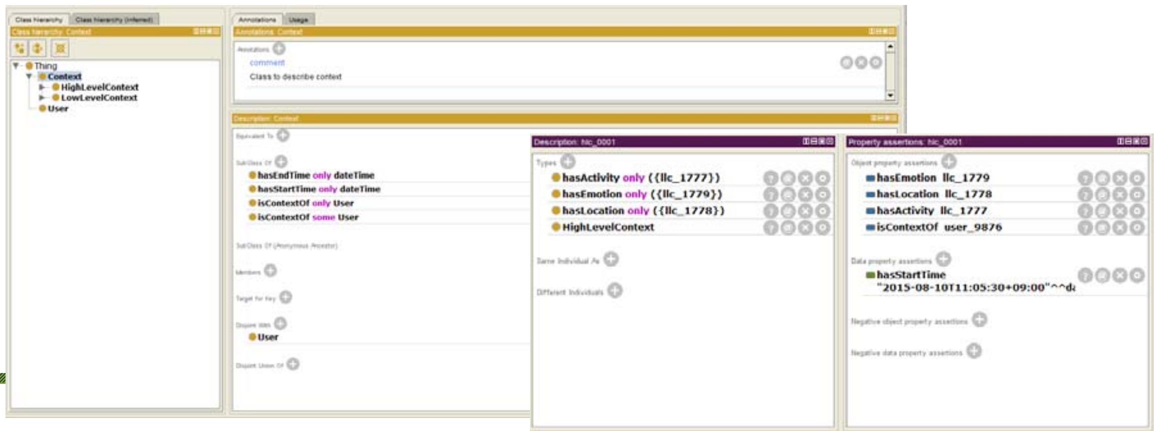


- Objectives:
  - Provide persistent storage for the Context Ontology and Context Instances
  - Management of the interactions with the persisted Context Information
- Subcomponents:
  - Context Ontology Storage
  - Ontology Model Manager
  - Context Handler
  - Context Query Generator

Context Ontology Manager

MM V2.0_ICL_T5_V0.2_151003_ClaudiaVillalonga

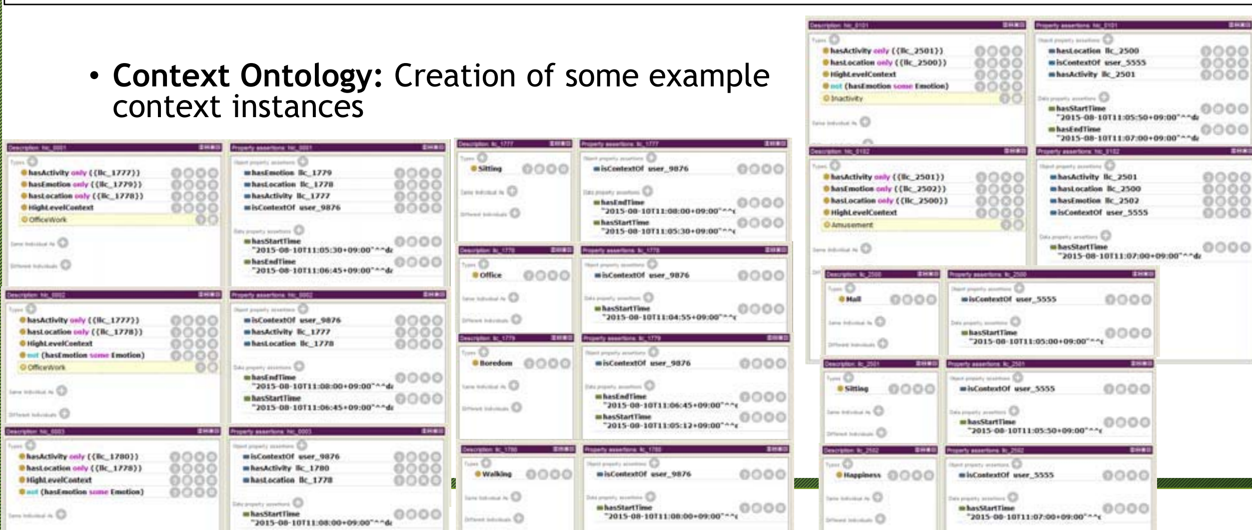
- **Context Ontology:** Update to include time and user.



Context Ontology Manager

MM V2.0_ICL_T5_V0.3_151010_ClaudiaVillalonga

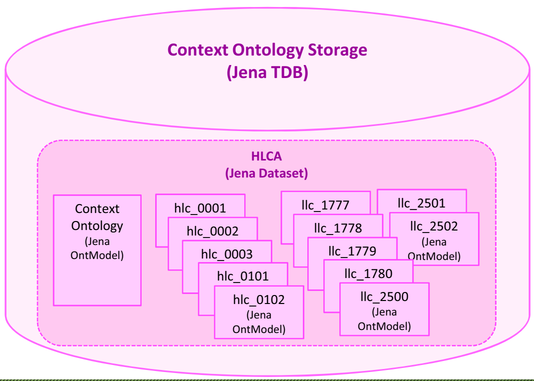
- **Context Ontology:** Creation of some example context instances



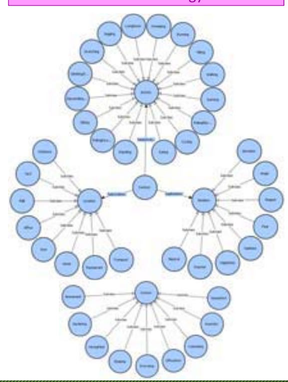
Context Ontology Manager

MM V2.0_ICL_T5_V0.4_151017_ClaudiaVillalonga

- **Context Ontology Storage:** Definition of an efficient storage mechanism for the Triplestore in order to optimize the queries and minimize the data transfer



Context Ontology



Context Instances

hl_0001

hlc_0001 rdf:type Context .
hlc_0001 rdf:type OfficeWork .
hlc_0001 hasActivity llc_1777 .
hlc_0001 rdf:type hasActivity only ({llc_1777}) .
hlc_0001 hasLocation llc_1778 .
hlc_0001 rdf:type hasLocation only ({llc_1778}) .
hlc_0001 hasEmotion llc_1779 .
hlc_0001 rdf:type hasEmotion only ({llc_1779}) .
hlc_0001 hasStartTime “2015-08-10T11:05:30”^^dateTime .
hlc_0001 isContextOf user_9876 .

llc_1777

llc_1777 rdf:type Sitting .
llc_1777 hasStartTime “2015-08-10T11:05:30”^^dateTime .
llc_1777 isContextOf user_9876 .

llc_1778

llc_1778 rdf:type Office .
llc_1778 hasStartTime “2015-08-10T11:04:55”^^dateTime .
llc_1778 isContextOf user_9876 .

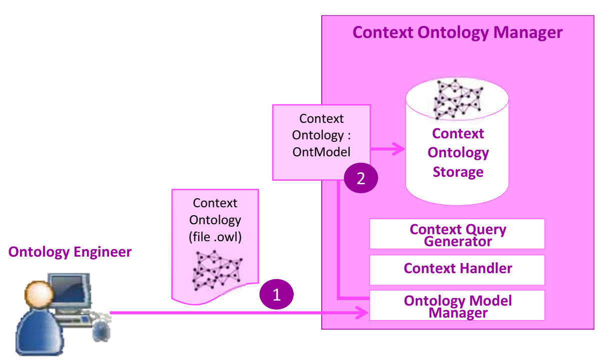
llc_1779

llc_1779 type Boredom .
llc_1779 hasStartTime “2015-08-10T11:05:12”^^dateTime .
llc_1779 isContextOf user_9876 .

Context Ontology Manager

MM V2.0_ICL_T5_V0.4_151017_ClaudiaVillalonga

- **Ontology Model Manager:** Business logic to store the context ontology to the Triplestore




```
public class OntologyModelManager {


    private Dataset dataset;

    public OntologyModelManager(String directory){

        this.dataset =  TDBFactory.createDataset(directory) ;


    }

    public void loadOntology(String uri, OntModel model) {

        dataset.begin(ReadWrite.WRITE);

        // Store context Ontology model to DB
        dataset.addNamedModel(uri, model);

        dataset.commit();
        dataset.end();

    }

    public void loadOntology(ContextOntology ont) {

        loadOntology(ont.getNS(), ont.getCtxModel());

    }

}
```

Context Ontology Manager

MM V2.0_ICL_T5_V0.2_151003_ClaudiaVillalonga

- **Ontology Model Manager:** Implementation of the Ontology Model verification mechanism.

1. Logical consistency
2. Handling of warnings in logical consistency check
3. Verification of the namespace
4. Verification of the definition of HLC class

```
public ValidityReport validate(){

    ValidityReport validity = ctxModel.validate();

    if(validity.isValid()){

        if(!validity.isClean()) {

            // there are warnings in the report (e.g., uninstanciable classes)
            StandardValidityReport report = new StandardValidityReport();
            report.add( true, "Context Ontology contains errors in the class definition", "" );

            Iterator<Report> it = validity.getReports();
            while (it.hasNext())
                report.add(it.next());

            return report;


        }

        if(ctxModel.getNsURIPrefix(ns) == null) {

                StandardValidityReport report = new StandardValidityReport();
                report.add( true, "Context Ontology does not contain the namespace", ns );
                return report;

        }

        if(ctxModel.getOntClass(hlcClass) == null) {

                StandardValidityReport report = new StandardValidityReport();
                report.add( true, "Context Ontology does not contain the class", hlcClass );
                return report;

        }


    }

    return validity;

}
```

Context Ontology Manager

MM V2.0_ICL_T5_V0.2_151003_ClaudiaVillalonga

- **Ontology Model Manager:** Testing of the Ontology Model verification

1. Logical consistency

```
Problems  Javadoc  Declaration  Search  Console
<terminated> TestHLCReasoner [Java Application] C:\Program Files (x86)\Java\jdk1.8.0_45\bin\javaw.exe (8 de oct. de 2015 18:26:18)
Error validating the ontology.
==============================
Error (KB is inconsistent!): Individual http://www.miningminds.re.kr/icl/context/context-v2.owl#hlc_0003 has more than one value for the functional property http://www.miningminds.re.kr/icl/context/conte
```

2. Handling of warnings in logical consistency check

```
Problems  Javadoc  Declaration  Search  Console
<terminated> TestHLCReasoner [Java Application] C:\Program Files (x86)\Java\jdk1.8.0_45\bin\javaw.exe (8 de oct. de 2015 18:28:42)
Error validating the ontology.
==============================
Error (Context Ontology contains errors in the class definition):
Warning (Unsatisfiable class): http://www.miningminds.re.kr/icl/context/context-v2.owl#OfficeWork
```

3. Verification of the namespace

```
Problems  Javadoc  Declaration  Search  Console
<terminated> TestHLCReasoner [Java Application] C:\Program Files (x86)\Java\jdk1.8.0_45\bin\javaw.exe (8 de oct. de 2015 18:23:57)
Error validating the ontology.
==============================
Error (Context Ontology does not contain the namespace): http://www.miningminds.re.kr/icl/context/context-v2.owl#
```

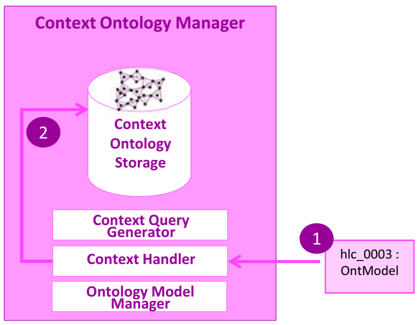
4. Verification of the definition of HLC class

```
Problems  Javadoc  Declaration  Search  Console
<terminated> TestHLCReasoner [Java Application] C:\Program Files (x86)\Java\jdk1.8.0_45\bin\javaw.exe (8 de oct. de 2015 18:25:32)
Error validating the ontology.
==============================
Error (Context Ontology does not contain the class): http://www.miningminds.re.kr/icl/context/context-v2.owl#HighLevelContext
```

Context Ontology Manager

MM V2.0_ICL_T5_V0.4_151017_ClaudiaVillalonga

- **Context Handler:** Business logic to store and update context instances in the Triplestore

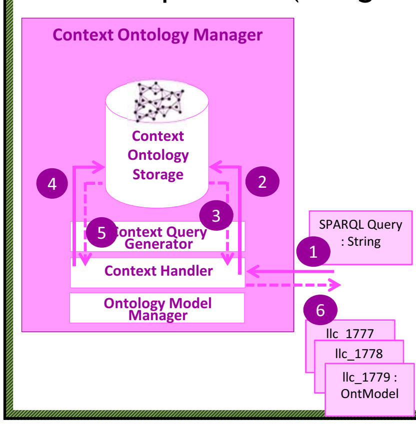



```java
public class ContextHandler {

    private Dataset dataset;

    public ContextHandler(String directory){

        this.dataset =  TDBFactory.createDataset(directory) ;

    }

    public void addContextInstance(String uri, OntModel model){

        //// WRITE to store Context Ontology

        dataset.begin(ReadWrite.WRITE);

        // Store context Ontology model to DB
        dataset.addNamedModel(uri, model);

        dataset.commit();
        dataset.end();

    }

    public void updateContextInstance(String uri, OntModel model){
        //TODO Maybe we want to do some verification first...
        addContextInstance(uri, model);
    }

    public void addHlcInstance(HighLevelContext hlc){

        addContextInstance(hlc.getHlcInstanceName(), hlc.getHlcModel());

    }

}
```

Context Ontology Manager

MM V2.0_ICL_T5_V0.4_151017_ClaudiaVillalonga

- **Context Handler:** Business logic to request context instances from the Triplestore (using SPARQL queries)




```java
*/
public class ContextQueryManager {


    private Dataset dataset;

    public ContextQueryManager(String directory){

        this.dataset =  TDBFactory.createDataset(directory) ;
        TDB.getContext().set(TDB.symUnionDefaultGraph, true) ;

    }


    public List<OntModel> getResultsFromSparqlQuery(String sparqlQuery){

        ResultSet rs = executeSparqlQuery(sparqlQuery);

        String varName = rs.getResultVars().iterator().next();

        List<OntModel> results = new ArrayList<OntModel>();

         while(rs.hasNext()) {

             results.add(retrieveContextInstance(rs.next(), varName));

         }

         return results;
    }
```

```java
    public ResultSet executeSparqlQuery(String sparqlQuery){


        dataset.begin(ReadWrite.WRITE) ;

         QueryExecution qExec = QueryExecutionFactory.create(sparqlQuery, dataset);
         ResultSet rs = qExec.execSelect() ;

         dataset.commit() ;

         dataset.end() ;

         return rs;

    }


    public OntModel retrieveContextInstance(QuerySolution sol, String varName){

        return retrieveContextInstance(sol.getResource(varName).getURI());

    }


    public OntModel retrieveContextInstance(String modelName){

        dataset.begin(ReadWrite.READ) ;

        Model inst = dataset.getNamedModel(modelName);
        dataset.end() ;

        return ModelFactory.createOntologyModel( OntModelSpec.OWL_DL_MEM, inst );
    }
```

Context Ontology Manager

MM V2.0_ICL_T5_V0.5_151024_ClaudiaVillalonga

- **Context Handler:** Method to store a new HLC and retrieve the previous one

```java
public static HighLevelContext retrievePreviousHlcAndStoreNew(HighLevelContext newHlc) {

    dataset.begin(ReadWrite.WRITE);

    Query sparqlQuery = ContextQueryGenerator.generateQueryForPreviousValidAndUnfinishedHlc(newHlc, ont);

    QueryExecution qExec = QueryExecutionFactory.create(sparqlQuery, dataset);
    ResultSet rs = qExec.execSelect();

    HighLevelContext previousHlc = null;

    if (rs.hasNext()) { // There is a HLC which is valid and open
                        // (unfinished) at the start time of the newHlc

        Model inst = dataset.getNamedModel(rs.next().getResource(HLCA.hlcSparqlVar).getURI());
        OntModel previousHlcModel = ModelFactory.createOntologyModel(OntModelSpec.OWL_DL_MEM, inst);
        previousHlc = new HighLevelContext(previousHlcModel, ont);

        previousHlc.setDataPropertyValue(ont.getEndTimeProp(), newHlc.getDataPropertyValue(ont.getStartTimeProp()));

    }

    else {

        sparqlQuery = ContextQueryGenerator.generateQueryForPreviousValidAndFinishedHlc(newHlc, ont);

        qExec = QueryExecutionFactory.create(sparqlQuery, dataset);
        rs = qExec.execSelect();

        if (rs.hasNext()) { // There is a HLC which was valid at the start
                            // time of the newHlc and which finished later

            Model inst = dataset.getNamedModel(rs.next().getResource(HLCA.hlcSparqlVar).getURI());
            OntModel previousHlcModel = ModelFactory.createOntologyModel(OntModelSpec.OWL_DL_MEM, inst);
            previousHlc = new HighLevelContext(previousHlcModel, ont);

            newHlc.setDataPropertyValue(ont.getEndTimeProp(),
                    previousHlc.getDataPropertyValue(ont.getEndTimeProp()));

            previousHlc.setDataPropertyValue(ont.getEndTimeProp(),
                    newHlc.getDataPropertyValue(ont.getStartTimeProp()));

        }
```

```java
        else {

            // There are no previous instances of HLC in the DB
            // Query to get the StartTime of the next HLC which is posterior
            // to the new Instance StartTime

            sparqlQuery = ContextQueryGenerator.generateQueryForStartTimeOfNextHlc(newHlc, ont);

            qExec = QueryExecutionFactory.create(sparqlQuery, dataset);
            rs = qExec.execSelect();

            if (rs.hasNext()) {

                QuerySolution soln = rs.next();

                Iterator<String> it = soln.varNames();

                if (it.hasNext()) // There is future instance of HLC in the
                                  // DB with the retrieved StartTime

                    newHlc.setDataPropertyValue(ont.getEndTimeProp(), soln.getLiteral(it.next()));
```

Context Ontology Manager

MM V2.0_ICL_T5_V0.5_151024_ClaudiaVillalonga

- **Context Handler:**Method to set end time of the previously stored LLC

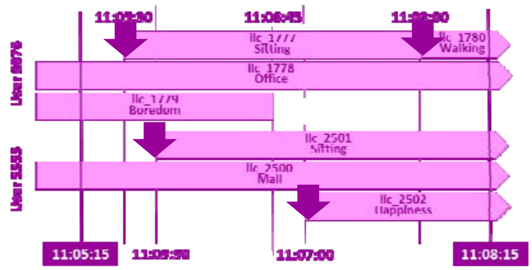
```java
public static void finalisePreviousLlc(OntClass category, Resource user, Literal time) {

    dataset.begin(ReadWrite.WRITE);

    Query sparqlQuery = ContextQueryGenerator.generateQueryForPreviousValidAndUnfinishedLlc(category, user, time,
            ont);

    QueryExecution qExec = QueryExecutionFactory.create(sparqlQuery, dataset);
    ResultSet rs = qExec.execSelect();

    if (rs.hasNext()) { // There is a HLC which is valid and open
                        // (unfinished) at the start time of the newHlc

        Model inst = dataset.getNamedModel(rs.next().getResource(HLCA.llcSparqlVar).getURI());
        OntModel previousLlcModel = ModelFactory.createOntologyModel(OntModelSpec.OWL_DL_MEM, inst);
        LowLevelContext previousLlc = new LowLevelContext(previousLlcModel, ont);

        previousLlc.setDataPropertyValue(ont.getEndTimeProp(), time);

    }

    else {

        sparqlQuery = ContextQueryGenerator.generateQueryForPreviousValidAndFinishedLlc(category, user, time, ont);

        qExec = QueryExecutionFactory.create(sparqlQuery, dataset);
        rs = qExec.execSelect();

        if (rs.hasNext()) { // There is a HLC which was valid at the start
                            // time of the newHlc and which finished later

            Model inst = dataset.getNamedModel(rs.next().getResource(HLCA.llcSparqlVar).getURI());
            OntModel previousLlcModel = ModelFactory.createOntologyModel(OntModelSpec.OWL_DL_MEM, inst);
            LowLevelContext previousLlc = new LowLevelContext(previousLlcModel, ont);

            previousLlc.setDataPropertyValue(ont.getEndTimeProp(), time);

        }

    }

    dataset.commit();

    dataset.end();

}
```

Context Ontology Manager

MM V2.0_ICL_T5_V0.6_151031_ClaudiaVillalonga

- **Context Handler:** Implementation of the method to retrieve from Triplestore all LLC which have their start time within a time window (to be used by the HLC Builder)

```
public static List<LowLevelContext> retrieveLlcStartingWithinWindow(Literal windowStart, Literal windowEnd) {

    Query sparqlQuery = ContextQueryGenerator.generateQueryForLlcWithStartTimeInWindow(windowStart, windowEnd, ont);

    ResultSet results = executeSparqlQuery(sparqlQuery);

    return (List<LowLevelContext>) (List<?>) getContextForSparqlResults(results);

}
```

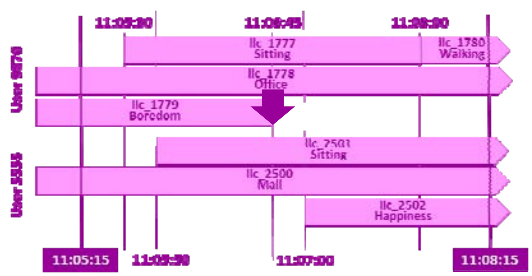


Context Ontology Manager

MM V2.0_ICL_T5_V0.6_151031_ClaudiaVillalonga

- **Context Handler:** Implementation of the method to retrieve from Triplestore all LLC which have their end time within a time window and do not have a contiguous LLC (to be used by the HLC Builder)

```
public static List<LowLevelContext> retrieveLlcEndingWithinWindow(Literal windowStart, Literal windowEnd) {

    Query sparqlQuery = ContextQueryGenerator.generateQueryForLlcWithEndTimeInWindowAndNoNextLlc(windowStart, windowEnd, ont);

    ResultSet results = executeSparqlQuery(sparqlQuery);

    return (List<LowLevelContext>) (List<?>) getContextForSparqlResults(results);

}
```

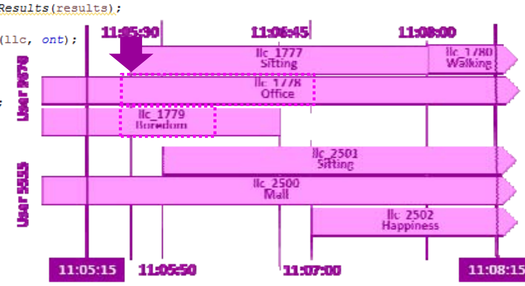


Context Ontology Manager

MM V2.0_ICL_T5_V0.6_151031_ClaudiaVillalonga

- **Context Handler:** Implementation of the method to retrieve from the Triplestore the LLC instances which are concurrent at the start time of a given LLC (to be used by the HLC Builder)

```java
public static List<LowLevelContext> retrieveConcurrentLlcAtStartTime(LowLevelContext llc) {

    Query sparqlQuery = ContextQueryGenerator.generateQueryForConcurrentAtStartTimeUnfinishedLlc(llc, ont);

    ResultSet results = executeSparqlQuery(sparqlQuery);

    List<LowLevelContext> llcList = (List<LowLevelContext>) (List<?>) getContextForSparqlResults(results);

    sparqlQuery = ContextQueryGenerator.generateQueryForConcurrentAtStartTimeFinishedLlc(llc, ont);

    results = executeSparqlQuery(sparqlQuery);

    llcList.addAll((List<LowLevelContext>) (List<?>) getContextForSparqlResults(results));

    return llcList;
}
```

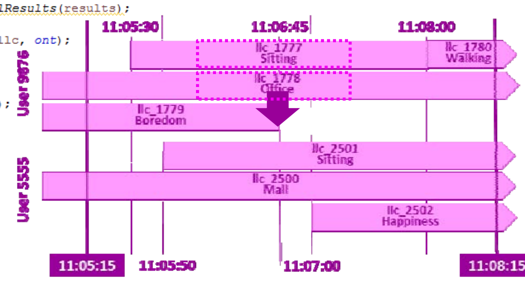


Context Ontology Manager

MM V2.0_ICL_T5_V0.6_151031_ClaudiaVillalonga

- **Context Handler:** Implementation of the method to retrieve from the Triplestore the LLC instances which are concurrent at the end time of a given LLC (to be used by the HLC Builder)

```java
public static List<LowLevelContext> retrieveConcurrentLlcAtEndTime(LowLevelContext llc) {

    Query sparqlQuery = ContextQueryGenerator.generateQueryForConcurrentAtEndTimeUnfinishedLlc(llc, ont);

    ResultSet results = executeSparqlQuery(sparqlQuery);

    List<LowLevelContext> llcList = (List<LowLevelContext>) (List<?>) getContextForSparqlResults(results);

    sparqlQuery = ContextQueryGenerator.generateQueryForConcurrentAtEndTimeFinishedLlc(llc, ont);

    results = executeSparqlQuery(sparqlQuery);

    llcList.addAll((List<LowLevelContext>) (List<?>) getContextForSparqlResults(results));

    return llcList;

}
```



Context Ontology Manager

MM V2.0_ICL_T5_V0.2_151003_ClaudiaVillalonga

- **Context Query Generator:** SPARQL queries for context retrieval
  - Definition of the SPARQL queries for context retrieval
  - Test of the SPARQL queries in Protégé

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX icl2: <http://www.miningminds.re.kr/icl/context/context-v2.owl#>

SELECT ?hlc ?starttime
        WHERE {
                ?hlc rdf:type icl2:HighLevelContext.
                ?hlc icl2:isContextOf icl2:user_9876.
                ?hlc icl2:hasStartTime ?starttime.

                FILTER NOT EXISTS { ?hlc icl2:hasEndTime ?endtime}
                FILTER (?starttime < "2015-08-10T11:08:45+09:00"^^xsd:dateTime)
        }
```

hlc	starttime
hlc_0003	"2015-08-10T11:08:00+09:00"^^<http://www.w3.org/2001/XMLSchema#dateTime>

High-Level Context Notifier

hlc_0010 rdf:type Context .
hlc_0010 rdf:type OfficeWork .
hlc_0010 hasActivity llc_1781 .
hlc_0010 hasLocation llc_1790 .
hlc_0010 hasEmotion llc_1775 .
hlc_0010 isContextOf user_9876 .
hlc_0010 hasStartTime “2015-08-10T11:08:45”^^dateTime .
hlc_0010 rdf:type hasActivity only ({llc_1781}) .
hlc_0010 rdf:type hasLocation only ({llc_1790}) .
hlc_0010 rdf:type hasEmotion only ({llc_1775}) .

Context Ontology Manager

MM V2.0_ICL_T5_V0.5_151024_ClaudiaVillalonga

- **Context Query Generator:** Automatic generation of SPARQL queries for HLC

1

```java
public static Query generateQueryForPreviousValidAndUnfinishedHlc(HighLevelContext newHlc, ContextOntology ont){

    ElementTriplesBlock block = createTriplesBlockForHlcUserStartTime(newHlc, ont);

    ElementFilter filterNeg = createFilterNotExistsEndTimeInHlc(ont);
    ElementFilter filterLessThan = createFilterStartTimeLessThan(newHlc.getDataPropertyValue(ont.getStartTimeProp()));

    return createHlcQuery(block, filterNeg, filterLessThan);

}
```

2

```java
public static Query generateQueryForPreviousValidAndFinishedHlc(HighLevelContext newHlc, ContextOntology ont){

    ElementTriplesBlock block = createTriplesBlockForHlcUserStartAndEndTime(newHlc, ont);

    Literal time = newHlc.getDataPropertyValue(ont.getStartTimeProp());
    ElementFilter filterLessThan = createFilterStartTimeLessThan(time);
    ElementFilter filterGreaterThan = createFilterEndTimeGreaterThan(time);

    return createHlcQuery(block, filterLessThan, filterGreaterThan);

}
```

3

```java
public static Query generateQueryForStartTimeOfNextHlc(HighLevelContext newHlc, ContextOntology ont){

    ElementTriplesBlock block = createTriplesBlockForHlcUserStartTime(newHlc, ont);

    Literal time = newHlc.getDataPropertyValue(ont.getStartTimeProp());
    ElementFilter filterGreaterThan = createFilterStartTimeGreaterThan(time);

    return createMinStartTimeQuery(block, filterGreaterThan);

}
```

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX icl2: <http://www.miningminds.re.kr/icl/context/context-v2.owl#>

SELECT ?hlc ?starttime
        WHERE {
                ?hlc rdf:type icl2:HighLevelContext.
                ?hlc icl2:isContextOf icl2:user_9876.
                ?hlc icl2:hasStartTime ?starttime.

                FILTER NOT EXISTS { ?hlc icl2:hasEndTime ?endtime}
                FILTER (?starttime < "2015-08-10T11:08:45+09:00"^^xsd:dateTime)
        }
```

hlc	starttime
hlc_0003	"2015-08-10T11:08:00+09:00"^^<http://www.w3.org/2001/XMLSchema

Context Ontology Manager

MM V2.0_ICL_T5_V0.6_151031_ClaudiaVillalonga

- **Context Query Generator:** Creation of the queries required by the Context Handler to retrieve LLC from the Triplestore.

1

```java
public static Query generateQueryForLlcWithStartTimeInWindow(Literal windowStart, Literal windowEnd,
        ContextOntology ont) {

    ElementTriplesBlock block = createTriplesBlockForLlcStartTime(ont);

    ElementFilter filterGreaterThanOrEqual = createFilterStartTimeGreaterThanOrEqual(windowStart);
    ElementFilter filterLessThan = createFilterStartTimeLessThan(windowEnd);

    Query q = createLlcQuery(block, filterGreaterThanOrEqual, filterLessThan);

    q.addOrderBy(new ExprVar(HLCA.startTimeSparqlVar), Query.ORDER_ASCENDING);

    return q;

}
```

2

```java
public static Query generateQueryForLlcWithEndTimeInWindowAndNoContiguousLlc(Literal windowStart, Literal windowEnd,
        ContextOntology ont) {

    ElementTriplesBlock block = createTriplesBlockForLlcUserAndEndTimeVar(ont);

    ElementFilter filterGreaterThanOrEqual = createFilterEndTimeGreaterThanOrEqual(windowStart);
    ElementFilter filterLessThan = createFilterEndTimeLessThan(windowEnd);
    ElementFilter filterNeg = createFilterNotExistsAdjacentLlc(ont);

    Query q = createLlcQuery(block, filterGreaterThanOrEqual, filterLessThan, filterNeg);

    q.addOrderBy(new ExprVar(HLCA.endTimeSparqlVar), Query.ORDER_ASCENDING);

    return q;

}
```

Context Ontology Manager

MM V2.0_ICL_T5_V0.6_151031_ClaudiaVillalonga

- **Context Query Generator:** Creation of the queries required by the Context Handler to retrieve LLC from the Triplestore.

3

```java
public static Query generateQueryForConcurrentAtStartTimeUnfinishedLlc(LowLevelContext llc, ContextOntology ont) {

    ElementTriplesBlock block = createTriplesBlockForLlcUserStartTime(
            llc.getObjectPropertyValue(ont.getContextOfProp()), ont);

    ElementFilter filterNeg = createFilterNotExistsEndTimeInLlc(ont);
    ElementFilter filterLessThanOrEqual = createFilterStartTimeLessThanOrEqual(
            llc.getDataPropertyValue(ont.getStartTimeProp()));
    ElementFilter filterNotEquals = createFilterLlcNotEquals(llc.getCtxIndiv());

    return createLlcQuery(block, filterNeg, filterLessThanOrEqual, filterNotEquals);

}
```

4

```java
public static Query generateQueryForConcurrentAtStartTimeFinishedLlc(LowLevelContext llc, ContextOntology ont) {

    ElementTriplesBlock block = createTriplesBlockForLlcUserStartAndEndTime(
            llc.getObjectPropertyValue(ont.getContextOfProp()), ont);

    Literal time = llc.getDataPropertyValue(ont.getStartTimeProp());
    ElementFilter filterLessThanOrEqual = createFilterStartTimeLessThanOrEqual(time);
    ElementFilter filterGreaterThan = createFilterEndTimeGreaterThan(time);
    ElementFilter filterNotEquals = createFilterLlcNotEquals(llc.getCtxIndiv());

    return createLlcQuery(block, filterLessThanOrEqual, filterGreaterThan, filterNotEquals);

}
```

Context Ontology Manager

MM V2.0_ICL_T5_V0.6_151031_ClaudiaVillalonga

- **Context Query Generator:** Creation of the queries required by the Context Handler to retrieve LLC from the Triplestore.

5

```
public static Query generateQueryForConcurrentAtEndTimeUnfinishedLlc(LowLevelContext llc, ContextOntology ont) {

    ElementTriplesBlock block = createTriplesBlockForLlcUserStartTime(
            llc.getObjectPropertyValue(ont.getContextOfProp()), ont);

    ElementFilter filterNeg = createFilterNotExistsEndTimeInLlc(ont);
    ElementFilter filterLessThanOrEqual = createFilterStartTimeLessThanOrEqual(
            llc.getDataPropertyValue(ont.getEndTimeProp()));

    return createLlcQuery(block, filterNeg, filterLessThanOrEqual);

}
```

6

```
public static Query generateQueryForConcurrentAtEndTimeFinishedLlc(LowLevelContext llc, ContextOntology ont) {

    ElementTriplesBlock block = createTriplesBlockForLlcUserStartAndEndTime(
            llc.getObjectPropertyValue(ont.getContextOfProp()), ont);

    Literal time = llc.getDataPropertyValue(ont.getEndTimeProp());
    ElementFilter filterLessThanOrEqual = createFilterStartTimeLessThanOrEqual(time);
    ElementFilter filterGreaterThan = createFilterEndTimeGreaterThan(time);

    return createLlcQuery(block, filterLessThanOrEqual, filterGreaterThan);

}
```

Context Ontology Manager

MM V2.0_ICL_T5_V0.5_151024_ClaudiaVillalonga

- **Context Ontology Manager:** Integration of its three subcomponents

```
public ContextOntologyManager(String directory, OntModel ontModel){

    this(directory);

    if(this.ont == null){ // OK. There is no Context Model stored in DB and this will be created

        OntModel infOntModel = ModelFactory.createOntologyModel( PelletReasonerFactory.THE_SPEC, ontModel);

        ContextOntology ont = new ContextOntology(ontModel);
        InferredContextOntology infOnt = new InferredContextOntology(infOntModel);

        if(infOnt.isValid()) {
            // Load ontology to DB
            OntologyModelManager.loadContextOntology(ont);

            this.ont = ont;
            this.infOnt = infOnt;

            ContextHandler.initialize(directory, this.ont);

        }

    }
    else
        // There is a Context Model stored in DB. Ignore the provided  ontology
        System.err.println("There is a Context Model stored in DB. Ignoring the provided  OntModel and using the available one to initialize

```

Knowledge Curation Layer (KCL 2.0)

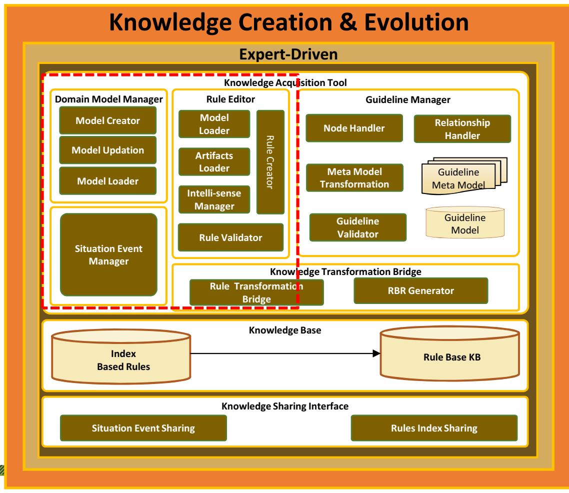
Principal Investigator:
Professor Sungyoung Lee

Expert-Driven Knowledge Acquisition

Taqdir Ali

Expert-Driven Knowledge Acquisition – Week 1 Progress

MM V2.0_KCL_T5_V0.1_20150925_Taqdir Ali



Component	Task	Responsible	19-25Sep	26Sep-2Oct	3-9Oct	10-16Oct
Configuration and environment setup	Learning Spring MVC, AngularJS, Installation, configuration	Taqdir Ali				
Implementation of Spring MVC	Implement a complete cycle of Spring MVC according to standard architecture	Taqdir Ali				
implementation of Integrated environment	Test implementation of Integrated work of Restful Service, AngularJS, and Spring MVC					
Schema Definition of the Knowledge Bases	Design and create Knowledge base schema in relational database	Taqdir				
Plain Rules Representation Mechanism	Design knowledge base for plain rules persistence					
Design User interfaces	Login screen Dashboard Rule Editor	Taqdir				
Restful web service	Development of Restful web service to use in Spring MVC	Taqdir				
library for using Hibernate	Design a library for using Hibernate	Taqdir				

Expert-Driven Knowledge Acquisition – Week 1 Progress

MM V2.0_KCL_T5_V0.1_20150925_Taqdir Ali

- Learning Development Technologies
  - Learn Spring MVC framework to develop flexible and loosely coupled web application
  - Learn AngularJS framework for client side scripting.
- Installation, configuration and Development
  - Install all required tools like eclipse, NetBeans
  - Configure Glass fish 4.1 and Apache Tomcat 8
  - Implement a complete cycle of Spring MVC according to standard architecture
  - Configure AngularJS with Spring MVC
  - Test implementation of Integrated work of Restful Service, AngularJS, and Spring MVC
- Challenges
  - HandlerMapping issue for calling the appropriate Controller by DispatcherServlet

Expert-Driven Knowledge Acquisition – Week 2 Future Plan

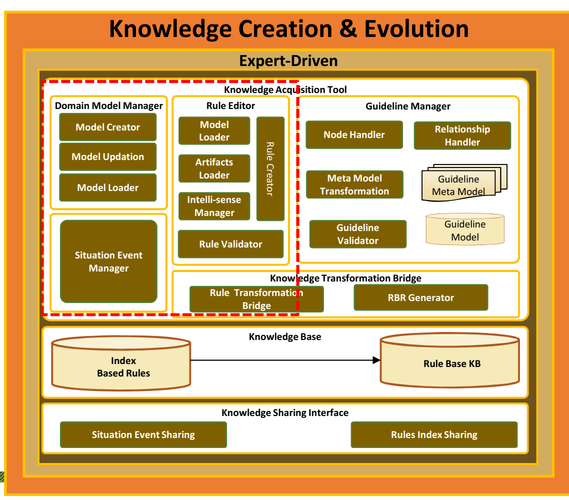
MM V2.0_KCL_T5_V0.1_20150925_Taqdir Ali

- Design and create Knowledge base schema in relational database
- Design knowledge base for plain rules persistence
- Design User interfaces
  - Login screen
  - Dashboard
  - Rule Editor
- Development of Restful web service to use in Spring MVC
- Design a library for using Hibernate

Component	Task	Responsible	19-25Sep	26Sep-2Oct	3-9Oct	10-16Oct
Configuration and environment setup	Learning Spring MVC, AngularJS, Installation, configuration	Taqdir Ali				
Implementation of Spring MVC	Implement a complete cycle of Spring MVC according to standard architecture	Taqdir Ali				
implementation of Integrated environment	Test implementation of Integrated work of Restful Service, AngularJS, and Spring MVC					
Schema Definition of the Knowledge Bases	Design and create Knowledge base schema in relational database	Taqdir				
Plain Rules Representation Mechanism	Design knowledge base for plain rules persistence					
Design User interfaces	Login screen Dashboard Rule Editor	Taqdir				
Restful web service	Development of Restful web service to use in Spring MVC	Taqdir				
library for using Hibernate	Design a library for using Hibernate	Taqdir				

Expert-Driven Knowledge Acquisition – Week 2 Progress

MM V2.0_KCL_T5_V0.2_20151003_Taqdir Ali

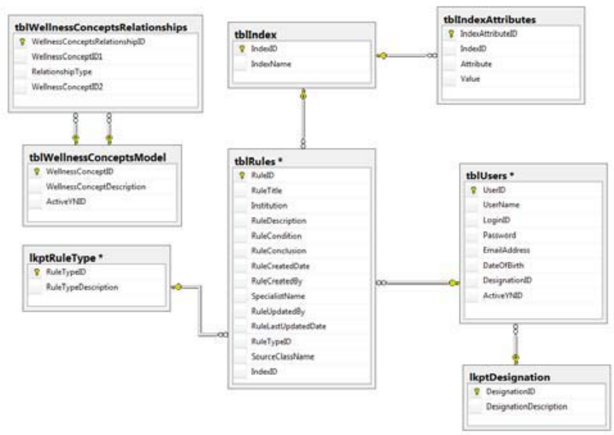


Component	Task	Responsible	19-25Sep	26Sep-2Oct	3-9Oct	10-16Oct
Configuration and environment setup	Learning Spring MVC, AngularJS, Installation, configuration	Taqdir Ali				
Implementation of Spring MVC	Implement a complete cycle of Spring MVC according to standard architecture	Taqdir Ali				
implementation of Integrated environment	Test implementation of Integrated work of Restful Service, AngularJS, and Spring MVC					
Schema Definition of the Knowledge Bases	Design and create Knowledge base schema in relational database	Taqdir				
Plain Rules Representation Mechanism	Design knowledge base for plain rules persistence					
Design User interfaces	Login screen Dashboard Rule Editor	Taqdir				
Restful web service	Development of Restful web service to use in Spring MVC	Taqdir				
library for using Hibernate	Design a library for using Hibernate	Taqdir				

Expert-Driven Knowledge Acquisition – Week 2 Progress

MM V2.0_KCL_T5_V0.2_20151003_Taqdir Ali

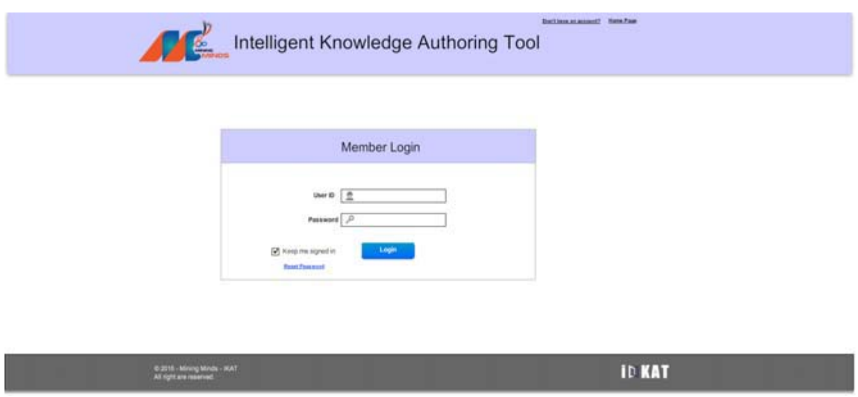
- Design and create database schema for knowledge base (Completed)
- Plain rule representation (In progress)
  - Storage in database
  - Storage in text files in folder hierarchy
- Restful Web services (In progress)
  - Implementation of services
  - Integration with data access library
- Design User Interfaces (Completed)
  - Login screen
  - Dashboard
  - Rule Editor
- Library Implementation using Hibernate (In progress)



Expert-Driven Knowledge Acquisition – Week 2 Progress

MM V2.0_KCL_T5_V0.2_20151003_Taqdir Ali

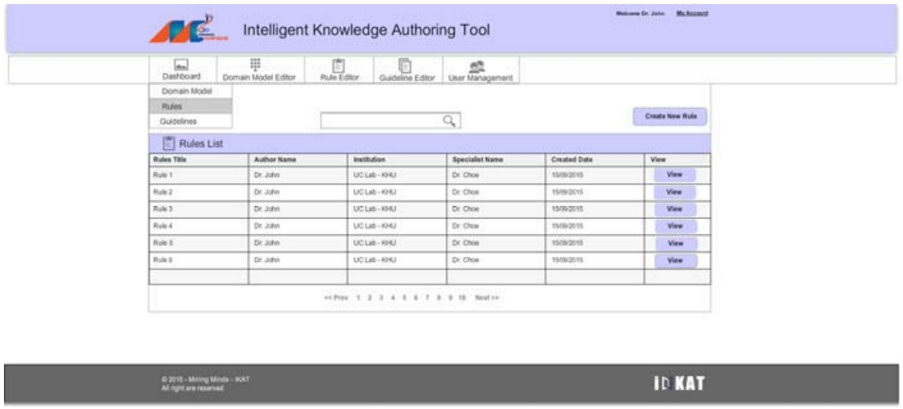
- Mockup for Login



Expert-Driven Knowledge Acquisition – Week 2 Progress

MM V2.0_KCL_T5_V0.2_20151003_Taqdir Ali

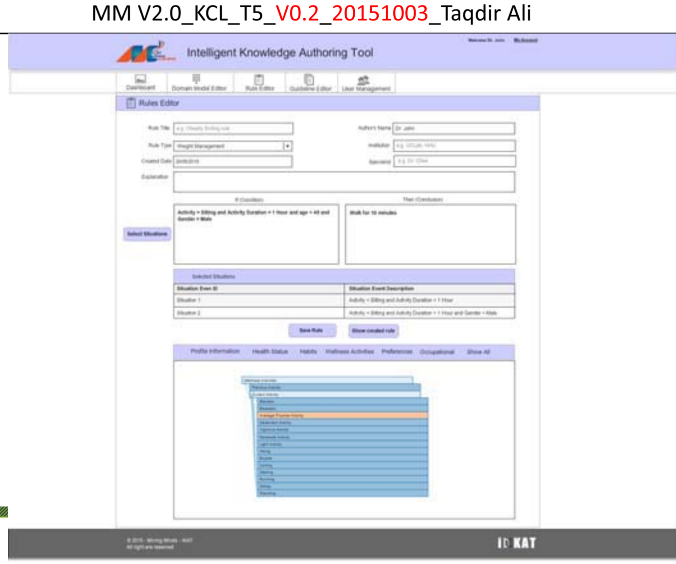
- Mockup for Dashboard



Expert-Driven Knowledge Acquisition – Week 2 Progress

MM V2.0_KCL_T5_V0.2_20151003_Taqdir Ali

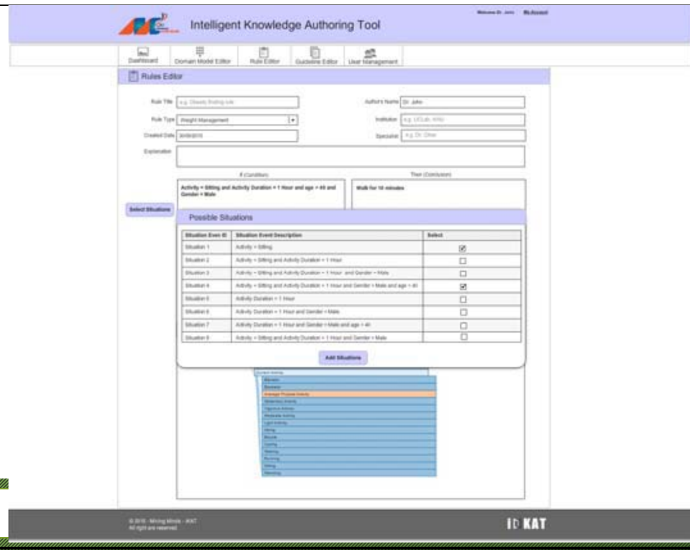
- Mockup for Rule Editor



Expert-Driven Knowledge Acquisition – Week 2 Progress

MM V2.0_KCL_T5_V0.2_20151003_Taqdir Ali

- Mockup for Rule Editor with situation



Expert-Driven Knowledge Acquisition – Week 3 Future Plan

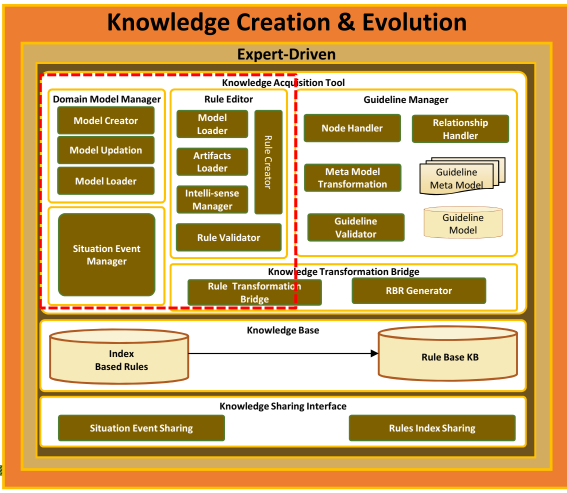
MM V2.0_KCL_T5_V0.3_20151003_Taqdir Ali

- Future Tasks
  - Continuation of Plain rule representation
    - Storage in database
  - Continuation of Restful Web services
    - Implementation of services
    - Integration with data access library
  - Continuation of library implementation using Hibernate
  - HTML creation and logic implementation
    - Login Screen
    - Dashboard
    - Rule Editor with Situation Handling

Component	Task	Responsible	19-25Sep	26Sep-2Oct	3-9Oct	10-16Oct
Configuration and environment setup	Learning Spring MVC, AngularJS, Installation, configuration	Taqdir Ali				
Implementation of Spring MVC	Implement a complete cycle of Spring MVC according to standard architecture	Taqdir Ali				
implementation of Integrated envirnoment	Test implementation of Integrated work of Restful Service, AngularJS, and Spring MVC					
Schema Definition of the Knowledge Bases	Design and create Knowledge base schema in relational database	Taqdir				
Plain Rules Representation Mechanism	Design knowledge base for plain rules persistence					
Design User interfaces	Login screen Dashboard Rule Editor	Taqdir				
Restful web service	Development of Restful web service to use in Spring MVC	Taqdir				
library for using Hibernate	Design a library for using Hibernate	Taqdir				
HTML Creation and logic implementation	Login screen Dashboard Rule Editor	Taqdir/Maqbool Hussain				

Expert-Driven Knowledge Acquisition – Week 3 Progress

MM V2.0_KCL_T5_V0.4_20151008_Taqdir Ali



Component	*Task*	*Responsible*	19-25Sep	26Sep-2Oct	3-9Oct	10-16Oct
Configuration and environment setup	Learning Spring MVC, AngularJS, Installation, configuration	Taqdir Ali				
Implementation of Spring MVC	Implement a complete cycle of Spring MVC according to standard architecture	Taqdir Ali				
implementation of Integrated environment	Test implementation of Integrated work of Restful Service, AngularJS, and Spring MVC					
Schema Definition of the Knowledge Bases	Design and create Knowledge base schema in relational database	Taqdir				
Plain Rules Representation Mechanism	Design knowledge base for plain rules persistence					
Design User interfaces	Login screen Dashboard Rule Editor	Taqdir				
Restful web service	Development of Restful web service to use in Spring MVC	Taqdir				
library for using Hibernate	Design a library for using Hibernate	Taqdir				
HTML Creation and logic implementation	Login screen Dashboard Rule Editor	Taqdir/Maqbool Hussain				

Expert-Driven Knowledge Acquisition – Week 3 Progress

MM V2.0_KCL_T5_V0.5_20151009_Taqdir Ali

- Meeting with Professor Kang (Recommended some changes)
- Continuation of Plain rule representation (Completed → In Progress)
  - Storage in database
- Continuation of Restful Web services (Shifted to local services due to Spring MVC)
  - Implementation of local services
  - Integration with data access library (No need)
- Continuation of library implementation using Hibernate (In progress)
- HTML creation and logic implementation (Re-design)
  - Login Screen
  - Dashboard
  - Rule Editor with Situation Handling

Expert-Driven Knowledge Acquisition – Week 4 Future Plan

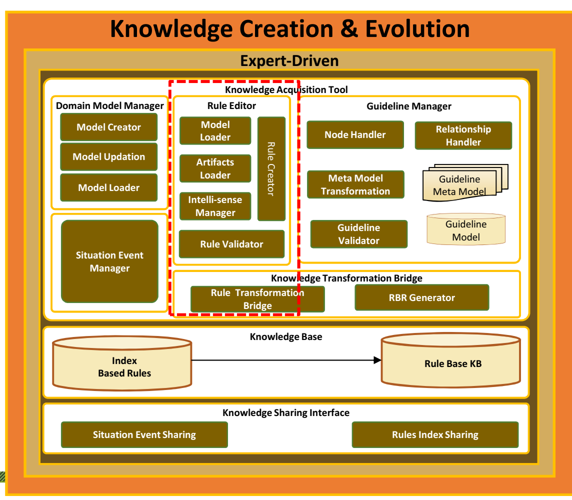
MM V2.0_KCL_T5_V0.5_20151009_Taqdir Ali

- Future Tasks
  - Plain rule representation and Schema changes
    - Storage in database
  - Re-design user interfaces
    - Rule Editor
  - Local Services for Spring MVC
  - Library for Hibernate
    - Object Model implementation
    - Data Access layer implementation
  - HTML creation and logic implementation
    - Login Screen
    - Dashboard
    - Rule Editor with Situation Handling

Component	*Task*	*Responsible*	10 – 16 Oct	17 - 23 Oct
Plain Rules Representation Mechanism	Re-design knowledge base for plain rules persistence	Taqdir		
Re-Design User interfaces	Login screen (No Need) Dashboard (No Need) Rule Editor	Taqdir		
Local services for spring MVC	Development of local service to use in Spring MVC	Taqdir		
library for using Hibernate	Object Model and Data Access Layer using Hibernate	Taqdir		
HTML Creation and logic implementation	Login screen Dashboard Rule Editor	Taqdir/Maqbool Hussain		

Expert-Driven Knowledge Acquisition – Week 4 Progress

MM V2.0_KCL_T5_V0.6_20151014_Taqdir Ali

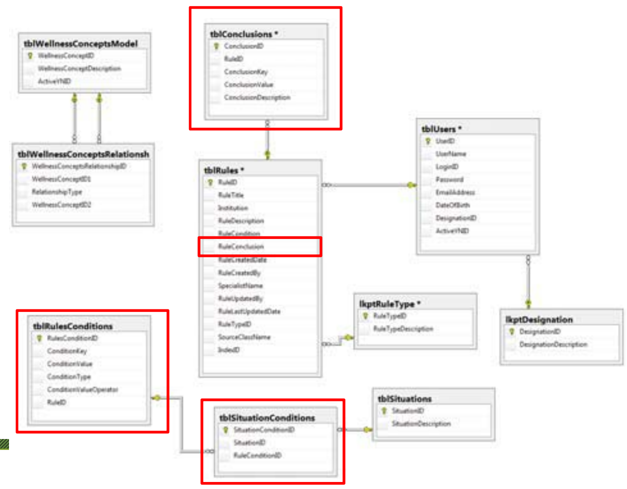


Component	*Task*	*Responsible*	10 – 16 Oct	17 - 23 Oct
Plain Rules Representation Mechanism	Re-design knowledge base for plain rules persistence	Taqdir		
Re-Design User interfaces	Login screen (No Need) Dashboard (No Need) Rule Editor	Taqdir		
Local services for spring MVC	Development of local service to use in Spring MVC	Taqdir		
library for using Hibernate	Object Model and Data Access Layer using Hibernate	Taqdir		
HTML Creation and logic implementation	Login screen Dashboard Rule Editor	Taqdir/Maqbool Hussain		

Expert-Driven Knowledge Acquisition – Week 4 Progress

MM V2.0_KCL_T5_V0.6_20151014_Taqdir Ali

- Table for conclusions, single rules may have multiple conclusions
  - One to many relationship
- Functionality of a single recommendation in text format
- SituationFacts → SituationCondition
- RulesFacts → RulesConditions



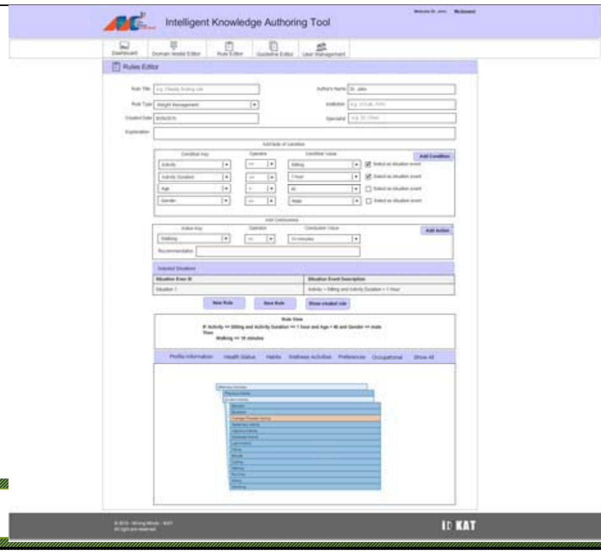
Expert-Driven Knowledge Acquisition – Week 4 Progress

MM V2.0_KCL_T5_V0.6_20151014_Taqdir Ali

<< As Is >>   << To Be >>

Expert-Driven Knowledge Acquisition – Week 4 Progress

MM V2.0_KCL_T5_V0.6_20151014_Taqdir Ali



Expert-Driven Knowledge Acquisition – Week 5 Future Plan

MM V2.0_KCL_T5_V0.7_20151023_Taqdir Ali

- Future Tasks
  - Local Services for Spring MVC (Continue)
  - Library for Hibernate (Continue)
    - Object Model implementation
    - Data Access layer implementation
  - HTML creation and logic implementation (Continue)
    - Login Screen
    - Dashboard
    - Rule Editor with Situation Handling

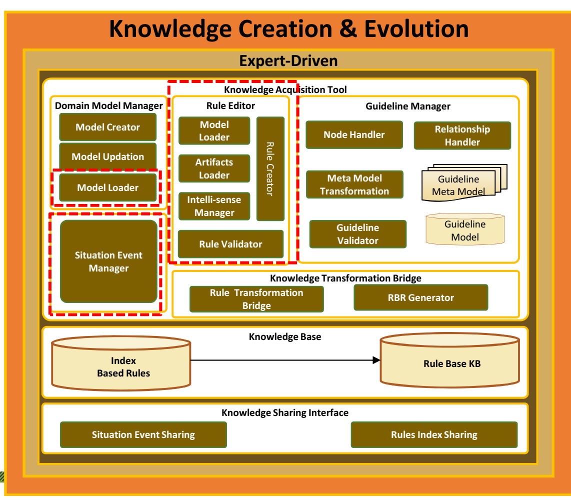
Component	*Task*	*Responsible*	10 – 16 Oct	17 - 23 Oct
Plain Rules Representation Mechanism	Re-design knowledge base for plain rules persistence	Taqdir		
Re-Design User interfaces	Login screen (No Need) Dashboard (No Need) Rule Editor	Taqdir		
Local services for spring MVC	Development of local service to use in Spring MVC	Taqdir		
library for using Hibernate	Object Model and Data Access Layer using Hibernate	Taqdir		
HTML Creation and logic implementation	Login screen Dashboard Rule Editor	Taqdir/Maqbool Hussain		

Recap of tasks of last 5 weeks

- Week 1
  - Learning Development Technologies (Completed)
  - Installation, configuration and Development (Completed)
  - Configure AngularJS with Spring MVC (Completed)
  - Test implementation of Integrated work of Restful Service, AngularJS, and Spring MVC (Completed)
- Week 2
  - Design and create database schema for knowledge base (Completed)
  - Plain rule representation (In progress)
  - Restful Web services (In progress)
  - Design User Interfaces (Completed)
  - Library Implementation using Hibernate
- Week 3
  - Meeting with Professor Kang (Recommended some changes)
  - Continuation of Plain rule representation (Completed In Progress)
  - Continuation of Restful Web services (Shifted to local services due to Spring MVC)
  - Continuation of library implementation using Hibernate (In progress)
  - HTML creation and logic implementation (Re-design)
- Week 4
  - Plain rule representation and Schema changes
  - Re-design user interfaces
  - Local Services for Spring MVC
  - Library for Hibernate
  - HTML creation and logic implementation
- Week 5
  - Local Services for Spring MVC (Continue)
  - Library for Hibernate (Continue)
  - HTML creation and logic implementation (Continue)

Expert-Driven Knowledge Acquisition – Week 5 Progress

MM V2.0_KCL_T5_V0.7_20151023_Taqdir Ali



Component	Task	Responsible	10 – 16 Oct	17 - 23 Oct
Plain Rules Representation Mechanism	Re-design knowledge base for plain rules persistence	Taqdir		
Re-Design User interfaces	Login screen (No Need) Dashboard (No Need) Rule Editor	Taqdir		
Local services for spring MVC	Development of local service to use in Spring MVC	Taqdir		
library for using Hibernate	Object Model and Data Access Layer using Hibernate	Taqdir		
HTML Creation and logic implementation	Login screen Dashboard Rule Editor	Taqdir/Maqbool Hussain		

Expert-Driven Knowledge Acquisition – Week 6 Future Plan

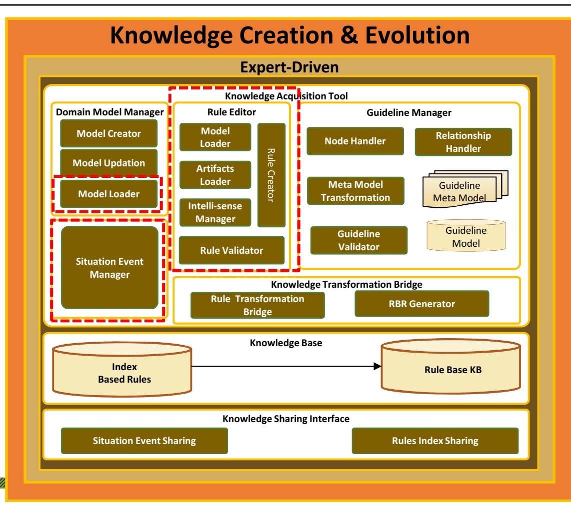
MM V2.0_KCL_T5_V0.7_20151023_Taqdir Ali

- Integration of Spring MVC and Hibernate
  - Integration of data access layer using Hibernate with Spring MVC infrastructure
- Data Access Interfaces
  - Rules DAO Interfaces
    - Conditions DAO Interface
    - Conclusion DAO Interface
  - Situation DAO Interface
    - Situation Condition Interface
  - Wellness DAO Interface
  - User (Expert) DAO Interface
- Logic Implementation (Continuation)
  - Login
  - Dashboard
  - Rule Editor

Component	*Task*	*Responsible*	24 – 30 Oct
Integration of Spring MVC and Hibernate	Integration of data access layer using Hibernate with Spring MVC infrastructure	Taqdir	
Data access interfaces for Rules	Using Hibernate annotations to develop DAO interface for Rule, Conditions, and Conclusion	Taqdir	
Data access interfaces for Situations	Using Hibernate annotations to develop DAO interface for Situation and Situation Condition	Taqdir	
Data access interfaces for Wellness Model and Users	Using Hibernate annotations to develop DAO interface for Wellness Model retrieval and Authentication of users.	Taqdir	
logic implementation	Dashboard Login screen Rule Editor	Taqdir	

Expert-Driven Knowledge Acquisition – Week 6 Progress

MM V2.0_KCL_T5_V0.7_20151030_Taqdir Ali



Component	*Task*	*Responsible*	24 – 30 Oct
Integration of Spring MVC and Hibernate	Integration of data access layer using Hibernate with Spring MVC infrastructure	Taqdir	
Data access interfaces for Rules	Using Hibernate annotations to develop DAO interface for Rule, Conditions, and Conclusion	Taqdir	
Data access interfaces for Situations	Using Hibernate annotations to develop DAO interface for Situation and Situation Condition	Taqdir	
Data access interfaces for Wellness Model and Users	Using Hibernate annotations to develop DAO interface for Wellness Model retrieval and Authentication of users.	Taqdir	
logic implementation	Dashboard Login screen Rule Editor	Taqdir	

Expert-Driven Knowledge Acquisition – Week 7 Future Plan

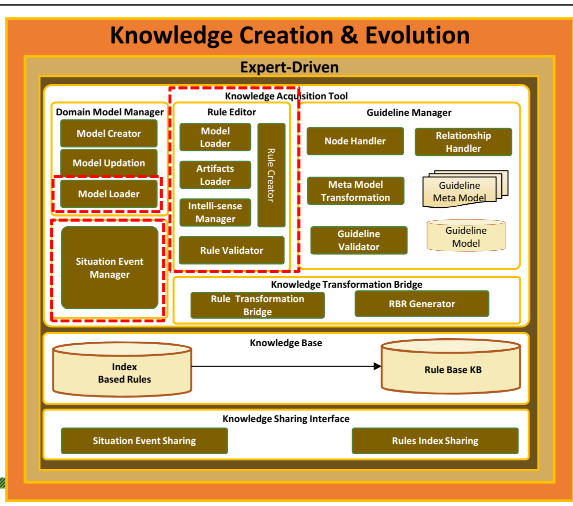
MM V2.0_KCL_T5_V0.7_20151031_Taqdir Ali

- Logic Implementation for Login, dashboard and rule editor
  - Dashboard
  - Login screen
  - Rule Editor
- DAO Implementation for Rules
  - Using Hibernate annotations to develop DAO Implementation for Rule
  - Condition
  - Conclusion
- DAO Implementation for Situations
  - Using Hibernate annotations to develop DAO Implementation for Situation and Situation Condition
- DAO Implementation for Wellness Model and Users
  - Using Hibernate annotations to develop DAO Implementation for Wellness Model retrieval and Authentication of users.
- Local Service Interfaces Rule, Condition and Conclusion
- Local Service Interfaces Situation, user and wellness model

Component	*Task*	*Responsible*	31 Oct – 06 Nov
Logic Implementation for Login, dashboard and rule editor	Dashboard Login screen Rule Editor	Taqdir	
DAO Implementation for Rules	Using Hibernate annotations to develop DAO Implementation for Rule, Conditions, and Conclusion	Taqdir	
DAO Implementation for Situations	Using Hibernate annotations to develop DAO Implementation for Situation and Situation Condition	Taqdir	
DAO Implementation for Wellness Model and Users	Using Hibernate annotations to develop DAO Implementation for Wellness Model retrieval and Authentication of users.	Taqdir	
Local Service Interfaces Rule, Condition and Conclusion	Using Spring MVC infrastructure to develop spring local service interfaces for Rules, Conditions and Conclusion	Taqdir	
Local Service Interfaces Situation, user and wellness model	Using Spring MVC infrastructure to develop spring local service interfaces for Situations, user and wellness model	Taqdir	

Expert-Driven Knowledge Acquisition – Week 7 Progress

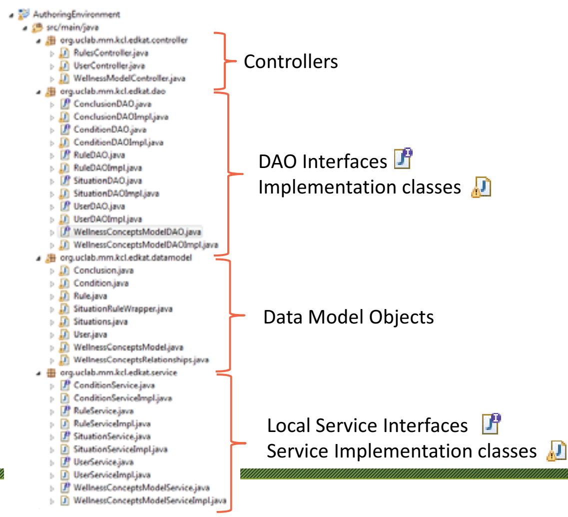
MM V2.0_KCL_T5_V0.7_20151105_Taqdir Ali

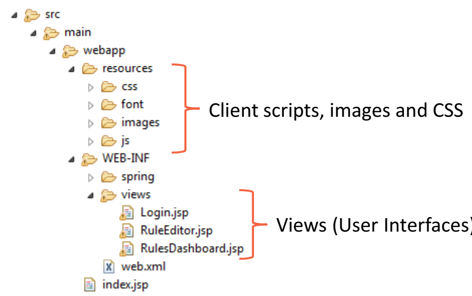


Component	*Task*	*Responsible*	31 Oct – 06 Nov
Logic Implementation for Login, dashboard and rule editor	Dashboard Login screen Rule Editor	Taqdir	
DAO Implementation for Rules	Using Hibernate annotations to develop DAO Implementation for Rule, Conditions, and Conclusion	Taqdir	
DAO Implementation for Situations	Using Hibernate annotations to develop DAO Implementation for Situation and Situation Condition	Taqdir	
DAO Implementation for Wellness Model and Users	Using Hibernate annotations to develop DAO Implementation for Wellness Model retrieval and Authentication of users.	Taqdir	
Local Service Interfaces Rule, Condition and Conclusion	Using Spring MVC infrastructure to develop spring local service interfaces for Rules, Conditions and Conclusion	Taqdir	
Local Service Interfaces Situation, user and wellness model	Using Spring MVC infrastructure to develop spring local service interfaces for Situations, user and wellness model	Taqdir	

Expert-Driven Knowledge Acquisition – Week 7 Progress

MM V2.0_KCL_T5_V0.7_20151105_Taqdir Ali





Expert-Driven Knowledge Acquisition – Week 8 Future Plan

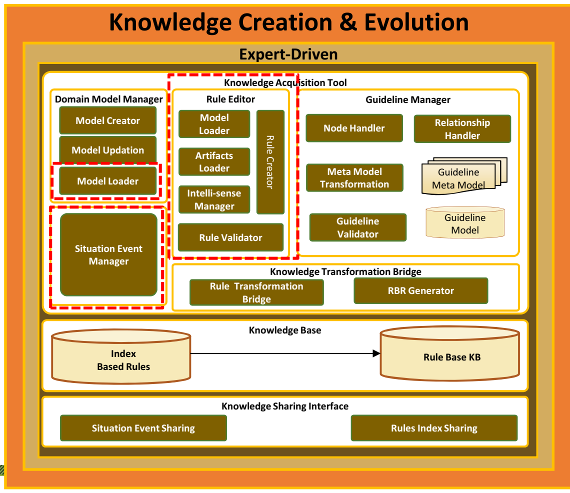
MM V2.0_KCL_T5_V0.8_20151107_Taqdir Ali

- Local Service Implementation for Rules, Conditions, and Conclusions
  - Using Spring MVC infrastructure to develop service Implementation for Rule, Conditions, and Conclusion
- Local Service Implementation for Situations, Users and Wellness Models
  - Using Spring MVC infrastructure to develop service Implementation for Situations, Users, and Wellness Models
- Providing Intelli-sense from Wellness model
  - Using jQuery and wellness services to provide Intelli-sense to text boxes of conditions and conclusions
- HTML final lookup.
  - Final look up to HTML CSS for the screens of Login, Dashboard and Rule Editor.
- Testing of Creating, Updating rules, Situations
  - Testing of different scenario for adding rules, conditions, conclusions and situations

Component	*Task*	*Responsible*	07 – 13 Nov
Local Service Implementation for Rules, Conditions, and Conclusions	Using Spring MVC infrastructure to develop service Implementation for Rule, Conditions, and Conclusion	Taqdir	
Local Service Implementation for Situations, Users and Wellness Models	Using Spring MVC infrastructure to develop service Implementation for Situations, Users, and Wellness Models	Taqdir	
Providing Intelli-sense from Wellness model	Using jQuery and wellness services to provide Intelli-sense to text boxes of conditions and conclusions	Taqdir	
HTML final lookup	Final look up to HTML CSS for the screens of Login, Dashboard and Rule Editor.	Taqdir/Jamil	
Testing of Creating, Updating rules, Situations	Testing of different scenario for adding rules, conditions, conclusions and situations	Taqdir	

Expert-Driven Knowledge Acquisition – Week 8 Progress

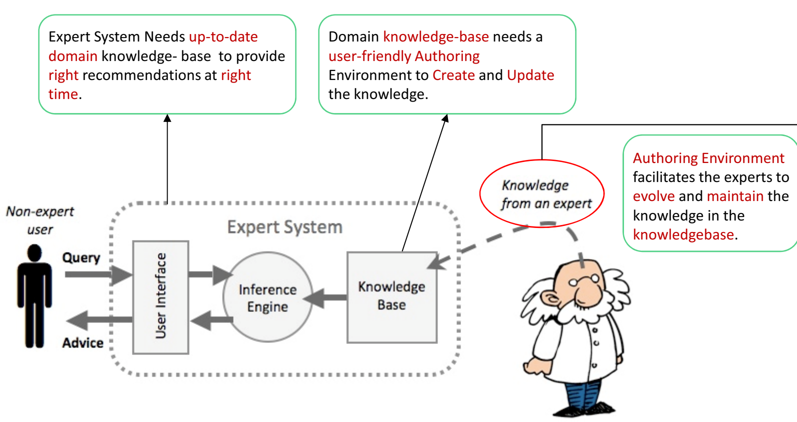
MM V2.0_KCL_T5_V0.8_20151110_Taqdir Ali



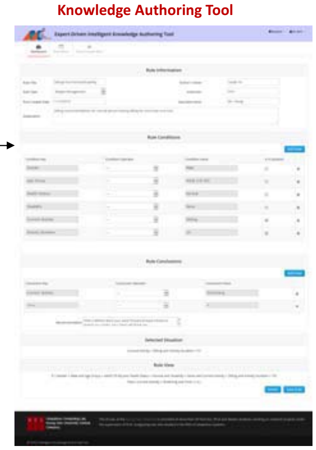
Component	*Task*	*Responsible*	07 – 13 Nov
Local Service Implementation for Rules, Conditions, and Conclusions	Using Spring MVC infrastructure to develop service Implementation for Rule, Conditions, and Conclusion	Taqdir	
Local Service Implementation for Situations, Users and Wellness Models	Using Spring MVC infrastructure to develop service Implementation for Situations, Users, and Wellness Models	Taqdir	
Providing Intelli-sense from Wellness model	Using jQuery and wellness services to provide Intelli-sense to text boxes of conditions and conclusions	Taqdir	
HTML final lookup	Final look up to HTML CSS for the screens of Login, Dashboard and Rule Editor.	Taqdir/Jamil	
Testing of Creating, Updating rules, Situations	Testing of different scenario for adding rules, conditions, conclusions and situations	Taqdir	

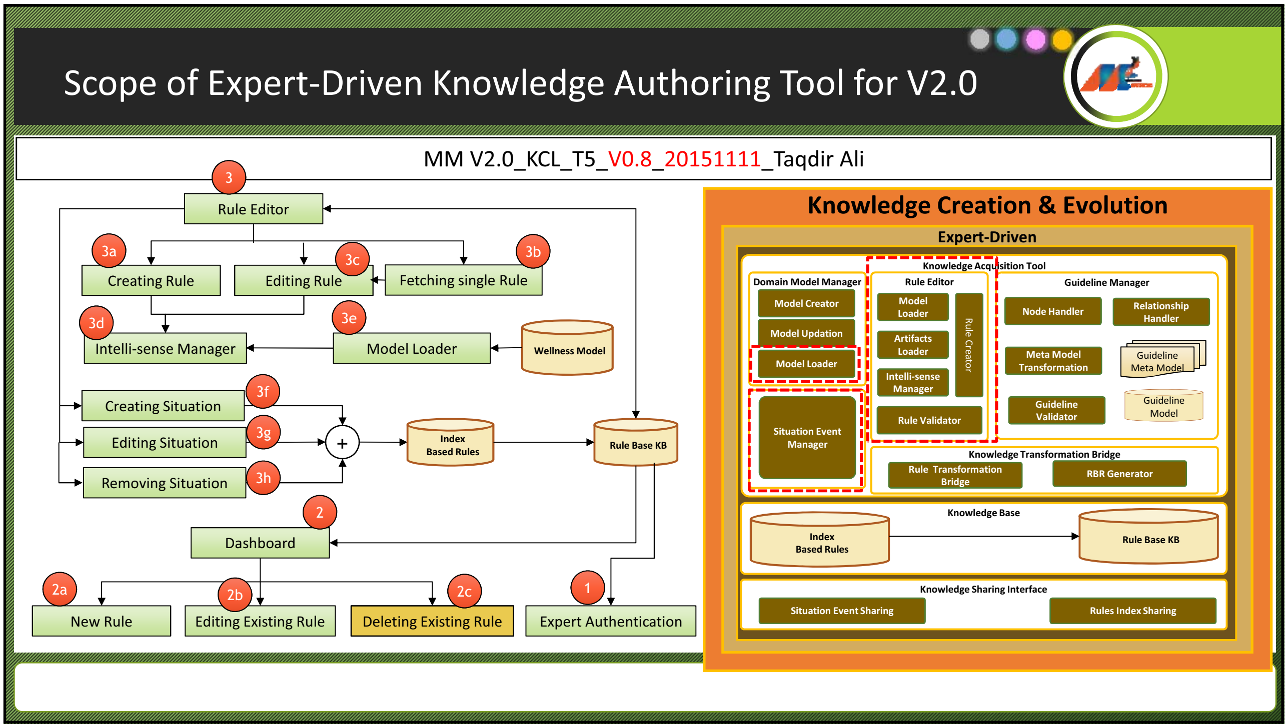
Purpose of Expert-Driven Knowledge Authoring Tool

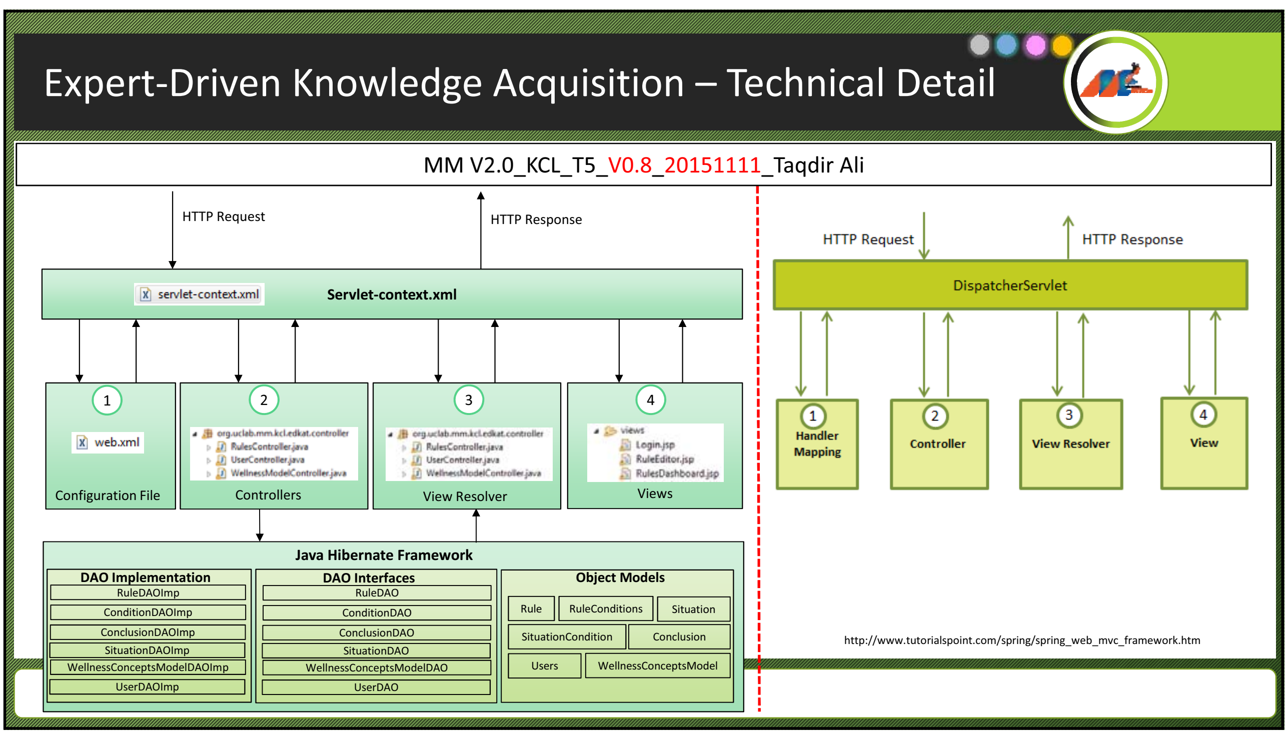
MM V2.0_KCL_T5_V0.8_20151111_Taqdir Ali

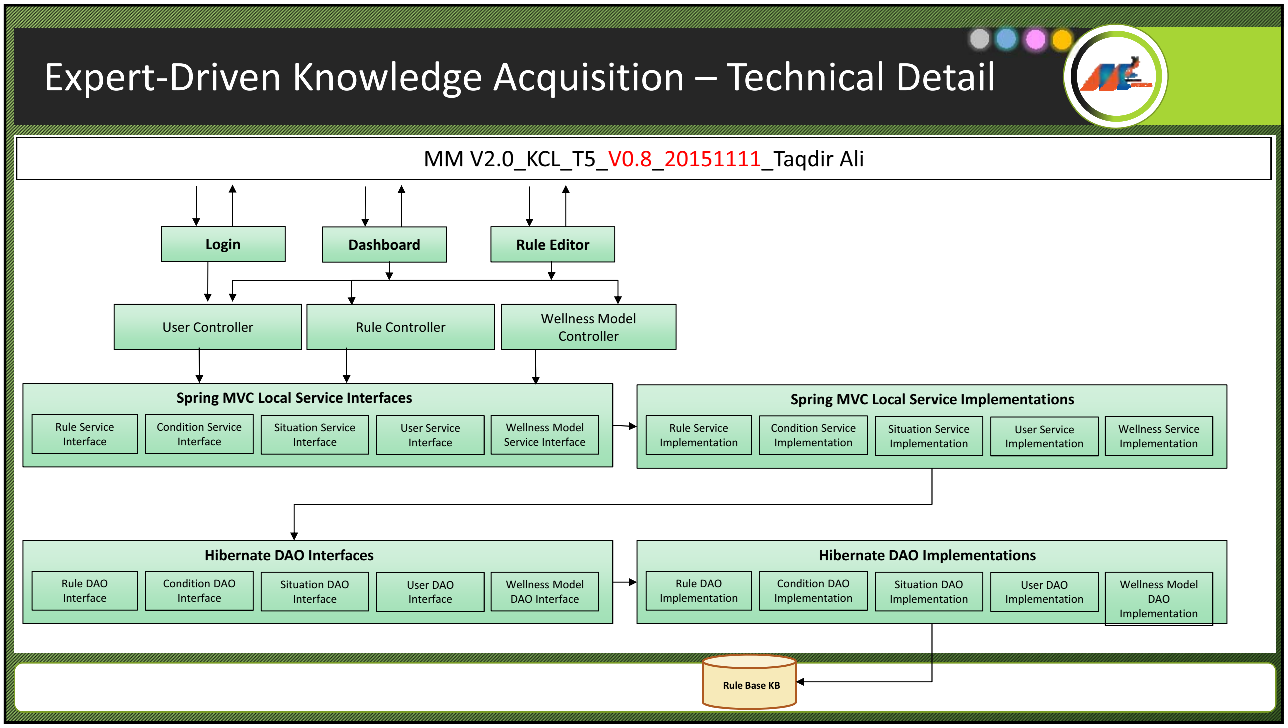


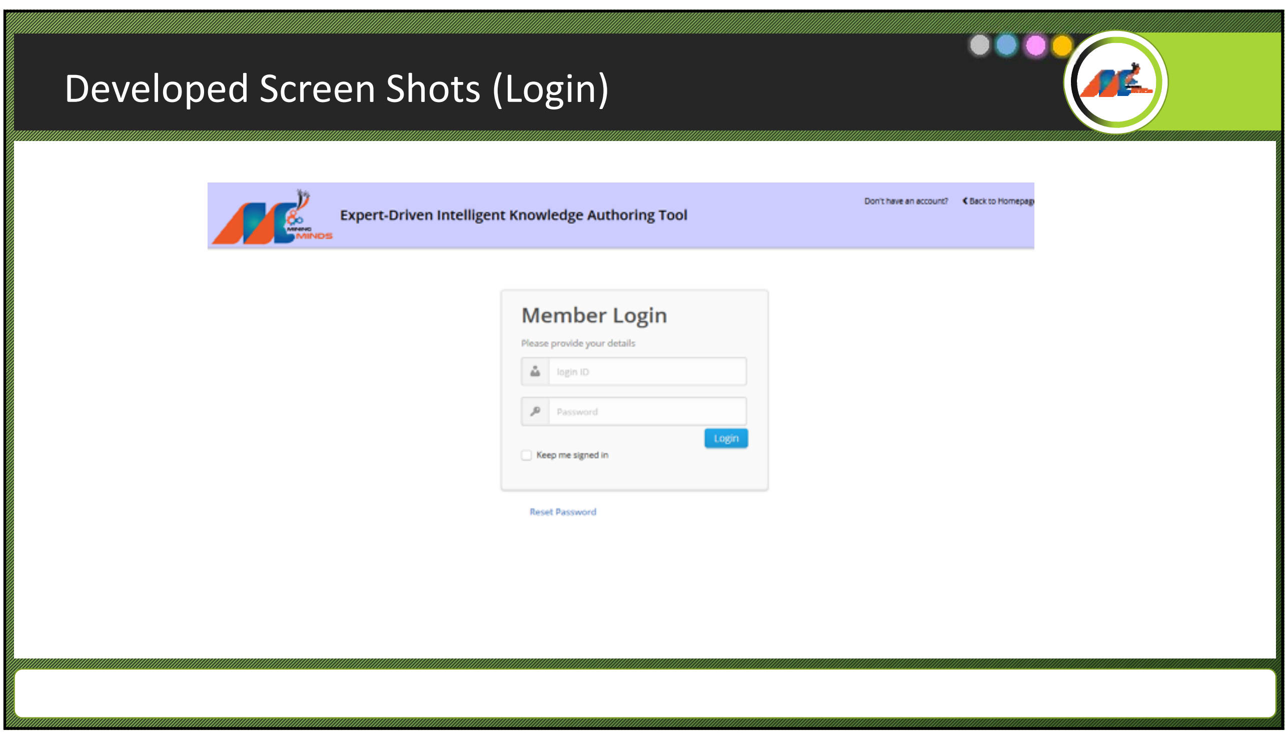
Knowledge Authoring Tool



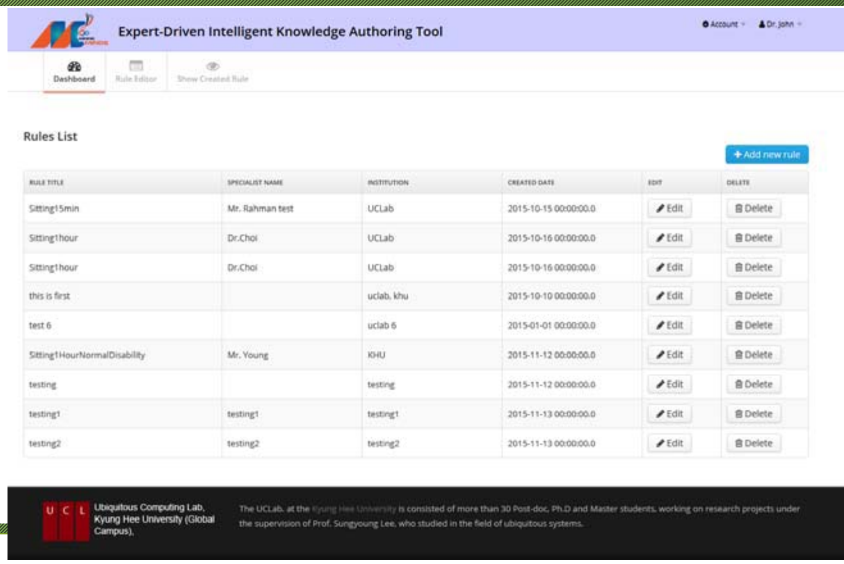

Scope of Expert-Driven Knowledge Authoring Tool for V2.0
MM V2.0_KCL_T5_V0.8_20151111_Taqdir Ali
3
Rule Editor
3a
Creating Rule
Editing Rule
3c
Fetching single Rule
3b
3d
Intelli-sense Manager
3e
Model Loader
Wellness Model
Creating Situation
3f
Editing Situation
3g
Removing Situation
3h
Index Based Rules
Rule Base KB
2
Dashboard
2a
New Rule
2b
Editing Existing Rule
2c
Deleting Existing Rule
1
Expert Authentication
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 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 – Technical Detail
MM V2.0_KCL_T5_V0.8_20151111_Taqdir Ali
HTTP Request
HTTP Response
servlet-context.xml
Servlet-context.xml
1
web.xml
Configuration File
2
org.uclab.mm.kcl.edkat.controller
RulesController.java
UserController.java
WellnessModelController.java
Controllers
3
org.uclab.mm.kcl.edkat.controller
RulesController.java
UserController.java
WellnessModelController.java
View Resolver
4
views
Login.jsp
RuleEditor.jsp
RulesDashboard.jsp
Views
Java Hibernate Framework
DAO Implementation
RuleDAOImp
ConditionDAOImp
ConclusionDAOImp
SituationDAOImp
WellnessConceptsModelDAOImp
UserDAOImp
DAO Interfaces
RuleDAO
ConditionDAO
ConclusionDAO
SituationDAO
WellnessConceptsModelDAO
UserDAO
Object Models
Rule
RuleConditions
Situation
SituationCondition
Conclusion
Users
WellnessConceptsModel
HTTP Request
HTTP Response
DispatcherServlet
1
Handler Mapping
2
Controller
3
View Resolver
4
View
http://www.tutorialspoint.com/spring/spring_web_mvc_framework.htm


Expert-Driven Knowledge Acquisition – Technical Detail
MM V2.0_KCL_T5_V0.8_20151111_Taqdir Ali
Login
Dashboard
Rule Editor
User Controller
Rule Controller
Wellness Model Controller
Spring MVC Local Service Interfaces
Rule Service Interface
Condition Service Interface
Situation Service Interface
User Service Interface
Wellness Model Service Interface
Spring MVC Local Service Implementations
Rule Service Implementation
Condition Service Implementation
Situation Service Implementation
User Service Implementation
Wellness Service Implementation
Hibernate DAO Interfaces
Rule DAO Interface
Condition DAO Interface
Situation DAO Interface
User DAO Interface
Wellness Model DAO Interface
Hibernate DAO Implementations
Rule DAO Implementation
Condition DAO Implementation
Situation DAO Implementation
User DAO Implementation
Wellness Model DAO Implementation
Rule Base KB


Developed Screen Shots (Login)
Expert-Driven Intelligent Knowledge Authoring Tool
Don't have an account?
Back to Homepag
Member Login
Please provide your details
login ID
Password
Login
Keep me signed in
Reset Password

Developed Screen Shots (Dashboard)



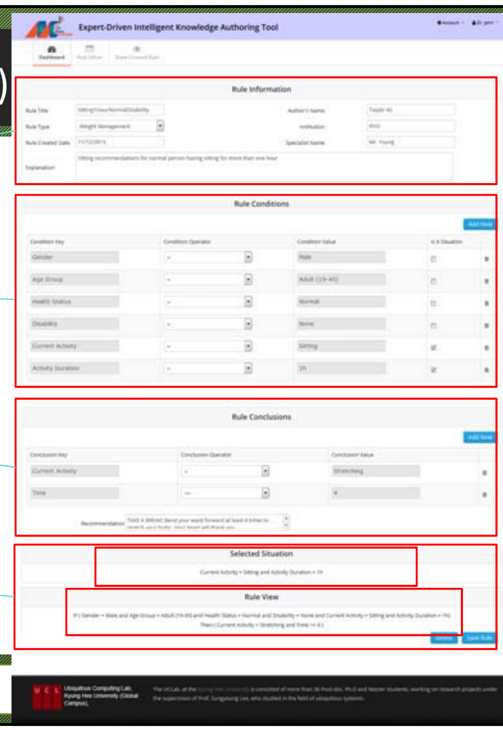
Developed Screen Shots (Rule Editor)

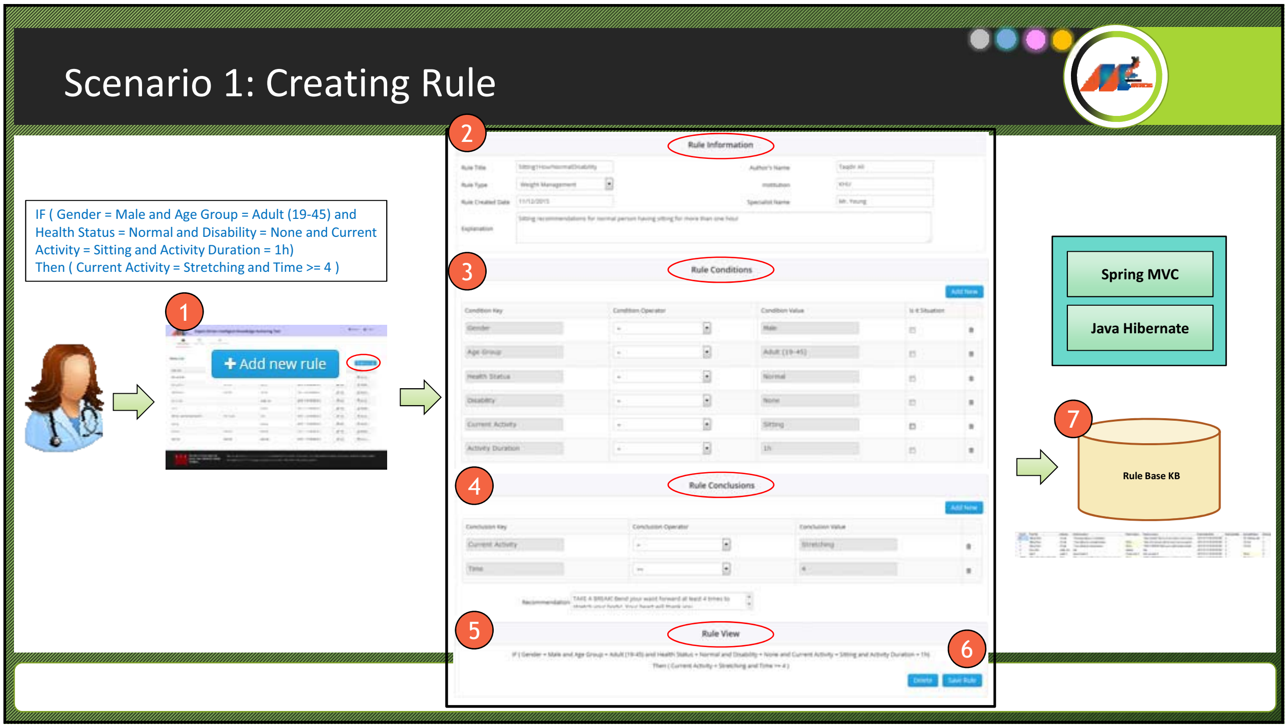
Rule Meta Information

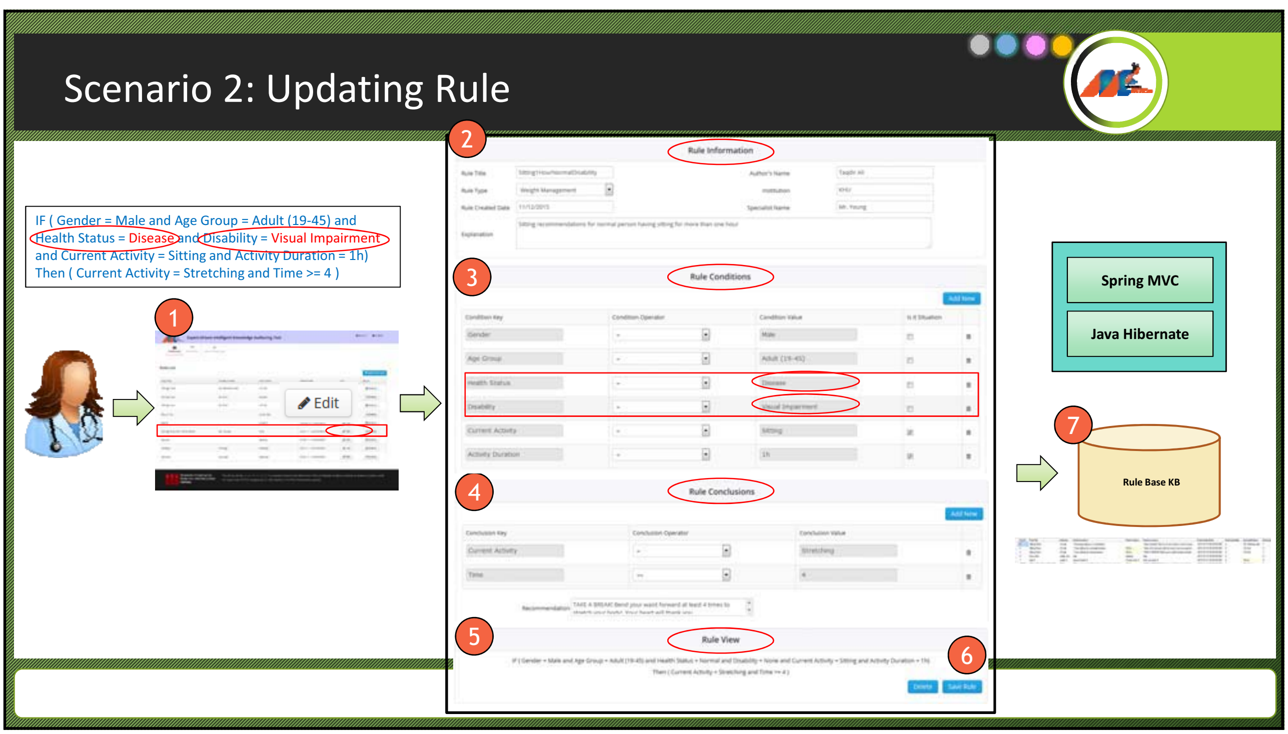
Multiple conditions in "IF" part

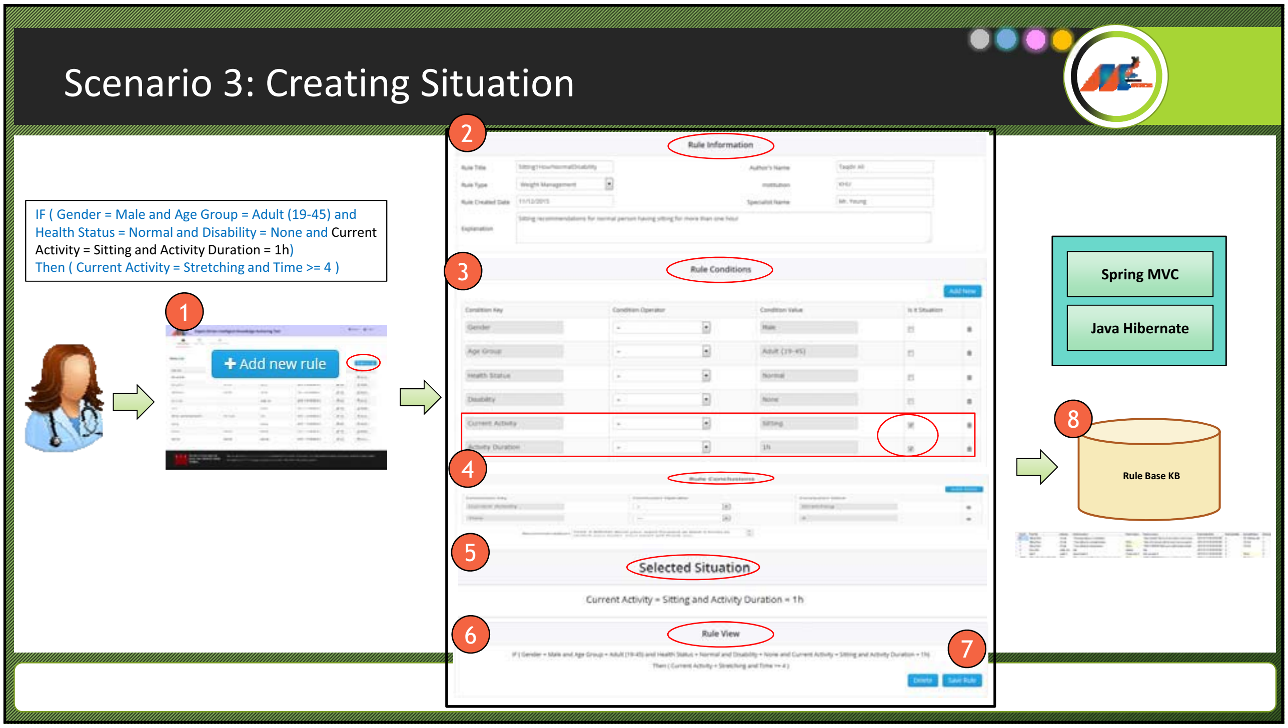
Multiple conclusions in "Then" part

Single view of selected situation and created rule

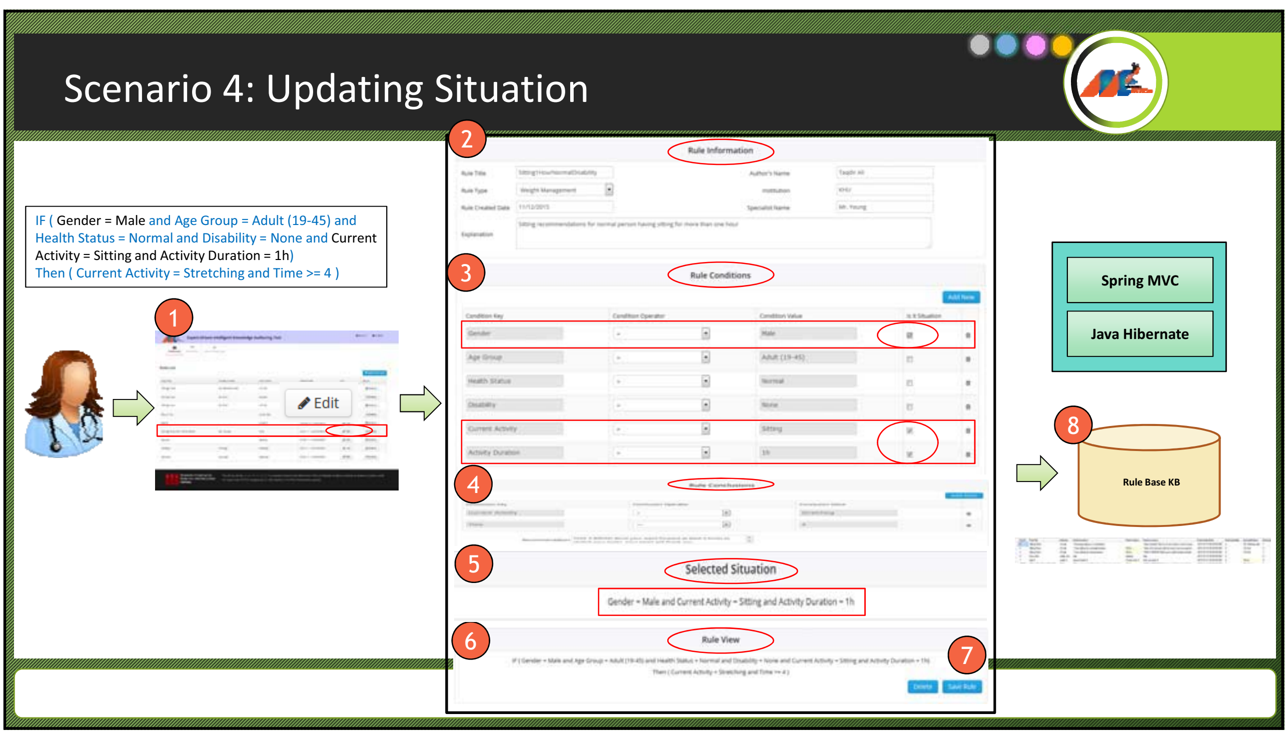



Scenario 1: Creating Rule
IF (Gender = Male and Age Group = Adult (19-45) and Health Status = Normal and Disability = None and Current Activity = Sitting and Activity Duration = 1h)
Then (Current Activity = Stretching and Time >= 4)
+ Add new rule
Rule Information
Rule Conditions
Rule Conclusions
Rule View
Spring MVC
Java Hibernate
Rule Base KB
1
2
3
4
5
6
7


Scenario 2: Updating Rule
IF (Gender = Male and Age Group = Adult (19-45) and Health Status = Disease and Disability = Visual Impairment and Current Activity = Sitting and Activity Duration = 1h)
Then (Current Activity = Stretching and Time >= 4)
Edit
Rule Information
Rule Conditions
Rule Conclusions
Rule View
Spring MVC
Java Hibernate
Rule Base KB
1
2
3
4
5
6
7



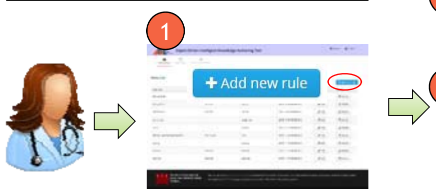
Scenario 3: Creating Situation
IF (Gender = Male and Age Group = Adult (19-45) and Health Status = Normal and Disability = None and Current Activity = Sitting and Activity Duration = 1h)
Then (Current Activity = Stretching and Time >= 4)
+ Add new rule
Rule Information
Rule Conditions
Selected Situation
Current Activity = Sitting and Activity Duration = 1h
Rule View
Spring MVC
Java Hibernate
Rule Base KB

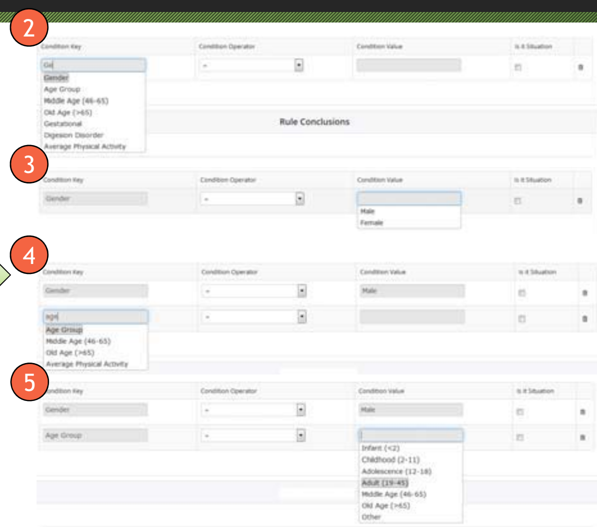


Scenario 4: Updating Situation
IF (Gender = Male and Age Group = Adult (19-45) and Health Status = Normal and Disability = None and Current Activity = Sitting and Activity Duration = 1h)
Then (Current Activity = Stretching and Time >= 4)
Edit
Rule Information
Rule Conditions
Selected Situation
Gender = Male and Current Activity = Sitting and Activity Duration = 1h
Rule View
Spring MVC
Java Hibernate
Rule Base KB

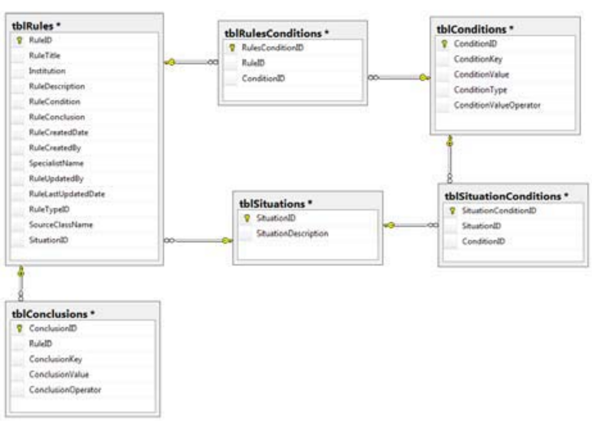
Scenario 5: Intelli-sense Scenario

IF (Gender = Male and Age Group = Adult (19-45) and Health Status = Normal and Disability = None and Current Activity = Sitting and Activity Duration = 1h)
Then (Current Activity = Stretching and Time >= 4)



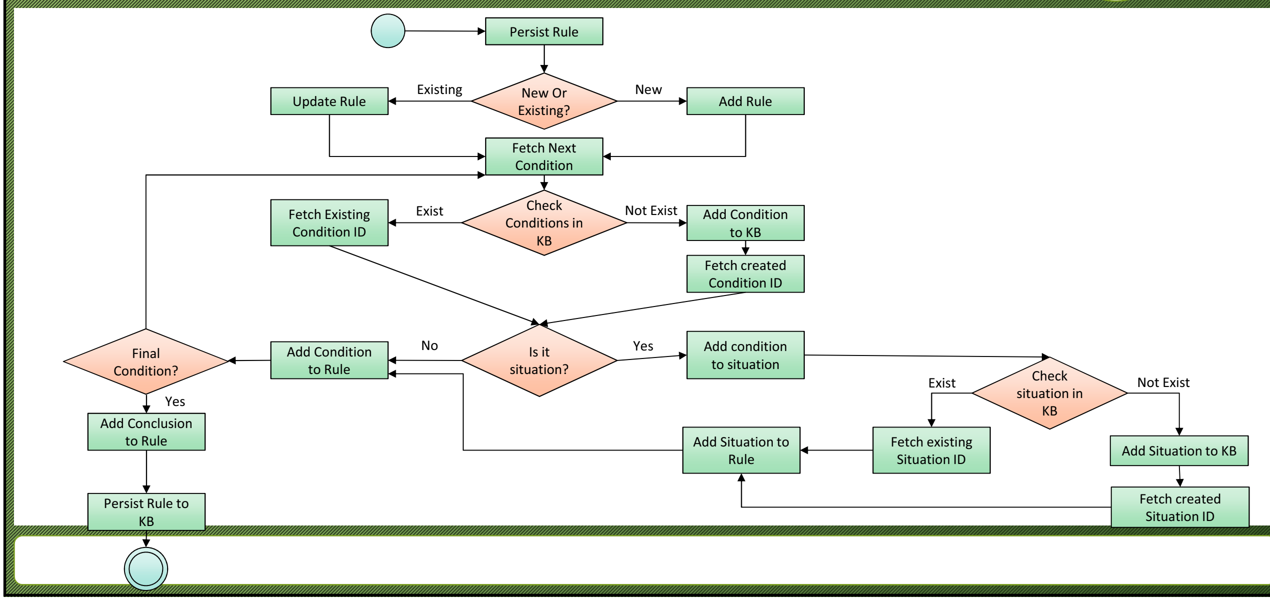


Scenario 6: Rules and Situation Persistence



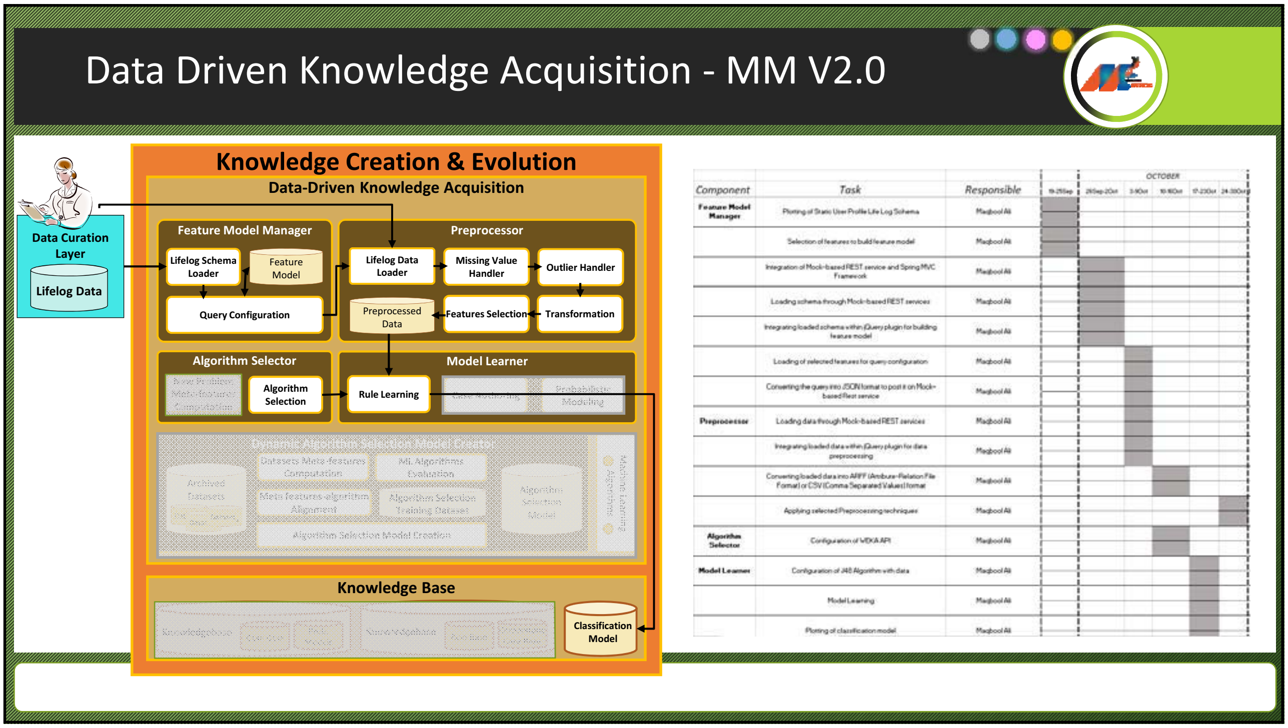
- Complicated Relationships
  - One Rule may have multiple Conditions
  - One Condition may be part of multiple Rules
  - One Situation may have multiple Conditions
  - One Condition may be part of multiple Situations
  - One Situation may have multiple Rules
- In order to maintain Rules and their Conditions relationships
  - No concept of deletion for Conditions
  - No concept of updating for Conditions
  - No Concept of deletion for Situation
  - No Concept of updating for Situation

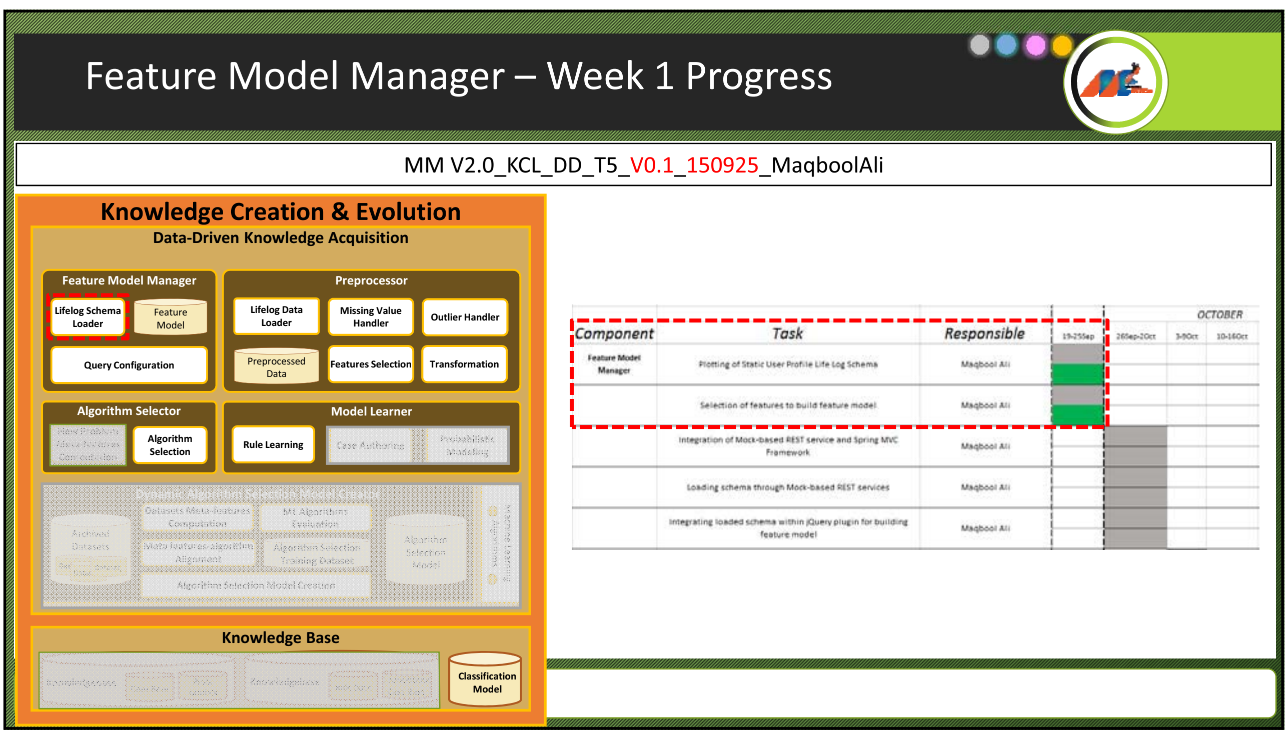
Scenario 6: Rules and Situation Persistence



Data Driven Knowledge Acquisition

Maqbool Ali

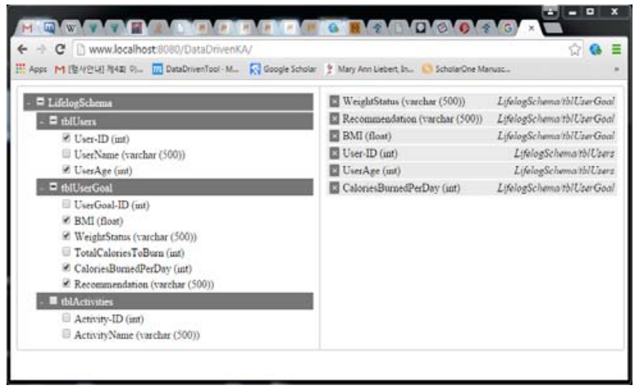

Data Driven Knowledge Acquisition - MM V2.0
Data Curation Layer
Lifelog Data
Knowledge Creation & Evolution
Data-Driven Knowledge Acquisition
Feature Model Manager
Lifelog Schema Loader
Feature Model
Query Configuration
Preprocessor
Lifelog Data Loader
Missing Value Handler
Outlier Handler
Preprocessed Data
Features Selection
Transformation
Algorithm Selector
Algorithm Selection
Model Learner
Rule Learning
Knowledge Base
Classification Model
OCTOBER
Component
Task
Responsible
Feature Model Manager
Plotting of Static User Profile Life Log Schema
Maqbool Ali
Selection of features to build feature model
Integration of Mock-based REST service and Spring MVC Framework
Loading schema through Mock-based REST services
Integrating loaded schema within jQuery plugin for building feature model
Loading of selected features for query configuration
Converting the query into JSON format to post it on Mock-based Rest service
Preprocessor
Loading data through Mock-based REST services
Integrating loaded data within jQuery plugin for data preprocessing
Converting loaded data into ARFF (Attribute-Relation File Format) or CSV (Comma Separated Values) format
Applying selected Preprocessing techniques
Algorithm Selector
Configuration of WEKA API
Model Learner
Configuration of J48 Algorithm with data
Model Learning
Plotting of classification model


Feature Model Manager – Week 1 Progress
MM V2.0_KCL_DD_T5_V0.1_150925_MaqboolAli
Knowledge Creation & Evolution
Data-Driven Knowledge Acquisition
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
Knowledge Base
Classification Model
OCTOBER
Component
Task
Responsible
19-25Sep
26Sep-2Oct
3-9Oct
10-16Oct
Feature Model Manager
Plotting of Static User Profile Life Log Schema
Maqbool Ali
Selection of features to build feature model
Integration of Mock-based REST service and Spring MVC Framework
Loading schema through Mock-based REST services
Integrating loaded schema within jQuery plugin for building feature model

Feature Model Manager – Week 1 Progress

MM V2.0_KCL_DD_T5_V0.1_150925_MaqboolAli

- Achievements
  - Plotting of Static User Profile Life Log Schema
  - Selection of features to build feature model
- Challenges
  - Spring MVC setup and configuration
  - Finding appropriate plugin according to requirement
  - Configuration of static resource (css, js, html) with Spring MVC Framework



Feature Model Manager – Week 2 Future Plan

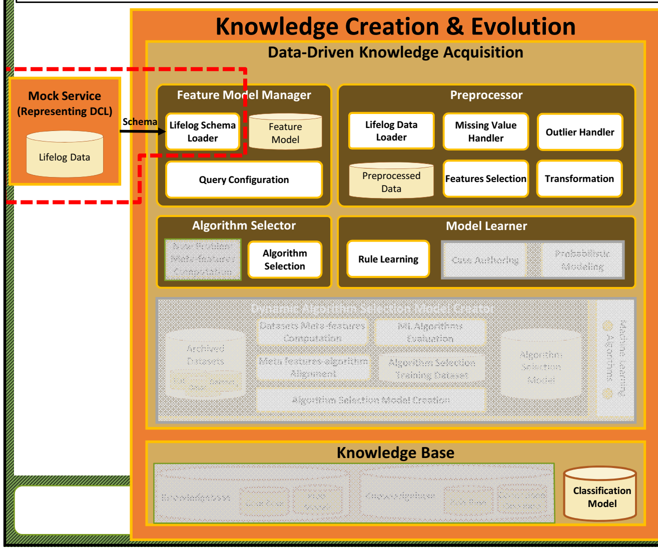
MM V2.0_KCL_DD_T5_V0.1_150925_MaqboolAli

- Integration of Mock-based REST service and Spring MVC Framework
  - What is Mock service?
    - No real implementation
    - For testing purpose only
  - Why Mock Service?
    - As no DCL end point is ready for communication
    - Mock service will mimic the exact behavior of DCL end point without any backend implementation
  - Available well-known platforms for Mock service
    - Apiary (https://apiary.io/)
    - RAML (http://raml.org)
  - Why Apiary?
    - Used to quickly design and test APIs
    - Easy to learn
    - Required Markdown markup language, which is a formatting "language" like HTML that is used to specify the final appearance of text
- Loading schema through Mock-based REST services
- Integrating loaded schema within jQuery plugin for building feature model

				OCTOBER		
Component	Task	Responsible	19-25Sep	26Sep-2Oct	3-9Oct	10-16Oct
Feature Model Manager	Plotting of Static User Profile Life Log Schema	Maqbool Ali				
	Selection of features to build feature model	Maqbool Ali				
	Integration of Mock-based REST service and Spring MVC Framework	Maqbool Ali				
	Loading schema through Mock-based REST services	Maqbool Ali				
	Integrating loaded schema within jQuery plugin for building feature model	Maqbool Ali				

Feature Model Manager – Week 2 Progress

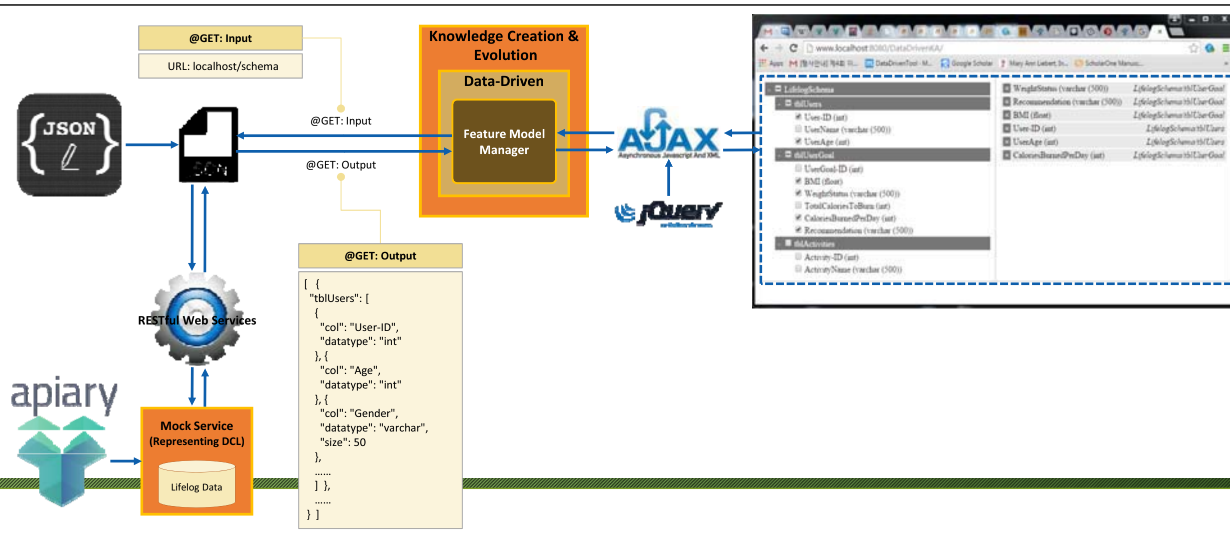
MM V2.0_KCL_DD_T5_V0.2_151002_MaqboolAli



Component	Task	Responsible	19-25Sep	28Sep-2Oct	3-9Oct	10-16Oct
				OCTOBER		
Feature Model Manager	Plotting of Static User Profile Life Log Schema	Maqbool Ali				
	Selection of features to build feature model	Maqbool Ali				
	Integration of Mock-based REST service and Spring MVC Framework	Maqbool Ali				
	Loading schema through Mock-based REST services	Maqbool Ali				
	Integrating loaded schema within jQuery plugin for building feature model	Maqbool Ali				
	Loading of selected features for query configuration	Maqbool Ali				
	Converting the query into JSON format to post it on Mock-based Rest service	Maqbool Ali				
Preprocessor	Loading data through Mock-based REST services	Maqbool Ali				
	Integrating loaded data within jQuery plugin for data preprocessing	Maqbool Ali				

Feature Model Manager – Week 2 Progress

MM V2.0_KCL_DD_T5_V0.2_151002_MaqboolAli



Feature Model Manager – Week 2 Progress

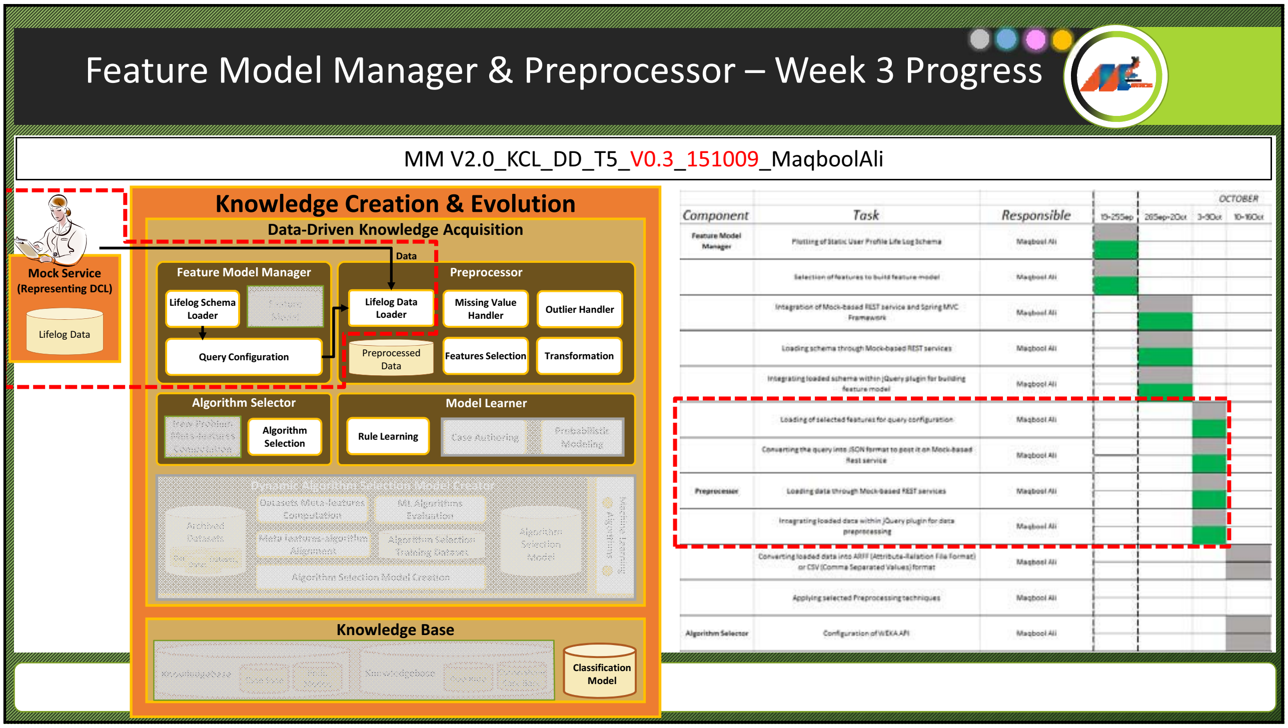
MM V2.0_KCL_DD_T5_V0.2_151002_MaqboolAli

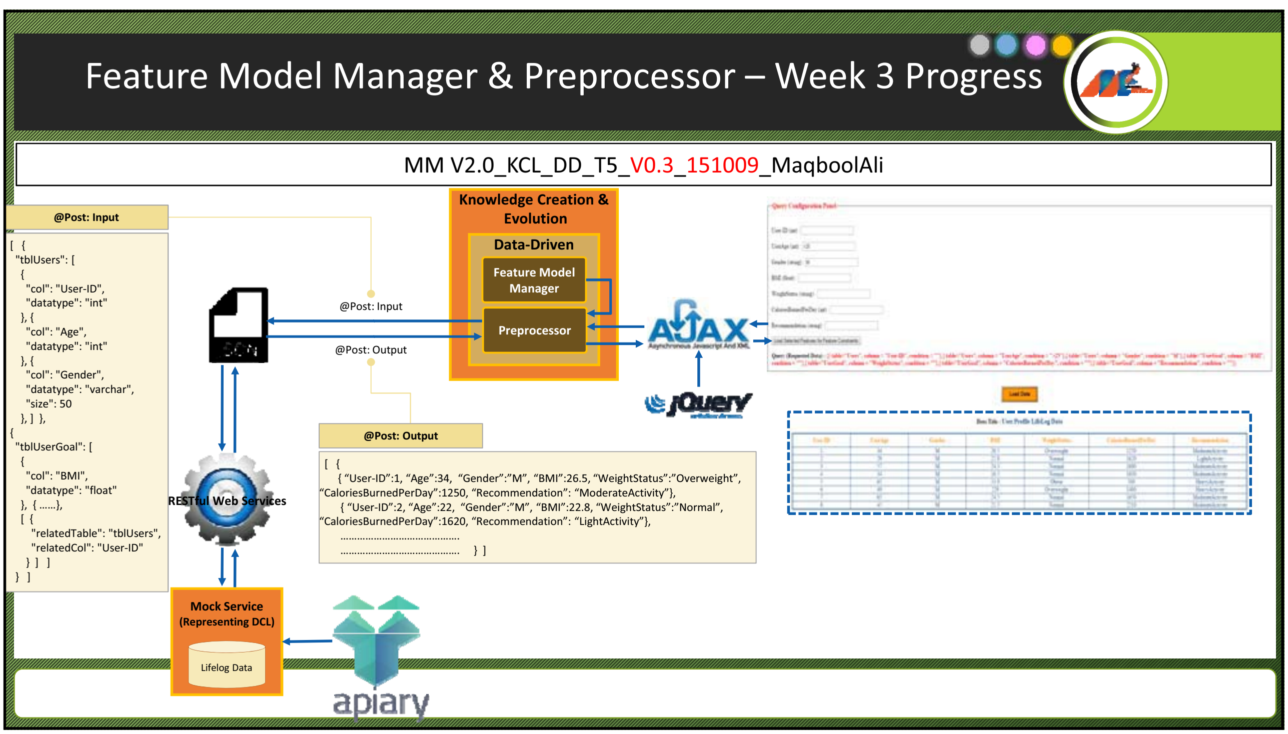
- Achievements
  - Integration of Mock-based REST service and Spring MVC Framework
  - Loading schema through Mock-based REST services
  - Integrating loaded schema within jQuery plugin for building feature model
- Challenges
  - Creating valid JSON object
  - Creating the Prototype in Apiary to build Mock service
  - Testing of Mock service
  - Parsing the JSON object using AJAX
  - Embedding returned JSON object within jQuery plugin
  - Configuration AJAX with Spring MVC Framework

Feature Model Manager – Week 3 Future Plan

MM V2.0_KCL_DD_T5_V0.2_151002_MaqboolAli

- 1 - Loading of selected features for query configuration
- 2 - Converting the query into JSON format to post it on Mock-based Rest service
- 3 - Loading data through Mock-based REST services
- 4 - Integrating loaded data within jQuery plugin for data preprocessing


Feature Model Manager & Preprocessor – Week 3 Progress
MM V2.0_KCL_DD_T5_V0.3_151009_MaqboolAli
Mock Service (Representing DCL)
Lifelog Data
Knowledge Creation & Evolution
Data-Driven Knowledge Acquisition
Data
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
Case Authoring
Probabilistic Modeling
Knowledge Base
Classification Model
OCTOBER
Component
Task
Responsible
19-25Sep
26Sep-2Oct
3-9Oct
10-16Oct
Feature Model Manager
Plotting of Static User Profile Life Log Schema
Maqbool Ali
Selection of features to build feature model
Maqbool Ali
Integration of Mock-based REST service and Spring MVC Framework
Maqbool Ali
Loading schema through Mock-based REST services
Maqbool Ali
Integrating loaded schema within jQuery plugin for building feature model
Maqbool Ali
Loading of selected features for query configuration
Maqbool Ali
Converting the query into JSON format to post it on Mock-based Rest service
Maqbool Ali
Preprocessor
Loading data through Mock-based REST services
Maqbool Ali
Integrating loaded data within jQuery plugin for data preprocessing
Maqbool Ali
Converting loaded data into ARFF (Attribute-Relation File Format) or CSV (Comma Separated Values) format
Maqbool Ali
Applying selected Preprocessing techniques
Maqbool Ali
Algorithm Selector
Configuration of WEKA API
Maqbool Ali


Feature Model Manager & Preprocessor – Week 3 Progress
MM V2.0_KCL_DD_T5_V0.3_151009_MaqboolAli
@Post: Input
[{
"tblUsers": [
{
"col": "User-ID",
"datatype": "int"
}, {
"col": "Age",
"datatype": "int"
}, {
"col": "Gender",
"datatype": "varchar",
"size": 50
},] },
{
"tblUserGoal": [
{
"col": "BMI",
"datatype": "float"
}, { ......},
[{
"relatedTable": "tblUsers",
"relatedCol": "User-ID"
}]]
}]
@Post: Input
@Post: Output
Knowledge Creation & Evolution
Data-Driven
Feature Model Manager
Preprocessor
AJAX
Asynchronous JavaScript And XML
jQuery
RESTful Web Services
@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"},
}]
Mock Service (Representing DCL)
Lifelog Data
apiary

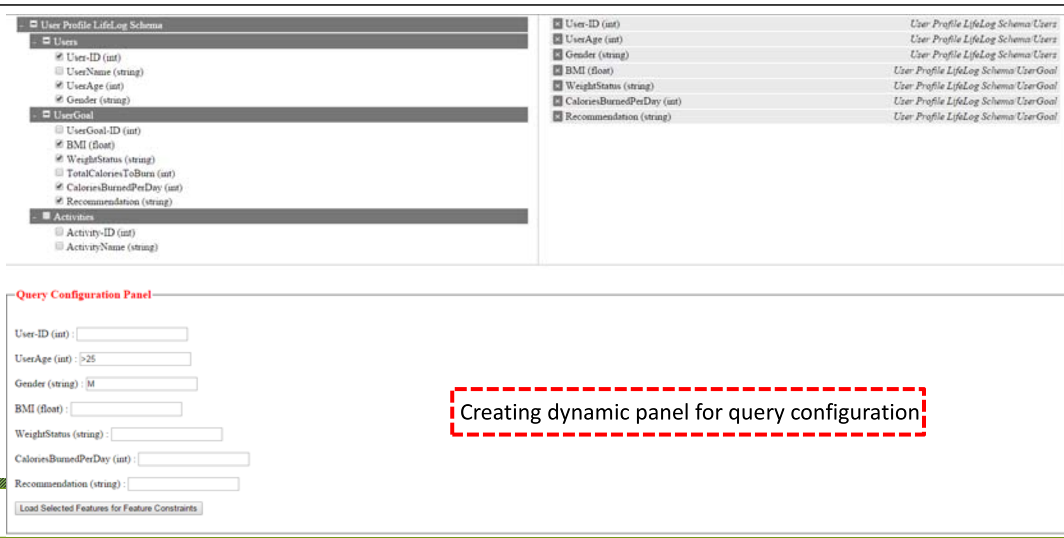
Feature Model Manager & Preprocessor – Week 3 Progress

MM V2.0_KCL_DD_T5_V0.3_151009_MaqboolAli

- Achievements
  - Feature Model Manager
    - Loading of selected features for query configuration
    - Converting the query into JSON format to post it on Mock-based Rest service
  - Preprocessor
    - Loading data through Mock-based REST services
    - Integrating loaded data within jQuery plugin for data preprocessing
- Challenges
  - Feature Model Manager
    - Creating dynamic panel for query configuration
    - Creating valid JSON object for sending query (data request)
  - Preprocessor
    - Creating valid JSON object for requested data
    - Creating the data prototype in Apiary
    - Testing of Mock service
    - Parsing the JSON object using AJAX
    - Creating valid HTML for displaying data

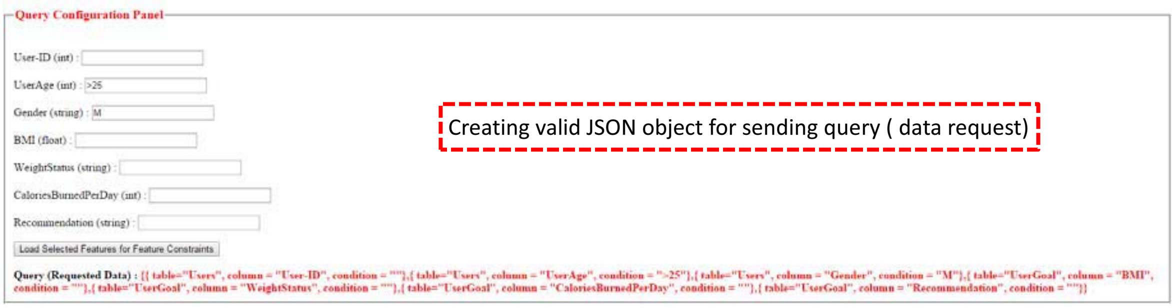
Feature Model Manager – Week 3 Progress

MM V2.0_KCL_DD_T5_V0.3_151009_MaqboolAli



Feature Model Manager – Week 3 Progress

MM V2.0_KCL_DD_T5_V0.3_151009_MaqboolAli



Model Learner – Week 4 Future Plan

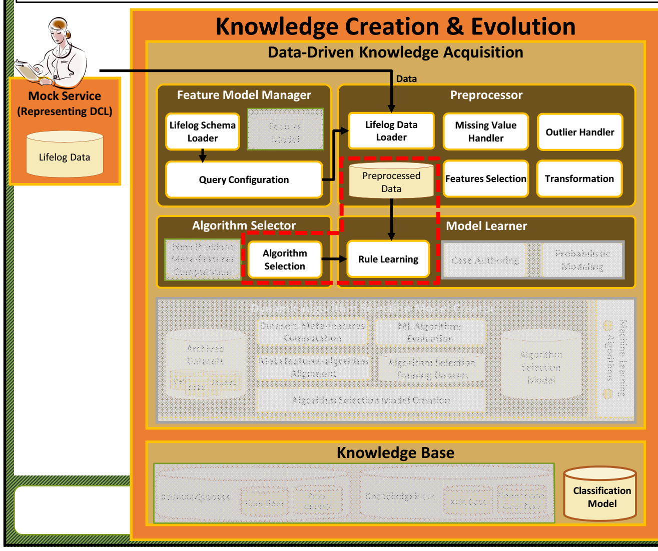
MM V2.0_KCL_DD_T5_V0.3_151009_MaqboolAli

- Converting loaded data into ARFF (Attribute-Relation File Format) or CSV (Comma Separated Values) format
- Configuration of WEKA API

					OCTOBER	
Component	Task	Responsible	19-25Sep	26Sep-2Oct	3-9Oct	10-16Oct
Feature Model Manager	Plotting of Static User Profile Life Log Schema	Maqbool Ali				
	Selection of features to build feature model	Maqbool Ali				
	Integration of Mock-based REST service and Spring MVC Framework	Maqbool Ali				
	Loading schema through Mock-based REST services	Maqbool Ali				
	Integrating loaded schema within jQuery plugin for building feature model	Maqbool Ali				
	Loading of selected features for query configuration	Maqbool Ali				
	Converting the query into JSON format to post it on Mock-based Rest service	Maqbool Ali				
Preprocessor	Loading data through Mock-based REST services	Maqbool Ali				
	Integrating loaded data within jQuery plugin for data preprocessing	Maqbool Ali				
	Converting loaded data into ARFF (Attribute-Relation File Format) or CSV (Comma Separated Values) format	Maqbool Ali				
	Applying selected Preprocessing techniques	Maqbool Ali				
Algorithm Selector	Configuration of WEKA API	Maqbool Ali				

Model Learner – Week 4 Progress

MM V2.0_KCL_DD_T5_V0.4_151016_MaqboolAli

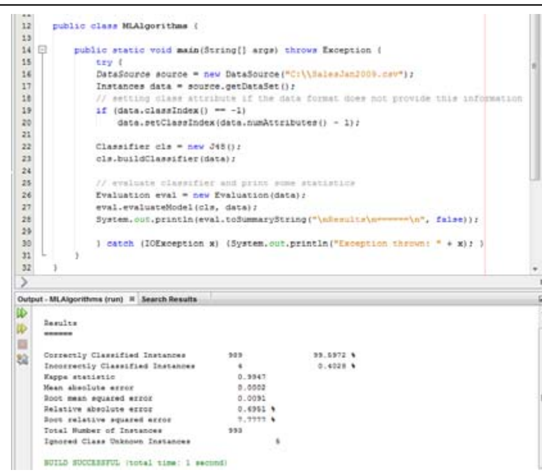


Component	Task	Responsible	19-25Sep	26Sep-2Oct	3-9Oct	10-16Oct	17-23Oct
Feature Model Manager	Plotting of Static User Profile Life Log Schema	Maqbool Ali					
	Selection of features to build feature model	Maqbool Ali					
	Integration of Mock-based REST service and Spring MVC Framework	Maqbool Ali					
	Loading schema through Mock-based REST services	Maqbool Ali					
	Integrating loaded schema within jQuery plugin for building feature model	Maqbool Ali					
	Loading of selected features for query configuration	Maqbool Ali					
	Converting the query into JSON format to post it on Mock-based Rest service	Maqbool Ali					
Preprocessor	Loading data through Mock-based REST services	Maqbool Ali					
	Integrating loaded data within jQuery plugin for data preprocessing	Maqbool Ali					
	Converting loaded data into CSV (Comma Separated Values) format	Maqbool Ali					
	Applying selected Preprocessing techniques	Maqbool Ali					
Algorithm Selector	Configuration of WEKA API	Maqbool Ali					
Model Learner	Configuration of J48 Algorithm with data	Maqbool Ali					
	Model Learning						

Model Learner – Week 4 Progress

MM V2.0_KCL_DD_T5_V0.4_151016_MaqboolAli

- Achievements
  - Configuration of WEKA API
  - Model Learning
- Challenges
  - Finding appropriate jar files
  - Getting data from CSV files through Java



Model Learner – Week 5 Future Plan

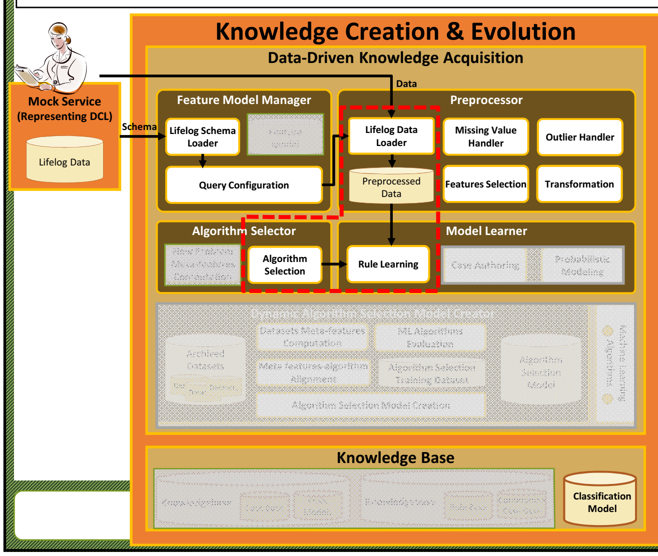
MM V2.0_KCL_DD_T5_V0.4_151016_MaqboolAli

- Converting loaded data into CSV (Comma Separated Values) format
- Configuration of J48 Algorithm WEKA API with Spring MVC Framework

Component	Task	Responsible	19-25Sep	26Sep-2Oct	3-9Oct	10-16Oct	17-23Oct
Feature Model Manager	Plotting of Static User Profile Life Log Schema	Maqbool Ali					
	Selection of features to build feature model	Maqbool Ali					
	Integration of Mock-based REST service and Spring MVC Framework	Maqbool Ali					
	Loading schema through Mock-based REST services	Maqbool Ali					
	Integrating loaded schema within jQuery plugin for building feature model	Maqbool Ali					
	Loading of selected features for query configuration	Maqbool Ali					
	Converting the query into JSON format to post it on Mock-based Rest service	Maqbool Ali					
Preprocessor	Loading data through Mock-based REST services	Maqbool Ali					
	Integrating loaded data within jQuery plugin for data preprocessing	Maqbool Ali					
	Converting loaded data into CSV (Comma Separated Values) format	Maqbool Ali					
	Applying selected Preprocessing techniques	Maqbool Ali					
Algorithm Selector	Configuration of WEKA API	Maqbool Ali					
Model Learner	Configuration of J48 Algorithm with data	Maqbool Ali					
	Model Learning						

Preprocessor & Model Learner – Week 5 Progress

MM V2.0_KCL_DD_T5_V0.5_151023_MaqboolAli



Component	Task	Responsible	19-25Sep	26Sep-2Oct	3-9Oct	10-16Oct	17-23Oct	24-30Oct
Preprocessor	Loading data through Mock-based REST services	Maqbool Ali						
	Integrating loaded data within jQuery plugin for data preprocessing	Maqbool Ali						
	Converting loaded data into CSV (Comma Separated Values) format	Maqbool Ali						
	Applying selected Preprocessing techniques	Maqbool Ali						
Algorithm Selector	Configuration of WEKA API	Maqbool Ali						
Model Learner	Configuration of J48 Algorithm with data	Maqbool Ali						
	Model Learning	Maqbool Ali						
	Plotting of classification model	Maqbool Ali						

Preprocessor & Model Learner – Week 5 Progress

MM V2.0_KCL_DD_T5_V0.5_151023_MaqboolAli

- Achievements
  - Converting loaded data into CSV (Comma Separated Values) format
  - Configuration of J48 Algorithm WEKA API with Spring MVC Framework
- Challenges
  - Maintaining JSON Object order and converting into CSV format
  - Configuration of J48 Algorithm with actual data

Model Learner – Week 6 Future Plan

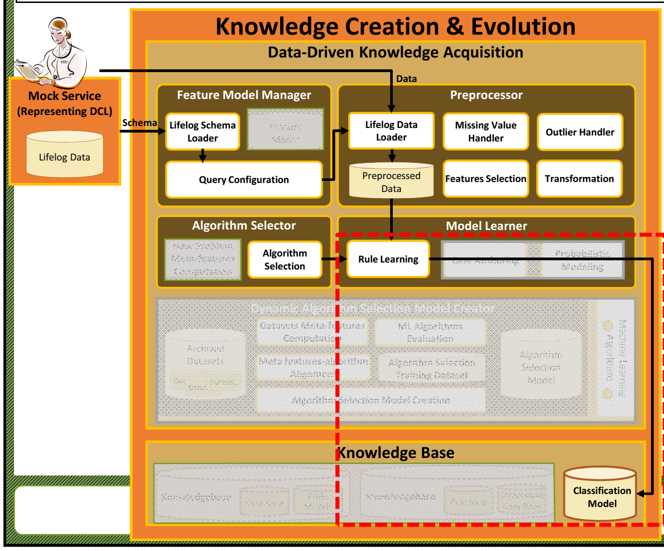
MM V2.0_KCL_DD_T5_V0.5_151023_MaqboolAli

- Plotting of classification model

Component	Task	Responsible	19-25Sep	26Sep-2Oct	3-9Oct	10-16Oct	17-23Oct	24-30Oct
Preprocessor	Loading data through Mock-based REST services	Maqbool Ali						
	Integrating loaded data within jQuery plugin for data preprocessing	Maqbool Ali						
	Converting loaded data into CSV (Comma Separated Values) format	Maqbool Ali						
	Applying selected Preprocessing techniques	Maqbool Ali						
Algorithm Selector	Configuration of WEKA API	Maqbool Ali						
Model Learner	Configuration of J48 Algorithm with data	Maqbool Ali						
	Model Learning	Maqbool Ali						
	Plotting of classification model	Maqbool Ali						

Model Learner – Week 6 Progress

MM V2.0_KCL_DD_T5_V0.6_151030_MaqboolAli

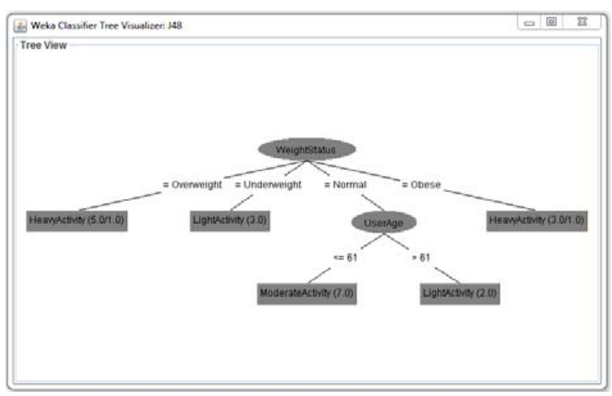


Component	Task	Responsible	3-9Oct	10-16Oct	17-23Oct	24-30Oct	31Oct-6No
Preprocessor	Loading data through Mock-based REST services	Maqbool Ali					
	Integrating loaded data within jQuery plugin for data preprocessing	Maqbool Ali					
	Converting loaded data into CSV (Comma Separated Values) format	Maqbool Ali					
	Applying selected Preprocessing techniques	Maqbool Ali					
Algorithm Selector	Configuration of WEKA API	Maqbool Ali					
Model Learner	Configuration of J48 Algorithm with data	Maqbool Ali					
	Model Learning	Maqbool Ali					
	Plotting of classification model	Maqbool Ali					

Model Learner – Week 6 Progress

MM V2.0_KCL_DD_T5_V0.6_151030_MaqboolAli

- Achievements
  - Plotting of classification model
  - Storing the classification model



Preprocessor – Week 7 Future Plan

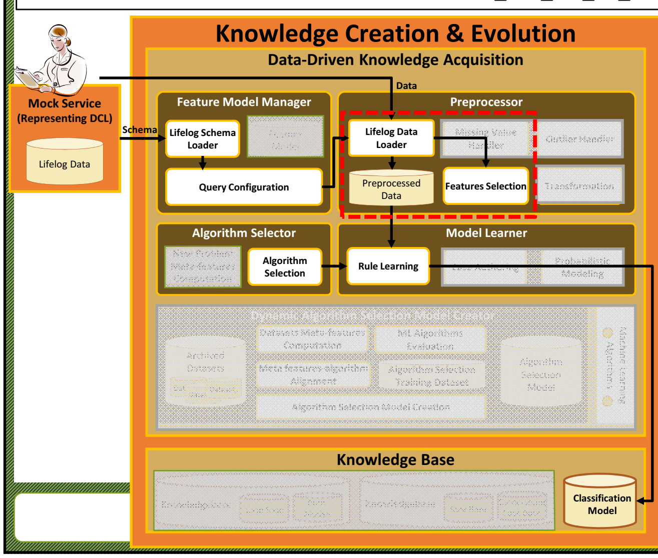
MM V2.0_KCL_DD_T5_V0.6_151030_MaqboolAli

- Features Selection

Component	Task	Responsible	3-9Oct	10-16Oct	17-23Oct	24-30Oct	1Oct-6No
Preprocessor	Loading data through Mock-based REST services	Maqbool Ali					
	Integrating loaded data within jQuery plugin for data preprocessing	Maqbool Ali					
	Converting loaded data into CSV (Comma Separated Values) format	Maqbool Ali					
	Applying selected Preprocessing techniques	Maqbool Ali					
Algorithm Selector	Configuration of WEKA API	Maqbool Ali					
Model Learner	Configuration of J48 Algorithm with data	Maqbool Ali					
	Model Learning	Maqbool Ali					
	Plotting of classification model	Maqbool Ali					

Preprocessor – Week 7 Progress

MM V2.0_KCL_DD_T5_V0.7_151106_MaqboolAli



Component	Task	Responsible	3-9Oct	10-16Oct	17-23Oct	24-30Oct	1Oct-6No	7-13Nov
Preprocessor	Loading data through Mock-based REST services	Maqbool Ali						
	Integrating loaded data within jQuery plugin for data preprocessing	Maqbool Ali						
	Converting loaded data into CSV (Comma Separated Values) format	Maqbool Ali						
	Applying selected Preprocessing techniques	Maqbool Ali						
Algorithm Selector	Configuration of WEKA API	Maqbool Ali						
Model Learner	Configuration of J48 Algorithm with data	Maqbool Ali						
	Model Learning	Maqbool Ali						
	Plotting of classification model	Maqbool Ali						
GUI Refinement	Integrating Bootstrap framework with GUI	Maqbool Ali						

Preprocessor – Week 7 Progress

MM V2.0_KCL_DD_T5_V0.7_151106_MaqboolAli

- Achievements
  - Features Selection

```java
@RequestMapping(value = "/featureselector", method = RequestMethod.POST)
public String featureSelector() throws Exception {

    CSVLoader loader=new CSVLoader();
    loader.setSource(new File("D:\\DDKAResources\\LifelogData.csv"));
    Instances data=loader.getDataSet();
          data.setClassIndex(data.numAttributes() - 1);
    AttributeSelection attsel = new AttributeSelection();
    CfsSubsetEval eval = new CfsSubsetEval();
    GreedyStepwise search = new GreedyStepwise();
    search.setSearchBackwards(true);
    attsel.setEvaluator(eval);
    attsel.setSearch(search);
    attsel.SelectAttributes(data);
    int[] indices = attsel.selectedAttributes();
    String selectedAttributesString="";
    for (int i=0; i < indices.length; i++) {
       System.out.println(data.attribute(i).toString());
       selectedAttributesString += data.attribute(i).toString() +", ";
     }
    selectedAttributesString = selectedAttributesString.substring(0, selectedAttributesString.length()-2);
    System.out.println("selected attribute : " + selectedAttributesString);
```

Selected Features

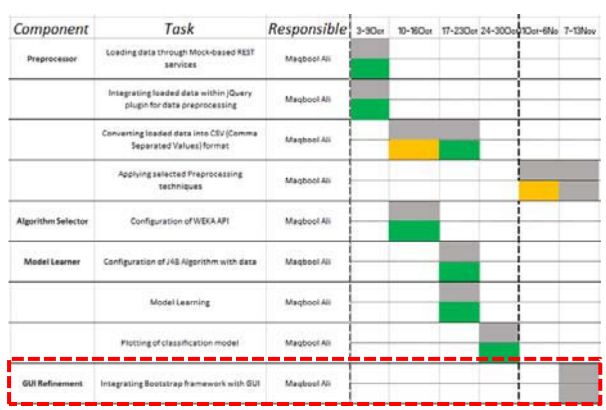
Features selected by Machine Learning Methods are : @attribute User-ID numeric, @attribute UserAge numeric, @attribute Gender {M,F}

OK

GUI Refinement – Week 8 Future Plan

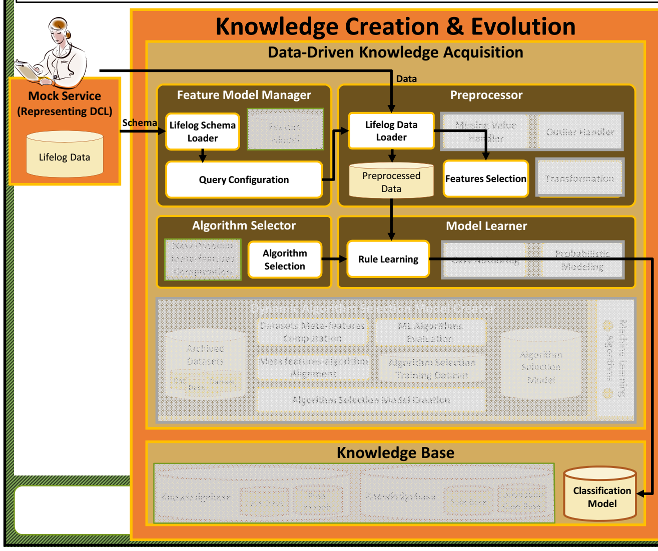
MM V2.0_KCL_DD_T5_V0.7_151106_MaqboolAli

- Integrating Bootstrap Framework



GUI Refinement – Week 8 Progress

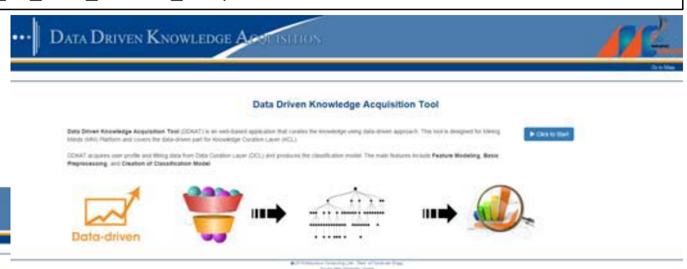
MM V2.0_KCL_DD_T5_V0.8_151113_MaqboolAli



GUI Refinement – Week 8 Progress

MM V2.0_KCL_DD_T5_V0.8_151113_MaqboolAli

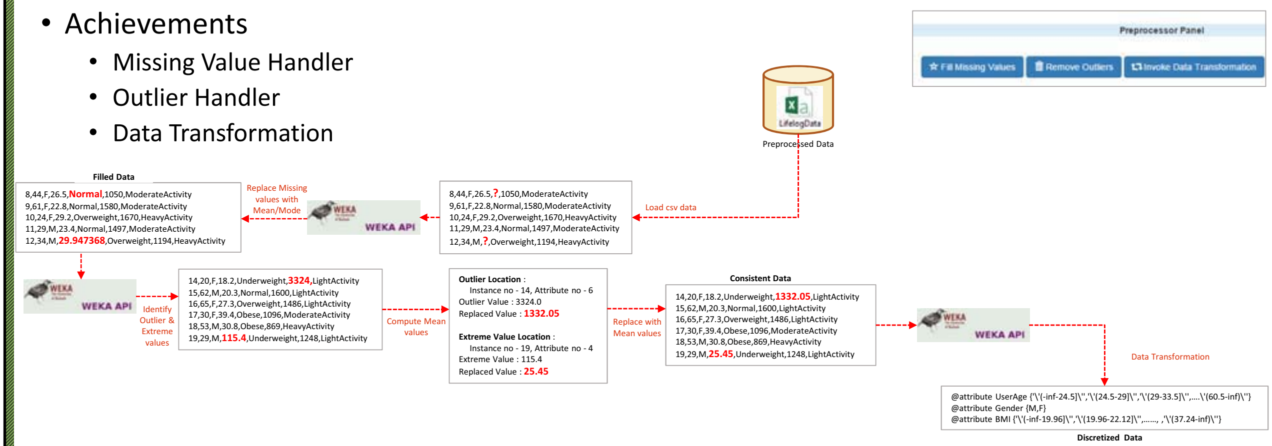
- Achievements
  - Integrating Bootstrap Framework
  - Fetching data from the server without loading a new page



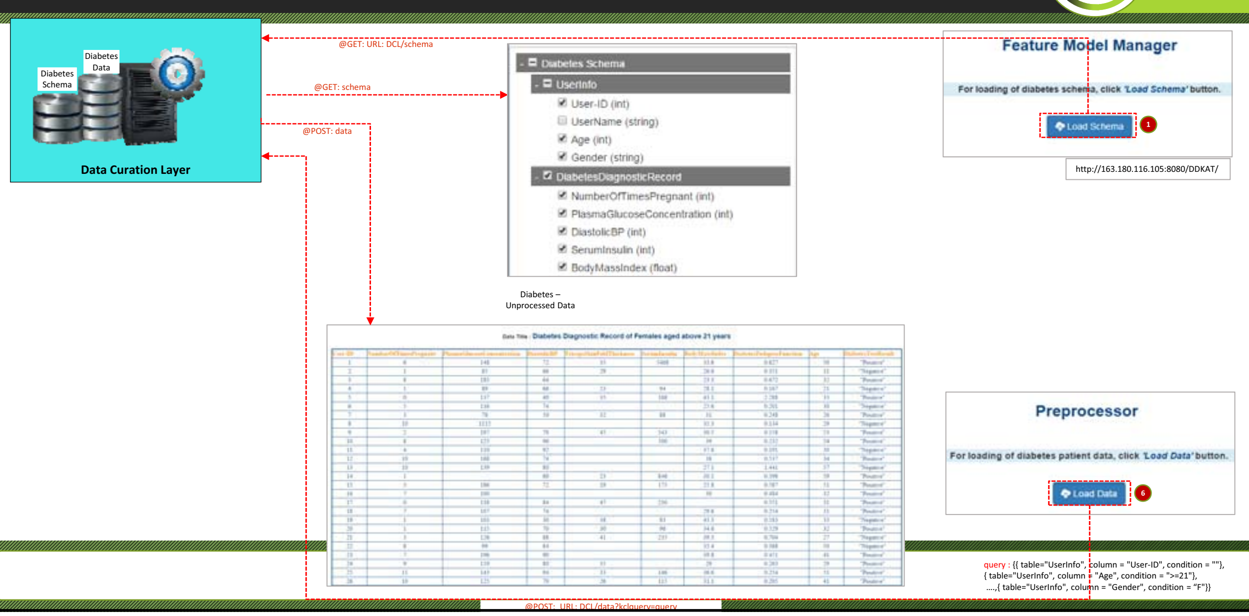
Preprocessing – Week 9 Progress

MM V2.0_KCL_DD_T5_V0.9_151120_MaqboolAli

- Achievements
  - Missing Value Handler
  - Outlier Handler
  - Data Transformation



Integration with DCL – Week 10 and Onward Progress



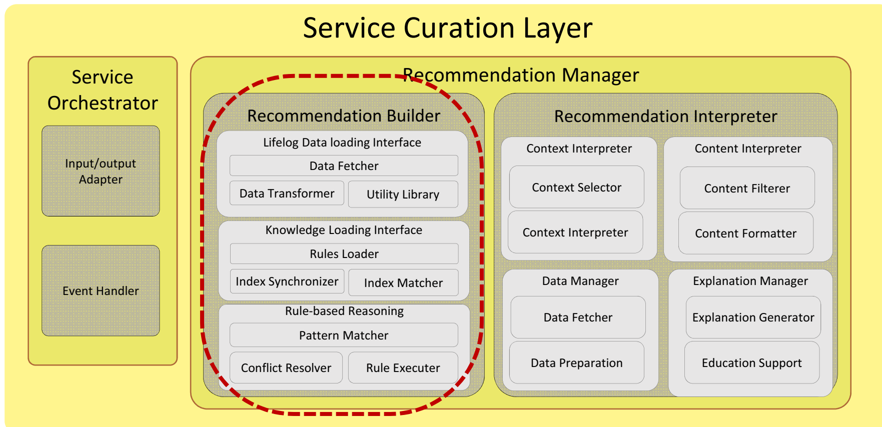
Service Curation Layer (SCL 2.0)

Principal Investigator:
Professor Sungyoung Lee

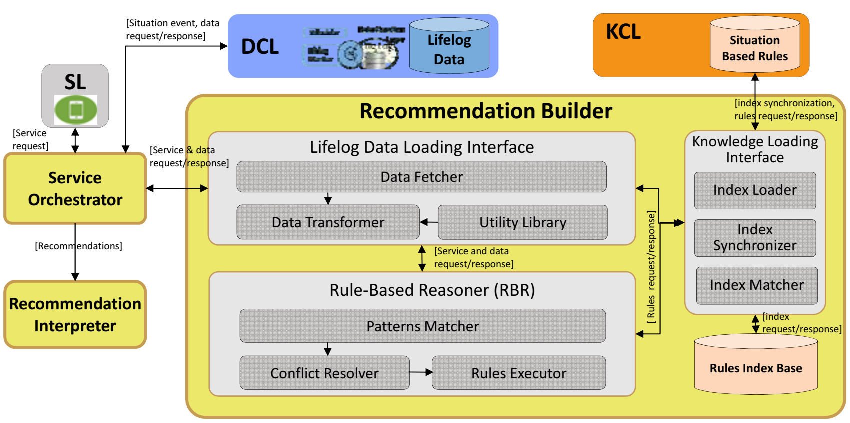
Recommendation Builder

Rahman Ali

Service Curation Layer

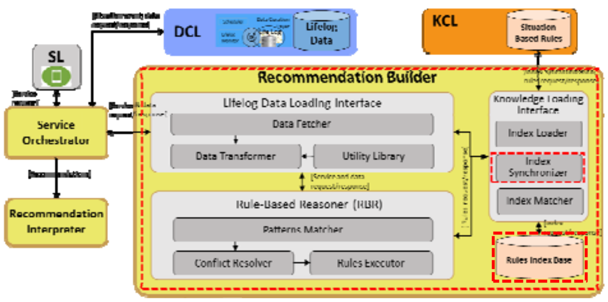


Recommendation Builder - Architecture



Recommendation Builder – Week 1 Progress

MM V2.0_SCL_T5_V0.2_20150925_Rahman Ali

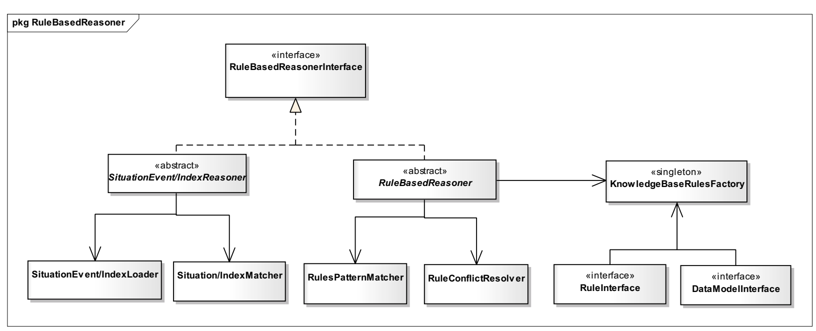


Component	Task	Responsible	19-25Sep	26Sep-2Oct	3-9Oct	10-16Oct
Rule-based reasoning Framework Design	Implementation design of the rule-based reasoning framework	Rahman, Maqbool Hussain Afzal				
Synchronization of KB	Skeleton definition of the local copy of Indexed Knowledge Base	Rahman, Afzal				
	Service definition for knowledge Base Synchronization					
Index-Based Reasoning	Schema Definition of the Index Knowledge Base	Rahman/Taqdir				
	Implementation of Index-based Reasoning algorithm over Indexed Knowledge Base					
Utility Librararies	Representation of utility library: categorization and definition of utility functions	Rahman, Imran				
	Implementation of utility functions over user profile data					
	Implementation of utility functions over user lifelog data					
	Integration of utility library with recommendation builder/Interpreter and Testing					

Recommendation Builder – Week 1 Progress

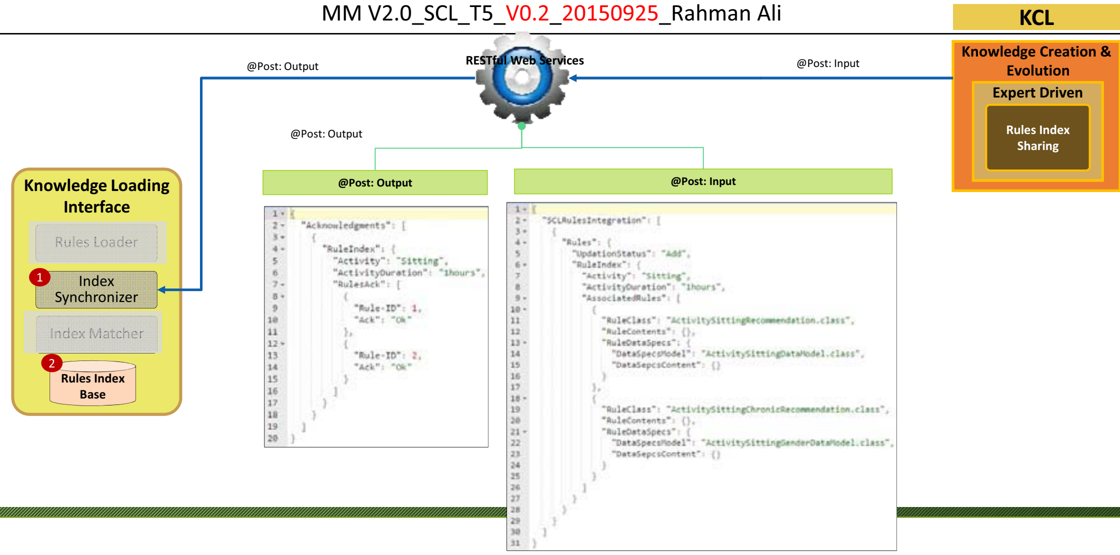
MM V2.0_SCL_T5_V0.2_20150925_Rahman Ali

- Environment setup
  - Eclipse Installation (JRE 1.8.0 (64 Bit))
- SCL-KCL Integration Framework
  - Rule-based Reasoning Framework implementation design
  - Skeleton definition of Indexed Knowledge Base
  - Restful Web Service contract specification for Knowledge and Index Synchronization
- Dependencies/Challenge
  - KCL Knowledge Base/Index Base



Recommendation Builder – Week 1 Progress

MM V2.0_SCL_T5_V0.2_20150925_Rahman Ali



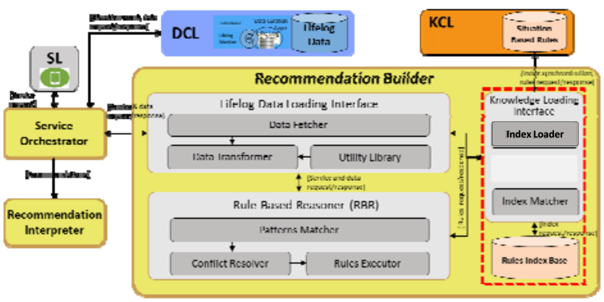
Recommendation Builder – Week 2 Future Plan

MM V2.0_SCL_T5_V0.2_20150925_Rahman Ali

- Schema definition for the Index Knowledge Base
- Index-Based Reasoning
  - Implementation of Index-based Reasoning algorithm for Index and Rules Loading

Recommendation Builder – Week 2 Progress

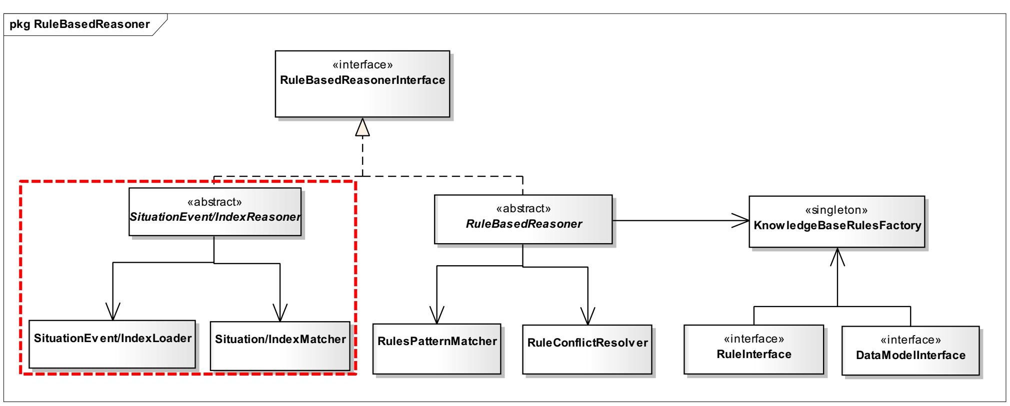
MM V2.0_SCL_T5_V0.3_20151003_Rahman Ali



Component	Task	Responsible	19-25Sep	26Sep-2Oct	3-9Oct	10-16Oct
Rule-based reasoning Framework Design	Implementation design of the rule-based reasoning framework	Rahman, Maqbool Hussain Afzal				
Synchronization of KB	Skeleton definition of the local copy of Indexed Knowledge Base	Rahman, Afzal				
	Service definition for knowledge Base Synchronization					
Index-Based Reasoning	Schema Definition of the Index Knowledge Base	Rahman/Taqdir				
	Implementation of Index-based Reasoning algorithm over Indexed Knowledge Base	Rahman				
Utility Librararies	Representation of utility library: categorization and definition of utility functions	Rahman, Imran				
	Implementation of utility functions over user profile data					
	Implementation of utility functions over user lifelog data					
	Integration of utility library with recommendation builder/Interpreter and Testing					

Recommendation Builder – Week 2 Progress

MM V2.0_SCL_T5_V0.3_20151003_Rahman Ali

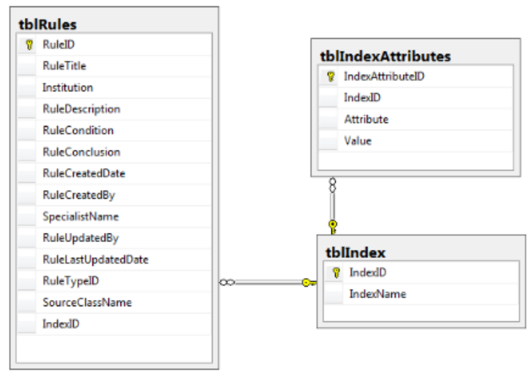


Recommendation Builder – Week 2 Progress

MM V2.0_SCL_T5_V0.3_20151003_Rahman Ali

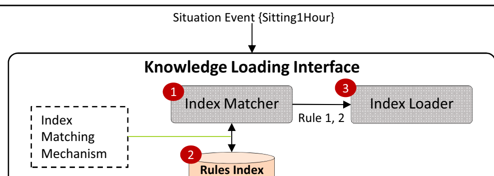
- Schema of Index Knowledge Base
  - Schema is defined for Index Base
  - Index-based Reasoning Mechanism is implemented using SQL Server Store Procedure
  - Challenges
    - The definition of generic store procedure to cover all types of situation indexes requests
    - How to handle situation indexes with varying number of salient features

Focus of MMV2.0



Recommendation Builder – Week 2 Progress

MM V2.0_SCL_T5_V0.3_20151003_Rahman Ali



Index

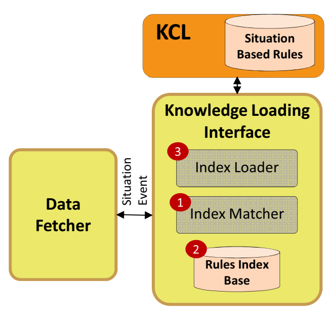
Index Id	Index Name
1	Sitting1Hour
2	Sitting1HourGender
.	
n	

Index Attributes

Index Id	Attribute Key	Value
1	Activity	Sitting
1	Activity duration	1Hour
2	Activity	Sitting
2	Activity duration	1Hour
2	Gender	Male
.		

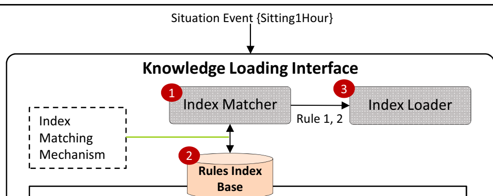
Indexed Rules

Rule Id	Rule Name	Index Id
1	Sitting1HoursRecommendation.cls	1
2	Sitting1HoursGenderRecommendation.cls	1
.		2
n		.



Recommendation Builder – Week 2 Progress

MM V2.0_SCL_T5_V0.3_20151003_Rahman Ali



Index

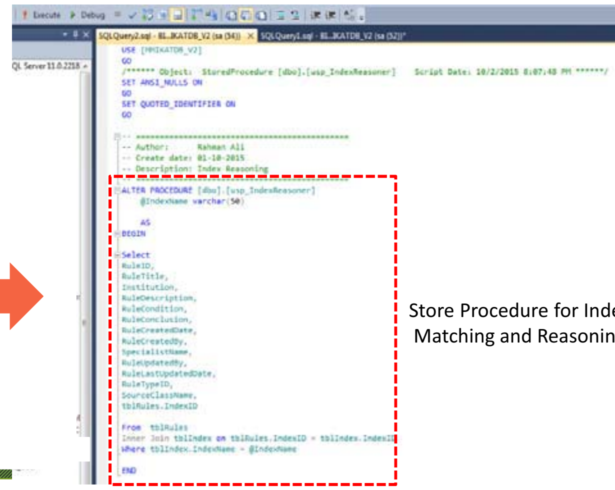
Index Id	Index Name
1	Sitting1Hour
2	Sitting1HourGender
.	
n	

Rules Index Attributes

Index Id	Attribute Key	Value
1	Activity	Sitting
1	Activity duration	1Hour
2	Activity	Sitting
2	Activity duration	1Hour
2	Gender	Male
.		

Indexed Rules

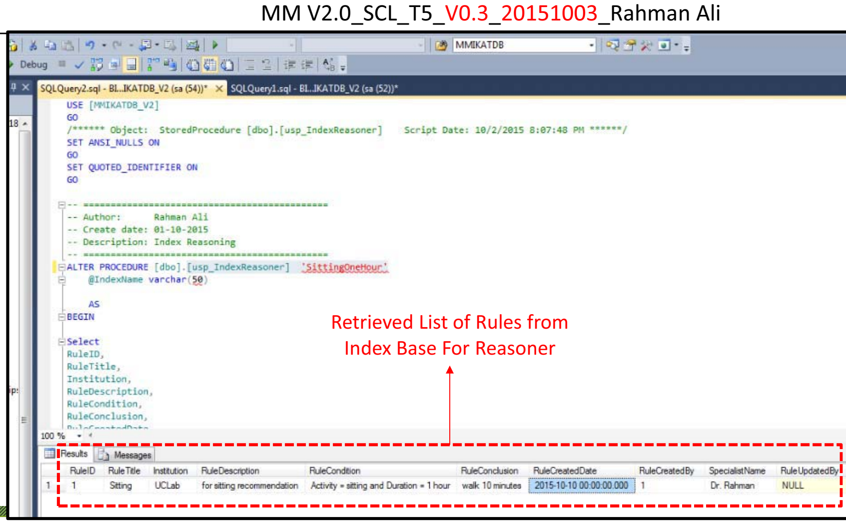
Rule Id	Rule Name	Index Id
1	Sitting1HoursRecommendation.cls	1
2	Sitting1HoursGenderRecommendation.cls	1
.		2
n		.



Store Procedure for Index Matching and Reasoning

Recommendation Builder – Week 2 Progress

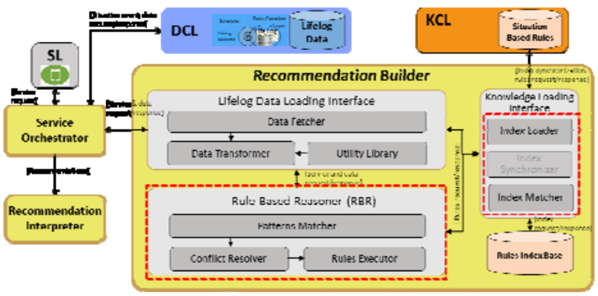
MM V2.0_SCL_T5_V0.3_20151003_Rahman Ali



	RuleID	RuleTitle	Institution	RuleDescription	RuleCondition	RuleConclusion	RuleCreatedDate	RuleCreatedBy	SpecialistName	RuleUpdatedBy
1	1	Sitting	UCLab	for sitting recommendation	Activity = sitting and Duration = 1 hour	walk 10 minutes	2015-10-10 00:00:00.000	1	Dr. Rahman	NULL

Recommendation Builder – Week 3 Future Plan

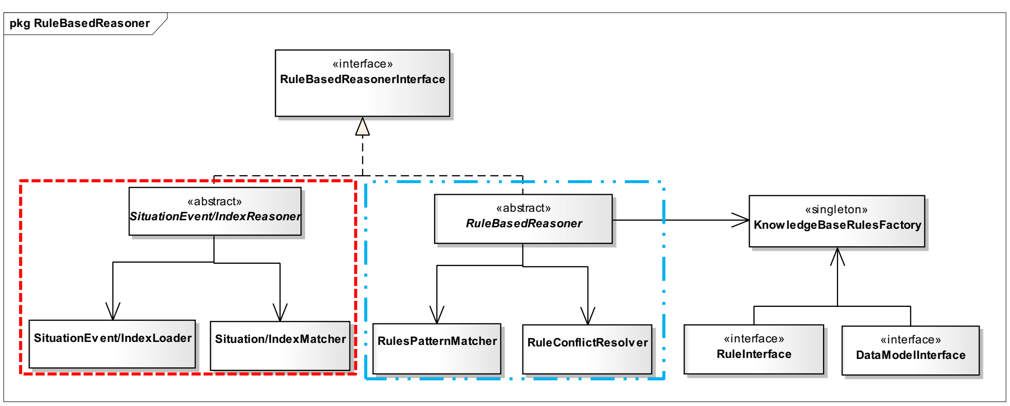
- Future Tasks
  - Completion of Index Reasoner Implementation
  - Skeleton definition for rule-based reasoner



Component	Task	Responsible	19-25Sep	26Sep-2Oct	3-9Oct	10-16Oct
Synchronization of KB	Skeleton definition of the local copy of Indexed Knowledge Base	Rahman, Afzal				
	Service definition for knowledge Base Synchronization					
Index-Based Reasoning	Schema Definition of the Index Knowledge Base	Rahman/Taqdir				
	Implementation of Index-based Reasoning algorithm over Indexed Knowledge Base					
Rule-based Reasoning	Skeleton definition for Rule-based Reasoning Engine	Rahman, Maqbool Hussain				

Recommendation Builder – Week 3 Progress

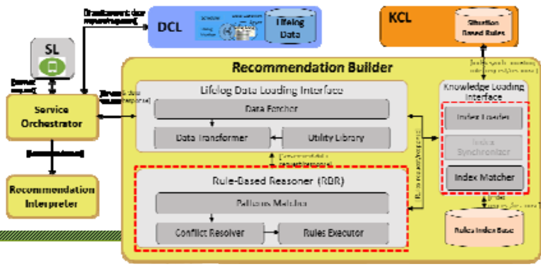
MM V2.0_SCL_T5_V0.3_20151003_Rahman Ali



Recommendation Builder – Week 3 Tasks

MM V2.0_SCL_T5_V0.3_20151016_Rahman Ali

- Tasks Performed
  - Issues resolution of the *Index-based Reasoner* at SCL side
  - Proper locations identification for *Rule Index Base* and *Knowledge Base* (KCL level)
  - Definition of the skeleton for Rule-based Reasoner

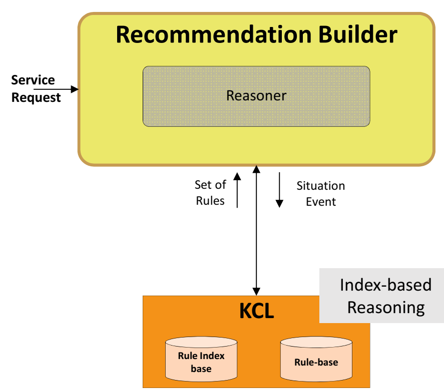


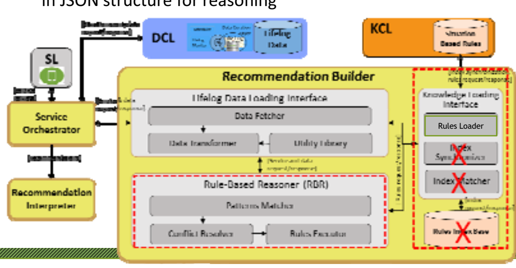
Component	*Task*	*Responsible*	19-25Sep	26Sep-2Oct	3-9Oct	10-16Oct
Synchronization of KB	Skeleton definition of the local copy of Indexed Knowledge Base	Rahman, Afzal				
	Service definition for knowledge Base Synchronization					
Index-Based Reasoning	Schema Definition of the Index Knowledge Base	Rahman/Taqdir				
	Implementation of Index-based Reasoning algorithm over Indexed Knowledge Base					
Rule-based Reasoning	Skeleton definition for Rule-based Reasoning Engine	Rahman, Maqbool Hussain				

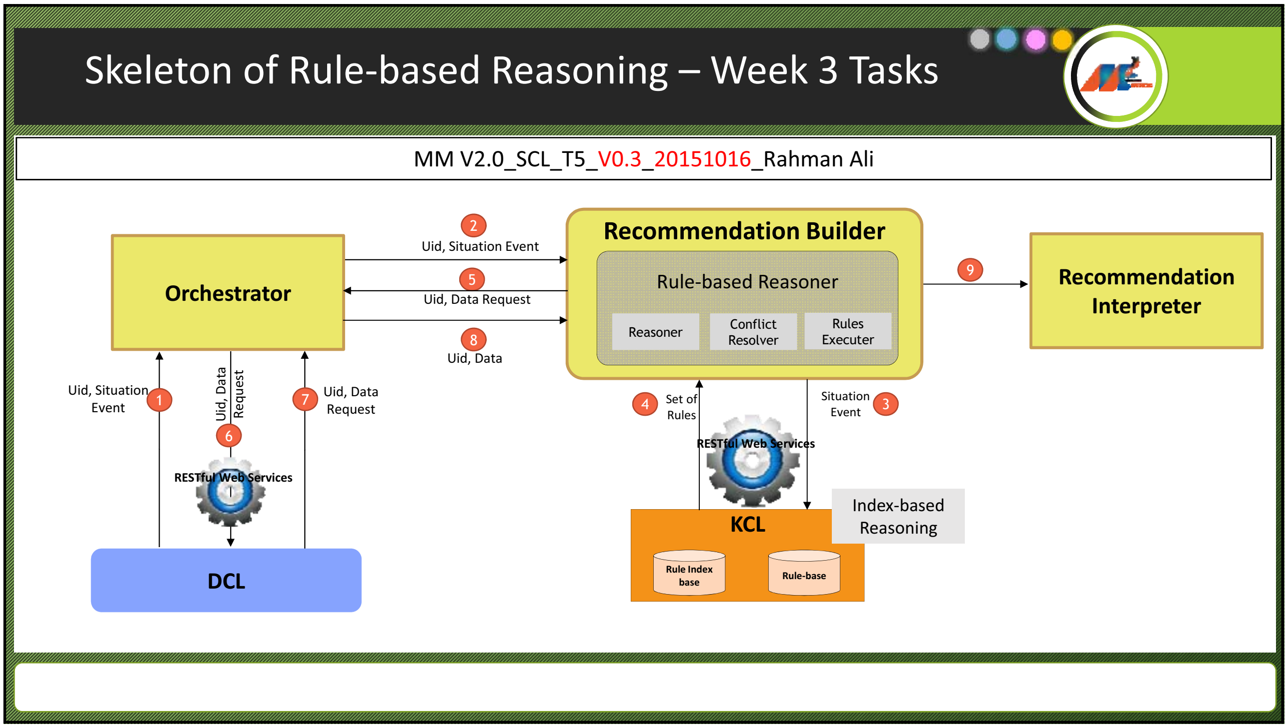
Index-based Reasoning (at KCL Side) – Week 3 Tasks

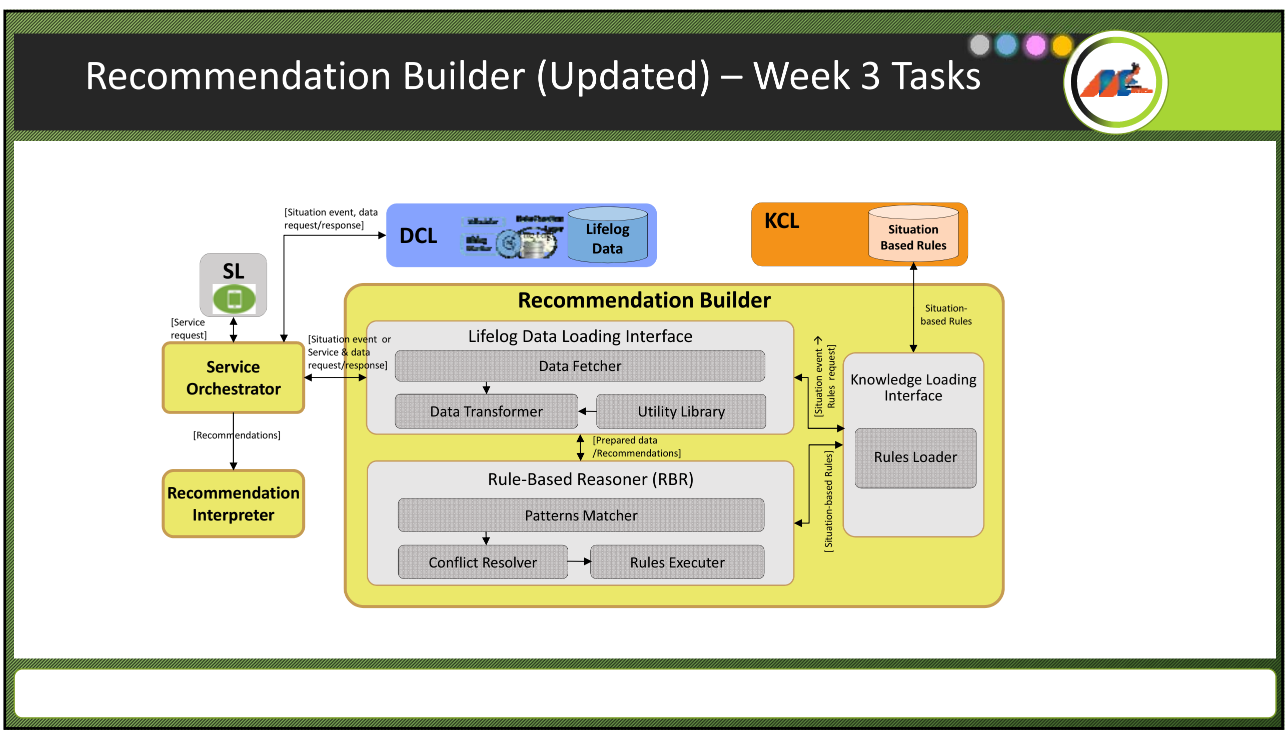
MM V2.0_SCL_T5_V0.3_20151016_Rahman Ali

- Index-based Reasoning at SCL level
  - If Rule Index Base is at the SCL side, it causes maintenance issue while updating the Rule Index Base.
  - Requires a synchronization mechanism for updation of the Rule Index Base
- Index-based Reasoning at KCL level
  - Rule Index-base and Rules Base both should reside at KCL side
  - SCL will first make a request form KCL for the rules , using the Situation Event
  - KCL will perform Index-based reasoning and will forward the set of rules in JSON structure for reasoning






Skeleton of Rule-based Reasoning – Week 3 Tasks
MM V2.0_SCL_T5_V0.3_20151016_Rahman Ali
Orchestrator
Recommendation Builder
Rule-based Reasoner
Reasoner
Conflict Resolver
Rules Executer
Recommendation Interpreter
2 Uid, Situation Event
5 Uid, Data Request
8 Uid, Data
9
1 Uid, Situation Event
6 Uid, Data Request
7 Uid, Data Request
RESTful Web Services
DCL
4 Set of Rules
3 Situation Event
RESTful Web Services
KCL
Rule Index base
Rule-base
Index-based Reasoning


Recommendation Builder (Updated) – Week 3 Tasks
[Situation event, data request/response]
DCL
Lifelog Data
KCL
Situation Based Rules
SL
[Service request]
Service Orchestrator
[Situation event or Service & data request/response]
[Recommendations]
Recommendation Interpreter
Recommendation Builder
Lifelog Data Loading Interface
Data Fetcher
Data Transformer
Utility Library
[Prepared data /Recommendations]
Rule-Based Reasoner (RBR)
Patterns Matcher
Conflict Resolver
Rules Executer
[Situation event → Rules request]
[Situation-based Rules]
Situation-based Rules
Knowledge Loading Interface
Rules Loader

Recommendation Builder – Week 4 Future Plan

MM V2.0_SCL_T5_V0.3_20151016_Rahman Ali

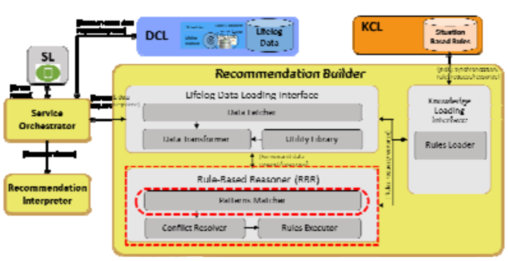
- Future Tasks
  - Implementation of the rule-based reasoning framework
  - Implementation of the rule-based reasoning algorithm with forward-chaining

Component	Task	Responsible	36Sep-2Oct	3-9Oct	10-16Oct	16-23Oct
Synchronization of KB	Skeleton definition of the local copy of Indexed Knowledge Base	Rahman, Afzal				
	Service definition for knowledge Base Synchronization					
Index-Based Reasoning	Schema Definition of the Index Knowledge Base	Rahman/Taqdir				
	Implementation of Index-based Reasoning algorithm over Indexed Knowledge Base					
Rule-based Reasoning	Skeleton definition for Rule-based Reasoning Engine	Rahman, Maqbool Hussain				
	Implementation of Rule-based Reasoning algorithm with forward-chaining mechanism					
	Implementation of Maximum Specificity conflict resolution algorithm for Final Rule (s) Identification					
	Implementation of the Rule-based Reasoning Execution Engine for Recommendation Generation and Testing					

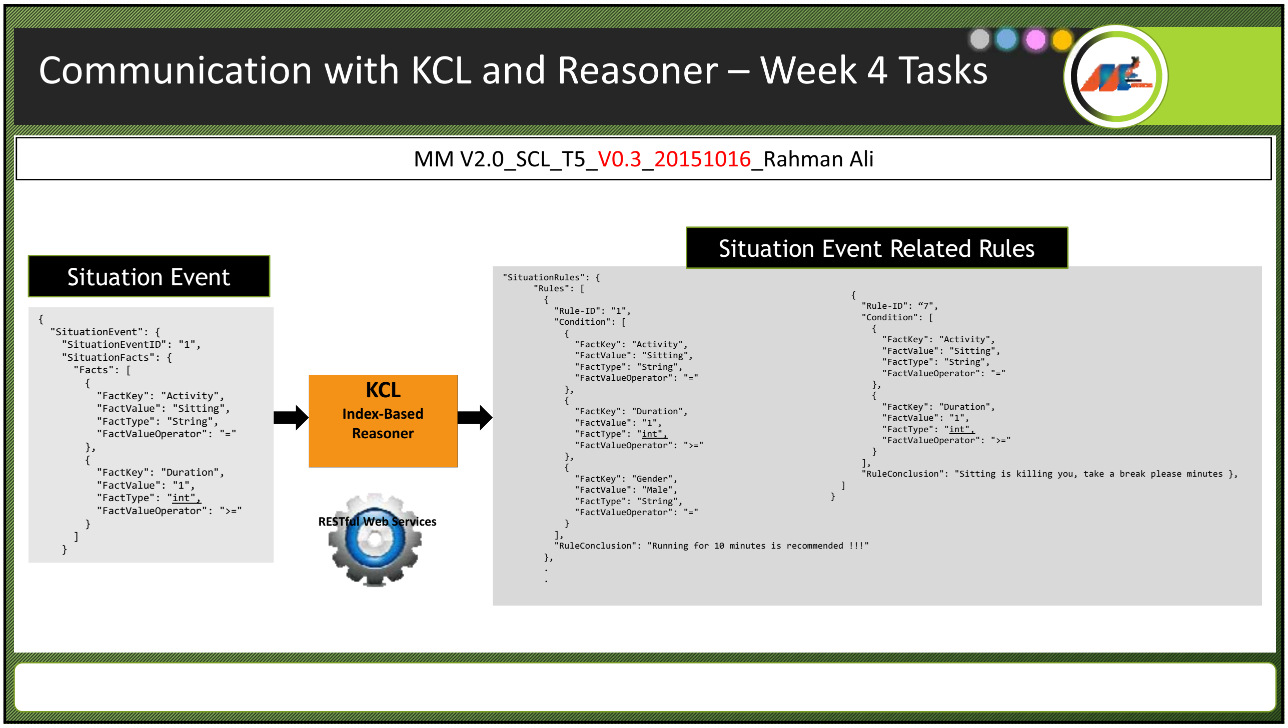
Recommendation Builder – Week 4 Tasks

MM V2.0_SCL_T5_V0.3_20151016_Rahman Ali

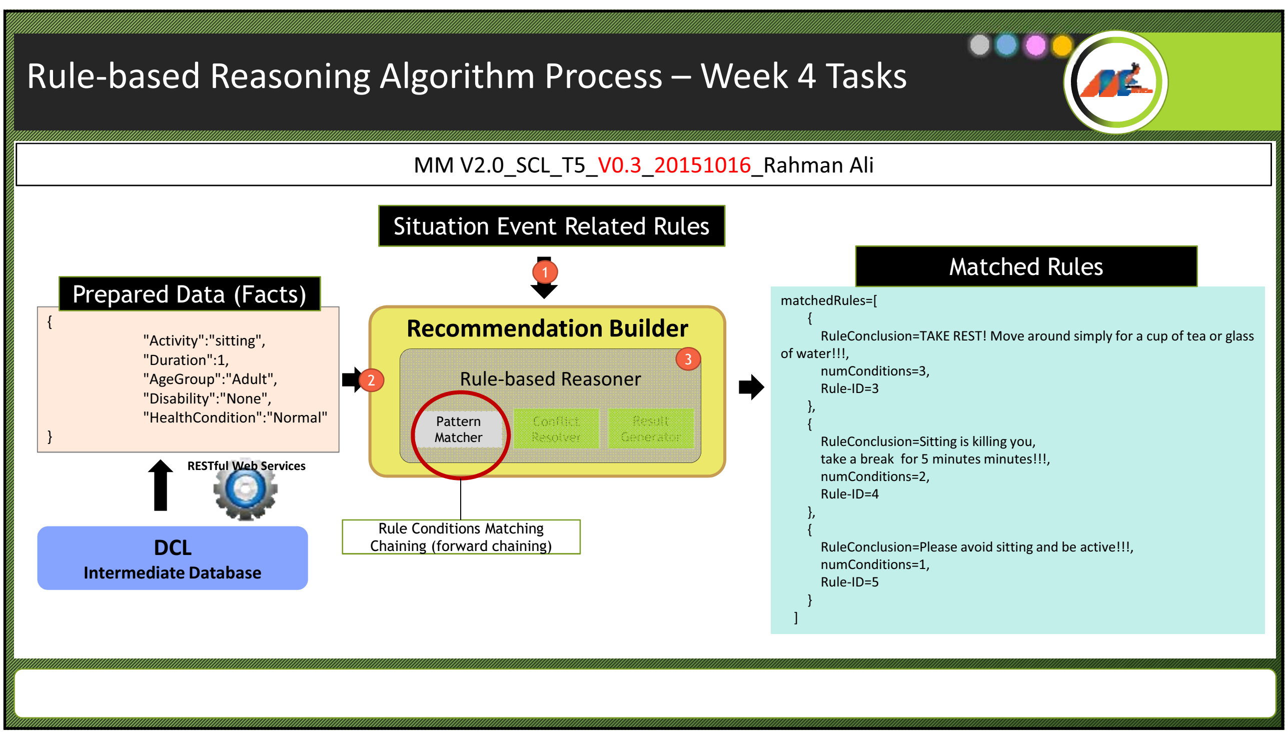
- Tasks Performed
  - Structured definition for the *Situation Event* request and *Situation Rules* with KCL
  - Implementation of the rule-based reasoning algorithm with forward-chaining mechanism



Component	Task	Responsible	36Sep-2Oct	3-9Oct	10-16Oct	16-23Oct
Synchronization of KB	Skeleton definition of the local copy of Indexed Knowledge Base	Rahman, Afzal				
	Service definition for knowledge Base Synchronization					
Index-Based Reasoning	Schema Definition of the Index Knowledge Base	Rahman/Taqdir				
	Implementation of Index-based Reasoning algorithm over Indexed Knowledge Base					
Rule-based Reasoning	Skeleton definition for Rule-based Reasoning Engine	Rahman, Maqbool Hussain				
	Implementation of Rule-based Reasoning algorithm with forward-chaining mechanism					
	Implementation of Maximum Specificity conflict resolution algorithm for Final Rule (s) Identification					
	Implementation of the Rule-based Reasoning Execution Engine for Recommendation Generation and Testing					



Communication with KCL and Reasoner – Week 4 Tasks
MM V2.0_SCL_T5_V0.3_20151016_Rahman Ali
Situation Event
{
"SituationEvent": {
"SituationEventID": "1",
"SituationFacts": {
"Facts": [
{
"FactKey": "Activity",
"FactValue": "Sitting",
"FactType": "String",
"FactValueOperator": "="
},
{
"FactKey": "Duration",
"FactValue": "1",
"FactType": "int",
"FactValueOperator": ">="
}
]
}
KCL
Index-Based Reasoner
RESTful Web Services
Situation Event Related Rules
"SituationRules": {
"Rules": [
{
"Rule-ID": "1",
"Condition": [
{
"FactKey": "Activity",
"FactValue": "Sitting",
"FactType": "String",
"FactValueOperator": "="
},
{
"FactKey": "Duration",
"FactValue": "1",
"FactType": "int",
"FactValueOperator": ">="
},
{
"FactKey": "Gender",
"FactValue": "Male",
"FactType": "String",
"FactValueOperator": "="
}
],
"RuleConclusion": "Running for 10 minutes is recommended !!!"
},
.
.
{
"Rule-ID": "7",
"Condition": [
{
"FactKey": "Activity",
"FactValue": "Sitting",
"FactType": "String",
"FactValueOperator": "="
},
{
"FactKey": "Duration",
"FactValue": "1",
"FactType": "int",
"FactValueOperator": ">="
}
],
"RuleConclusion": "Sitting is killing you, take a break please minutes },
]
}



Rule-based Reasoning Algorithm Process – Week 4 Tasks
MM V2.0_SCL_T5_V0.3_20151016_Rahman Ali
Situation Event Related Rules
Prepared Data (Facts)
{
"Activity":"sitting",
"Duration":1,
"AgeGroup":"Adult",
"Disability":"None",
"HealthCondition":"Normal"
}
RESTful Web Services
DCL
Intermediate Database
1
2
3
Recommendation Builder
Rule-based Reasoner
Pattern Matcher
Conflict Resolver
Result Generator
Rule Conditions Matching Chaining (forward chaining)
Matched Rules
matchedRules=[
{
RuleConclusion=TAKE REST! Move around simply for a cup of tea or glass of water!!!,
numConditions=3,
Rule-ID=3
},
{
RuleConclusion=Sitting is killing you,
take a break for 5 minutes minutes!!!,
numConditions=2,
Rule-ID=4
},
{
RuleConclusion=Please avoid sitting and be active!!!,
numConditions=1,
Rule-ID=5
}
]

Rule-based Reasoning – Forward Chaining – Week 4 Tasks

```
RecommendationBuilder
{
        BuildRecommendation (facts, rules)
  {
```

rule-Id	conditions	conclusion

```
        MatchedRules = Reasoner.fireRule(facts, conditions);

            FiredRule = ConflictResolver.resolveConflict(MatchedRules)
  }
}
```

Rules Matching Using Forward Chaining – Week 4 Tasks

MatchedRules = Reasoner.fireRule(conditions, facts);

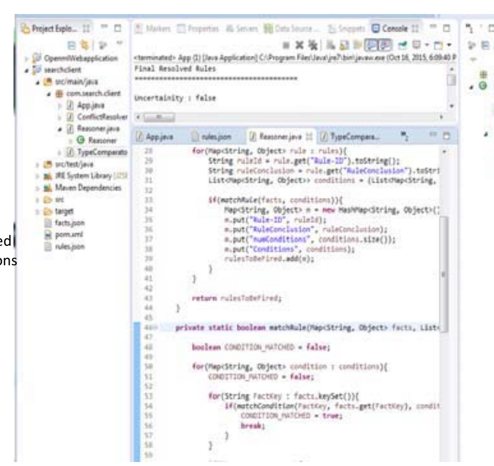
Type	Operator	Key	Value

String	Int	Boolean

String Operators (==, !=)	Integer Operators (==, !=, <, >, <=, >=)	Boolean (=, !=)

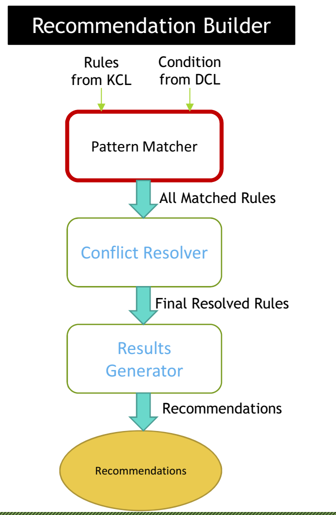
Matching

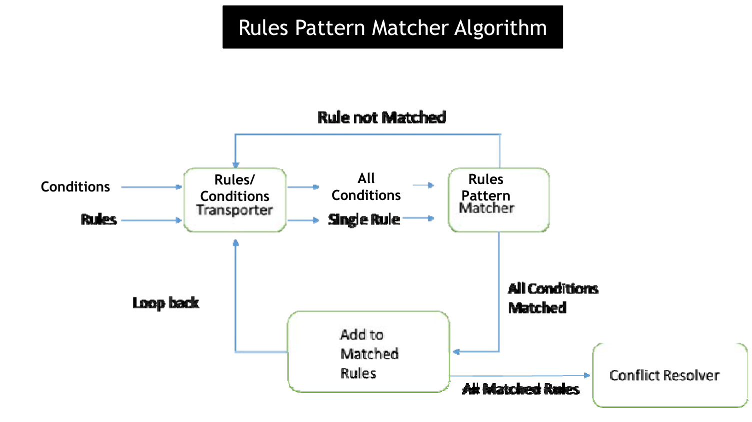
Matched Conditions



Examples:
- If (age>=60)
- If (activity=='sitting')

Recommendation Builder – Dataflow Diagram





Recommendation Builder – Week 5 Future Plan

MM V2.0_SCL_T5_V0.3_20151016_Rahman Ali

- Future Tasks
  - Implementation of Maximum Specificity conflict resolution algorithm for Final Rule (s) Identification
  - Testing rule-based reasoner

Component	*Task*	*Responsible*	10-16Oct	16-23Oct
Rule-based Reasoning	Skeleton definition for Rule-based Reasoning Engine	Rahman, Maqbool Hussain		
	Implementation of Rule-based Reasoning algorithm with forward-chaining mechanism	Rahman		
	Implementation of Maximum Specificity conflict resolution algorithm for Final Rule (s) Identification	Rahman		

Recap of the last 7 Weeks

MM V2.0_SCL_T5_V0.3_20151016_Rahman Ali

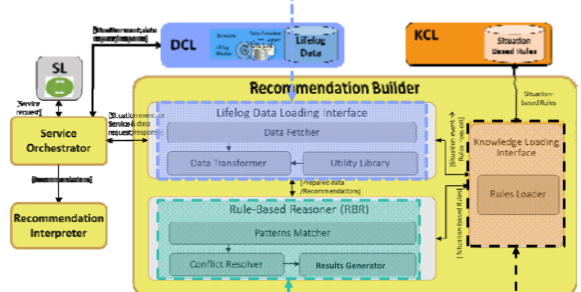
- **Week 1**
  - Development environment setup
    - (tools selection, installation etc.)
  - Design of the Rule-based Reasoning framework
    - (index-based reasoning, rule-based reasoning, class diagram etc.)
- **Week 2**
  - Index-based reasoning
    - (Schema-definition, store procedure-based implementation)
- **Week 3**
  - Issues with Index-based Reasoning
    - (Synchronization and maintenance issues at SCL side)
  - Defining Index-based reasoning at KCL side
    - (Situation-based Reasoning using store procedure)
- **Week 4**
  - Skeleton definition for Rule-based Reasoning Engine
    - Communication pattern (KCL, DCL)
    - Skeleton (pattern matcher, conflict resolver, result generator)
  - Implemented Pattern Matcher with forward-chaining algorithm
- **Week 5**
  - Testing rule-based reasoner
  - Definition of the conflict resolution strategy
  - Implementation of Maximum Specificity conflict resolution algorithm
- **Week 6**
  - Testing reasoner and conflict resolver
  - Data integration
    - (transformation of DCL data → utility library)
- **Week 7**
  - Results Generation
    - Complex Object / Recommendation Packet Creation
- **Week 8**
  - Recommendation Builder Online Demo
    - Pattern matcher – Forward Chaining Algorithm
    - Conflict Resolver – Maximum Specificity Algorithm
    - Results Generator – Complex Object Creation

Recommendation Builder – Overall Concept (Week 5)

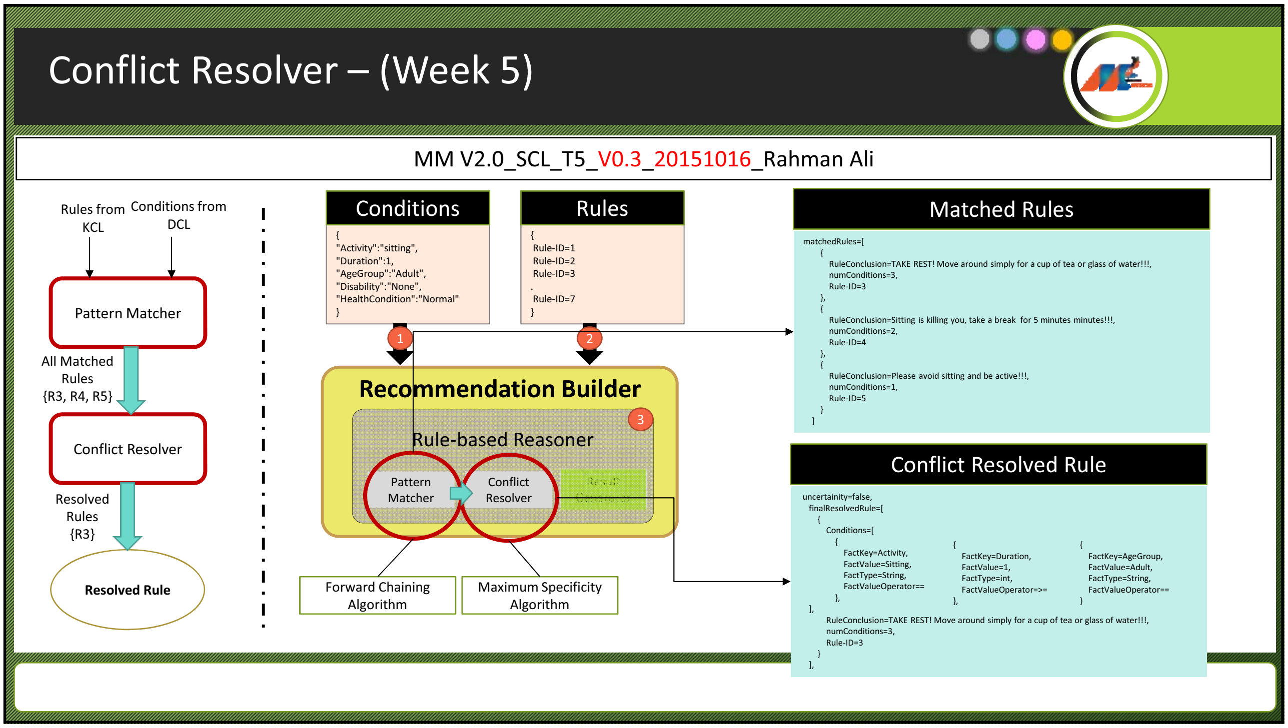
- Data integration
  - Objective
    - DCL data loading for new instance generation
  - Method
    - Restful web service -- data transformation → utility library
- Knowledge integration
  - Objective
    - KCL Knowledge loading
  - Method
    - Restful web service -- JSON
- Reasoning
  - Pattern matcher
    - Objective
      - Matching rules against new instance
    - Method
      - Pattern matching forward chaining algorithm
  - Conflict resolver
    - Objective
      - Selecting final rule from a set of matched rule
    - Method
      - Maximum specificity conflict resolving algorithm
  - Results generator
    - Objective
      - Communicating the results of fired rule with Interpreter
    - Method
      - Complex object definition – recommendation packet

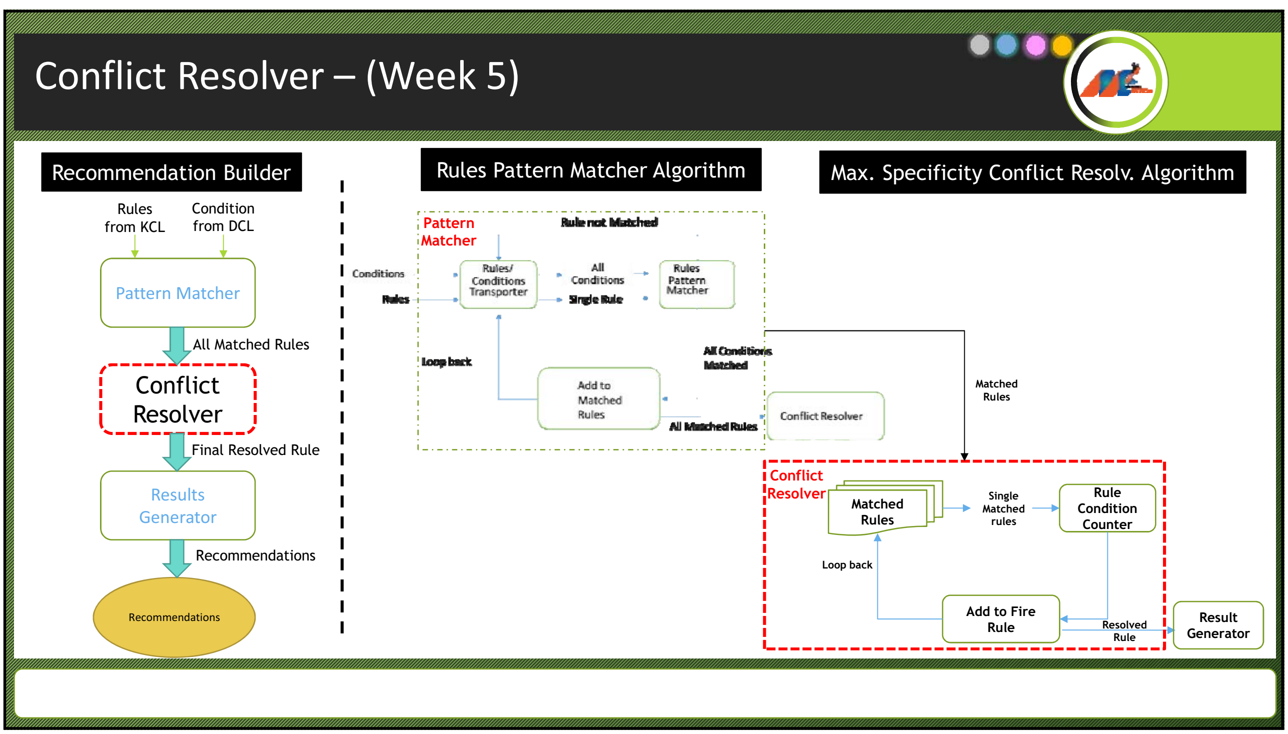
Loading and Preparing Data

Loading Knowledge Rules



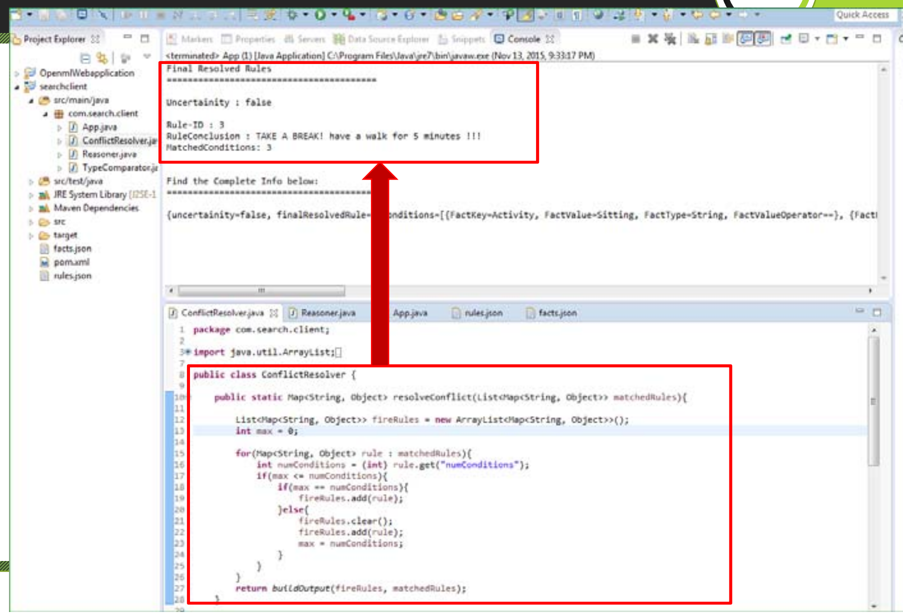
Rules matching, conflict resolving and recommendations generation


Conflict Resolver – (Week 5)
MM V2.0_SCL_T5_V0.3_20151016_Rahman Ali
Rules from KCL
Conditions from DCL
Pattern Matcher
All Matched Rules {R3, R4, R5}
Conflict Resolver
Resolved Rules {R3}
Resolved Rule
Conditions
{ "Activity":"sitting", "Duration":1, "AgeGroup":"Adult", "Disability":"None", "HealthCondition":"Normal" }
Rules
{ Rule-ID=1 Rule-ID=2 Rule-ID=3 . Rule-ID=7 }
1
2
3
Recommendation Builder
Rule-based Reasoner
Pattern Matcher
Conflict Resolver
Result Generator
Forward Chaining Algorithm
Maximum Specificity Algorithm
Matched Rules
matchedRules=[{ RuleConclusion=TAKE REST! Move around simply for a cup of tea or glass of water!!!, numConditions=3, Rule-ID=3 }, { RuleConclusion=Sitting is killing you, take a break for 5 minutes minutes!!!, numConditions=2, Rule-ID=4 }, { RuleConclusion=Please avoid sitting and be active!!!, numConditions=1, Rule-ID=5 }]
Conflict Resolved Rule
uncertainity=false, finalResolvedRule=[{ Conditions=[{ FactKey=Activity, FactValue=Sitting, FactType=String, FactValueOperator== }, { FactKey=Duration, FactValue=1, FactType=int, FactValueOperator=>= }, { FactKey=AgeGroup, FactValue=Adult, FactType=String, FactValueOperator== }], RuleConclusion=TAKE REST! Move around simply for a cup of tea or glass of water!!!, numConditions=3, Rule-ID=3 }],


Conflict Resolver – (Week 5)
Recommendation Builder
Rules from KCL
Condition from DCL
Pattern Matcher
All Matched Rules
Conflict Resolver
Final Resolved Rule
Results Generator
Recommendations
Recommendations
Rules Pattern Matcher Algorithm
Pattern Matcher
Rule not Matched
Conditions
Rules
Rules/ Conditions Transporter
All Conditions
Single Rule
Rules Pattern Matcher
Loop back
All Conditions Matched
Add to Matched Rules
All Matched Rules
Conflict Resolver
Max. Specificity Conflict Resolv. Algorithm
Matched Rules
Conflict Resolver
Matched Rules
Single Matched rules
Rule Condition Counter
Loop back
Add to Fire Rule
Resolved Rule
Result Generator

Conflict Resolver – (Week 5)

Rules Conflict Resolution using Maximum Specificity Algorithm



Data Transformation & Refinement – (Week 6)

MM V2.0_SCL_T5_V0.3_20151016_Rahman Ali

Example Rule:

If Activity = Sitting, ActivityDuration >=1h, Gender = Male, AgeGroup = Adult, WeightStatus = Overweight → **"RuleConclusion": "TAKE REST! Please have walk for 10 minutes outdoor !!!"**

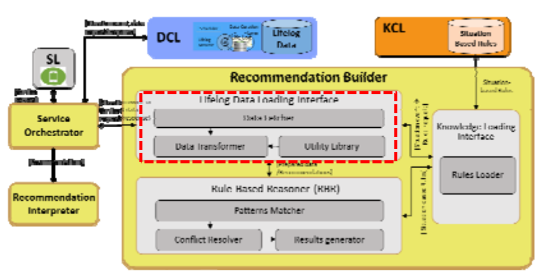
Data Fetcher → **Data Transformer** ← **Utility Library**

<u>Loaded Data from Lifelog</u>

- Gender: Male
- DoB: 1985
- Height: 6ft
- Weight: 89Kg
- Activity: Sitting
- ActivityDuration: >=1h

Transformation

- **Rule Parsing Only Condition Part**
  - **(Activity, Gender, AgeGroup, WeightStatus)**
- **Conformance check**
  - (AgeGroup, WeightStatus)
- **Refinement/computation**
  - Dob → **AgeGroup** (transformation)
  - Height, Weight→ **WeightStatus** (transformation)
    - Height, Weight → BMI
    - BMI → WeightStatus



경희대학교 KYUNG HEE UNIVERSITY

Results Generation – (Week 7)

MM V2.0_SCL_T5_V0.3_20151016_Rahman Ali

Rules from KCL
Condition from DCL

Pattern Matcher

All Matched Rules

Conflict Resolver

Final Resolved Rule

Results Generator

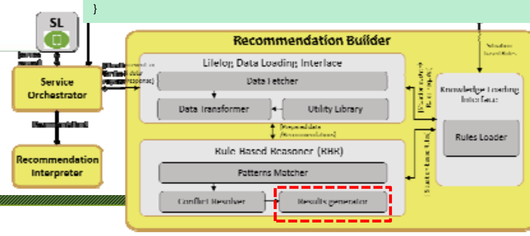
Recommendations

Recommendations

Recommendation Interpreter

{
ruleConclusion: "TAKE A BREAK! Bend your waist forward at least 4 times to stretch your body! Your heart will thank you."
conclusionList: [
2]0: {
conclusionKey: "Activity"
conclusionValue: "Stretching"
conclusionOperator: "="
ruleID: 4
conclusionID: 8
}
-
1: {
conclusionKey: "Times"
conclusionValue: "4"
conclusionOperator: "="
ruleID: 4
conclusionID: 9
}

Recommendation will be Shared in Complex Object Form



경희대학교
KYUNG HEE UNIVERSITY

Scope of Recommendation Builder Demo – (Week 8)

MM V2.0_SCL_T5_V0.3_20151016_Rahman Ali

- Purpose: to show how reasoned generates recommendations in different scenarios

Recommendation Builder → Recommendations → Recommendation Interpreter

KCL
DCL

Rules from KCL
Conditions from DCL

Pattern Matcher

All Matched Rules

Conflict Resolver

Final Resolved Rule

Results Generator

Recommendations

Recommendations

Recommendations in Complex Object – Recommendation Packet

Recommendation Interpreter

Scope of Recommendation Builder Demo (Reasoner)

Recommendation Builder Demo Scenarios – (Week 8)

MM V2.0_SCL_T5_V0.3_20151016_Rahman Ali

- **Scenario 1**
  - Pattern Matcher – Forward Chaining Algorithm
    - Sub-scenario 1
      - Only one rule matched
    - Sub-scenario 2
      - More than 1 rules matched
    - Sub-scenario 3
      - No rule matched

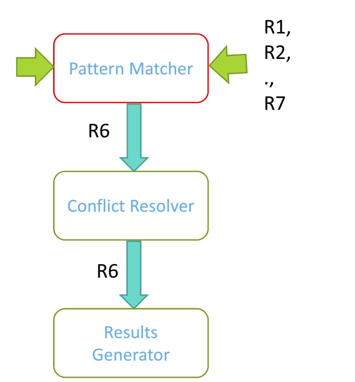
- **Scenario 2**
  - Conflict Resolver – Maximum Specificity Algorithm
    - Sub-scenario 1
      - More than 1 rules matched → Resolved to → Single rule
    - Sub-scenario 2 - Uncertainty
      - More than 1 rules matched → Resolved to → More than 1 rules

Recommendation Builder Demo Scenarios – (Week 8)

MM V2.0_SCL_T5_V0.3_20151016_Rahman Ali

- **Scenario 1**
  - Pattern Matcher – Forward Chaining Algorithm
    - Sub-scenario 1
      - Only one rule matched
    - Sub-scenario 2
      - More than 1 rules matched
    - Sub-scenario 3
      - No rule matched

{
"Activity":"LyingDown",
"Duration":6,
"Chronic":"Yes"
}

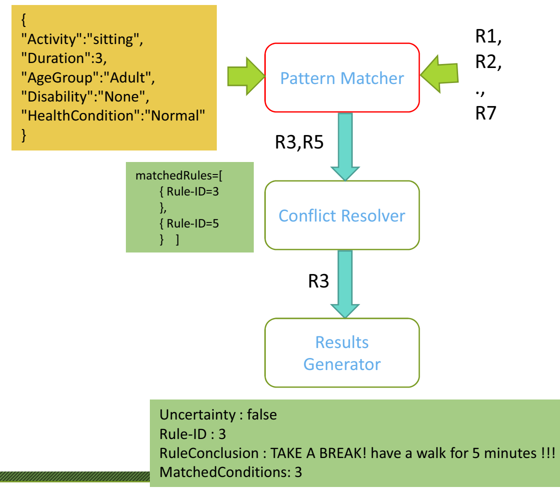


Rule-ID : 6
RuleConclusion : Walking for atleast 10 minutes after prolonged lying down adds more to your health !!!
MatchedConditions: 3

Recommendation Builder Demo Scenarios – (Week 8)

MM V2.0_SCL_T5_V0.3_20151016_Rahman Ali

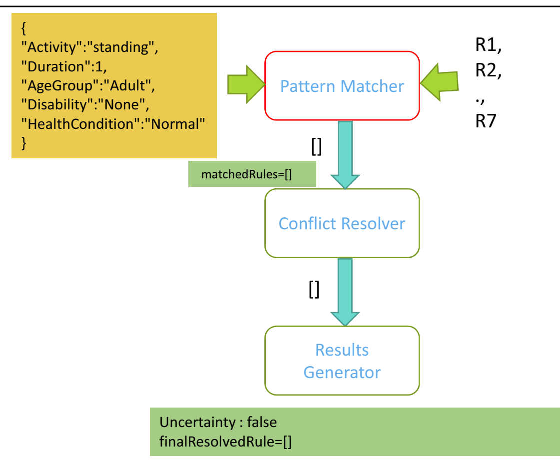
- **Scenario 1**
  - Pattern Matcher – Forward Chaining Algorithm
    - Sub-scenario 1
      - Only one rule matched
    - Sub-scenario 2
      - More than 1 rules matched
    - Sub-scenario 3
      - No rule matched



Recommendation Builder Demo Scenarios – (Week 8)

MM V2.0_SCL_T5_V0.3_20151016_Rahman Ali

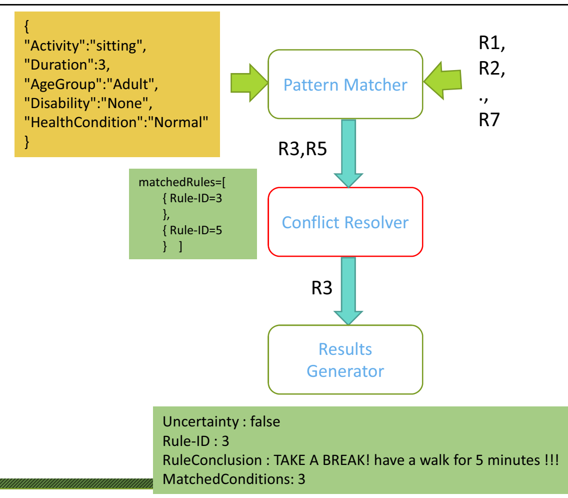
- **Scenario 1**
  - Pattern Matcher – Forward Chaining Algorithm
    - Sub-scenario 1
      - Only one rule matched
    - Sub-scenario 2
      - More than 1 rules matched
    - Sub-scenario 3
      - No rule matched



Recommendation Builder Demo Scenarios – (Week 8)

MM V2.0_SCL_T5_V0.3_20151016_Rahman Ali

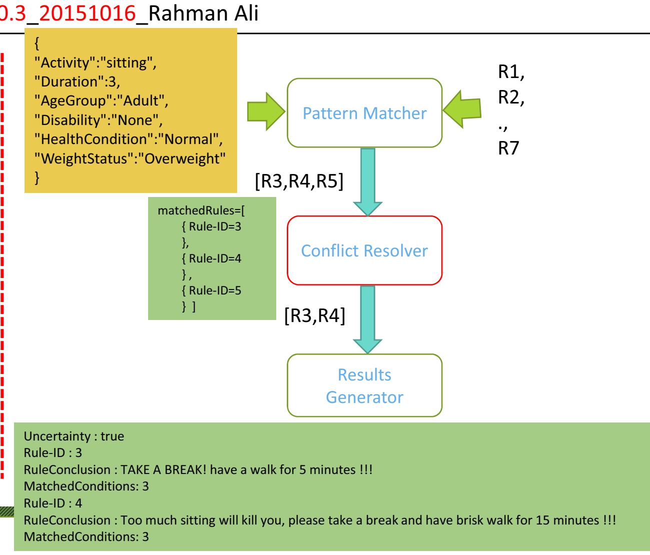
- **Scenario 2**
  - Conflict Resolver – Maximum Specificity Algorithm
    - Sub-scenario 1
      - More than 1 rules matched → Resolved to → Single rule
    - Sub-scenario 2 - Uncertainty
      - More than 1 rules matched → Resolved to → More than 1 rules



Recommendation Builder Demo Scenarios – (Week 8)

MM V2.0_SCL_T5_V0.3_20151016_Rahman Ali

- **Scenario 2**
  - Conflict Resolver – Maximum Specificity Algorithm
    - Sub-scenario 1
      - More than 1 rules matched → Resolved to → Single rule
    - Sub-scenario 2 - Uncertainty
      - More than 1 rules matched → Resolved to → More than 1 rules

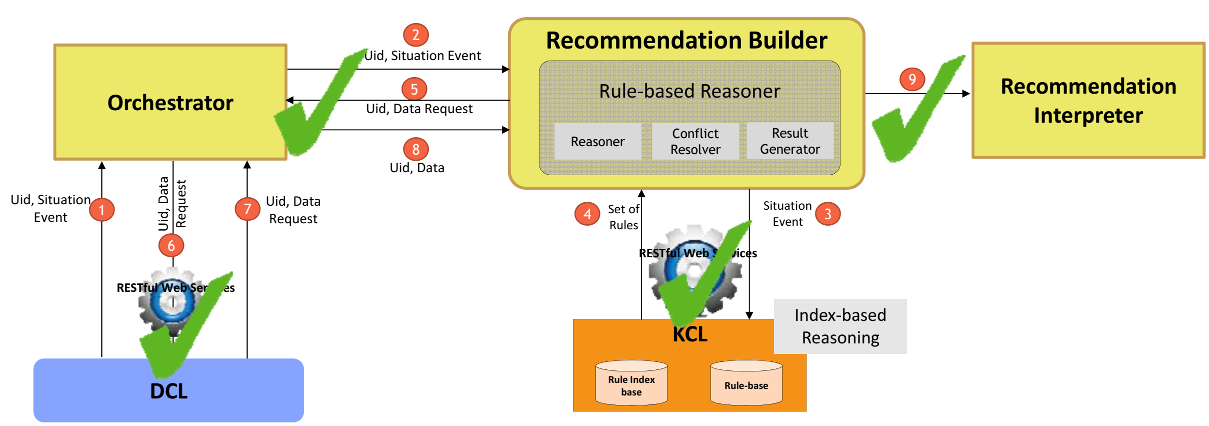


Future Work (Week 9-10)

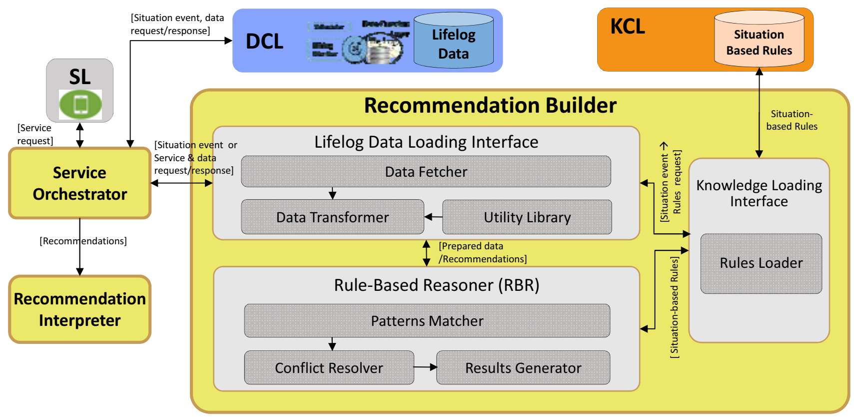
- Intra layer integration
  - Recommendation builder → Recommendation Interpreter
- Inter layer integration
  - Recommendation builder → KCL
  - Recommendation builder → DCL
- Data Integration and transformation
  - Utility library

Integration – Week 9 and 10 Tasks

MM V2.0_SCL_T5_V0.5_20151116_Rahman Ali



Appendix A: Recommendation Builder



Appendix B: Reasoner Input Cases Scenarios

MM V2.0_SCL_T5_V0.3_20151016_Rahman Ali

```
{
"Activity":"Standing",
"Duration":1,
"Chronic":"Yes"
}
```

→ No Match

```
{
"Activity":"LyingDown",
"Duration":6,
"Chronic":"Yes"
}
```

→ R6 ↓ R6

```
{
"Activity":"sitting",
"Duration":3,
"AgeGroup":"Adult",
"Disability":"None",
"HealthCondition":"Normal",
"WeightStatus":"Overweight"
}
```

R3, R4, R5 ↓ R3,R4

```
{
"Activity":"sitting",
"Duration":3,
"AgeGroup":"Adult",
"Disability":"None",
"HealthCondition":"Normal"
}
```

→ R3, R5 ↓ R3

Appendix C: Conflict Resolution Strategies

- 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

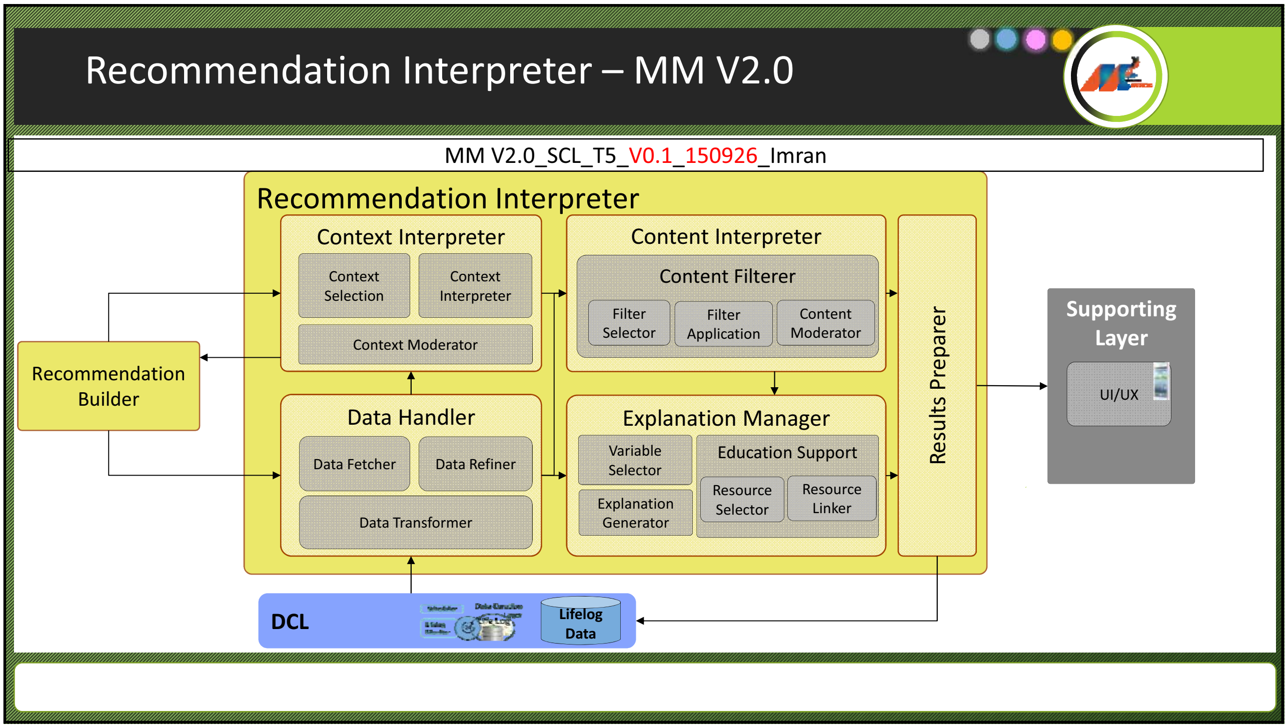
- Fire All Selected Rules
  - Executes all the matched rules
- Context Limiting
  - partition rule base into disjoint subsets
  - doing this we can have subsets and we may also have preconditions
- Execution Time
  - Chose the one with faster execution
- Physically ordering of rules
  - hard to add rules based on their physical order
- Random
  - Randomly pick one rule for execution

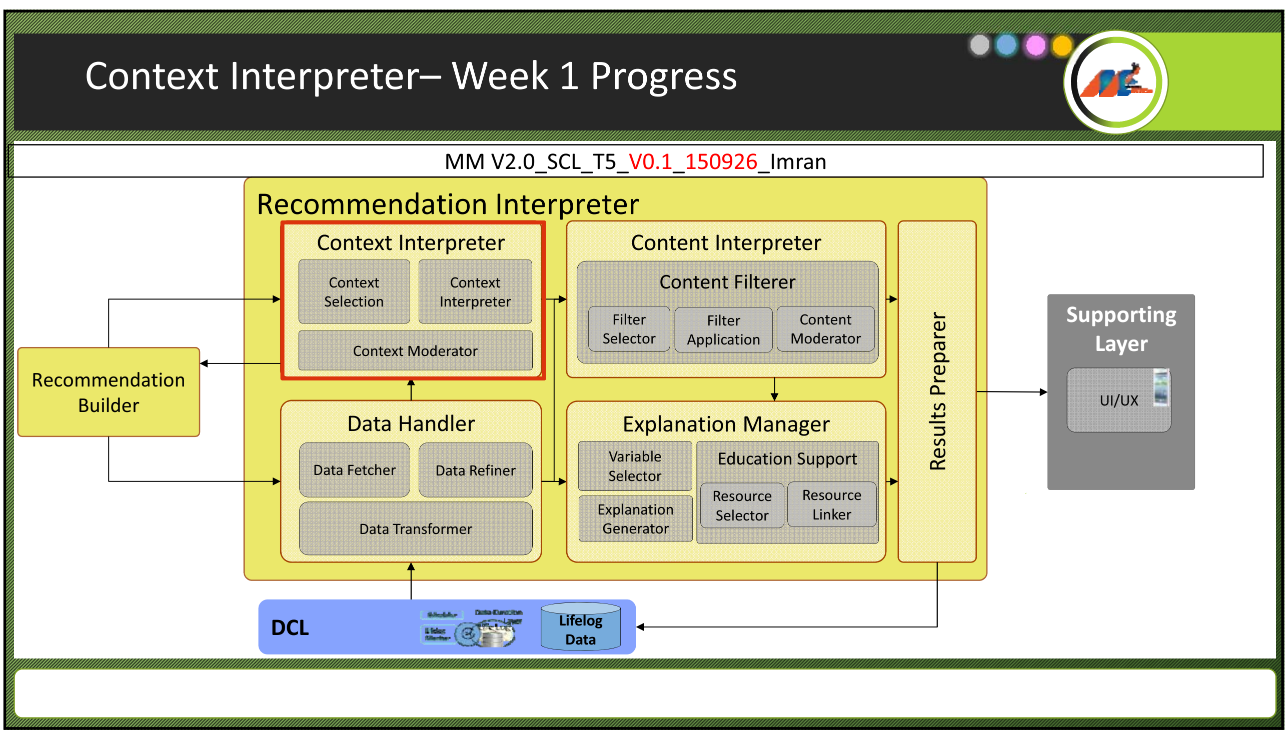
경희대학교 KYUNG HEE UNIVERSITY

Recommendation Interpreter

Imran Ali

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


Recommendation Interpreter – MM V2.0
MM V2.0_SCL_T5_V0.1_150926_Imran
Recommendation Interpreter
Context Interpreter
Context Selection
Context Interpreter
Context Moderator
Content Interpreter
Content Filterer
Filter Selector
Filter Application
Content Moderator
Data Handler
Data Fetcher
Data Refiner
Data Transformer
Explanation Manager
Variable Selector
Education Support
Explanation Generator
Resource Selector
Resource Linker
Results Preparer
Recommendation Builder
Supporting Layer
UI/UX
DCL
Lifelog Data


Context Interpreter– Week 1 Progress
MM V2.0_SCL_T5_V0.1_150926_Imran
Recommendation Interpreter
Context Interpreter
Context Selection
Context Interpreter
Context Moderator
Content Interpreter
Content Filterer
Filter Selector
Filter Application
Content Moderator
Data Handler
Data Fetcher
Data Refiner
Data Transformer
Explanation Manager
Variable Selector
Education Support
Explanation Generator
Resource Selector
Resource Linker
Results Preparer
Recommendation Builder
Supporting Layer
UI/UX
DCL
Lifelog Data

Context Interpreter– Week 1 Progress

MM V2.0_SCL_T5_V0.1_150926_Imran

- Required Software Installation
  - Eclipse EE + GlassFish Server + Weka
  - Understanding Implementation of Mining Minds 1.5 (SCL)
- Prototype for Context Interpretation (Simulation)
  - Dataset generation
  - Data annotation (label assignments)

Component	Task	19-25Sep	26Sep-2Oct	3-9Oct	10-16Oct
			OCTOBER		
Context Interpretation	Dataset Genenartion for Context Interpreter				
	Data Annotations (Label assignments)				
	Model Selection and Learning (Offline Experiment)				
	Model Implementation and Testing				
Content Filteration and Alteration	Finding Alternate Recommendations on the basis of different context (MET Index)				
	Criteria definitions for filteration (preferences/special condition)				

Context Interpreter– Week 2 Progress

MM V2.0_SCL_T5_V0.2_151003_Imran

What I Did:

1. Finalized Dataset and label annotation
2. Evaluated DT using Weka
3. Evaluation of Context Selection (prototype)
4. Alternate Recommendation Table (for weather and location)
5. Filtration Criteria (for special condition)

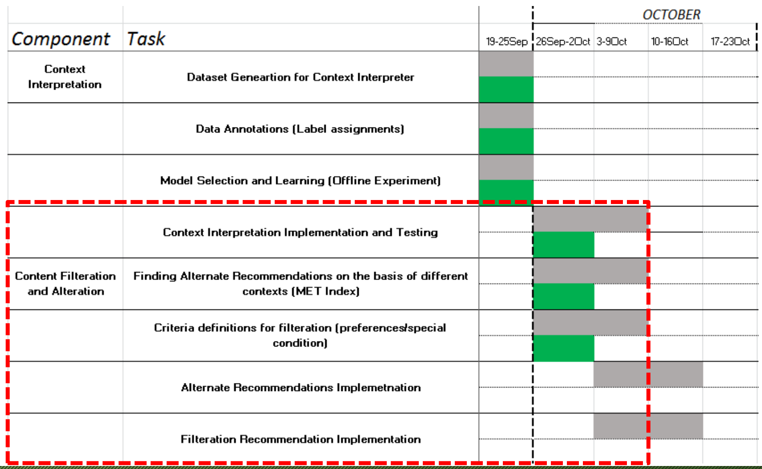
Component	Task	19-25Sep	26Sep-2Oct	3-9Oct	10-16Oct
			OCTOBER		
Context Interpretation	Dataset Genenartion for Context Interpreter				
	Data Annotations (Label assignments)				
	Model Selection and Learning (Offline Experiment)				
	Context Interpretation Implementation and Testing				
Content Filteration and Alteration	Finding Alternate Recommendations on the basis of different contexts (MET Index)				
	Criteria definitions for filteration (preferences/special condition)				

Context Interpreter– Week 3 Future Plan

MM V2.0_SCL_T5_V0.2_151003_Imran

What I will do:

1. Finish Implementation of Context Interpreter
2. Complete Design for Alternative Recommendation
3. Complete Design for Recommendation Filtration
4. Prototype Alternative Recommendation
5. Prototype Recommendation Filtration

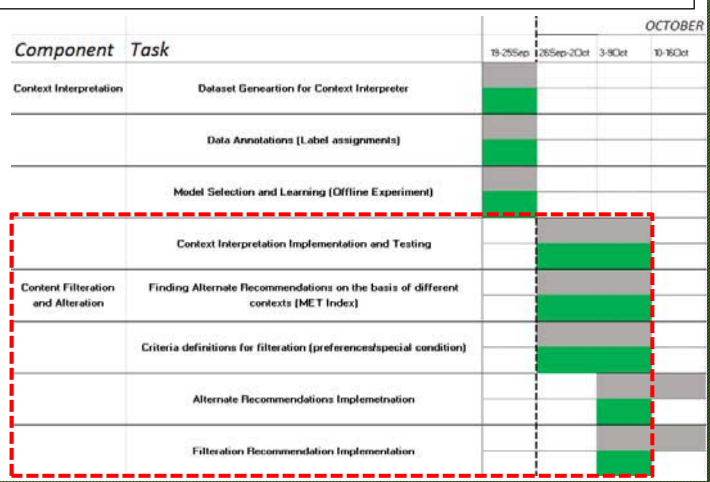


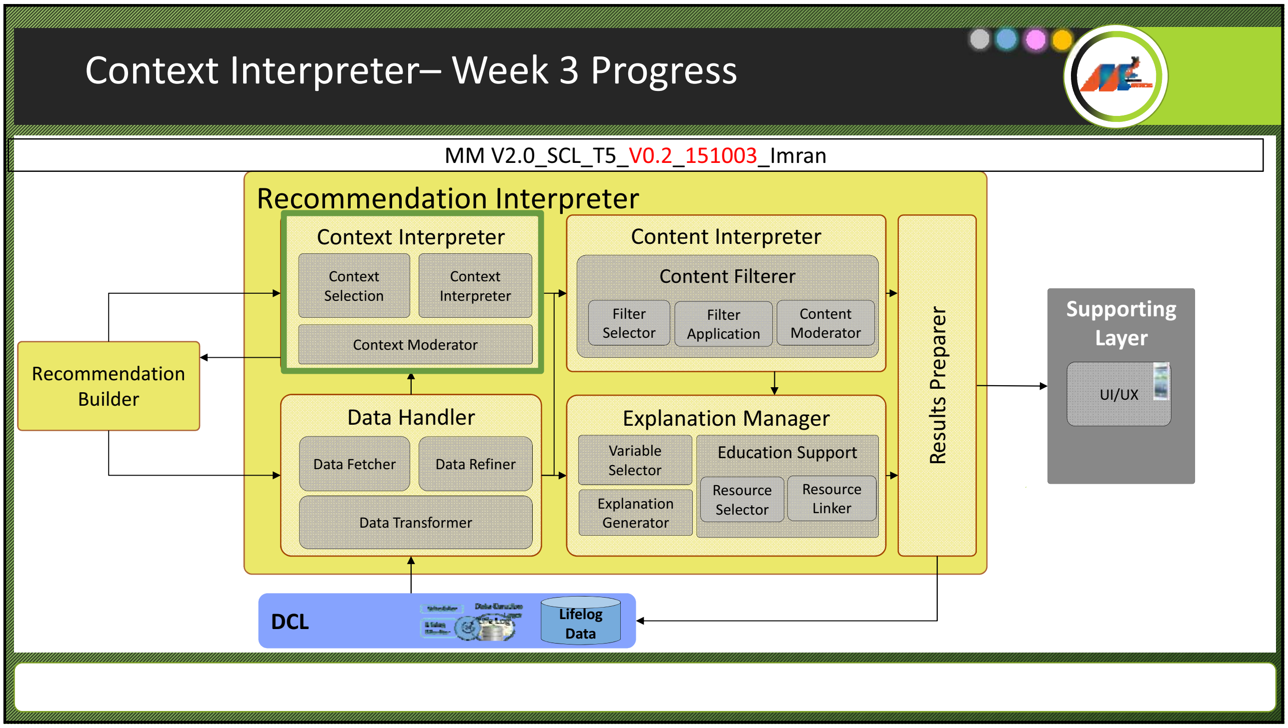
Context Interpreter– Week 3 Progress

MM2.0_SCL_T5_V0.3_20151021_Imran

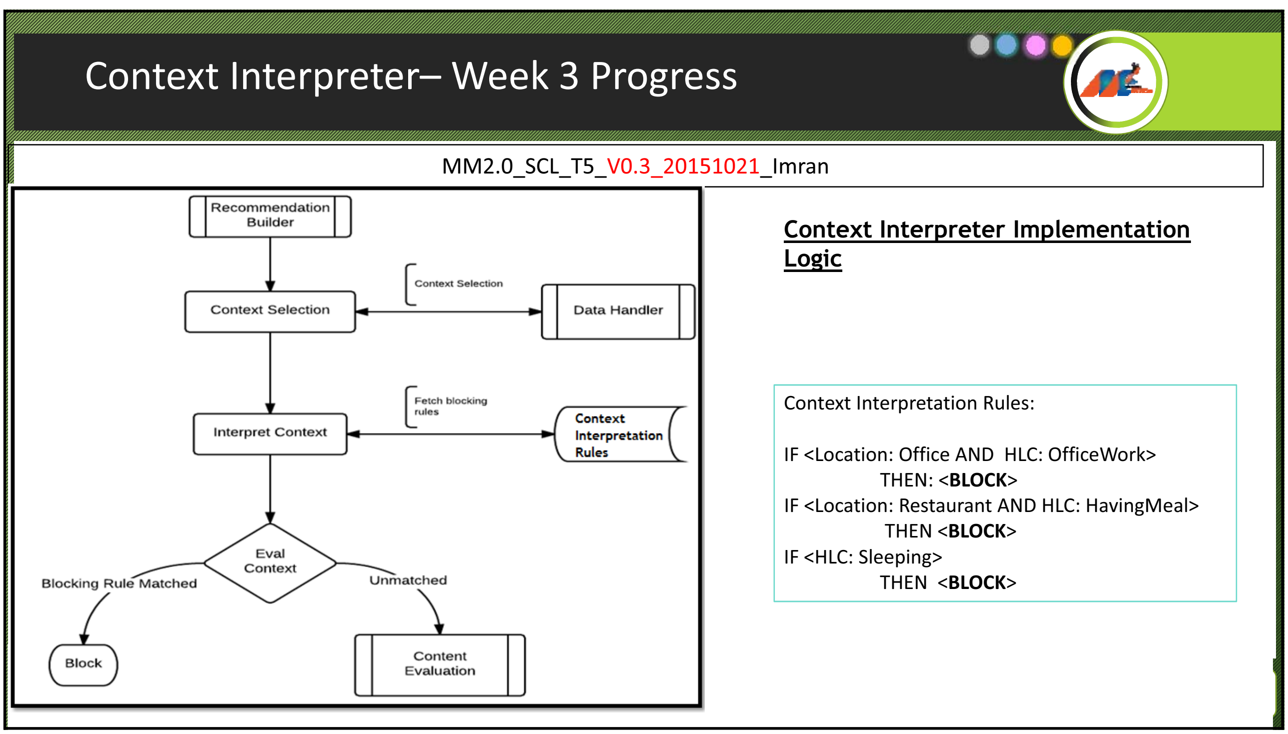
What I Did:

0. Redesigned Dataset
1. Implemented Context Interpreter
2. Designed Alternative Recommendation
3. Designed Recommendation Filtration
4. Prototyped Alternative Recommendation
5. Prototype Recommendation Filtration





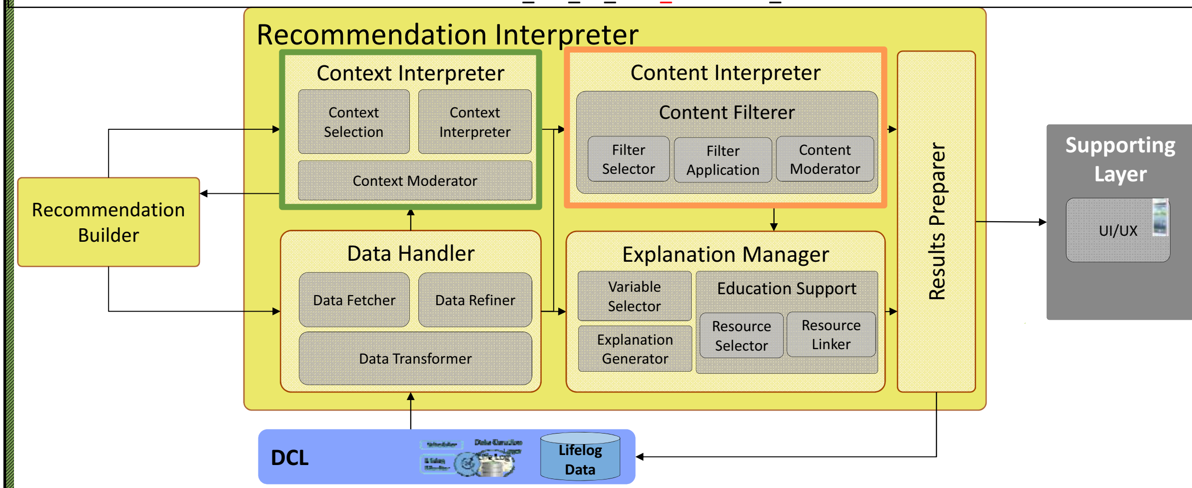
Context Interpreter– Week 3 Progress
MM V2.0_SCL_T5_V0.2_151003_Imran
Recommendation Interpreter
Context Interpreter
Context Selection
Context Interpreter
Context Moderator
Content Interpreter
Content Filterer
Filter Selector
Filter Application
Content Moderator
Results Preparer
Supporting Layer
UI/UX
Recommendation Builder
Data Handler
Data Fetcher
Data Refiner
Data Transformer
Explanation Manager
Variable Selector
Explanation Generator
Education Support
Resource Selector
Resource Linker
DCL
Lifelog Data



Context Interpreter– Week 3 Progress
MM2.0_SCL_T5_V0.3_20151021_Imran
Recommendation Builder
Context Selection
Data Handler
Context Selection
Fetch blocking rules
Interpret Context
Context Interpretation Rules
Eval Context
Blocking Rule Matched
Unmatched
Block
Content Evaluation
Context Interpreter Implementation Logic
Context Interpretation Rules:
IF <Location: Office AND HLC: OfficeWork>
THEN: <BLOCK>
IF <Location: Restaurant AND HLC: HavingMeal>
THEN <BLOCK>
IF <HLC: Sleeping>
THEN <BLOCK>

Context Interpreter– Week 3 Progress

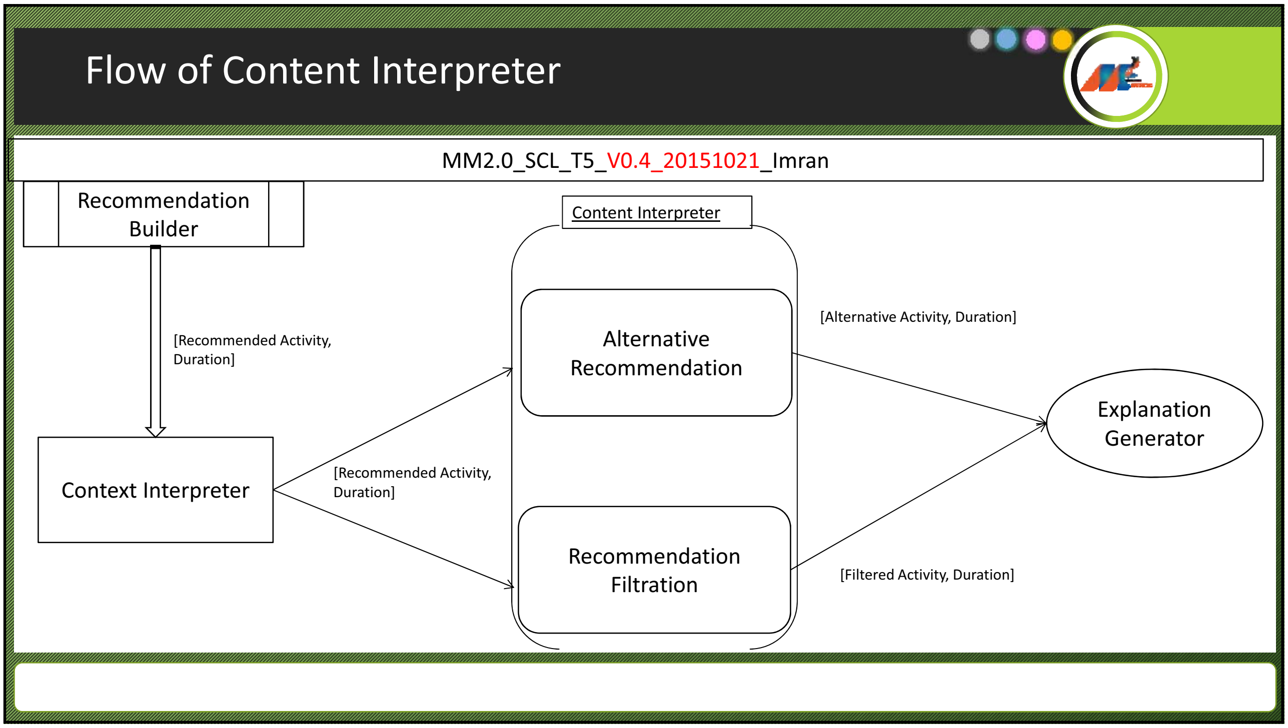
MM V2.0_SCL_T5_V0.3_20151021_Imran

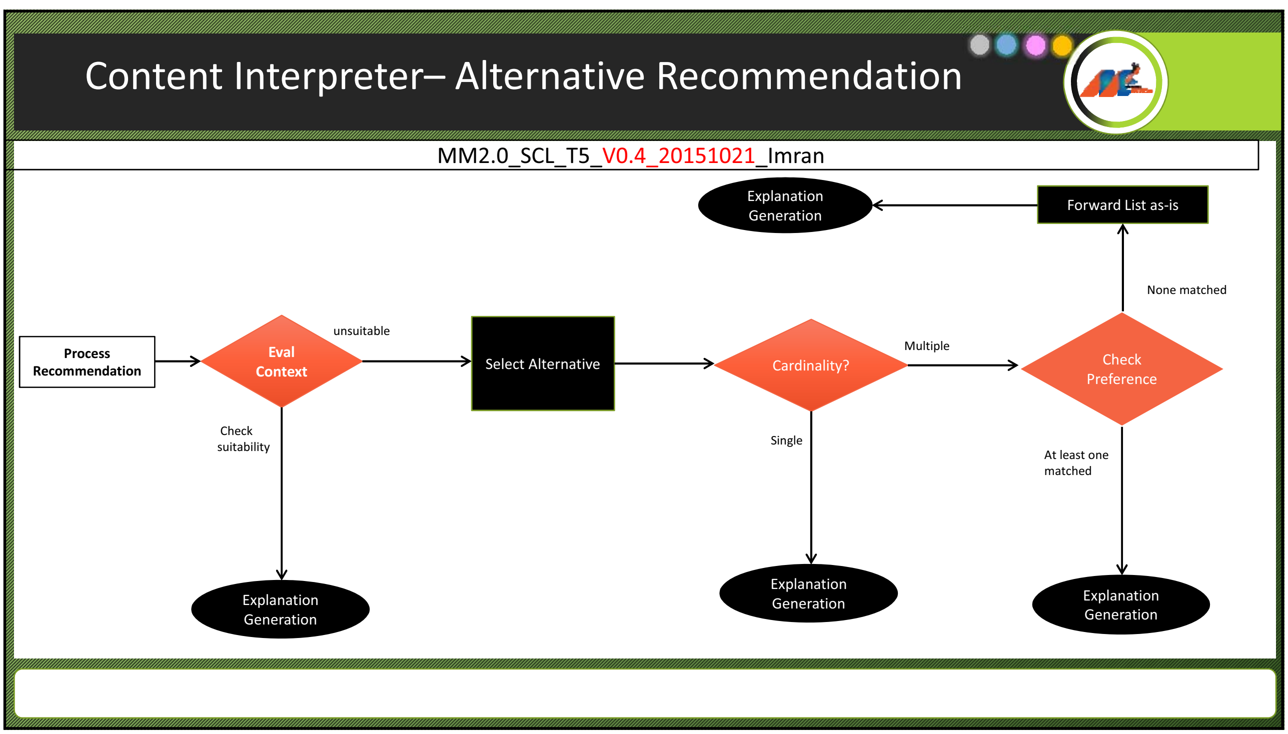


Content Interpreter– Purpose

MM2.0_SCL_T5_V0.4_20151021_Imran

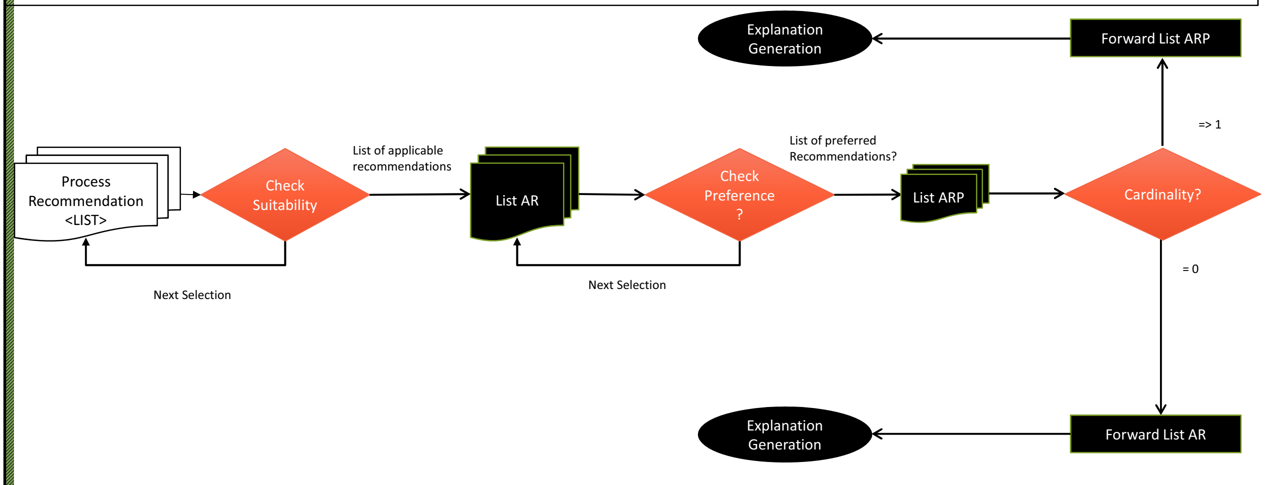
- Provide a mechanism for alternative recommendation
- Provide mechanism for filtration of recommendations based on context and/or preferences


Flow of Content Interpreter
MM2.0_SCL_T5_V0.4_20151021_Imran
Recommendation Builder
[Recommended Activity, Duration]
Context Interpreter
[Recommended Activity, Duration]
Content Interpreter
Alternative Recommendation
Recommendation Filtration
[Alternative Activity, Duration]
[Filtered Activity, Duration]
Explanation Generator


Content Interpreter– Alternative Recommendation
MM2.0_SCL_T5_V0.4_20151021_Imran
Explanation Generation
Forward List as-is
None matched
Process Recommendation
Eval Context
unsuitable
Select Alternative
Cardinality?
Multiple
Check Preference
Check suitability
Single
At least one matched
Explanation Generation
Explanation Generation
Explanation Generation

Content Interpreter– Recommendation Filtration

MM2.0_SCL_T5_V0.4_20151021_Imran

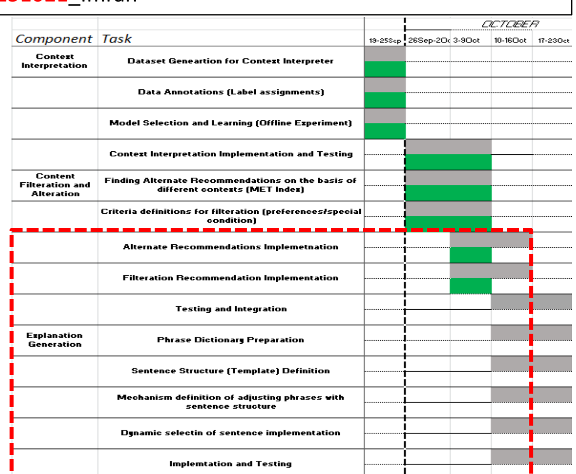


Context Interpreter– Week 4 Future Plan

MM2.0_SCL_T5_V0.4_20151021_Imran

What I will do:

1. Complete Implementation of Alternative Recommendation
2. Complete Implementation of Recommendation Filtration
3. Testing and Integration
4. Design mechanism for Explanation Generation

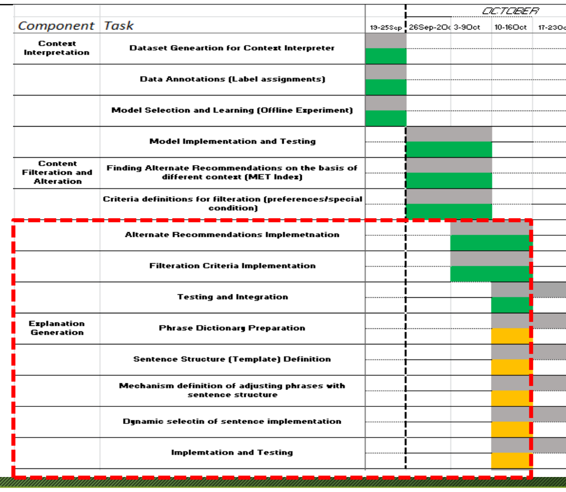


Context Interpreter– Week 4 Progress

MM2.0_SCL_T5_V0.4_20151021_Imran

What I Did:

1. Implemented Alternative Recommendation
2. Implemented Recommendation Filtration
3. Completed Testing and Integration
4. Design of Explanation Generation in process



Content Interpreter– Implementation

MM2.0_SCL_T5_V0.4_20151021_Imran

Representation of Contextual Data:

How to represent Contextual Data for Content Interpretation for each Context (Location, HLC, Weather, Emotion, etc)?

Content Interpreter– Implementation

MM2.0_SCL_T5_V0.4_20151021_Imran

Representation of Contextual Data:

How to represent Contextual Data for Content Interpretation for each Context (Location, HLC, Weather, Emotion, etc)?

OPTION 1: Rules Form

Issue: number of rules creation and maintenance

(LOCATION Rules)

IF <Location: Office> → Stretching
IF <Location: Office> → Standing
IF <Location: Office> → Sitting
…
IF <Location: OutDoors> → Hiking
IF <Location: OutDoors> → Cycling
IF <Location: OutDoors> → Stretching
IF <Location: OutDoors> → Jogging

…
(Key,Value) = <"Office" , "Stretching, Standing, Sitting"
(Key,Value) = <"OutDoors" , "Hiking, Cycling, Stretching, Jogging"
…

Content Interpreter– Implementation

MM2.0_SCL_T5_V0.4_20151021_Imran

Representation of Contextual Data:

How to represent Contextual Data for Content Interpretation for each Context (Location, HLC, Weather, Emotion, etc)?

OPTION 1: Rules Form

OPTION 2: Matrix Form

Benefits: Easy to maintain and process

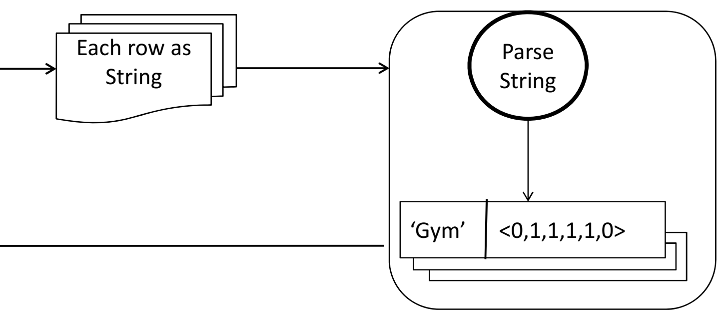
Location	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Gym	0	1	1	1	1	0
Home	0	1	1	1	0	1
Office	1	1	0	1	1	0
Transport	0	1	1	0	1	0
Outdoors	1	1	0	1	0	1
Don't Care	1	1	1	1	1	1

Content Interpreter– Processing Data Matrix

MM2.0_SCL_T5_V0.4_20151021_Imran

(Key,Value) CSV file

Location	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Gym	0	1	1	1	1	0
Home	0	1	1	1	0	1
Office	1	1	0	1	1	0
Transport	0	1	1	0	1	0
Outdoors	1	1	0	1	0	1
Don't Care	1	1	1	1	1	1



Loc_HashMap

Gym: 0,1,1,1,1,0
Home: 0,1,1,1,0,1
...

Content Interpreter– Implementation

MM2.0_SCL_T5_V0.4_20151021_Imran

Context_HashMaps

Location	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Gym	0	1	1	1	1	0
Home	0	1	1	1	0	1
Office	1	1	0	1	1	0
Transport	0	1	1	0	1	0
Outdoors	1	1	0	1	0	1
Don't Care	0	1	0	1	0	1

HLC	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Amusement	0	1	1	1	1	0
Commuting	0	0	1	1	0	1
Exercising	1	1	0	1	1	1
Gardening	0	0	1	0	1	0
HavingMeal	0	1	0	1	1	1
Inactivity	1	1	0	1	1	1

Weather	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Rainy	0	1	1	1	0	1
Sunny	0	0	1	1	0	1
Windy	1	1	0	1	1	1
Cloudy	0	0	1	0	1	0
Don't Care	1	1	1	1	1	1

Emotion	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Anger	1	1	0	1	1	0
Boredom	0	0	1	1	0	1
Happy	1	1	0	1	1	1
Fear	0	0	1	1	1	0
Sad	1	1	0	1	1	0
Don't Care	1	1	1	1	1	1

Disability	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Injury1	0	1	1	1	1	0
Injury2	0	0	1	1	0	1
Injury3	1	1	0	1	1	1
Injury4	0	0	1	0	1	0
Injury5	1	1	0	1	0	1
Don't Care	1	1	1	1	1	1

Original Recommendation: **Hiking**
Context: Gym, Exercising, Rainy, DC, Injury5

Content Interpreter– Implementation

Context_HashMaps

MM2.0_SCL_T5_V0.4_20151021_Imran

Location	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Gym	0	1	1	1	1	0
Home	0	1	1	1	0	1
Office	1	1	0	1	1	0
Transport	0	1	1	0	1	0
Outdoors	1	1	0	1	0	1
Don't Care	0	1	0	1	0	1

HLC	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Amusement	0	1	1	1	1	0
Commuting	0	0	1	1	0	1
Exercising	1	1	0	1	1	1
Gardening	0	0	1	0	1	0
HavingMeal	0	1	0	1	1	1
Inactivity	1	1	0	1	1	1

Weather	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Rainy	0	1	1	1	0	1
Sunny	0	0	1	1	0	1
Windy	1	1	0	1	1	1
Cloudy	0	0	1	0	1	0
Don't Care	1	1	1	1	1	1

Emotion	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Anger	1	1	0	1	1	0
Boredom	0	0	1	1	0	1
Happy	1	1	0	1	1	1
Fear	0	0	1	1	1	0
Sad	1	1	0	1	1	0
Don't Care	1	1	1	1	1	1

Disability	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Injury1	0	1	1	1	1	0
Injury2	0	0	1	1	0	1
Injury3	1	1	0	1	1	1
Injury4	0	0	1	0	1	0
Injury5	1	1	0	1	0	1
Don't Care	1	1	1	1	1	1

Original Recommendation: **Hiking**

Context: Gym, Exercising, Rainy, DC, Injury5

Combined	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Location=Gym	0	1	1	1	1	0
HLC=Exercising	1	1	0	1	1	1
Weather=Rainy	0	1	1	1	0	1
Emotion=Don't Care	1	1	1	1	1	1
Disability=Injury5	1	1	0	1	0	1

Content Interpreter– Implementation

Context_HashMaps

MM2.0_SCL_T5_V0.4_20151021_Imran

Location	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Gym	0	1	1	1	1	0
Home	0	1	1	1	0	1
Office	1	1	0	1	1	0
Transport	0	1	1	0	1	0
Outdoors	1	1	0	1	0	1
Don't Care	0	1	0	1	0	1

HLC	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Amusement	0	1	1	1	1	0
Commuting	0	0	1	1	0	1
Exercising	1	1	0	1	1	1
Gardening	0	0	1	0	1	0
HavingMeal	0	1	0	1	1	1
Inactivity	1	1	0	1	1	1

Weather	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Rainy	0	1	1	1	0	1
Sunny	0	0	1	1	0	1
Windy	1	1	0	1	1	1
Cloudy	0	0	1	0	1	0
Don't Care	1	1	1	1	1	1

Emotion	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Anger	1	1	0	1	1	0
Boredom	0	0	1	1	0	1
Happy	1	1	0	1	1	1
Fear	0	0	1	1	1	0
Sad	1	1	0	1	1	0
Don't Care	1	1	1	1	1	1

Disability	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Injury1	0	1	1	1	1	0
Injury2	0	0	1	1	0	1
Injury3	1	1	0	1	1	1
Injury4	0	0	1	0	1	0
Injury5	1	1	0	1	0	1
Don't Care	1	1	1	1	1	1

Original Recommendation: **Hiking**

Context: Gym, Exercising, Rainy, DC, Injury5

Combined	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Location=Gym	0	1	1	1	1	0
HLC=Exercising	1	1	0	1	1	1
Weather=Rainy	0	1	1	1	0	1
Emotion=Don't Care	1	1	1	1	1	1
Disability=Injury5	1	1	0	1	0	1
AGGREGATE	**0**	**1**	**0**	**1**	**0**	**0**

Logical AND

Content Interpreter– Implementation

Context_HashMaps

MM2.0_SCL_T5_V0.4_20151021_Imran

Location	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Gym	0	1	1	1	1	0
Home	0	1	1	1	0	1
Office	1	1	0	1	1	0
Transport	0	1	1	0	1	0
Outdoors	1	1	0	1	0	1
Don't Care	0	1	0	1	0	1

HLC	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Amusement	0	1	1	1	1	0
Commuting	0	0	1	1	0	1
Exercising	1	1	0	1	1	1
Gardening	0	0	1	0	1	0
HavingMeal	0	1	0	1	1	1
Inactivity	1	1	0	1	1	1

Weather	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Rainy	0	1	1	1	0	1
Sunny	0	0	1	1	0	1
Windy	1	1	0	1	1	1
Cloudy	0	0	1	0	1	0
Don't Care	1	1	1	1	1	1

Emotion	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Anger	1	1	0	1	1	0
Boredom	0	0	1	1	0	1
Happy	1	1	0	1	1	1
Fear	0	0	1	1	1	0
Sad	1	1	0	1	1	0
Don't Care	1	1	1	1	1	1

Disability	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Injury1	0	1	1	1	1	0
Injury2	0	0	1	1	0	1
Injury3	1	1	0	1	1	1
Injury4	0	0	1	0	1	0
Injury5	1	1	0	1	0	1
Don't Care	1	1	1	1	1	1

Original Recommendation: **Hiking**

Context: Gym, Exercising, Rainy, DC, Injury5

Combined	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Location=Gym	0	1	1	1	1	0
HLC=Exercising	1	1	0	1	1	1
Weather=Rainy	0	1	1	1	0	1
Emotion=Don't Care	1	1	1	1	1	1
Disability=Injury5	1	1	0	1	0	1
AGGREGATE	**0**	**1**	**0**	**1**	**0**	**0**

Content Interpreter– Implementation

MM2.0_SCL_T5_V0.4_20151021_Imran

Location	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Gym	0	1	1	1	1	0
Home	0	1	1	1	0	1
Office	1	1	0	1	1	0
Transport	0	1	1	0	1	0
Outdoors	1	1	0	1	0	1
Don't Care	0	1	0	1	0	1

HLC	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Amusement	0	1	1	1	1	0
Commuting	0	0	1	1	0	1
Exercising	1	1	0	1	1	1
Gardening	0	0	1	0	1	0
HavingMeal	0	1	0	1	1	1
Inactivity	1	1	0	1	1	1

Weather	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Rainy	0	1	1	1	0	1
Sunny	0	0	1	1	0	1
Windy	1	1	0	1	1	1
Cloudy	0	0	1	0	1	0
Don't Care	1	1	1	1	1	1

Emotion	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Anger	1	1	0	1	1	0
Boredom	0	0	1	1	0	1
Happy	1	1	0	1	1	1
Fear	0	0	1	1	1	0
Sad	1	1	0	1	1	0
Don't Care	1	1	1	1	1	1

Disability	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Injury1	0	1	1	1	1	0
Injury2	0	0	1	1	0	1
Injury3	1	1	0	1	1	1
Injury4	0	0	1	0	1	0
Injury5	1	1	0	1	0	1
Don't Care	1	1	1	1	1	1

Original Recommendation: **Hiking**

Context: Gym, Exercising, Rainy, DC, Injury5

Combined	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Location=Gym	0	1	1	1	1	0
HLC=Exercising	1	1	0	1	1	1
Weather=Rainy	0	1	1	1	0	1
Emotion=Don't Care	1	1	1	1	1	1
Disability=Injury5	1	1	0	1	0	1
AGGREGATE	**0**	**1**	**0**	**1**	**0**	**0**

Alternative Recommendations: <**Jogging OR Stretching**>

Content Interpreter– Implementation

MM2.0_SCL_T5_V0.4_20151021_Imran

Two possible Scenarios for Aggregate vector

Combined	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Location=Gym	0	1	1	1	1	0
HLC=Exercising	1	1	0	1	1	1
Weather=Rainy	0	1	1	1	0	1
Emotion=Don't Care	1	1	1	1	1	1
Disability=Injury5	1	1	0	1	0	1
AGGREGATE	**0**	**1**	**0**	**1**	**0**	**0**

Aggregate vector contains one or more '1' bits

Aggregate vector contains all '0' bits

Content Interpreter– Implementation

MM2.0_SCL_T5_V0.4_20151021_Imran

Aggregate vector contains all '0' bits

Combined	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Location=Gym	0	1	1	1	1	0
HLC=Exercising	1	1	0	1	1	1
Weather=Rainy	0	1	1	1	0	1
Emotion=Don't Care	1	1	1	1	1	1
Disability=Injury5	0	0	1	0	0	1
AGGREGATE	**0**	0	**0**	0	**0**	**0**

Consequences:

- Original Recommendation was not suitable against the current context
- Alternative Recommendations blocked due to certain inapplicable contexts
- Neither Original or Alternative Recommendations can forwarded

Content Interpreter– Implementation

MM2.0_SCL_T5_V0.4_20151021_Imran

Critical Context overrides Aggregate vector

Combined	Recommendations					
Context	*Hiking*	*Jogging*	*Sitting*	*Stretching*	*Cycling*	*Walking*
Location=Gym	0	1	1	1	1	0
HLC=Exercising	1	1	0	1	1	1
Weather=Rainy	0	1	1	1	0	1
Emotion=Don't Care	1	1	1	1	1	1
Disability=Injury5	0	0	1	0	0	1
AGGREGATE	0	0	0	0	0	0

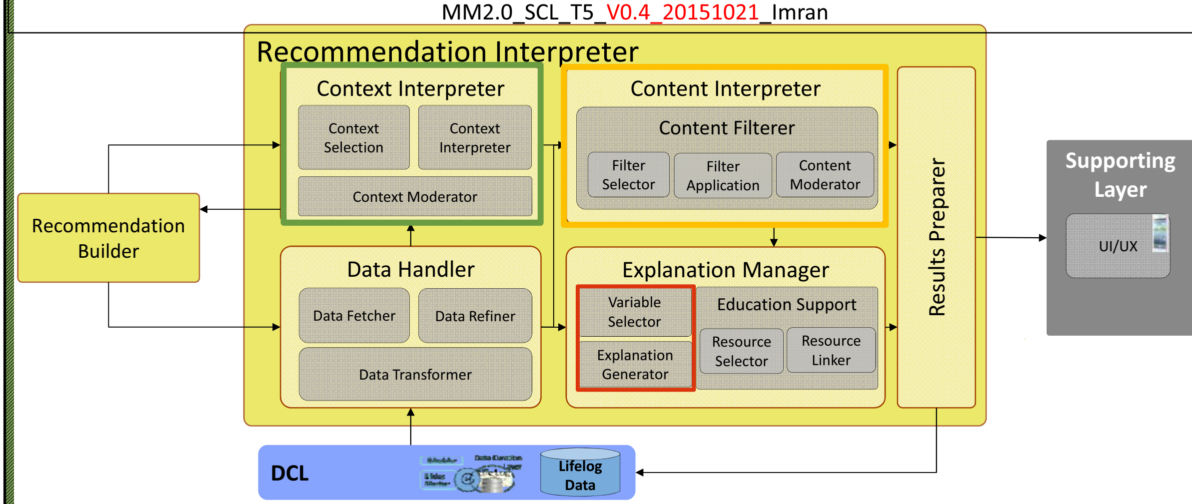
AND

One Possible Solution:
In case of deadlock among contexts, let the 'critical context' override results of aggregate vector

Forwarded Recommendations = Sitting, Walking

Recommendation Interpreter – Week 5 Plan

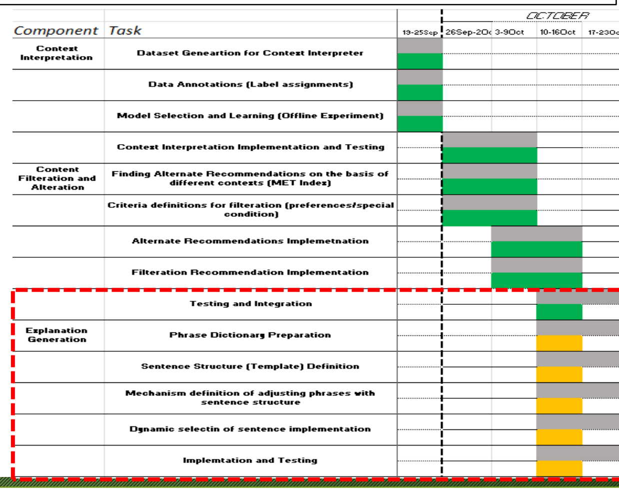
MM2.0_SCL_T5_V0.4_20151021_Imran



Context Interpreter – Week 5 Future Plan

MM2.0_SCL_T5_V0.4_20151021_Imran

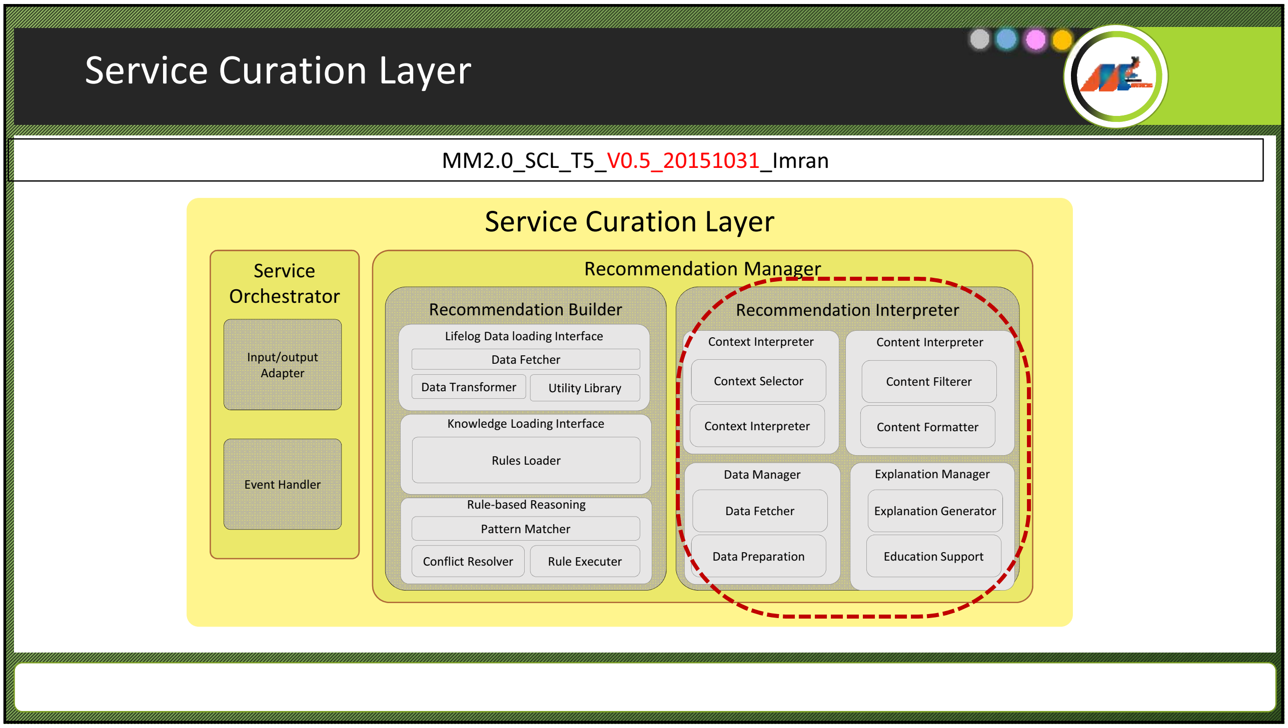
- Content Interpreter: Testing and Integration
- Explanation Generation (Design and Implementation)
  - Phrase Dictionary Preparation
  - Template Definition
  - Phrase, Sentence Matching
  - Dynamic Selection of Phrases
  - Implementation and Testing

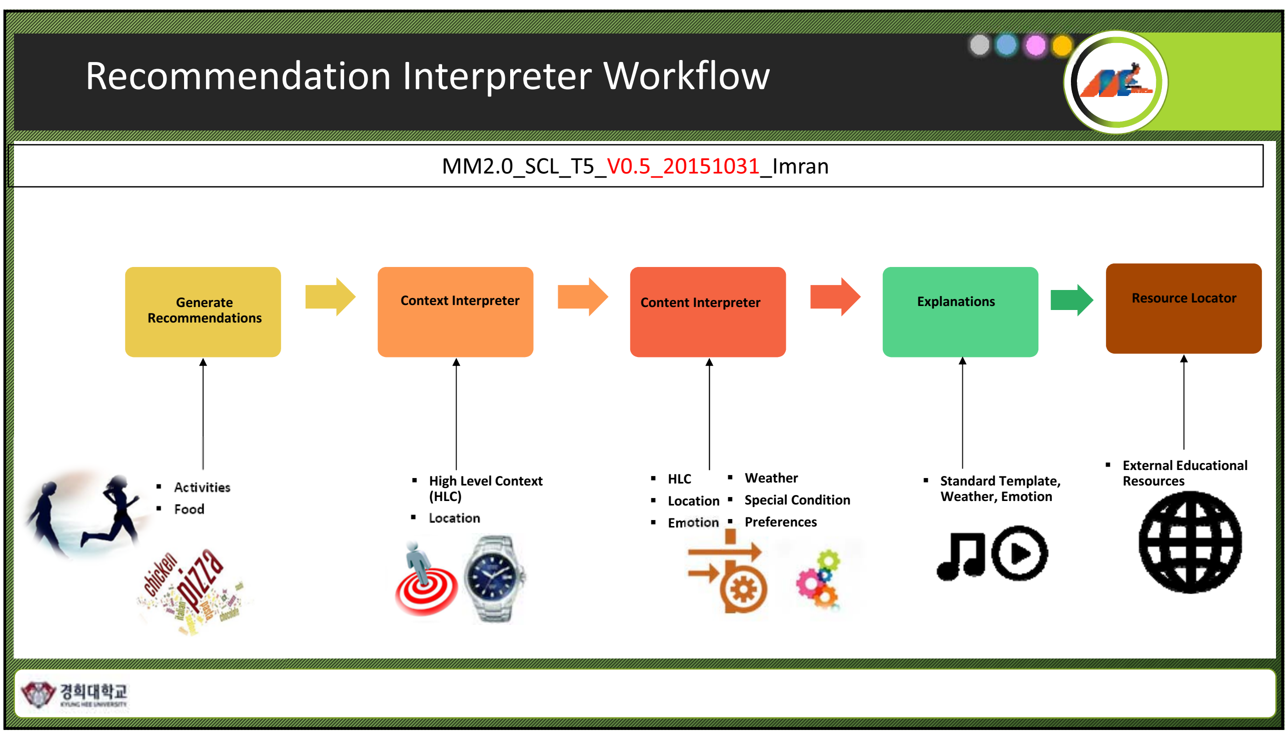


Context Interpreter – Week 5

MM2.0_SCL_T5_V0.5_20151024_Imran

- Explanation Generation
  - Description of the recommendation
  - Associated Video URL
- Standard Template for Description Generation
  - Description = “You are Recommended” + [Recommendation] + “for” + [Duration] + “units”
- Post Appending Sentences for selected contexts
  - Selected Contexts: Weather := Rainy, Weather := Sunny, Emotion := Anger/Sadness
- URL Association
  - Each description is associated with a URL which is fetched according to the type of the recommended activity


Service Curation Layer
MM2.0_SCL_T5_V0.5_20151031_Imran
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
Rule Executer
Recommendation Interpreter
Context Interpreter
Context Selector
Context Interpreter
Content Interpreter
Content Filterer
Content Formatter
Data Manager
Data Fetcher
Data Preparation
Explanation Manager
Explanation Generator
Education Support


Recommendation Interpreter Workflow
MM2.0_SCL_T5_V0.5_20151031_Imran
Generate Recommendations
Context Interpreter
Content Interpreter
Explanations
Resource Locator
Activities
Food
chicken
pizza
High Level Context (HLC)
Location
HLC
Location
Emotion
Weather
Special Condition
Preferences
Standard Template, Weather, Emotion
External Educational Resources
경희대학교

Scenarios

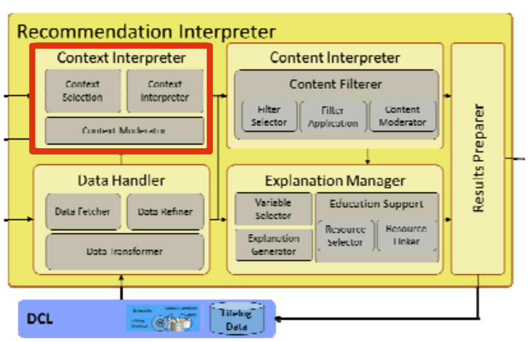
MM2.0_SCL_T5_V0.5_20151031_Imran

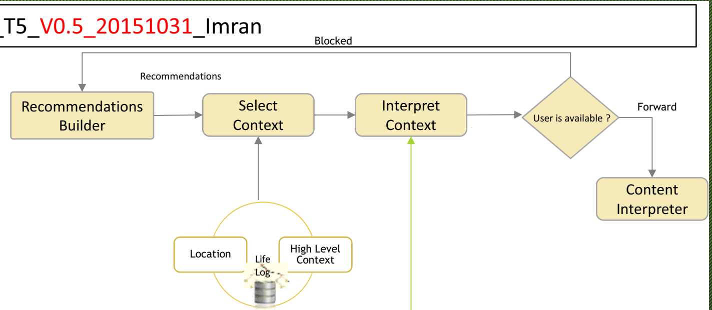
1. Inapplicable Context in terms of user availability
2. Applicable Original Recommendation for the given Context (Normal Scenario)
3. Inapplicable Original Recommendation for the current user Context (Alternative Recommendation)
4. Preference Check to filter multiple recommendations (Preference based filtration)
5. Explanation Generation (No Alteration)
6. Explanation Generation (Sentence generation with Standard Template)
7. Explanation (Post Appended Phrase)

1. Inapplicable Context

MM2.0_SCL_T5_V0.5_20151031_Imran

Purpose: To checking availability of the user
Input: Recommendations, High Level Context, Location
Output: Applicable/Inapplicable





Context Interpretation Rules:

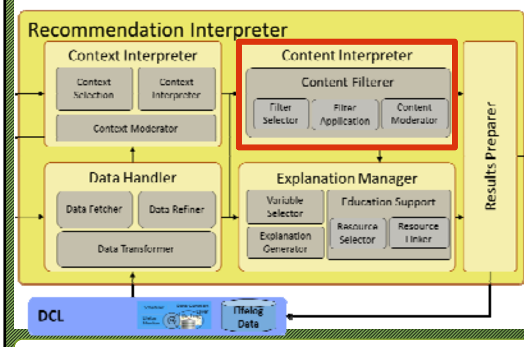
IF <Location: Office AND HLC: OfficeWork>
THEN: <**BLOCK**>
IF <Location: Restaurant AND HLC: HavingMeal>
THEN <**BLOCK**>
IF <HLC: Sleeping>
THEN <**BLOCK**>

경희대학교

Scenario 3: Alternative Recommendation

MM2.0_SCL_T5_V0.5_20151031_Imran

- Purpose: To evaluate Recommendation against the current Context
- Input: HLC, Location, Emotion, Weather, Special Condition, Recommendation List
- Output:
  - Recommended Activity

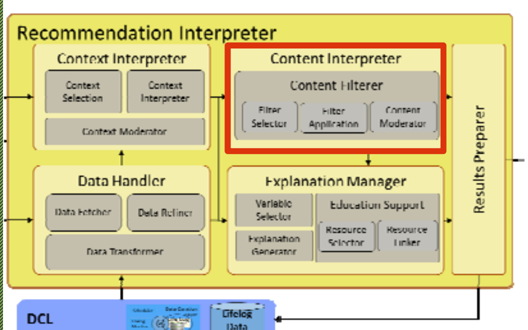


Current Context/Condition		Recommendations				
Context	Value	Jogging	Walking	Running	Stretching	Dancing
Location	**Mall**	0	1	0	0	0
HLC	**Inactivity**	0	1	0	0	0
Weather	**Sunny**	1	1	1	1	1
Emotion	**Neutral**	1	1	1	1	1
Injury	**No**	1	1	1	1	1
Aggregate		**0**	**1**	**0**	**0**	**0**

Scenario 4: Preference based Filtration of Multiple Recommendations

MM2.0_SCL_T5_V0.5_20151031_Imran

- Purpose: To evaluate Recommendation against the current Context
- Input: HLC, Location, Emotion, Weather, Special Condition, Recommendation List
- Output:
  - Recommended Activity



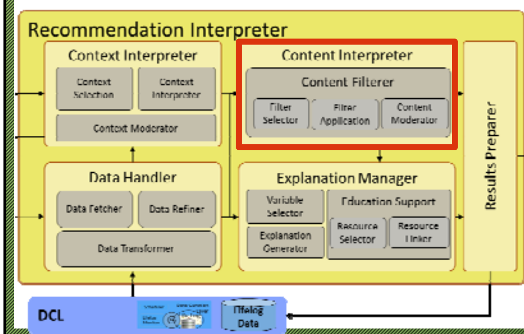
Current Context/Condition		Recommendations				
Context	Value	Jogging	Walking	Running	Stretching	Dancing
Location	**Yard**	1	1	1	1	0
HLC	**Gardening**	0	1	0	1	0
Weather	**Sunny**	1	1	1	1	1
Emotion	**Neutral**	1	1	1	1	1
Injury	**No**	1	1	1	1	1
Aggregate (user1)		**0**	**1**	**0**	**1**	**0**

Scenario 4: Preference based Filtration of Multiple Recommendations

MM2.0_SCL_T5_V0.5_20151031_Imran

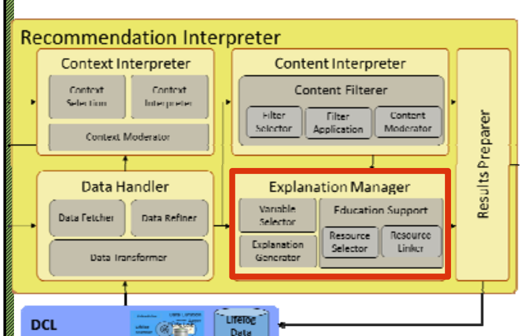
- Purpose: To evaluate Recommendation against the current Context
- Input: HLC, Location, Emotion, Weather, Special Condition, Recommendation List
- Output:
  - Recommended Activity

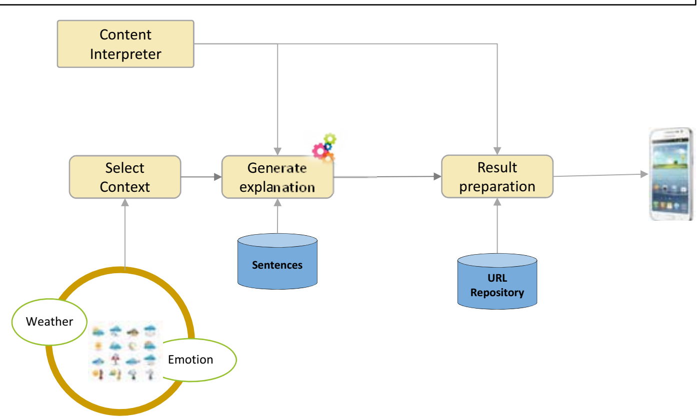
Current Context/Condition		Recommendations				
Context	Value	Jogging	Walking	Running	Stretching	Dancing
Location	**Yard**	1	1	1	1	0
HLC	**Gardening**	0	1	0	1	0
Weather	**Sunny**	1	1	1	1	1
Emotion	**Neutral**	1	1	1	1	1
Injury	**No**	1	1	1	1	1
Aggregate (user2)		**0**	**1**	**0**	**1**	**0**



Explanation Generation

MM2.0_SCL_T5_V0.5_20151031_Imran

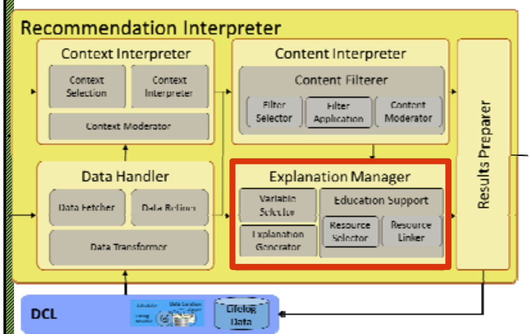




Scenario 5: Explanation Generation (as-it-is)

MM2.0_SCL_T5_V0.5_20151031_Imran

- Purpose: To generate sentence
- Input:
  - Current Context: *Outdoors, Amusement, Sunny, Happiness, No*
  - Recommended Activity: *Stretching*
- Output:
  - Recommended Activity: *Stretching*
  - Duration: X minutes
  - Sentence: Take a 2min bouts of light-intensity activity every 20min to increase the chance of diabetes prevention.



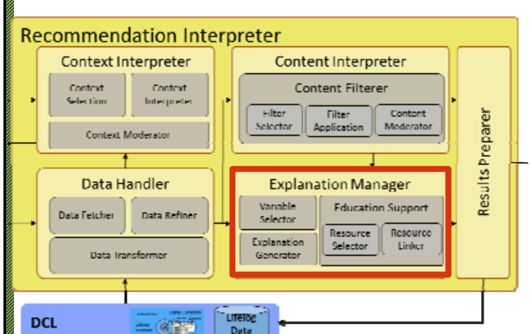
Scenario 6: Explanation Generation (Standard Template)

MM2.0_SCL_T5_V0.5_20151031_Imran

Standard Template

”you are recommended” <Activity Name>+“ for ”+<Duration>+” <unit>. “

- Purpose: To generate sentence
- Input:
  - Current Context: *Outdoors, Inactivity, Windy, Neutral, No*
  - Recommended Activity: *Dancing*
- Output:
  - Recommended Activity: *Walking*
  - Duration: x minutes
  - Sentence: why don't you go for walking for X mins
  - For more information please visit: href1/w1

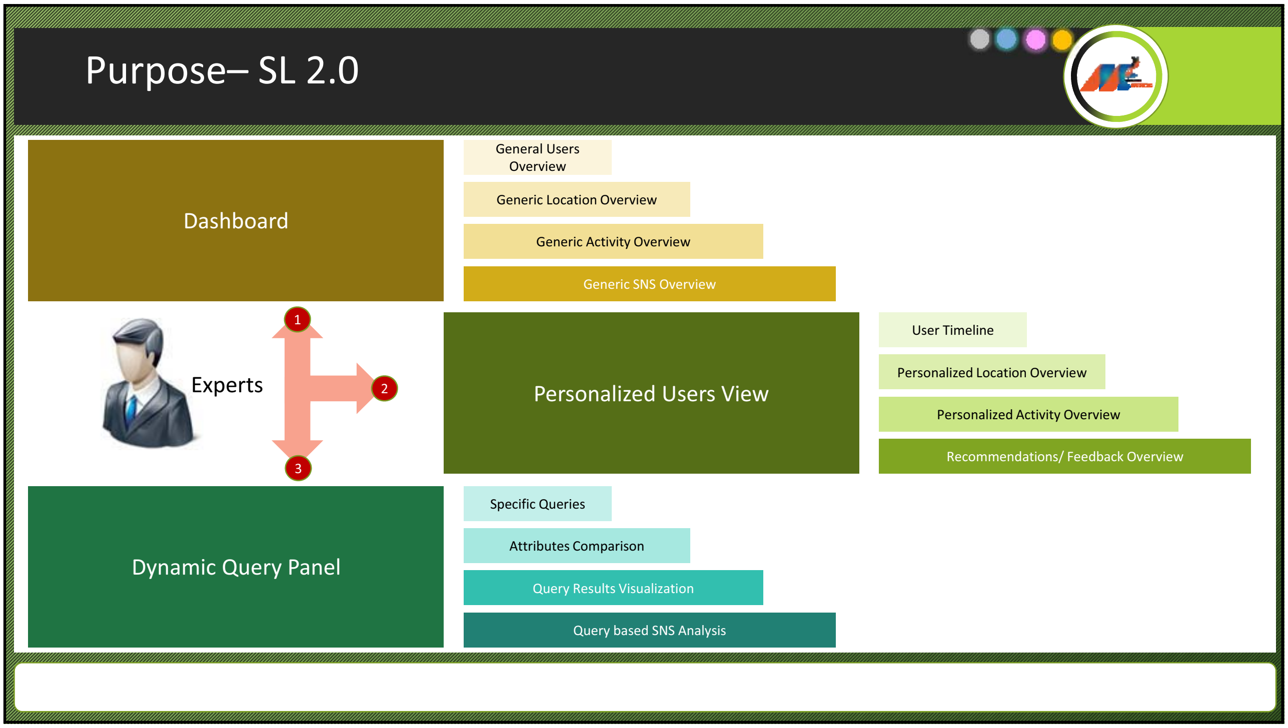


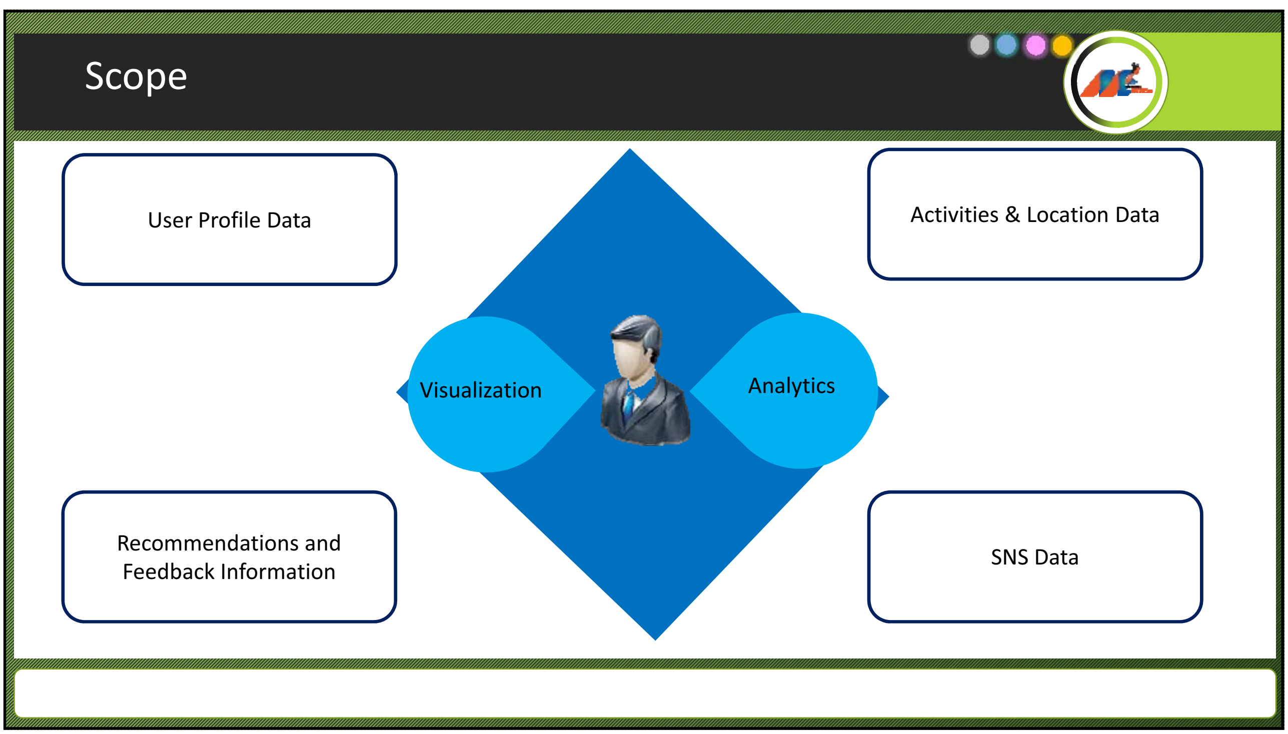
Supporting Layer (SL 2.0)

Principal Investigator:
Professor Sungyoung Lee

Descriptive Analytics

Shujaat Hussain


Purpose– SL 2.0
Dashboard
General Users Overview
Generic Location Overview
Generic Activity Overview
Generic SNS Overview
Experts
1
2
3
Personalized Users View
User Timeline
Personalized Location Overview
Personalized Activity Overview
Recommendations/ Feedback Overview
Dynamic Query Panel
Specific Queries
Attributes Comparison
Query Results Visualization
Query based SNS Analysis

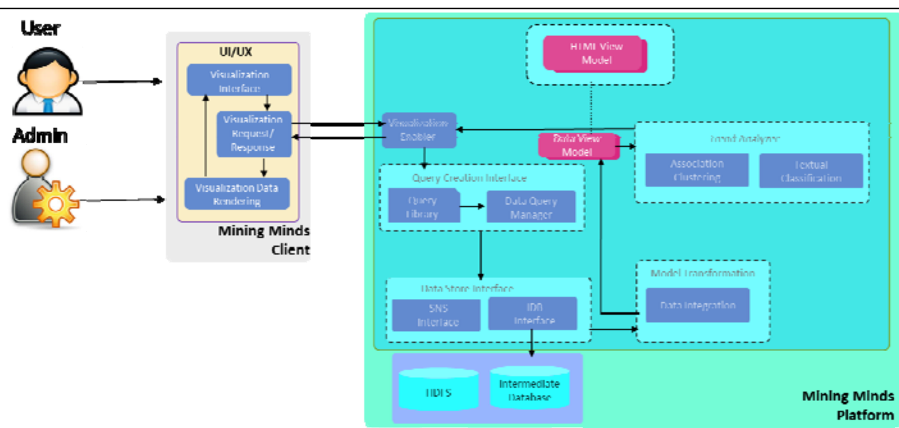

Scope
User Profile Data
Activities & Location Data
Visualization
Analytics
Recommendations and Feedback Information
SNS Data

Tools and Technologies

V2.0_SL_T5_V0.7_151114_Shujaat

Descriptive Analytics

V2.0_SL_T5_V0.1_151002_Shujaat



Descriptive Analytics– Week 1 Progress

MM V2.0_SL_T5_V0.1_151002_Shujaat

- Query Breakdown Implementation
  - Query Interface Design
  - Query Mapping
  - DB Schema Modification
- Open Issues
  - *Mapping queries with big data repository*

Descriptive Analytics– Week 2 Future Plan

MM V2.0_SL_T5_V0.1_151002_Shujaat

- Discussion with data curation layer on Query Library w.r.t.
  - *Activity and Location (Group)*
  - *Activity and Location (Individual)*
  - *Restful Web Services and SQL procedures*
  - *Big Data Storage Mechanism*

Descriptive Analytics– Week 2 Progress

MM V2.0_SL_T5_V0.1_151002_Shujaat

- Dashboard Design and Group Analytics
  - Location
  - Age
  - Activity
  - User recommendations
- Query Panel Design (Partial Implementation)

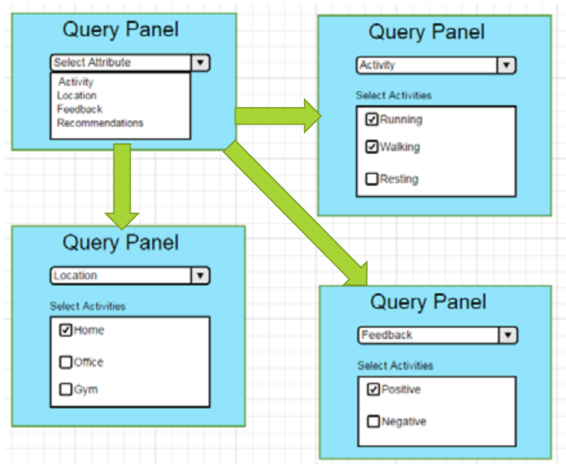
Dashboard View

MM V2.0_SL_T5_V0.1_151002_Shujaat

Query Panel View

MM V2.0_SL_T5_V0.1_151002_Shujaat

- Through the query panel a main attribute is selected.
  - After selecting the main attribute conditional attribute is displayed as shown in the figure
- The selected conditional attribute makes up a parameters to be sent to the data curation layer.



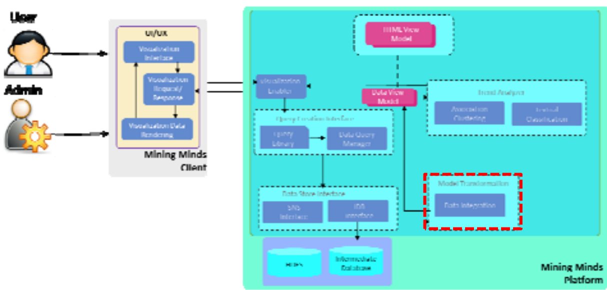
Descriptive Analytics– Week 3 Future Plan

MM V2.0_SupportingLayer_T5_V0.1_151008_Shujaat

- Defining the list of services/stored procedures
- Defining the input parameters and their feasibility with DCL.
- Discuss DB schema changes for data facts

Descriptive Analytics– Week 3 Progress

V2.0_SL_T5_V0.3_151016_Shujaat

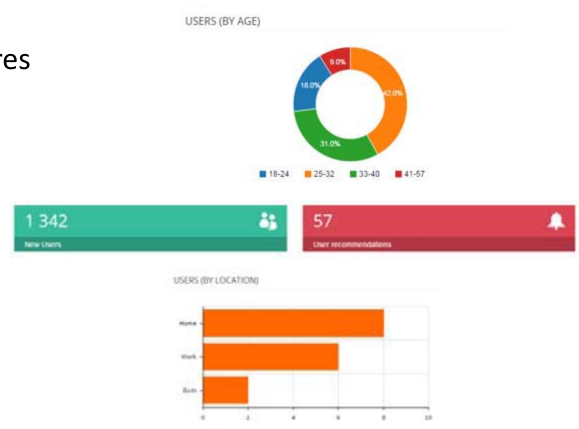


Component	Task	Responsible	19-25Sep	26Sep-2Oct	3-9Oct	10-16Oct	17-23Oct
Query Library and Mapping	Query interface design and mapping it to Query	Shujaat					
	Restful web service design for queries and DB schema						
Model Transformation	Intermediate DB transformation to JSON model	Shujaat					
	Big Data Transformation to JSON model						
Trend Analysis	Designing the interface for visualization (Mockups and UI Design)	Shujaat					
	Data Aggregation and data association						
	Text Classification						
Visualization	Selecting graphs and their attributes, Schema Design	Shujaat					
	Graph Selection and Data Facts	Shujaat					

Descriptive Analytics– Week 3 Progress

V2.0_SL_T5_V0.3_151016_Shujaat

- Defining the list of services/stored procedures
  - Total user recommendations
  - Total users
  - Users with age
  - Users Location



Descriptive Analytics– Week 3 Progress

V2.0_SL_T5_V0.3_151016_Shujaat

- Improving the implementation of MMV 1.5
  - Weekly /Monthly Activities
  - Weekly /Monthly Calories Burned
  - Feedback on recommendations
- Expert requests gave decreased from 38 restful web service calls to 3 web service calls.
  - Loads the expert panel quickly.
  - Reduces the load significantly on data curation layer.

Old Format

[{"activityDescription":"Walking","activityId":2,"duration":600,"userId":39},
{"activityDescription":"Sitting","activityId":3,"duration":7437,"userId":39},
{"activityDescription":"LyingDown","activityId":7,"duration":2028,"userId":39},
{"activityDescription":"Standing","activityId":8,"duration":7758,"userId":39},
{"activityDescription":"Jogging","activityId":9,"duration":120,"userId":39},{
"activityDescription":"Stretching","activityId":13,"duration":48,"userId":39}]

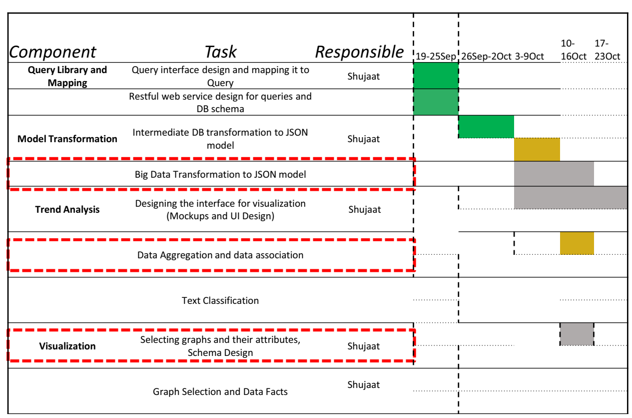
New Format

[{"activityDescription":"Walking","activityId":2,"duration":600,"userId":39,
"startDate":"2015-05-18 10:40:59.0" },
{"activityDescription":"Sitting","activityId":3,"duration":7437,"userId":39, ,
"startDate":"2015-05-19 10:40:59.0"},
{"activityDescription":"LyingDown","activityId":7,"duration":2028,"userId":39 ,
"startDate":"2015-05-20 10:40:59.0"},
{"activityDescription":"Standing","activityId":8,"duration":7758,"userId":39, ,
"startDate":"2015-05-21 10:40:59.0"},
{"activityDescription":"Jogging","activityId":9,"duration":120,"userId":39, ,
"startDate":"2015-05-22 10:40:59.0"},{
"activityDescription":"Stretching","activityId":13,"duration":48,"userId":39, ,
"startDate":"2015-05-23 10:40:59.0"}]

Descriptive Analytics– Week 4 Future Plan

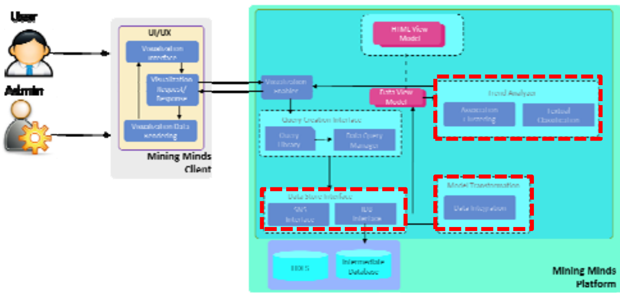
V2.0_SL_T5_V0.3_151016_Shujaat

- Discussion with Tapacross with respect to integration of SNS
  - *Focus on* Tapacross *API access*
  - *Query Construction to get relevant expert data*
  - *Understand the input and output of the query*



Descriptive Analytics– Week 4 Progress

V2.0_SL_T5_V0.4_151022_Shujaat



Component	Task	Responsible	19-25Sep	26Sep-2Oct	3-9Oct	10-16Oct	17-23Oct
Query Library and Mapping	Query interface design and mapping it to Query	Shujaat					
	Restful web service design for queries and DB schema						
Model Transformation	Intermediate DB transformation to JSON model	Shujaat					
	Big Data Transformation to JSON model						
Trend Analysis	Designing the interface for visualization (Mockups and UI Design)	Shujaat					
	Data Aggregation and data association						
	Text Classification						
Visualization	Selecting graphs and their attributes, Schema Design	Shujaat					
	Graph Selection and Data Facts	Shujaat					

Descriptive Analytics– Week 4 Progress

V2.0_SL_T5_V0.4_151022_Shujaat

- We discussed the query construction and the parameters for their SNS data API.
- A list of keywords will be provided to tapacross for static query for MM V2.0 development
- After November dynamic query processing capability will be given.
- The time, place, occasion analysis done by tapacross will be considered for next version.

검색식	우유		우유		우유	
기간	20130901 ~ 2015101		20130901 ~ 2015101		20130901 ~ 20151014	
총 버즈량	2966646		2966646		2966646	
순위	time	빈도	place	빈도	occasion	빈도
1	아침	199944	카페	115009	집	215581
2	저녁	112696	학교	61062	이벤트	136394
3	점심	100196	마트	46631	처음	92483
4	오후	57524	회사	37693	식사	70959
5	여름	55348	맛집	29975	요리	62993
6	밤	49301	스타벅스	29970	추첨	54593
7	주말	39594	부산	24517	사과	51196
8	오전	37632	동네	23523	주문	49920
9	새벽	26187	바다	21613	여행	39237
10	봄	25581	유럽	20861	구입	39088
11	가을	22654	파리바게!	18933	출시	39053
12	월요일	16109	버스	15419	겨울	36101
13	일요일	14623	화장실	14849	발견	33848
14	토요일	14202	입구	14700	일상	31772
15	이틀	11845	홍대	14660	사고	29725
16	추석	10943	백화점	14619	마실	29195
17	크리스마:	10800	우리집	14041	홍수	27314
18	화요일	7798	도로	13980	공부	26812
19	수요일	7670	호텔	13867	출근	23208
20	여름	7246	식당	13700	생일	21762

Descriptive Analytics– Week 4 Progress

V2.0_SL_T5_V0.4_151022_Shujaat

- The keywords are selected from the given topics covered by tapacross.
- They are selected based on relevance to he expert panel and the trends which will guide the expert.
- They are based on activity, location and food.

Sentiment	Volume/Sentiment	Volume/Sentiment
• Walk	• Home	• Diet
• Run	• Cafe	• Vegetables
• Sleep	• School	• Fruit
• Tired	• Restaurant	• Alcohol
• Exercise	• Hotel	• Cholesterol
• Pilates	• Office	• Protein
• Yoga	• Gym	• Carbohydrates
• Sit		• Recommendation

Descriptive Analytics– Week 5 Future Plan

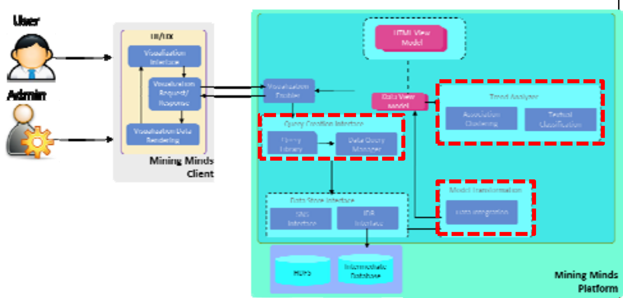
V2.0_SL_T5_V0.4_151022_Shujaat

- Integration of SNS data from tapacross
  - *Query the data from tapacross.*
  - *Parse the data from tapacross.*
  - *Aggregate and visualize the SNS trend.*
  - *Interface and dashboard designing with the help of Jamil*
- Selecting graphs through attributes

Component	*Task*	*Responsible*	19-25Sep	26Sep-2Oct	3-9Oct	10-16Oct	17-23Oct
Query Library and Mapping	Query interface design and mapping it to Query	Shujaat					
	Restful web service design for queries and DB schema						
Model Transformation	Intermediate DB transformation to JSON model	Shujaat					
	Big Data Transformation to JSON model						
Trend Analysis	Designing the interface for visualization (Mockups and UI Design)	Shujaat/Jamil					
	Data Aggregation and data association						
	Text Classification						
Visualization	Selecting graphs and their attributes, Schema Design	Shujaat					
	Graph Selection and Data Facts	Shujaat					

Descriptive Analytics– Week 5 Progress

V2.0_SL_T5_V0.5_151030_Shujaat



Component	Task	Responsible	19-25Sep	26Sep-2Oct	3-9Oct	10-16Oct	17-23Oct	23-30 October
Query Library and Mapping	Query interface design and mapping it to Query	Shujaat						
	Restful web service design for queries and DB schema							
Model Transformation	Intermediate DB transformation to JSON model	Shujaat						
	Big Data Transformation to JSON model							
Trend Analysis	Designing the interface for visualization (Mockups and UI Design)	Shujaat						
	Data Aggregation and data association							
	Text Classification							
Visualization	Selecting graphs and their attributes, Schema Design	Shujaat						
	Graph Selection and Data Facts	Shujaat						

Descriptive Analytics– Week 5 Progress

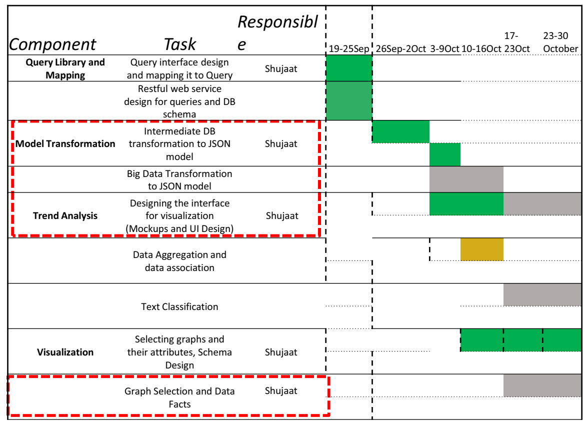
V2.0_SL_T5_V0.5_151030_Shujaat

- A Query library menu is created for generating dynamic comparison of different attributes in different scenarios
- The dynamic graph generation is done for different queries and it's attributes.
- JSON transformation is done for visualization and quick execution time.

Descriptive Analytics– Week 6 Future Plan

V2.0_SL_T5_V0.6_151030_Shujaat

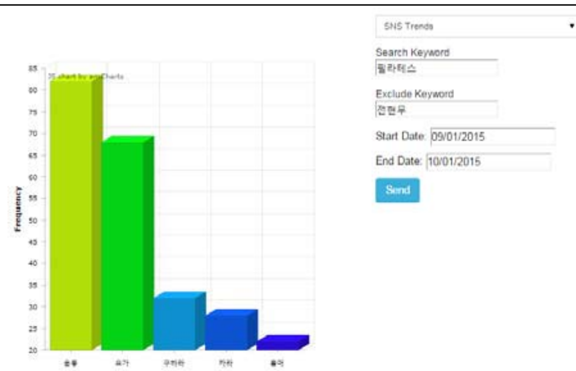
- Analytics extraction through date and time data
  - Datetime integration with DCL is a little tricky.
- Integration of SNS data from tapacross if possible
  - *Query the data from tapacross.*
  - *Parse the data from tapacross.*
  - *Aggregate and visualize the SNS trend.*
  - *Interface and dashboard designing with the help of Jamil*



Descriptive Analytics– Week 6 Progress

V2.0_SL_T5_V0.6_151106_Shujaat

- A Query library menu is created for generating SNS data from the tapacross.
- The dynamic graph generation is done for different queries and it's attributes.
- JSON transformation is done for visualization and quick execution time.



Descriptive Analytics– Week 7 Future Plan

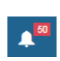
V2.0_SL_T5_V0.6_151106_Shujaat

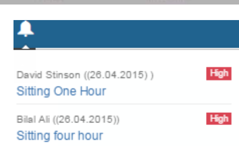
- LLM user categorization
  - Three categories of users based on the lifelog monitoring
- Integration of Push based notifications from the LLM.
- Creating Emotions and high level context for particular user.

Descriptive Analytics– Week 7 Progress

V2.0_SL_T5_V0.7_151106_Shujaat

- LLM user categorization was done on 3 categories.
  - Bad Habituation
  - Improper Routine
  - Good Routine
- Push based notifications are done for expert view.



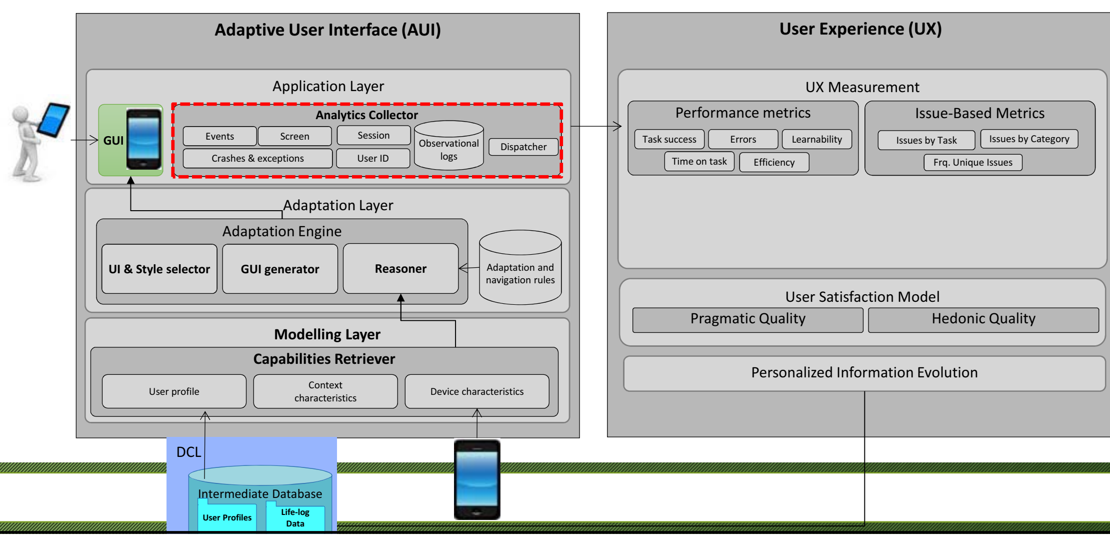


UI/UX Authoring Tool

Jamil Hussain

Analytics Collector

MM V2.0_SL_T5_V0.1_150926_Jamil_Hussain



Analytics Collector – Week 1 Progress

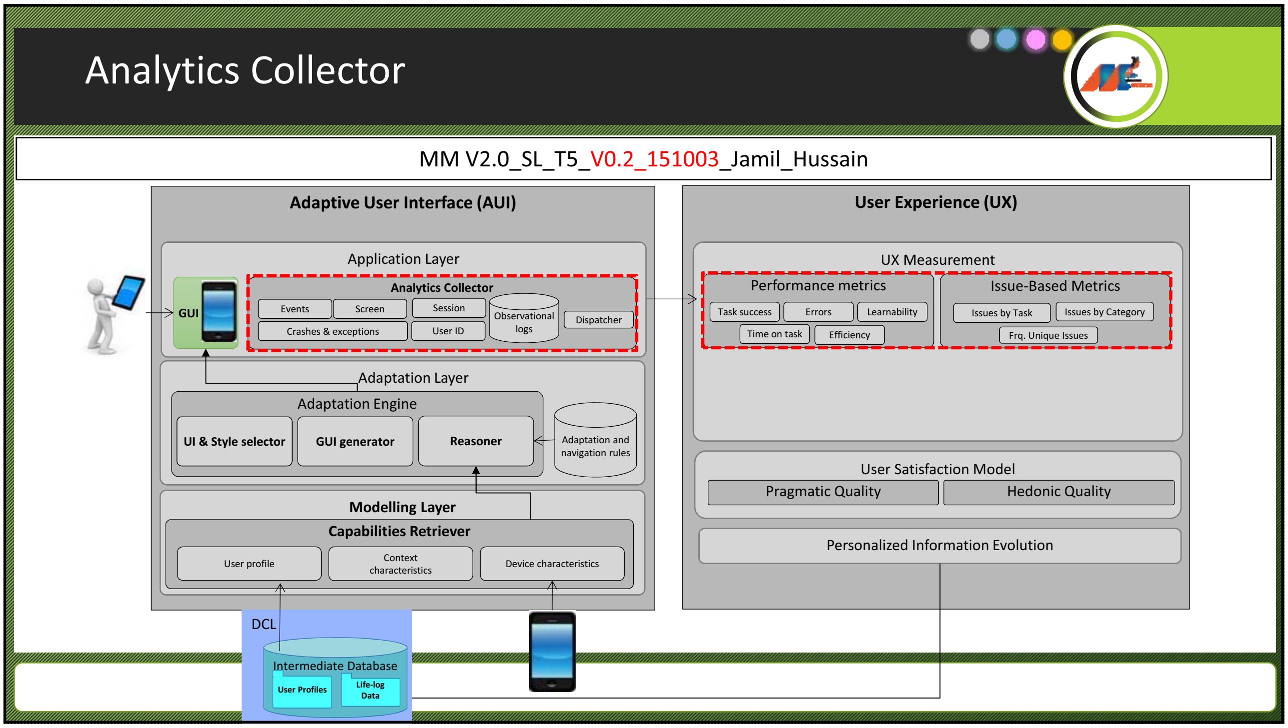
MM V2.0_SL_T5_V0.1_150926_Jamil_Hussain

- Android Studio installation and configuration
- Learned how to build interactive UI in android Studio
  - Layouts (Linear layout, relative layout, grid View, list view)
  - Input Controls, & Events
  - Menus & Action Bar
  - Styles and Themes
  - Accessibility
- Learned Google material design [1]
  - Color, Font, style, layout, component, patterns, usability
- Google analytics and Tag manager [2]
  - Create tracking ID at Google analytics
  - Download the Google Analytics SDK and integrated with app

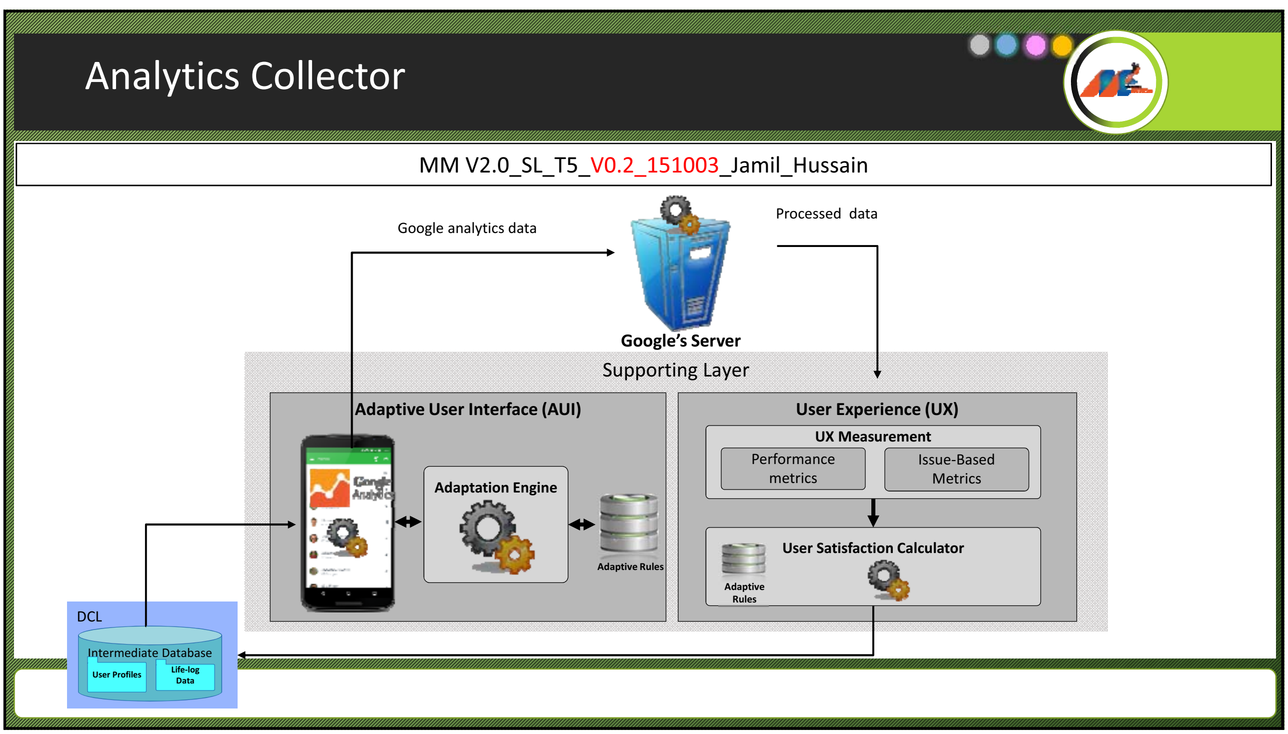
Analytics Collector – Week 2 Future Plan

MM V2.0_SL_T5_V0.1_151003_Jamil_Hussain

- Android Studio Learning – User Interface + App Components
  - Dialogs
  - Notifications
  - Drag and Drop
  - Custom Components
  - Intents and Intent Filters
  - Services
- Implementation of Google analytics events and tag manager containers
  - Implementation of Google analytics API for User Experience Calculation for (expert Panel)
  - Implantation of usability metrics calculator from collected data
- Wireframes for MM V2.0
  - Which adaptation aspects we are considered for MM V2.0 demo



Analytics Collector
MM V2.0_SL_T5_V0.2_151003_Jamil_Hussain
Adaptive User Interface (AUI)
Application Layer
GUI
Analytics Collector
Events
Screen
Session
Observational logs
Dispatcher
Crashes & exceptions
User ID
Adaptation Layer
Adaptation Engine
UI & Style selector
GUI generator
Reasoner
Adaptation and navigation rules
Modelling Layer
Capabilities Retriever
User profile
Context characteristics
Device characteristics
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
User Satisfaction Model
Pragmatic Quality
Hedonic Quality
Personalized Information Evolution
DCL
Intermediate Database
User Profiles
Life-log Data

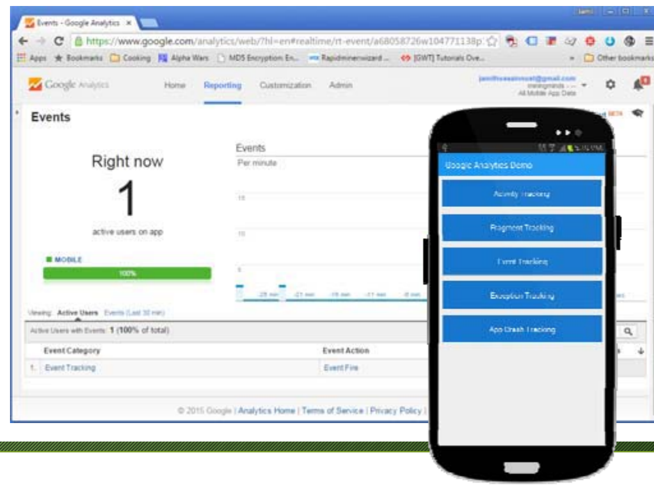


Analytics Collector
MM V2.0_SL_T5_V0.2_151003_Jamil_Hussain
Google analytics data
Processed data
Google's Server
Supporting Layer
Adaptive User Interface (AUI)
Adaptation Engine
Adaptive Rules
User Experience (UX)
UX Measurement
Performance metrics
Issue-Based Metrics
User Satisfaction Calculator
Adaptive Rules
DCL
Intermediate Database
User Profiles
Life-log Data

Analytics Collector – Week 2 Progress [1/3]

MM V2.0_SL_T5_V0.2_151003_Jamil_Hussain

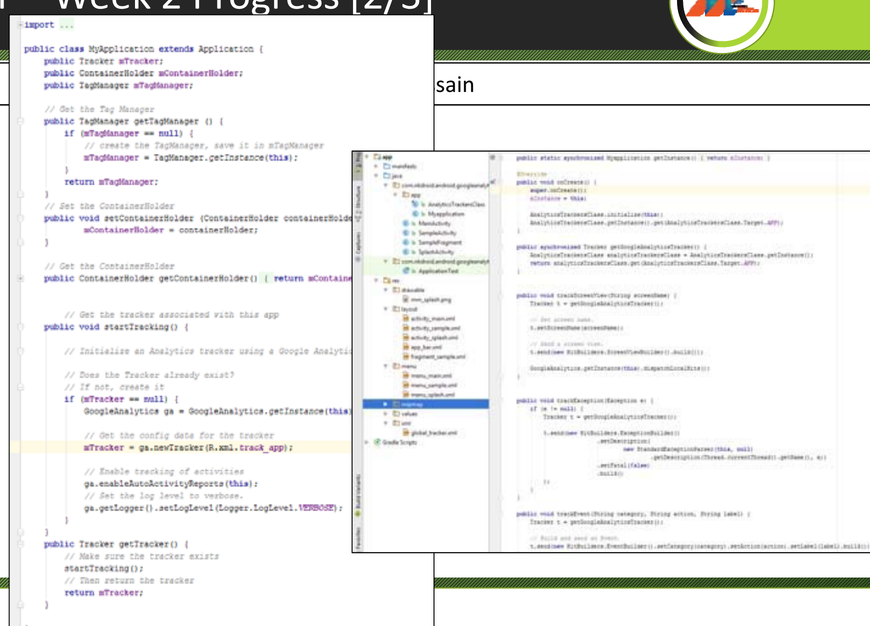
- Made class for Google Analytics Tracker
- All function associated to tracker are done such as
  - Track Events
  - Track Session
  - Track User_ID
  - Track Screen
  - Track Crashes & exceptions



Analytics Collector – Week 2 Progress [2/3]

sain

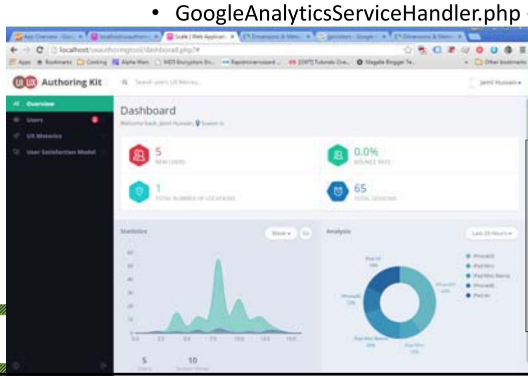
- Analytics Tracking integration Code
- Tracking Class Code



UX Measurement – Week 2 Progress [3/3]

MM V2.0_SL_T5_V0.2_151003_Jamil_Hussain

- Google Analytics Core API integration in UX Authoring Tool for expert
  - Class for Google Analytics Service Handling
    - GoogleAnalyticsServiceHandler.php {class name}



User Experience (UX)

UX Measurement

Performance metrics | Issue-Based Metrics

User Satisfaction Calculator

UX Measurement - Week 3 Future Plan

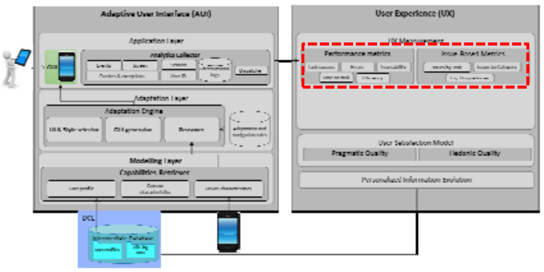
MM V2.0_SL_T5_V0.2_151003_Jamil_Hussain

- UX Expert Panel
  - Mockups for other pages
  - Additional functions for data retrieval from google analytics
  - Graphs selections for data visualization
  - Data preparation methods according to graphs
- Usability Metric Calculation - Performance metrics
- Android Learning

					OCTOBER					NOVEMBER				DECEMBER		
	Component	Task	Responsible	19-25Sep	26Sep-2Oct	3-9Oct	10-16Oct	17-23Oct	24-30Oct	31Oct-6Nov	7-13Nov	14-20Nov	21-27Nov	28Nov-4Dec	5-11Dec	12-18Dec
1	**Google analytics intergration in APP**	Tracker ID account creation and SDK intergration in android APP	Jamil Hussain													
		Data sending to google analtytics server for events such as Screen, Session, User ID, and others	Jamil Hussain													
2	**UX Mearument**	Data Collection from Google Analytics server using google	Jamil Hussaim													
		Usibility Metric Calcuation (Performance metrics) Usibility Metric Calcuation (Issue-Based Metrics)	Jamil Hussaim													
		User Satisfication Calculation Rules	Jamil Hussain, Wajahat ali Khan													
3	**Adaptation Engine**	Adaptation Rules Creations	Jamil Hussain, Wajahat ali Khan													
		Reasoner Implemtation	Jamil Hussain													
		Adaptive GUI generator	Jamil Hussain													
4	**Intergration**	Overall Intergration and Testing	Jamil Hussain, Wajahat ali Khan													

UX Measurement - Week 3 Progress

MM V2.0_SL_T5_V0.3_151009_Jamil_Hussain

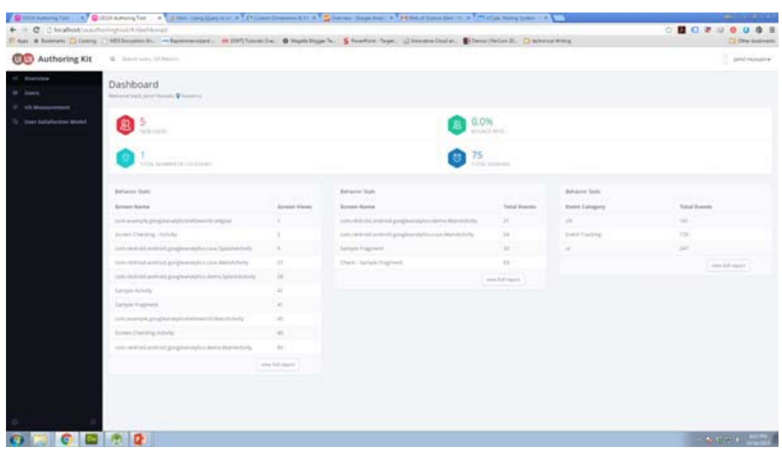


					OCTOBER					NOVEMBER				DECEMBER		
	Component	Task	Responsible	19-25Sep	26Sep-2Oct	3-9Oct	10-16Oct	17-23Oct	24-30Oct	31Oct-6Nov	7-13Nov	14-20Nov	21-27Nov	28Nov-4Dec	5-11Dec	12-18Dec
1	Google analytics intergration in APP	Tracker ID account creation and SDK intergration in android APP	Jamil Hussain													
		Data sending to google analtytics server for events such as Screen, Session, User ID, and others	Jamil Hussain													
2	UX Mearument	Data Collection from Google Analtyics server using google	Jamil Hussaim													
		Usibility Metric Calcuation (Performance metrics) Usibility Metric Calcuation (Issue-Based Metrics)	Jamil Hussaim													
		User Satisfication Calculation Rules	Jamil Hussain, Wajahat ali Khan													
3	Adaptation Engine	Adaptation Rules Creations	Jamil Hussain, Wajahat ali Khan													
		Reasoner Implemtation	Jamil Hussain													
		Adaptive GUI generator	Jamil Hussain													
4	Intergration	Overall Intergration and Testing	Jamil Hussain, Wajahat ali Khan													

UX Measurement - Week 3 Progress

MM V2.0_SL_T5_V0.3_151009_Jamil_Hussain

- UX Expert Panel
  - App Tracking
    - Screen Analysis (Time on Screen, screen depth, events on screen etc.)
    - Preference Metrics for app tracking dimensions

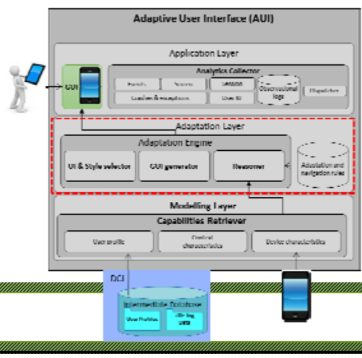


http://163.180.116.121/uxauthoringtool/

UX Measurement & Rule Engine - Week 4 Future Plan

MM V2.0_SL_T5_V0.3_151009_Jamil_Hussain

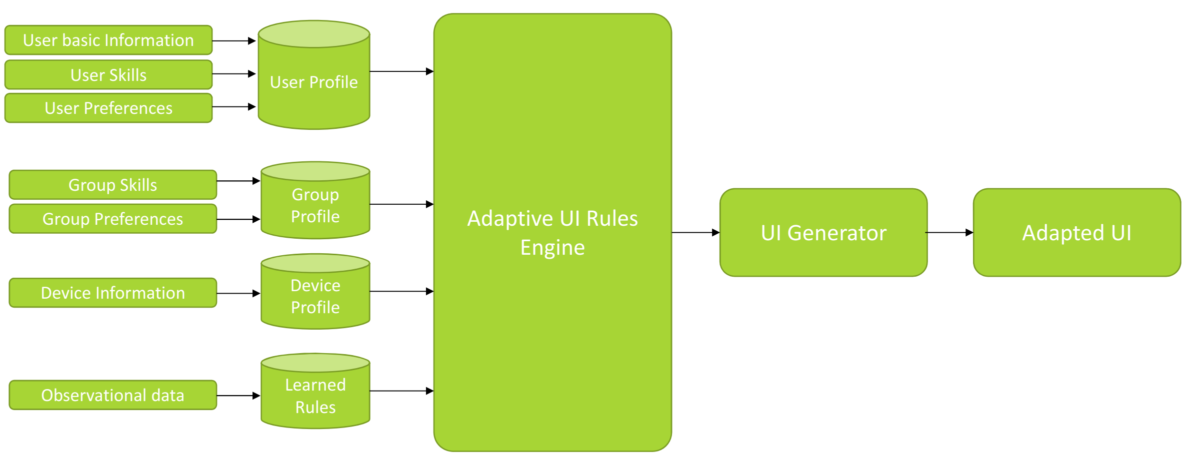
- Rule Engine Framework setup
- Rule Format setting
- Performance metrics



				OCTOBER						NOVEMBER				DECEMBER		
	Component	Task	Responsible	19-25Sep	26Sep-2Oct	3-9Oct	10-16Oct	17-23Oct	24-30Oct	31Oct-6Nov	7-13Nov	14-20Nov	21-27Nov	28Nov-4Dec	5-11Dec	12-18Dec
1	Google analytics intergration in APP	Tracker ID account creation and SDK intergration in android APP	Jamil Hussain													
		Data sending to google analtytics server for events such as Screen, Session, User ID, and others	Jamil Hussain													
2	UX Mearument	Data Collection from Google Analtyics server using google	Jamil Hussaim													
		Usibility Metric Calcuation (Performance metrics) Usibility Metric Calcuation (Issue-Based Metrics)	Jamil Hussaim													
		User Satisfication Calculation Rules	Jamil Hussain, Wajahat ali Khan													
3	Adaptation Engine	Adaptation Rules Creations	Jamil Hussain, Wajahat ali Khan													
		Reasoner Implemtation	Jamil Hussain													
		Adaptive GUI generator	Jamil Hussain													
4	Intergration	Overall Intergration and Testing	Jamil Hussain, Wajahat ali Khan													

Adaptation Rule Engine - Week 4 Progress

MM V2.0_SL_T5_V0.4_151016_Jamil_Hussain



Adaptation Rule Engine - Week 4 Progress

MM V2.0_SL_T5_V0.4_151016_Jamil_Hussain

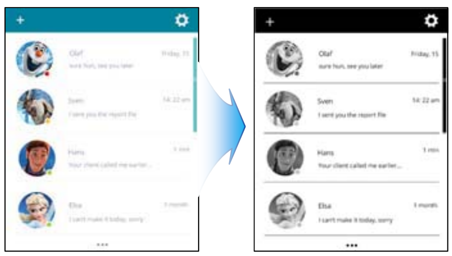
Rule Format: Event / Condition / Action

Example

Event: On Render

Condition : The user has color-blind

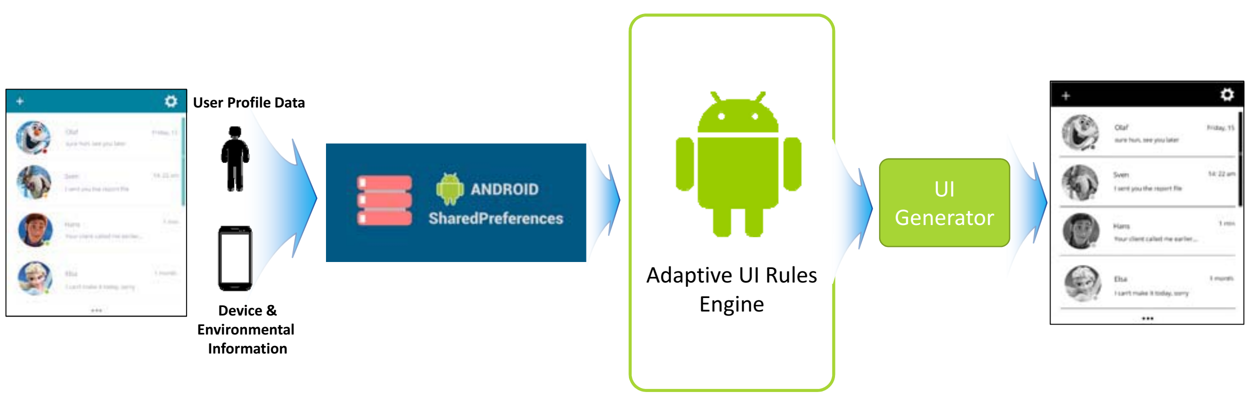
Action: Change the foreground color to white and the background color to black in order to provide a high-contrast UI.



```java
/**
 * Created by jamil on 10/10/2015.
 */
import org.easyrules.annotation.Action;
import org.easyrules.annotation.Condition;
import org.easyrules.annotation.Priority;
import org.easyrules.annotation.Rule;

@Rule(name = "ColourBlindRule",
        description = "Check if person's is colour-blind and set the UI accordingly")

public class ColourBlindRule {

    Users person;
    private Boolean colorBlind=true;

    @Condition
    public boolean isColourBlind() {

        return  person.getColourBlind() == colorBlind;
    }
    @Action
    public void markAsLowVision(){

        AdaptUI AUI=new AdaptUI();
        AUI.setColorBlind();
    }

}
```


Adaptation Rule Engine - Week 4 Progress

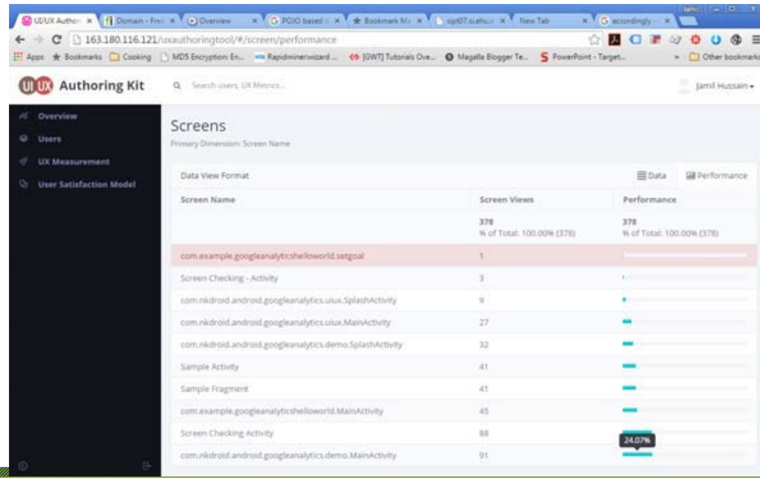
MM V2.0_SL_T5_V0.4_151016_Jamil_Hussain



Performance metrics (Screen View) - Week 4 Progress

MM V2.0_SL_T5_V0.4_151016_Jamil_Hussain

- Usability Metric Calculation
  - Performance metrics

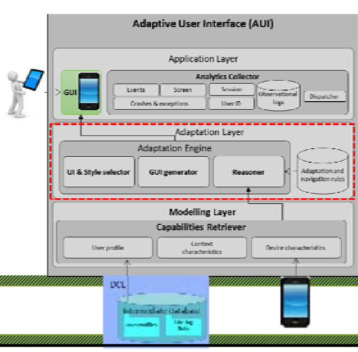


http://163.180.116.121/uxauthoringtool/#/screen/performance

Adaptive UI - Week 5 Future Plan

MM V2.0_SL_T5_V0.4_151016_Jamil_Hussain

- Adaption Rules Creation
- Adaptive GUI Engine implementation
- UI/UX Authoring Tool Expert View completion



				OCTOBER						NOVEMBER				DECEMBER		
	Component	Task	Responsible	19-25Sep	26Sep-2Oct	3-9Oct	10-16Oct	17-23Oct	24-30Oct	31Oct-6Nov	7-13Nov	14-20Nov	21-27Nov	28Nov-4Dec	5-11Dec	12-18Dec
1	**Google analytics intergration in APP**	Tracker ID account creation and SDK intergration in android APP	Jamil Hussain													
		Data sending to google analtytics server for events such as Screen, Session, User ID, and others	Jamil Hussain													
2	**UX Mearument**	Data Collection from Google Analytics server using google	Jamil Hussaim													
		Usibility Metric Calcuation (Performance metrics) Usibility Metric Calcuation (Issue-Based Metrics)	Jamil Hussaim													
		User Satisfication Calculation Rules	Jamil Hussain, Wajahat ali Khan													
3	**Adaptation Engine**	Adaptation Rules Creations	Jamil Hussain, Wajahat ali Khan													
		Reasoner Implemtation	Jamil Hussain													
		Adaptive GUI generator	Jamil Hussain													
4	**Intergration**	Overall Intergration and Testing	Jamil Hussain, Wajahat ali Khan													

UX Measurement - Week 3 Future Plan

MM V2.0_SL_T5_V0.4_151016_Jamil_Hussain

- UX Expert Panel
  - Mockups for other pages
  - Additional functions for data retrieval from google analytics
  - Graphs selections for data visualization
  - Data preparation methods according to graphs
- Usability Metric Calculation - Performance metrics
- Android Learning

					OCTOBER					NOVEMBER				DECEMBER		
	Component	Task	Responsible	19-25Sep	26Sep-2Oct	3-9Oct	10-16Oct	17-23Oct	24-30Oct	31Oct-6Nov	7-13Nov	14-20Nov	21-27Nov	28Nov-4Dec	5-11Dec	12-18Dec
1	**Google analytics intergration in APP**	Tracker ID account creation and SDK intergration in android APP	Jamil Hussain													
		Data sending to google analtytics server for events such as Screen, Session, User ID, and others	Jamil Hussain													
2	**UX Mearument**	Data Collection from Google Analytics server using google	Jamil Hussaim													
		Usibility Metric Calcuation (Performance metrics) Usibility Metric Calcuation (Issue-Based Metrics)	Jamil Hussaim													
		User Satisfication Calculation Rules	Jamil Hussain, Wajahat ali Khan													
3	**Adaptation Engine**	Adaptation Rules Creations	Jamil Hussain, Wajahat ali Khan													
		Reasoner Implemtation	Jamil Hussain													
		Adaptive GUI generator	Jamil Hussain													
4	**Intergration**	Overall Intergration and Testing	Jamil Hussain, Wajahat ali Khan													

Recap of the last 7 Weeks

MM V2.0_SL_T5_V0.7_151113_Jamil_Hussain

- **Week 1**
  - Development environment setup
    - (tools selection, installation etc.)
  - Account creation at google analytics and tag Manager
    - To track the user behavior data
- **Week 2**
  - Google analytics services API integration in applications and its core function implementations
    - Track Events, Session, User ID, Screen and Crashes & exceptions
  - Google Analytics Core API integration in UX Authoring Tool for expert
- **Week 3**
  - UX Expert Panel
  - App Tracking
    - Screen Analysis (Time on Screen, screen depth, events on screen etc.)
    - Preference Metrics for app tracking dimensions
- **Week 4**
  - Rule Engine Framework setup
  - Rule Format setting
  - Performance metrics
- **Week 5**
  - Testing rule-based reasoner
    - Rule creation and testing
- **Week 6**
  - Data fetching and sending clients code
  - Themes Setting against each rules
- **Week 8**
  - Demo

UI/UX Authoring Tool - Purpose

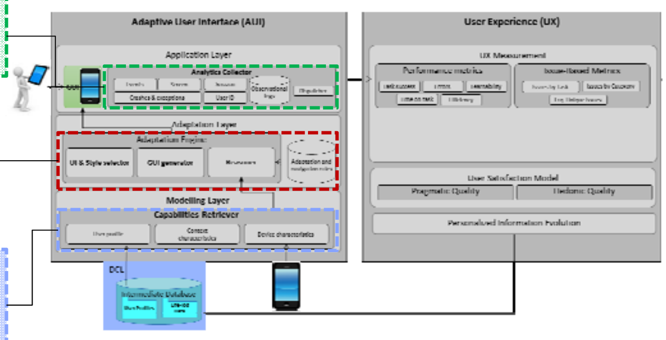
MM V2.0_SL_T5_V0.7_151113_Jamil_Hussain

Purpose: Show, how the UI is adapted based on user , environmental and device data, in order to improve the User experience by
- **Increasing the accessibility of information**
- **Analytics data Collection for UI/UX Authoring Expert Panel**

Analytics Collector
- **User interaction behavioral data collection**
  - ✓ **To measure the Pragmatic qualities**

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**



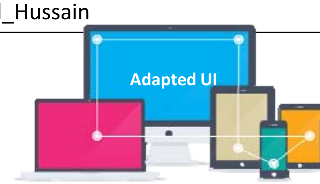
Adaptation Engine

MM V2.0_SL_T5_V0.7_151113_Jamil_Hussain

User Profile Data → Adaptive UI Rules Engine

UI adaptive Rules → Adaptive UI Rules Engine

Adaptive UI Rules Engine → UI Generator



UI adaptive Rules
- **Adaptation rules**
- **Used Objected oriented approach for rules creation**

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.

경희대학교 KYUNG HEE UNIVERSITY

http://www.easyrules.org/index.html

Adaptation Scenarios

MM V2.0_SL_T5_V0.7_151113_Jamil_Hussain

Adaptation Rules

#	Target User	Adaptation Type
1	Low Vision User	Adapted UI
2	Colour Blind User	Adapted UI
3	Paraplegia	Adaptive UI
4	Mild Deafness	Adaptive UI
5	Cognitive	Adaptive UI

Rule 1

Event: the size of the text of the UI is smaller than 16 dp
Condition: the user has low vision
Action: increase the size of the text of the UI to 20 dp

Rule 2

Event: onRender
Condition: the user is color-blind
Action: change the foreground color to black and background color to white interaction element.

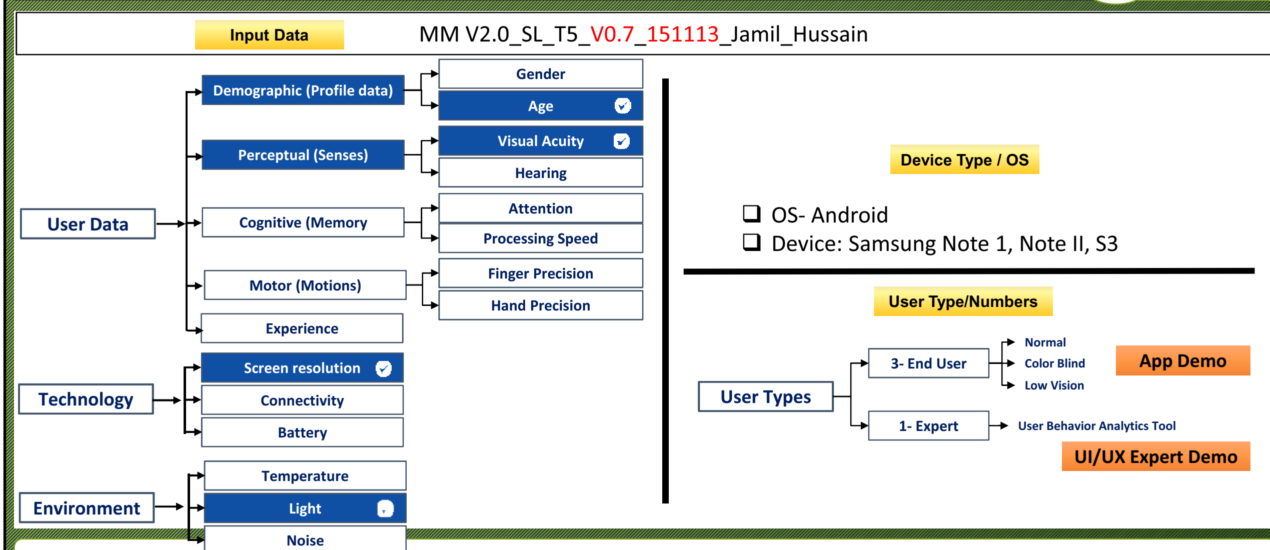
Rule 3

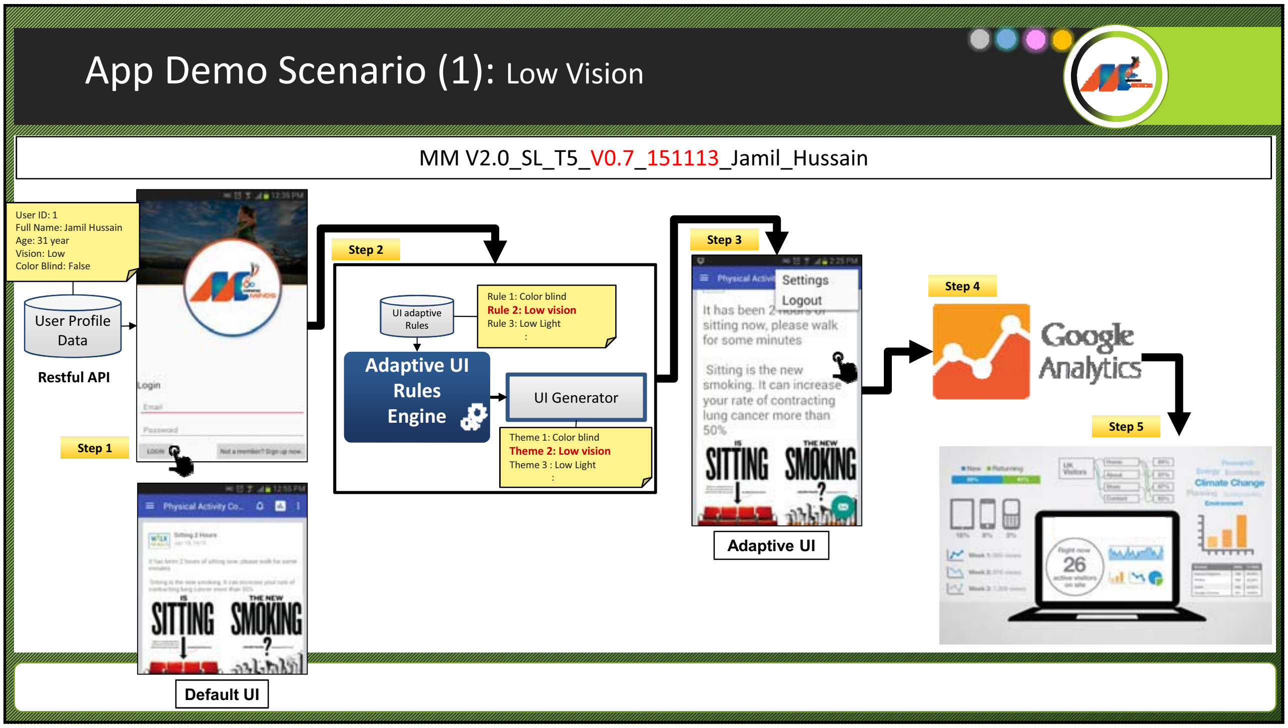
Event: Based on light sensor lux values
Condition: The light level is too low
Action: the user interface is changed to night mood.

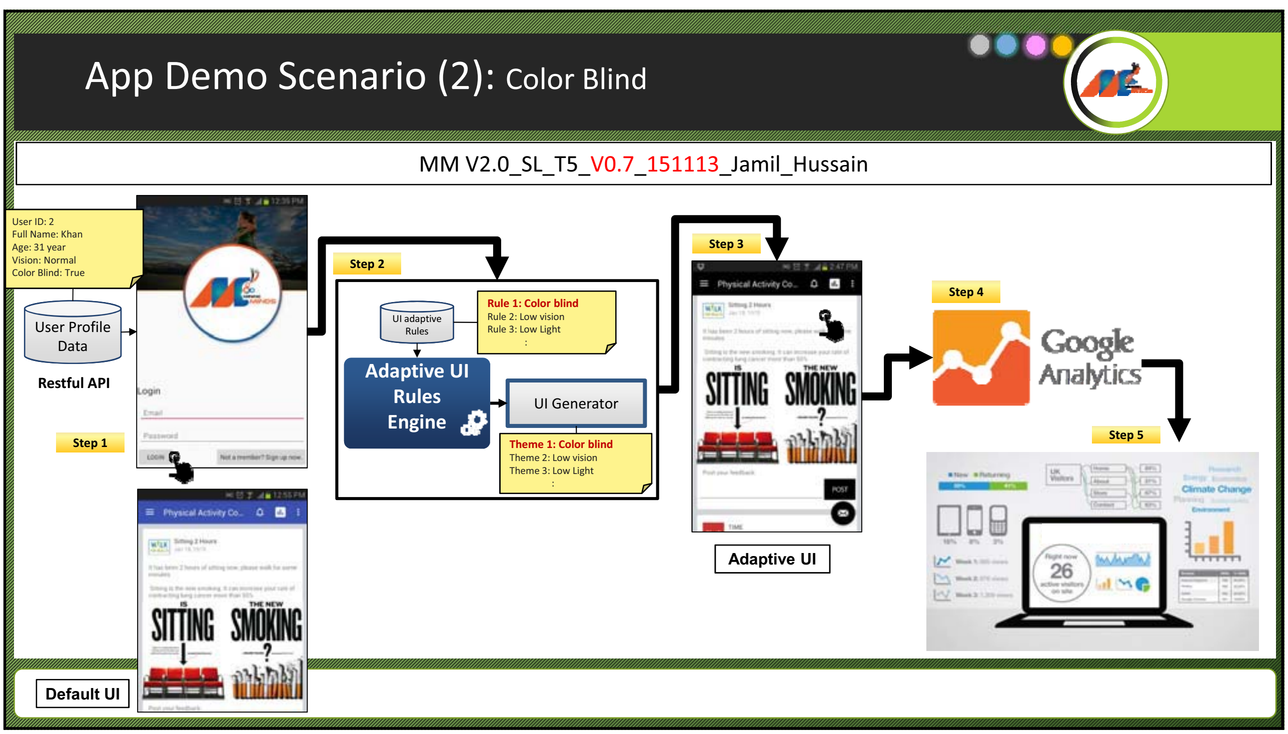
Miñón, Raúl, et al. "Integrating adaptation rules for people with special needs in model-based UI development process." *Universal Access in the Information Society*: 1-16.
http://sipt07.si.ehu.es/aptrep/html/adaptation_rules.html

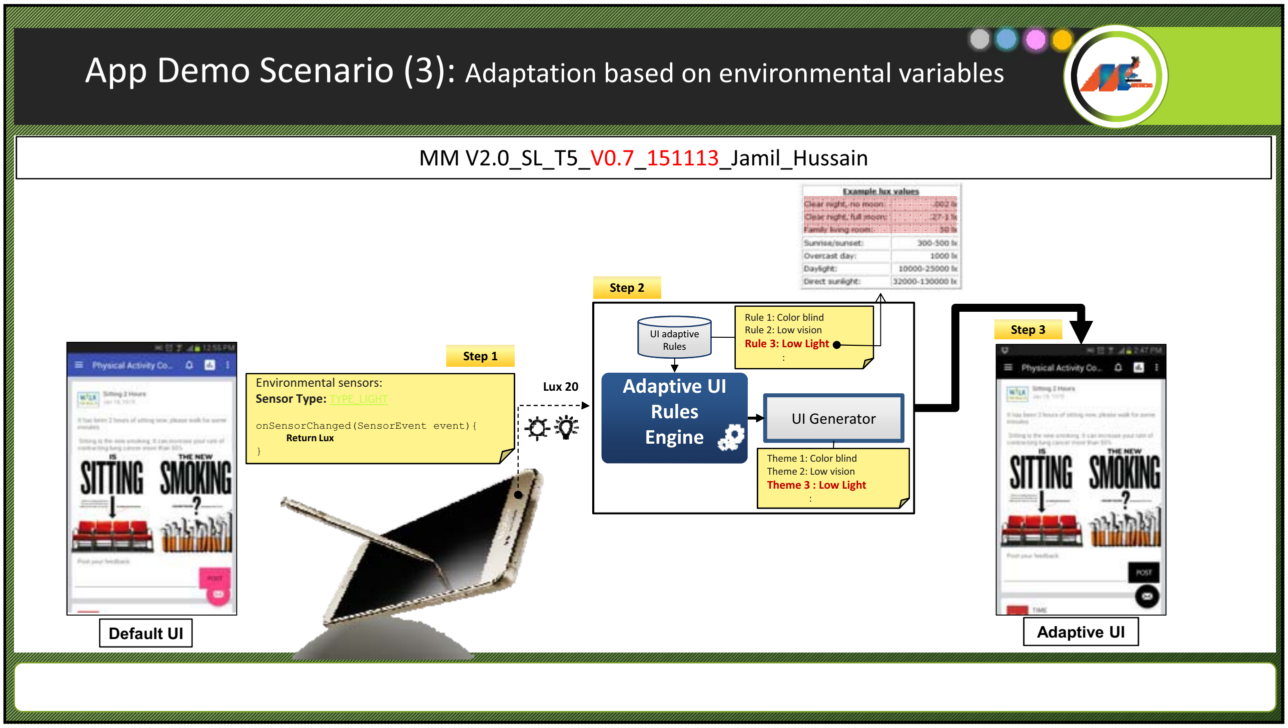
Scope

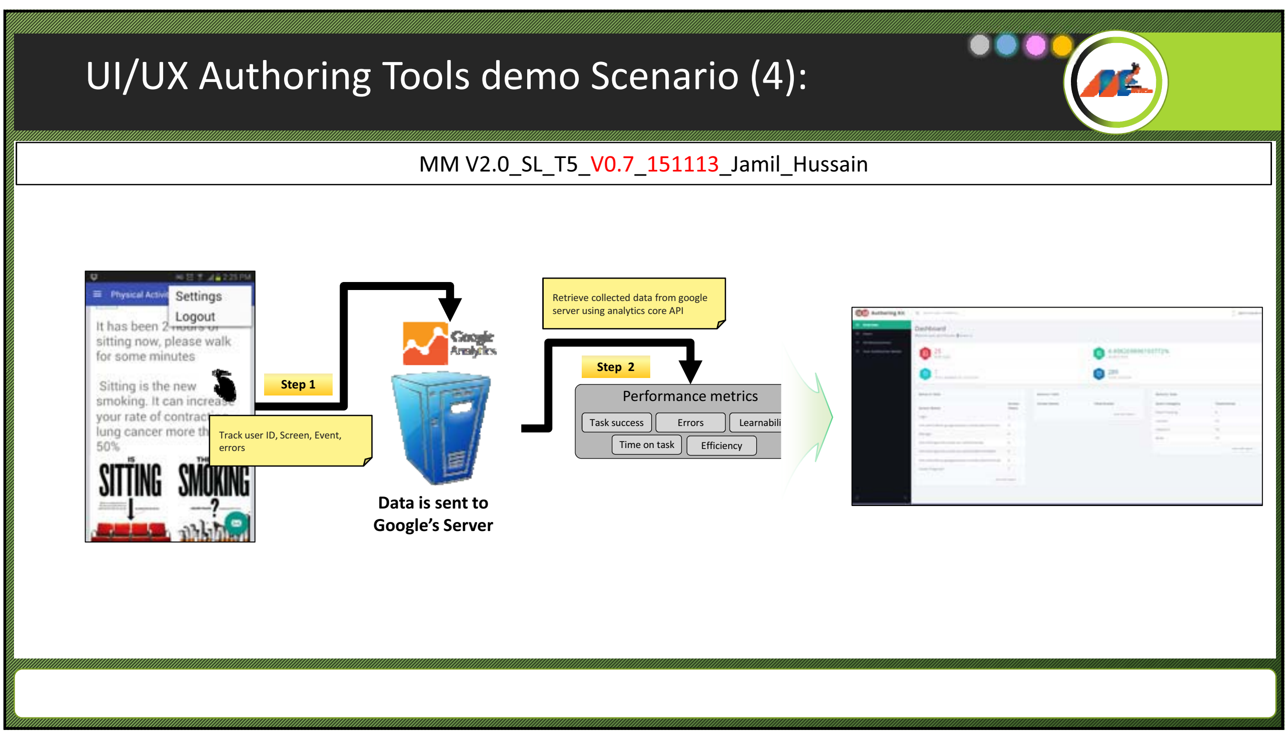
MM V2.0_SL_T5_V0.7_151113_Jamil_Hussain




App Demo Scenario (1): Low Vision
MM V2.0_SL_T5_V0.7_151113_Jamil_Hussain
User ID: 1
Full Name: Jamil Hussain
Age: 31 year
Vision: Low
Color Blind: False
User Profile Data
Restful API
Step 1
Login
Default UI
Step 2
UI adaptive Rules
Rule 1: Color blind
Rule 2: Low vision
Rule 3: Low Light
Adaptive UI Rules Engine
UI Generator
Theme 1: Color blind
Theme 2: Low vision
Theme 3 : Low Light
Step 3
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
Adaptive UI
Step 4
Google Analytics
Step 5


App Demo Scenario (2): Color Blind
MM V2.0_SL_T5_V0.7_151113_Jamil_Hussain
User ID: 2
Full Name: Khan
Age: 31 year
Vision: Normal
Color Blind: True
User Profile Data
Restful API
Step 1
Login
Default UI
Step 2
UI adaptive Rules
Rule 1: Color blind
Rule 2: Low vision
Rule 3: Low Light
Adaptive UI Rules Engine
UI Generator
Theme 1: Color blind
Theme 2: Low vision
Theme 3: Low Light
Step 3
SITTING
SMOKING
Adaptive UI
Step 4
Google Analytics
Step 5


App Demo Scenario (3): Adaptation based on environmental variables
MM V2.0_SL_T5_V0.7_151113_Jamil_Hussain
Example lux values
Step 1
Environmental sensors:
Sensor Type: TYPE_LIGHT
onSensorChanged(SensorEvent event){
Return Lux
}
Lux 20
Step 2
UI adaptive Rules
Rule 1: Color blind
Rule 2: Low vision
Rule 3: Low Light
Adaptive UI Rules Engine
UI Generator
Theme 1: Color blind
Theme 2: Low vision
Theme 3 : Low Light
Step 3
Default UI
Adaptive UI


UI/UX Authoring Tools demo Scenario (4):
MM V2.0_SL_T5_V0.7_151113_Jamil_Hussain
Settings
Logout
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
Time on task
Efficiency

Behavior Change

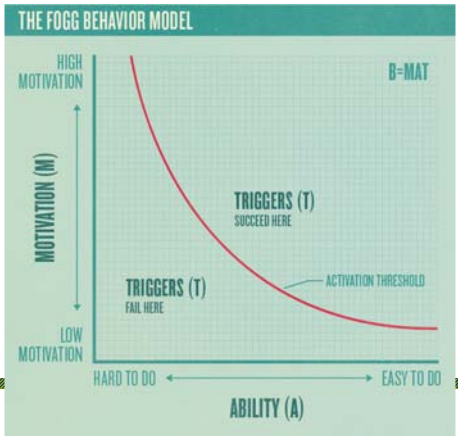
MM V2.0_SL_T5_V0.8_151128_Jamil_Hussain

- A large portion of the current research on behavior change focuses on what goes on inside of people's heads as they consciously decide to act. The two most prominent models were:
  - **Ajzen's Theory of Planned Behavior :**
    Focuses on how our intentions to act are formed as a product of attitudes, norms, and perceived control over the behavior
  - **Prochaska and Velicer's Transtheoretical Model :**
    Looks at the stages of change that a person goes through from starting to anticipate an action to changing the behavior and maintaining it.
- In 2009 BJ Fogg's proposed a model that is used in behavior change product design called **The Fogg Behavior Model (FBM)**

The Fogg Behavior Model (FBM)

MM V2.0_SL_T5_V0.8_151128_Jamil_Hussain

The Fogg Behavior Model (FBM) shows that three elements must meet at the same movement for the behavior to occur: Motivation, Ability, and Trigger (MAT)



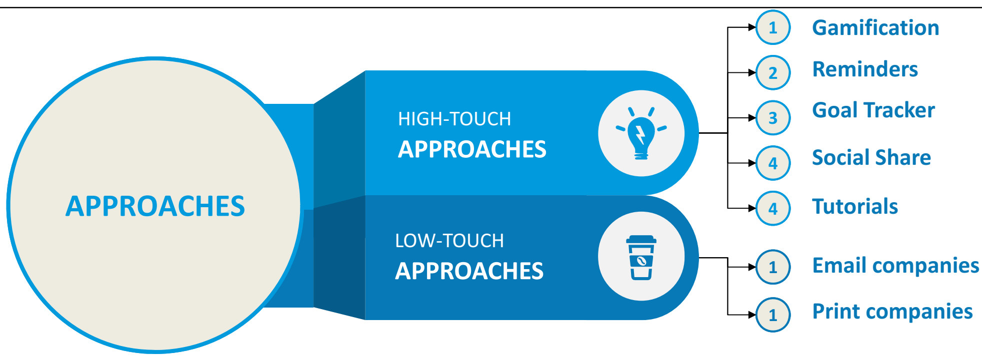
Product Design Patterns for Behavior Change

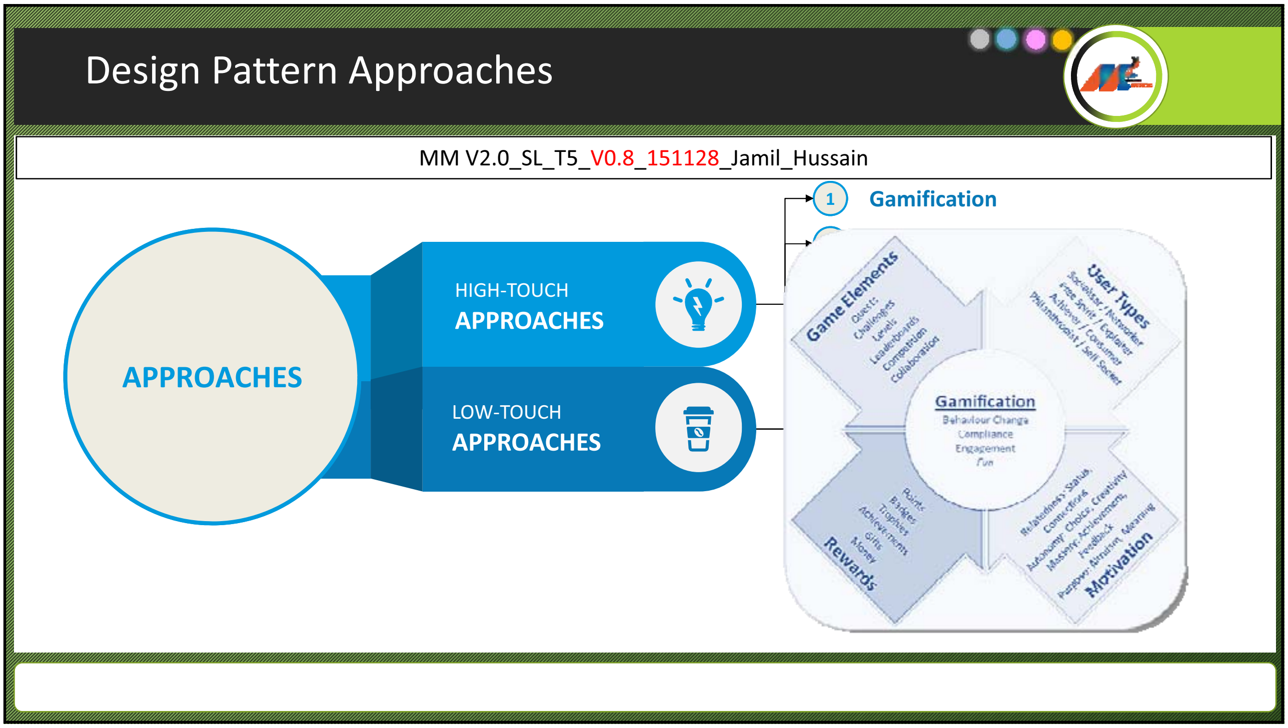
MM V2.0_SL_T5_V0.7_151128_Jamil_Hussain

- In the behavior change space, a variety of templates/patterns have already been established that can be used for mobile and online applications.
- Often a given product employs more than one of these templates/patterns in different parts of the application.
- Each pattern provides guidelines for promoting behavior change in a particular context, and a template for the look and feel of that part of the application

Design Pattern Approaches

MM V2.0_SL_T5_V0.8_151128_Jamil_Hussain



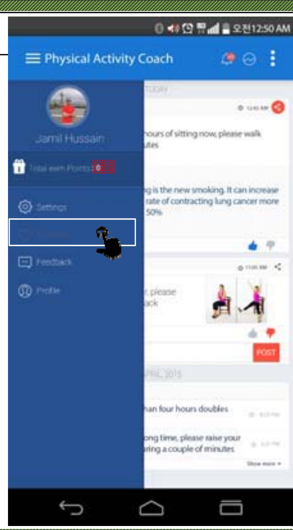

Design Pattern Approaches
MM V2.0_SL_T5_V0.8_151128_Jamil_Hussain
1
Gamification
APPROACHES
HIGH-TOUCH
APPROACHES
LOW-TOUCH
APPROACHES
Game Elements
User Types
Gamification
Rewards
Motivation


Our approach to behavior change design
Behavior Change
Adaptation
Rewards
Self-Reporting
Group-reporting
Analytics (Usage)
Series 1
Series 2
Series 3
10
0

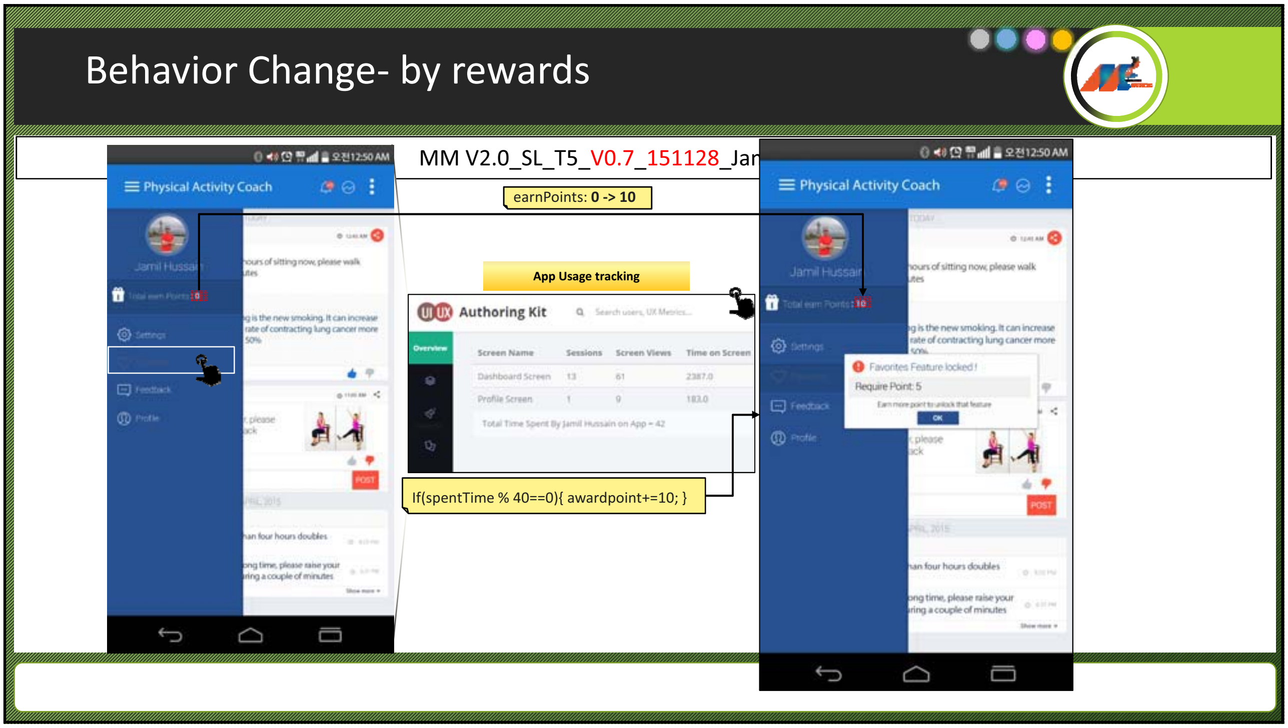
Our approach to behavior change design

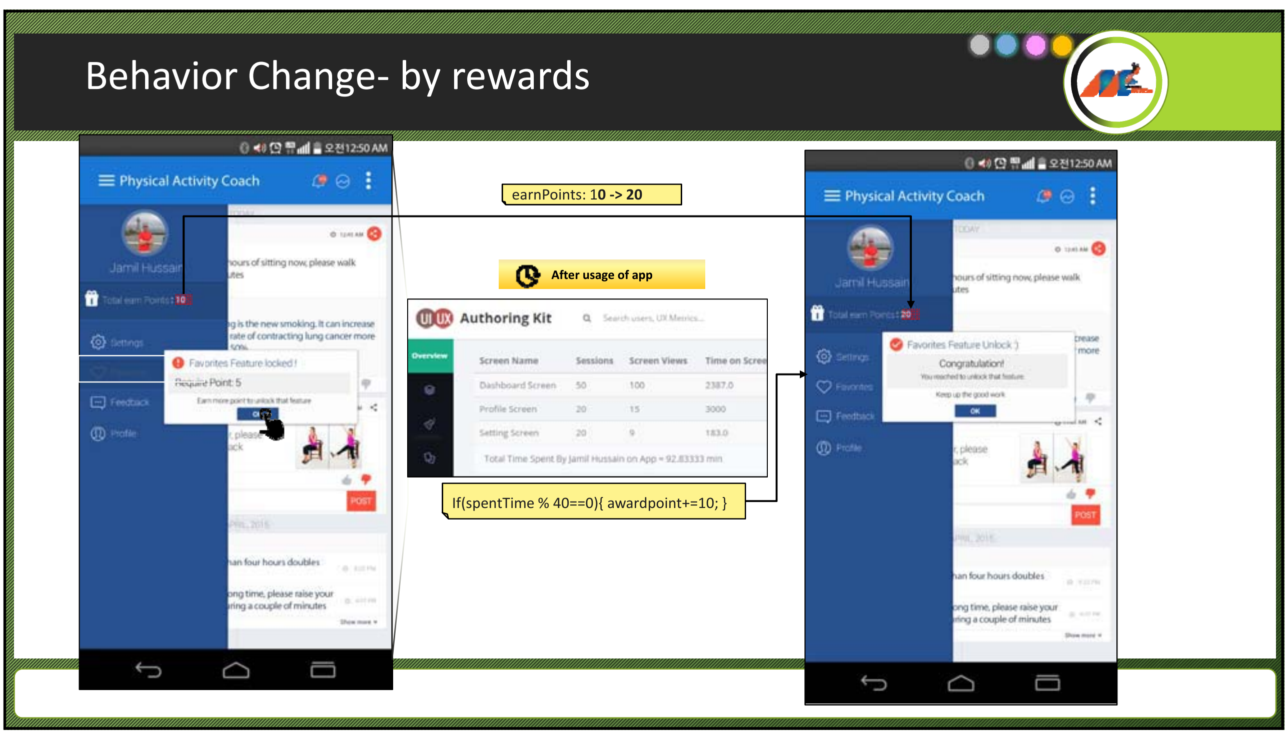
Behavior Change- by rewards

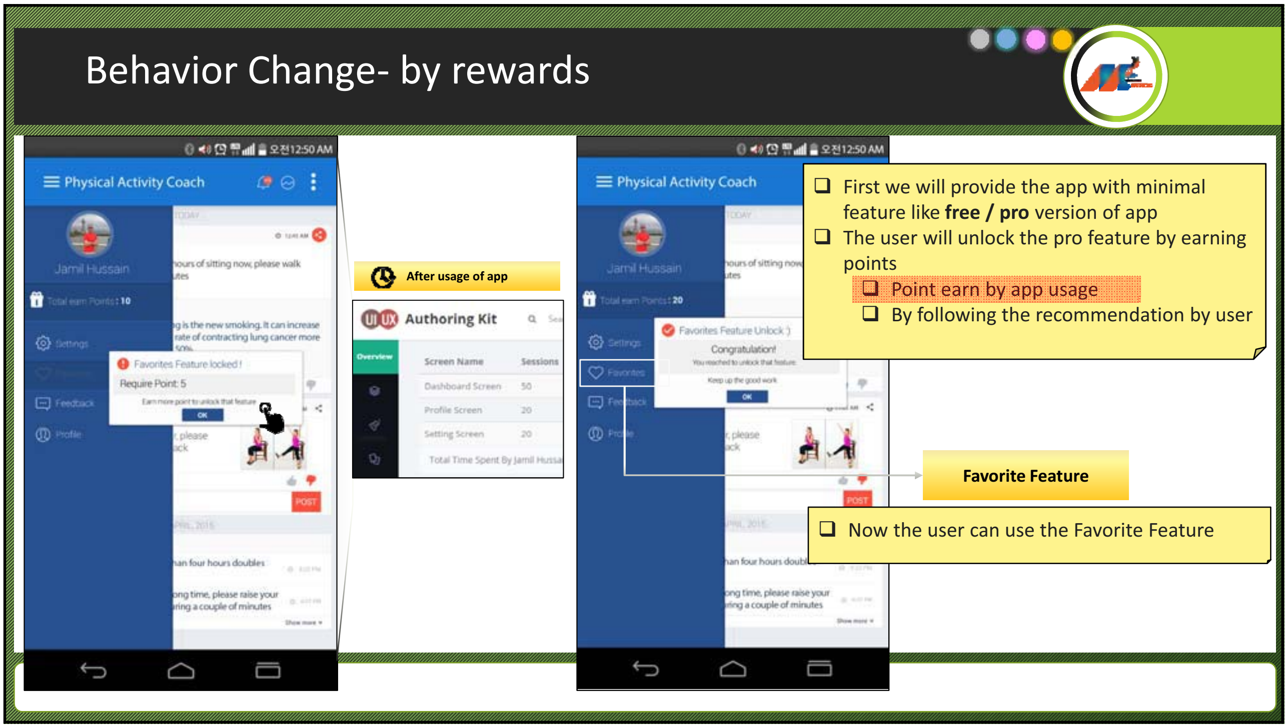
0.8_151128_Jamil_Hussain



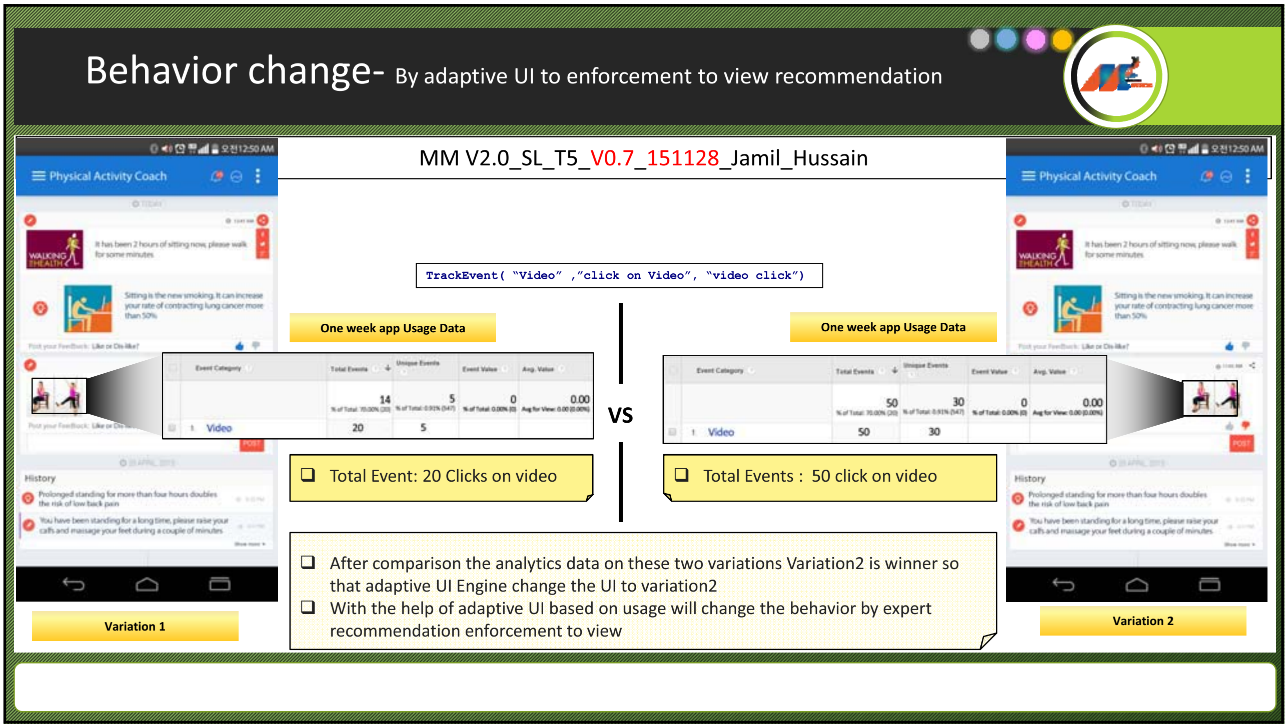
- First we will provide the app with minimal feature like **free / pro** version of app
- The user will unlock the pro feature by earning points
  - Point earn by app usage
  - By following the recommendation


Behavior Change- by rewards
MM V2.0_SL_T5_V0.7_151128_Jan
earnPoints: 0 -> 10
App Usage tracking
Authoring Kit
If(spentTime % 40==0){ awardpoint+=10; }
Physical Activity Coach
Physical Activity Coach

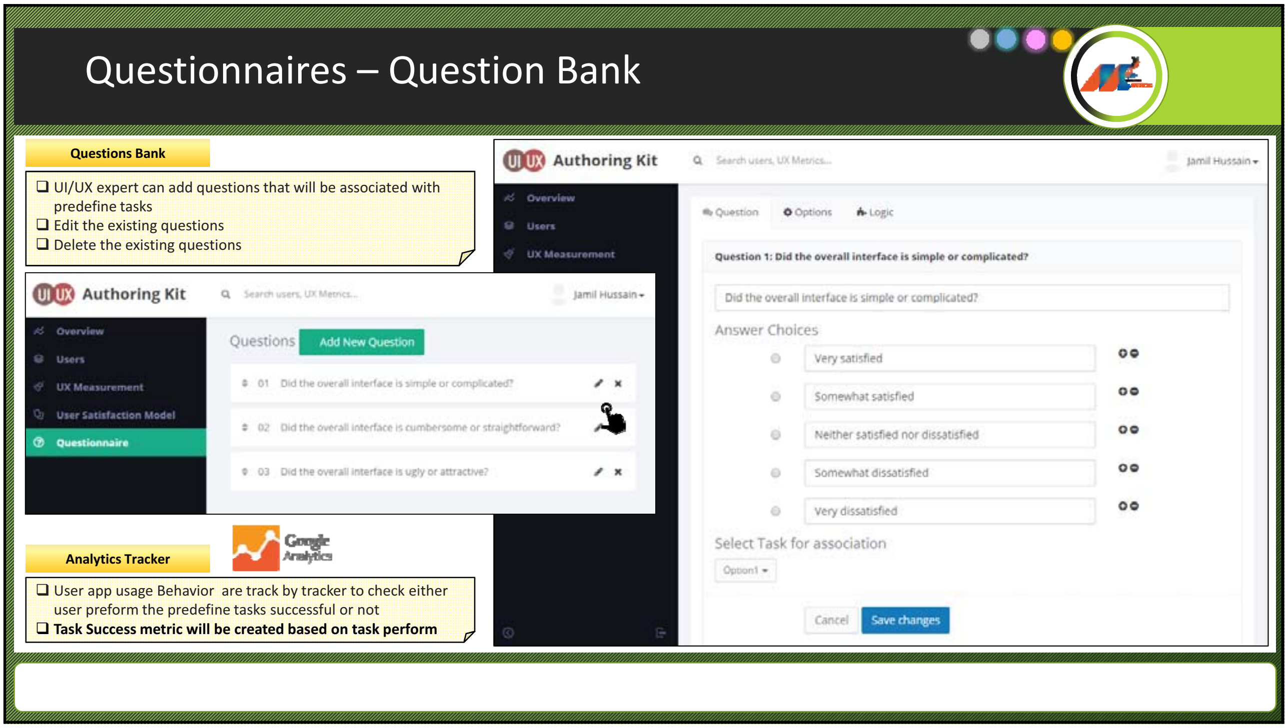

Behavior Change- by rewards
earnPoints: 10 -> 20
After usage of app
Authoring Kit
If(spentTime % 40==0){ awardpoint+=10; }
Physical Activity Coach
Physical Activity Coach

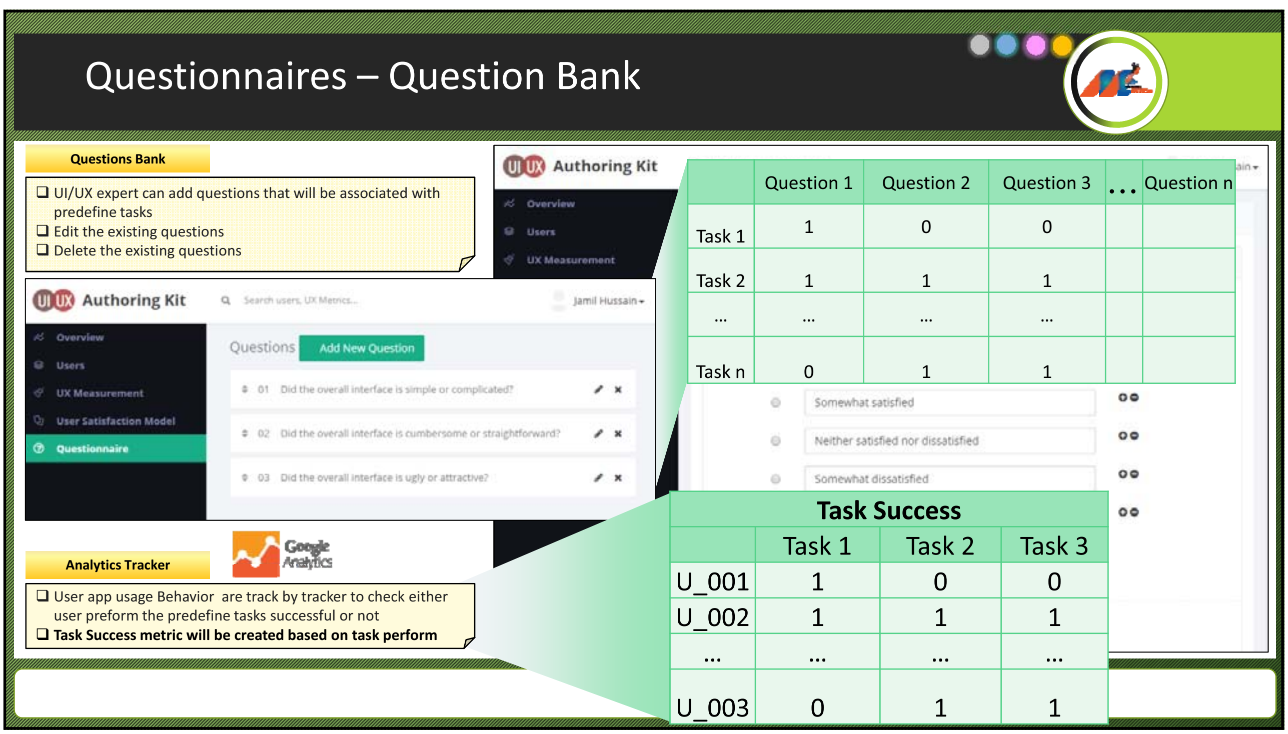

Behavior Change- by rewards
Physical Activity Coach
Jamil Hussain
Favorites Feature locked !
Require Point: 5
After usage of app
Authoring Kit
Overview
Screen Name
Sessions
Dashboard Screen
50
Profile Screen
20
Setting Screen
20
Favorites Feature Unlock :)
Congratulation!
First we will provide the app with minimal feature like free / pro version of app
The user will unlock the pro feature by earning points
Point earn by app usage
By following the recommendation by user
Favorite Feature
Now the user can use the Favorite Feature

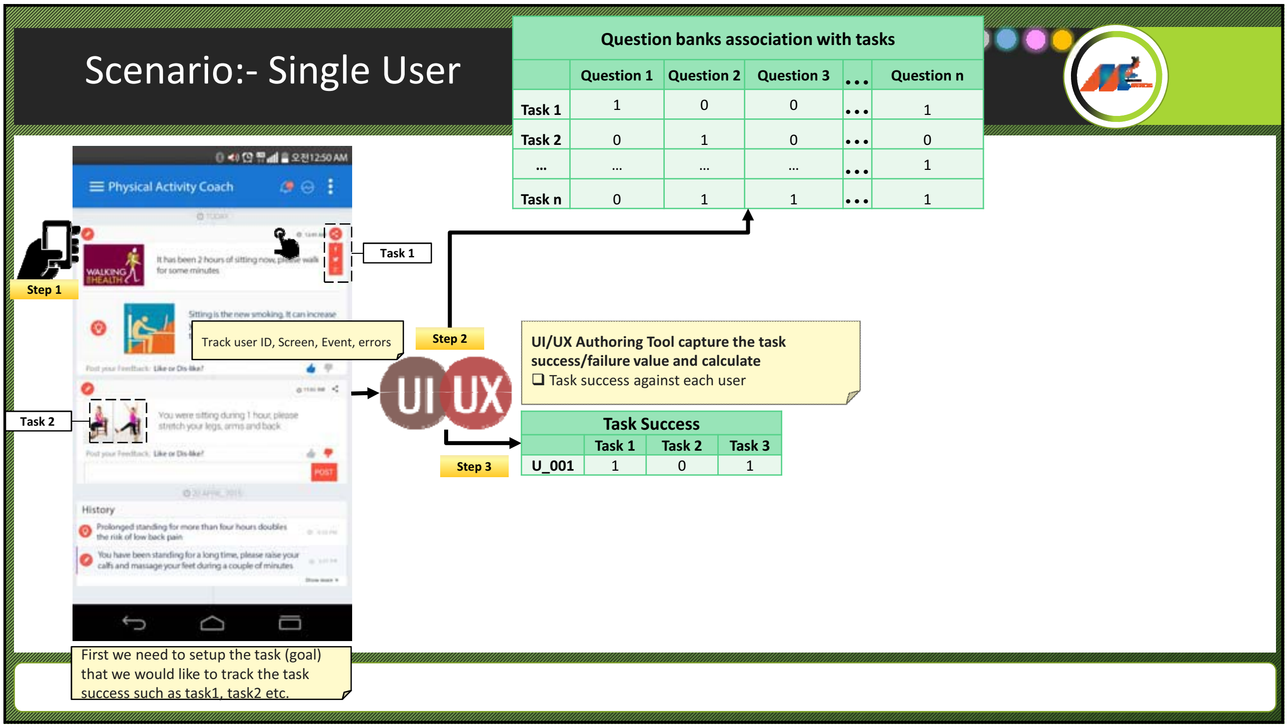

Our approach to behavior change design
Behavior Change
Adaptation
Series 1
Series 2
Series 3
10
0
Rewards
Self-Reporting
Group-reporting
Analytics (Usage)


Behavior change- By adaptive UI to enforcement to view recommendation
MM V2.0_SL_T5_V0.7_151128_Jamil_Hussain
Physical Activity Coach
TrackEvent("Video" ,"click on Video", "video click")
One week app Usage Data
One week app Usage Data
VS
Video
Total Event: 20 Clicks on video
Total Events : 50 click on video
After comparison the analytics data on these two variations Variation2 is winner so that adaptive UI Engine change the UI to variation2
With the help of adaptive UI based on usage will change the behavior by expert recommendation enforcement to view
History
Variation 1
Variation 2


Our approach to behavior change design
Behavior Change
Adaptation
Rewards
Self-Reporting
Group-reporting
Analytics (Usage)
Series 1
Series 2
Series 3


Questionnaires – Question Bank
Questions Bank
UI/UX expert can add questions that will be associated with predefine tasks
Edit the existing questions
Delete the existing questions
Authoring Kit
Overview
Users
UX Measurement
User Satisfaction Model
Questionnaire
Questions
Add New Question
01 Did the overall interface is simple or complicated?
02 Did the overall interface is cumbersome or straightforward?
03 Did the overall interface is ugly or attractive?
Analytics Tracker
Google Analytics
User app usage Behavior are track by tracker to check either user preform the predefine tasks successful or not
Task Success metric will be created based on task perform
Question
Options
Logic
Question 1: Did the overall interface is simple or complicated?
Did the overall interface is simple or complicated?
Answer Choices
Very satisfied
Somewhat satisfied
Neither satisfied nor dissatisfied
Somewhat dissatisfied
Very dissatisfied
Select Task for association
Option1
Cancel
Save changes


Questionnaires – Question Bank
Questions Bank
UI/UX expert can add questions that will be associated with predefine tasks
Edit the existing questions
Delete the existing questions
Authoring Kit
Overview
Users
UX Measurement
User Satisfaction Model
Questionnaire
Questions
Add New Question
01 Did the overall interface is simple or complicated?
02 Did the overall interface is cumbersome or straightforward?
03 Did the overall interface is ugly or attractive?
Analytics Tracker
Google Analytics
User app usage Behavior are track by tracker to check either user preform the predefine tasks successful or not
Task Success metric will be created based on task perform
Question 1 | Question 2 | Question 3 | … | Question n
Task 1 | 1 | 0 | 0
Task 2 | 1 | 1 | 1
… | … | … | …
Task n | 0 | 1 | 1
Somewhat satisfied
Neither satisfied nor dissatisfied
Somewhat dissatisfied
Task Success
Task 1 | Task 2 | Task 3
U_001 | 1 | 0 | 0
U_002 | 1 | 1 | 1
… | … | … | …
U_003 | 0 | 1 | 1

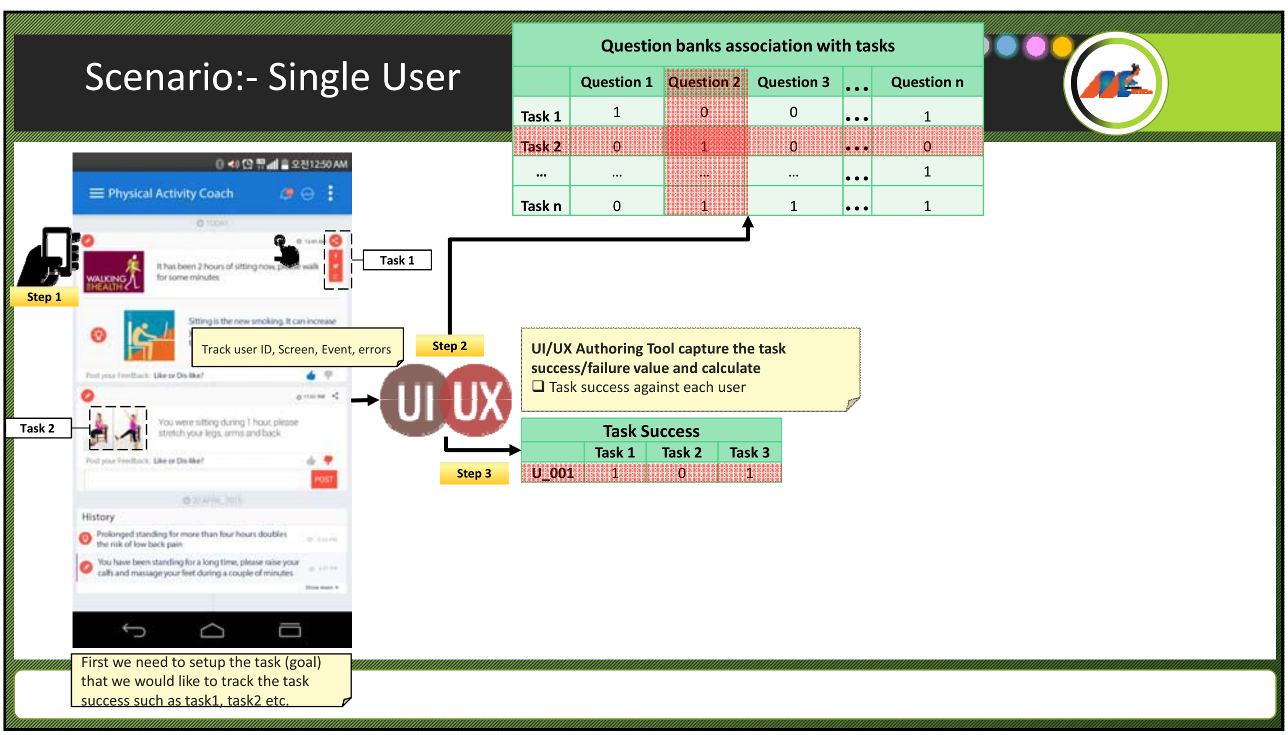

Scenario:- Single User
Question banks association with tasks

	Question 1	Question 2	Question 3	...	Question n
Task 1	1	0	0	...	1
Task 2	0	1	0	...	0
...	...	...	...	...	1
Task n	0	1	1	...	1

Physical Activity Coach
Step 1
Task 1
Task 2
Track user ID, Screen, Event, errors
Step 2
UI UX
Step 3
UI/UX Authoring Tool capture the task success/failure value and calculate
❑ Task success against each user
Task Success

	Task 1	Task 2	Task 3
U_001	1	0	1

History
First we need to setup the task (goal) that we would like to track the task success such as task1, task2 etc.

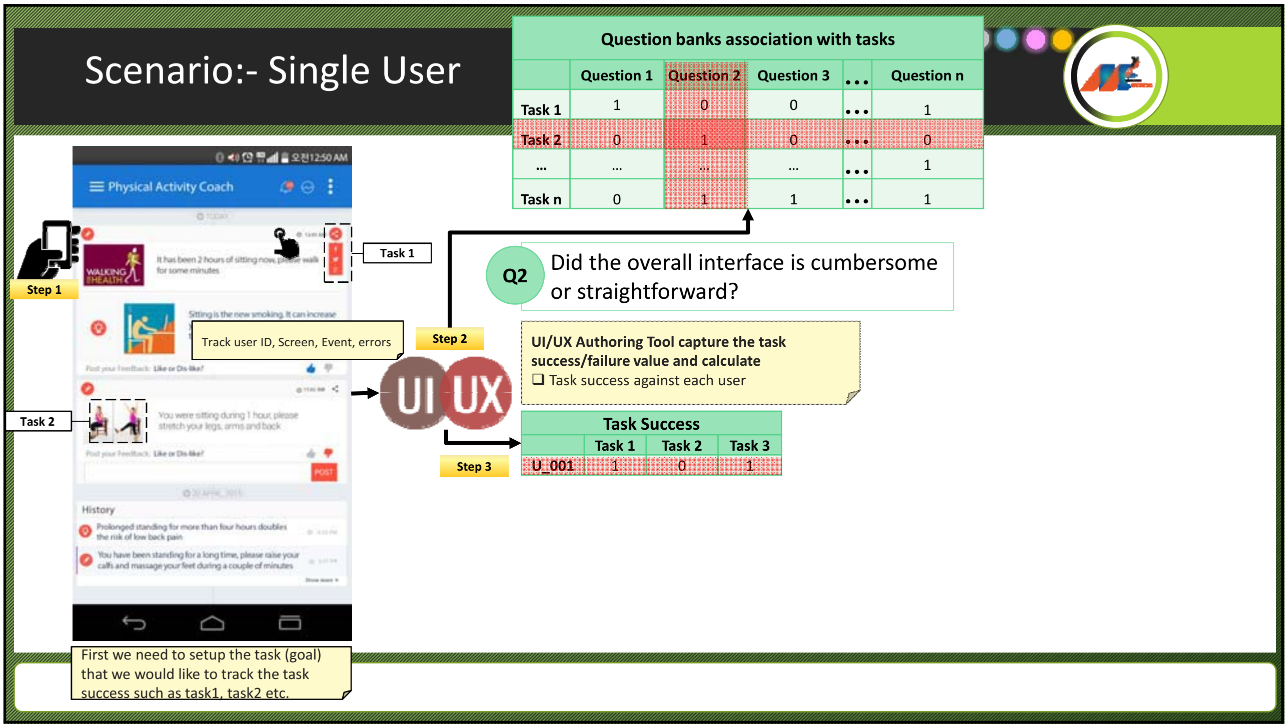

Scenario:- Single User
Question banks association with tasks

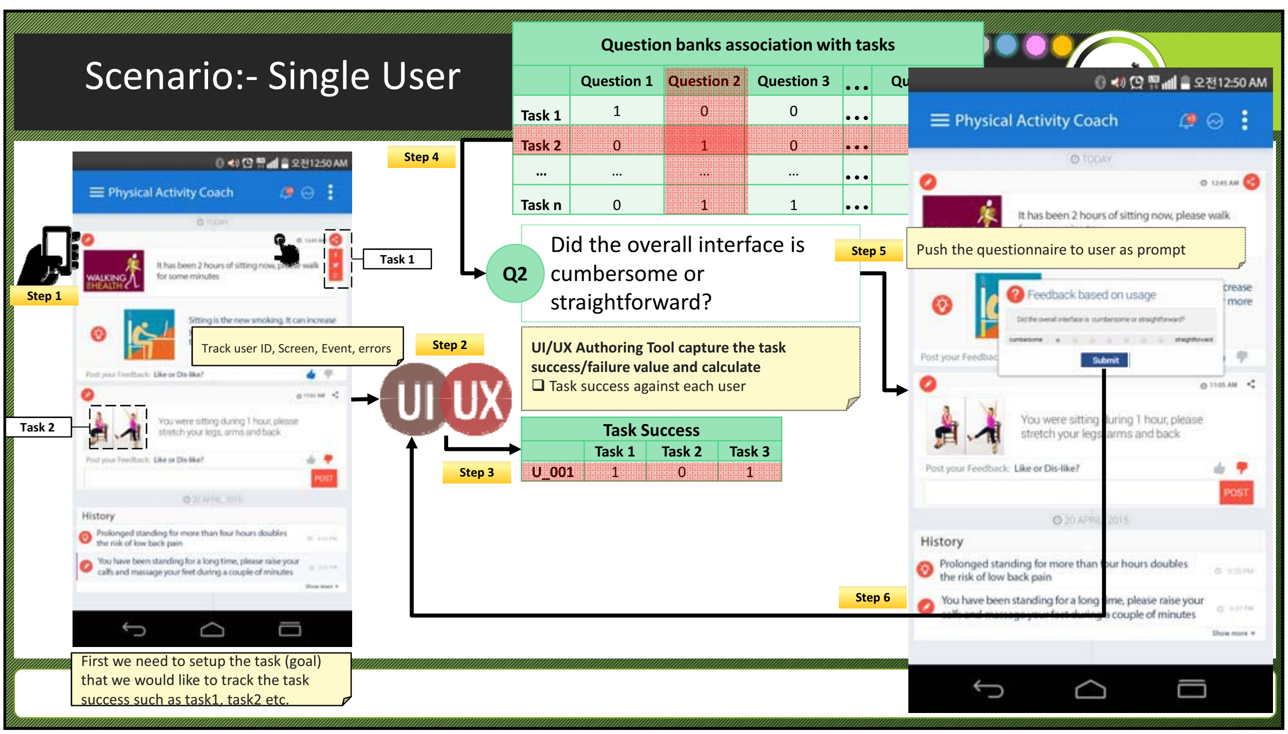
	Question 1	Question 2	Question 3	...	Question n
Task 1	1	0	0	...	1
Task 2	0	1	0	...	0
...	...	...	...	...	1
Task n	0	1	1	...	1

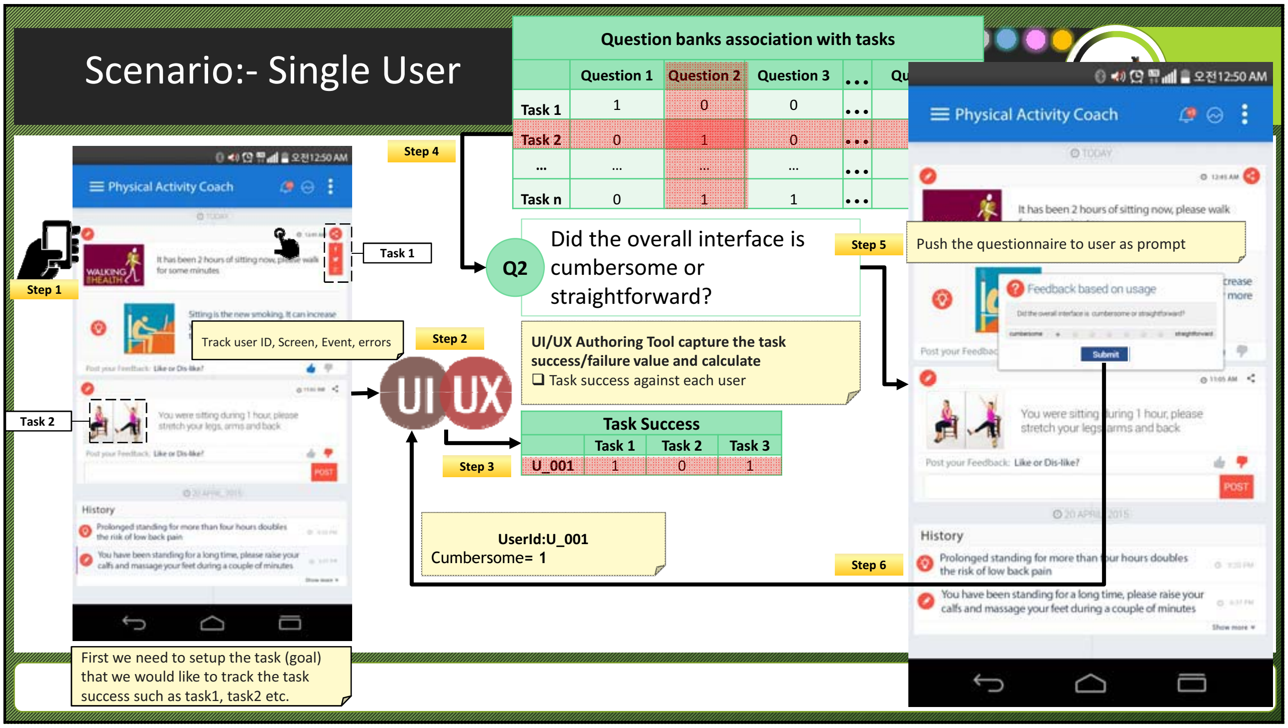
Physical Activity Coach
Step 1
Task 1
Task 2
Track user ID, Screen, Event, errors
Step 2
UI UX
Step 3
UI/UX Authoring Tool capture the task success/failure value and calculate
❑ Task success against each user
Task Success

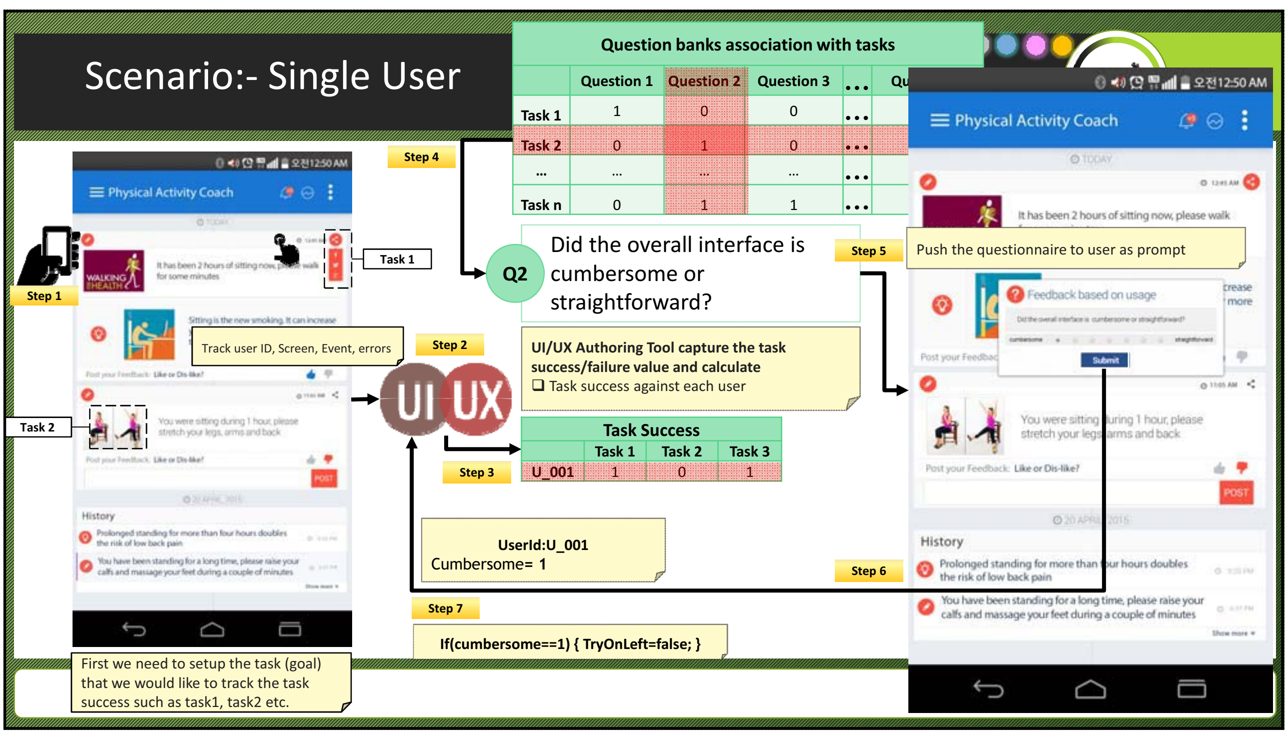
	Task 1	Task 2	Task 3
U_001	1	0	1

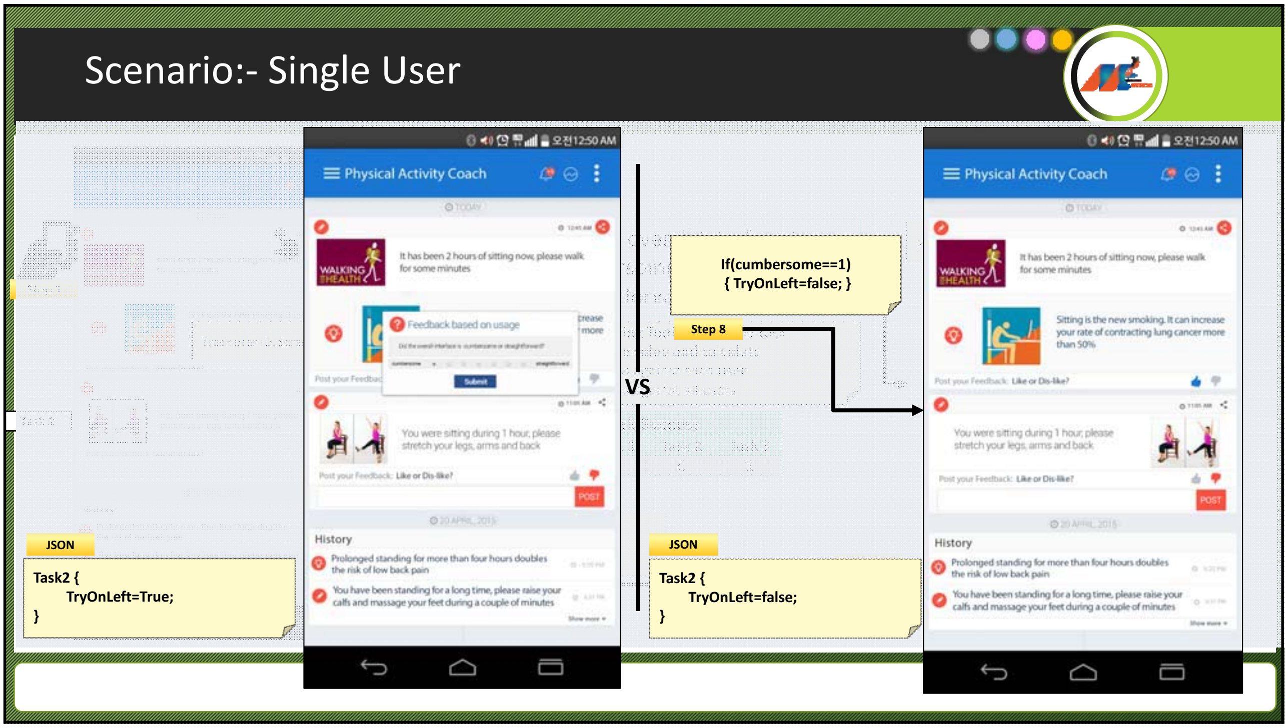
History
First we need to setup the task (goal) that we would like to track the task success such as task1, task2 etc.


Scenario:- Single User
Question banks association with tasks
Question 1
Question 2
Question 3
Question n
Task 1
Task 2
Task n
Did the overall interface is cumbersome or straightforward?
Q2
Step 1
Step 2
Step 3
Task 1
Task 2
Track user ID, Screen, Event, errors
UI/UX Authoring Tool capture the task success/failure value and calculate
Task success against each user
Task Success
U_001
Physical Activity Coach
First we need to setup the task (goal) that we would like to track the task success such as task1, task2 etc.

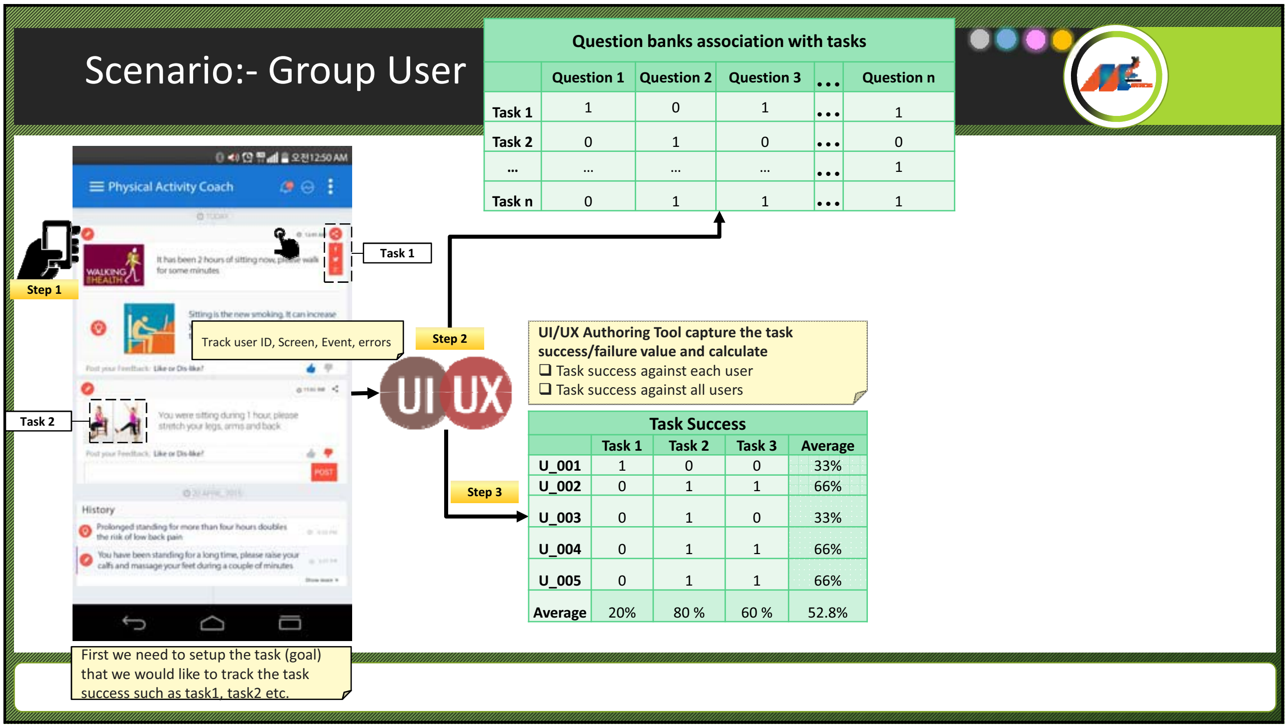

Scenario:- Single User
Question banks association with tasks
Question 1
Question 2
Question 3
Task 1
Task 2
Task n
Did the overall interface is cumbersome or straightforward?
Q2
Step 1
Step 2
Step 3
Step 4
Step 5
Step 6
Track user ID, Screen, Event, errors
Push the questionnaire to user as prompt
UI/UX Authoring Tool capture the task success/failure value and calculate
Task success against each user
Task Success
U_001
Physical Activity Coach
Feedback based on usage
Submit
First we need to setup the task (goal) that we would like to track the task success such as task1, task2 etc.

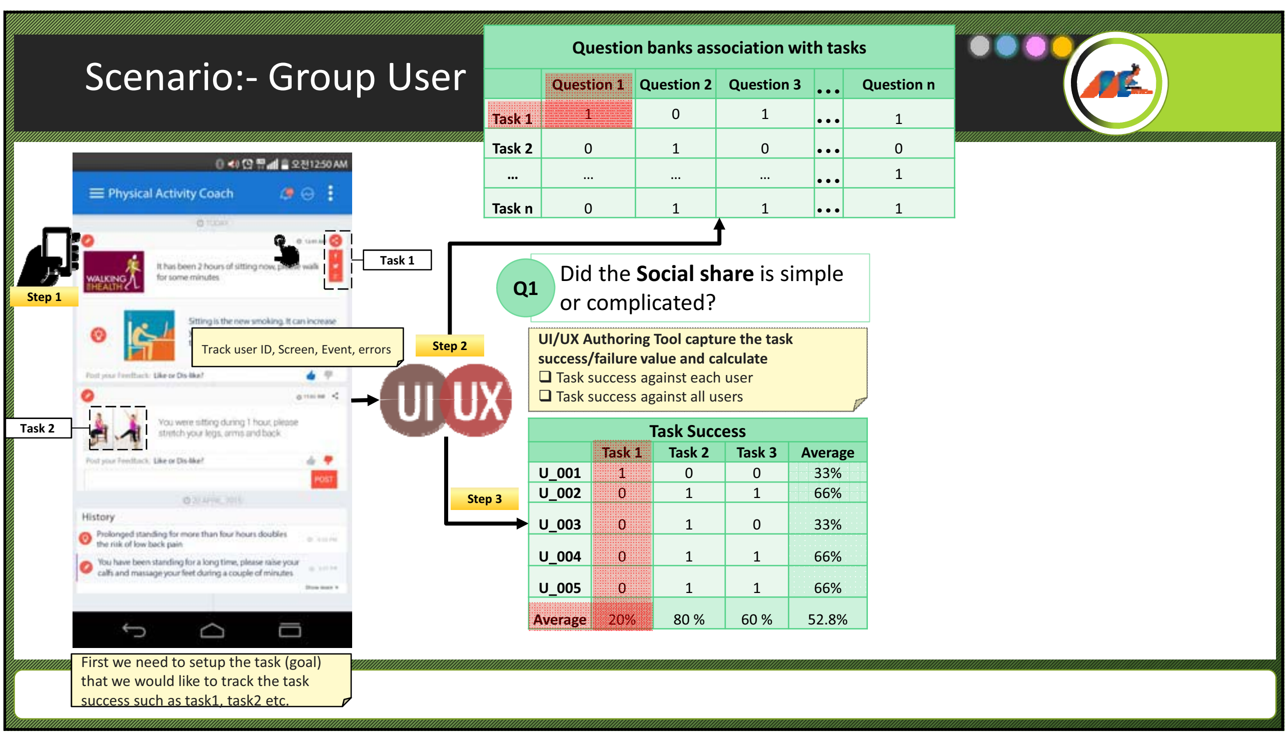

Scenario:- Single User
Question banks association with tasks
Question 1
Question 2
Question 3
Task 1 1 0 0
Task 2 0 1 0
Task n 0 1 1
Step 1
Step 2
Step 3
Step 4
Step 5
Step 6
Task 1
Task 2
Physical Activity Coach
Track user ID, Screen, Event, errors
Q2
Did the overall interface is cumbersome or straightforward?
Push the questionnaire to user as prompt
UI/UX Authoring Tool capture the task success/failure value and calculate
Task success against each user
UI UX
Task Success
Task 1 Task 2 Task 3
U_001 1 0 1
UserId:U_001
Cumbersome= 1
First we need to setup the task (goal) that we would like to track the task success such as task1, task2 etc.
Feedback based on usage
Submit
History
POST

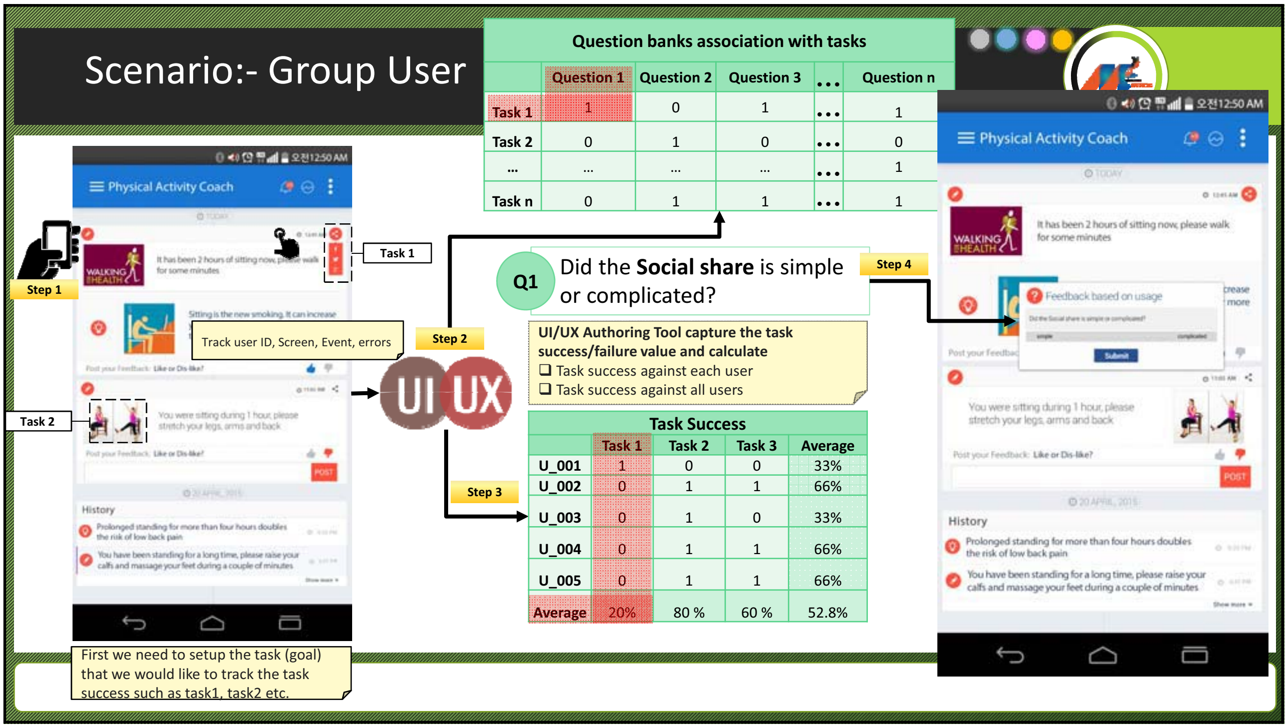

Scenario:- Single User
Question banks association with tasks
Question 1
Question 2
Question 3
Task 1 1 0 0
Task 2 0 1 0
Task n 0 1 1
Step 1
Step 2
Step 3
Step 4
Step 5
Step 6
Step 7
Task 1
Task 2
Physical Activity Coach
Track user ID, Screen, Event, errors
Q2
Did the overall interface is cumbersome or straightforward?
Push the questionnaire to user as prompt
UI/UX Authoring Tool capture the task success/failure value and calculate
Task success against each user
UI UX
Task Success
Task 1 Task 2 Task 3
U_001 1 0 1
UserId:U_001
Cumbersome= 1
If(cumbersome==1) { TryOnLeft=false; }
First we need to setup the task (goal) that we would like to track the task success such as task1, task2 etc.
Feedback based on usage
Submit
History
POST

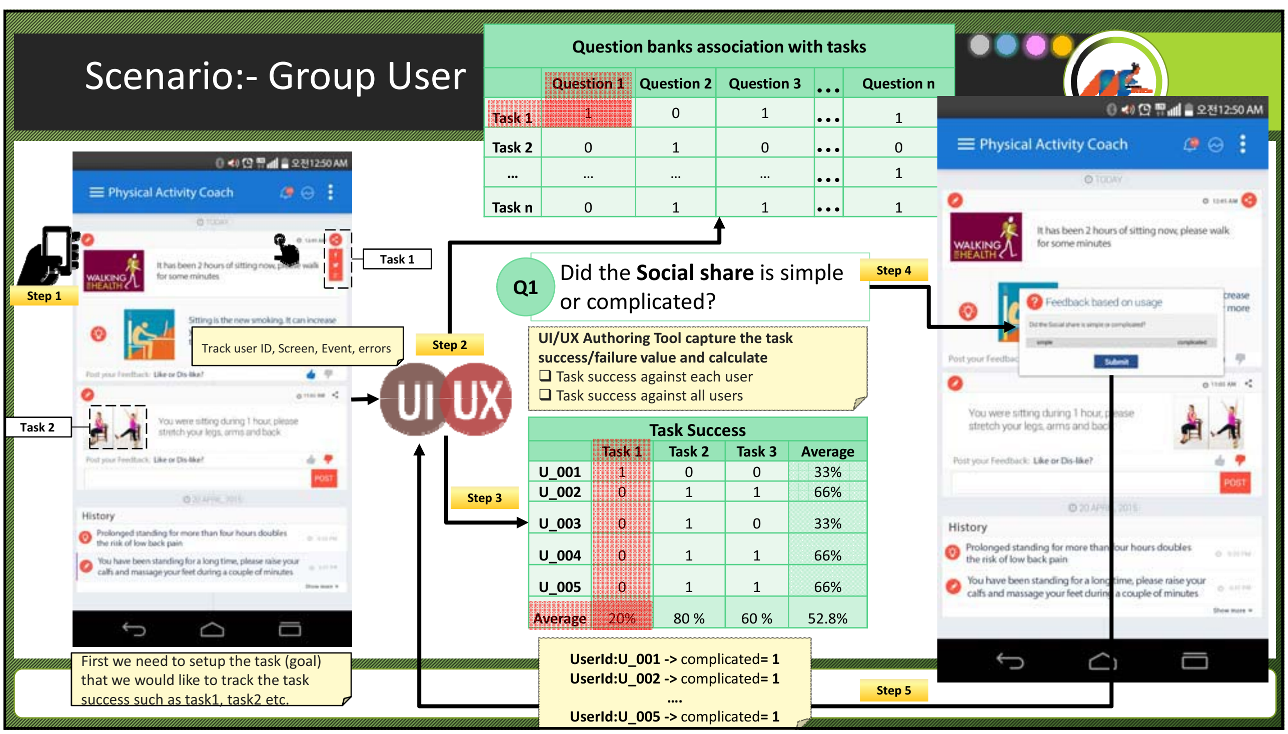

Scenario:- Single User
Physical Activity Coach
It has been 2 hours of sitting now, please walk for some minutes
Feedback based on usage
Submit
You were sitting during 1 hour, please stretch your legs, arms and back
Post your Feedback: Like or Dis-like?
POST
History
Prolonged standing for more than four hours doubles the risk of low back pain
You have been standing for a long time, please raise your calfs and massage your feet during a couple of minutes
JSON
Task2 {
TryOnLeft=True;
}
VS
If(cumbersome==1)
{ TryOnLeft=false; }
Step 8
JSON
Task2 {
TryOnLeft=false;
}
Sitting is the new smoking. It can increase your rate of contracting lung cancer more than 50%


Our approach to behavior change design
Behavior Change
Adaptation
Rewards
Self-Reporting
Group-reporting
Analytics (Usage)
Series 1
Series 2
Series 3
10
0
10
0
Series 1
Series 2
Series 3


Scenario:- Group User
Question banks association with tasks
	Question 1	Question 2	Question 3	...	Question n
Task 1	1	0	1	...	1
Task 2	0	1	0	...	0
...	...	...	...	...	1
Task n	0	1	1	...	1
Physical Activity Coach
Step 1
Task 1
Task 2
Track user ID, Screen, Event, errors
Step 2
UI UX
Step 3
UI/UX Authoring Tool capture the task success/failure value and calculate
❑ Task success against each user
❑ Task success against all users
Task Success
	Task 1	Task 2	Task 3	Average
U_001	1	0	0	33%
U_002	0	1	1	66%
U_003	0	1	0	33%
U_004	0	1	1	66%
U_005	0	1	1	66%
Average	20%	80 %	60 %	52.8%
First we need to setup the task (goal) that we would like to track the task success such as task1, task2 etc.


Scenario:- Group User
Question banks association with tasks
	Question 1	Question 2	Question 3	...	Question n
Task 1	1	0	1	...	1
Task 2	0	1	0	...	0
...	...	...	...	...	1
Task n	0	1	1	...	1
Physical Activity Coach
Step 1
Task 1
Task 2
Q1
Did the Social share is simple or complicated?
Track user ID, Screen, Event, errors
Step 2
UI UX
Step 3
UI/UX Authoring Tool capture the task success/failure value and calculate
❑ Task success against each user
❑ Task success against all users
Task Success
	Task 1	Task 2	Task 3	Average
U_001	1	0	0	33%
U_002	0	1	1	66%
U_003	0	1	0	33%
U_004	0	1	1	66%
U_005	0	1	1	66%
Average	20%	80 %	60 %	52.8%
First we need to setup the task (goal) that we would like to track the task success such as task1, task2 etc.


Scenario:- Group User
Question banks association with tasks
Question 1 | Question 2 | Question 3 | ... | Question n
Task 1 | 1 | 0 | 1 | ... | 1
Task 2 | 0 | 1 | 0 | ... | 0
... | ... | ... | ... | ... | 1
Task n | 0 | 1 | 1 | ... | 1
Physical Activity Coach
Step 1
Task 1
Task 2
Track user ID, Screen, Event, errors
Step 2
UI UX
Step 3
Q1
Did the Social share is simple or complicated?
Step 4
UI/UX Authoring Tool capture the task success/failure value and calculate
❑ Task success against each user
❑ Task success against all users
Task Success
| Task 1 | Task 2 | Task 3 | Average
U_001 | 1 | 0 | 0 | 33%
U_002 | 0 | 1 | 1 | 66%
U_003 | 0 | 1 | 0 | 33%
U_004 | 0 | 1 | 1 | 66%
U_005 | 0 | 1 | 1 | 66%
Average | 20% | 80 % | 60 % | 52.8%
History
First we need to setup the task (goal) that we would like to track the task success such as task1, task2 etc.


Scenario:- Group User
Question banks association with tasks
Question 1 | Question 2 | Question 3 | ... | Question n
Task 1 | 1 | 0 | 1 | ... | 1
Task 2 | 0 | 1 | 0 | ... | 0
... | ... | ... | ... | ... | 1
Task n | 0 | 1 | 1 | ... | 1
Physical Activity Coach
Step 1
Task 1
Task 2
Track user ID, Screen, Event, errors
Step 2
UI UX
Step 3
Q1
Did the Social share is simple or complicated?
Step 4
UI/UX Authoring Tool capture the task success/failure value and calculate
❑ Task success against each user
❑ Task success against all users
Task Success
| Task 1 | Task 2 | Task 3 | Average
U_001 | 1 | 0 | 0 | 33%
U_002 | 0 | 1 | 1 | 66%
U_003 | 0 | 1 | 0 | 33%
U_004 | 0 | 1 | 1 | 66%
U_005 | 0 | 1 | 1 | 66%
Average | 20% | 80 % | 60 % | 52.8%
History
First we need to setup the task (goal) that we would like to track the task success such as task1, task2 etc.
UserId:U_001 -> complicated= 1
UserId:U_002 -> complicated= 1
....
UserId:U_005 -> complicated= 1
Step 5

Scenario:- Group User

App Usage Analytics Data

Task Success				
	Task 1	Task 2	Task 3	Average
U_001	1	0	0	33%
U_002	0	1	1	100 %
U_003	0	1	0	33 %
U_004	0	1	1	
U_005	0	1	1	
Average	66 %	66 %	33 %	55 %

+

User Feedback Data – Self-reporting

User feedback Data			
	Task ID	Question 1	
		Simple	Complicated
U_001	Task 1		
U_002	Task 1	0	1
U_003	Task 1	0	1
U_004	Task 1	0	1
U_005	Task 1	0	1

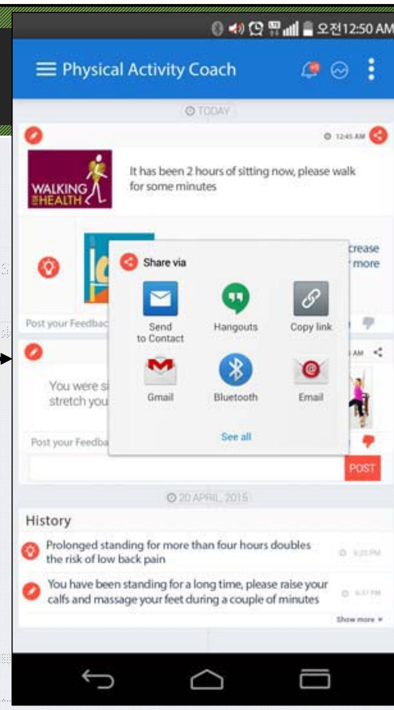
UI/UX Authoring Tool

- use majority vote to get action

Result = Complicate

Question 1: Action

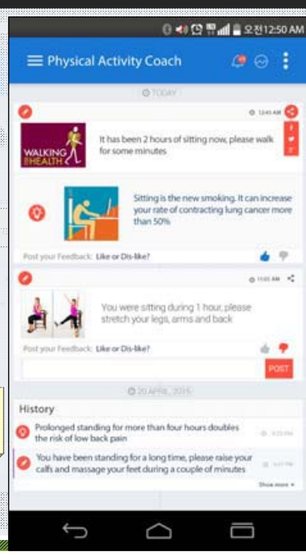
If(Comlicated==1) {
SocialTemplate=2;
}



Scenario:- Single User

JSON

Task1 {
SocialTemplate=1
}



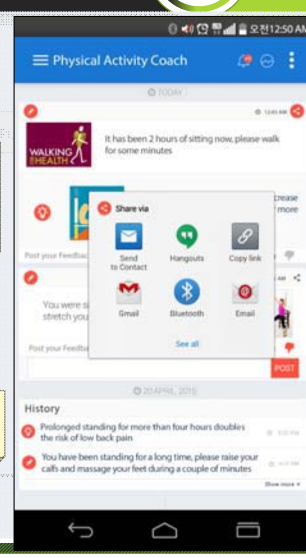
VS

Question 1: Action

If(Comlicated==1) {
SocialTemplate=2;
}

JSON

Task1 {
SocialTemplate=2
}



Thank You