Secured WSN-integrated Cloud Computing for Ubiquitous Life Care (SC^3)

Date: December 2012

Technical Report
Version: 2.0

Approved by ______________________________

Project Supervisor

____

Date ______________________________

Ubiquitous Computing Lab
Department of Computer Engineering
Kyung Hee University, Global Campus, Korea

ABSTRACT

Project Supervision: Professor Sungyoung Lee
Ubiquitous Computing Lab, Department of Computer Engineering
Kyung Hee University (Suwon Campus), South Korea

To provide robust Lifecare services and personalized recommendations; user daily life activity recognition, profile information, and user personal experience are the most important information sources and they facilitate appropriate service execution. SC^3 focuses on the improvement in general life status of users through the use of an innovative service to align all input sources like: activity information, social media information, profile information, dietary intake information, environment information, and clinical information. Ubiquitous Lifecare (u-Lifecare) nowadays becomes more attractive to computer science researchers due to a demand on a high quality and low cost of care services at anytime and anywhere. Many works exploit sensor networks to monitor user's health status and movements to provide care services to them. It requires the above discussed data to be quickly transmitted and processed so that physicians, clinics, and other caregivers can access conveniently via Internet and when required the system can generate recommendations. Most of the existing lifecare systems rely on their own data center to store and process the data. It brings a high cost to maintain the system, yet the performance is not reliable and a limited number of services can be provided.

This Technical Report presents our study and development of a Secured Wireless Sensor Network (WSN) - integrated Cloud Computing for u-LifeCare (SC^3). We abstract SC^3 in three levels: 1) Lifecare Decision Support System for Wellness (LDSS for Wellness), 2) Lifecare Decision Support System for Chronic Disease (LDSS for Chronic Disease), and 3) SmartCDSS for Head & Neck Cancer. Sensor data, environment data, profile data, diet data, social media data, and health data is collected and sent to the Clouds. In the Cloud, a three of the services are deployed that interpret the incoming data with the help of expert knowledge and our proposed algorithm and produce corresponding services and recommendations. Different users, committees, such as hospitals, clinics, researchers, relatives, or even users themselves can access their data in the Clouds. We have also introduced a new methodology of separating Lifecare from healthcare and disease treatment. We have also presented details of each component and their working scenarios. Our proposed SC^3 can help in enhancing capabilities and provides tremendous value by achieving efficient use of software and hardware investments. Our Cloud infrastructure drives profitability by improving resources utilization and increasing their scalability while maintaining strong privacy and security essential in u-Lifecare. For the proof of concept, we have provided four detailed scenarios where working of different SC^3 services is demonstrated[1].

[1] This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency).

Table of Contents

INTRODUCTION

1.1. u-Lifecare

- u-Lifecare is the ubiquitous lifecare and lifecare services which are delivered to the user anytime, anywhere.
- It encapsulates medical and health management aspects in addition to lifecare and wellness services.
- The main concern points are as follow and also shown in Figure 1.
  - Ubiquitous Life
  - Medical Services
  - Healthcare and Management Services

Figure 1 Abstract concept of u-Lifecare

- u-Lifecare is a package (shown in Figure 2) that provides set of advantages like:
  - Reduced medical cost
  - Improved personalized healthcare/lifecare services
  - Improved healthcare/lifecare services quality
  - Improved healthcare/lifecare services accessibility
  - Reduce hospital visits
  - Reduce caregivers burden and also reduce utilization of hospital resources

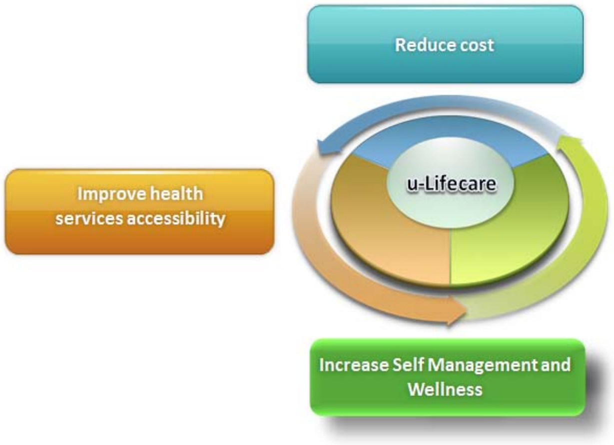


Figure 2 u-Lifecare and its effects and advantages

- To achieve the objectives, different dimensions of the system needs to be understood and worked out. These dimensions are diverse in their nature and critical in their effects. The dimensions are 1) Social Dimension, 2) Physical Dimension, and 3) Personal Dimension. These are shown in Figure 3.

Figure 3 Dimensions of u-Lifecare

- These factors given in Figure 3 incorporates factor [Little2008] which they are based on. These factors are the target to achieve and make the vision of u-Lifecare a reality. Solution to these factors (shown in Figure 4) is also the motivating point towards the conceptualization and development of u-Lifecare system.

Figure 4 Factors effecting and motivating the conceptualization and development of u-Lifecare System

1.2. Cloud Computing

"A Cloud is a type of parallel and distributed system consisting of a *collection of interconnected and virtualized computers* that are *dynamically provisioned* and presented as one or more *unified computing resources* based on service-level agreements established through negotiation between the service provider and customers and can be ubiquitously accessed from any connected devices over the internet"

- The Cloud computing, coined in late of 2007, currently emerges as a hot topic due to its abilities to offer flexible dynamic IT infrastructures, QoS guaranteed computing environments and configurable software services.
- Cloud computing started quietly from several seeding technologies such as grid computing, virtualization, SalesForce.com innovative subscription-based business model or Amazon's effort to scale their e-commerce platform.
- It differs from traditional ones in that:
    - It is massively scalable
    - Encapsulated as an abstract entity that delivers different levels of services to customers anywhere, anytime
    - It is driven by economies of scale that is the services can be dynamically configured (via virtualization or other approaches) and delivered on-demand.

- The Web search popularity, as measured by the Google search trends in 2008, for terms "Cluster computing", "Grid computing", and "Cloud computing" is shown in Figure 5.
    - From the Google trends, it can be observed that cluster computing was a popular term during 1990s, from early 2000 Grid computing become popular, and recently Cloud computing started gaining popularity.
    - Market-research firm IDC expects IT Cloud-services spending to grow from about $16 billion in 2008 to about $42 billion by 2012 as Figure 6 shows.

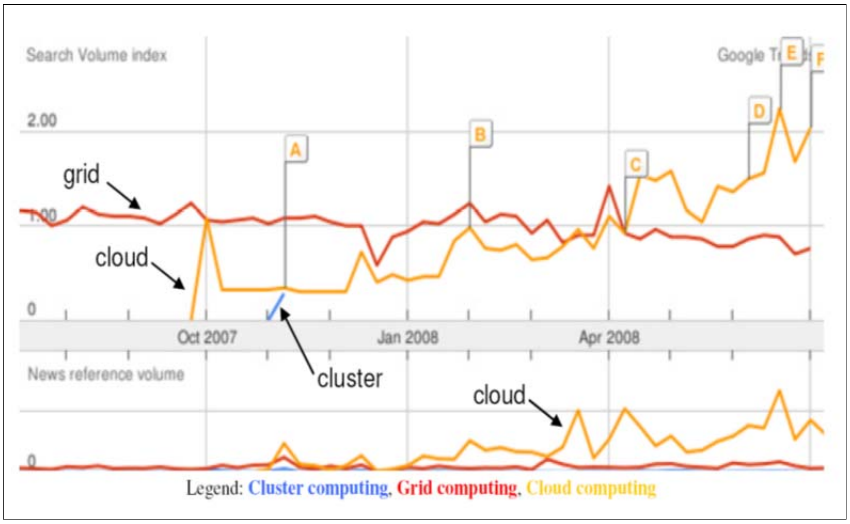


Figure 5 Cloud computing in Google Trends

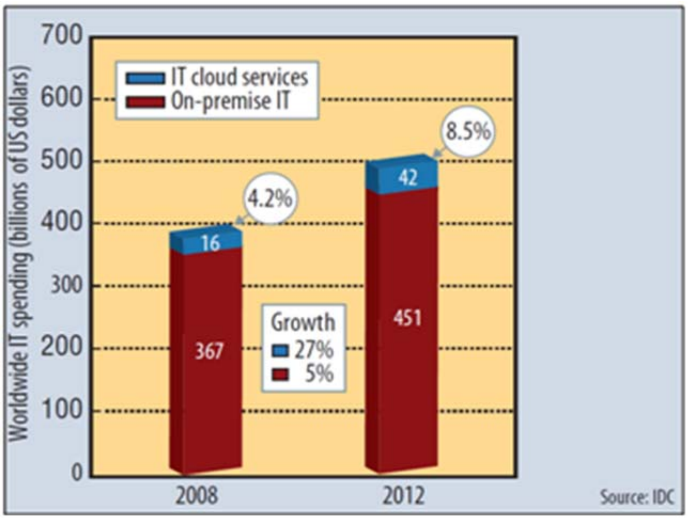


Figure 6 IT service investment

- Cloud Computing has many benefits that the public sector and government IT organizations are certain to want to take advantage of. In very brief summary form they are as follows:
  - **Reduced cost, higher gains**: Cloud technology is paid incrementally, saving organizations money.
  - **Increased storage**: Organizations can store more data than on private computer systems.
  - **Highly automated**: No longer do IT personnel need to worry about keeping software up to date.

- **Flexibility**: Cloud computing offers much more flexibility than past computing methods.
- **More mobility**: Employees can access information wherever they are, rather than having to remain at their desks.
- **Allows IT to shift focus**: No longer having to worry about constant server updates and other computing issues, government organizations will be free to concentrate on innovation.

▪ In Cloud computing, customers do not own the infrastructure they are using; they basically rent it, or pay as they use it.
▪ One of the major selling points of cloud computing is lower costs. Companies will have lower technology-based capital expenditures, which should enable companies to focus their money on delivering the goods and services that they specialize in. The general architecture of Cloud computing is shown in Figure 7.

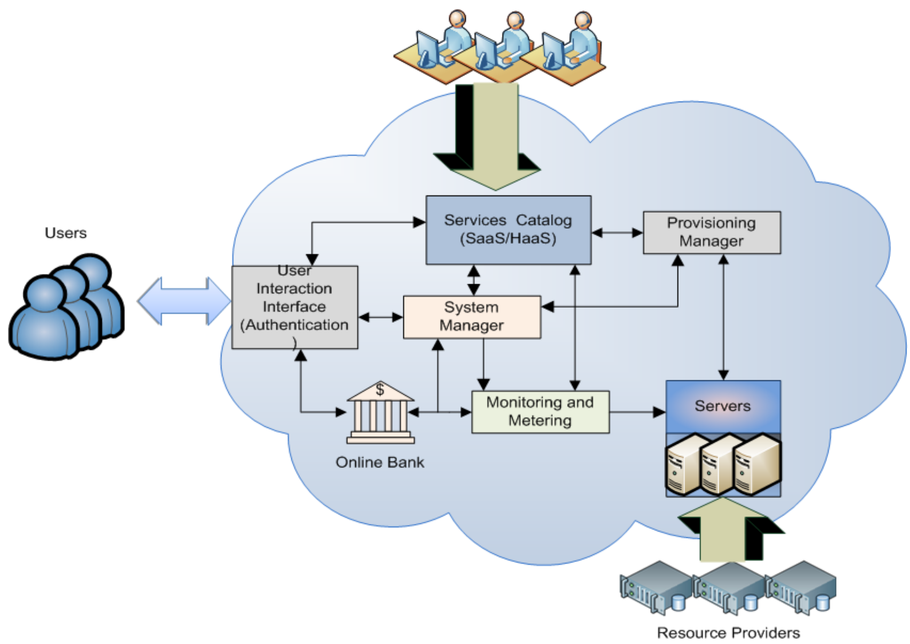


Figure 7 General Architecture of Cloud Computing

MOTIVATION

2.1. Cloud Computing in u-Lifecare

- As the standard of living rises, people are more interested in their health and desire well-being life.
- Today due to aging of population, rising cost of workforce and high quality treatment, threat of new panepidemies and diseases, the cost of life care or healthcare system is increasing worldwide.

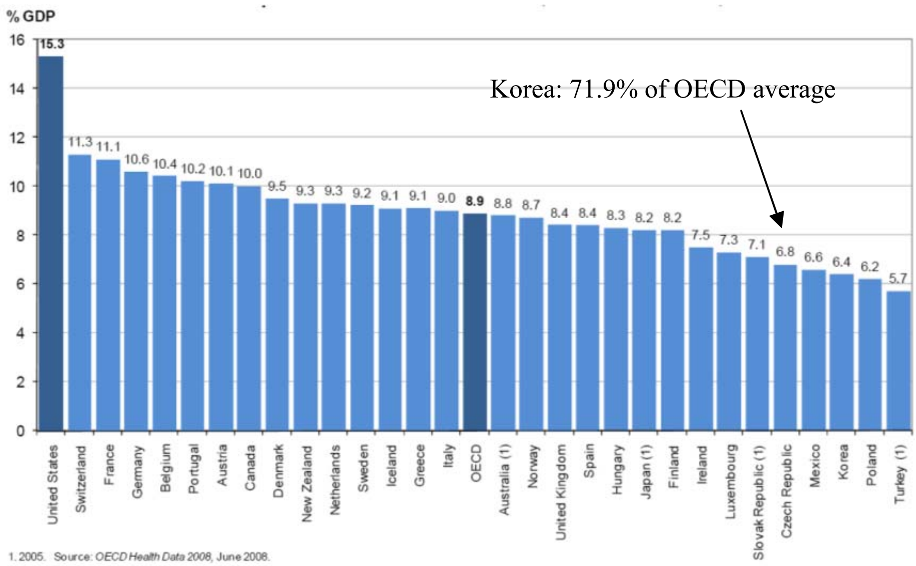


Figure 8 Health expenditure as a share of GDP, OECD countries, 2006

- Accoridng to OECD (Organizaiton of Economic Cooperation and Development) Health data 2008 (shown in Figure 8), total health spending accounted for 15.3% of GDP in the United States in 2006, the highest share in the OECD, and more than six percentage points higher than the average of 8.9% in OECD countries. Korea was 6.4% of GDP to health in 2006. The United States also ranks far ahead of other OECD countries in terms of total health spending per capita, with spending of 6,714 USD (adjusted for purchasing power parity (PPP)), more than twice the OECD average of 2,824 USD in 2006. For Korea it was 1480 USD.
- In Korea, the number of aging population is increasing as shown in Figure 9.
- The national health expenditure of Korea as compare to GDP is shown in Figure 10. We can see from that the financial crisis in healthcare is increasing and Governments are trying to increase the budget every year.

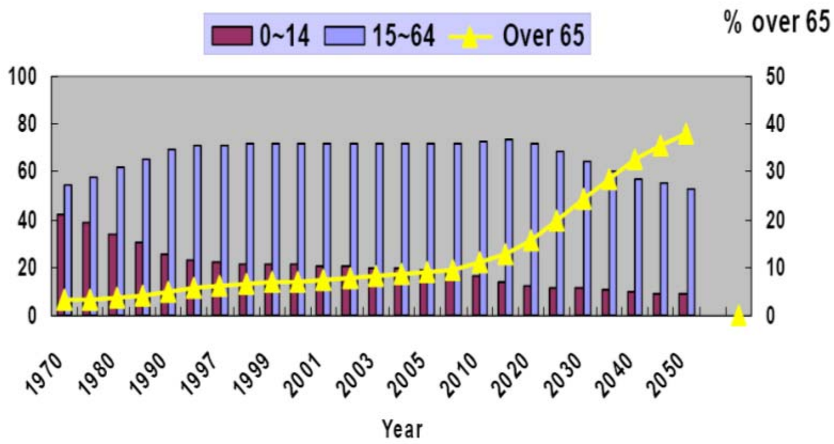


Figure 9 Population by Age group in Korea

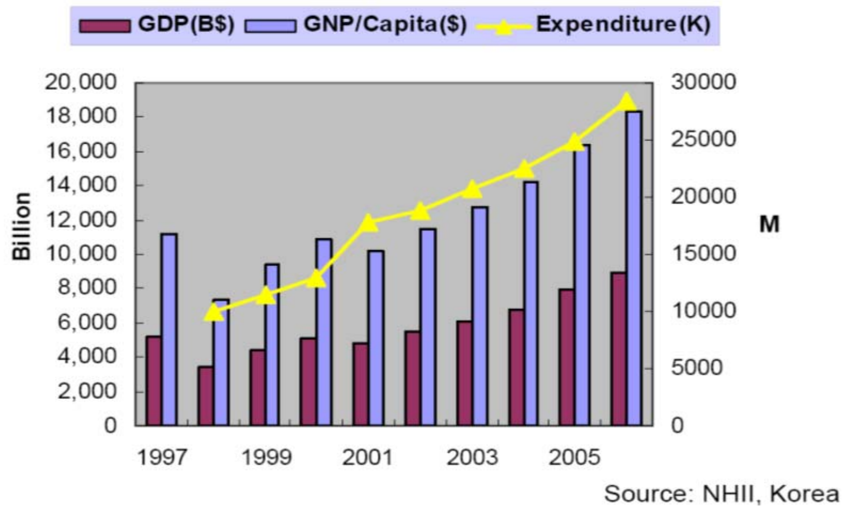


Figure 10 National Healthcare expenditure compare to GDP

- To maintain the quality and availability level of lifecare services with minimum cost, Cloud computing can provide a powerful, flexible, and cost-effective infrastructure for lifecare services. The cloud infrastructure for vendor and institute is shown in Figure 11.

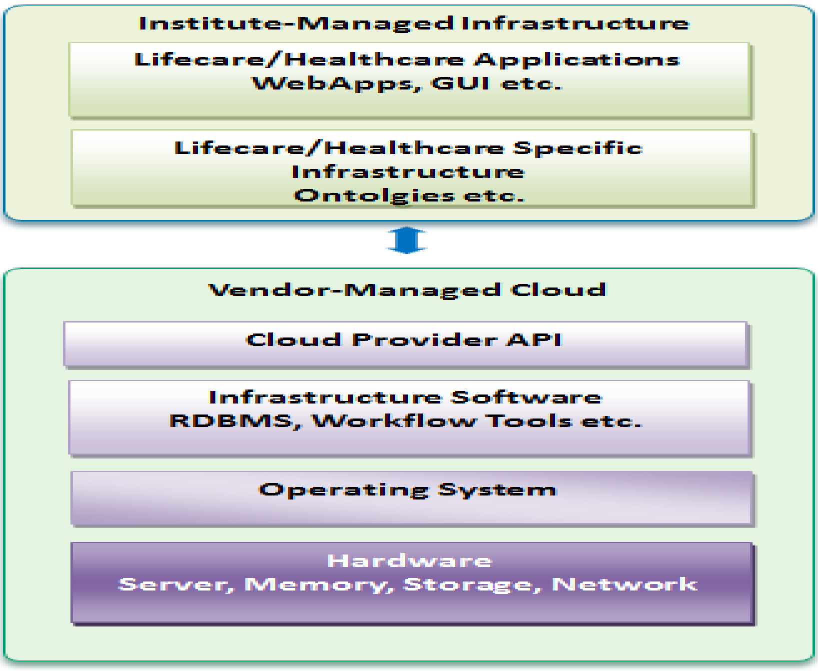


Figure 11 Cloud Computing infrastructure for u-Lifecare

- This can fulfill the vision of "Ubiquitous Lifecare" that is providing life care to people anywhere at any time while increasing both the coverage and the quality.
- Because of its elasticity, scalability, pay-as-you-go model, Cloud computing can potentially provide huge cost savings, flexible high-throughput, and ease of use for life care services.
- The motivations for Cloud Computing in u-Lifecare are:
  - With lifecare providers looking at automating processes at lower cost and higher gains, Cloud computing can act as an ideal platform in the Life care IT space.
  - Hospitals could share Cloud computing infrastructure with vast number of systems linked together for reducing cost and increasing efficiency.
  - Patient information and data can be accessed globally and resources can be shared by a group of hospitals rather than each hospital having a separate IT infrastructure.
  - Cloud computing would help hospitals to achieve more efficient use of their hardware and software investments and increase profitability by improving the utilization of resources to the maximum.
  - By pooling the various Lifecare IT resources into large clouds, hospitals can reduce the cost and increase utilization as the resources are delivered only, when they are required.
  - The use of cloud computing architecture helps is in eliminating the time and effort needed to roll a Lifecare IT application in a hospital.
  - Share information stored across disparate Healthcare information Systems.
  - Integrating with wireless sensor networks, the cloud can provide real time information of the environment for collaboration and knowledge sharing in data intensive research and analysis, especially in the health and biomedical arena.
  - The Cloud could, for instance, provide a flexible platform for public-health departments to upload real time health data in a timely manner to assist state and national health offi-

cials in the early identification and tracking of disease outbreaks, environmental-related health problems, and other issues.
- Provides agility to the deployed Healthcare services with scaling upwards or downwards as needed.
- Provides healthcare organizations with the opportunity to improve quality of care as well as access to care with its pay-as-you-go cost model.
- Cloud Vendors sell computation and storage resources as commodities
- Providing users with the illusion of indefinite resource power
- Clouds use large data centers with numerous processors
- Clouds are managed by organizations with specialists in several areas of data processing security and storage
- Within the datacenter, the network bandwidth is usually high, allowing the underlying computers to share data with one another efficiently

2.2. Problems of Existing Cloud Computing to Support u-Lifecare

▪ **Poor Security and Privacy Support**
  - Data for lifecare services normally composes of personal information, contextual information (e.g. location, user activity information), medical data (e.g. medical history, drug information, and medical health record), etc.
  - Such information is highly sensitive and people do not want to disclose them to the public.
    - For example, a patient with HIV positive test may not want to expose his result to the other, even to their family.
  - Storing data in Cloud leads to more security and privacy problems than traditional computing systems such as distributed systems or grid computing systems.
  - Sensitive data processed outside the enterprise brings with it an inherent level of risk, because outsourced services bypass the "physical, logical and personnel controls"
  - IT shops exert over in-house programs. More dangers and vulnerabilities may cause disrupts of services, theft of information, loss of privacy, damage of information.
  - On the other hand, because any one can access to Clouds, it brings more chances for malicious users to launch their hostile programs.

▪ **No Existing Infrastructure for Integration of WSN to Cloud**
  - In the past few years, wireless sensor networks (WSNs) have been gaining increasing attention to create decision making capabilities and alert mechanisms, in many Life care application areas including Lifecare monitoring for patients, environmental monitoring, pollution control, disaster recovery, military surveillance etc.
    - For example, MIT wireless sensor ring as shown in Figure 12 can measure heart rate, heart rate variability, Oxygen saturation and blood pressure for the person wearing the ring.

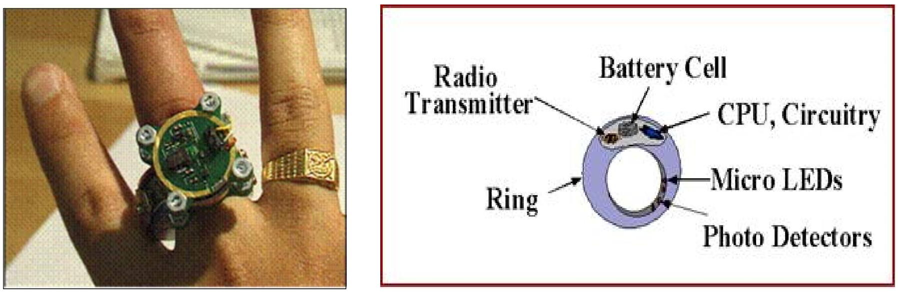


Figure 12 MIT wireless sensor ring and its internal architecture

- Collection, analysis (knowledge processing, ontology reasoning etc.), storing and disseminating of these sensor data is a great challenge since sensor nodes constituting a WSN have limited sensing capability, processing power, and communication bandwidth.
  - There is a lack of uniform operations and standard representation for sensor data.
- Cloud computing provides a powerful, flexible, cost-effective and massively scalable platform that enables real-time sensor data collection and the sharing of huge computational and storage resources for sensor data processing and management.
- Using this collection of real-time sensor derived data to various Cloud computing applications, we can have a significant transformation in our ability to "see" ourselves and our planet.
  - For example, in case of health care, integrating WSN to Cloud computing can provide real-time information of the environment for collaboration, analysis and knowledge-sharing of the early identification and tracking of disease outbreaks, environmental-related health problems, and other issues

▪ **Lacking of Efficient Information Dissemination Mechanism**
- To deliver published sensor data or events to appropriate users or applications from Sensor-Cloud, there is a need of an information dissemination mechanism that matches published events with subscriptions efficiently.
- Designing an efficient content or event matching algorithm is a key especially for the range predicate or overlapping predicate case.
- It is difficult to construct an effective index for multidimensional range predicates. It is even more challenging if these predicates are highly overlapping.
- In u-Lifecare application scenario, doctors and caregivers may express their interests into a range (*i.e.* 35 < *body temperature* < 37)). As patient states are changed continuously within a certain range (min, max) like body temperature and heartbeat, each classified consumer will require events in various ranges to safely take care of patients.
- Also there are other issues to be considered in the design of information dissemination system like maintaining the flexible expressiveness of predicates that is the ability to provide powerful subscription schema capable to capture information about events, guarantees the scalability with respect to the number of subscribers and the published events and supporting system adaptability.

▪ **No Support of Dynamic Collaboration between Cloud Providers in case of Service Level Agreement Violation (SLA)**
- As consumers of different Cloud applications rely on Cloud Providers (CLP) to supply

all their computing needs (process, store and analyze huge sensor data and user generated data) on demand.
- They will require specific QoS to be maintained by their providers in order to meet their objectives and sustain their operations.
- Existing commercial Cloud services are proprietary in nature. They are owned and operated by individual companies.
  - Each of them has created its own closed network, which is expensive to setup and maintain. So, if the CLP is unable to provide quality of service to the end-user requests (in the case when huge sensor data needs to process for critical U-Life care scenario), it may result in *Service Level Agreement* (SLA) violation and end up costing the provider.
- There is a need of a dynamic collaboration between Cloud providers.
- Choosing the best combination of Cloud providers for dynamic collaboration is the major challenge in terms of cost, conflicts between providers, time and QoS.

▪ **Limitations of Existing Systems to be ported to Cloud Computing**
- Minimal support for human aware technologies
- Mostly focus on medical data storage and management: focused on providing PHR (Personal Health Record) Service and EMR (Electronic Medical Record) Service
- Lack of intelligent processing for Lifecare/Healthcare information
- Lack of sharing personalized healthcare information for different stakeholders
- Lack of wellness services provisioning to consumers of the systems
- No sufficient security and privacy support

RELATED WORK

3.1. u-Lifecare Systems

- There are variety of Lifecare and Healthcare systems that are currently in use and are providing healthcare and lifecare services.
- The start of healthcare/lifecare systems started around 1994 for Hospital Information System as shown in Figure 13.
- The journey has led to the current lifecare or wellness system's needs which way more sophisticated, reliable, and cost effective.

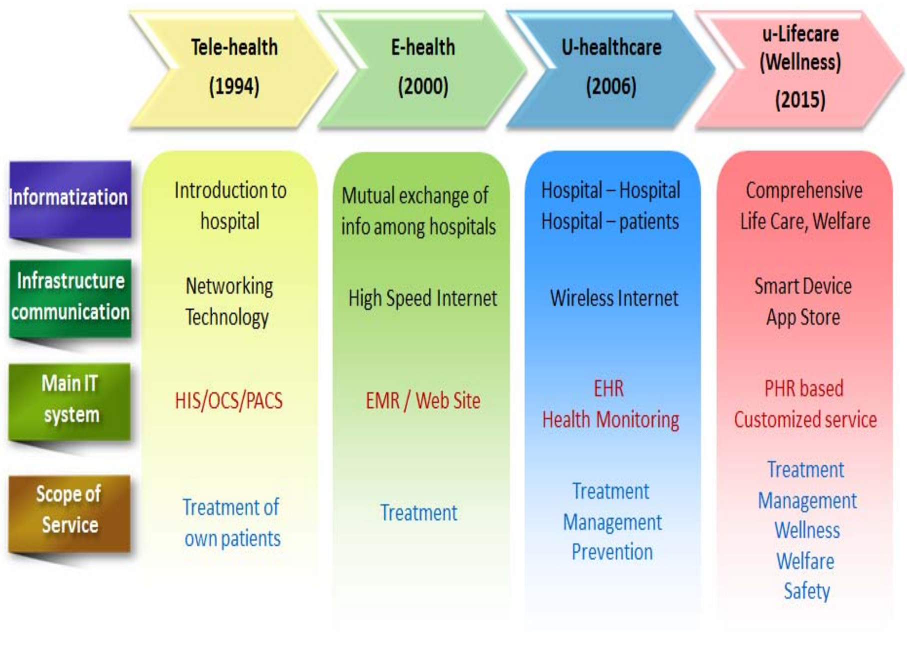


Figure 13 Lifecare / Healthcare systems evolution

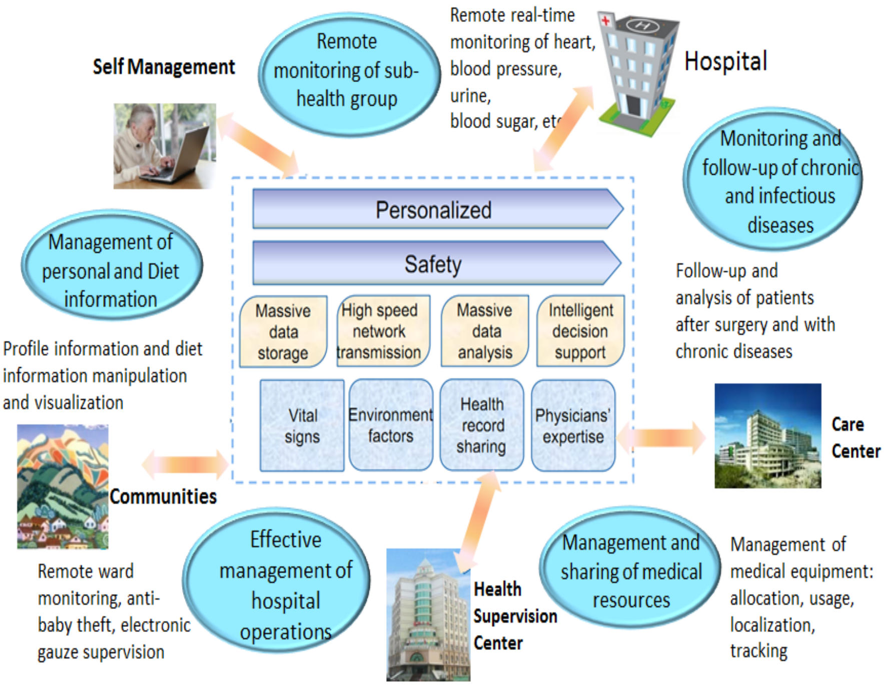


Figure 14 Lifecare / Healthcare system interaction and processing model [Lionel2011]

- To manage and execute u-Lifecare system according to stakeholders requirements and service consumer's needs, there are list of input, processing and output needs that need to be satisfied.
- These needs are independent and inter-related which results in smooth working of the overall system and completion of user requirements.
- The requirements, their source of input, the components of execution, the output, and the interaction entities are all elaborated and presented in Figure 14.
- Some existing Cloud computing efforts for healthcare/lifecare are as follows:
  - IBM Introduces 'Blue Cloud' Computing, CIO Today - Nov 15 2007
  - IBM, EU Launch RESERVOIR Research Initiative for Cloud Computing, IT News Online - Feb 7, 2008
  - Google and Salesforce.com in Cloud computing deal, Siliconrepublic.com - Apr 14 2008
  - Demystifying Cloud Computing, Intelligent Enterprise - Jun 11 2008
  - Yahoo realigns to support Cloud computing, 'core strategies', San Antonio Business Journal - Jun 27 2008
  - Merrill Lynch Estimates "Cloud Computing" To Be $100 Billion Market, SYS-CON Media - Jul 8

Now we will describe some existing work related to u-Life care.

3.2. Korea u-Care System for a Solitary Senior Citizen

- Numerous projects within industry and academia are already started or being used in reality.
- In 17 July 2008, an RFP of Ministry for Health, Welfare and Family Affairs, Korea has released u-Care System for a Solitary Senior Citizen (SSC) [Korea2008] as shown in Figure 15.

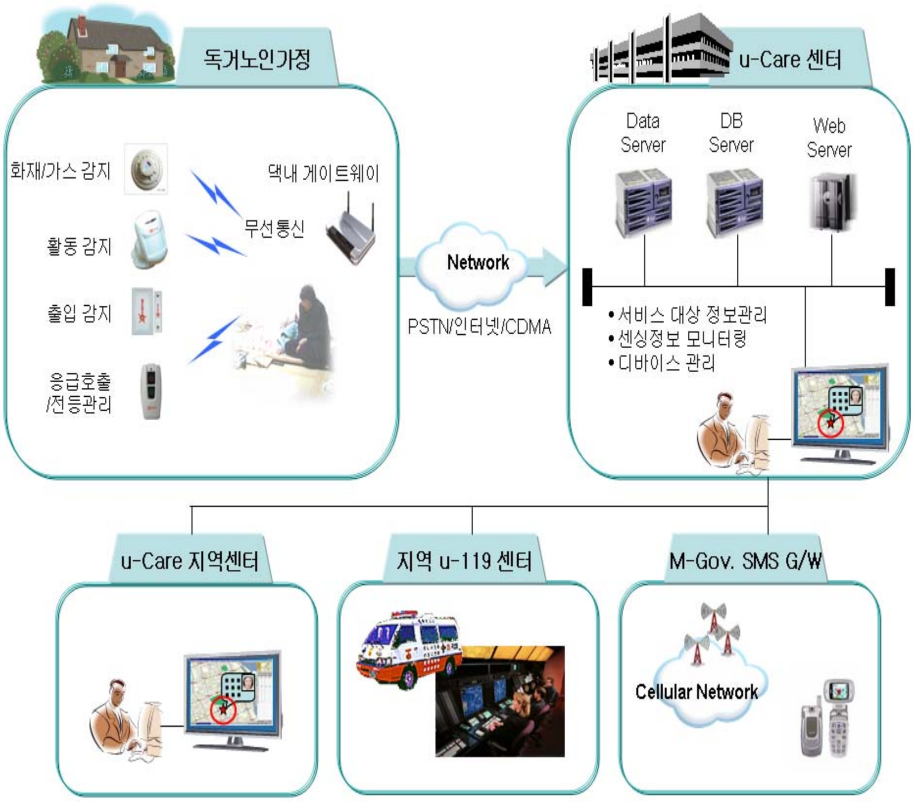


Figure 15 Hardware architecture of u-Care System for a Solitary Senior Citizen (source: Korea Ministry)

- The system provides a number of featured services including:
- **24 hours × 365 days safety monitoring services for a SSC**: monitoring SSC's movement or location by using motion or tripwire sensors, voice communication between a SSC and a life-care giver at all times, emergency call service for heart attack, etc.
- **Emergency-connection services**: real-time monitoring of gas spillage or fire by using gas and detection sensors, if any emergency situation occurs like gas or fire, automatically notify to u-119 system.
- **Information sharing services**: sharing a SSC database to related organization such as fire-station, care givers, or central information system, and a history of care givers activities to build an efficient SSC management systems, etc.
- In this system, *u-Care Center* maintains Data Server, DB Server, and Web Server to store data from SSC's home and provide access to various services, as illustrated in Figure 15.

However, its biggest issue is a high cost of deployment when it becomes widely used.

- With recent advanced in MEMS technology which enables low cost sensors, each user may carry on a number of sensors and wearable devices, leading to a huge amount of data produced.
- On the other hand, more and more life care applications and services access to the system to provide better care for users. It obviously causes a problem of economy to maintain the system.
- In another sector of healthcare applications, several providers explore advantages of Cloud computing to reduce cost and increase care services.
    - Among those, Microsoft and Google are two pioneers who bring Cloud healthcare platforms to reality.

3.3. Microsoft HealthVault

- Microsoft developed a platform to store and maintains health and fitness information, called HealthVault [Microsoft].
- It is a cloud service that helps people collect, store, and share their personal health information.
- HealthVault's data can come from providers, pharmacies, plans, government, employers, labs, equipment and devices, and from consumers themselves.
- Access to a record is through a HealthVault account, which may be authorized to access records for multiple individuals.
    - A mother may manage records for each of her children or a son may have access to his father's record to help the father deal with medical issues.

3.4. Google Health

- Meanwhile, Google provides a personal health information centralization services, known as Google Health [Google].
- The service allows Google users to volunteer their health records, either manually or by logging into their accounts at partnered health services providers, into the Google Health system.
- Thereby merging potentially separate health records into one centralized Google Health profile.
- Volunteered information can include health conditions, medications, allergies, and lab results.
- Once entered, Google Health uses the information to provide the user with a merged health record, information on conditions, and possible interactions between drugs, conditions, and allergies.
- In general, HealthVault and Google Health serve as Cloud health information storages and operate separately.
- As consumers of different Cloud applications rely on Cloud Providers (CLP) to supply all their computing needs (process, store and analyze huge sensor data and user generated data) on demand.
- They will require specific QoS to be maintained by their providers in order to meet their objectives and sustain their operations.
    - To solve the problem of Cloud interoperation, a Unified Cloud Interface (UCI) standardization has been proposed.

- Recently, this service has been stopped by Google.

3.5. Amazon's Cloud Computing Based Healthcare

- At an invitation-only event sponsored by Harvard Medical School and Amazon Web Services, a few dozen experts convened in Boston for a day to ponder the possibilities of cloud computing in their work.
- Participants included health care IT leaders, academics, biomedical researchers, medical and scientific consulting firm representatives, and officials from vendors like Amazon, Oracle, and Hewlett-Packard [Amazon].
- For its part, Amazon in recent weeks unveiled the AWS Hosted Public Data Sets, or "Public Data Computing Initiative,"
- This provides on the cloud a "hosted-for-free, centralized public repository" for data -- such as United States census and human genome research data -- useful to researchers,

Now we will describe some existing solutions or standard regarding Cloud computing.

3.6. Secured Cloud Overlay: VPN-Cubed

- One of the security challenges of Cloud Computing - and specifically Infrastructure as a Serv ice (IaaS) is securely connecting enterprise network to one or more Cloud providers without deploying VPN hardware.
- VPN-Cubed™ is the first commercial solution by Cohesive FT that enables customer control in a cloud, across multiple clouds, and between private infrastructure and the clouds.
- VPN-Cubed provides an overlay network that allows the user to control of addressing, topology, protocols, and encrypted communications for devices deployed to virtual infrastructure or cloud computing centers [SecureCloud].
- When using public clouds corporate assets are going into 3rd party controlled infrastructure. VPN-Cubed gives you flexibility with control in 3rd party environments.
- Here consumer has most of the responsibility to secure his/her own data

3.7. Unified Cloud Interface (UCI) Standardization

- The Unified Cloud Interface (UCI) standardization [UnifiedCloud] or Cloud broker will serve as a common interface for the interaction with remote platforms, systems, networks, data, identity, applications and services.
- UCI will be composed of a semantic specification and an ontology.
- The ontology provides the actual model descriptions, while the specification defines the details for integration with other management models.
- One of the key drivers of the unified cloud interface is to create an API about other API's.
- A singular programmatic point of contact that can encompass the entire infrastructure stack as well as emerging cloud centric technologies all through a unified interface.
- The draft proposal will be submitted for approval by the Internet Engineering Task Force (IETF) for inclusion as a XMPP Extension

3.8. Problems of Existing Works

- **Weak security support**
  - First, the customer needs to know her data is encrypted so nosey sysadmins at the cloud data center can't troll through the data for interesting tidbits.
    - If the information is encrypted, who controls the encryption/decryption keys, the customer or the cloud vendor?
  - Integrity relates to the integrity of the data, in that it changes only in response to duly authorized transactions. So we need standards to ensure that. But they don't exist -- yet.
  - The last nagging security issue is availability: Will the data be there whenever you need it?
    - The answer here is an unqualified "maybe."
    - In February 2009, Amazon's S3 went down for almost four hours, wreaking havoc on several companies that use and depend on the S3 Cloud.
    - Amazon ascribed the cause to an unexpected spike in customer transactions.

- **Weak Privacy Support**
  - Private health data can goes public by mistake: Part of consumers' reticence to sign up for electronic personal health-care records — with or without services "in the cloud" — has to do with a handful of recent high-profile data breaches.
  - In April, the largest health insurer in the U.S., WellPoint, disclosed that records on as many as 130,000 of its customers had leaked out and become publicly available over the Internet.
  - User heath data and information uploaded into Clouds are not controlled by user.
  - Consumer's privacy may get loss in the cloud: Is there a law that keeps your data from being misused?
    - Yes. It is Health Insurance Portability and Accountability Act (HIPAA), but it does not offer lifecare service themselves.
    - Right now, disclosure of health information is out of control.

- **No infrastructure to support WSN integration to Cloud**
  - The existing Cloud based Healthcare system does not integrate wireless sensor network which is necessary to get real time information of patient or environment to monitor and analysis emergency situation.
  - Appropriate information dissemination mechanism is not explicitly addressed in the existing system to deliver sensor data or events to appropriate users of Cloud applications who subscribed.
  - Also there is a need to match published events with subscriptions efficiently.
  - A fast, scalable and efficient event matching algorithm is required for information dissemination system on Sensor-Cloud framework.

- **Lack of Dynamic Collaboration between Cloud providers**
  - When the Cloud provider is unable to provide quality of service to the end-user requests, it may result in service–level agreement (SLA) violation and end up costing the provider.
  - Existing commercial Cloud services are proprietary in nature.
  - They are owned and operated by individual companies. Each of them has created its own closed network, which is expensive to setup and maintain.

- Existing Cloud based solution does not consider the dynamic collaboration between Cloud providers which is obvious in near future.

- **Limitations of Existing Systems for**
  - Minimal support for human aware technologies
  - No support for Social Media Interaction and their usage
  - Mostly focus on medical data storage and management: focused on providing PHR (Personal Health Record) Service and EMR (Electronic Medical Record) Service
  - Lack of intelligent processing for Lifecare/Healthcare information
  - No use of standard ontologies for healthcare/lifecare data processing and usage as well as no use of modern technologies
  - Lack of sharing personalized healthcare information for different stakeholders
  - Lack of wellness services provisioning to consumers of the systems
  - No sufficient security and privacy support
  - Table 1 shows a comparison of the above existing work.

Table 1, A Comparison of Existing Works

Existing Works / Features	Korea u-Care System	MS. HealthVault	Google Health	Amazon	VPN-Cube™	UCI
Security	X	Weak	Weak	Weak	Weak	X
Privacy Control	Weak	Weak	Weak	Weak	Weak	X
USN Integration to Cloud	X	X	X	X	X	X
Sensor data dissemination to Cloud	X	X	X	X	X	X
Collaboration btw Clouds	X	X	X	X	V	V

OVERVIEW OF SC³

- Our research scope falls into Wireless Sensor Network, Social Media Interaction, Activity Recognition, Wellness, Lifecare, Healthcare, Cloud Computing, and Security & Privacy.
- In this section, we present an overview of our proposed solution, Secured WSN-integrated Cloud Computing for u-Life Care, called SC³.
- The abstract model of SC³ is shown in Figure 1816 where different services of SC³, their deployment, and their interactions are displayed.

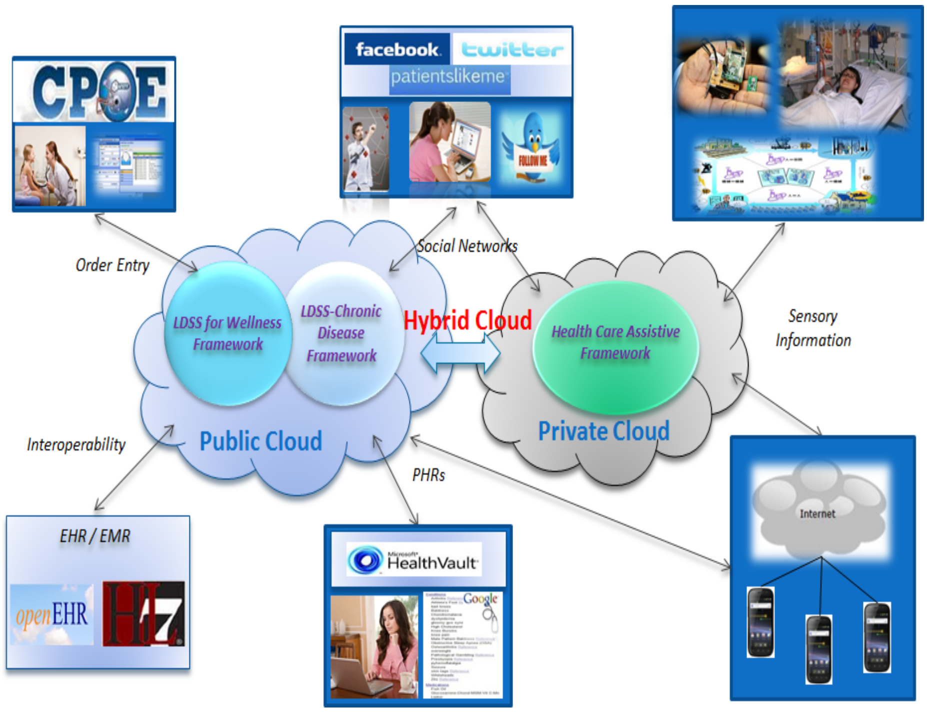


Figure 16 Abstract Architecture of SC³ for Wellness, Lifecare and Healthcare services

- We deploy a secure wireless sensor networks in u-Home environments for a purpose of monitoring and collecting sensor data.
- To enable u-Lifecare applications, we propose an Activity Recognition using sensors and smartphones.
- We have also incorporated the data share by user on social media in which they mostly share their experience. In addition, to support the healthcare applications and services, we have also used standard ontologies, guidelines, and methodologies.
- SC³ use technologies to provide personalized healthcare services based on activity/emotion recognition and context-awareness.
- It deals with diverse datasets, such as clinical, social and sensory data for constructing intelligent Knowledge-Base (KB) where these are mixed together and based on that final services

are provided as shown in Figure 17.

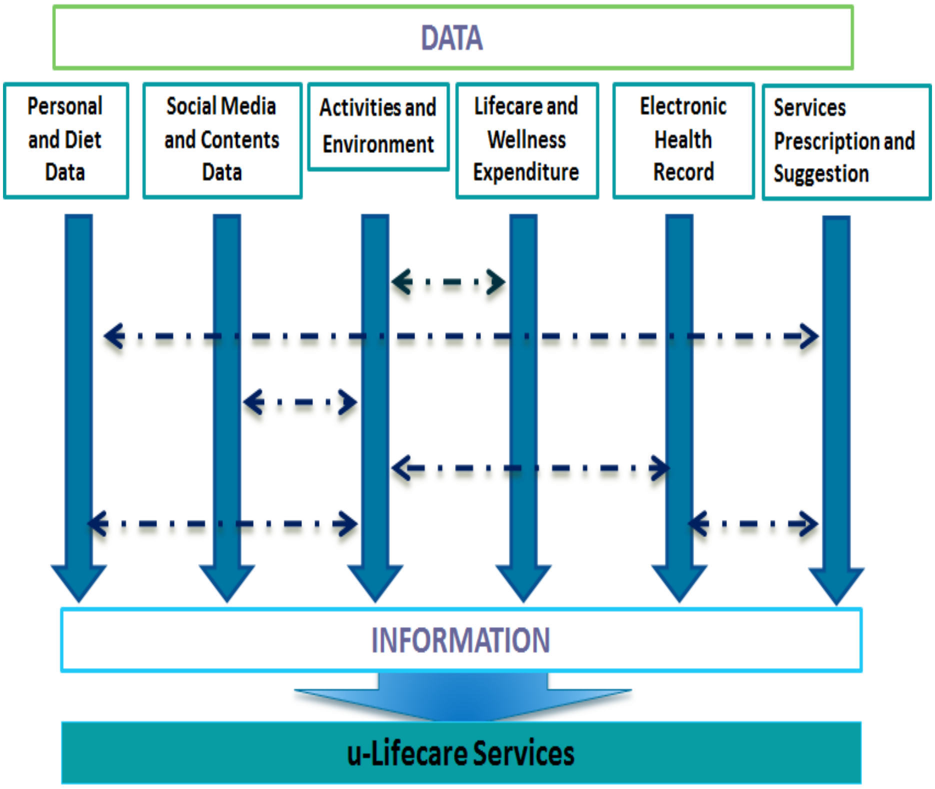


Figure 17 Diverse input modalities and their integration in SC^3 for Wellness, u-Lifecare and u-Healthcare services

- We utilized the cloud infrastructure for reducing u-Lifecare and u-healthcare cost.
- To facilitate interoperability among existing healthcare standards.
- To develop state-of-the-art, standard-based u-Lifecare Decision Support System (LDSS) that provides recommendations for chronic disease patients.
- We provide a security and privacy control of data and applications stored in Clouds.
- Different Clouds can collaborate with each other by using our dynamic collaboration method.
- Figure 1918 and Figure 2119 shows the abstract architectures of proposed SC^3.

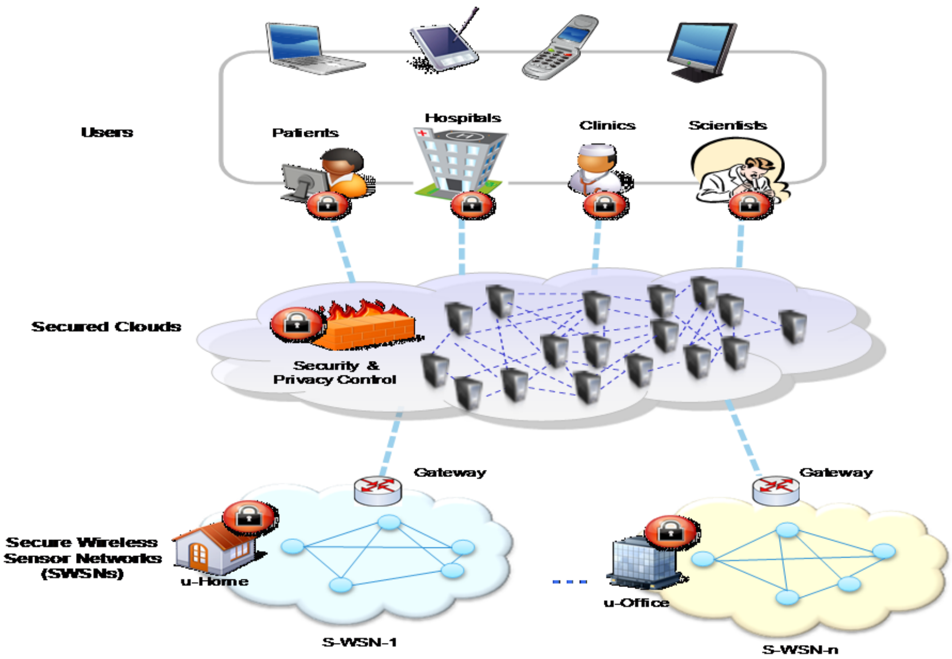


Figure 18 Overall Architecture of SC3

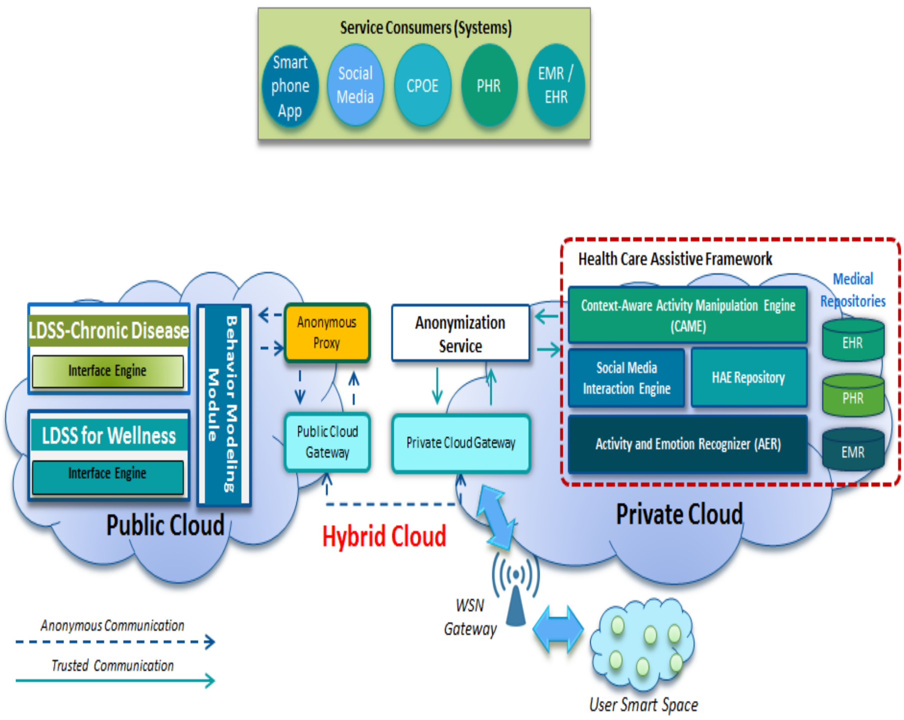


Figure 19 Abstract Architecture of SC^3

- SC^3 as an infrastructure is composed of three main services that facilitate six different applications based on information collected from different diverse input sources as shown in Figure 20.

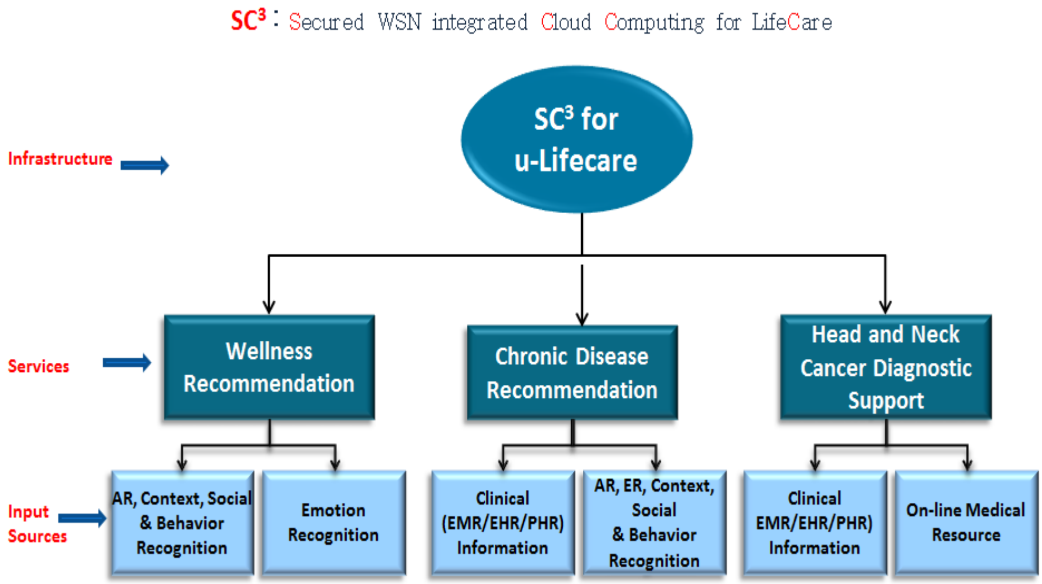


Figure 20 Service and input sources level decomposition of SC^3

- The layered architecture of SC^3 is given in Figure 21 that shows the overall components of SC^3 supporting Wellness Service, u-Lifecare Service, u-Healthcare Service, Privacy and Security Service, and Cloud infrastructure support for all the services execution.

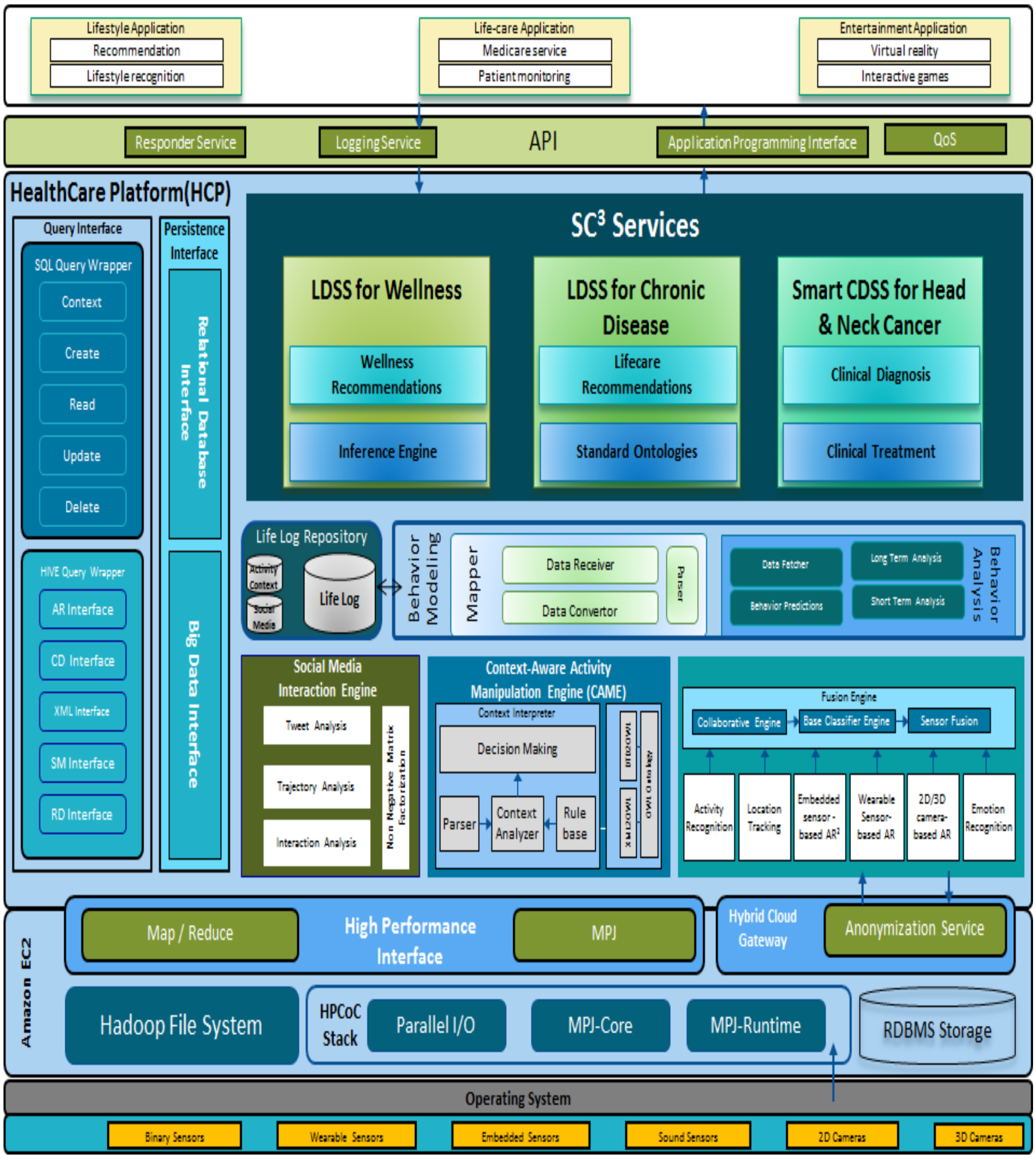


Figure 21 Detailed Layered Architecture of SC^3

4.1. Services of SC³

- In this section, we present different services SC³ like LDSS for Wellness, LDSS for Chronic Disease, and SmartCDSS for Head & Neck Cancer.
- Each of the service is briefly discussed in their respective sections and then their detail is presented in Chapter 5 and 6.

4.1.1. LDSS for Wellness

- Wellness is to keep yourself well, and do things to help yourself feel better.
- Goals of Wellness
  - Increase the span of healthy life
  - Eliminate health disparities
  - Increase access to personalized information and services for all people
  - To make people responsible for your own healthy life
- Wellness is the product of healthy lifestyles just like fitness is the product of regular exercise. There are four essentials for wellness as shown in Figure 22.

Figure 22 Essentials for Wellness

- The wellness services needs to integrate the different context information coming from diverse modalities and use it for lifestyle analysis, behavior identification, and personalized lifecare and wellness recommendations as shown in Figure 23 and Figure 24.

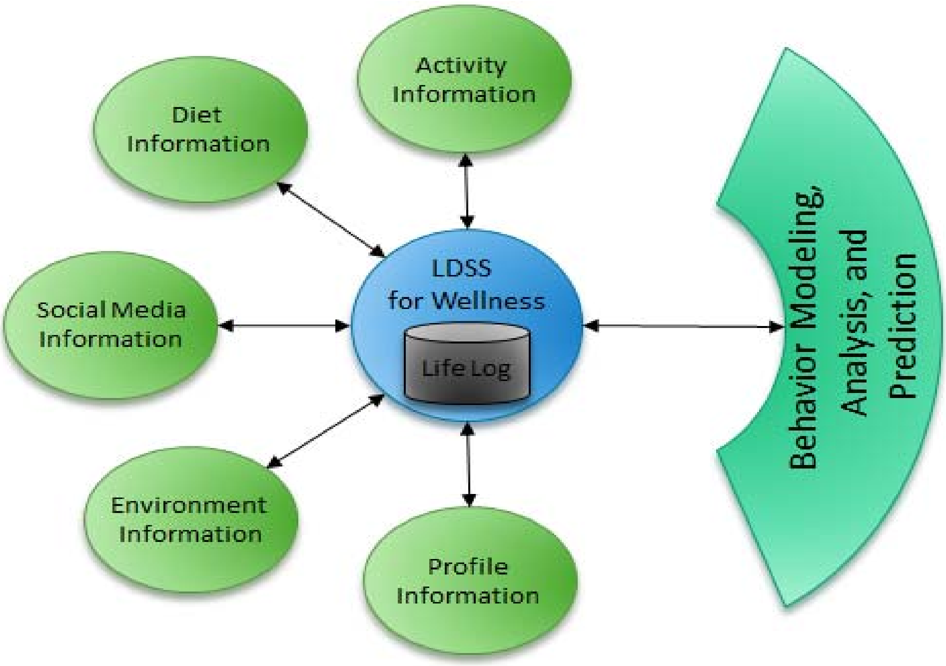


Figure 23 Integration of input data

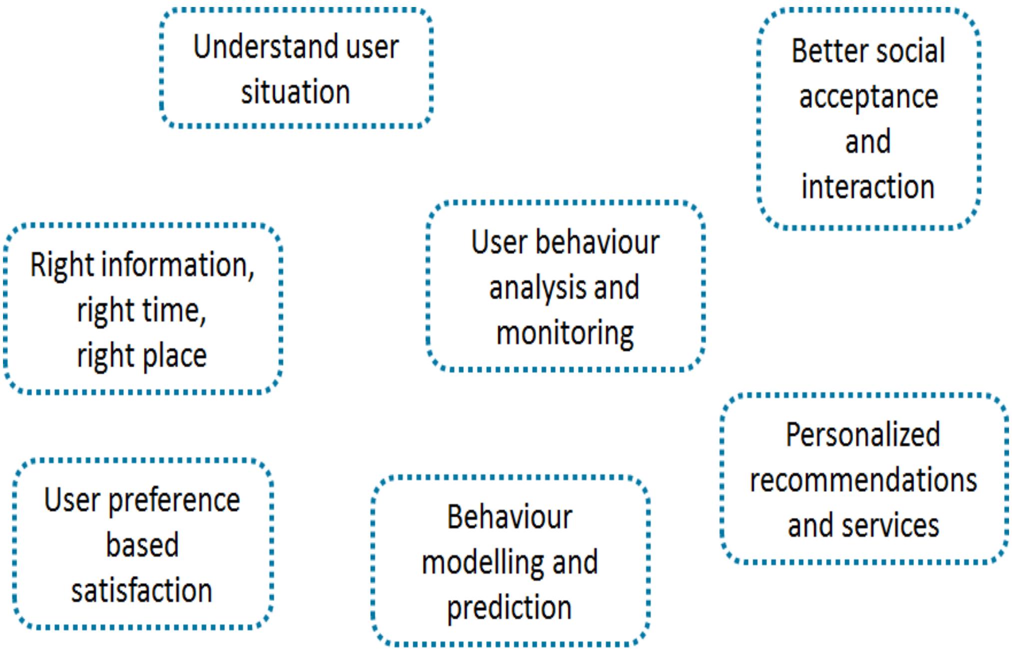


Figure 24 Types of wellness recommendations based on the input context/data

- The diverse input context is collected using digital devices and social media interactions, and are merged together and stored in Life Log as shown in Figure 25.
- This Life Log context is later used for wellness services.
- The overall layered architecture of LDSS for Wellness is shown in Figure 26. The design of LDSS for Wellness is for:
  - To use technologies for activity recognition, social interaction, diet information, environment information and provide personalized lifecare services based on the context of information
  - To deal with information from diverse input sources
  - To use the integrated information for behavior analysis, behavior prediction, and personalized recommendations
  - To utilize cloud infrastructure for reducing cost and high performance
  - To support security and privacy

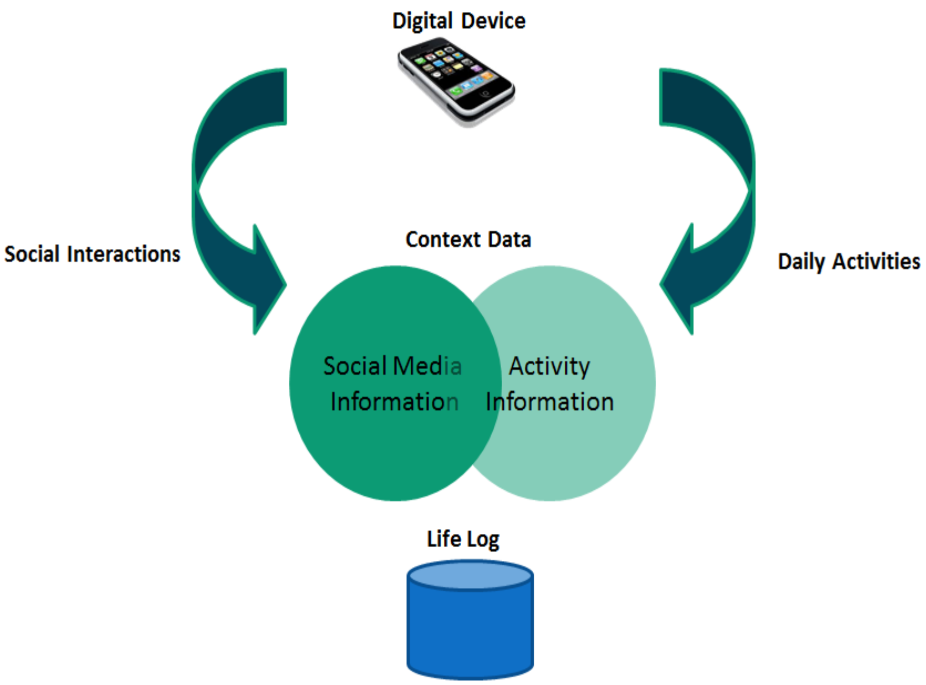


Figure 25 Composing Life Log and logging information from diverse sources into Life Log

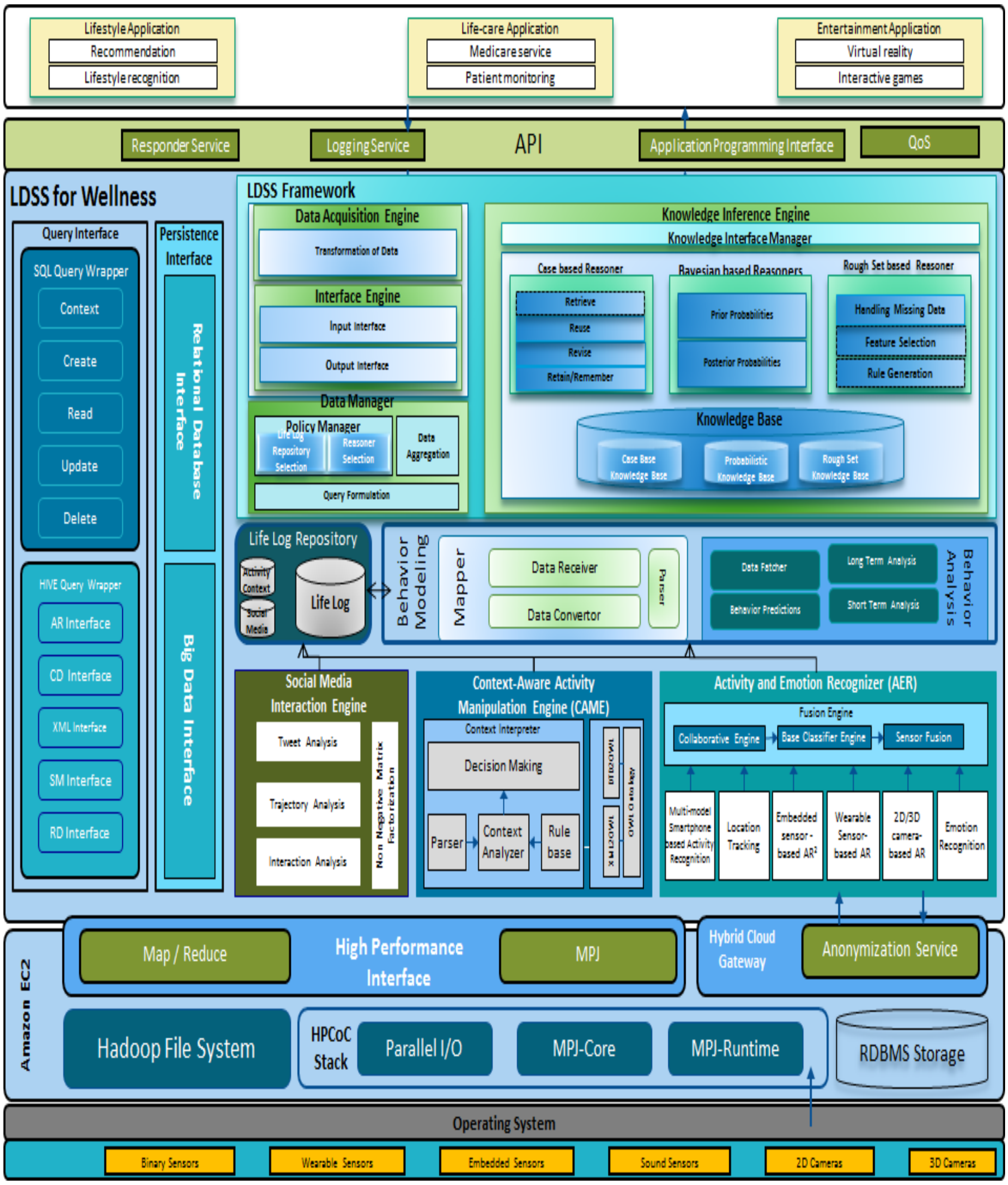


Figure 26 Layered Architecture of LDSS for Wellness

4.1.2. LDSS for Chronic Disease

- Like Wellness, LDSS for Chronic Disease facilitate chronic disease patients in their home environment for their daily life activities and prescribed medication and treatment.
- This helps in reduction of hospitalization and reduces healthcare cost.
- Goals of LDSS for Chronic Disease are:
  - To increase the span of healthy life for chronic disease patients and facilitate them to perform daily life activities.
  - Eliminate health disparities
  - Increase access to personalized information and services recommended by doctors and the developed system
  - To make people responsible for your own healthy life
- The over abstract architecture of LDSS for Chronic Disease showing its interaction with different external agents is shown in Figure 27.

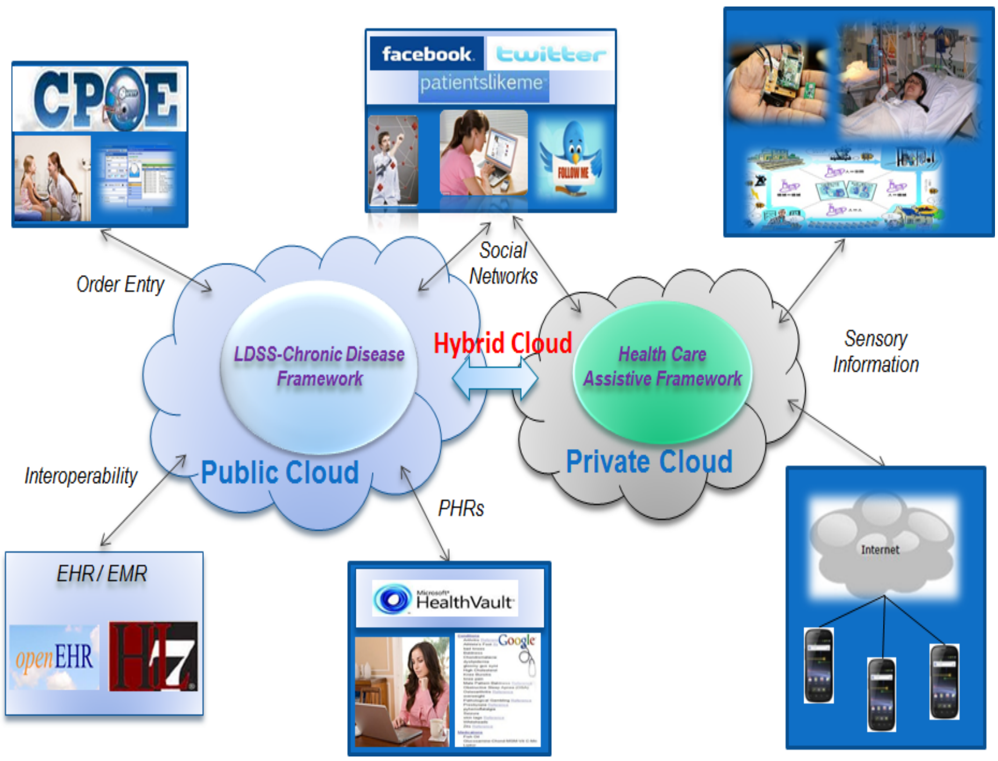


Figure 27 Abstract architecture of LDSS for Chronic Disease patients and its interaction with external agents

- Subcomponents are:
  - It contains self-evolutionary and dynamic knowledge base
  - Standard interfaces allow integration with diverse applications
  - Decoupled Knowledge bases (KBs) to support heterogeneous KB Reasoner

- Individual ownership on KBs using hybrid cloud
- Secured interfaces to share anonymized user's private data
- Comprehensive guidelines and recommendation for chronic disease patient based on social, sensory and clinical datasets

▪ The diverse input context is collected using digital devices and social media interactions are also used for LDSS for Chronic Disease.

▪ The overall layered architecture of LDSS for Chronic Disease is shown in Figure 28. The design of LDSS for Chronic Disease is for:
  - To use technologies for activity recognition, social interaction, diet information, environment information and provide personalized lifecare services based on the context of information
  - To deal with information from diverse input sources and uses standard technologies for personalized and prescribed service provisioning.
  - To utilize cloud infrastructure for reducing cost and high performance
  - To support security and privacy

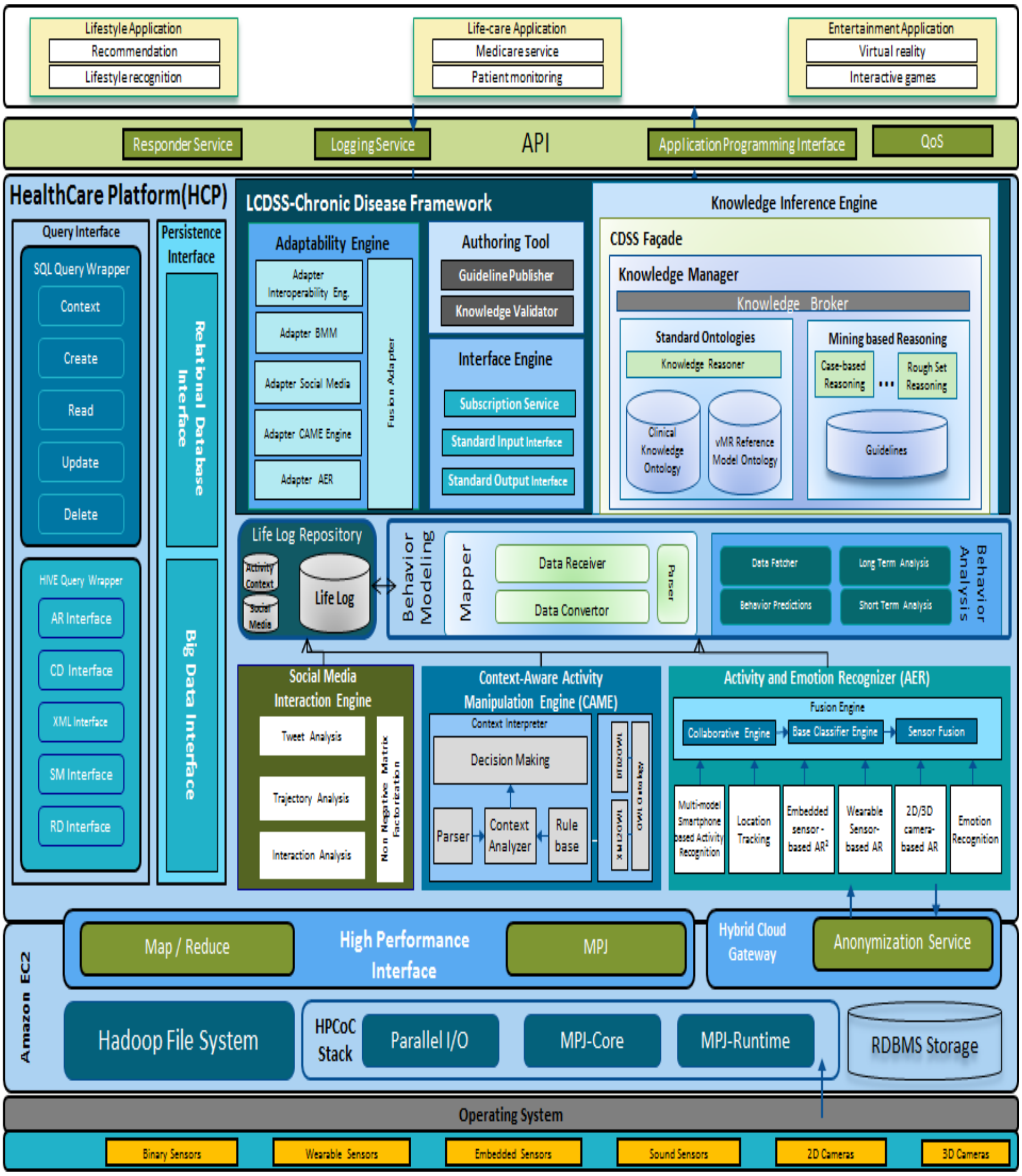


Figure 28 Layered Architecture of LDSS for Chronic Disease

4.1.3. SmartCDSS for Head and Neck Cancer

- Any tool that helps us make better medical decisions thereby reduces clinical errors and improves quality of care. The goals are:
  - Generating alerts and reminders
  - Diagnostic assistance
  - Therapy critiquing and planning
  - Image recognition and interpretation
- CDSS improves performance of clinicians and cares for patients in different application domains. Some of the statistics are shown in Figure 29.
- The proposed SmartCDSS for Head & Neck Cancer has following contributions.
  - Self-evolutionary and dynamic knowledge base using online resources
  - Standard interfaces which allow integration with diverse applications
  - Decoupled Knowledge bases (KBs) to support heterogeneous KB Reasoner
  - Individual ownership on KBs using hybrid cloud

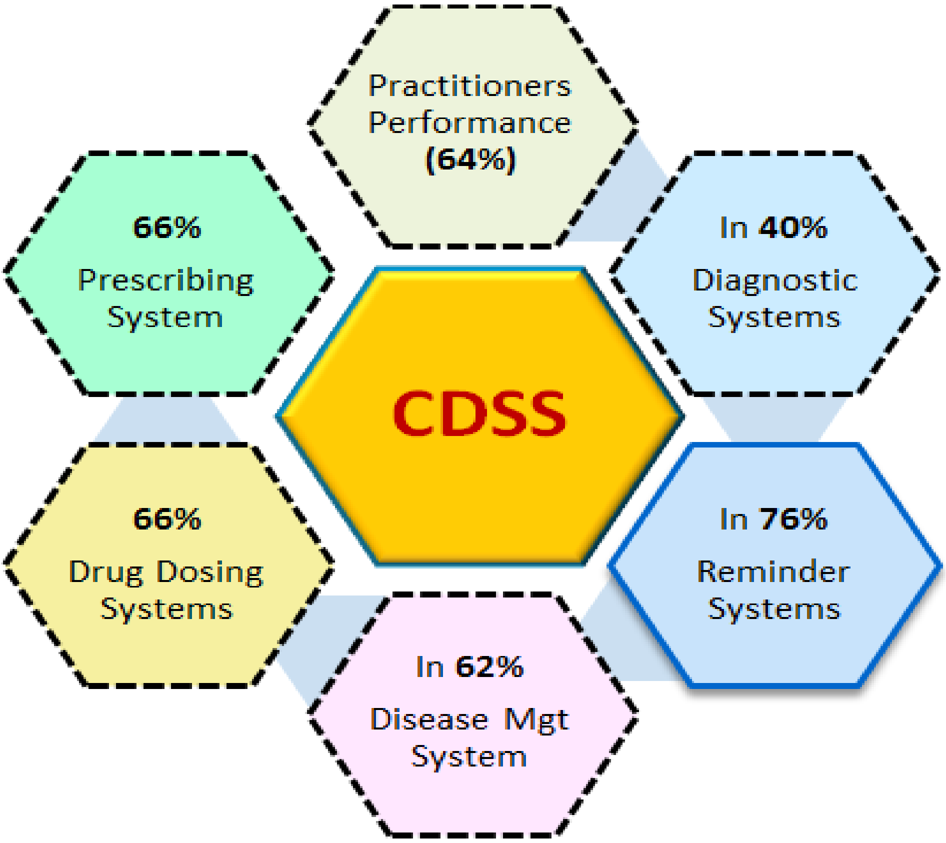


Figure 29 Statistics of CDSS usage in different domains and its effectiveness

- The overall architecture of SmartCDSS for Head & Neck Cancer is shown in Figure 30. Its design is as follow:

- To use technologies for diagnosis and treatment.
- To use expert knowledge and design and development of the system.
- To use hospital workflow for development of system logic and execution.
- To deal with information from diverse input sources and uses standard technologies for personalized and prescribed service provisioning.
- To utilize cloud infrastructure for reducing cost and high performance
- To support security and privacy

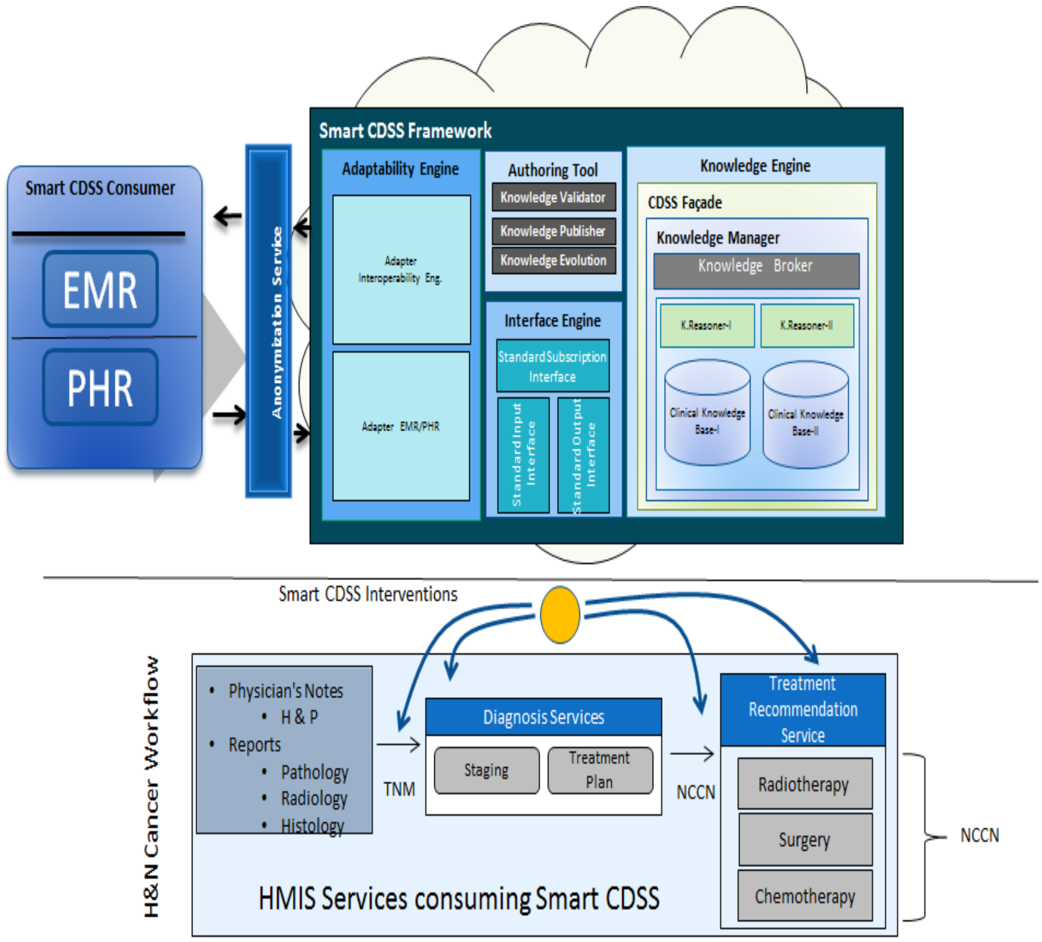


Figure 30 Overall architecture of SmartCDSS for Head & Neck Cancer

4.2. Our Challenges

- **Integration of IT and u-Lifecare Technologies**
  - Different sensors, smartphones, social media, standard knowledgebase, standard technologies, and cloud infrastructure exist that can benefit in achieving better u-Lifecare systems and services.
  - So the objective is to find a way of integrating the technologies for lifecare systems and services to achieve the vision of u-Lifecare as shown in Figure 31.

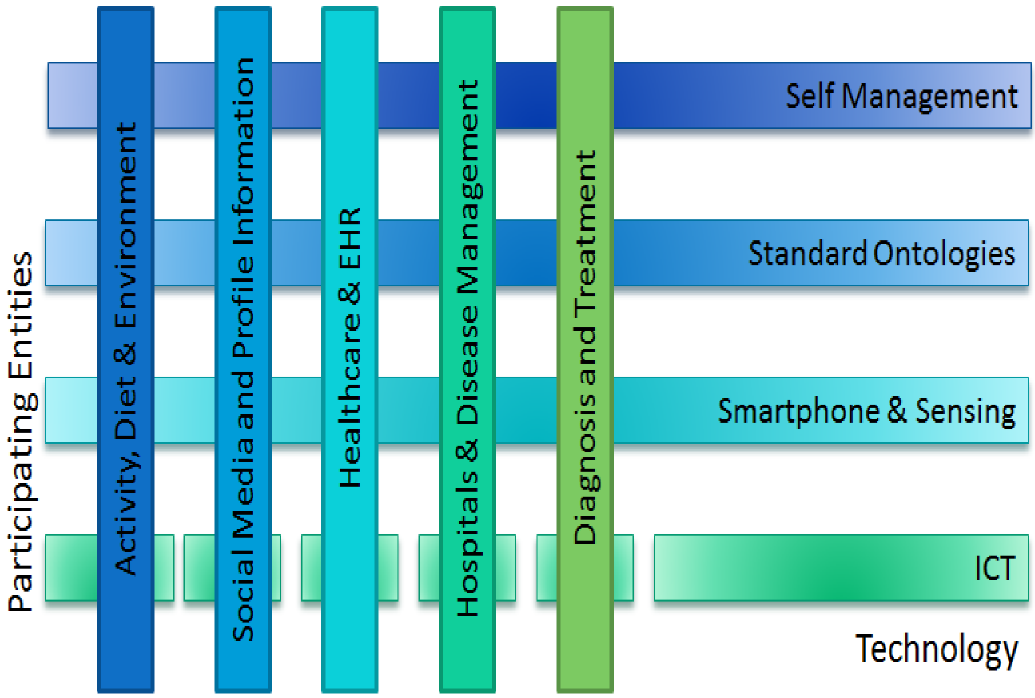


Figure 31 Integration of IT and Lifecare to achieve u-Lifecare with SC^3

- **Low resource sensors and smartphones**
  - Sensor nodes are very limited in term of energy, communication, and computation. Therefore, in order to make the algorithms feasible on sensor devices, they must be lightweight and energy-efficient.
  - To make the system and services more usable, we are using the inbuilt sensors of smartphones and developing applications on smartphones so that the services are easily consumable.

- **A huge number of users, and it increases dramatically**
  - As the number of users accessing Clouds increase dramatically, how to support individual users to declare their privacy preferences accurately.
  - Authentication method must be usable on various devices with wired or wireless-enable connection over the Internet.

- Besides, appropriate privacy policy implementation is very hard. User must agree to provide his/her sensitive information which is not always possible

- **Data dissemination challenges**
  - In case of dissemination of information to mobile clients, the mobility can cause their access brokers to be changed, which can bring problems in dissemination of subscriptions and distribution of matching results.
  - Cloud computing can be used to provide robustness and scalability at runtime.

- **Dynamic collaboration challenges**
  - Finding appropriate group strategy to minimize collaboration cost in dynamic collaboration is really a major challenge.

- **Standard u-Lifecare services and deployment of system in real environment**
  - The services need to be based on standards which are acceptable for the stakeholders.
  - The deployment of system in real environment like nursing home.

ANALYSIS OF SC^3 SERVICES

- In this section, we present the detailed analysis of all services of SC^3.
- These services include LDSS for Wellness, LDSS for Chronic Disease and Smart CDSS for Head and Neck Cancer.
- The details on the components of the service and the main contribution of that component.
- The following sections will present each service in details.

5.1.LDSS for Wellness

- LDSS for Wellness system facilitates users of the system to manage their daily life activity, social, and diet data and then reuse it for the behavior analysis.
- This data is also used to analyze user situation and provide wellness recommendations which keep the user active and healthy.
- It is very time and resource consuming for a user to visit hospital or caregivers for suggestions which he/she can normally avail using the wellness system.
- The daily life activities of users are monitored and collected from different sources, such as social media, sensors, camera, smartphones, and environmental sensors.
- Therefore the system has input from different sources which is logged in life log used for behavior analysis and daily life recommendations.

5.1.1. Human Activity Recognition (HAR)

- It consists of the multimodal sensor based activity and emotion recognition modules.
- The outputs of these individual independent modules are fuse together to predict the real-time high level contexts.
- The whole architecture is decomposed into the subcomponents and details are provided in the subsequent section.

5.1.1.1. Audio Based Emotion Recognition

- **Introduction**
  - Emotion is a mental state that arises spontaneously. In daily life, emotion is not only an effective way to convey our intention in communication but also a good indicator of our mental health. That is the reason why automatic detection of human emotions is an important factor to enhance the quality of the service provided by the computer such as human-computer interaction [Cowie2001, Schuller2004], lifestyle monitoring in ubiquitous health care systems [Tacconi2008].
  - While human emotion can be expressed by a variety of physiological changes such as speech, blood pressure, heart rate, facial expression, etc.; many researchers prefer acoustic speech as a source of emotion [Ayadi2011, Bitouk2010, Iliev2010, Lee2005] because speech signal is the most commonly used and most natural method of human communi-

cation.

- Finding the best feature extraction and classification for emotion recognition from speech signal still are the current challenge for researchers.

- **Problems in Existing Systems**
  - When classification methods have achieved successful results with different classifiers, feature extraction for emotion recognition is still an open area.
  - There are a lot of feature extraction methods that have been used to extract more and more relevant features for emotion recognition from speech signal [Cowie2001, Lee2005, Banse1996, Gobl2003, Nwe2003]. But we still don't know what the best features for this task of recognition are.

- **Requirements**
  - In order for system working properly and efficiently, there are many requirements are needed, some of important requirement as follows:
    - Training data is needed to be recorded as much as possible with different voices and emotions. All other sounds affecting to voice quality should be avoided.
    - Emotions to be recognized should be predefined and differentiate together clear enough.

- **Proposed Solution**
  - The main function of this component is recognizing emotional content of unknown speech signal. It takes a speech sentence from audio sensor as input and then process to classify to one of different emotions. The output emotion has the form of text label such as anger, happiness, boredom, sadness, normal, etc.
  - This AER component has 3 main sub-components called:
    - Data Acquisition
    - Feature Extraction
    - Classification
  - Each of them has a unique important role in the whole process of this component. It shares the similar architecture with almost classification applications that need to have at least three parts of collecting data, extracting features and classifying.

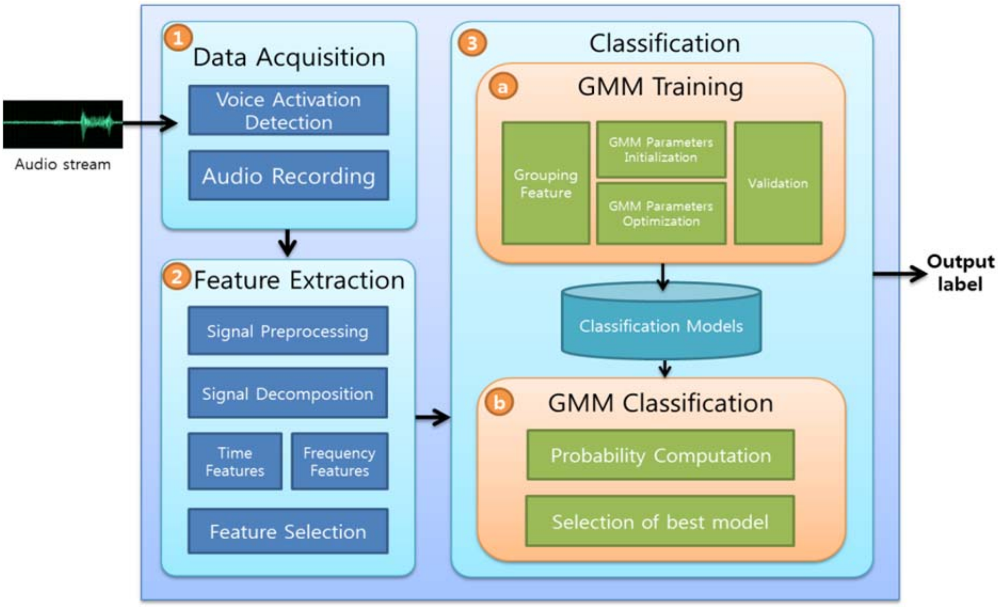


Figure 32 General Architecture of Cloud Computing

- **Data Acquisition:** The main purpose of this sub-component is to detect and record audio data for further process. Input audio stream from sensor is continuous and contains audio and speech signal as well.
- We need to detect when speech signal appears in audio stream so that we can record speech signal. To reduce the complexity of this process, we can assume that the audio stream contains speech signal only, so we just need to detect speech signal by using energy threshold.
- Whenever the average energy is larger than threshold, the audio recording module is activated to start recording speech signal. If average energy becomes lower than threshold, it means a sentence is completed, recording module is stopped, and then recorded speech signal is stored as a file.

- **Feature Extraction:** This is one of two most important parts that takes a speech sample as input and then generates a feature vector or a sequence of feature vectors. The main purpose of this process is extract the most relevant features that can describe clearly about the processing signal, and can be used to classify between different classes of signal. Specially, we need to extract the most valuable features that can classify emotional states from speech signal. Depend on what kind of signal and application, there are different processes inside feature exaction component. In this architecture, we have:
  - Signal Preprocessing: to reduce noise that can affect to quality of signal and reduce the accuracy of classification. In this process, we can apply spectrum subtraction to reduce background noise.
  - Signal Decomposition: Matching Pursuit is applied to decompose the input signal into a block of atoms that are small-predefined signals [Mallat1993, Umapathy2005a, Ummapathy2005b, Chu2009]. And then depend on the distribution of

these atom we can extract different features.

- Time feature and frequency features are extracted from output block of atoms by using temporal and spectrum histogram.
- Feature Selection: In feature extraction stage, we can use different kind of features, some of them are good, but some of them are not good, they affect to accuracy of classification. We apply feature selection here to reduce the irrelevant features in order to increase the accuracy.

- **Classification:** In this stage, a machine learning algorithm is applied to learn the properties of processing signal, and to differentiate between different classes of signal. To make balance between simplicity and performance, we apply Gaussian Mixture Model (GMM) as classifier to recognize different emotional states in speech signal. GMM is a lightweight classification algorithm, so that it is easier to apply on smartphone that have limited computational resources. Similar with other classification, there are 2 main processes training and classifying.

  - ***Training process:*** This training is used to learn properties of each class of signal by group all training data of class and apply Expectation Maximization to generate the parameters of GMM model. Each emotional class of speech signals is used to generate a GMM model. For example we have 4 classes Normal, Angry, Happy and Sad, so 4 GMM models are trained in this step.

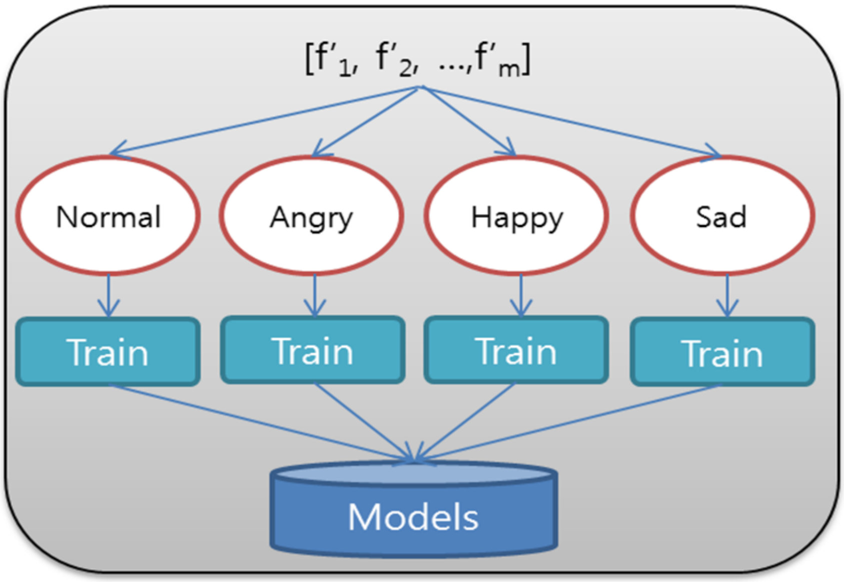


Figure 33 GMM Training process for 4 emotional classes

  - Grouping features: In order to train models for different classes, we have to extract features for all training data and manually split them into different groups according to emotional classes. For each group, we apply GMM training to generate model for this one.

- GMM Parameters Initialization: we initialize parameters for each GMM model randomly
- GMM Parameters Optimization: in this process, EM algorithm is used to optimize parameters of a model. This is a recursion process to fit parameters of GMM model with distribution of training data in iteration. The process is stopped when it reaches to a number of iterations or is below a threshold.
- Validation: This process is to validate predefined parameters of a model by using them to recognize a set of validating data. Which parameters have better performance and accuracy are chosen for classifying stage.
- All trained models are stored in a local database or provided to mobile side for further process.

- ***Classifying stage:*** This process is deployed at PC side or mobile side as well depending on what kind of application. The main purpose of this process is to find from all trained models the best model that has the largest probability with unknown input feature vector compare with other ones.

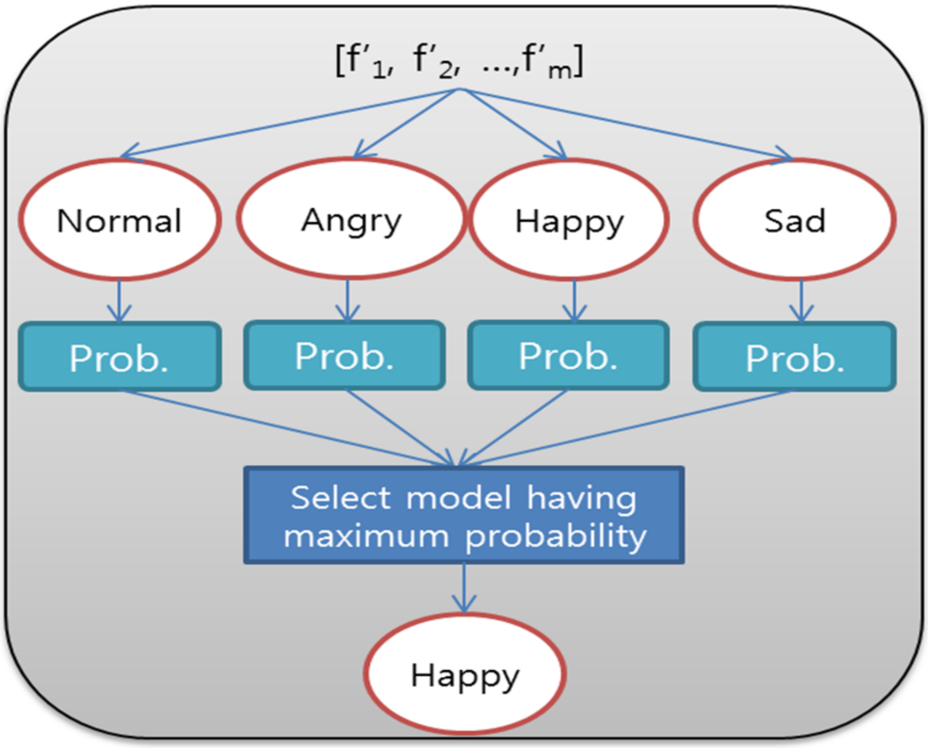


Figure 34 Classifying unknown input signal given 4 GMM models

- Probability Computation: when the input feature vector is available, its probability with each GMM model is computed using Gaussian distribution.
- All probabilities are compared to select the largest one. And then we consider the

model with the largest probability is the output of classifying stage.

- **Overall System Flowchart**
  - Following flowchart show how the system works. It contains two phases: training and testing.
  - This shows that how different components work together and how the output of one component flows to the other component where it is used as input for further processing.

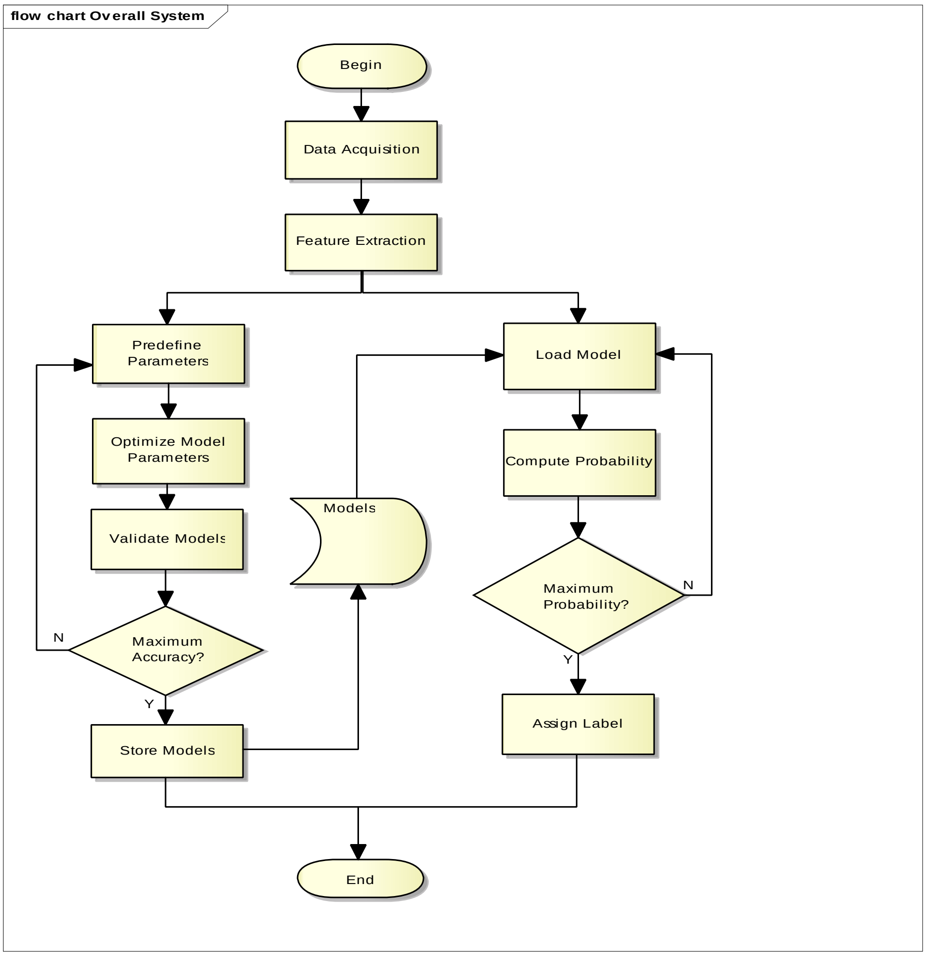


Figure 35 Overall system flowchart

5.1.1.2. Video Based Activity Recognition

- Automated systems are becoming increasingly involved in our daily lives. To correctly handle a situation either to help people to finish a job in a friendly environment or to prevent people from error in a hostile situation, the system must know what is happening. Human activity is a very important factor in this process. Knowledge of what people are doing can enable a wide range of assisting technologies, smart appliances, and aware environments.

- Human activity recognition has a good application in smart home technologies in all over the world to monitor human activities and take necessary steps. For instance, the SmartBo project in Norway includes a two-room house for the elderly with mobility impairments and cognitive disabilities. In this project, its main system including lighting, doors, windows, and shutters is controlled by smart devices and sensors and generates an alarm when something goes wrong [Elger1998].
- Security is another major application domain that requires an understanding of activity. Video surveillance is becoming standard in all sorts of public areas as well as the private backyard. Having a person behind each camera is simply impossible. An automated filter system as well as abnormal situation detection system has generated enough market demand for companies, which provides shopping customer identification.

- **Introduction**
  - Video-based activity recognition refers to an algorithm that a computer system uses to automatically recognize what human activity is being or was performed, given a sequence of images (video frames). In recent years, this problem has caught the awareness of researchers from manufacturing, academia, safety agencies, consumer agencies, and the general populace too [Turaga2008]. In the vision literature, the words ‘Action’ and ‘Activity’ are habitually utilized identical. The word ‘Action’ means the patterns of simple motion that usually executed by a single person and normally fixed for short durations of time, on the order of tens of seconds [Turaga2008]. Examples of actions include bending, walking, swimming etc., as shown in Figure 36.

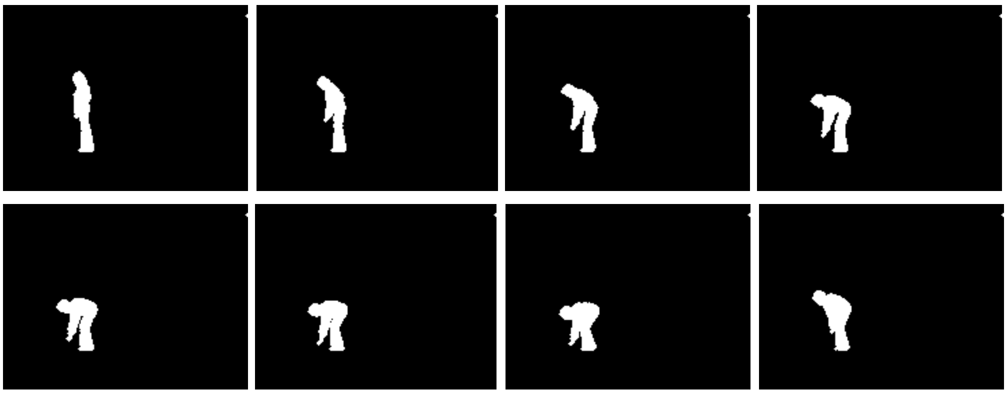


Figure 36 Examples of bending actions.

  - On the other hand, the word ‘Activities’ refers as the complex sequences of actions that are performed by several humans who could be interrelating with each other in an inhibited manner. They are normally characterized by much longer temporal durations, e.g. two persons shaking hands, a football team scoring a goal or a co-ordinated bank attack by multiple robbers [Georis2004] as shown in Figure 37.

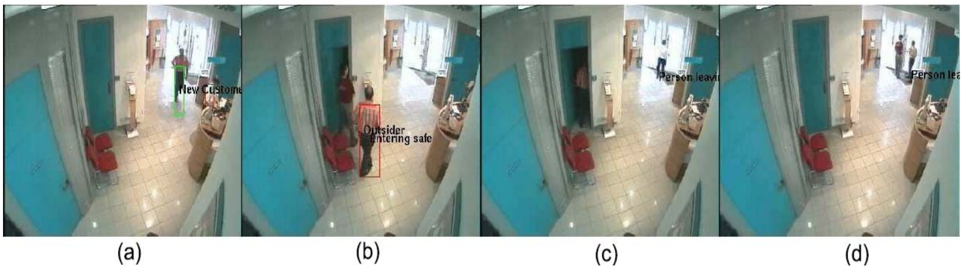


Figure 37 Medium field sequence of a simulated bank attack (courtesy [Georis2004]).

- (a) Person enters the bank. (b) Robber is recognized to be a stranger, and is entering the bank secure. (c) A customer flee, (d) Robber creates an exit [Turaga2008].
- Generally there are two types of activities that a human is performing commonly. The first one is low-level activities also called micro activities (μ-activities) e.g. sitting, standing, running, walking, one hand waving or two hands waving etc. The second one is high-level activities also called macro activities e.g. watching TV, playing tennis, ridding bus etc. the general architecture of the high-level activities is given as in Figure 38.

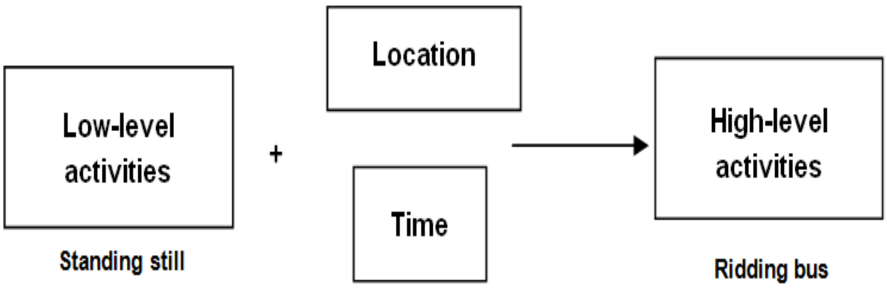


Figure 38 General architecture of high-level (macro) activities.

- The architecture of human activity recognition system consists of three main modules: segmentation, feature extraction, and recognition. *Segmentation* is used to extract the human body shape from video frames and generate corresponding binary images. *Feature extraction* deals with getting the distinguishable features each body shape and quantizing it as a discrete symbol. In *recognition* module, a classifier such as HMM (Hidden Markov Model), or GMM (Gaussian Mixture of Model), or SVM (Support Vector Machine) is first trained with training data and then used to generate the label of human activity contained in the incoming video data as shown in Figure 39.

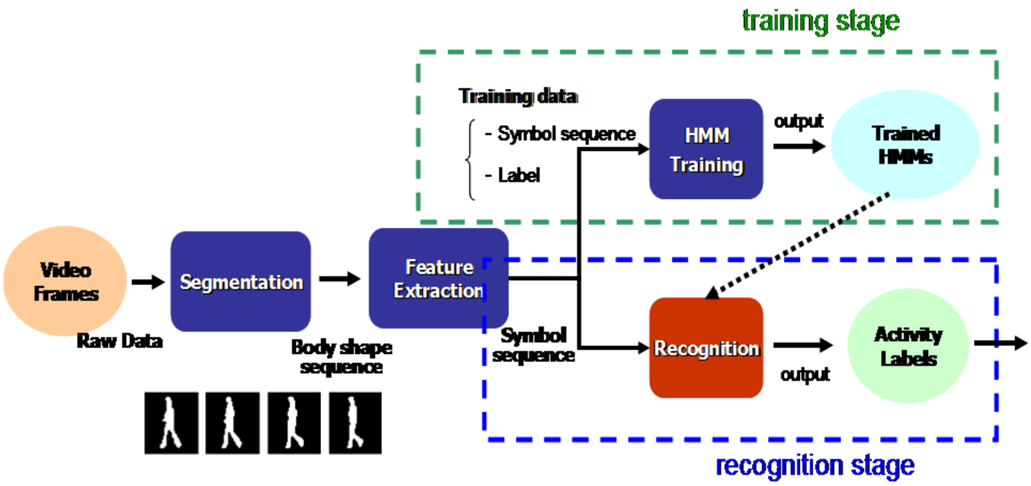


Figure 39 Block diagram of a typical video-based AR system.

- **Problems in Existing Systems**
  - Numerous video-based human activity recognition systems have been proposed; however, recognizing human activity recognition accurately is still a major concern for most of these systems. This lack of accuracy could be attributed to various causes, such as different light conditions for training and test images, lack of accurate automatic human body segmentation methods, and high similarity among different activities that occurs in the presence of low between-class variance in the feature space. A typical activity recognition system consists of three basic modules.

  - **Human Body Segmentation.** The accuracy of recognition module is completely depends on automatic human body segmentation. Some methods have been investigated for human body segmentation including [Siddiqi2010, Uddin2008]; they just subtracted the empty frames from the video frame to segment the human body as shown in Figure 40.

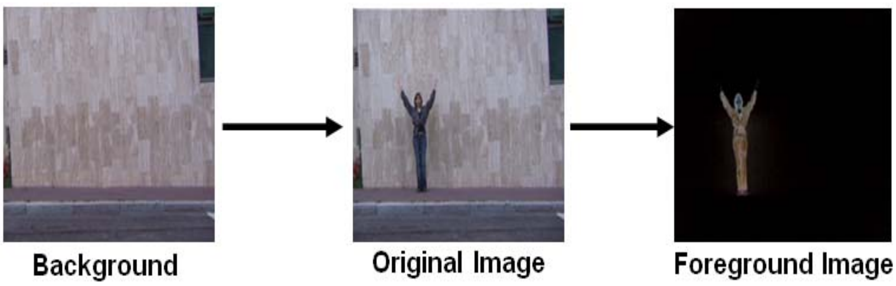


Figure 40 Body segmentation using empty frame subtraction from the video frame.

  - However, these techniques were not applicable in real world environment, because in some situation, there is no empty frame, so it is very hard to segment the human body by using these techniques. Due to these limitations, these methods are known as heuristic techniques. Another method has been developed by [Doulamis1999] for segmentation, but this technique produced artifacts due to occlusion. A well-known method was proposed by [Kass1988], but mostly, the human body is in concave form, and to segment the concave objects is one of the limitations of this method. Some of the existing methods often involve modeling of the human body and/or the background, which normally requires extensive amount of training data and cannot efficiently handle changes over time. Recently, active contours [Chan2001] have been emerging as an effective segmentation technique in still images.

- **Feature Extraction.** Moreover, the accuracy is also depends on the extraction of good features that means, the effectiveness of the extracted features will also affect the accuracy of recognition module Some well-known techniques employed for the feature extraction module include [Uddin2008, Uddin2010]. They used Principal Component Analysis (PCA) and Independent Component Analysis (ICA) for feature extraction. However, PCA yields uncorrelated components. If the data have a Gaussian distribution, the uncorrelated components are independent. However, if the data are merged of non-Gaussian components, then PCA fails to extract components having non-Gaussian distribution [Buciu2009]. Also, PCA is an unsupervised technique that locates PCs at the optimally diminished dimension of the input. For human activity, it only focuses on the global information from the binary silhouettes, which results in low accuracy [Kim2010]. To solve the limitation of PCA, a higher order statistical approach Independent Component Analysis (ICA) has been employed by [Uddin2010] for feature extraction. However, ICA is slow to train when the dimension of the data is bulky. Moreover, these features are receptive to decipher and scaling of human body postures which concerns the silhouettes extraction procedure. Also, ICA is very weak in managing the inputs. If there are plenty of video frames, exploited as input, ICA does not have the capability to organize it, due to which some time ICA cannot retrieve the desire features as shown in Figure 41.

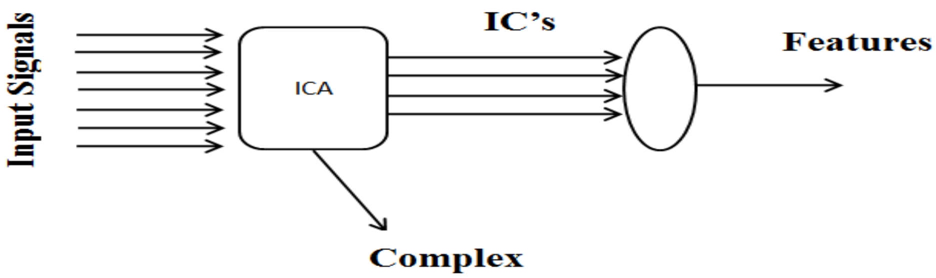


Figure 41 Weakness of ICA in-term of feature extraction.

- **Recognition.** While there is quite a large number of recent research focusing on improving the feature extraction stage [Huang2007, Robertson2006, Xue2010], almost all the proposed activity recognition systems utilize conventional learning methods, such as Hidden Markov Model (HMM), Support Vector Machine (SVM), Gaussian Mixture Model (GMM), Artificial Neural Network (ANN), etc. Among these classifiers, HMM is pointed out by several studies to be the most commonly used method [Uddin2008, Uddin2010]. Even though HMM is widely used for activity recognition, still it has serious deficiencies. It is very difficult to represent multiple interacting activities by using HMMs [Gu2009]. HMMs are also incapable of capturing long-range or transitive dependencies. Additionally, without considerable training, an HMM may not be proficient to recognize all of the possible observation sequences that can be consistent with a particular activity [Kim2010].

- **Requirements**
  - To overcome the limitations of the existing systems and to provide better decision on activity recognition in smart homes, there are different requirements that need to be satisfied.
    - Information coming from video (2D) camera.
    - To store the information (i.e., human activities) coming from input source, we need to decompose it into frames and then store them for decision making.
    - For the system to work properly, smooth communication among all the components is required.
    - The need is to develop an automatic human body segmentation technique.
    - For knowing the activity, a recognition engine is required that will facilitate the users to understand their daily activities.

- **Proposed Solution**
  - **Unsupervised Segmentation Technique.** Accordingly, the purpose of this study was to propose an accurate and robust activity recognition system capable of exhibiting high recognition accuracy. A model is proposed that use unsupervised segmentation in combination with motion information. Specifically, in each video frame, an active contour (AC) [Kass1988] is evolved to capture the human body and the motion information (i.e., optical flow) is used to move the contour toward the human position in the next frame, where the contour is again evolved to detect the exact boundary of the body that avoid the need of supervised learning. In AC models, the initial contour should be close to the object in order to converge correctly. This can be done manually in static images but will not be feasible in video data which have a large number of frames. Automating the initialization of AC is therefore needed. Optical flow is a good candidate for this purpose because it contains the relatively exact direction and magnitude of the motion between consecutive frames. According to the afore-mentioned classification, the proposed approach falls into the shape-based category with partial incorporation of temporal context. By “partial”, means that the temporal information is not embedded in the contour evolution itself but instead is used as a guideline for positioning the initial contour. The proposed approach is unsupervised, i.e., no training data or prior human model is needed. Experimental results show that AR using segmentation results of our approach has accuracy comparable to the one using manual segmentation, proving the feasibility of our model in practical activity recognition.

  - **Robust Feature Extraction & Recognition.** We proposed a complete approach for video based human activity recognition that employs wavelet transform and hidden conditional random fields model. To obtain the feature vectors, symlet wavelet family, was tested and decomposed up to 4 levels to recognize the human activities. Some of the highest coefficients were extracted from each level of decomposition at the feature extraction stage. These coefficients were based on the average frequency of each video frames and the time difference between each frame, and then a novel hidden conditional random fields model was applied for recognition. The proposed method was tested on a database of nine activities. Data collected by ten different people, so the whole collected samples of activities were 90. The recognition results were compared with some of the existing techniques that used Principle Component Analysis (PCA), Independent Component Analysis (ICA), and Hidden Markov Model (HMM). These results showed better improvement in performance when compared with the existing techniques applied on the same dataset.

- **Overall System Flowchart**
  - The overall flowchart for the proposed activity recognition system is shown in Figure 42.

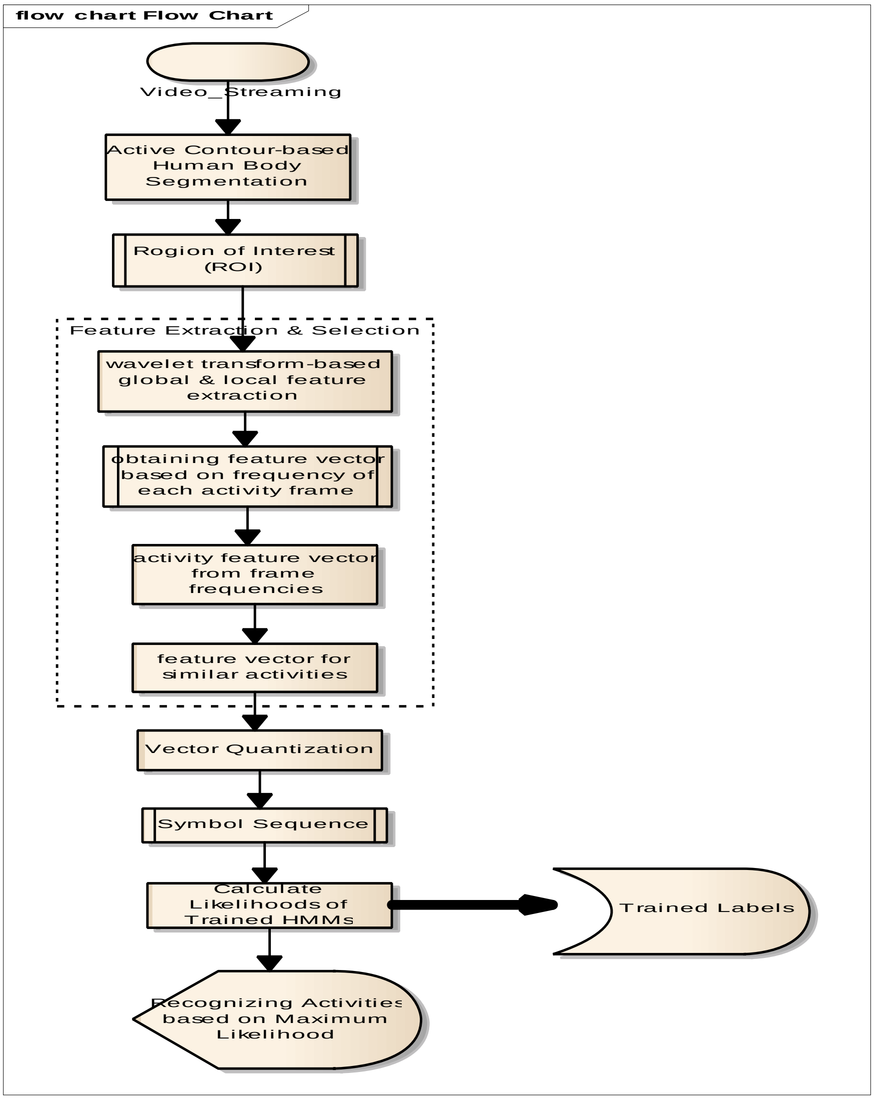


Figure 42 A video-based activity recognition system

5.1.1.3. Multimodal Sensor Fusion

- **Introduction**
  - Now-a-days, sensor technology is the more robust, cost-effective, less-intrusive and easy to install solution for recognizing the human activities [Turaga2008].
  - Single sensor is not adequate to reflect the true context of humans, so multimodal sensors are required such as embedded sensors, wearable sensors and location sensors to recognize the true context.
  - Fusion engine is an effective solution to combine the information from these heterogeneous modalities [Dante2010].

- **Problems in Existing Systems**
  - Many researchers have devised numerous methods that can be divided into three categories: rule-based methods, classification-based methods, and estimation-based methods [Pradeep2010].
  - Classification-based methods include a range of classification techniques that have been used to classify the multimodal observation into one of the pre-defined classes [Pradeep2010]. These techniques are inadequate in context recognition due to lack of support to handle mutually exclusive hypothesis and become hard to handle large number of combinations of hypothesis.
  - The estimation-based fusion methods such as Kalman filter and particle filter are generally used to estimate and predict the fused observations over a period. These methods are suitable for object localization and tracking tasks [Kulkarni2011].
  - According to the nature of the problem, we explore the Rule-based methods and proposed our novel design to fulfill the requirement of healthcare domain that demands high accuracy.
  - The existing rule based methods such as majority voting and linear weighted fusion are the simplest techniques to combine the output of different modalities. It is computationally less expensive approaches but major issues are optimal weight assignment, less specific to the problem domain, and less flexible to add new rules [Pradeep2010].
  - New methods are required, which are capable of fully automated adjustment and self-adaptation to recognize the true context. Our fusion engine will automatically extracts the rules and utilize probabilistic theory to increase the level of accuracy.

- **Requirements**
  - To overcome the limitations of the existing systems and provide healthcare and smart homes services, there are different requirements that need to be satisfied.
    - Information coming from diverse sources intelligently fuse together like motion sensor, accelerometer, audio/video sensor, and location sensor.
    - To deal with different sources some synchronization process is required.
    - For the understandable output some auto-generated rules mechanism is to be needed to fulfill the requirements.

- **Proposed Solution**
  - Our proposed solution is based on evolutionary technique with probabilistic approach. It will be capable to fuse audio/video, accelerometer, location, and embedded sensors data

to improve the robustness and accuracy of the recognized context.

- Fusion engine is comprised on preprocessing step and three layers architecture to process the sensor modalities. The detail diagram of the fusion engine is illustrated in Figure 43.
- Let $\Omega = \{S_1, \ldots, S_n\}$ be a set of n sensors, e.g., {wearable, location, camera, embedded etc.} characterized by m attributes $A = [a_1, \ldots, a_m]^T \in \Re$, which include binary or numerical value, to express the individual activity recognition for specific instance of time.
- Let C classes $C_1, \ldots, C_n$ has the objective to recognize the human activities automatically, so rule set (RS) for base learner is:

$$\mathrm{RS}\colon\ \Re^n \longrightarrow [C_1, \ldots, C_n] \qquad (1)$$

i.e., the output RS is n-dimensional rule vector, whose i^{th} component denotes the "rule" for recognizing the activity, which comes from sensor S_i by evolving genetic algorithm. The combination of the outputs of individual base learners "BL" is:

$$\mathrm{DF} = \max\big(\mathrm{RS}_1(\mathrm{BL}) + \ldots + \mathrm{RS}_n(\mathrm{BL})\big) \qquad (2)$$

i.e., DF is sensor fusion with the highest supporting class, determined by majority voting, accuracy matrix calculated by maximum likelihood probabilistic technique and RS(BL) is an aggregation operator to combine the each base learner rule set. The architecture is as follows:

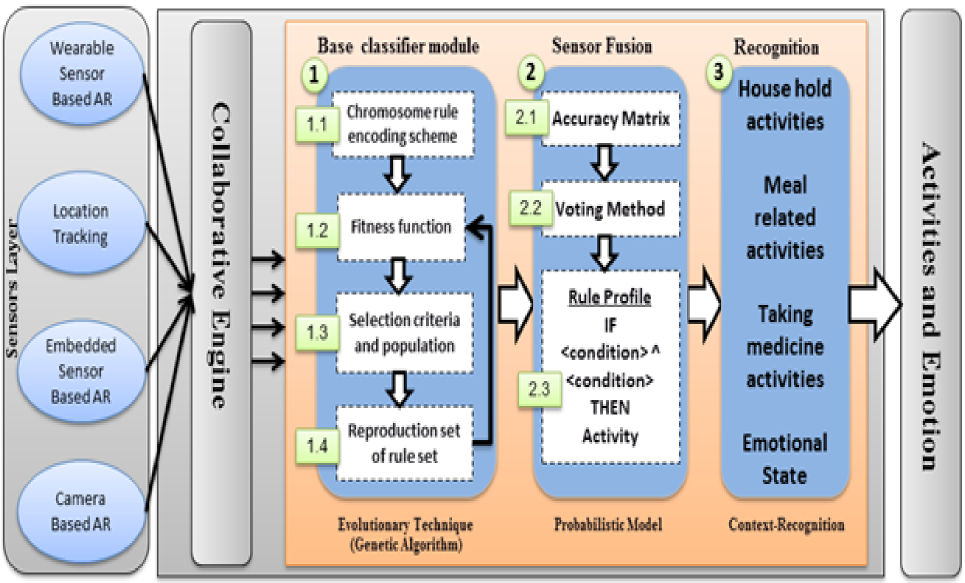


Figure 43 Multimodal sensor fusion architecture

- **Base classifier module**
  - Base classifier is based on genetic algorithm. In this phase, candidate rule set solutions are randomly generated. To alive these candidate solutions, it needs to survive, otherwise it is dominant by the good candidates [Jiao2006]. Initial population is randomly generated and then improves the quality of the candidate rules by evaluating the training instances. Following are the steps to construct the base learner compo-

nent.

- **Chromosome rule encoding scheme**
  - The rule set is the representation of sensory data; more commonly known as a chromosome encoding structure. In our case, we adopt the Michigan approach [Rogova2004] to encode the sensor data. In which each individual encodes a single rule. Each rule contains a logical combination of sensor value in the form of$\mathrm{attribute_1} \cap \mathrm{attribute_2} \cap \mathrm{attribute_3}, \ldots, \mathrm{attribute_n}$.

- **Fitness function**
  - Evolutionary algorithm is guided search space by the fitness function; it must include the quality measures that reflect the accuracy of the candidate solutions [Rogova2004]. Fitness function is defined as:

$$\mathcal{F} = \sum_{i}^{n}\sum_{j}^{m} reward\ (Candiate_i\ |example_j) - \sum_{i}^{n}\sum_{j}^{m} payoff\ (Candiate_i\ |example_j)$$

$$= \sum_{i}^{n}\sum_{j}^{m} [reward\ (Candiate_i\ |example_j) - payoff\ (Candiate_i\ |example_j)]$$

Where, $reward = \begin{cases} 1 & if\ candidate \equiv example\ \cap classLabel \equiv Correct, \\ 0 & otherewise. \end{cases}$

$$payoff = \begin{cases} 1 & if\ candidate \equiv example\ \cap classLabel \equiv Incorrect, \\ 0 & otherewise. \end{cases}$$

  - The fitness function “F” evaluates the individual rule in the rule set. It assesses how many training instances are correctly classified by paying reward and also payoff in case of wrong class identification.

- **Parent Selection**
  - Once these rules are evaluated according to the above fitness criteria then they are ranked. The best fittest rules are passed to next generation as elite.
  - From the remaining solution, some rules are selected based on certain probability x of whole population for crossover operation. This mechanism prevent from losing the best rules which are found in each generation and increase the probability of selection for weak candidates.
  - Fitness ranking and elitism ensures that rules are converging to the global optimum point without stuck into local optima.

- **Reproduction set of rule set**
  - It applies the stochastic operator crossover, mutation and elitism to generate the new population to search the global optimum.

- **Sensor fusion**
- This module is based on the following three components.

- **Accuracy Matrix**
  - This module will again process the rule to maximize the accuracy and assign the weights to each rule according to the importance of context.

Feature Function: f: X * Y → R

$$p(X|Y,w) = \frac{\exp\left[\sum_{j=1}^{n} w_j F_j(x,y)\right]}{z\,(x_i,w)}$$

$$F_j = \sum_{j=1}^{n} w(i, R_x^i, C)$$

$$z\,(x_i,w) = \sum_{y\in y'} \exp[w_j F_j(x,y')]$$

Taking log on both sides

$$\mathbf{log}\, p(X|Y,w) = \mathbf{log}\, \frac{\exp\left[\sum_{j=1}^{n} w_j F_j(x,y)\right]}{z\,(x_i,w)}$$

$$\mathbf{log}\, p(X|Y,w) = \sum_{j=1}^{n} w_j F_j(x,y) - \sum_{y\in y'} \exp[w_j F_j(x,y')]$$

Maximize the likelihood of data and calculating the gradient with respect to "w"

$$\mathrm{W}_{ML} = \mathrm{argmax}_{\mathbf{W}\in\mathbb{R}^m} L(\mathrm{W})$$

$$L(\mathbf{W}) = \sum_{i=1}^{n} \log P(y_i \mid x_i)$$

$$= \sum_{j=1}^{n} w_j F_j(x,y) - \sum_{y\in y'} \exp[w_j F_j(x,y')]$$

$$\left.\frac{dL}{d\mathbf{W}}\right|_{\mathbf{W}} = \sum_{j=1}^{n} F_j(x,y) - \sum_{i=1}^{n} \sum_{y\in y'} \exp[w_j F_j(x,y')]$$

- It is a probabilistic and discriminative model. It has a number of features and each feature has a weight for correct classification. Feature weights are optimized on development data, so that the system output matches correct translations as close as possible and minimize the error rate.
- The goal is to maximize the likelihood of the data and consequently results in good feature weights.

- **Majority voting**
  - It is most common method for combining the output of multiple learners [Rogova2004]. Let assume we have "L" base learners and each have "n" rules based on sensors value to recognize the activities. Then we can define majority voting for the base learners as depicted in equation 3.

    $$\sum_{i=1}^{L} \mathrm{DF} = \max_{j=1}^{C}\left(\sum_{i=1}^{n} \mathrm{RS}_n(\mathrm{BL}_n)\right) \quad (3)$$
  - In Equation.3 "DF" is the decision fusion vector which determines the most confident class label for any instance.

- **Rule profiling**
  - Each rule contains a logical combination of sensor value in the form $\mathrm{attribute}_1 \cap \mathrm{attribute}_2 \cap \mathrm{attribute}_3, \ldots, \mathrm{attribute}_n$ is followed by a class label at a specific instance of time.

- **Recognition**
  - The final step of fusion engine is to recognize the context of real world. So different sort of house hold activities, exercise, meal preparation and taking medicine context will be recognized with confidence values.

- **Overall System Flowchart**
  - Training and testing module flow chart provides the internal components interaction and flow of the process.
  - At the end of the process we get the rule profile along accuracy matrix and component is ready to classify the situation.

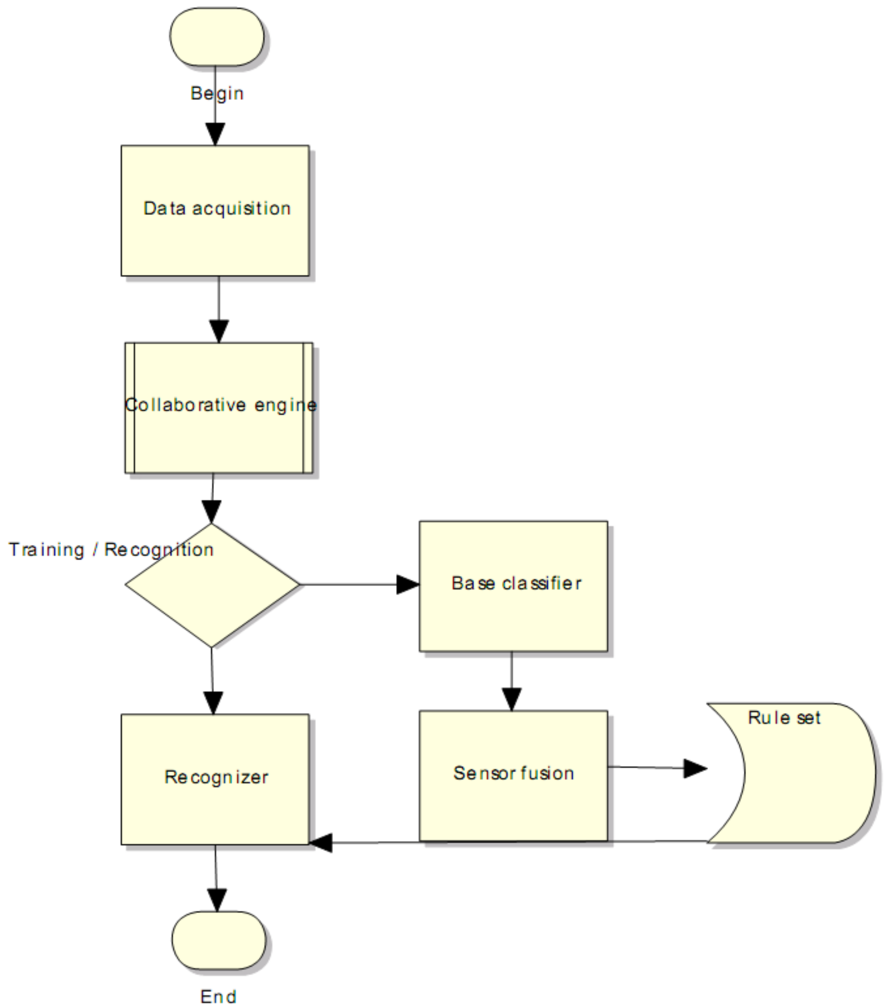


Figure 44 Complete flow chart of Multimodal Sensor Fusion

5.1.1.4. Emotion Recognition

- Emotions are a universal means of communication which can be expressed non-verbally without any language constraints. They are recognized through facial expressions, voice tones, speech and physiological signals.
- Communication through facial expressions plays a significant role in social interactions.
- Over the last decade, automatic facial expressions recognition (AFER) has become an important research area for many applications such as child development, neuroscience and psychology, access control and surveillance and human behavior understanding.

- **Introduction**
  - Anytime and anywhere, people have the ability to sense and express emotion, which can help make decisions, handle crises, and maintain relationships. Plentiful words and phrases are used to express moods and feelings. The ability of sensing and expressing emotion in a machine is often a luxury, considered unnecessary and functionless in basic computer intelligence. Programmers hardly encode emotion descriptions into computers. Why are people still trying to give computer emotional ability? [Malika2009].
  - When a human user interacts with a computer agent, emotion expressed by the agent affects the emotional response from user. In a study of game agent, if the virtual game agent opponent behaves “naturally" in that it follows its own goals and expresses associated positively or negatively affect behaviors, users will be less stressed than when the agent does not do so [Prendinger2006]. A frustrating interaction with a computer system can also leave a user feeling negative toward the system and its maker [Klein2002], so the consideration of emotion for computers and agents becomes important.
  - Before giving computer emotion ability, recognizing human emotion is a preliminary requirement. A computer agent needs to know how a person feels via emotion recognitions, and then the computer agent can perform a reasonable emotional feedback to the person. This feedback of computer is a design of human computer interaction. In the design of human computer interaction human desire and feeling should be considered. Furthermore, it needs to obtain the emotion via contain electronic media before it recognizes such emotion [Malika2009].
  - The study of facial expression recognition (FER) has consistently been an active and exigent research area in recent years, which has a significant contribution in many applications such as communication, personality and child development [Bartlett1999], neuroscience and psychology [Mehrabian1968], access control and surveillance [Bettadapura2009] and human behavior understanding. There are three basic modules in FER systems, preprocessing, feature extraction and recognition as shown in Figure 45.

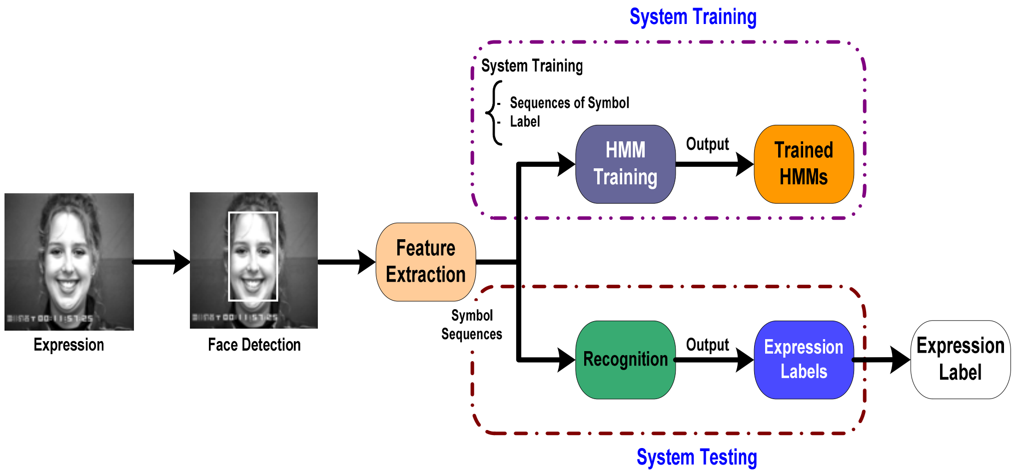


Figure 45 General architecture of video-based emotion recognition system.

- **Problems in Existing Systems**
  - Several video-based emotion recognition systems have been proposed; however, recognizing human facial expressions accurately is still a major concern for most of these systems. This lack of accuracy could be attributed to various causes, such as different light conditions for training and test images, lack of accurate automatic face detection methods, and high similarity among different facial expressions that occurs in the presence of low between-class variance in the feature space.

  - **Feature Extraction.** The accuracy of recognition module is completely reliant on the extraction of good features that means, the effectiveness of the extracted features will also affect the accuracy of recognition module [Huang1997]. Moreover, a good classifier is required to classify the expressions accurately.
  - Many of the existing works have been investigated new algorithms to extract good features. The authors of [Wu2010] have proposed a complete approach for facial expression recognition. They employed canny edge detection and a well-known statistical approach such as Principal Component Analysis (PCA) for features extraction. However, canny edge detection computational wise very much expensive, and some time it is not particularly successful technique because mostly the edges are not continuous and serious edges might be presented due to noise, which might a complex task for canny edge detection. On the other hand, PCA yields uncorrelated components. If the data have a Gaussian distribution, the uncorrelated components are independent. However, if the data are merged of non-Gaussian components, then PCA fails to extract components having non-Gaussian distribution [Buciu2009].
  - Most of the significant facial expressions information can be extracted from lips, nose and eyes known as local features that have an important role in achieving the best accuracy of recognition. Therefore, to extract the local features, one of the higher order feature extraction techniques such as Independent Component Analysis (ICA) [Bartlett2002, Chuang2006, Buciu2003] has been extensively employed for FER systems. The peak state of the facial expression was proposed to recognize the facial expression images;

however, the localized features were ignored [Kim2009]. Furthermore, ICA is very weak in managing the inputs. If there are plenty of video frames, exploited as input, ICA does not have the capability to organize it, due to which some time ICA cannot retrieve the desire features.

- Another method like Local Binary Patterns (LBP) has been exploited by [Shan2009] for feature extraction. In this method a 3 x 3 operator is used in which each pixel is compared with its eight neighbors by subtracting the center pixel value. The resultant positive values are encoded to 1 otherwise encoded to 0. However, the dominant features cannot be extracted by this small 3 x 3 LBA operator. Also it does not provide directional information of the facial frame because it only captures the relations with its surrounding eight neighbor pixels. Moreover, LBP uses first order local patterns and cannot extract more detailed information. Therefore, to solve this problem and to extract the best local features, high order local pattern descriptor such as Local Directional Pattern (LDP) has been presented by [Huang2011] and [Jabid2010]. By using LDP, we can extract best local features that will further be used for recognition. However, computational wise, LDP is much expensive. Moreover, lots of memory size is required for LDP to store the binary codes that is four times bigger than LBP [Jabeed2010].

- **Recognition.** A large number of classifiers have been employed for video-based facial expression recognition. Among them, the HMM is still the most commonly used method, and it produces accuracies that are comparable to other well-known classifiers such as GMMs, ANNs, and SVMs. In addition, HMMs have their own advantage of handling sequential data when frame-level features are used. In such cases, other vector-based classifiers like GMMs, ANNs, and SVMs, fail to learn the sequence of the feature vectors. However, it is still very difficult for HMMs to represent multiple interacting facial expressions.
- A recent work by Zia et al. [Uddin2009] proposed a complete approach for FER systems that achieved high classification accuracy for the Cohn-Kanade dataset of facial expressions. In their work, they employed well-known statistical methods such as principal component analysis (PCA) and independent component analysis (ICA) for global and local feature extraction, respectively. Furthermore, they used a single-LDA- and a single-HMM-based recognition technique for expression recognition. However, their technique failed to exhibit the same high accuracy when used for other databases, such as the Cohn-Kanade dataset (87%), Japanese (JAFFE) dataset 83%), and AT&T dataset (72%) of facial expressions. Low accuracy in these new experiments could be attributed to the following three reasons. First, most of the expressions share high similarity and thus overlap significantly in the feature space, as shown in Figure 46. Zia et al [Uddin2009] applied a single LDA with the assumption that this variance is distributed uniformly among all the classes in the feature space. However, this is not the case. For example, expressions like happy and sad are very similar to each other but can easily be distinguished from anger and fear (another pair with high similarity).

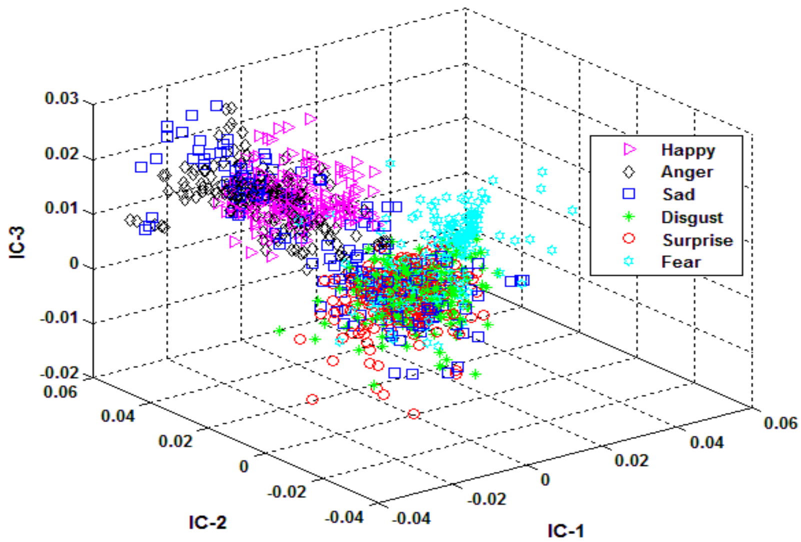


Figure 46 3D-feature plots for six different types of facial expressions. The features are highly merged, due to the presence of similarity between the expressions, which could later results in a high misclassification rate

- **Requirements**
  - To overcome the limitations of the existing systems and to provide better decision on facial expression recognition in healthcare domain, there are different requirements that need to be satisfied.
  - Information coming from video (2D) camera.
  - To store the information (i.e., Facial expressions) coming from input source, we need to decompose it into frames and then store them for decision making.
  - For the system to work properly, smooth communication among all the components is required.
  - The need is to develop an automatic face detection and extraction technique.
  - For knowing the expression, a recognition engine is required that will facilitate the doctors to find the aggressiveness of patients in healthcare environments.

- **Proposed Solution**
  - **Preprocessing.** Accordingly, the purpose of this study was to propose an accurate and robust facial expression recognition system capable of exhibiting high recognition accuracy. The proposed system is based on the idea of a hierarchical recognition approach, and it was tested using three publicly available datasets. Each dataset consisted of six facial expressions: happy, sad, surprise, disgust, anger, and fear. Each expression was performed by different people. Mostly, the image data in these datasets contained frontal views of the face, and each expression was composed of several sequences of expression frames.
  - In these datasets, facial expressions were recorded under different light conditions, which could reduce the recognition accuracy. To solve this problem, a method based on global histogram equalization was proposed. This method reduces such noise by increasing the dynamic range of the intensity using the histogram of the whole image. The proposed method finds the running sum of the histogram values and then normalizes it by dividing

it by the total number of pixels. This value is then multiplied by the maximum gray level value and then mapped onto the previous values in a one-to-one correspondence. The proposed method outperforms previously used techniques, such as histogram equalization (HE) and local histogram equalization (LHE) that produce unwanted artifacts and washed-out images. Furthermore, LHE causes over-enhancement and sometimes produces checkerboards of the enhanced image; therefore, several studies suggested avoiding the use of these methods.

- **Face Detection.** As stated earlier, an accurate facial expression recognition system requires automatic face detection. Thus we proposed, a novel technique that automatically detects and extracts faces from a given image using two methods: gray-level- and skin-tone-based methods.

- **Feature Extraction & Selection.** We employed the well-known statistical approaches, PCA and ICA, to extract local and global features, respectively. Since most expressions share high similarity, their features overlap significantly in the feature space. This can result in the presence of very low between-class and high within-class variances in the feature space, which in turn can lead to low recognition accuracy. Numerous methods have been presented in the machine learning literature to solve this problem, such as LDA. However, our experiments showed that applying LDA directly to the whole feature space failed to resolve the overlap or low between-class variance among facial expressions. This failure could be attributed to the fact that LDA is a linear technique, which limits its flexibility when applied to complex datasets. Moreover, the assumption made in using LDA that all classes share the same within-class covariance matrix is not valid in this case.

- **Hierarchical Scheme-based Recognition** To overcome this problem, we proposed a hierarchical recognition scheme that is a combination of PCA, ICA, LDA, and HMMs. Based on the parts of the face that create an expression, expressions were divided into three categories: lips-based, lips-eyes-based; or lips-eyes-forehead-based expressions. At the first level of the proposed hierarchical recognition scheme, LDA was applied to the features (PCs and ICs), and the resulting LDA features were fed to a single HMM to recognize the category for the given expression (as shown in Figure 47). Once the category of the given expression was determined, the label for the expression within the recognized category was recognized at the second level by again feeding the features to a combination of LDA and HMM, trained specifically for the recognized state as shown in Figure 47.

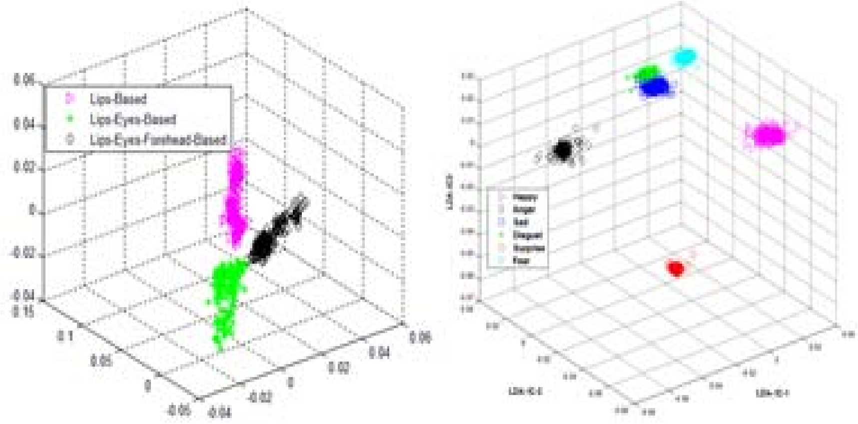


Figure 47 (a) 3D feature plots for the three expression-categories after applying LDA at the first level. (b) 3D-feature plot of the proposed hierarchical recognition scheme for six different types of facial expressions after hierarchical LDA.

- **Overall System Flowchart**
  - The overall flowchart for the proposed system at the training stage is shown in Figure 48.

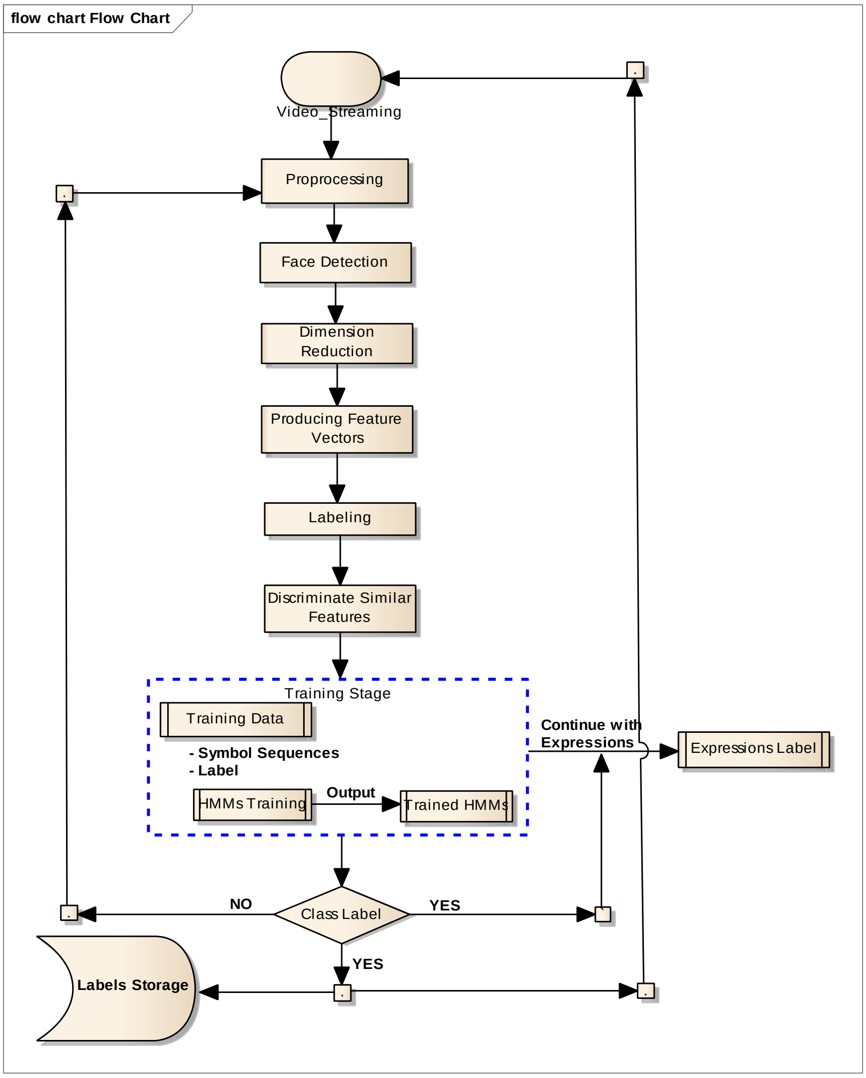


Figure 48 A video-based emotion recognition system

5.1.2. Context-aware Activity Manipulation Engine

- Semantics is the meaning of a resource in the context in which it is referred while these meanings are provided by using modeling languages like rdf(s) and OWL.
- Modeling information in a formally structured and canonical representational format is called as ontology. So ontology is basically the representation of information combined together using some modeling language.
- LDSS for Wellness takes the input form CAME as well.
- Context-aware Activity Manipulation Engine (CAME) is one of the main components of SC3 and also an inessential component of LDSS for Wellness.
- CAME is the process of inferring high level activities from low level activities recognized by different sensors.

- **Introduction**
  - Use of ontology provides basic description to events and activities, and from this information we can deduce new knowledge.
  - Use of ontology in activity recognition is relatively a new area of research. Using ontology help us better understand the activity in a given context.
  - Activities recognized with the help of different sensors (i.e., body, location, motion, and video sensors) are low level activities and they are not in a capacity to be used for certain types of analysis and decision making.
  - With the help of ontology, where we use the context information and link all the related activities in a chain, then with the help of customized rules we get the higher level activities that are more usable for decision making.
  - Ontology helps in properly extracting the higher level activity of a set of activities in a series, e.g., series of low level activities like bending, sitting, jumping and walking with the help of ontology will result in a higher level activity i.e., exercising.

- **Problems in Existing Systems**
  - In today's society, people are interested in their improved healthcare. Hence, a challenge exists to both provide and maintain the quality and availability of healthcare services with a minimum cost [Khattak2010].
  - Various wireless technologies have been used for improved healthcare services. CodeBlue [Gao2008] supports physicians and nurses to monitor patients.
  - Reminder based systems which have primarily focused on plan-based approaches to decide how and when to prompt subjects effectively [Mama2000] in addition to offering location-based reminders [Sohn2005]. Nevertheless, the context for the delivery of reminders involves more information than simply location or time.
  - HYCARE [Du2008] system takes context in consideration and develops a schedule for reminder based services with a goal of avoiding possible conflicts.
  - In [Wang2009], an ontology is used to incorporate context related information for intelligent processing; however, the system developed was more of a homecare system rather than a healthcare system.
  - Work presented in [Chen2012] details an ontology based reminder system; incorporates rules for manipulating the recognized activities of the elderly.
  - In [Khattak2011], real-time activities are recognized using diverse sensors are used for situation analysis.
  - Research in [Fitma2011] focused on the social interaction of patients. Based on the expe-

riences shared, it generates intelligent service recommendations and eliminates their isolation.
- Existing systems aiming to offer a form of support are mostly based on one input modality and in some cases use imperfect contextual information [Henr2004]. These solutions subsequently do not consider the integration of activity from diverse input modalities.

- **Requirements**
  - To overcome the limitations of the existing systems and provide healthcare and smart homes services, there are different requirements that need to be satisfied.
    - Information coming from diverse sources likes motion sensor, accelerometer, video sensor, and location sensor.
    - To store the information coming from input modalities, we need to store them. For the storage structure, a storage model is designed for the knowledgebase. OWL is used for the storage.
    - For the system to work properly, smooth communication among all the components is required.
    - The need is to design and model the rules for developing the filtering system.
    - For Context Analyzer, a reasoning engine is required that will facilitate the match making process.

- **Proposed Solution**
  - CAME is one of the main components of SC^3. SC^3-CAME is the process of inferring high level activities from low level activities recognized by different sensors.
  - The component based framework architecture diagram of CAME and the information flow is given in Figure 49, while the detail description of all the components are given in their corresponding sections.

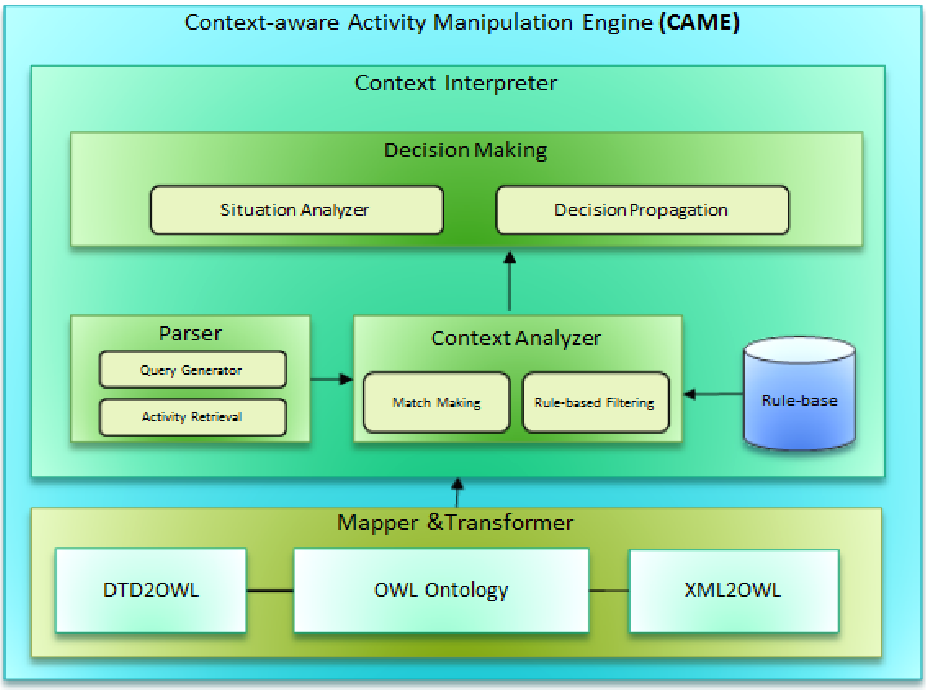


Figure 49 Overall architecture of Context-aware Activity Manipulation Engine (CAME)

- **Mapper & Transformer**
  - Mapper & Transformer is responsible for recognizing data inputs from diverse sources. Video-based, sensor-based, motion-based, and location-based activity recognition engines will provide output in different formats like XML (example given below in Figure 50) and simple text.
  - So there is need of recognizer and convertor engine that will convert the simple representation of activity into OWL representation. The OWL (N3) representation of the activity is given in Figure 51.
  - The three subcomponents are: 1) DTD2OWL, 2) OWL Ontology, and 3) XML2OWL.

```
<?xml version="1.0" encoding="UTF-8"?>
<activities>
   <activity type="Motion">
            <detectedBy>Motion Sensor</detectedBy>
          <hasName>Prof. SY Lee</hasName>
            <activityName>Entering Class</activityName>
          <id>345</id>
          <time>2009:06:14:14:00:13</time>
   </activity>
   </activities>
```

Figure 50 XML representation of activity information produced by motion sensor.

```
activityOnto:Activity_Instance_20090614140013345
a           activityOnto:Activity ;
activityOnto:hasConsequentAction    activityOnto:Action_Instance_145413546;
activityOnto:hasID                               345;
activityOnto:hasName                      "Entering Class";
activityOnto:hasType                                 "Motion";
activityOnto:isA                          activityOnto:Room_Instance_Class;
activityOnto:performedAtTime               2009:06:14:14:00:13;
activityOnto:performedBy                       activityOnto:Person_Instance_345.
```

Figure 51 OWL representation (N3) of activity detected using motion sensor.

- **Knowledgebase**
  - Knowledgebase (KB) serves as the back bone of CAME.
  - It is responsible for proper communication of information among all the components of CAME.
  - It stores all the possible types of activities that a human body can perform in different context/situation, with the information of different activities priority for different users and group of users.
  - The proper engineering of the KB is most important activity in the development of CAME.
  - To engineer the KB (see Figure 52) we have to look at the same problem from different dimensions.
  - When an activity is recognized by the sensors then this knowledgebase is parsed and if required in different situations then inferencing is done for decisions against activities.
  - So it is actually the ontology, where all the activities are semantically modeled and available for analysis and decision making.

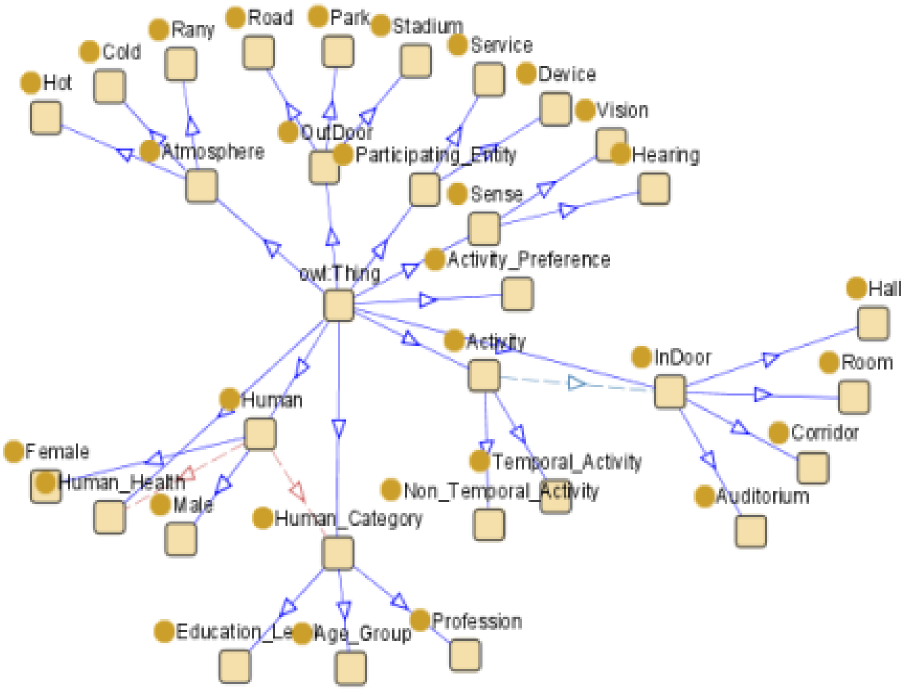


Figure 52 Knowledgebase (Human Activities Ontology)

- **Parser**
  - For any type of information manipulation from the Knowledgebase, Parser is responsible to properly handle all the operation regarding that matter.
  - The Parser normally communicates with Activity Representation component to properly represent the activity, it also parse the Knowledgebase for the Context Analyzer for verity of different reasons like verification of activity and Match Making.
  - To populate the KB for newly recognized activity, the Parser is also used in that case.

- **Context Analyzer**
  - To understanding the context of an activity and to extract high level (abstract) activities from low level activities recognized by sensors, we need to have a Context Analyzer for analysis of these activities and to make proper situation analysis.
  - The Context Analyzer is very important component of CAME. It uses the activities information with respect to their context information and infers high level activities. For this purpose Match Making is activated.
  - The decisions or suggestions of Context Analyzer are very much dependent on the domain and user intensions.
  - So for this reason we also introduced the user defined customized rules in the Context Analyzer. So the rules are used in Rule-Based Filtering to filter out the unintended activities information.

- **Rule Base**
  - Every organizations have their own customized rules. These are used to implement the organization's policies.
  - So for these sorts of situations and actions, we need to define customized rules for different activities.
  - The rules are shown in Figure 53.

Rule1

$\exists \text{Activity}(a1) \sqcap \neg \text{hasContents}(\text{taking medicine}) \sqcap \text{hasNextActivity}(a2) \sqcap \exists \text{Activity}(a2) \sqcap \text{hasContents}(\text{eating}) \rightarrow \text{Activity.Create}(a1) \sqcap \text{Activity.Create}(a2) \sqcap \text{reminder}(\text{take medicine})$

Rule2

$\exists \text{Activity}(a1) \sqcap \text{hasContents}(\text{reading}) \sqcap \text{hasNextActivity}(a2) \sqcap \exists \text{Activity}(a2) \sqcap \text{hasContents}(\text{TV On}) \rightarrow \text{Activity.Create}(a1) \sqcap \text{Activity.Create}(a2) \sqcap \text{turnOff}(\text{TV})$

Rule3

$\exists \text{Activity}(a1) \sqcap \text{hasContents}(\text{unknown activity}) \sqcap \text{hasNextActivity}(\text{null}) \rightarrow \text{Activity.Create}(a1) \sqcap \text{reminder}(\text{movements are wrong})$

Rule4

$\exists \text{Activity}(a1) \sqcap \text{hasContents}(\text{entering kitchen}) \sqcup \exists \text{Activity}(a2) \sqcap \text{hasContents}(\text{entering bedroom}) \rightarrow \text{Activity.Create}(a1) \sqcup \text{Activity.Create}(a2) \sqcap \text{turnOn}(\text{lights})$

Figure 53 Customized rules for rule-based filtering

- **Decision Making**
  - After the process of context analysis, the system can take decisions or give suggestion against different activities.
  - So this module is responsible for performing some actions against the suggestions made by the Context Analyzer.
  - This module has two sub modules: 1) Situation Analyzer that analyzes the situation and stores the high level activities in Life Log for later usage. 2) Decision Propagation module instruct different hardware or connected systems for appropriate actions to be take based on the analysis.

- **Overall System Flowchart**
  - The overall CAME system flow chart with all components working is given in Figure 54.
  - This shows that how different components of CAME work together and how the output of one component flows to the other component where it is used as input for further processing.

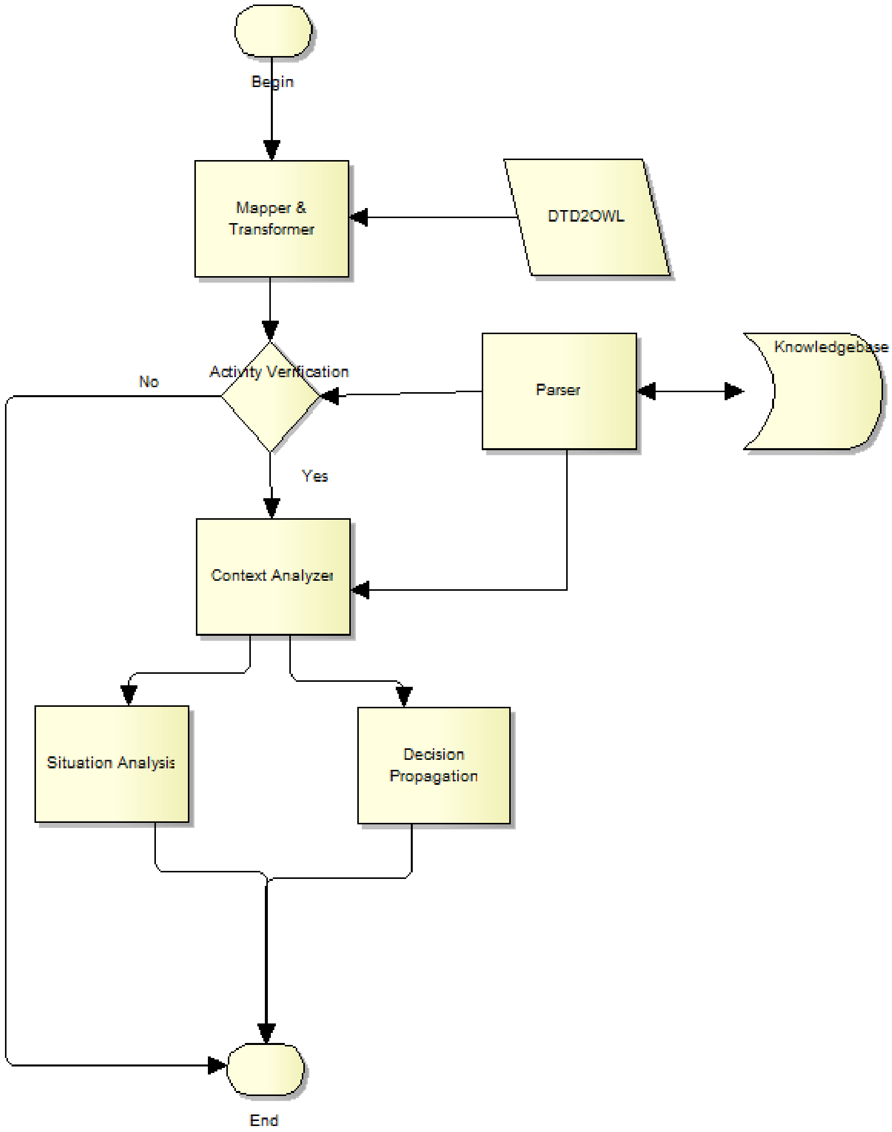


Figure 54 Flow Chart for overall working

5.1.3. Social Media Interaction

- In this module, we aim to improve the users' health by utilizing his social interaction in order to suggest them appropriate lifestyle patterns. For instance, after observing a user's daily routines, our proposed Social Media Interaction Engine (SMIE) is able to finds some complications with his lifestyle. He/she usually sleeps late; does not exercise regularly; does not take eats on time; eats too much. Obviously, these lifestyles are not good for healthy living. The proposed SMIE is integrated through life log in the behavioral analysis module which is under development at UC Lab. The lifelog will take the information and integrate it with patient demographic to facilitate the behavioral analysis and suggest changes in unhealthy life patterns through better way.

- To achieve above goals, we design SMIE with several novel ideas. Firstly, tweet analysis extracts user interest, health conditions and sentiment from user tweets. Secondly, trajectory in terms of outdoor movement of the patient is tracked using GPS enabled location aware mobile devices, such as smart phones. Finally, email interaction analyzes the users' actions to identify significant behavior and life threatening complications in daily routines to gain knowledge about their habits and preferences. The detail of each component is described in the subsequent sections Tweet Analysis.

5.1.3.1. Tweet Analysis

- **Introduction**
  - By monitoring person's social activities, interest and emotions can be extracted, help to provide personalized services to person.
  - System monitors user stream from Twitter and process tweets to extracts user interest and sentiments from tweets.
  - Our proposed system integrates as a plug-in application, while extracting user related information including profile information, person interests and emotions.
  - Analyzing the interest, behavior and lifestyle of person provides assistance in better decision making and personalized services.

- **Problems in Existing Systems**
  - Twitter data must be collected to use it for research purpose. Different analysis tools are available to collect twitter public data. Archivist is one of the tool to collect tweets.
  - Grabeeter is another tool to get individual personal and public tweets and store them for future use. Twitter data is available for mining unstructured data and analyzing hidden patterns in it. J. Chen et al [Chen2010] introduced a system for URL recommendations on Twitter using data stream technique.
  - The system was based on content sources, topic interest models, and social voting to design URL recommender and compare different recommender techniques.
  - Fabian Abel et al. [Abel2011a, Abel2011b, Abel2011c] analyzed user modeling on Twitter for personalized news recommendation and enrich news with tweets to improve the semantic of Twitter activities. The work used methods including topic, entity, and hashtag based to analyze the user modeling.
  - It also focused on temporal pattern extraction in user profile. Ilknur Celik et al. [Celik2011] studied semantic relationship between entities in Twitter to provide a medium

where users can easily access relevant content for what they are interested in.

- Eleanor Clark et al. [Clark2011] introduced a system to apply text normalization for Twitter. System categorized errors and irregular languages used in casual English of social media into different groups and then applied natural language processing techniques to correct common phonetic and slang mistakes in tweets.
- Tetsuya Nasukawa et al. [Nasukawa2003] used natural language processing techniques to identify sentiment related to particular subject in a document.
- They used Markov-modal based tagger for recognizing part of speech and then applied statistics based techniques to identify sentiments related to subject in speech.
- Jeonghee Yi et al. [Yi2003] presented a model to extract sentiments about particular subject rather than extracting sentiment of whole document collectively.
- This model proceeded by extracting topics, then sentiments and then mixture model to detect relation of topics with sentiments. Whereas Namrata Godbole et al. [Godbole2007] introduced a sentiment analysis system for news and blog entities.
- This system determined the public sentiment on each of the entities in posts and measured how this sentiment varies with time. They used synonyms and antonyms to find path between positive and negative polarity to increase the seed list.
- Andranik Tumasjan et al. [Tumasjan2010] analyzed Twitter as a source of predicting elections. They used the context of the German federal election to investigate whether Twitter is used as a forum for political deliberation. They used LIWC2007 [Liwc2012], a text analysis software developed to assess emotional, cognitive, and structural components of text samples using a psychometrically validated internal dictionary.
- They focused on 12 dimensions in order to profile political sentiment: Future orientation, past orientation, positive emotions, negative emotions, sadness, anxiety, anger, tentativeness, certainty, work, achievement, and money.
- Bernard J et al. [Jansen2009] performed analysis of Twitter as electronic word of mouth in the product marketing domain. They analyzed filtered tweets for frequency, range, timing and content.
- They found that 19% of a random sample of tweets contained mentions of a brand or product and that an automated classification was able to extract statistically significant differences of customer sentiment.
- Similarly Archivist [Archivist2012] is a service that uses the Twitter Search API to find and archive tweets. It helps to have a look at trends such as frequency of Tweets over time, top users and words and many more. It also helps user to get real time trend information on Twitter.
- Jeff Clark used Venn diagram to show illustrating pattern in twitter data [Clark2008]. By using Venn diagram he explores overlap of different topics with each other in tweets. Collecting tweets alone and analyzing them for sentiments just on the available keyword is not enough to understand the real semantics of tweets.
- There is a need to precisely parse and process the tweet for the contained knowledge.

▪ **Requirements**

- Semantically processing of Natural language from Twitter to extract user interest and sentiments to build user profile. This profile would be used to provide personalized services.
- Classification of user interest into different categorizes to enhance system efficacy in providing domain specific services.

▪ **Proposed Solution**

- The proposed system architecture consists of four main components. Data Manager, Knowledge Generator, architecture of whole system is shown in Figure 55.
- Knowledge Enhancer and Filter Engine. These components are elaborated as follows.

- **Data Manager**
  - Data Manager is our plugable application that interacts with Twitter. It consists of the following subcomponents.
  - **Data Fetcher**: Data Fetcher sends request to Twitter for stream of user. The fetched data is in JSON format [Archivist2012] that is a lightweight data-interchange format.
  - **Data Processor**: Fetched data requires some pre-processing before analyzing. Data Processor converts data in required useable format. It also removes user slangs from tweets.

- **Knowledge Generator**
  - Our purpose is to extract valuable information hidden in tweets and build user profile. Twitter data collected by our system is given to Alchemy API.
  - It accepts unstructured text, processes it using natural language processing and machine learning techniques, and returns keywords and sentiments of users about keywords. We extract participating keywords and sentiments associated with those keywords.
  - After extraction of knowledge, all tweets, participating keywords, and associated sentiments are stored in the repository for further processing as discussed below. However, the information extracted by knowledge generator is of low precision.
  - So, it needs further processing to better identify user interest and analyze sentiments of domain specific interest.

- **Knowledge Enhancer**
  - Knowledge enhancer module adds additional knowledge which was not extracted as keyword by Alchemy API.
  - The proposed system uses part of speech tagging and entity extraction on tweets and then adds additional data in the knowledge, extracted by Alchemy API.
  - Entity extraction by using Alchemy API helps by extracting entities, not extracted as keyword.
  - The proposed system have been tested with addition of subjects, verbs, objects, and entities in knowledge; however, just addition of verb and entities increases information collected from tweets.

- **Filter Engine**
  - For classifying tweets into different categories on the basis of knowledge extracted from tweets, the proposed system applies filtering on the extracted data. The filtering process is domain specific.

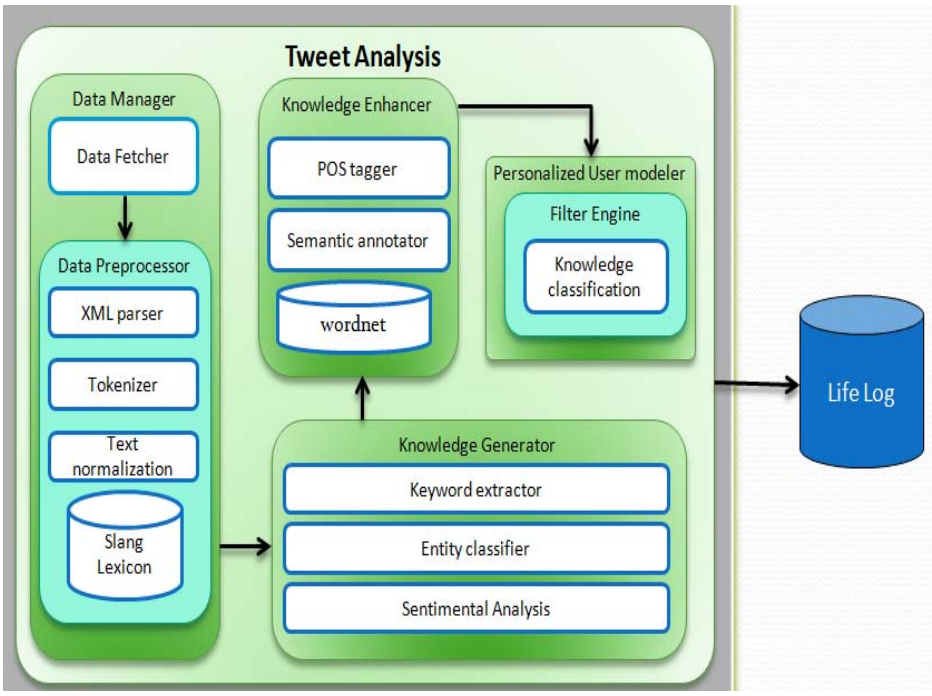


Figure 55 Overall Flow of Tweet Analysis

- **Overall System Flowchart**
  - The overall system flow chart with all components working is given in Figure 56.
  - This shows that how different components work together and how the output of one component flows to the other component where it is used as input for further processing

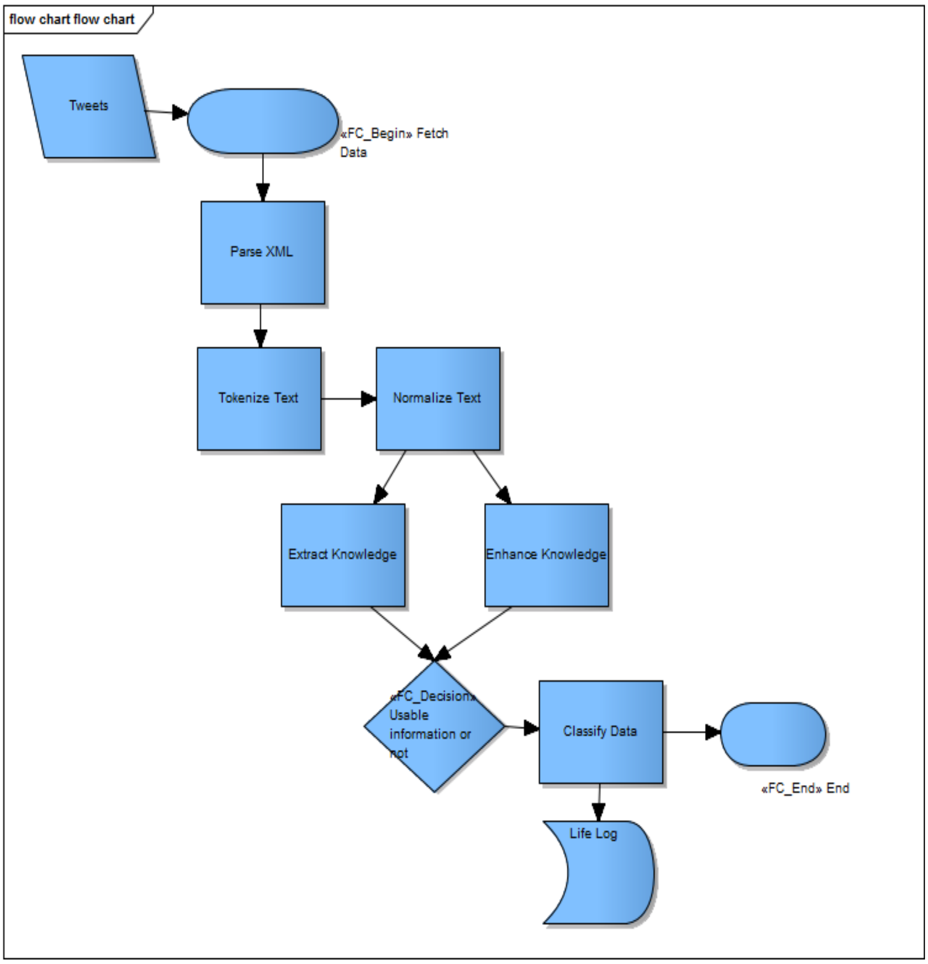


Figure 56 Flow chart of Tweet Analysis

5.1.3.2. Trajectory Analysis

- **Introduction**
  - GPS is a commendable technology to find location related activities. Presently a huge number of devices are enabled with this technology and getting more common with rapid speed. Currently more of these technologies are used for advancement of overall society and mankind. Like how to control traffic in a better way, finding peak and low rush hours and movement behavior of people of a particular area.
  - Our proposed system is incorporating trajectory mining techniques to a person's direct life and particularly its usage in healthcare domain is targeted. A GPS based real time activity monitoring system is developed which is used for tracking daily life routine of a user. Further these patterns are compared with the patterns prescribed by physicians and also recommendations for the better carrying out of prescribed schedule are presented to the user.

- **Problems in Existing Systems**
  - A recent study used trajectory information of people for finding people attractive areas and their related movement patterns, which can lead to instructive insight to transport management, urban planning and location-based services (LBS).
  - They considered taxi as an important mode of transport and acquired road traffic condition, travel patterns, average speed estimation and attractive places where people often visit [Yue2009].
  - In [Coll2011] the trajectory mining is used for mining ship spatial trajectory and an Automatically Identification System (AIS) is developed as a result of this study in which GPS enabled technology is used for finding the paths of ships [Coll2011].
  - The basic purpose of this study was self-navigation and collision avoidance but it can be extended for better marine traffic management and distribution. But both of these studies cannot be applied to human life because of the limitations and restriction involved like the most appropriate device for tracking and recording all of this information is mobile phone, which has very limited computation power and storage space.
  - Correspondingly all of these parameters are not required when we discuss about human life as we only are interested in daily routine activities and its positions so we have modified our approach accordingly.

- **Requirements**
  - GPS enabled smart phone with the availability of internet at each imperative location is required to fetch all the required information. Further processing of this data involves the Google API for conversion of GPS coordinates to Geo tags.

- **Proposed Solution**
  - The proposed method is a healthcare service which monitors user's routine activities and assists to follow prescribed schedule. Detailed architecture of the proposed system is shown in Figure 57. As shown in the figure that architecture is divided into 3 main processing module, Data Preprocessor, Schedule Manager and Activity Manager. Each of these modules is discussed below in detail.

  - **Data Preprocessor**
    - GPS coordinates of the user's position are fetched by using a GPS receiver, after a minimum time interval T_{min}. After that each of these coordinates are sent to Data preprocessor. The main module of data processor, imperative location finder confirms the significance of the position. Two main parameters of time and distance are used for conformance of imperative location. Position and corresponding activity is only treated as imperative if both of these thresholds are satisfied.
    - Then these coordinates of imperative location are sent to Geo tag transformer where Google API is used to convert it to Geo tags. Semantic tag of the imperative location is also fetched from the user for the acquiring contextual information about the location.

  - **Schedule Manager**
    - After preprocessing of data, all of the information is sent to the schedule manager for further processing. First of all semantic tag mappers plays its role of mapping all of the corresponding information and sent it to the repository for the storage. Followed schedule of the user is stored it followed patterns.

- Prescribed schedule of physician is stored in prescribed patterns and detail of each of activity in prescribed schedule is stored in the recommendations. Activity analysis is the main part of our proposed system with the purpose of comparing both followed and prescribed schedule. Inconsistencies of these schedules are shown to the user and physician as well for further improvement of the daily routine.

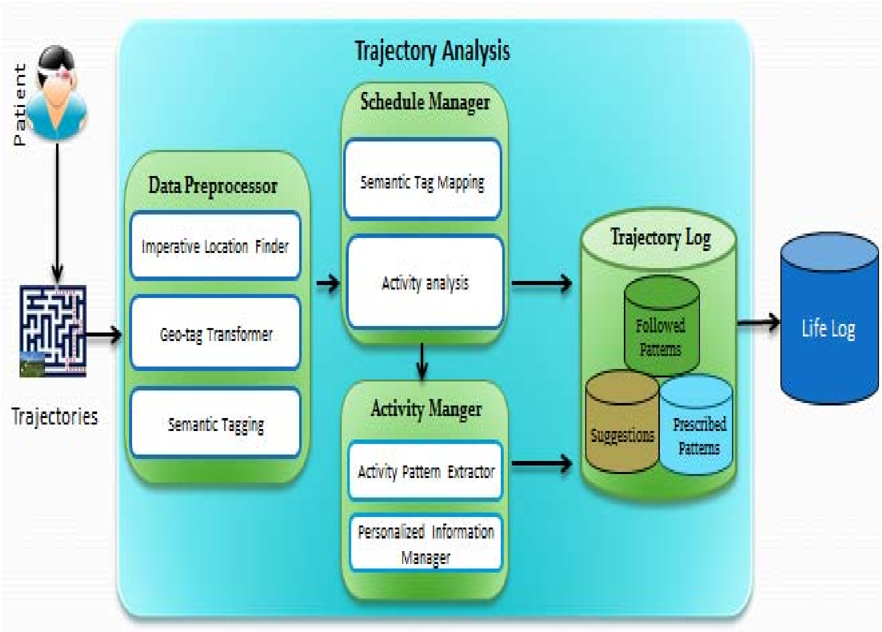


Figure 57 Main Architecture of Trajectory Analysis

- **Overall System Flowchart**
  - The overall Trajectory Analysis system flow chart with all components working is given in Figure 58 Overall System Flow for Trajectory Analysis. This shows that how different components of interaction analysis work together and how the output of one component flows to the other component where it is used as input for further processing.

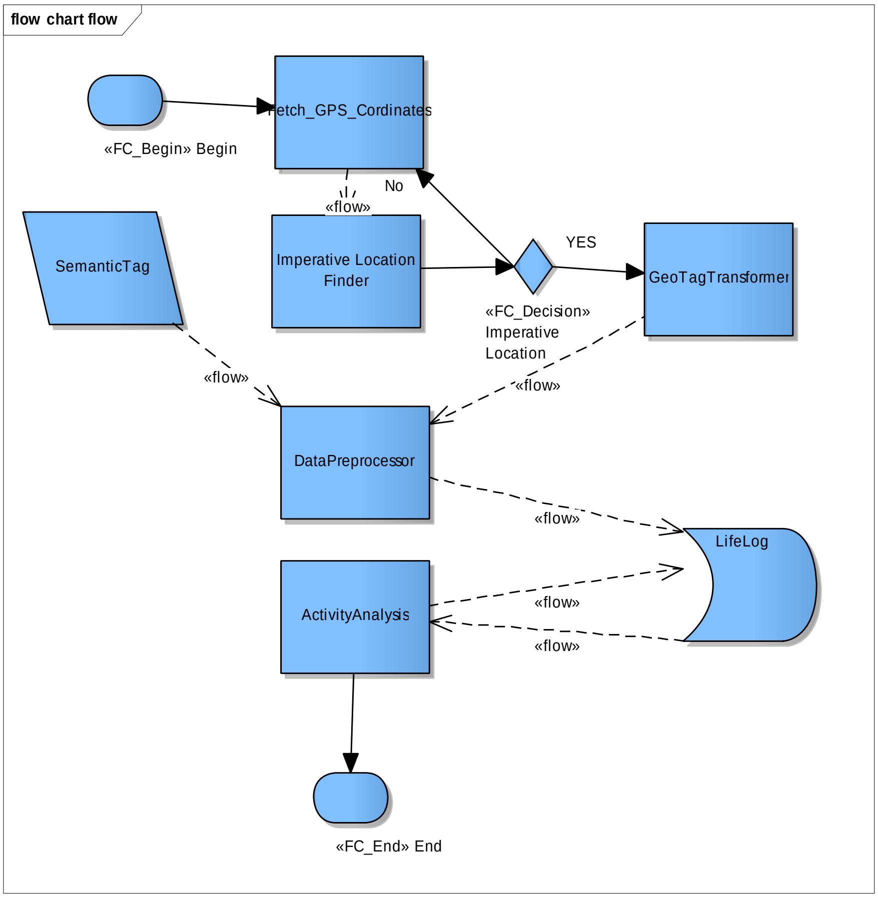


Figure 58 Overall System Flow for Trajectory Analysis

5.1.3.3. Interaction Analysis

- **Introduction**
  - In this module, we aim to improve the users' health by utilizing his social interaction in order to suggest them appropriate lifestyle patterns.
    - For instance, after observing a user's daily routines, our proposed interaction analysis is able to finds some complications with his lifestyle.
    - He/she usually sleeps late; does not exercise regularly; does not take eats on time. Obviously, these lifestyles are not good for healthy living.
  - The proposed method is integrated through lifelog in the behavioral analysis module which is under development at UC Lab.
  - The lifelog will take the information and integrate it with patient demographic to facilitate the behavioral analysis and suggest changes in unhealthy life patterns through better way.

- **Problems in Existing Systems**
  - The main motivation of our work is that time varying interaction data is collected in very diverse settings which need social network analysis to identify the meaningful information.
  - The network analysis has been used in a variety of fields to analyze the huge amount of data such as the Internet [Faloutsos1999], animal behavior [Fischhoff2007] [Sundaresan2007], e-mail habits [Chapanond2005, Diesner2005, Lahiri2010], mobile phone usage patterns [Nanavati2006], co-authorship patterns in research publications [Barabási2001, Newman2001], Dominance behavior and social association patterns of the animals [Juang2002].
  - In this module we analyzed the email interaction data with intensions to extract information that can facilitate the LDSS system in taking the more meaningful decisions.
  - We analyzed the recurring patterns correspond to seasonal and other recurrent association patterns. The similar work is done for analyzing the human behavior with location-aware cellphones [Eagle2006].
  - We applied it on email interaction of from daily living of users. We are interested to identify the typical periodicities and specific interaction patterns that may affect the patients' health

- **Requirements**
  - In this module we mine the patients' frequent and periodic interaction patterns that change over time.
  - The purpose is to gain knowledge about the preferences, needs and habits of the user.
  - Users can act in two different roles: senders and receivers. These two roles are not interchangeable while mining the patterns of interest from his daily interaction routine

- **Proposed Solution**
  - We propose a two-phase strategy to identify the hidden structures shared across different dimensions in dynamic networks such as type of interaction, time of interaction, interaction intervals and interaction response based on priorities.
  - We extract structural features from each dimension of the email network via periodic and frequent interaction mining, and then integrate them to find out robust patterns about patients as shown in Figure 59.
  - Furthermore, with the right formal definition of what constitutes periodic behavior, the aggregate periodicities of an entire set of mined interaction patterns can assist LDSS in better decision making.
  - Therefore, learning patients' common behaviors becomes an important step towards allowing LDSS to provide personalized services more accurately and effectively.

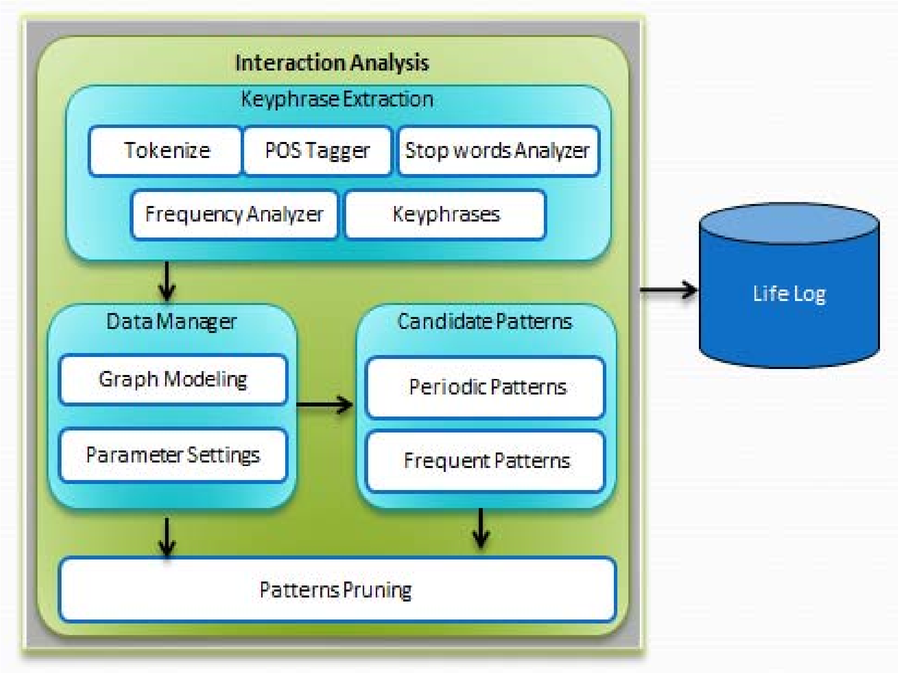


Figure 59 Overall Architecture for Interaction Analysis

- **Keyphrase Extraction**
  - The keyphrases are extracted from the contents of the email by using the Algorithm shown in Figure 60.
  - Extraction of the semantic keyphrases is the essential requirement of the accurate data modeling with the user interaction. First of all parameters of the extraction algorithm KEA++ [KEA] are set with respect to keyphrases' length in the taxonomy and length of the documents. Secondly train KEA++ on the set of Emails using taxonomy.
  - Then apply KEA++ on actual Emails (data). First email contents are tokenized by using POS tagger and stop words analyzer. The frequency of each word is counted in the email and then KEA++ return the relevant keypharses.
  - The keyphrases returned by KEA++ is processed to get its level label in the taxonomy. Identify level labels is required before applying the refinement rules because they represent the hierarchical order of the keyphrases.
  - If the KEA++ result has training level keyphrases then these training level keyphrases are retained in the result set as shown in steps 5 to 12 of Algorithm 1.
  - Lower level keyphrases are stemmed to their training level keyphrases and kept in the result set if they are associated with the general category at the lower level in taxonomy. Otherwise lower level keyphrases are discarded.
  - Upper level keyphrases are discarded after identifying and preserving their equivalent keyphrases from taxonomy which belong to the same level of training level keyphrases.
  - If the initial result does not contain any training level keyphrases then lower level

keyphrases of the result are preserved and added in the final refined result.

- Upper level keyphrases are discarded after identifying and preserving their equivalent keyphrases from taxonomy which belong to the same level of training level keyphrases.
- This process is executed from steps 13 to 21 of the algorithm. Finally redundant keyphrases are removed from the final refined set of keyphrases.

- **Data Manager:**
  - This module helps in data modeling and parameter settings before applying the mining algorithm.
  - It extracts a population of interest from messy email interaction data by removing noise.
  - The extracted information is modeled in graphs based on user defined interactions intervals and extracted keyphrases.
  - In each graph nodes are the individuals with keypharses as node label and directed edge represents the interaction between them.
  - Parameters set the thresholds of frequency and periodicity to identify the patterns of interest. For that, it is necessary to define a demanded minimum level (minimum confidence), so that all those sets of actions that have higher confidence level than the minimum confidence are considered as basic frequent periodic patterns.

```
Input: Training:
    (a) Set the parameters of KEA++ by keeping in view the keyphrase length in the taxonomy and email type
    (b) email along with their keyphrase and taxonomy
  Dataset for Extraction:
    (a) email with unknown keyphrases
Output: Set of refined keyphrases

1: TrainLevel ← KEA++ TrainLevel //(Rule I)
2: resultSet [] ← returned keyphrases by KEA++[]
3: resultSet [] ← level labels (Resultset [])
4: for resultSet[] <> empty do
5:     if (resultSet(training level)) then
6:         if (keyphrase level = lower level keyphrases) then
7:             processSet[] = preserving lower level keyphrases //Rule II
8:         else
9:             processSet[] ← identifying and preserving training level equivalent //Rule V
10:            processSet[] ← remove redundant keyphrases //Rule VI
11:            refineSet[] ← processSet[]
12:        end if
13:    else
14:        if (keyphrase level = training level) then
15:            refineSet[] ← processSet[]
16:        else
17:            if (keyphrase level = upper level) then
18:                processSet[] ← identifying and preserving training level equivalent keyphrases //Rule V
19:            else
20:                processSet[] ← stemming lower level general keyphrases //Rule II
21:            end if
22:            processSet[] ← remove redundant keyphrases //Rule VI
23:            refineSet[] ← processSet[]
24:        end if
25:    end if
26: end for
27: return refineSet[] //refine result set of semantic keyphrases.
```

Figure 60 Keyphrases Extraction from Email Contents

- **Candidate Patterns**
  - This module identifies a set of frequent and periodic patterns from email interaction graphs.
  - Frequent patterns emphasize the significance of patterns and periodic patterns consider the regularity between them. Frequents patterns are mined by using the FP tree based approach while periodic patterns are mined using PSEMiner with integration optimization.
  - The objective is to identify the sets of actions that frequently and periodically occur together.
  - Once basic frequent periodic patterns have been discovered, an aspect to consider is if there is any action which is not frequent taking into accounts all the periods, so that it is not discovered in the previous task, but it is frequent enough in those periods where the basic frequent periodic patterns occur.

- **Patterns Pruning**
  - This module applies one mining process to identify frequent and periodic patterns under the given parameter settings.
  - Patterns pruning reflects the common characteristics of a typical email interaction with some unusual association between patterns. For that, the starting point will be the candidate patterns are transformed into integrated set in order to make it useful comprehensively.
  - Briefly explained, it infers meaningful actions from the data collected by email data and then it splits the string of actions into periodic sequences based on some frequent support.
  - Combining these two concepts allows us to define periodic patterns in a way that avoids any redundant information.

▪ **Overall System Flowchart**
  - The overall CAME system flow chart with all components working is given in Figure 61. This shows that how different components of interaction analysis work together and how the output of one component flows to the other component where it is used as input for further processing.

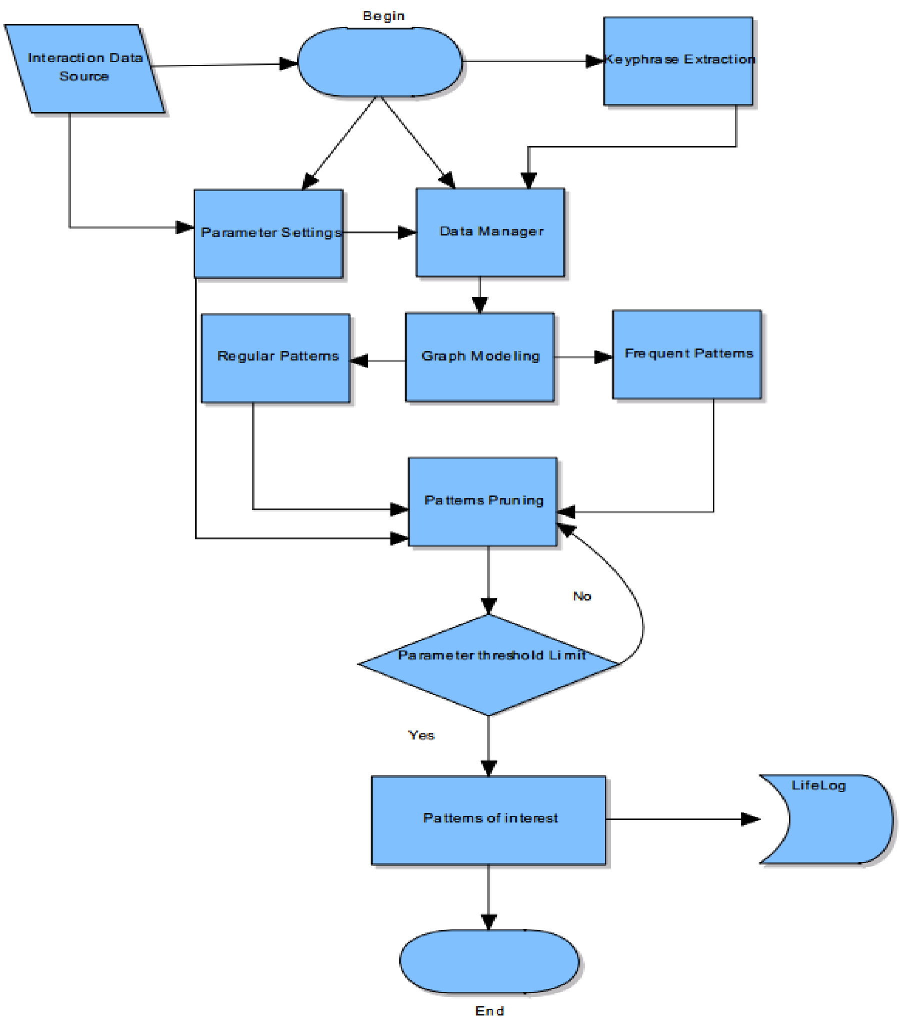


Figure 61 Overall System Flowchart

5.1.4. Behavior Modeling Module (BMM)

- BMM is mainly responsible for collecting the information coming from diverse input sources and logging them in the life log.
- It takes input form HAR, CAME, and Social Media and log them in life Log.
- The life log is the main component of BMM that provide data for life style analysis and prediction. It also provides the facility of data for recommendations and chronic disease recommendations.

- **Introduction**
  - Use of ontology provides basic description to events, activities and interactions, and from this information we can deduce new knowledge.
  - Use of ontology in life log design and development is one of the main contributions of BMM.
  - The life log is modeled as ontology using OWL for all the activities and social interactions.
  - Later it uses the Life Log information for service provisioning.
  - The Life Log information is used for analyzing user behavior for shorter time interval and for longer time intervals which can be even years.
  - On the other hand, BMM also provide the facility for behavior prediction in case the current behavior is matching with some existing (stored, learned) behaviors.
  - The component based framework architecture diagram of BMM is given in main architecture; however, the detailed diagram of BMM is given in Figure 62.
  - The data access and storage is all handled using, Parser designed and developed for manipulating the information handled in BMM.

- **Problems in Existing Systems**
  - To provide better lifecare services, information from different sources is necessary that can avoid any missing information which might result in wrong services. Hence, a challenge exists to both provide and maintain the quality and availability of lifecare services with a minimum cost [Khattak2010] and better accessibility.
  - The systems based on reminder have primarily focused on plan-based approaches to decide how and when to prompt subjects effectively [Mama2000] in addition to offering location-based reminders [Sohn2005]. Nevertheless, the context for the delivery of reminders involves more information than simply location or time. However, these systems are not based on data from different sources and also on data of historical nature.
  - In [Wang2009], an ontology is used to incorporate context related information for intelligent processing; however, these systems provide the services for home held devices interaction and usage. However, they lack in service provision for lifecare services.
  - Work presented in [Chen2012] details an ontology based reminder system; incorporates rules for manipulating the recognized activities of the elderly.
  - In [Khattak2011], real-time activities are recognized using diverse sensors are used for situation analysis.
  - Research in [Fitma2011] focused on the social interaction of patients. Based on the experiences shared, it generates intelligent service recommendations and eliminates their isolation.
  - Existing systems aiming to offer a form of support are mostly based on one input modality, in some cases use imperfect contextual information [Henr2004], and if using di-

verse information with context information then they still provide the facilities of on time service provisioning. These solutions subsequently do not consider the lifecare services provisioning.

- **Requirements**
  - To overcome the limitations of the existing systems and provide better lifecare, life style, and analysis services, there is a need of sophisticated information collection, integration, and manipulation methodologies.
  - The basic requirements are discussed below.
    - Need to collect, integrate, and manipulate information based on the storage structure provided in the Life Log.
    - Need to model a methodological storage structure for information to be stored in Life Log.
    - For the storage structure, a storage model is designed and developed for the Life Log using OWL.
    - For the system to work properly, smooth communication among all the components is required. This needs the components to be atomic in their working.
    - To provide the lifestyle analysis and behavior prediction services, we need to design and develop or use sophisticated data mining algorithms which can fulfill our system needs for service provisioning.

- **Proposed Solution**
  - The Life Log information is collected from input sources i.e., Activity recognition, Social Media, and diet information and used for analysis of user lifestyle.
  - On the other hand, BMM also provide the facility for behavior prediction in case the current behavior is matching with some existing (stored, learned) behaviors.
  - The component based framework architecture diagram of BMM is given in Figure 62.
  - The details of all the components are given in the below given subsections.

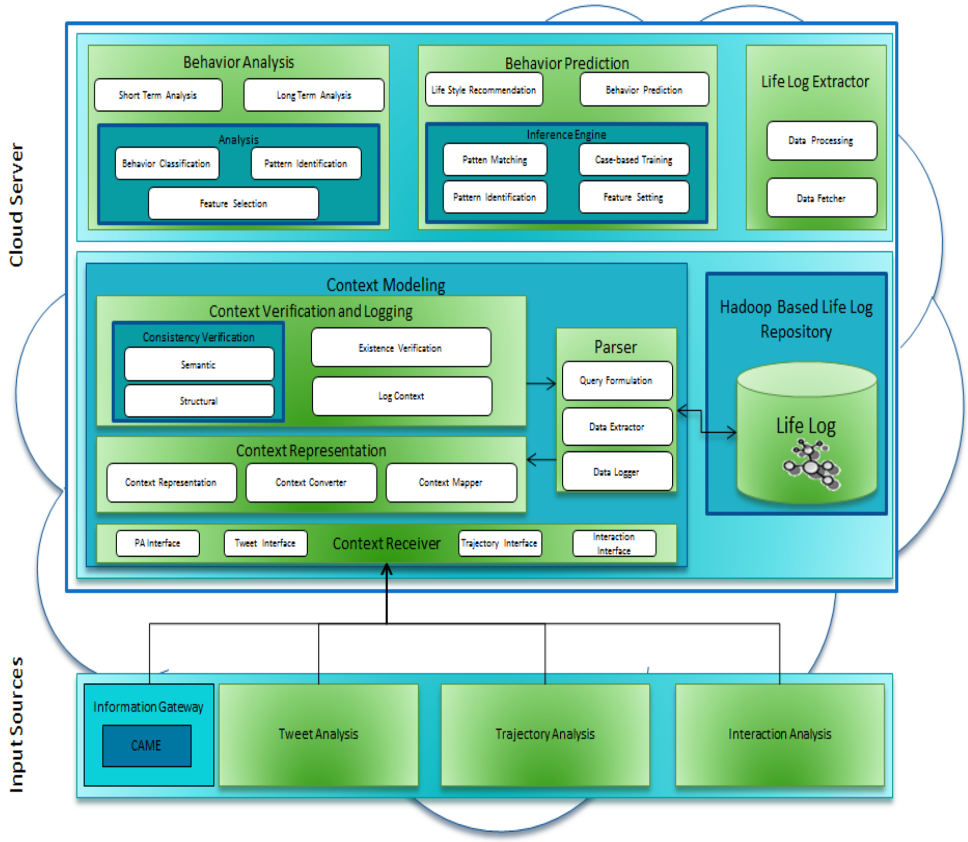


Figure 62 Overall architecture of Behavior Modeling Module (BMM)

- **Data Receiver**
  - This component is responsible for recognizing data inputs from diverse sources. Activities, Social Media, and diet information is provided in different formats like text (example given below in Figure 63) and/or in XML.
  - So there is need of recognizer that will recognize the simple representation of incoming information for their type and will initiate their respective queries in the Parser component.
  - There are different interface implemented in Data Receiver component that listen for appropriate incoming information.

- **Parser**
  - To store and access the information in and from Life Log, appropriated access patterns are defined.
  - It contains three main components: 1) Query Formulation, to formulate the query. 2) Data Extractor, for extracting data from Life Log. 3) Data Logger, for logging the data in Life Log once the information is formalized.

```
20121104      I'm  thinking,  LONG:0,  ACCx:-
0.1915396558,  ORIx:42.0038197266,  ORIz:-1.0,
MAGz:-46.0000697266,Instance:1
```

```
Topic: Exercise
Interest: Food
Time: 10:10:32 hr
Sentiments: Positive
User: u2
```

Figure 63 Text representation of activity information and social media information.

```
Lifelog: Activity_Instance
a                        LifeLog:Activity
Lifelog: hasInterest     Food
Lifelog:talkAbout        Exercise
Lifelog:hasTime:         2012:11:10:10:10:32
Lifelog:User:            User1
```

Figure 64 OWL representation (N3) of social media information collected using tweet analysis.

- **Context Representation**
  - This module is mainly responsible for representing the information the representational structural returned by Parser.
  - It consists of three main sub components: 1) Context Mapper which maps the incoming context information with the storage structure. 2) Context Convertor which convert the context from simple text format to OWL format. 3) Context representation that represent the final converted context in OWL (N3) representational format as shown in Figure 64.

- **Context Verification and Logging**
  - This module is mainly responsible for verification of the context represented in the previous module. The verification is for structure, existence, and semantics.
  - Once the represented information is verified then the information is logged using Log Context.

- **Life Log**
  - Life Log is the main repository of BMM and LDSS for Wellness and LDSS for Chronic Disease.
  - It is responsible for proper communication of information among all the components of BMM.
  - It stores all the activities, diet, and social media information in different context/situation.
  - The proper engineering of the Life Log (See Figure 65) is most important activity in the development of BMM.
  - Life Log is designed as an ontology, where all the information is semantically modeled and available for lifestyle analysis and behavior prediction.

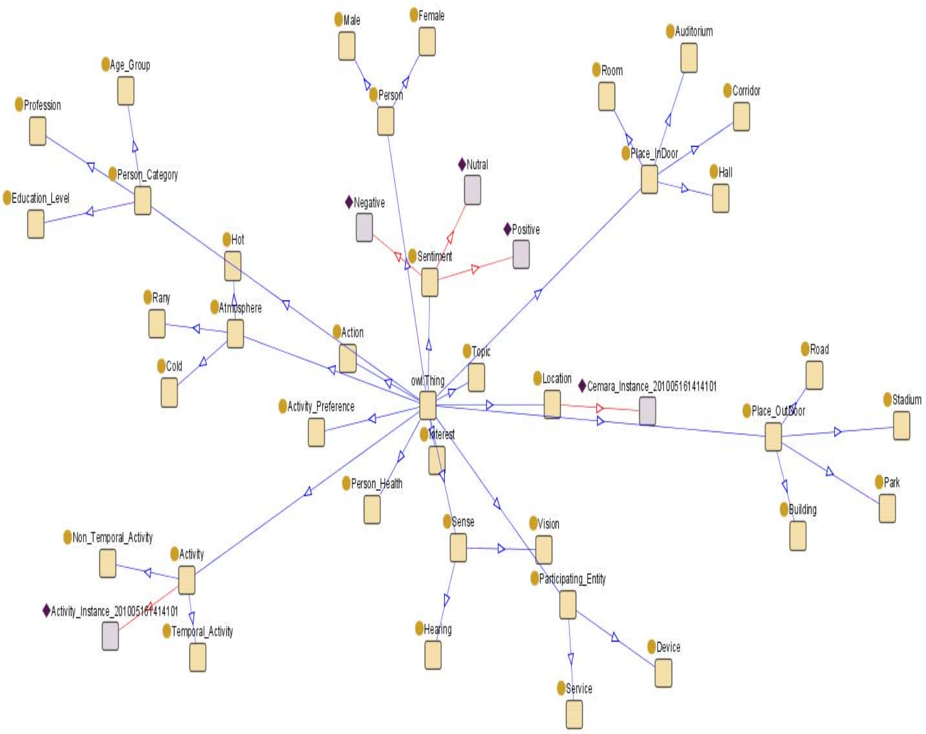


Figure 65 Life Log for information storage to be used in BMM

- **Life Log Extractor**
  - This component is responsible extracting information from Life Log.
  - It consists of two main components: 1) Data Fetcher is responsible for fetching appropriate information from Life Log to provide the information for appropriate services. 2) Data Processing is responsible for preprocessing of information extracted from Life Log and provides it to services.

- **Behavior Analysis**
  - It is responsible for analyzing the Behavior of a person for its activities and his consumed services.
  - These services help the user to have a look at his life from different aspects and find out any problems which if removed can make his/her lifestyle better.
  - These services have two subservices: 1) Short Term Analysis that provides the analysis services for shorter time like for a week or 10 days. 2) Long Term Analysis provides the services for lifetime activities and information.

- **Behavior Prediction**
  - This service is responsible for providing the behavior prediction services.
  - This service is also composed of two subservices: 1) Lifestyle Prediction that predicts the user lifestyle based on his /her activities and matching it with the already

learned lifestyles stored in the repository. If the lifestyles are matching in some of its aspects then the system generates the predicted lifestyle. 2) Behavior Prediction is based on user everyday behavior and lifelong behavior. Based on the user behavior monitored using activity recognition systems and social media information, the system analyze the behavior and predict the possible future behavior of the user in a given situation.

- **Overall System Flowchart**
  - The overall BMM system flow chart with all components working is given in Figure 66.
  - This shows that how different components of BMM work together in integration and how the output of one component flows to the other component where it is used as input in the second component for further processing.

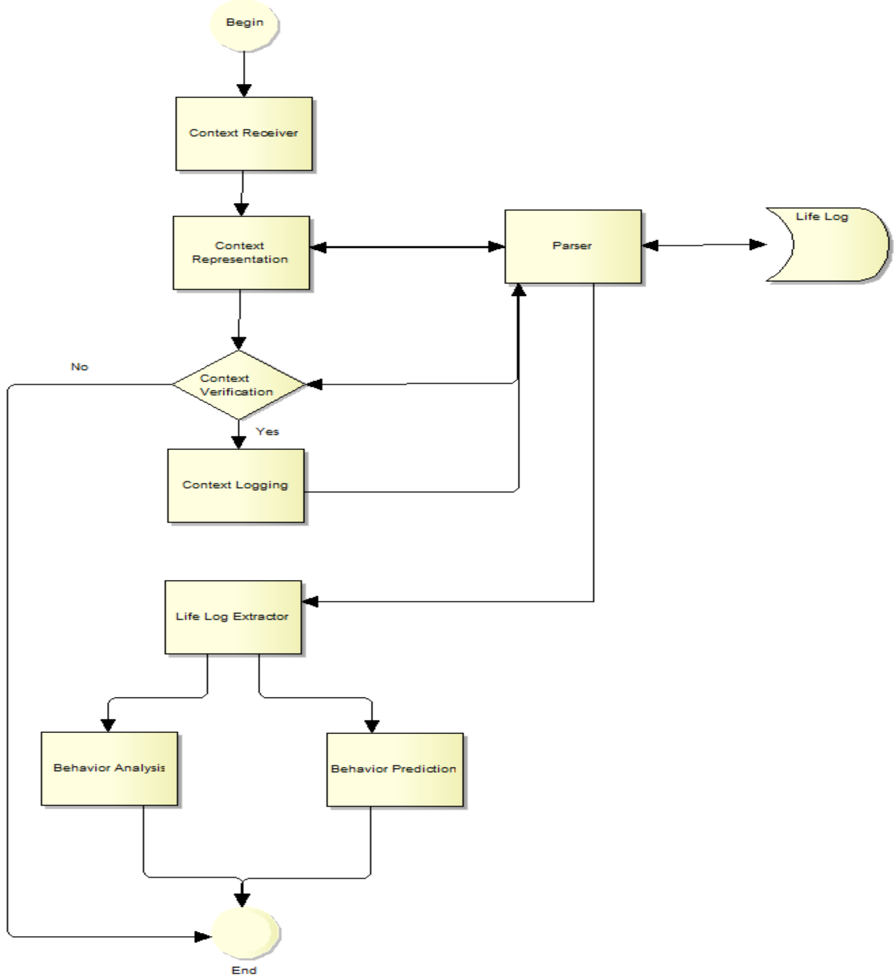


Figure 66 Flow Chart for overall working of BMM

5.1.5. Automatic Rules Generation and Inferenceing Using Rough Set Theory

- **Introduction**
  - The analysis of a huge volume of data for generating rules is a difficult task in term of labor, time and cost.
  - To make this process easier and faster, a mathematical tool, called rough set theory, is used to analyze such huge data.
  - The purpose of this module is: 1-To automatically generate knowledge base form a huge volume of data with the help of Rough Set Theory and 2-To generate wellness recommendation for the users.
  - We are proposing Rough set base inferencing, Case base inferencing and Bayesian based inferencing for generating recommendation (for the time being we are focusing on rough set theory).

- **Problems in Existing Systems**
  - Personalized wellness therapy recommendation system [Husain, W., 2010] has been developed that uses hybrid case-based reasoning and rule-based reasoning for recommendations.
  - Paparo [2005] has developed an automated wellness system with which individuals can manage their personalized fitness program on the Internet.
  - [Aino Ahtinen et al., 2009] have presented their research findings on designing social features for mobile wellness applications. The have focused on opportunities to support and motivate wellness by utilizing and enhancing social interaction between users.

- **Requirements**
  - To overcome the limitations of the existing systems and provide healthcare and smart homes recommendation services, there are different requirements that need to be satisfied.
    - It should be determined in advanced that from which component of the lifelog data should be extracted for the requested query of the user so that an appropriate recommendation is generated.
    - To process the data retrieved from life-log for rules generation, a specific technique is needed to represent the data.
    - For the system to work properly, smooth communication among all the components is required.
    - For rough set based approach discretization of the input data both for rules generation and user's query is needed.
    - For rules generation using rough set, an effective features selection technique is needed to filter out those features that don't play any prominent in the decision process.

- **Proposed Solution**
  - The proposed framework for automatic rules generation and recommendations is given below in Figure 67.
  - Main components of the system are: data manager, interface engine, data acquisition engine, knowledge inference engine and rule-base.

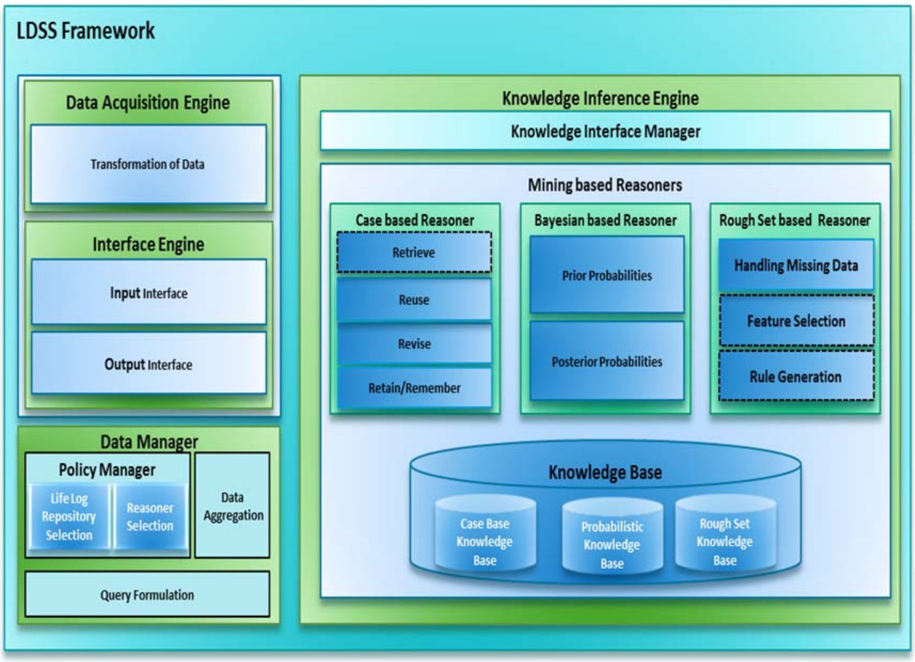


Figure 67 Architecture for Automatic Rules Generation and Inferencing.

- **Data manager**
  - This module is used to select particular reasoning techniques out of rough set, Bayesian and case-base and retrieve data from lifelog using formulator and data aggregator.

- **Interface engine**
  - This component provides the standard input and output interfaces for interring and correcting input data and displaying the output results.

- **Data acquisition engine**
  - This component transforms the input data into a standard process able format of a particular technique selected by data manager.

- **Knowledge inference engine**
  - This is the main component of the LDSS Framework which further consists of: knowledge interface engine, reasoners and knowledge bases.

- **Knowledge interface manager**
  - It is an interface between the data acquisition engine and the reasoning module.

- **Knowledge reasoner**
  - We have proposed three reasoned for generating recommendation which are: rough set based reasoned, Bayesian reasoned and case-base reasoned.

- **Knowledge base**
  - This is the storage for the knowledge generated in the first step of rules generation from the data.

▪ **Overall System Flowchart**

- **Flowchart for rules (knowledge-base) generation (Figure 68)**

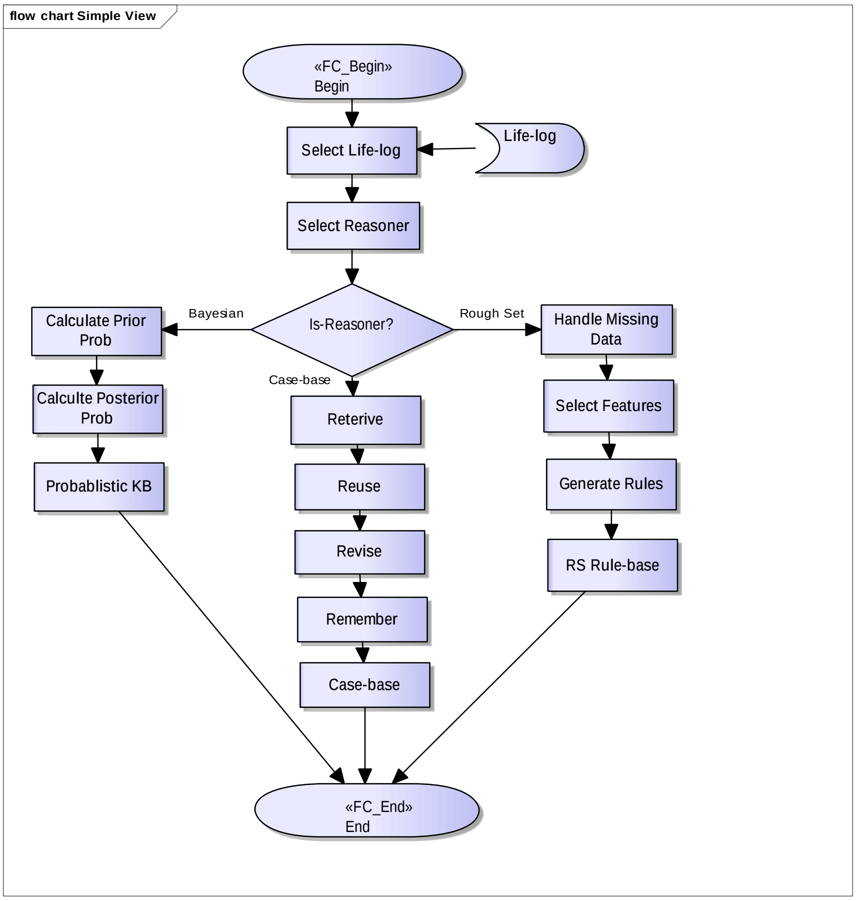


Figure 68 Flow chart for rules generation

- **Flowchart for recommendation generation (Figure 69)**

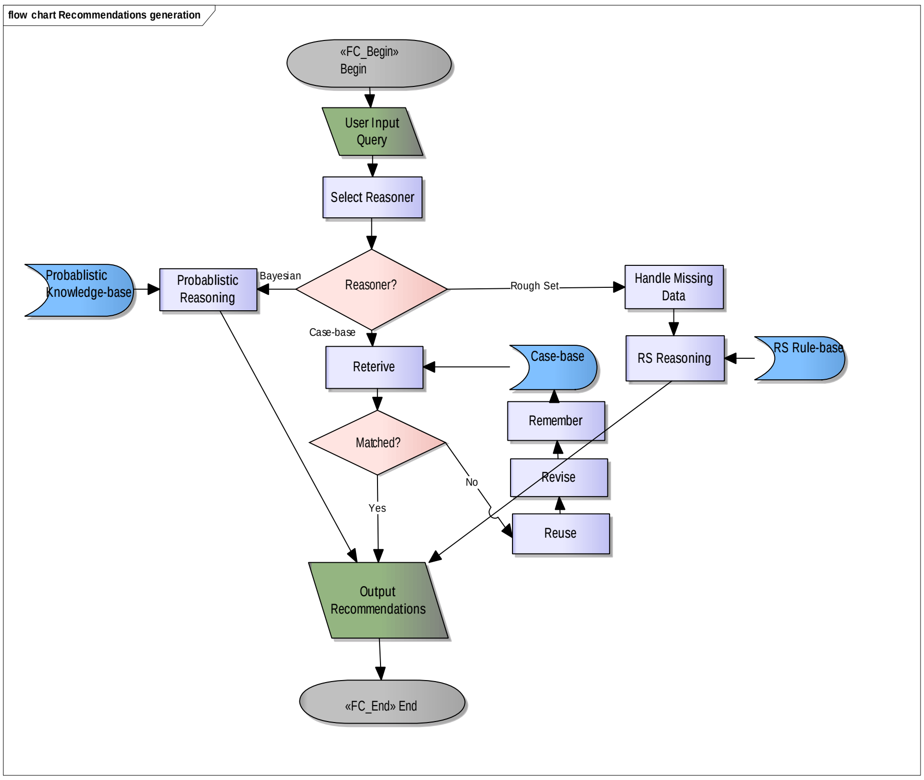


Figure 69 Flow chart for recommendation generation.

5.2. LDSS for Chronic Disease

- LDSS-Chronic Disease is a system that facilitates physicians and chronic patients in management of their time and resources effectively.
- Physicians are too busy to monitor chronic patient activities regularly or to inquire about their health regularly, whereas it's very difficult for chronic disease patients to regularly visit physician clinic for routine checkup.
- Therefore, LDSS-Chronic Disease is necessary for physicians and chronic patients to monitor daily life activities of the chronic patients and find abnormalities in it to inform patients and physicians on time for taking preventive measures.
- The activities can be monitored from different sources such as social media, sensors and cameras, clinical and monitoring life log patterns.
- Therefore the system has input from different sources that are concatenated for reasoning to provide recommendation to the user.

5.2.1. Human Activity Recognition (HAR)

- It consists of the multimodal sensor based activity and emotion recognition modules.
- The outputs of these individual independent modules are fuse together to predict the real-time high level contexts.
- The whole architecture is decomposed into the subcomponents and details are provided in the subsequent section.

5.2.1.1. Audio Based Emotion Recognition

- **Introduction**
  - Emotion is a mental state that arises spontaneously. In daily life, emotion is not only an effective way to convey our intention in communication but also a good indicator of our mental health. That is the reason why automatic detection of human emotions is an important factor to enhance the quality of the service provided by the computer such as human-computer interaction [Cowie2001, Schuller2004], lifestyle monitoring in ubiquitous health care systems [Tacconi2008].
  - While human emotion can be expressed by a variety of physiological changes such as speech, blood pressure, heart rate, facial expression, etc.; many researchers prefer acoustic speech as a source of emotion [Ayadi2011, Bitouk2010, Iliev2010, Lee2005] because speech signal is the most commonly used and most natural method of human communication.
  - Finding the best feature extraction and classification for emotion recognition from speech signal still are the current challenge for researchers.

- **Problems in Existing Systems**
  - When classification methods have achieved successful results with different classifiers, feature extraction for emotion recognition is still an open area.
  - There are a lot of feature extraction methods that have been used to extract more and more relevant features for emotion recognition from speech signal [Cowie2001, Lee2005, Banse1996, Gobl2003, Nwe2003]. But we still don't know what the best features for this task of recognition are.

- **Requirements**
  - In order for system working properly and efficiently, there are many requirements are needed, some of important requirement as follows:
    - Training data is needed to be recorded as much as possible with different voices and emotions. All other sounds affecting to voice quality should be avoided.
    - Emotions to be recognized should be predefined and differentiate together clear enough.

- **Proposed Solution**
  - The main function of this component is recognizing emotional content of unknown speech signal. It takes a speech sentence from audio sensor as input and then process to classify to one of different emotions. The output emotion has the form of text label such as anger, happiness, boredom, sadness, normal, etc.
  - This AER component has 3 main sub-components called:
    - Data Acquisition
    - Feature Extraction
    - Classification
  - Each of them has a unique important role in the whole process of this component. It shares the similar architecture with almost classification applications that need to have at least three parts of collecting data, extracting features and classifying.

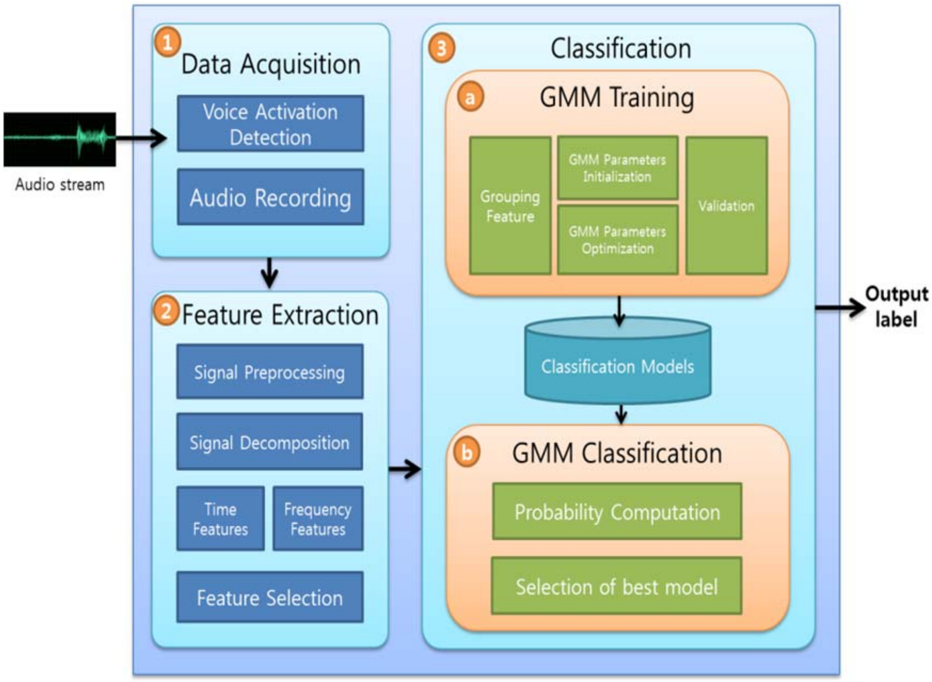


Figure 70 General Architecture of Cloud Computing

- **Data Acquisition:** The main purpose of this sub-component is to detect and record audio data for further process. Input audio stream from sensor is continuous and contains audio and speech signal as well.

- We need to detect when speech signal appears in audio stream so that we can record speech signal. To reduce the complexity of this process, we can assume that the audio stream contains speech signal only, so we just need to detect speech signal by using energy threshold.
- Whenever the average energy is larger than threshold, the audio recording module is activated to start recording speech signal. If average energy becomes lower than threshold, it means a sentence is completed, recording module is stopped, and then recorded speech signal is stored as a file.

- **Feature Extraction:** This is one of two most important parts that takes a speech sample as input and then generates a feature vector or a sequence of feature vectors. The main purpose of this process is extract the most relevant features that can describe clearly about the processing signal, and can be used to classify between different classes of signal. Specially, we need to extract the most valuable features that can classify emotional states from speech signal. Depend on what kind of signal and application, there are different processes inside feature exaction component. In this architecture, we have:
  - Signal Preprocessing: to reduce noise that can affect to quality of signal and reduce the accuracy of classification. In this process, we can apply spectrum subtraction to reduce background noise.
  - Signal Decomposition: Matching Pursuit is applied to decompose the input signal into a block of atoms that are small-predefined signals [Mallat1993, Umapathy2005a, Ummapathy2005b, Chu2009]. And then depend on the distribution of these atom we can extract different features.
  - Time feature and frequency features are extracted from output block of atoms by using temporal and spectrum histogram.
  - Feature Selection: In feature extraction stage, we can use different kind of features, some of them are good, but some of them are not good, they affect to accuracy of classification. We apply feature selection here to reduce the irrelevant features in order to increase the accuracy.

- **Classification:** In this stage, a machine learning algorithm is applied to learn the properties of processing signal, and to differentiate between different classes of signal. To make balance between simplicity and performance, we apply Gaussian Mixture Model (GMM) as classifier to recognize different emotional states in speech signal. GMM is a lightweight classification algorithm, so that it is easier to apply on smartphone that have limited computational resources. Similar with other classification, there are 2 main processes training and classifying.

  - ***Training process:*** This training is used to learn properties of each class of signal by group all training data of class and apply Expectation Maximization to generate the parameters of GMM model. Each emotional class of speech signals is used to generate a GMM model. For example we have 4 classes Normal, Angry, Happy and Sad, so 4 GMM models are trained in this step.

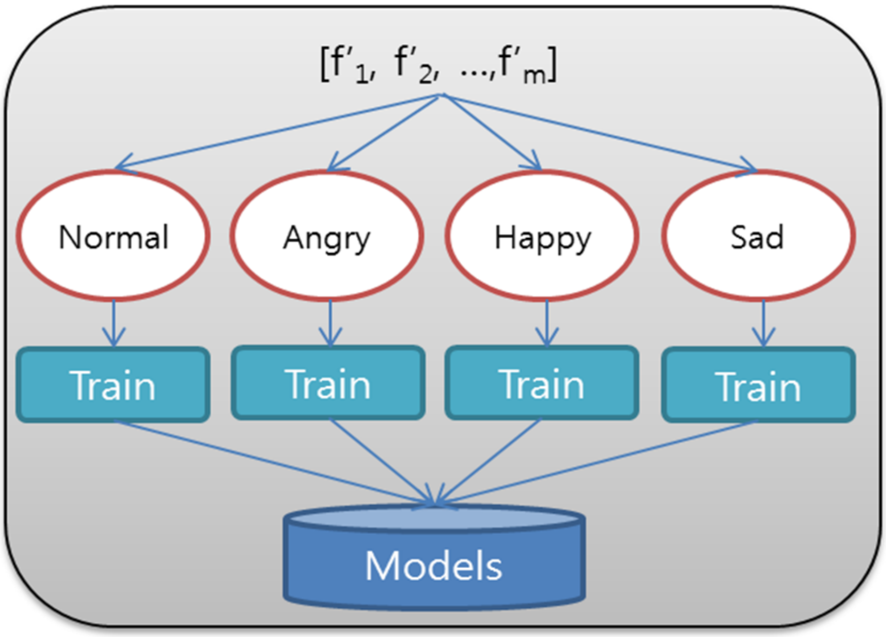


Figure 71 GMM Training process for 4 emotional classes

- Grouping features: In order to train models for different classes, we have to extract features for all training data and manually split them into different groups according to emotional classes. For each group, we apply GMM training to generate model for this one.
- GMM Parameters Initialization: we initialize parameters for each GMM model randomly
- GMM Parameters Optimization: in this process, EM algorithm is used to optimize parameters of a model. This is a recursion process to fit parameters of GMM model with distribution of training data in iteration. The process is stopped when it reaches to a number of iterations or is below a threshold.
- Validation: This process is to validate predefined parameters of a model by using them to recognize a set of validating data. Which parameters have better performance and accuracy are chosen for classifying stage.
- All trained models are stored in a local database or provided to mobile side for further process.

- ***Classifying stage:*** This process is deployed at PC side or mobile side as well depending on what kind of application. The main purpose of this process is to find from all trained models the best model that has the largest probability with unknown input feature vector compare with other ones.

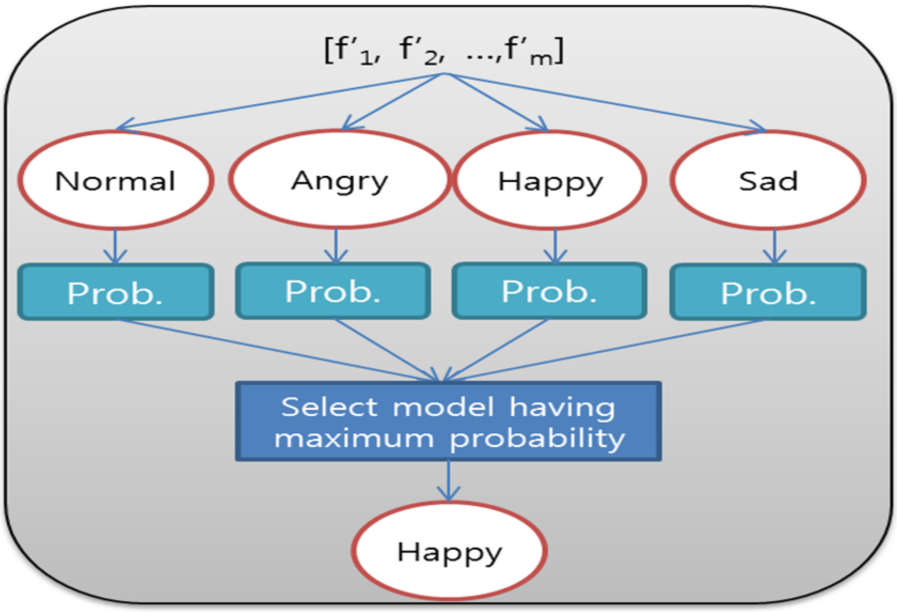


Figure 72 Classifying unknown input signal given 4 GMM models

- Probability Computation: when the input feature vector is available, its probability with each GMM model is computed using Gaussian distribution.
- All probabilities are compared to select the largest one. And then we consider the model with the largest probability is the output of classifying stage.

- **Overall System Flowchart**
  - Following flowchart show how the system works. It contains two phases: training and testing.
  - This shows that how different components work together and how the output of one component flows to the other component where it is used as input for further processing.

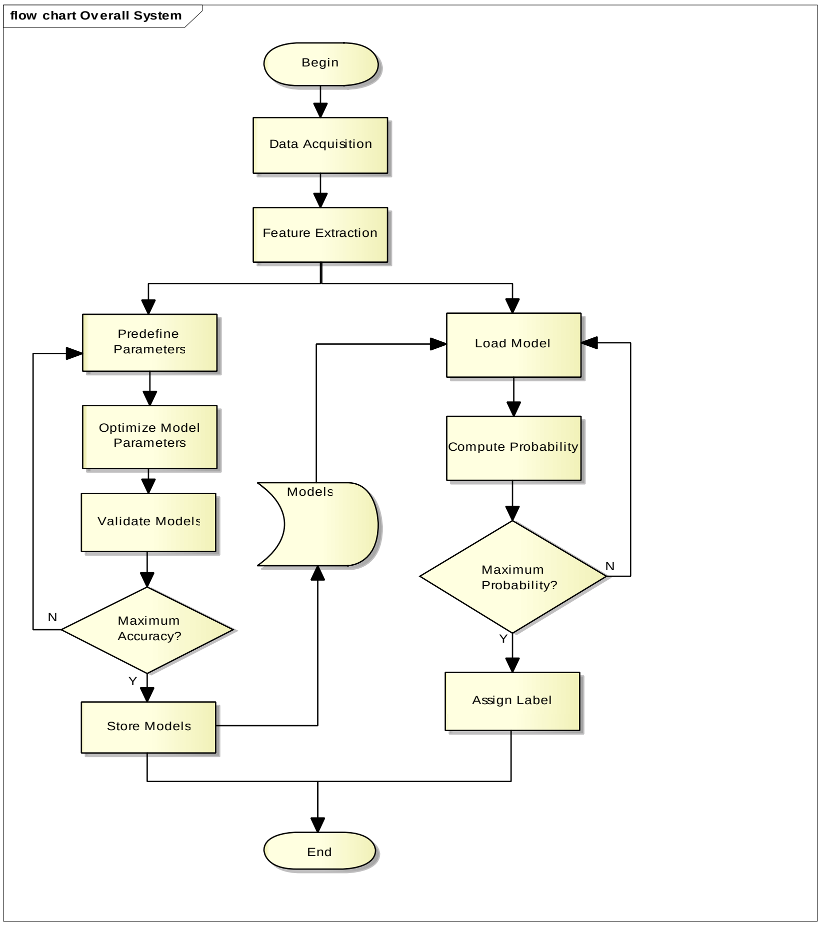


Figure 73 Overall system flowchart

5.2.1.2. Video Based Activity Recognition

- Automated systems are becoming increasingly involved in our daily lives. To correctly handle a situation either to help people to finish a job in a friendly environment or to prevent people from error in a hostile situation, the system must know what is happening. Human activity is a very important factor in this process. Knowledge of what people are doing can enable a wide range of assisting technologies, smart appliances, and aware environments.
- Human activity recognition has a good application in smart home technologies in all over the world to monitor human activities and take necessary steps. For instance, the SmartBo project in Norway includes a two-room house for the elderly with mobility impairments and cognitive disabilities. In this project, its main system including lighting, doors, windows, and shutters is controlled by smart devices and sensors and generates an alarm when something goes wrong [Elger1998].
- Security is another major application domain that requires an understanding of activity. Video

surveillance is becoming standard in all sorts of public areas as well as the private backyard. Having a person behind each camera is simply impossible. An automated filter system as well as abnormal situation detection system has generated enough market demand for companies, which provides shopping customer identification.

- **Introduction**
  - Video-based activity recognition refers to an algorithm that a computer system uses to automatically recognize what human activity is being or was performed, given a sequence of images (video frames). In recent years, this problem has caught the awareness of researchers from manufacturing, academia, safety agencies, consumer agencies, and the general populace too [Turaga2008]. In the vision literature, the words ‘Action’ and ‘Activity’ are habitually utilized identical. The word ‘Action’ means the patterns of simple motion that usually executed by a single person and normally fixed for short durations of time, on the order of tens of seconds [Turaga2008]. Examples of actions include bending, walking, swimming etc., as shown in Figure 74.

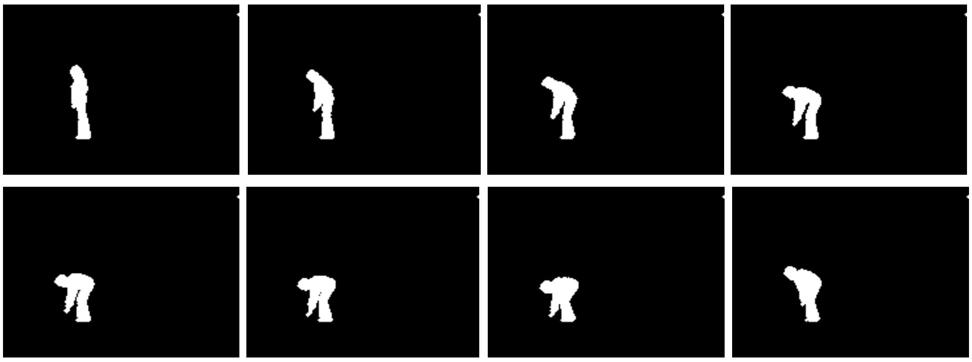


Figure 74 Examples of bending actions.

  - On the other hand, the word ‘Activities’ refers as the complex sequences of actions that are performed by several humans who could be interrelating with each other in an inhibited manner. They are normally characterized by much longer temporal durations, e.g. two persons shaking hands, a football team scoring a goal or a co-ordinated bank attack by multiple robbers [Georis2004] as shown in Figure 75.

Figure 75 Medium field sequence of a simulated bank attack (courtesy [Georis2004]).

  - (a) Person enters the bank. (b) Robber is recognized to be a stranger, and is entering the bank secure. (c) A customer flee, (d) Robber creates an exit [Turaga2008].
  - Generally there are two types of activities that a human is performing commonly. The

first one is low-level activities also called micro activities (μ-activities) e.g. sitting, standing, running, walking, one hand waving or two hands waving etc. The second one is high-level activities also called macro activities e.g. watching TV, playing tennis, ridding bus etc. the general architecture of the high-level activities is given as in Figure 76.

-

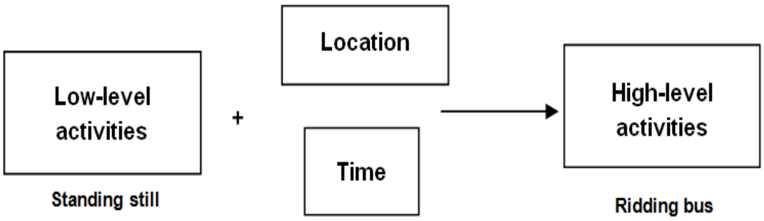


Figure 76 General architecture of high-level (macro) activities.

- The architecture of human activity recognition system consists of three main modules: segmentation, feature extraction, and recognition. *Segmentation* is used to extract the human body shape from video frames and generate corresponding binary images. *Feature extraction* deals with getting the distinguishable features each body shape and quantizing it as a discrete symbol. In *recognition* module, a classifier such as HMM (Hidden Markov Model), or GMM (Gaussian Mixture of Model), or SVM (Support Vector Machine) is first trained with training data and then used to generate the label of human activity contained in the incoming video data as shown in Figure 77.

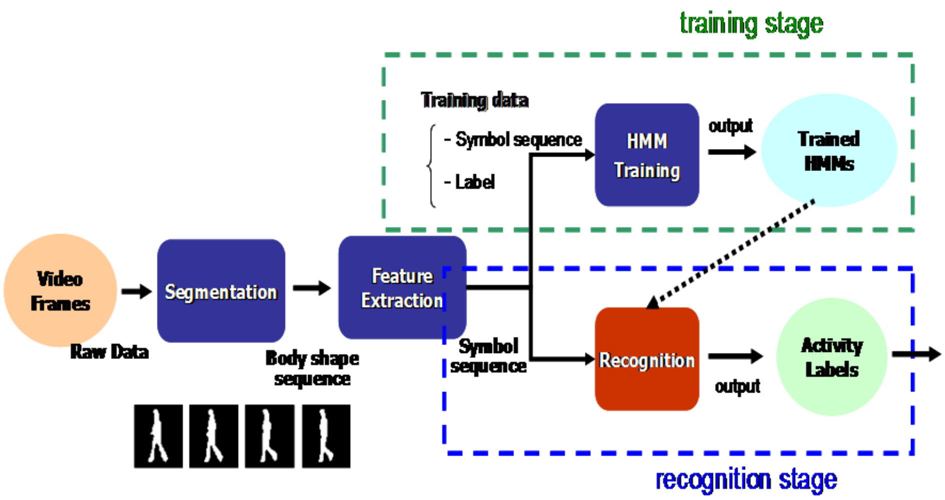


Figure 77 Block diagram of a typical video-based AR system.

- **Problems in Existing Systems**
  - Numerous video-based human activity recognition systems have been proposed; however, recognizing human activity recognition accurately is still a major concern for most of these systems. This lack of accuracy could be attributed to various causes, such as different light conditions for training and test images, lack of accurate automatic human body segmentation

methods, and high similarity among different activities that occurs in the presence of low between-class variance in the feature space. A typical activity recognition system consists of three basic modules.

- **Human Body Segmentation.** The accuracy of recognition module is completely depends on automatic human body segmentation. Some methods have been investigated for human body segmentation including [Siddiqi2010, Uddin2008]; they just subtracted the empty frames from the video frame to segment the human body as shown in Figure 78.

Figure 78 Body segmentation using empty frame subtraction from the video frame.

- However, these techniques were not applicable in real world environment, because in some situation, there is no empty frame, so it is very hard to segment the human body by using these techniques. Due to these limitations, these methods are known as heuristic techniques. Another method has been developed by [Doulamis1999] for segmentation, but this technique produced artifacts due to occlusion. A well-known method was proposed by [Kass1988], but mostly, the human body is in concave form, and to segment the concave objects is one of the limitations of this method. Some of the existing methods often involve modeling of the human body and/or the background, which normally requires extensive amount of training data and cannot efficiently handle changes over time. Recently, active contours [Chan2001] have been emerging as an effective segmentation technique in still images.

- **Feature Extraction.** Moreover, the accuracy is also depends on the extraction of good features that means, the effectiveness of the extracted features will also affect the accuracy of recognition module Some well-known techniques employed for the feature extraction module include [Uddin2008, Uddin2010]. They used Principal Component Analysis (PCA) and Independent Component Analysis (ICA) for feature extraction. However, PCA yields uncorrelated components. If the data have a Gaussian distribution, the uncorrelated components are independent. However, if the data are merged of non-Gaussian components, then PCA fails to extract components having non-Gaussian distribution [Buciu2009]. Also, PCA is an unsupervised technique that locates PCs at the optimally diminished dimension of the input. For human activity, it only focuses on the global information from the binary silhouettes, which results in low accuracy [Kim2010]. To solve the limitation of PCA, a higher order statistical approach Independent Component Analysis (ICA) has been employed by [Uddin2010] for feature extraction. However, ICA is slow to train when the dimension of the data is bulky. Moreover, these features are receptive to decipher and scaling of human body postures which concerns the silhouettes extraction procedure. Also, ICA is very weak in managing the inputs. If there are plenty of video frames, exploited as input, ICA does not have the capability to organize it, due to which some time ICA cannot retrieve the desire features as shown in Figure 79.

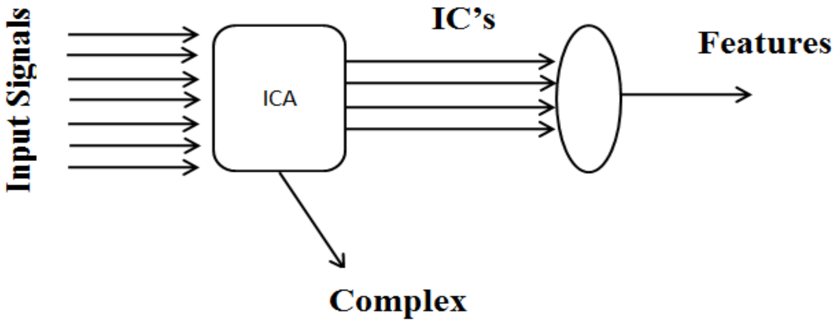


Figure 79 Weakness of ICA in-term of feature extraction.

- **Recognition.** While there is quite a large number of recent research focusing on improving the feature extraction stage [Huang2007, Robertson2006, Xue2010], almost all the proposed activity recognition systems utilize conventional learning methods, such as Hidden Markov Model (HMM), Support Vector Machine (SVM), Gaussian Mixture Model (GMM), Artificial Neural Network (ANN), etc. Among these classifiers, HMM is pointed out by several studies to be the most commonly used method [Uddin2008, Uddin2010]. Even though HMM is widely used for activity recognition, still it has serious deficiencies. It is very difficult to represent multiple interacting activities by using HMMs [Gu2009]. HMMs are also incapable of capturing long-range or transitive dependencies. Additionally, without considerable training, an HMM may not be proficient to recognize all of the possible observation sequences that can be consistent with a particular activity [Kim2010].

- **Requirements**
  - To overcome the limitations of the existing systems and to provide better decision on activity recognition in smart homes, there are different requirements that need to be satisfied.
    - Information coming from video (2D) camera.
    - To store the information (i.e., human activities) coming from input source, we need to decompose it into frames and then store them for decision making.
    - For the system to work properly, smooth communication among all the components is required.
    - The need is to develop an automatic human body segmentation technique.
    - For knowing the activity, a recognition engine is required that will facilitate the users to understand their daily activities.

- **Proposed Solution**
  - **Unsupervised Segmentation Technique.** Accordingly, the purpose of this study was to propose an accurate and robust activity recognition system capable of exhibiting high rec ognition accuracy. A model is proposed that use unsupervised segmentation in combinati on with motion information. Specifically, in each video frame, an active contour (AC) [K ass1988] is evolved to capture the human body and the motion information (i.e., optical f low) is used to move the contour toward the human position in the next frame, where the contour is again evolved to detect the exact boundary of the body that avoid the need of s upervised learning. In AC models, the initial contour should be close to the object in orde

r to converge correctly. This can be done manually in static images but will not be feasibl e in video data which have a large number of frames. Automating the initialization of AC is therefore needed. Optical flow is a good candidate for this purpose because it contains the relatively exact direction and magnitude of the motion between consecutive frames. According to the afore-mentioned classification, the proposed approach falls into the sha pe-based category with partial incorporation of temporal context. By "partial", means tha t the temporal information is not embedded in the contour evolution itself but instead is u sed as a guideline for positioning the initial contour. The proposed approach is unsupervi sed, i.e., no training data or prior human model is needed. Experimental results show that AR using segmentation results of our approach has accuracy comparable to the one usin g manual segmentation, proving the feasibility of our model in practical activity recognit ion.

- **Robust Feature Extraction & Recognition.** We proposed a complete approach for video based human activity recognition that employs wavelet transform and hidden conditional random fields model. To obtain the feature vectors, symlet wavelet family, was tested and decomposed up to 4 levels to recognize the human activities. Some of the highest coefficients were extracted from each level of decomposition at the feature extraction stage. These coefficients were based on the average frequency of each video frames and the time difference between each frame, and then a novel hidden conditional random fields model was applied for recognition. The proposed method was tested on a database of nine activities. Data collected by ten different people, so the whole collected samples of activities were 90. The recognition results were compared with some of the existing techniques that used Principle Component Analysis (PCA), Independent Component Analysis (ICA), and Hidden Markov Model (HMM). These results showed better improvement in performance when compared with the existing techniques applied on the same dataset.

▪ **Overall System Flowchart**

- The overall flowchart for the proposed activity recognition system is shown in Figure 80.

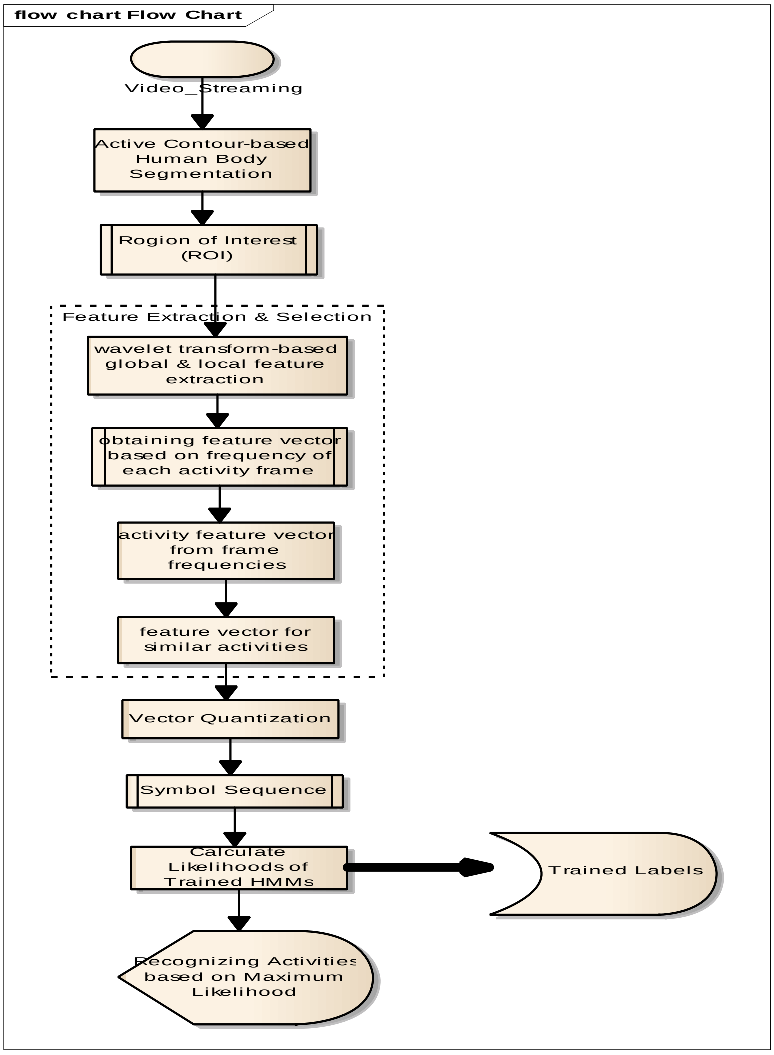


Figure 80 A video-based activity recognition system

5.2.1.3. Multimodal Sensor Fusion

- **Introduction**
  - Now-a-days, sensor technology is the more robust, cost-effective, less-intrusive and easy to install solution for recognizing the human activities [Turaga2008].
  - Single sensor is not adequate to reflect the true context of humans, so multimodal sensors are required such as embedded sensors, wearable sensors and location sensors to recognize the true context.
  - Fusion engine is an effective solution to combine the information from these heterogeneous modalities [Dante2010].

- **Problems in Existing Systems**
  - Many researchers have devised numerous methods that can be divided into three categories: rule-based methods, classification-based methods, and estimation-based methods [Pradeep2010].
  - Classification-based methods include a range of classification techniques that have been used to classify the multimodal observation into one of the pre-defined classes [Pradeep2010]. These techniques are inadequate in context recognition due to lack of support to handle mutually exclusive hypothesis and become hard to handle large number of combinations of hypothesis.
  - The estimation-based fusion methods such as Kalman filter and particle filter are generally used to estimate and predict the fused observations over a period. These methods are suitable for object localization and tracking tasks [Kulkarni2011].
  - According to the nature of the problem, we explore the Rule-based methods and proposed our novel design to fulfill the requirement of healthcare domain that demands high accuracy.
  - The existing rule based methods such as majority voting and linear weighted fusion are the simplest techniques to combine the output of different modalities. It is computationally less expensive approaches but major issues are optimal weight assignment, less specific to the problem domain, and less flexible to add new rules [Pradeep2010].
  - New methods are required, which are capable of fully automated adjustment and self-adaptation to recognize the true context. Our fusion engine will automatically extracts the rules and utilize probabilistic theory to increase the level of accuracy.

- **Requirements**
  - To overcome the limitations of the existing systems and provide healthcare and smart homes services, there are different requirements that need to be satisfied.
    - Information coming from diverse sources intelligently fuse together like motion sensor, accelerometer, audio/video sensor, and location sensor.
    - To deal with different sources some synchronization process is required.
    - For the understandable output some auto-generated rules mechanism is to be needed to fulfill the requirements.

- **Proposed Solution**
  - Our proposed solution is based on evolutionary technique with probabilistic approach. It will be capable to fuse audio/video, accelerometer, location, and embedded sensors data to improve the robustness and accuracy of the recognized context.
  - Fusion engine is comprised on preprocessing step and three layers architecture to process the sensor modalities. The detail diagram of the fusion engine is illustrated in Figure 81.
  - Let $\Omega = \{S_1, \dots, S_n\}$ be a set of n sensors, e.g., {wearable, location, camera, embedded etc.} characterized by m attributes $A = [a_1, \dots, a_m]^T \in \Re$, which include binary or numerical value, to express the individual activity recognition for specific instance of time.
  - Let C classes $C_1, \dots, C_n$ has the objective to recognize the human activities automatically, so rule set (RS) for base learner is:

    $$\text{RS:}\ \Re^n \longrightarrow [C_1, \dots, C_n] \qquad (4)$$

    i.e., the output RS is n-dimensional rule vector, whose i^{th} component denotes the "rule" for recognizing the activity, which comes from sensor S_i by evolving genetic algorithm. The combination of the outputs of individual base learners "BL" is:

$$\mathrm{DF} = \max\left(\mathrm{RS}_1(\mathrm{BL}) + \ \ldots + \ \mathrm{RS}_n(\mathrm{BL})\right) \qquad (5)$$

i.e., DF is sensor fusion with the highest supporting class, determined by majority voting, accuracy matrix calculated by maximum likelihood probabilistic technique and RS(BL) is an aggregation operator to combine the each base learner rule set. The architecture is as follows:

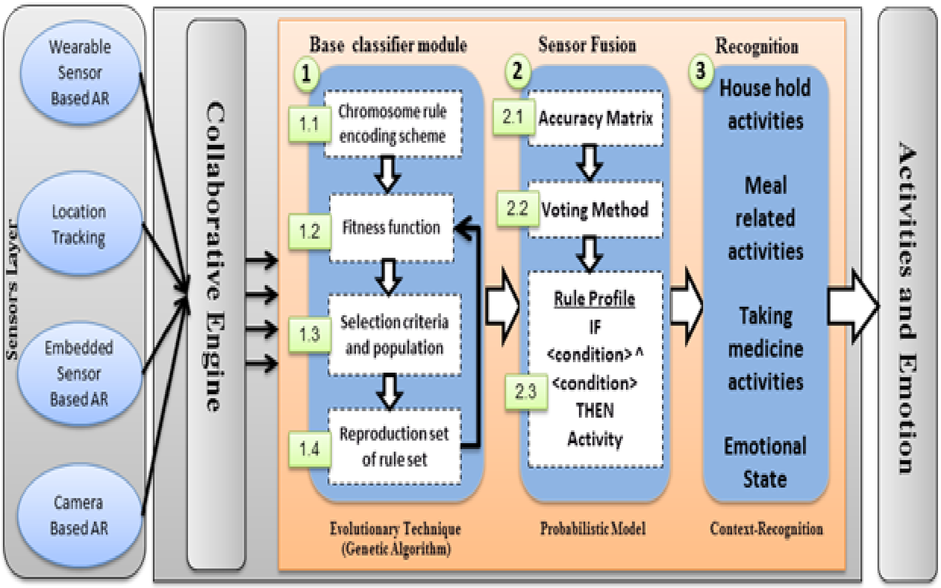


Figure 81 Multimodal sensor fusion architecture

- **Base classifier module**
  - Base classifier is based on genetic algorithm. In this phase, candidate rule set solutions are randomly generated. To alive these candidate solutions, it needs to survive, otherwise it is dominant by the good candidates [Jiao2006]. Initial population is randomly generated and then improves the quality of the candidate rules by evaluating the training instances. Following are the steps to construct the base learner component.

- **Chromosome rule encoding scheme**
  - The rule set is the representation of sensory data; more commonly known as a chromosome encoding structure. In our case, we adopt the Michigan approach [Rogova2004] to encode the sensor data. In which each individual encodes a single rule. Each rule contains a logical combination of sensor value in the form of$\mathrm{attribute}_1 \cap \mathrm{attribute}_2 \cap \mathrm{attribute}_3, \ldots, \mathrm{attribute}_n$.

- **Fitness function**
  - Evolutionary algorithm is guided search space by the fitness function; it must include the quality measures that reflect the accuracy of the candidate solutions

[Rogova2004]. Fitness function is defined as:

$$\mathcal{F} = \sum_{i}^{n}\sum_{j}^{m} reward\ (Candiate_i\ |example_j) - \sum_{i}^{n}\sum_{j}^{m} payoff\ (Candiate_i\ |example_j)$$

$$= \sum_{i}^{n}\sum_{j}^{m} [reward\ (Candiate_i\ |example_j) - payoff\ (Candiate_i\ |example_j)]$$

Where, $reward = \begin{cases} 1 & if\ candidate \equiv example\ \cap classLabel \equiv Correct, \\ 0 & otherwise. \end{cases}$

$$payoff = \begin{cases} 1 & if\ candidate \equiv example\ \cap classLabel \equiv Incorrect, \\ 0 & otherwise. \end{cases}$$

- The fitness function "F" evaluates the individual rule in the rule set. It assesses how many training instances are correctly classified by paying reward and also payoff in case of wrong class identification.

- **Parent Selection**
  - Once these rules are evaluated according to the above fitness criteria then they are ranked. The best fittest rules are passed to next generation as elite.
  - From the remaining solution, some rules are selected based on certain probability x of whole population for crossover operation. This mechanism prevent from losing the best rules which are found in each generation and increase the probability of selection for weak candidates.
  - Fitness ranking and elitism ensures that rules are converging to the global optimum point without stuck into local optima.

- **Reproduction set of rule set**
  - It applies the stochastic operator crossover, mutation and elitism to generate the new population to search the global optimum.

- **Sensor fusion**
- This module is based on the following three components.

- **Accuracy Matrix**
  - This module will again process the rule to maximize the accuracy and assign the weights to each rule according to the importance of context.

    Feature Function: f: X * Y → R

$$p(X|Y,w) = \frac{\exp\left[\sum_{j=1}^{n} w_j F_j(x,y)\right]}{z\ (x_i,w)}$$

$$F_j = \sum_{j=1}^{n} w(i, R_x^i, C)$$

$$z\ (x_i,w) = \sum_{y\in y'} \exp[w_j F_j(x,y')]$$

Taking log on both sides

$$\mathbf{log}\ p(X|Y,w) = \mathbf{log}\ \frac{\exp\left[\sum_{j=1}^{n} w_j F_j(x,y)\right]}{z\ (x_i,w)}$$

$$\mathbf{log}\ p(X|Y,w) = \sum_{j=1}^{n} w_j F_j(x,y) - \sum_{y\in y'} \exp[w_j F_j(x,y')]$$

Maximize the likelihood of data and calculating the gradient with respect to “w”

$$W_{ML} = \text{argmax}_{\mathbf{W}\in\mathbb{R}^m} L(\mathbf{W})$$

$$L(\mathbf{W}) = \sum_{i=1}^{n} \log P(y_i \mid x_i)$$

$$= \sum_{j=1}^{n} w_j F_j(x, y) - \sum_{y\in y'} \exp[w_j F_j(x, y')]$$

$$\left.\frac{dL}{d\mathbf{W}}\right|_{\mathbf{W}} = \sum_{j=1}^{n} F_j(x, y) - \sum_{i=1}^{n} \sum_{y\in y'} \exp[w_j F_j(x, y')]$$

- It is a probabilistic and discriminative model. It has a number of features and each feature has a weight for correct classification. Feature weights are optimized on development data, so that the system output matches correct translations as close as possible and minimize the error rate.
- The goal is to maximize the likelihood of the data and consequently results in good feature weights.

- **Majority voting**
  - It is most common method for combining the output of multiple learners [Rogova2004]. Let assume we have “L” base learners and each have “n” rules based on sensors value to recognize the activities. Then we can define majority voting for the base learners as depicted in equation 3.

$$\sum_{i=1}^{L} \mathrm{DF} = \max_{j=1}^{C}\left(\sum_{i=1}^{n} \mathrm{RS_n}(\mathrm{BL_n})\right) \qquad (6)$$

  - In Equation.3 “DF” is the decision fusion vector which determines the most confident class label for any instance.

- **Rule profiling**
  - Each rule contains a logical combination of sensor value in the form $\mathrm{attribute}_1 \cap \mathrm{attribute}_2 \cap \mathrm{attribute}_3, \ldots, \mathrm{attribute}_n$ is followed by a class label at a specific instance of time.

- **Recognition**
  - The final step of fusion engine is to recognize the context of real world. So different sort of house hold activities, exercise, meal preparation and taking medicine context will be recognized with confidence values.

- **Overall System Flowchart**
  - Training and testing module flow chart provides the internal components interaction and flow of the process.
  - At the end of the process we get the rule profile along accuracy matrix and component is ready to classify the situation.

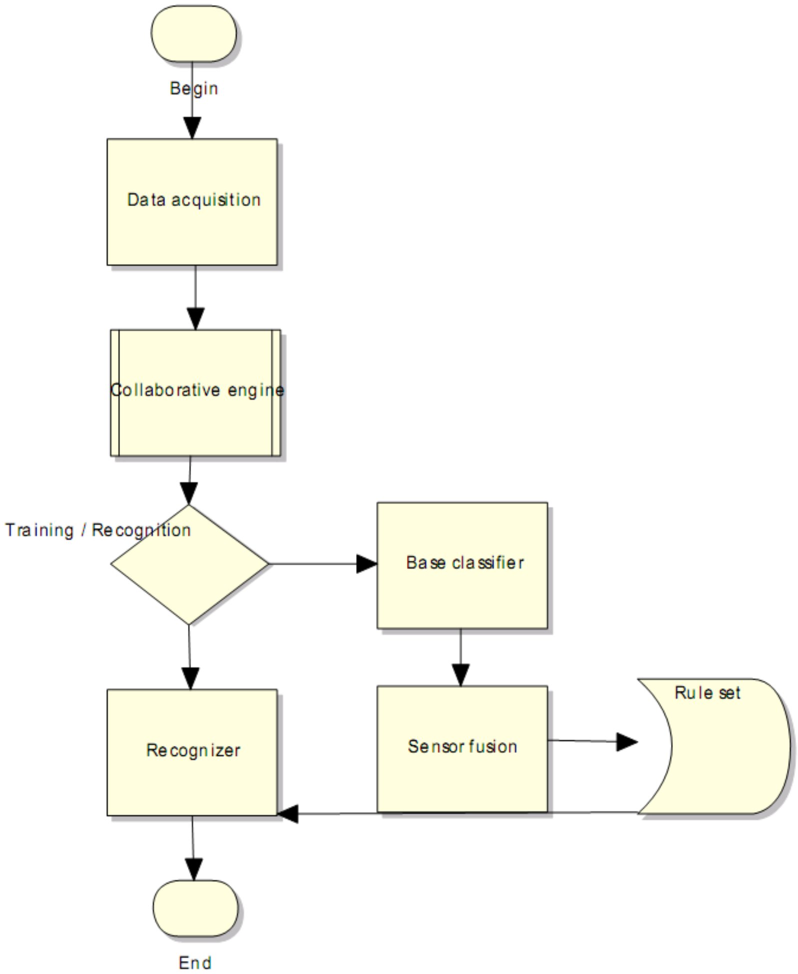


Figure 82 Complete flow chart of Multimodal Sensor Fusion

5.2.1.4. Emotion Recognition

- Emotions are a universal means of communication which can be expressed non-verbally without any language constraints. They are recognized through facial expressions, voice tones, speech and physiological signals.
- Communication through facial expressions plays a significant role in social interactions.
- Over the last decade, automatic facial expressions recognition (AFER) has become an important research area for many applications such as child development, neuroscience and psychology, access control and surveillance and human behavior understanding.

- **Introduction**

- Anytime and anywhere, people have the ability to sense and express emotion, which can help make decisions, handle crises, and maintain relationships. Plentiful words and phrases are used to express moods and feelings. The ability of sensing and expressing emotion in a machine is often a luxury, considered unnecessary and functionless in basic computer intelligence. Programmers hardly encode emotion descriptions into computers. Why are people still trying to give computer emotional ability? [Malika2009].
- When a human user interacts with a computer agent, emotion expressed by the agent affects the emotional response from user. In a study of game agent, if the virtual game agent opponent behaves “naturally" in that it follows its own goals and expresses associated positively or negatively affect behaviors, users will be less stressed than when the agent does not do so [Prendinger2006]. A frustrating interaction with a computer system can also leave a user feeling negative toward the system and its maker [Klein2002], so the consideration of emotion for computers and agents becomes important.
- Before giving computer emotion ability, recognizing human emotion is a preliminary requirement. A computer agent needs to know how a person feels via emotion recognitions, and then the computer agent can perform a reasonable emotional feedback to the person. This feedback of computer is a design of human computer interaction. In the design of human computer interaction human desire and feeling should be considered. Furthermore, it needs to obtain the emotion via contain electronic media before it recognizes such emotion [Malika2009].
- The study of facial expression recognition (FER) has consistently been an active and exigent research area in recent years, which has a significant contribution in many applications such as communication, personality and child development [Bartlett1999], neuroscience and psychology [Mehrabian1968], access control and surveillance [Bettadapura2009] and human behavior understanding. There are three basic modules in FER systems, preprocessing, feature extraction and recognition as shown in Figure 83.

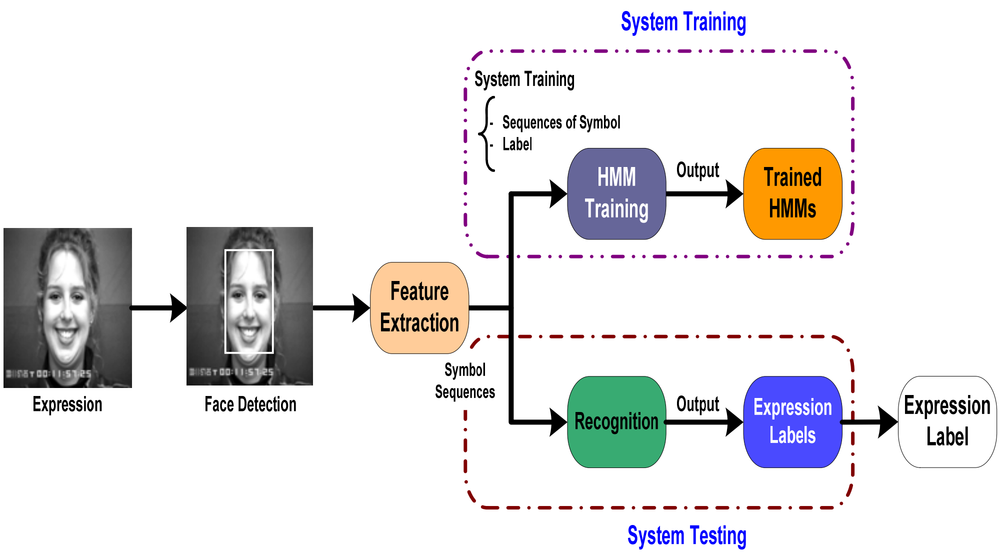


Figure 83 General architecture of video-based emotion recognition system.

- **Problems in Existing Systems**
  - Several video-based emotion recognition systems have been proposed; however, recognizing human facial expressions accurately is still a major concern for most of these systems. This lack of accuracy could be attributed to various causes, such as different light conditions for training and test images, lack of accurate automatic face detection methods, and high similarity among different facial expressions that occurs in the presence of low between-class variance in the feature space.

  - **Feature Extraction.** The accuracy of recognition module is completely reliant on the extraction of good features that means, the effectiveness of the extracted features will also affect the accuracy of recognition module [Huang1997]. Moreover, a good classifier is required to classify the expressions accurately.
  - Many of the existing works have been investigated new algorithms to extract good features. The authors of [Wu2010] have proposed a complete approach for facial expression recognition. They employed canny edge detection and a well-known statistical approach such as Principal Component Analysis (PCA) for features extraction. However, canny edge detection computational wise very much expensive, and some time it is not particularly successful technique because mostly the edges are not continuous and serious edges might be presented due to noise, which might a complex task for canny edge detection. On the other hand, PCA yields uncorrelated components. If the data have a Gaussian distribution, the uncorrelated components are independent. However, if the data are merged of non-Gaussian components, then PCA fails to extract components having non-Gaussian distribution [Buciu2009].
  - Most of the significant facial expressions information can be extracted from lips, nose and eyes known as local features that have an important role in achieving the best accuracy of recognition. Therefore, to extract the local features, one of the higher order feature extraction techniques such as Independent Component Analysis (ICA) [Bartlett2002, Chuang2006, Buciu2003] has been extensively employed for FER systems. The peak state of the facial expression was proposed to recognize the facial expression images; however, the localized features were ignored [Kim2009]. Furthermore, ICA is very weak in managing the inputs. If there are plenty of video frames, exploited as input, ICA does not have the capability to organize it, due to which some time ICA cannot retrieve the desire features.
  - Another method like Local Binary Patterns (LBP) has been exploited by [Shan2009] for feature extraction. In this method a 3 x 3 operator is used in which each pixel is compared with its eight neighbors by subtracting the center pixel value. The resultant positive values are encoded to 1 otherwise encoded to 0. However, the dominant features cannot be extracted by this small 3 x 3 LBA operator. Also it does not provide directional information of the facial frame because it only captures the relations with its surrounding eight neighbor pixels. Moreover, LBP uses first order local patterns and cannot extract more detailed information. Therefore, to solve this problem and to extract the best local features, high order local pattern descriptor such as Local Directional Pattern (LDP) has been presented by [Huang2011] and [Jabid2010]. By using LDP, we can extract best local features that will further be used for recognition. However, computational wise, LDP is much expensive. Moreover, lots of memory size is required for LDP to store the binary codes that is four times bigger than LBP [Jabeed2010].

  - **Recognition.** A large number of classifiers have been employed for video-based facial expression recognition. Among them, the HMM is still the most commonly used method, and it produces accuracies that are comparable to other well-known classifiers such as GMMs, ANNs, and SVMs. In addition, HMMs have their own advantage of handling sequential data when frame-level features are used. In such cases, other vector-based classifiers like GMMs, ANNs, and SVMs, fail to learn the sequence of the feature vectors. However, it is still very difficult for HMMs to represent multiple interacting facial expressions.

- A recent work by Zia et al. [Uddin2009] proposed a complete approach for FER systems that achieved high classification accuracy for the Cohn-Kanade dataset of facial expressions. In their work, they employed well-known statistical methods such as principal component analysis (PCA) and independent component analysis (ICA) for global and local feature extraction, respectively. Furthermore, they used a single-LDA- and a single-HMM-based recognition technique for expression recognition. However, their technique failed to exhibit the same high accuracy when used for other databases, such as the Cohn-Kanade dataset (87%), Japanese (JAFFE) dataset 83%), and AT&T dataset (72%) of facial expressions. Low accuracy in these new experiments could be attributed to the following three reasons. First, most of the expressions share high similarity and thus overlap significantly in the feature space, as shown in Figure 84. Zia et al [Uddin2009] applied a single LDA with the assumption that this variance is distributed uniformly among all the classes in the feature space. However, this is not the case. For example, expressions like happy and sad are very similar to each other but can easily be distinguished from anger and fear (another pair with high similarity).

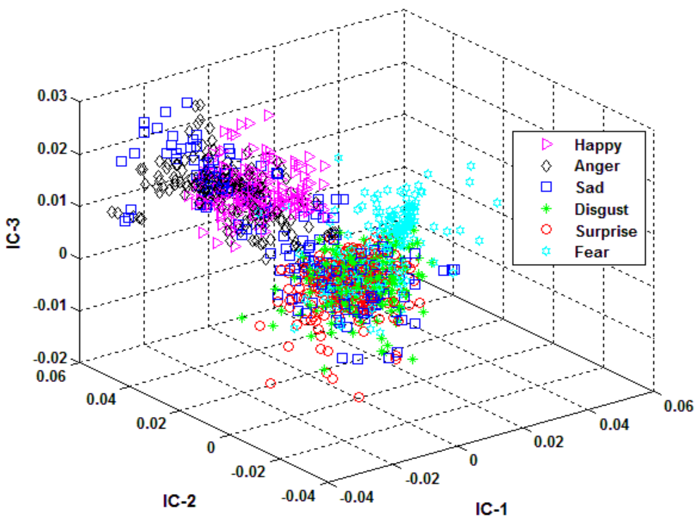


Figure 84 3D-feature plots for six different types of facial expressions. The features are highly merged, due to the presence of similarity between the expressions, which could later results in a high misclassification rate

- **Requirements**
  - To overcome the limitations of the existing systems and to provide better decision on facial expression recognition in healthcare domain, there are different requirements that

need to be satisfied.
- Information coming from video (2D) camera.
- To store the information (i.e., Facial expressions) coming from input source, we need to decompose it into frames and then store them for decision making.
- For the system to work properly, smooth communication among all the components is required.
- The need is to develop an automatic face detection and extraction technique.
- For knowing the expression, a recognition engine is required that will facilitate the doctors to find the aggressiveness of patients in healthcare environments.

▪ **Proposed Solution**
- **Preprocessing.** Accordingly, the purpose of this study was to propose an accurate and robust facial expression recognition system capable of exhibiting high recognition accuracy. The proposed system is based on the idea of a hierarchical recognition approach, and it was tested using three publicly available datasets. Each dataset consisted of six facial expressions: happy, sad, surprise, disgust, anger, and fear. Each expression was performed by different people. Mostly, the image data in these datasets contained frontal views of the face, and each expression was composed of several sequences of expression frames.
- In these datasets, facial expressions were recorded under different light conditions, which could reduce the recognition accuracy. To solve this problem, a method based on global histogram equalization was proposed. This method reduces such noise by increasing the dynamic range of the intensity using the histogram of the whole image. The proposed method finds the running sum of the histogram values and then normalizes it by dividing it by the total number of pixels. This value is then multiplied by the maximum gray level value and then mapped onto the previous values in a one-to-one correspondence. The proposed method outperforms previously used techniques, such as histogram equalization (HE) and local histogram equalization (LHE) that produce unwanted artifacts and washed-out images. Furthermore, LHE causes over-enhancement and sometimes produces checkerboards of the enhanced image; therefore, several studies suggested avoiding the use of these methods.

- **Face Detection.** As stated earlier, an accurate facial expression recognition system requires automatic face detection. Thus we proposed, a novel technique that automatically detects and extracts faces from a given image using two methods: gray-level- and skin-tone-based methods.

- **Feature Extraction & Selection.** We employed the well-known statistical approaches, PCA and ICA, to extract local and global features, respectively. Since most expressions share high similarity, their features overlap significantly in the feature space. This can result in the presence of very low between-class and high within-class variances in the feature space, which in turn can lead to low recognition accuracy. Numerous methods have been presented in the machine learning literature to solve this problem, such as LDA. However, our experiments showed that applying LDA directly to the whole feature space failed to resolve the overlap or low between-class variance among facial expressions. This failure could be attributed to the fact that LDA is a linear technique, which limits its flexibility when applied to complex datasets. Moreover, the assumption made in using LDA that all classes share the same within-class covariance matrix is not valid in this case.

- **Hierarchical Scheme-based Recognition** To overcome this problem, we proposed a hierarchical recognition scheme that is a combination of PCA, ICA, LDA, and HMMs. Based on the parts of the face that create an expression, expressions were divided into three categories: lips-based, lips-eyes-based; or lips-eyes-forehead-based expressions. At the first level of the proposed hierarchical recognition scheme, LDA was applied to the features (PCs and ICs), and the resulting LDA features were fed to a single HMM to recognize the category for the given expression (as shown in Figure 85). Once the category of the given expression was determined, the label for the expression within the recognized category was recognized at the second level by again feeding the features to a combination of LDA and HMM, trained specifically for the recognized state as shown in Figure 85.

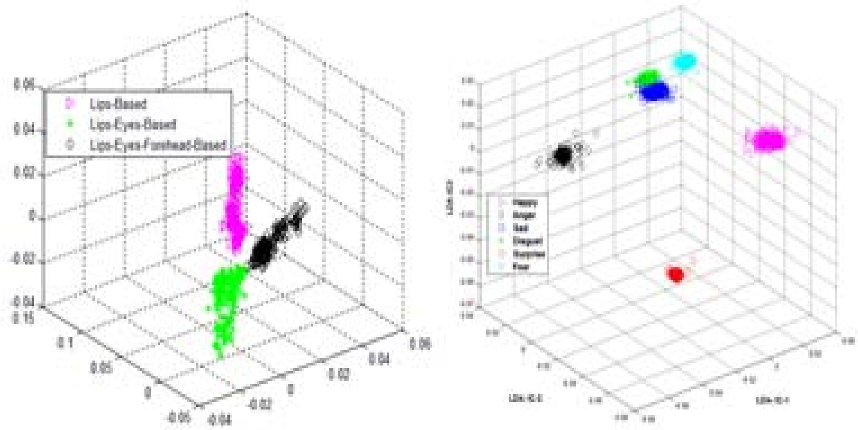


Figure 85 3D feature plots for the three expression-categories after applying LDA at the first level. (b) 3D-feature plot of the proposed hierarchical recognition scheme for six different types of facial expressions after hierarchical LDA.

- **Overall System Flowchart**
  - The overall flowchart for the proposed system at the training stage is shown in Figure 86.

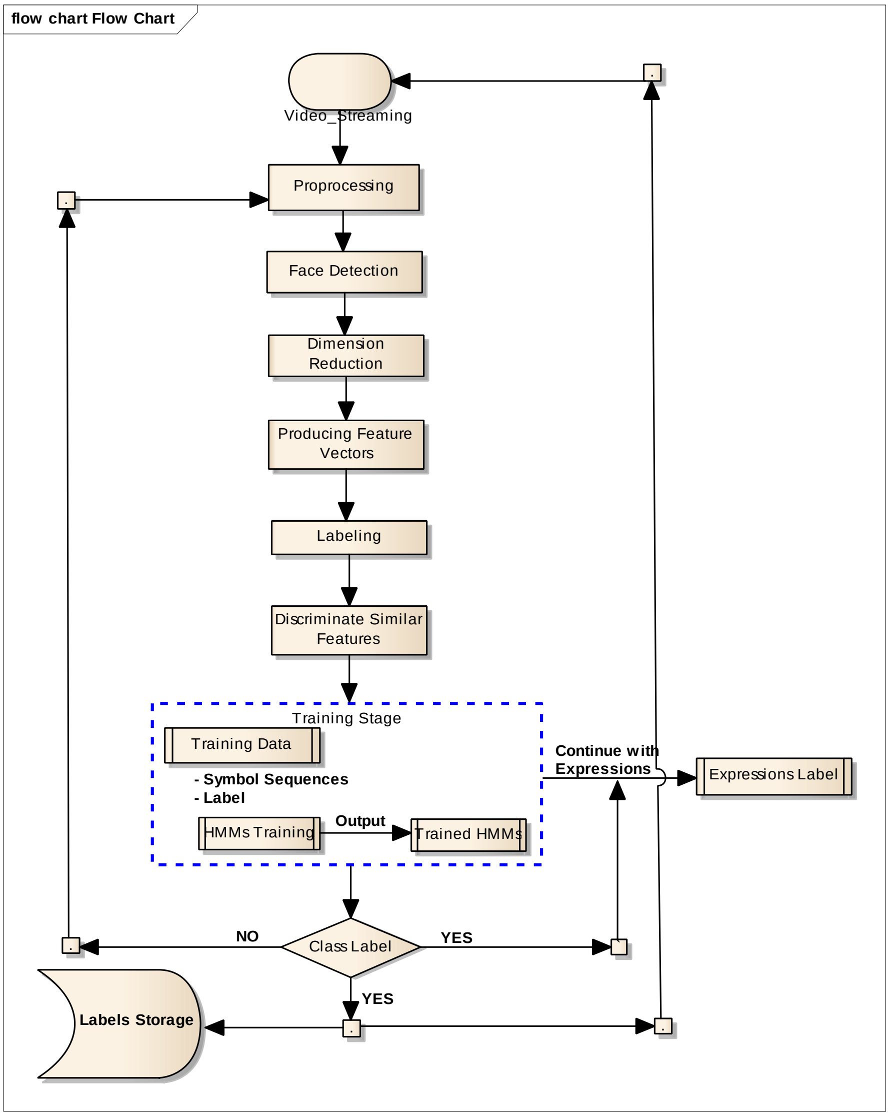


Figure 86 A video-based emotion recognition system

5.2.2. Context-aware Activity Manipulation Engine

- Semantics is the meaning of a resource in the context in which it is referred while these meanings are provided by using modeling languages like rdf(s) and OWL.
- Modeling information in a formally structured and canonical representational format is called as ontology. So ontology is basically the representation of information combined together using some modeling language.
- LDSS for Chronic Disease takes the input form CAME engine as well.
- Context-aware Activity Manipulation Engine (CAME) is one of the main components of SC3 and also an inessential component of LDSS for Chronic Disease.
- CAME is the process of inferring high level activities from low level activities recognized by different sensors.

- **Introduction**
  - Use of ontology provides basic description to events and activities, and from this information we can deduce new knowledge.
  - Use of ontology in activity recognition is relatively a new area of research. Using ontology help us better understand the activity in a given context.
  - Activities recognized with the help of different sensors (i.e., body, location, motion, and video sensors) are low level activities and they are not in a capacity to be used for certain types of analysis and decision making.
  - With the help of ontology, where we use the context information and link all the related activities in a chain, then with the help of customized rules we get the higher level activities that are more usable for decision making.
  - Ontology helps in properly extracting the higher level activity of a set of activities in a series, e.g., series of low level activities like bending, sitting, jumping and walking with the help of ontology will result in a higher level activity i.e., exercising.

- **Problems in Existing Systems**
  - In today's society, people are interested in their improved healthcare. Hence, a challenge exists to both provide and maintain the quality and availability of healthcare services with a minimum cost [Khattak2010].
  - Various wireless technologies have been used for improved healthcare services. CodeBlue [Gao2008] supports physicians and nurses to monitor patients.
  - Reminder based systems which have primarily focused on plan-based approaches to decide how and when to prompt subjects effectively [Mama2000] in addition to offering location-based reminders [Sohn2005]. Nevertheless, the context for the delivery of reminders involves more information than simply location or time.
  - HYCARE [Du2008] system takes context in consideration and develops a schedule for reminder based services with a goal of avoiding possible conflicts.
  - In [Wang2009], an ontology is used to incorporate context related information for intelligent processing; however, the system developed was more of a homecare system rather than a healthcare system.
  - Work presented in [Chen2012] details an ontology based reminder system; incorporates rules for manipulating the recognized activities of the elderly.
  - In [Khattak2011], real-time activities are recognized using diverse sensors are used for situation analysis.
  - Research in [Fitma2011] focused on the social interaction of patients. Based on the expe-

riences shared, it generates intelligent service recommendations and eliminates their isolation.

- Existing systems aiming to offer a form of support are mostly based on one input modality and in some cases use imperfect contextual information [Henr2004]. These solutions subsequently do not consider the integration of activity from diverse input modalities.

▪ **Requirements**

- To overcome the limitations of the existing systems and provide healthcare and smart homes services, there are different requirements that need to be satisfied.
    - Information coming from diverse sources like motion sensor, accelerometer, video sensor, and location sensor.
    - To store the information coming from input modalities, we need to store them. For the storage structure, a storage model is designed for the knowledgebase. OWL is used for the storage.
    - For the system to work properly, smooth communication among all the components is required.
    - The need is to design and model the rules for developing the filtering system.
    - For Context Analyzer, a reasoning engine is required that will facilitate the match making process.

▪ **Proposed Solution**

- CAME is one of the main components of SC^3. SC^3-CAME is the process of inferring high level activities from low level activities recognized by different sensors.
- The component based framework architecture diagram of CAME and the information flow is given in Figure 87, while the detail description of all the components are given in their corresponding sections.

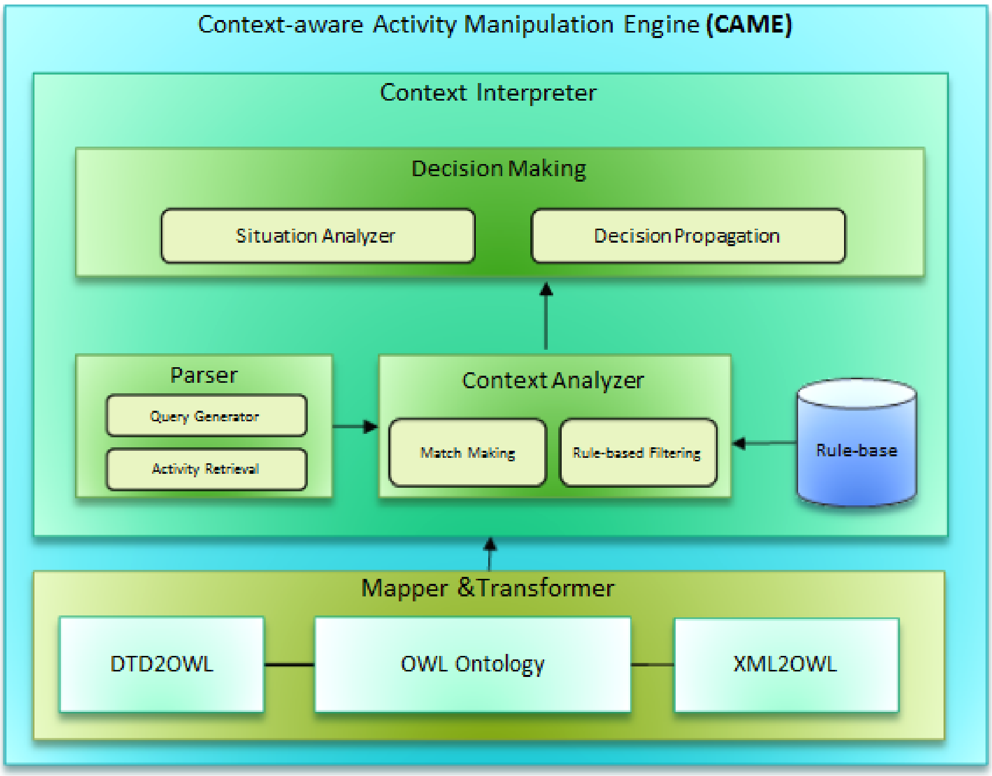


Figure 87 Overall architecture of Context-aware Activity Manipulation Engine (CAME)

- **Mapper & Transformer**
  - Mapper & Transformer is responsible for recognizing data inputs from diverse sources. Video-based, sensor-based, motion-based, and location-based activity recognition engines will provide output in different formats like XML (example given below in Figure 88) and simple text.
  - So there is need of recognizer and convertor engine that will convert the simple representation of activity into OWL representation. The OWL (N3) representation of the activity is given in Figure 89.
  - The three subcomponents are: 1) DTD2OWL, 2) OWL Ontology, and 3) XML2OWL.

```
<?xml version="1.0" encoding="UTF-8"?>
<activities>
  <activity type="Motion">
        <detectedBy>Motion Sensor</detectedBy>
       <hasName>Prof. SY Lee</hasName>
        <activityName>Entering Class</activityName>
       <id>345</id>
       <time>2009:06:14:14:00:13</time>
  </activity>
  </activities>
```

Figure 88 XML representation of activity information produced by motion sensor.

```
activityOnto:Activity_Instance_20090614140013345
a           activityOnto:Activity ;
activityOnto:hasConsequentAction   activityOnto:Action_Instance_145413546;
activityOnto:hasID                               345;
activityOnto:hasName                      "Entering Class";
activityOnto:hasType                                  "Motion";
activityOnto:isA                          activityOnto:Room_Instance_Class;
activityOnto:performedAtTime              2009:06:14:14:00:13;
activityOnto:performedBy                       activityOnto:Person_Instance_345.
```

Figure 89 OWL representation (N3) of activity detected using motion sensor.

- **Knowledgebase**
  - Knowledgebase (KB) serves as the back bone of CAME.
  - It is responsible for proper communication of information among all the components of CAME.
  - It stores all the possible types of activities that a human body can perform in different context/situation, with the information of different activities priority for different users and group of users.
  - The proper engineering of the KB is most important activity in the development of CAME.
  - To engineer the KB (see Figure 90) we have to look at the same problem from different dimensions.
  - When an activity is recognized by the sensors then this knowledgebase is parsed and if required in different situations then inferencing is done for decisions against activities.
  - So it is actually the ontology, where all the activities are semantically modeled and available for analysis and decision making.

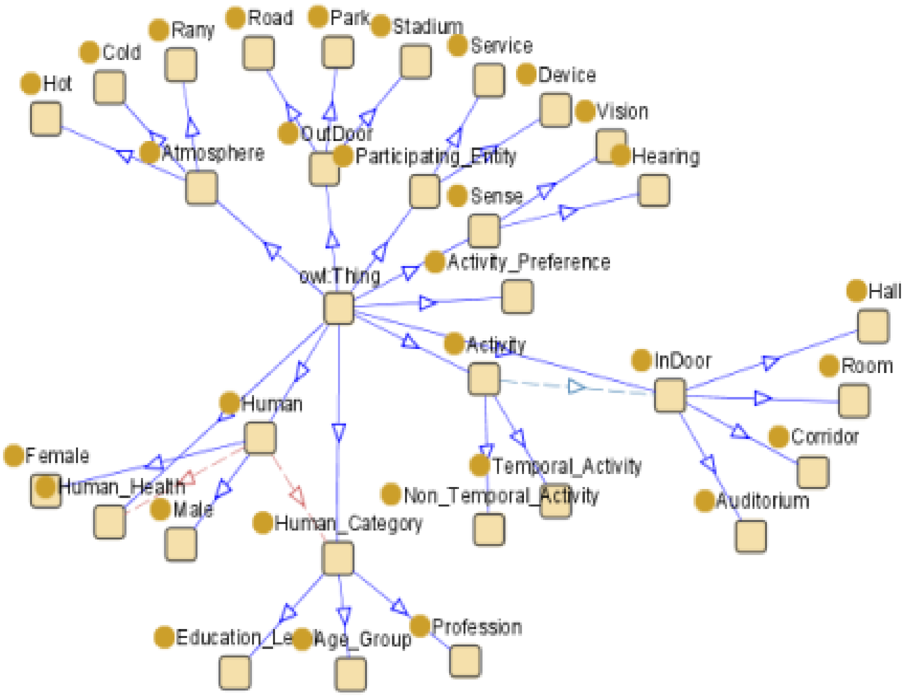


Figure 90 Knowledgebase (Human Activities Ontology)

- **Parser**
  - For any type of information manipulation from the Knowledgebase, Parser is responsible to properly handle all the operation regarding that matter.
  - The Parser normally communicates with Activity Representation component to properly represent the activity, it also parse the Knowledgebase for the Context Analyzer for verity of different reasons like verification of activity and Match Making.
  - To populate the KB for newly recognized activity, the Parser is also used in that case.

- **Context Analyzer**
  - To understanding the context of an activity and to extract high level (abstract) activities from low level activities recognized by sensors, we need to have a Context Analyzer for analysis of these activities and to make proper situation analysis.
  - The Context Analyzer is very important component of CAME. It uses the activities information with respect to their context information and infers high level activities. For this purpose Match Making is activated.
  - The decisions or suggestions of Context Analyzer are very much dependent on the domain and user intensions.
  - So for this reason we also introduced the user defined customized rules in the Context Analyzer. So the rules are used in Rule-Based Filtering to filter out the unintended activities information.

- **Rule Base**
  - Every organizations have their own customized rules. These are used to implement the organization’s policies.
  - So for these sorts of situations and actions, we need to define customized rules for different activities.
  - The rules are shown in Figure 91.

Rule1
∃Activity(a1) ⊓ ┘hasContents(taking medicine) ⊓ hasNextActivity(a2) ⊓ ∃Activity(a2) ⊓ hasContents(eating) → Activity.Create(a1) ⊓ Activity.Create(a2) ⊓ reminder(take medicine)

Rule2
∃Activity(a1) ⊓ hasContents(reading) ⊓ hasNextActivity(a2) ⊓ ∃Activity(a2) ⊓ hasContents(TV On) → Activity.Create(a1) ⊓ Activity.Create(a2) ⊓ turnOff(TV)

Rule3
∃Activity(a1) ⊓ hasContents(unknown activity) ⊓ hasNextActivity(null) → Activity.Create(a1) ⊓ reminder(movements are wrong)

Rule4
∃Activity(a1) ⊓ hasContents(entering kitchen) ⊔ ∃Activity(a2) ⊓ hasContents(entering bedroom) → Activity.Create(a1) ⊔ Activity.Create(a2) ⊓ turnOn(lights)

Figure 91 Customized rules for rule-based filtering

- **Decision Making**
  - After the process of context analysis, the system can take decisions or give suggestion against different activities.
  - So this module is responsible for performing some actions against the suggestions made by the Context Analyzer.
  - This module has two sub modules: 1) Situation Analyzer that analyzes the situation and stores the high level activities in Life Log for later usage. 2) Decision Propagation module instruct different hardware or connected systems for appropriate actions to be take based on the analysis.

- **Overall System Flowchart**
  - The overall CAME system flow chart with all components working is given in Figure 92.
  - This shows that how different components of CAME work together and how the output of one component flows to the other component where it is used as input for further processing.

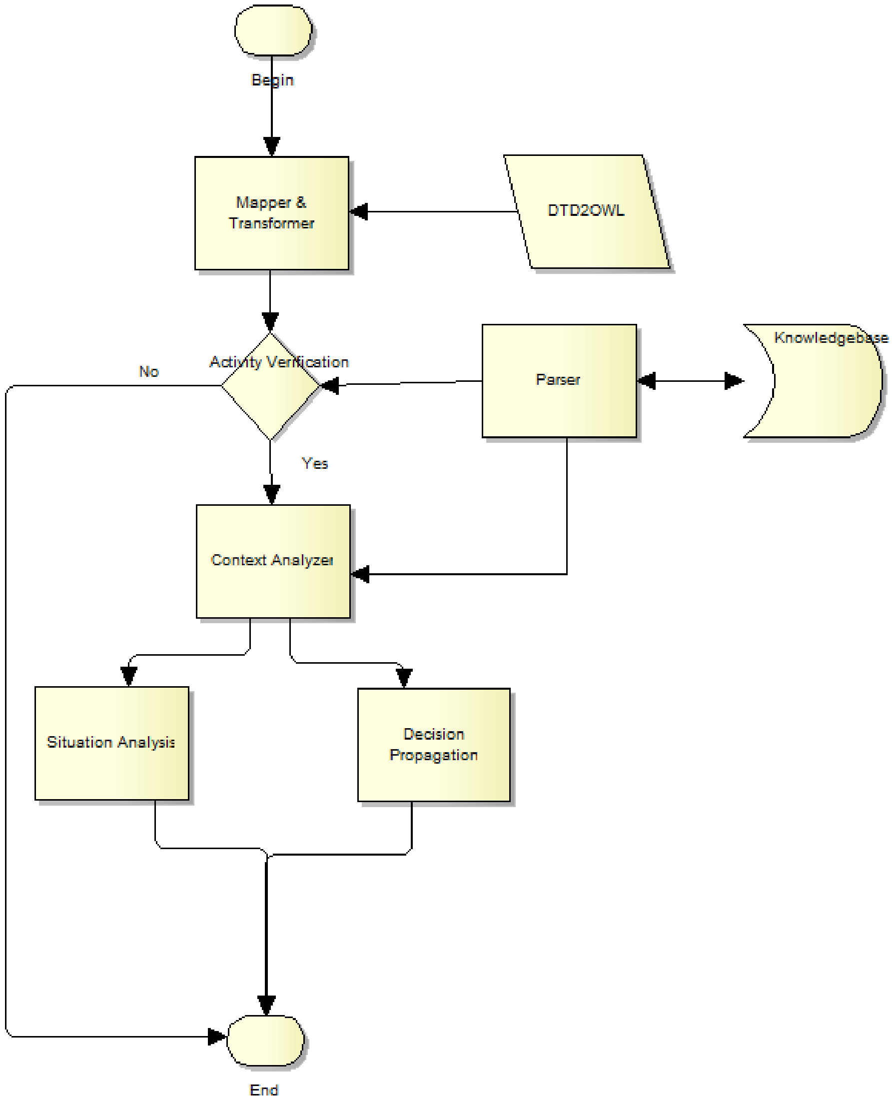


Figure 92 Flow Chart for overall working

5.2.3. Social Media Interaction

5.2.3.1. Tweet Analysis

- **Introduction**
  - By monitoring person's social activities, interest and emotions can be extracted, help to provide personalized services to person.
  - System monitors user stream from Twitter and process tweets to extracts user interest and sentiments from tweets.
  - Our proposed system integrates as a plug-in application, while extracting user related information including profile information, person interests and emotions.
  - Analyzing the interest, behavior and lifestyle of person provides assistance in better decision making and personalized services.

- **Problems in Existing Systems**
  - Twitter data must be collected to use it for research purpose. Different analysis tools are available to collect twitter public data. Archivist is one of the tool to collect tweets.
  - Grabeeter is another tool to get individual personal and public tweets and store them for future use. Twitter data is available for mining unstructured data and analyzing hidden patterns in it. J. Chen et al [Chen2010] introduced a system for URL recommendations on Twitter using data stream technique.
  - The system was based on content sources, topic interest models, and social voting to design URL recommender and compare different recommender techniques.
  - Fabian Abel et al. [Abel2011a, Abel2011b, Abel2011c] analyzed user modeling on Twitter for personalized news recommendation and enrich news with tweets to improve the semantic of Twitter activities. The work used methods including topic, entity, and hashtag based to analyze the user modeling.
  - It also focused on temporal pattern extraction in user profile. Ilknur Celik et al. [Celik2011] studied semantic relationship between entities in Twitter to provide a medium where users can easily access relevant content for what they are interested in.
  - Eleanor Clark et al. [Clark2011] introduced a system to apply text normalization for Twitter. System categorized errors and irregular languages used in casual English of social media into different groups and then applied natural language processing techniques to correct common phonetic and slang mistakes in tweets.
  - Tetsuya Nasukawa et al. [Nasukawa2003] used natural language processing techniques to identify sentiment related to particular subject in a document.
  - They used Markov-modal based tagger for recognizing part of speech and then applied statistics based techniques to identify sentiments related to subject in speech.
  - Jeonghee Yi et al. [Yi2003] presented a model to extract sentiments about particular subject rather than extracting sentiment of whole document collectively.
  - This model proceeded by extracting topics, then sentiments and then mixture model to detect relation of topics with sentiments. Whereas Namrata Godbole et al. [Godbole2007] introduced a sentiment analysis system for news and blog entities.
  - This system determined the public sentiment on each of the entities in posts and measured how this sentiment varies with time. They used synonyms and antonyms to find path between positive and negative polarity to increase the seed list.
  - Andranik Tumasjan et al. [Tumasjan2010] analyzed Twitter as a source of predicting elections. They used the context of the German federal election to investigate whether

Twitter is used as a forum for political deliberation. They used LIWC2007 [Liwc2012], a text analysis software developed to assess emotional, cognitive, and structural components of text samples using a psychometrically validated internal dictionary.
- They focused on 12 dimensions in order to profile political sentiment: Future orientation, past orientation, positive emotions, negative emotions, sadness, anxiety, anger, tentativeness, certainty, work, achievement, and money.
- Bernard J et al. [Jansen2009] performed analysis of Twitter as electronic word of mouth in the product marketing domain. They analyzed filtered tweets for frequency, range, timing and content.
- They found that 19% of a random sample of tweets contained mentions of a brand or product and that an automated classification was able to extract statistically significant differences of customer sentiment.
- Similarly Archivist [Archivist2012] is a service that uses the Twitter Search API to find and archive tweets. It helps to have a look at trends such as frequency of Tweets over time, top users and words and many more. It also helps user to get real time trend information on Twitter.
- Jeff Clark used Venn diagram to show illustrating pattern in twitter data [Clark2008]. By using Venn diagram he explores overlap of different topics with each other in tweets. Collecting tweets alone and analyzing them for sentiments just on the available keyword is not enough to understand the real semantics of tweets.
- There is a need to precisely parse and process the tweet for the contained knowledge.

- **Requirements**
  - Semantically processing of Natural language from Twitter to extract user interest and sentiments to build user profile. This profile would be used to provide personalized services.
  - Classification of user interest into different categorizes to enhance system efficacy in providing domain specific services.

- **Proposed Solution**
  - The proposed system architecture consists of four main components. Data Manager, Knowledge Generator, architecture of whole system is shown in Figure 93.
  - Knowledge Enhancer, and Filter Engine. These components are elaborated as follows.

  - **Data Manager**
    - Data Manager is our plugable application that interacts with Twitter. It consists of the following subcomponents.
    - **Data Fetcher**: Data Fetcher sends request to Twitter for stream of user. The fetched data is in JSON format [Archivist2012] that is a lightweight data-interchange format.
    - **Data Processor**: Fetched data requires some pre-processing before analyzing. Data Processor converts data in required useable format. It also removes user slangs from tweets.

  - **Knowledge Generator**
    - Our purpose is to extract valuable information hidden in tweets and build user profile. Twitter data collected by our system is given to Alchemy API.
    - It accepts unstructured text, processes it using natural language processing and machine learning techniques, and returns keywords and sentiments of users about key-

words. We extract participating keywords and sentiments associated with those keywords.

- After extraction of knowledge, all tweets, participating keywords, and associated sentiments are stored in the repository for further processing as discussed below. However, the information extracted by knowledge generator is of low precision.
- So, it needs further processing to better identify user interest and analyze sentiments of domain specific interest.

- **Knowledge Enhancer**
  - Knowledge enhancer module add additional knowledge which was not extracted as keyword by Alchemy API.
  - The proposed system uses part of speech tagging and entity extraction on tweets and then adds additional data in the knowledge, extracted by Alchemy API.
  - Entity extraction by using Alchemy API helps by extracting entities, not extracted as keyword.
  - The proposed system have been tested with addition of subjects, verbs, objects, and entities in knowledge; however, just addition of verb and entities increases information collected from tweets.

- **Filter Engine**
  - For classifying tweets into different categories on the basis of knowledge extracted from tweets, the proposed system applies filtering on the extracted data. The filtering process is domain specific.

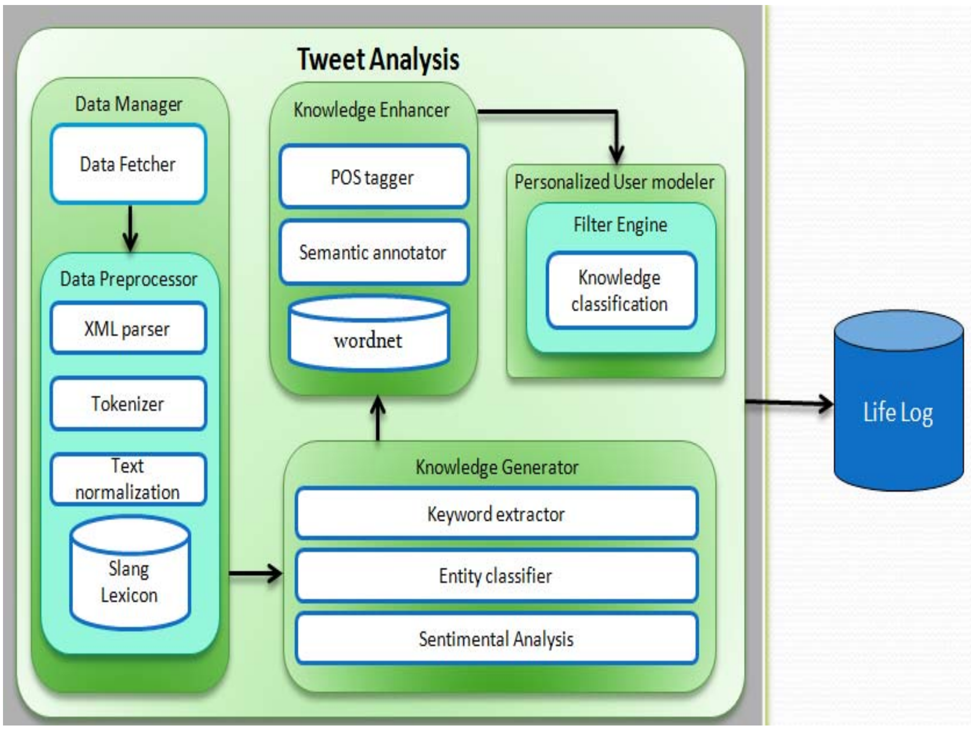


Figure 93 Overall Flow of Tweet Analysis

- **Overall System Flowchart**
  - The overall system flow chart with all components working is given in Figure 94.
  - This shows that how different components work together and how the output of one component flows to the other component where it is used as input for further processing

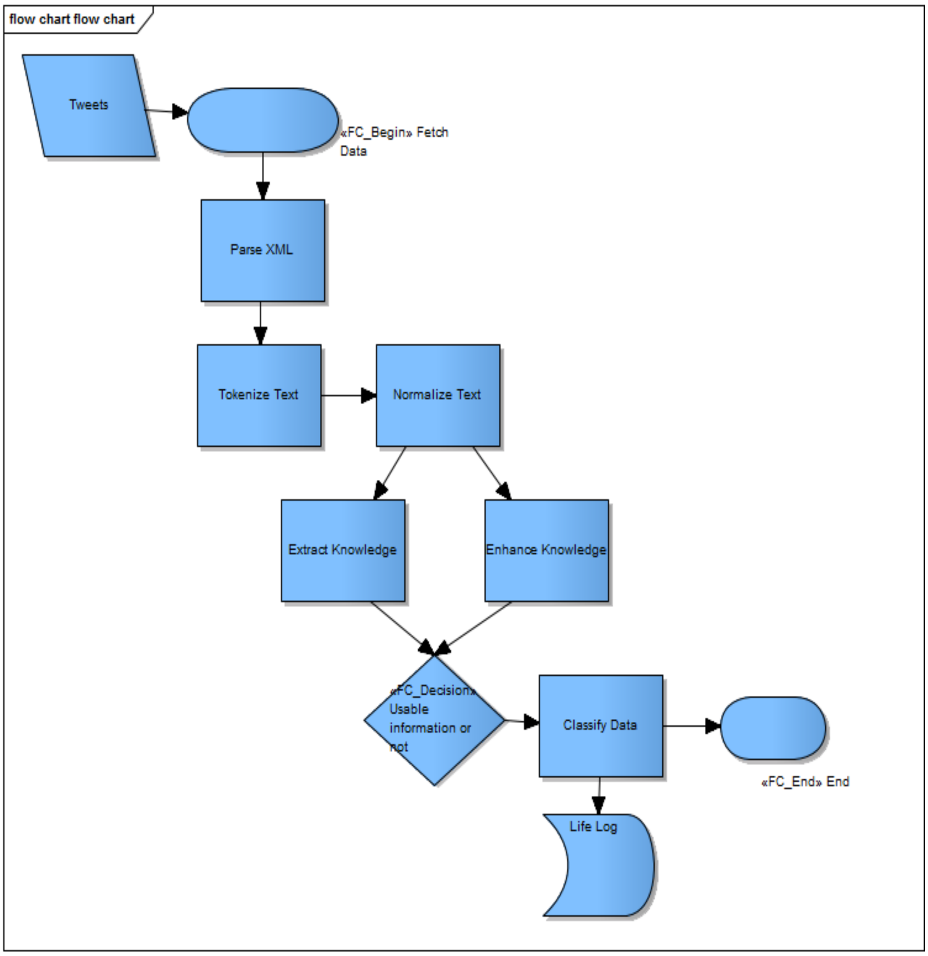


Figure 94 Flow chart of Tweet Analysis

5.2.3.2. Trajectory Analysis

- **Introduction**
  - GPS is a commendable technology to find location related activities. Presently a huge

number of devices are enabled with this technology and getting more common with rapid speed. Currently more of these technologies are used for advancement of overall society and mankind. Like how to control traffic in a better way, finding peak and low rush hours and movement behavior of people of a particular area.
- Our proposed system is incorporating trajectory mining techniques to a person's direct life and particularly its usage in healthcare domain is targeted. A GPS based real time activity monitoring system is developed which is used for tracking daily life routine of a user. Further these patterns are compared with the patterns prescribed by physicians and also recommendations for the better carrying out of prescribed schedule are presented to the user.

- **Problems in Existing Systems**
  - A recent study used trajectory information of people for finding people attractive areas and their related movement patterns, which can lead to instructive insight to transport management, urban planning and location-based services (LBS).
  - They considered taxi as an important mode of transport and acquired road traffic condition, travel patterns, average speed estimation and attractive places where people often visit [Yue2009].
  - In [Coll2011] the trajectory mining is used for mining ship spatial trajectory and an Automatically Identification System (AIS) is developed as a result of this study in which GPS enabled technology is used for finding the paths of ships [Coll2011].
  - The basic purpose of this study was self-navigation and collision avoidance but it can be extended for better marine traffic management and distribution. But both of these studies cannot be applied to human life because of the limitations and restriction involved like the most appropriate device for tracking and recording all of this information is mobile phone, which has very limited computation power and storage space.
  - Correspondingly all of these parameters are not required when we discuss about human life as we only are interested in daily routine activities and its positions so we have modified our approach accordingly.

- **Requirements**
  - GPS enabled smart phone with the availability of internet at each imperative location is required to fetch all the required information. Further processing of this data involves the Google API for conversion of GPS coordinates to Geo tags.

- **Proposed Solution**
  - The proposed method is a healthcare service which monitors user's routine activities and assists to follow prescribed schedule. Detailed architecture of the proposed system is shown in Figure 95. As shown in the figure that architecture is divided into 3 main processing module, Data Preprocessor, Schedule Manager and Activity Manager. Each of these modules are discussed below in detail.

  - **Data Preprocessor**
    - GPS coordinates of the user's position are fetched by using a GPS receiver, after a minimum time interval T_{min}. After that each of these coordinates are sent to Data preprocessor. The main module of data processor, imperative location finder confirms the significance of the position. Two main parameters of time and distance are used for conformance of imperative location. Position and corresponding activity is only treated as imperative if both of these thresholds are satisfied.

- Then these coordinates of imperative location are sent to Geo tag transformer where Google API is used to convert it to Geo tags. Semantic tag of the imperative location is also fetched from the user for the acquiring contextual information about the location.

- **Schedule Manager**
  - After preprocessing of data, all of the information is sent to the schedule manager for further processing. First of all semantic tag mappers plays its role of mapping all of the corresponding information and sent it to the repository for the storage. Followed schedule of the user is stored it followed patterns.
  - Prescribed schedule of physician is stored in prescribed patterns and detail of each of activity in prescribed schedule is stored in the recommendations. Activity analysis is the main part of our proposed system with the purpose of comparing both followed and prescribed schedule. Inconsistencies of these schedules are shown to the user and physician as well for further improvement of the daily routine.

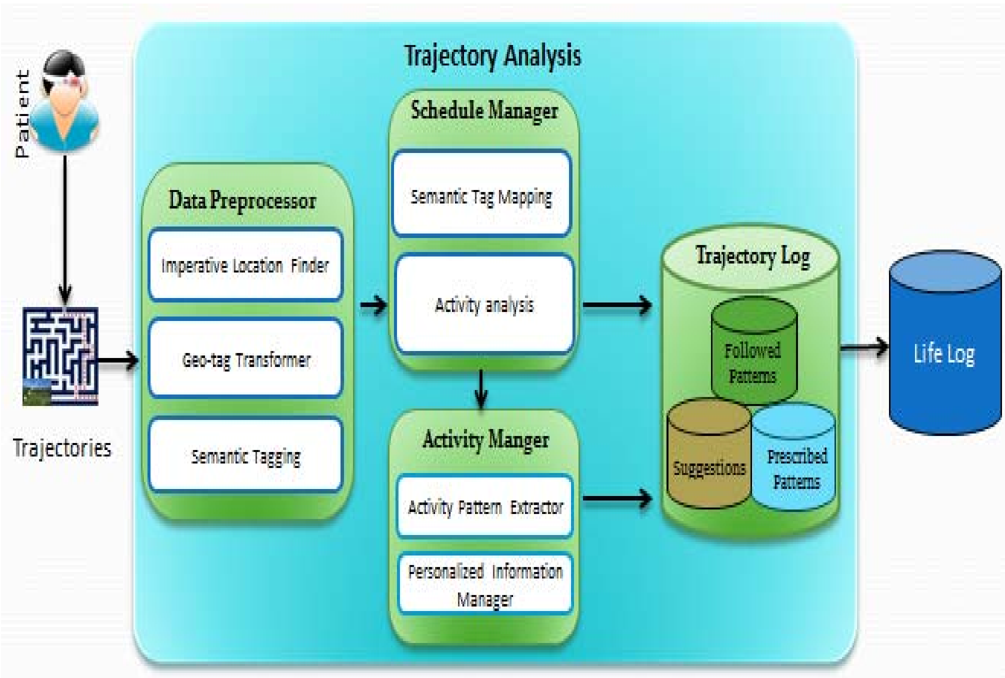


Figure 95 Trajectory1 Main Architecture of Trajectory Analysis

- **Overall System Flowchart**
  - The overall Trajectory Analysis system flow chart with all components working is given in Figure 96 Overall System Flow for Trajectory Analysis. This shows that how different components of interaction analysis work together and how the output of one component flows to the other component where it is used as input for further processing.

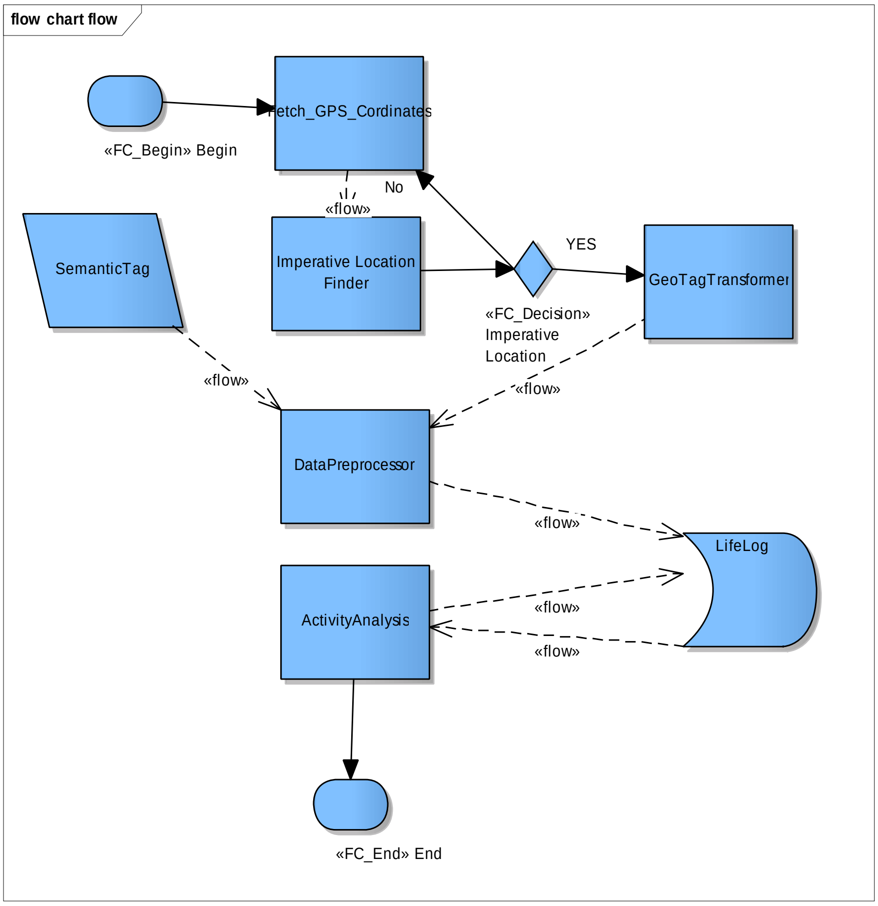


Figure 96 Overall System Flow for Trajectory Analysis

5.2.3.3. Interaction Analysis

- **Introduction**
  - In this module, we aim to improve the users' health by utilizing his social interaction in order to suggest them appropriate lifestyle patterns.
    - For instance, after observing a user's daily routines, our proposed interaction analysis is able to finds some complications with his lifestyle.
    - He/she usually sleeps late; does not exercise regularly; does not take eats on time. Obviously, these lifestyles are not good for healthy living.
  - The proposed method is integrated through lifelog in the behavioral analysis module which is under development at UC Lab.
  - The lifelog will take the information and integrate it with patient demographic to facilitate the behavioral analysis and suggest changes in unhealthy life patterns through better way.

- **Problems in Existing Systems**
  - The main motivation of our work is that time varying interaction data is collected in very diverse settings which need social network analysis to identify the meaningful information.
  - The network analysis has been used in a variety of fields to analyze the huge amount of data such as the Internet [Faloutsos1999], animal behavior [Fischhoff2007] [Sundaresan2007], e-mail habits [Chapanond2005, Diesner2005, Lahiri2010], mobile phone usage patterns [Nanavati2006], co-authorship patterns in research publications [Barabási2001, Newman2001], Dominance behavior and social association patterns of the animals [Juang2002].
  - In this module we analyzed the email interaction data with intensions to extract information that can facilitate the LDSS system in taking the more meaningful decisions.
  - We analyzed the recurring patterns correspond to seasonal and other recurrent association patterns. The similar work is done for analyzing the human behavior with location-aware cellphones [Eagle2006].
  - We applied it on email interaction of from daily living of users. We are interested to identify the typical periodicities and specific interaction patterns that may affect the patients' health

- **Requirements**
  - In this module we mine the patients' frequent and periodic interaction patterns that change over time.
  - The purpose is to gain knowledge about the preferences, needs and habits of the user.
  - Users can act in two different roles: senders and receivers. These two roles are not interchangeable while mining the patterns of interest from his daily interaction routine

- **Proposed Solution**
  - We propose a two-phase strategy to identify the hidden structures shared across different dimensions in dynamic networks such as type of interaction, time of interaction, interaction intervals and interaction response based on priorities.
  - We extract structural features from each dimension of the email network via periodic and frequent interaction mining, and then integrate them to find out robust patterns about patients as shown in Figure 97.
  - Furthermore, with the right formal definition of what constitutes periodic behavior, the aggregate periodicities of an entire set of mined interaction patterns can assist LDSS in better decision making.
  - Therefore, learning patients' common behaviors becomes an important step towards allowing LDSS to provide personalized services more accurately and effectively.

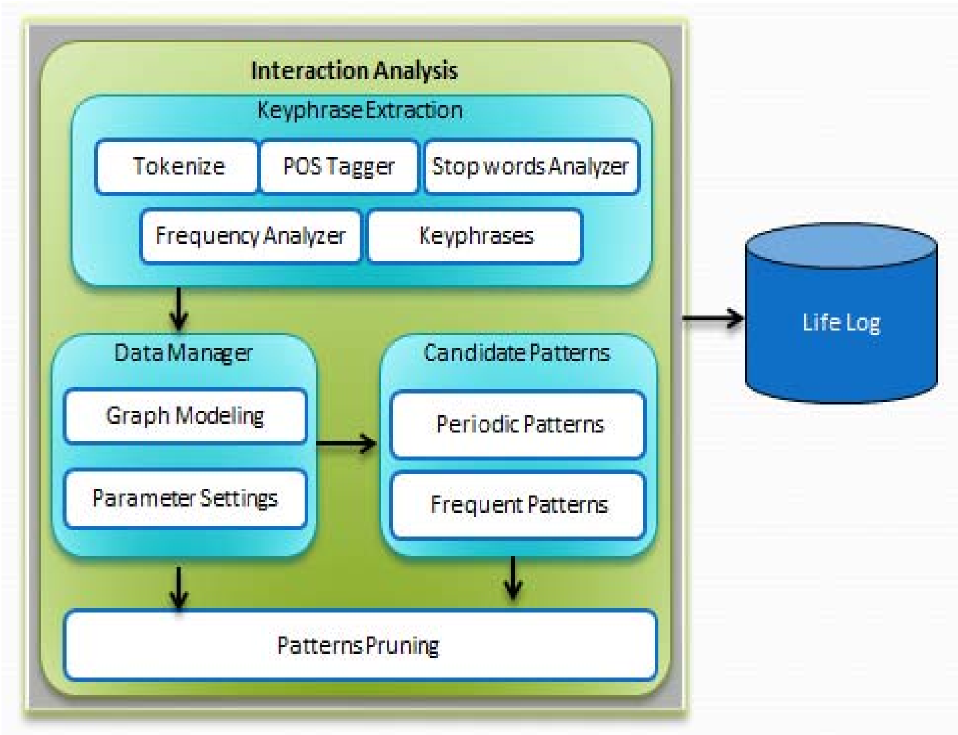


Figure 97 Overall Architecture for Interaction Analysis

- **Keyphrase Extraction**
  - The keypharses are extracted from the contents of the email by using the Algorithm shown in Figure 98.
  - Extraction of the semantic keyphrases is the essential requirement of the accurate data modeling with the user interaction. First of all parameters of the extraction algorithm KEA++ [KEA] are set with respect to keyphrases' length in the taxonomy and length of the documents. Secondly train KEA++ on the set of Emails using taxonomy.
  - Then apply KEA++ on actual Emails (data). First email contents are tokenized by using POS tagger and stop words analyzer. The frequency of each word is counted in the email and then KEA++ return the relevant keypharses.
  - The keyphrases returned by KEA++ is processed to get its level label in the taxonomy. Identify level labels is required before applying the refinement rules because they represent the hierarchical order of the keyphrases.
  - If the KEA++ result has training level keyphrases then these training level keyphrases are retained in the result set as shown in steps 5 to 12 of Algorithm 1.
  - Lower level keyphrases are stemmed to their training level keyphrases and kept in the result set if they are associated with the general category at the lower level in taxonomy. Otherwise lower level keyphrases are discarded.
  - Upper level keyphrases are discarded after identifying and preserving their equivalent keyphrases from taxonomy which belong to the same level of training level

keyphrases.
- If the initial result does not contain any training level keyphrases then lower level keyphrases of the result are preserved and added in the final refined result.
- Upper level keyphrases are discarded after identifying and preserving their equivalent keyphrases from taxonomy which belong to the same level of training level keyphrases.
- This process is executed from steps 13 to 21 of the algorithm. Finally redundant keyphrases are removed from the final refined set of keyphrases.

- **Data Manager:**
  - This module helps in data modeling and parameter settings before applying the mining algorithm.
  - It extracts a population of interest from messy email interaction data by removing noise.
  - The extracted information is modeled in graphs based on user defined interactions intervals and extracted keyphrases.
  - In each graph nodes are the individuals with keypharses as node label and directed edge represents the interaction between them.
  - Parameters set the thresholds of frequency and periodicity to identify the patterns of interest. For that, it is necessary to define a demanded minimum level (minimum confidence), so that all those sets of actions that have higher confidence level than the minimum confidence are considered as basic frequent periodic patterns.

```
Input: Training:
    (a) Set the parameters of KEA++ by keeping in view the keyphrase length in the taxonomy and email type
    (b) email along with their keyphrase and taxonomy
  Dataset for Extraction:
    (a) email with unknown keyphrases
Output: Set of refined keyphrases

TrainLevel ← KEA++ TrainLevel //(Rule I)
resultSet [] ← returned keyphrases by KEA++[]
resultSet [] ← level labels (Resultset [])
for resultSet[] <> empty do
    if (resultSet(training level)) then
        if (keyphrase level = lower level keyphrases) then
            processSet[] = preserving lower level keyphrases //Rule II
        else
            processSet[] ← identifying and preserving training level equivalent //Rule V
            processSet[] ← remove redundant keyphrases //Rule VI
            refineSet[] ← processSet[]
        end if
    else
        if (keyphrase level = training level) then
            refineSet[] ← processSet[]
        else
            if (keyphrase level = upper level) then
                processSet[] ← identifying and preserving training level
                    equivalent keyphrases //Rule V
            else
                processSet[] ← stemming lower level general keyphrases //Rule II
            end if
            processSet[] ← remove redundant keyphrases //Rule VI
            refineSet[] ← processSet[]
        end if
    end if
end for
return refineSet[] //refine result set of semantic keyphrases.
```

Figure 98 Keyphrases Extraction from Email Contents

- **Candidate Patterns**
    - This module identifies a set of frequent and periodic patterns from email interaction graphs.
    - Frequent patterns emphasize the significance of patterns and periodic patterns consider the regularity between them. Frequents patterns are mined by using the FP tree based approach while periodic patterns are mined using PSEMiner with integration optimization.
    - The objective is to identify the sets of actions that frequently and periodically occur together.
    - Once basic frequent periodic patterns have been discovered, an aspect to consider is if there is any action which is not frequent taking into accounts all the periods, so that it is not discovered in the previous task, but it is frequent enough in those periods where the basic frequent periodic patterns occur.

- **Patterns Pruning**
    - This module applies one mining process to identify frequent and periodic patterns under the given parameter settings.
    - Patterns pruning reflects the common characteristics of a typical email interaction with some unusual association between patterns. For that, the starting point will be the candidate patterns are transformed into integrated set in order to make it useful

comprehensively.

- Briefly explained, it infers meaningful actions from the data collected by email data and then it splits the string of actions into periodic sequences based on some frequent support.
- Combining these two concepts allows us to define periodic patterns in a way that avoids any redundant information.

- **Overall System Flowchart**
  - The overall CAME system flow chart with all components working is given in Figure 99. This shows that how different components of interaction analysis work together and how the output of one component flows to the other component where it is used as input for further processing.

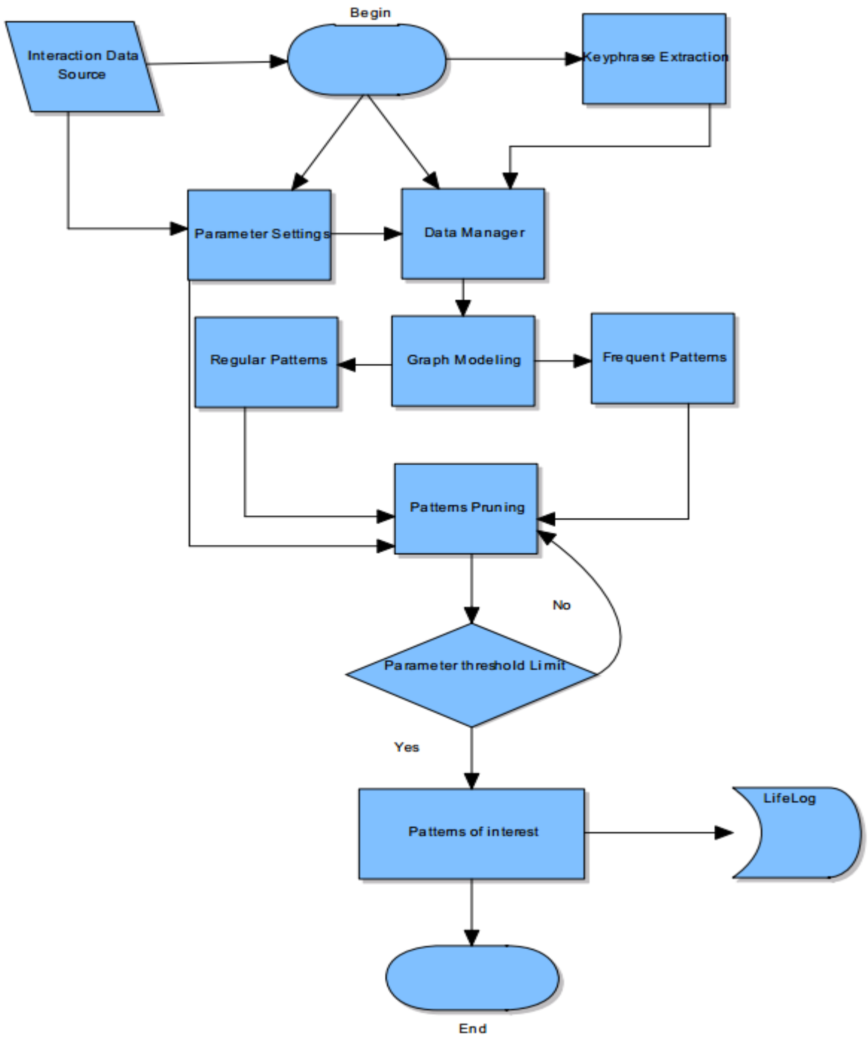


Figure 99 Overall System Flowchart

5.2.4. Behavior Modeling Module

- BMM is mainly responsible for collecting the information coming from diverse input sources and logging them in the life log.
- It takes input form HAR, CAME, and Social Media and log them in life Log.
- The life log is the main component of BMM that provide data for life style analysis and prediction. It also provides the facility of data for recommendations and chronic disease recommendations.

- **Introduction**
  - Use of ontology provides basic description to events, activities and interactions, and from this information we can deduce new knowledge.
  - Use of ontology in life log design and development is one of the main contributions of BMM.
  - The life log is modeled as ontology using OWL for all the activities and social interactions.
  - Later it uses the Life Log information for service provisioning.
  - The Life Log information is used for analyzing user behavior for shorter time interval and for longer time intervals which can be even years.
  - On the other hand, BMM also provide the facility for behavior prediction in case the current behavior is matching with some existing (stored, learned) behaviors.
  - The component based framework architecture diagram of BMM is given in main architecture; however, the detailed diagram of BMM is given in Figure 100.
  - The data access and storage is all handled using Parser designed and developed for manipulating the information handled in BMM.

- **Problems in Existing Systems**
  - To provide better lifecare services, information from different sources is necessary that can avoid any missing information which might result in wrong services. Hence, a challenge exists to both provide and maintain the quality and availability of lifecare services with a minimum cost [Khattak2010] and better accessibility.
  - The systems based on reminder have primarily focused on plan-based approaches to decide how and when to prompt subjects effectively [Mama2000] in addition to offering location-based reminders [Sohn2005]. Nevertheless, the context for the delivery of reminders involves more information than simply location or time. However, these systems are not based on data from different sources and also on data of historical nature.

- In [Wang2009], an ontology is used to incorporate context related information for intelligent processing; however, these systems provide the services for home held devices interaction and usage. However, they lack in service provision for lifecare services.
- Work presented in [Chen2012] details an ontology based reminder system; incorporates rules for manipulating the recognized activities of the elderly.
- In [Khattak2011], real-time activities are recognized using diverse sensors are used for situation analysis.
- Research in [Fitma2011] focused on the social interaction of patients. Based on the experiences shared, it generates intelligent service recommendations and eliminates their isolation.
- Existing systems aiming to offer a form of support are mostly based on one input modality, in some cases use imperfect contextual information [Henr2004], and if using diverse information with context information then they still provide the facilities of on time service provisioning. These solutions subsequently do not consider the lifecare services provisioning.

▪ **Requirements**
  - To overcome the limitations of the existing systems and provide better lifecare, life style, and analysis services, there is a need of sophisticated information collection, integration, and manipulation methodologies.
  - The basic requirements are discussed below.
    - Need to collect, integrate, and manipulate information based on the storage structure provided in the Life Log.
    - Need to model a methodological storage structure for information to be stored in Life Log.
    - For the storage structure, a storage model is designed and developed for the Life Log using OWL.
    - For the system to work properly, smooth communication among all the components is required. This needs the components to be atomic in their working.
    - To provide the lifestyle analysis and behavior prediction services, we need to design and develop or use sophisticated data mining algorithms which can fulfill our system needs for service provisioning.

▪ **Proposed Solution**
  - The Life Log information is collected from input sources i.e., Activity recognition, Social Media, and diet information and used for analysis of user lifestyle.
  - On the other hand, BMM also provide the facility for behavior prediction in case the current behavior is matching with some existing (stored, learned) behaviors.
  - The component based framework architecture diagram of BMM is given in Figure 100.
  - The details of all the components are given in the below given subsections.

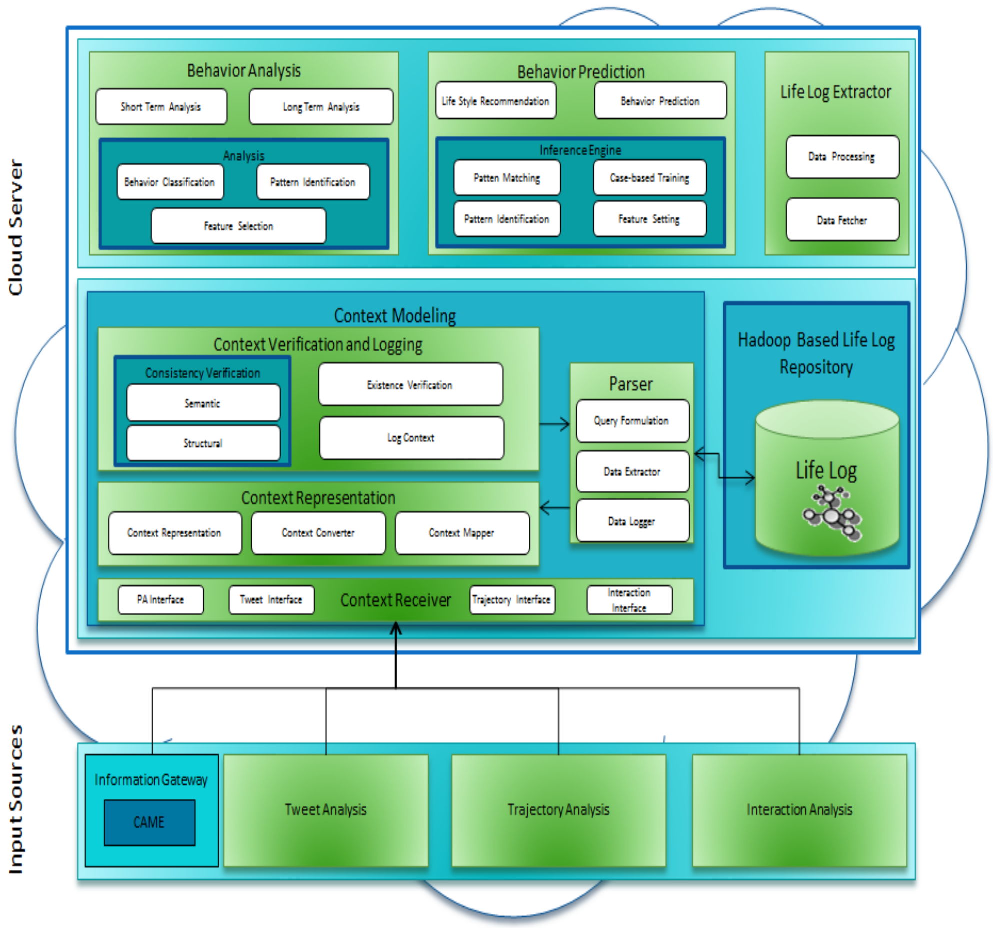


Figure 100 Overall architecture of Behavior Modeling Module (BMM).

- **Data Receiver**
  - This component is responsible for recognizing data inputs from diverse sources. Activities, Social Media, and diet information is provided in different formats like text (example given below in Figure 101) and/or in XML.
  - So there is need of recognizer that will recognize the simple representation of incoming information for their type and will initiate their respective queries in the Parser component.
  - There are different interface implemented in Data Receiver component that listen for appropriate incoming information.

- **Parser**
  - To store and access the information in and from Life Log, appropriated access patterns are defined.
  - It contains three main components: 1) Query Formulation, to formulate the query. 2)

Data Extractor, for extracting data from Life Log. 3) Data Logger, for logging the data in Life Log once the information is formalized.

```
20121104      I'm thinking, LONG:0, ACCx:-0.1915396558, ORIx:42.0038197266, ORIz:-1.0, MAGz:-46.0000697266,Instance:1
```

```
Topic: Exercise
Interest: Food
Time: 10:10:32 hr
Sentiments: Positive
User: u2
```

Figure 101 Text representation of activity information and social media information.

```
Lifelog: Activity_Instance
a                          LifeLog:Activity
Lifelog: hasInterest       Food
Lifelog:talkAbout          Exercise
Lifelog:hasTime:           2012:11:10:10:10:32
Lifelog:User:              User1
```

Figure 102 OWL representation (N3) of social media information collected using tweet analysis.

- **Context Representation**
  - This module is mainly responsible for representing the information the representational structural returned by Parser.
  - It consists of three main sub components: 1) Context Mapper which map the incoming context information with the storage structure. 2) Context Convertor which convert the context from simple text format to OWL format. 3) Context representation that represent the final converted context in OWL (N3) representational format as shown in Figure 102.

- **Context Verification and Logging**
  - This module is mainly responsible for verification of the context represented in the previous module. The verification is for structure, existence, and semantics.
  - Once the represented information is verified then the information is logged using Log Context.

- **Life Log**
  - Life Log is the main repository of BMM and LDSS for Wellness and LDSS for Chronic Disease..
  - It is responsible for proper communication of information among all the components of BMM.
  - It stores all the activities, diet, and social media information in different context/situation.
  - The proper engineering of the Life Log (See Figure 103) is most important activity in the development of BMM.
  - Life Log is designed as an ontology, where all the information is semantically mod-

eled and available for lifestyle analysis and behavior prediction.

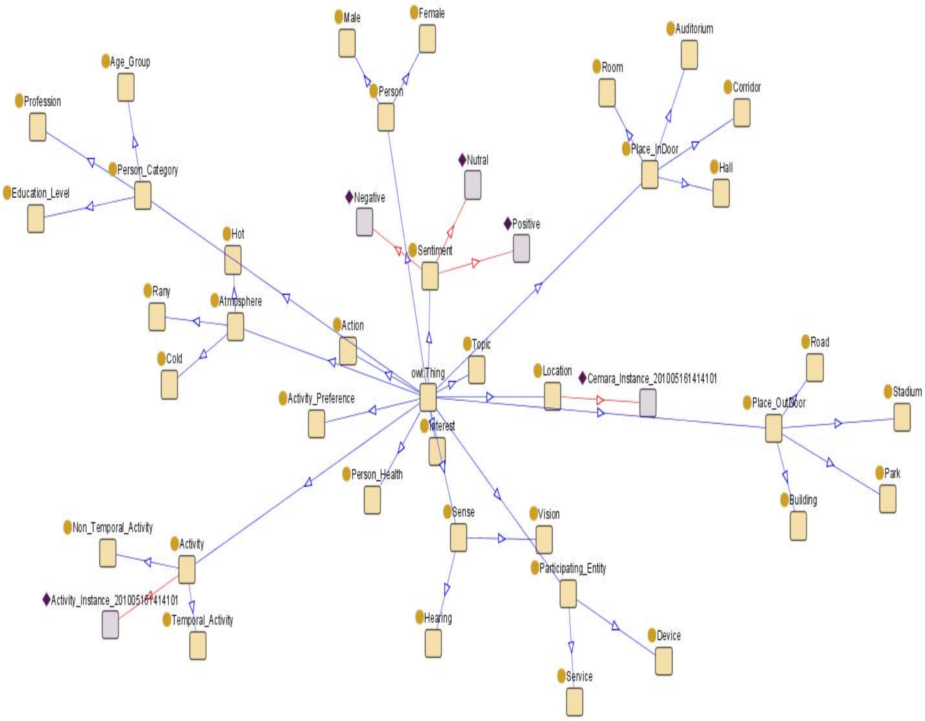


Figure 103 Life Log for information storage to be used in BMM

- **Life Log Extractor**
  - This component is responsible extracting information from Life Log.
  - It consist of two main components: 1) Data Fetcher is responsible for fetching appropriate information from Life Log to provide the information for appropriate services. 2) Data Processing is responsible for preprocessing of information extracted from Life Log and provide it to services.

- **Behavior Analysis**
  - It is responsible for analyzing the Behavior of a person for its activities and his consumed services.
  - This services help the user to have a look at his life from different aspects and find out any problems which if removed can make his/her lifestyle better.
  - This service have two subservices: 1) Short Term Analysis that provide the analysis services for shorter time like for a week or 10 days. 2) Long Term Analysis provide the services for lifetime activities and information.

- **Behavior Prediction**
  - This service is responsible for providing the behavior prediction services.

- This service is also composed of two subservices: 1) Lifestyle Prediction that predict the user lifestyle based on his /her activities and matching it with the already learned lifestyles stored in the repository. If the lifestyles are matching in some of its aspects then the system generates the predicted lifestyle. 2) Behavior Prediction is based on user everyday behavior and lifelong behavior. Based on the user behavior monitored using activity recognition systems and social media information, the system analyze the behavior and predict the possible future behavior of the user in a given situation.

- **Overall System Flowchart**
  - The overall BMM system flow chart with all components working is given in Figure 104.
  - This shows that how different components of BMM work together in integration and how the output of one component flows to the other component where it is used as input in the second component for further processing.

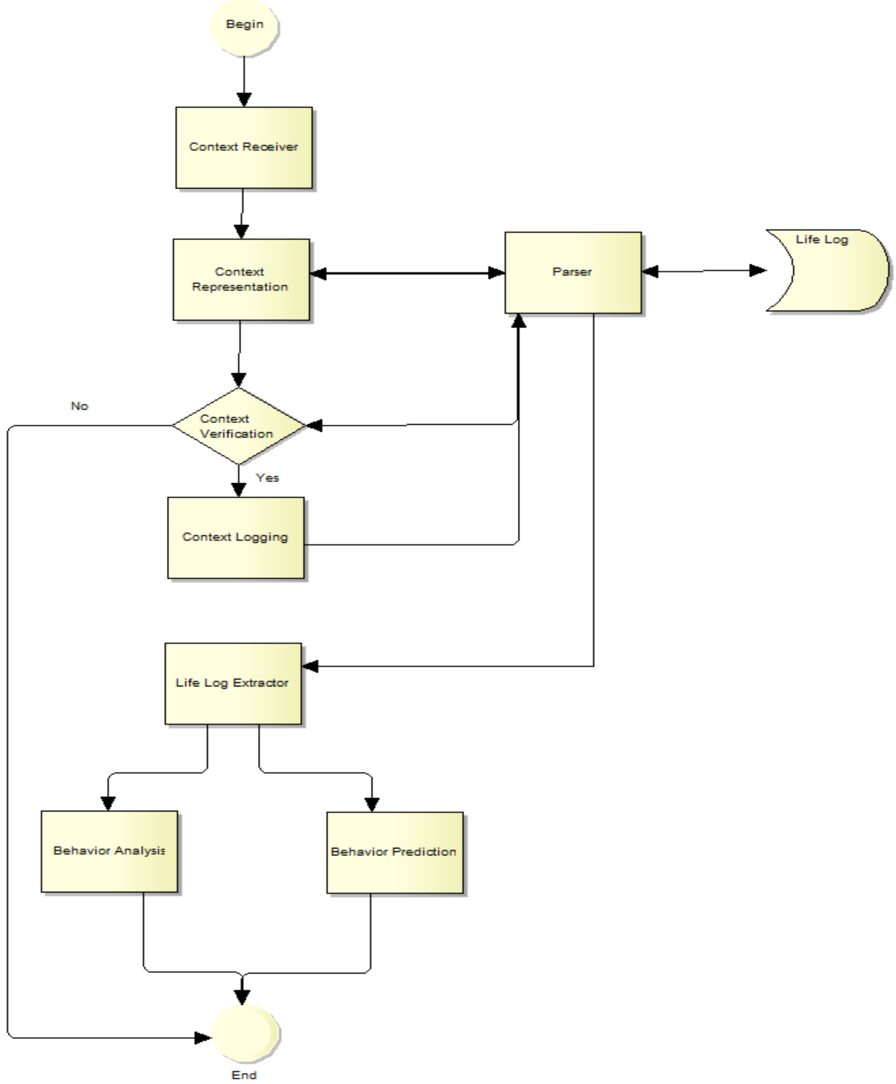


Figure 104 Flow Chart for overall working of BMM

5.2.5. Adapter Interoperability Engine

- **Introduction**
  - One of the key aspects of LDSS- Chronic Disease is its interoperability with Health Management Information Systems (HMIS) compliant to different healthcare standards. LDSS service can only be utilized by HMIS's if processing of information is performed in their respective interpretable formats. **Adapter Interoperability Engine (ARIEN)** resolves these heterogeneities for HMIS's to interact easily with LDSS to utilize its services. ARIEN is subcomponent of LDSS-Chronic Disease Adaptability Engine and behaves as a mediator between LDSS and HMIS systems. HMISs compliant to different healthcare standards understand only the standardized format such as: HL7 CDA, openEHR, CEN 13606, while LDSS can only process virtual Medical Record (vMR) format. ARIEN provides bridge services that use ontology matching techniques to generate mappings between heterogeneous healthcare standards for automatic transformation of information to enable interoperable communication among healthcare systems. A demonstration of resolving heterogeneities between HL7 and openEHR standards at model level is already presented in our previous work [WKhan2009].
  - ARIEN takes as input standard form (e.g. HL7 CDA) and transforms it to vMR format using already generated and stored ontology mappings in repository. This transformed vMR format is processed by smart CDSS for generating recommendations. After processing the information is again required to be converted from vMR to HMIS complaint standard format. AREIN converts it to standard format and passes information to HMIS in standard format. So far transformation between HL7 CDA and vMR is successfully completed with some deficiencies while openEHR and vMR transformation is underway. Complete mappings cannot be generated using ontology matching techniques therefore manual mappings are also used to increase the accuracy of mappings while transformation.
  - The main objectives of ARIEN are accuracy and continuity of mappings. While accuracy is achieved using ontology matching with manual matching scheme, continuity of mappings is realized by ontology change management.

- **Problems in Existing Systems**
  - For achieving interoperability in healthcare domain, some systems have used ontology matching, SOA architecture, and also semantic web services framework. Some of these systems, closely align with the proposed system are discussed below;
  - Jini Health Interoperability Framework (HIF-J) [Ducrou2009] uses Jini technology which is based on SOA. The main purpose of HIF-J is to exchange semantically interoperable messages. It provides translation services that behave as a mediator between standards. These translation services convert message instances HL7 V2 and V3 and also HL7 and openEHR message instances. It is based on XSLT transformations between message instances of different standards. Since standards are growing with new domains, so managing XSLT becomes very difficult. Moreover, XSLT is just transforming syntactic structure and semantic transformation is not achieved.
  - Artemis [Dogac2006] project is based on achieving semantic interoperability between healthcare systems by using semantic web services. It also uses the concept of semantic mediation which focuses on resolving the heterogeneities between different standards. It

mainly focuses on resolving the heterogeneities between HL7 V2 and V3 standards. Artemis uses OWLmt tool which an ontology mapping tool providing a graphical user interface to define the mappings between two ontology schemas. It is limited only to conversion between HL7 V2 and V3 standards.

- PPEPR [Sahay2008] project is an integration platform that focuses on resolving the heterogeneity problem between two version of the same standard HL7 (V2 and V3). It is based on semantic SOA concepts and solves the problem of interoperability at the semantic level. It used Web Service Modeling Ontology (WSMO) approach unlike Artemis which uses OWL-S. It mainly focuses on integration of Electronic Patient Records and conversion between HL7 V2 and V3 is specified. The scope is only limited to transformations between standards that comes under the umbrella of HL7.
- Ortho-EPR [Magni2007] standard is a proposed standard that is based on the integration of HL7 and DICOM standards for electronic orthodontic patient records. The main purpose of this standard is storage and communication of orthodontic patient records. The message part is handled by HL7 while imaging is handled by DICOM and there integration results in Ortho-EPR standard. Its main purpose is the integration of two standards and not interoperability between standards.
- In [Khan2009], the authors focus on semantic process interoperability with the help of interaction ontology in HL7 V3. Interaction ontology is responsible for handling the heterogeneities between processes of different healthcare organizations compliant to HL7 V3 standard. This work is only related to semantic process interoperability using standard HL7 V3 and semantic data interoperability is not discussed.
- Existing systems mainly focuses on the conversion of instances between different standards while our focus is on the accuracy of mappings in addition to conversion of instances. Other than accuracy of mapping, we are also considering the continuity of mapping that is not the focus of the existing systems.

▪ **Requirements**
  - The requirement of the systems for achieving true semantic interoperability is as follows:
    - Handling heterogeneous data formats of different standards such as HL7, openEHR, CDA and CEN 13606.
    - Creation of bridge ontology to store all the mappings generated from the heterogeneous sources.
    - Transformation from source data format into target data format for interoperability.

▪ **Proposed Solution**
  - Adapter interoperability engine main focus is on resolving heterogeneities among heterogeneous healthcare standards. The main components of the system are shown in Figure 105 and described as follows:

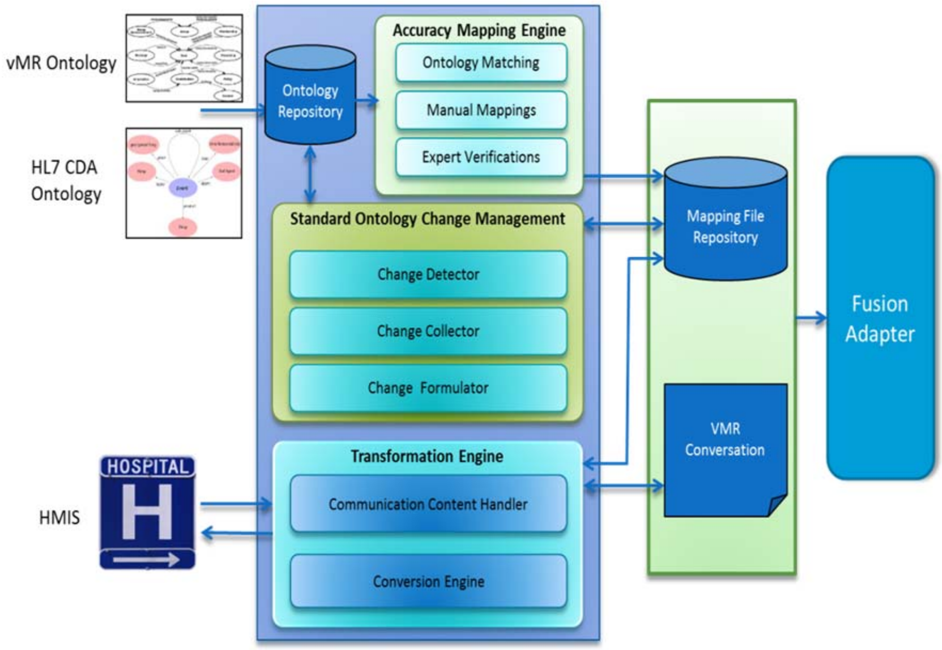


Figure 105 Adapter Interoperability Engine Working Model

- **Accuracy Mapping Engine**
  - Ontology matching component is responsible for matching the source and target ontologies of different standards. In LDSS-Chronic Disease the source and target ontologies are vMR ontology and HL7 CDA Ontology.
  - The mappings generated are stored in Ontology Repository.
  - Manual Mappings are required where the ontology matching techniques fails to find the appropriate mappings.
  - Expert verifications are necessary in the end to make sure that there are no impurities in the mapping repository.

- **Ontology Repository**
  - The mappings generated by the Accuracy Mapping Engine are stored in the ontology repository. The mappings are stored in the form of bridge ontology.

- **Standard Ontology Change Management**
  - Changes in ontologies should be propagated in the mapping files. Therefore, Standard Ontology Change Management module is used for incorporating the changes of the ontologies.
  - Change Detector components detect the change in the ontology and initiate the process of ontology change management.
  - Change Detector component provides the information of change to the Change Collector component that indexes the information about the change and pass it to Change Formulator.

- Change Formulator formulates the change according to the format in which the matching would take place and mappings can be stored.

- **Transformation Engine**
  - Communication Content Handler component is responsible for communicating with the HMIS and internal component of Adapter Interoperability Engine to transform the source standard format into target standard format for processing.
  - Conversion Engine converts CDA standard format to vMR and vice versa. As the HMIS only understands CDA (in this case only) and the LDSS-Chronic Disease can process the data in vMR format, therefore, transformation is necessary.

▪ **Overall System Flowchart**

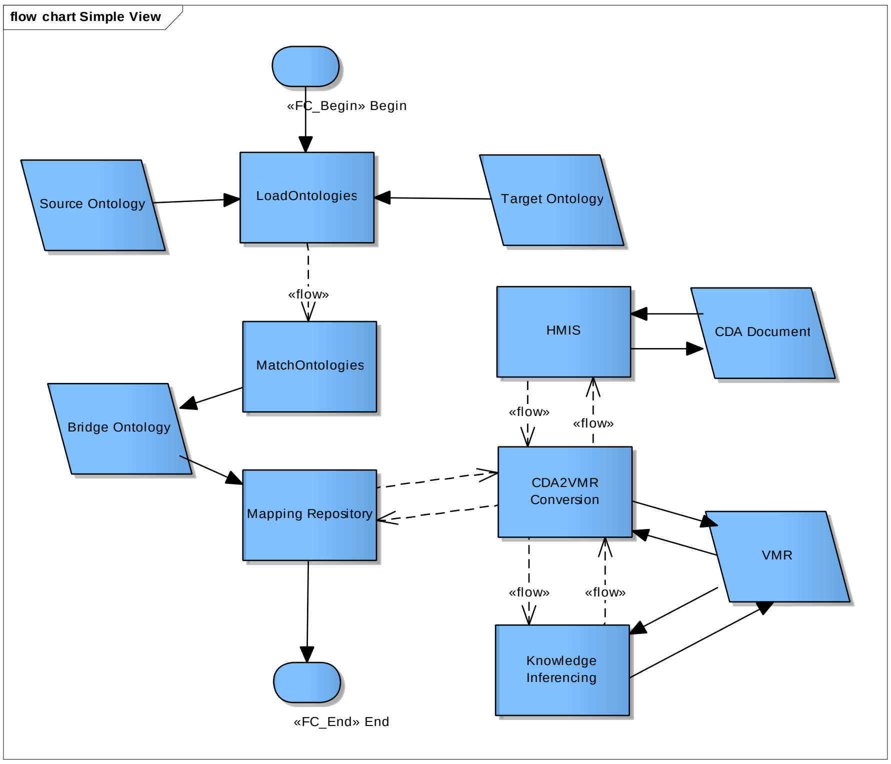


Figure 106 Adapter Interoperability Engine overall Flowchart

5.3. SmartCDSS for Head & Neck Cancer

- **Introduction**
  - Smart CDSS is a clinical decision support system that assists physicians to provide guidelines and recommendations for Head and Neck Cancer (HNC) patients.
  - The architecture of Smart CDSS has several important components. Knowledge Base (KB) that practice standards like HL7 Arden Syntax for ease of sharing clinical knowledge.
  - It incorporates emerging HL7 vMR (virtual Medical Record) to allow access to KB using standard base interfaces for seamless integration with diverse healthcare systems.
  - In order to achieve interoperability, the output of the smart home application is transformed to standard input based on vMR and get reminders and guidelines from Smart CDSS in standard format.
  - In order to reach to appropriate guidelines, a Reasoner component performs reasoning on KB.

- **Problems in Existing Systems**
  - Most of the existing CDSS are proprietary and highly coupled to medical systems having non-standard knowledge base. These systems facing problems during sharing the knowledge of the practitioners across the organization boundaries.
  - Existing CDSS also facing challenge of maintaining the knowledge base. As knowledge bases are growing with time, the alerts and recommendations are duplicated or contradicted with existing knowledge rules.
  - CDSS adoption becomes nightmare due to complex interfaces for practitioners. The physicians feeling difficulties in publishing their experience to the knowledge base that can be executed in their practices and sharable across the organization.

- **Requirements**
  - Objectives of the Smart CDSS include facilitating physicians during diagnosis of head and neck cancer, provide meaningful and timely base recommendations for nurses during patient care plan and allow patients to manage cancer treatment with provision of physician guidelines and online resource guidelines from authentic cancer networks.
  - In most general form, Smart CDSS will provide stakeholders with following main functions;
    - Allow physicians to incorporate their experiences into shareable and executable knowledge base for head and neck cancer.
    - Facilitate diagnosis process of head and neck cancer patient at different level by incorporating interventions, such as helping in finding appropriate staging level and facilitating multi-disciplinary conference panel in assigning appropriate treatment plan for patient.
    - Helping nurses and patient with displaying appropriate guidelines during treatment.
    - To strengthen physician experiences, evolving clinical knowledge base from online cancer resources.
    - Providing stakeholder with easy to use interface for interacting with system and hence improving patient care.

- **Proposed Solution**
  - Developing Knowledge Base (KB) that practice standards like HL7 Arden Syntax for

ease of sharing clinical knowledge.

- Smart CDSS incorporates emerging HL7 vMR (virtual Medical Record) to allow access to KB using standard base interfaces for seamless integration with diverse healthcare systems.
- Providing physician with easy to use tool to develop their practices and publish in to standard executable knowledge base.
- Enable KB evolution from online guideline resources to become part of physician KBs.
- Following is detail architecture of Smart CDSS for Head and Neck Cancer.

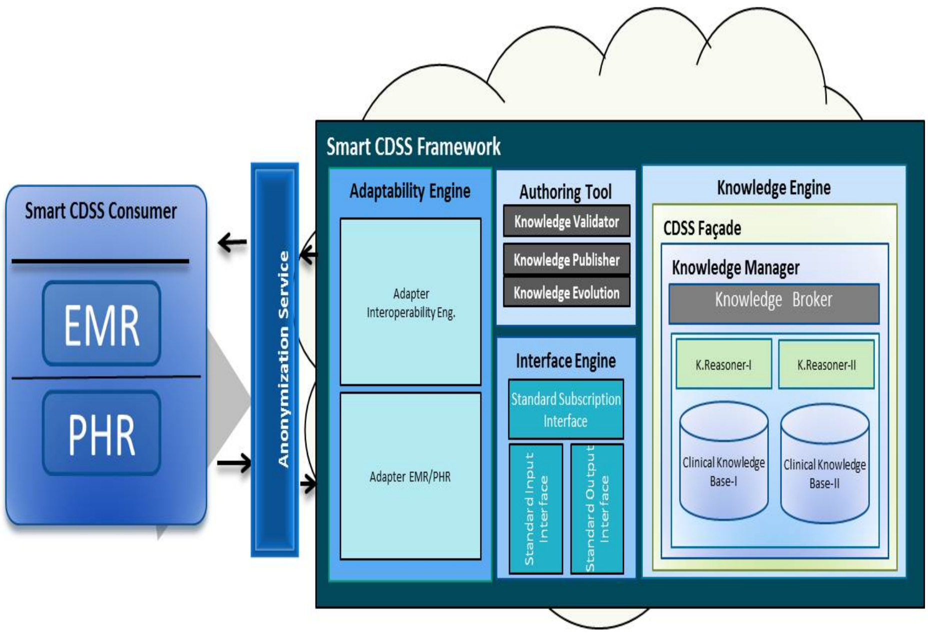


Figure 107 Smart CDSS Architecture

- **Adaptability Engine**
  - It allows supporting diverse applications that interact with Smart CDSS. The candidates application are PHR, EMR and EHRs where the proprietary input is converted to standard HL7 vMR format of Smart CDSS.

- **Interface Engine**
  - Interface Engine encapsulates HL7 vMR standard specifications and validates the incoming request and outgoing recommendations according to HL7 vMR standard schema.

- **Knowledge Engine**
  - Knowledge Engine is core module of the system which composes of clinical knowledge bases, knowledge reasoner and appropriate knowledge broker. The knowledge base contains executable clinical knowledge published by physicians or transformed from online resources. Current knowledge base covers the Head and

Neck Cancer and is represented in HL7 Arden Syntax for sharing across the organization.

- Knowledge broker allow picking appropriate knowledge from different knowledge owners or institutions. Knowledge reasoner schedule appropriate rules according to event criteria and application input.

- **Authoring Tool**
  - Authoring Tool allows the physician to create and publish the knowledge into knowledge base. Moreover, Authoring Tool also allow to integrate knowledge from online resources.

▪ **Overall System Flowchart**

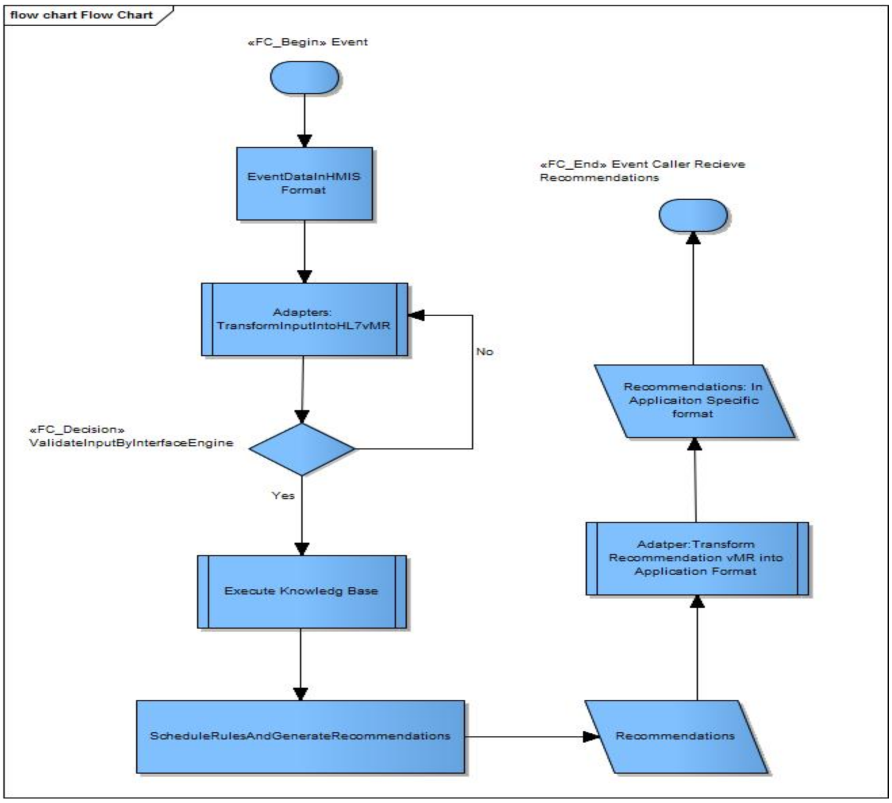


Figure 108 Smart CDSS Overall flowchart diagram.

5.4. Cloud Computing and Security

- Java being the most used language globally attracts more developers and engineers to develop their scientific applications; however, HPC based hardware is still a requirement for scientific application execution. So far Clusters have been providing backbone architecture for HPC applications.
- Message Passing Interface (MPI) is a de-facto standard of HPC-based communication over clustered, facilitated by numerous C, C++ and FORTRAN based implementations.
- A Java-based scientific application can only utilize MPI via Interoperability layers, which inflict performance degradation at large. With this point of view, Several Java-based MPI implementations are currently available, facilitating a bypass over interoperability and accessing clusters HPC. This technique has worked really well for clusters but has shown bottlenecks over virtualized platforms.
- Cloud computing being an emerging market does provide commodity hardware to perform in distributed fashion and can be exploited to perform HPC; however, traditional Java-based MPI-libraries (MPJ) show reasonable lag over virtualized platform (Hypervisor) of cloud computing.
- High Performance Computing over Cloud (HPCoC) is our contribution in regards for studying possible bottlenecks involved and enabling cloud's virtualized platform to perform HPC over Clouds.

5.4.1. MPJ-Core

- **Introduction**
  - Java-based High Performance API in production ready mode is already rare. As most of the research intentions are either using the industry standard but outdated MPI-Like APIs built in C, C++ and FORTRAN. However the success of MPJ Express [Carpenter2000] in last few years have given us the opportunity to extend that design and enhance it for its efficient utilization over Cloud.
  - In first year we will go over the current implementations of parallel messaging API's for our general understanding of core focus areas built earlier and how can they be improved and implemented over a cloud like platform i.e. Virtualization-awareness.
  - This Literature survey will include the benchmarking of commercially available MPI-like implementations over both bare metal systems and virtualized private cloud environment. This experiment will provide us the figure of performance loss on average, a message library experiences over a virtualized environment. This figure will support our implementation to quantify the amount of improvement required in the MPJ-Core implementation.
  - Apart from literature survey, in-depth requirement engineering process will be executed by keeping the understanding of current MPI-like API's in view and will be validated with High Performance Computing community and MPI Specification. MPJ-Core Execution Flow Algorithm will be created and validated.
  - From the extracted requirements, MPJ-Core API will be implemented and published in HPC and Cloud-based conferences for validation and feedback.
  - MPJ-Core API will lead us to provide a detailed architecture for MPJ-Core Implementation phase that will be executed in 2nd year of our project.

- **Problems in Existing Systems**
  - There have been various efforts in the last decade to develop Java messaging systems. Many of these systems were experimental and are no longer supported, or the software is not available. As a response to the appearance of several prototype MPI-like systems, the Message-Passing Working Group of the Java Grande Forum was formed in late 1998. This working group came up with an initial draft for a common messaging API, which was distributed at the Supercomputing conference in 1998. Since then, two APIs namely mpiJava 1.2 [Carpetner1999] and MPJ [Carpenter2000] have appeared. The main difference between these two APIs lies in the naming conventions of variables and functions.
  - Java messaging systems for HPC typically follow one of three approaches: use a JNI interface to a native MPI implementation; implement Java messaging from scratch on top of Java Remote Method Invocation (RMI); or implement the system on top of lower level Java sockets.
  - MPJava [Pugh2003] was a message-passing library that was implemented using the Java NIO package by The University of Maryland. This package demonstrated that Java messaging systems based on NIO could achieve performance comparable to that of C or FORTRAN message passing libraries. The runtime infrastructure consisted of shell scripts that allowed processes to be started on Linux-based nodes. Figure 109 in [Pugh2003] shows the ping-pong performance comparison between the java.io (bytes), java.io (doubles), MPJava and LAM/MPI 6.5.8. It shows that the LAM/MPI implementation attained greater throughput than MPJava for message sizes up to about 1000 doubles. For message sizes larger than 7000 doubles, MPJava attained a greater throughput than LAM-MPI. The source code of MPJava has not been released publicly.
  - mpiJava [Baker1999] is a Java messaging system that uses JNI to interact with the underlying native MPI library. The project started in 1997 at NPAC (Syracuse University), later moved to the University of Florida, and then Indiana University. The mpiJava software API is currently in version 1.2, although there are plans to move to the MPJ API in the future.

- **Requirements**
  - To overcome the limitations of the implemented systems following requirements need to be implemented.
    - For High Performance Computing (HPC), a high-speed and reliable, inter-node and intra-node communication middleware is a must. For HPC over Cloud (HPCoC) this middleware must be virtualization-aware with ability to support the Vertical Scaling of HPC (Performance / node) & Horizontal Scaling of Cloud Computing (Availability & persistence / node).
    - Inter-node and Intra-node communication gap between sequential application over a cluster or a cloud is minimal and un-noticeable however while executing parallel applications over the similar environment can lead towards significant communication lags and delays in exponential order

- **Proposed Solution**

  We are proposing a device system composed of multiple device layers exploiting the strength of Java's IO library and in parallel providing support for conventional mpiJava based communication devices.

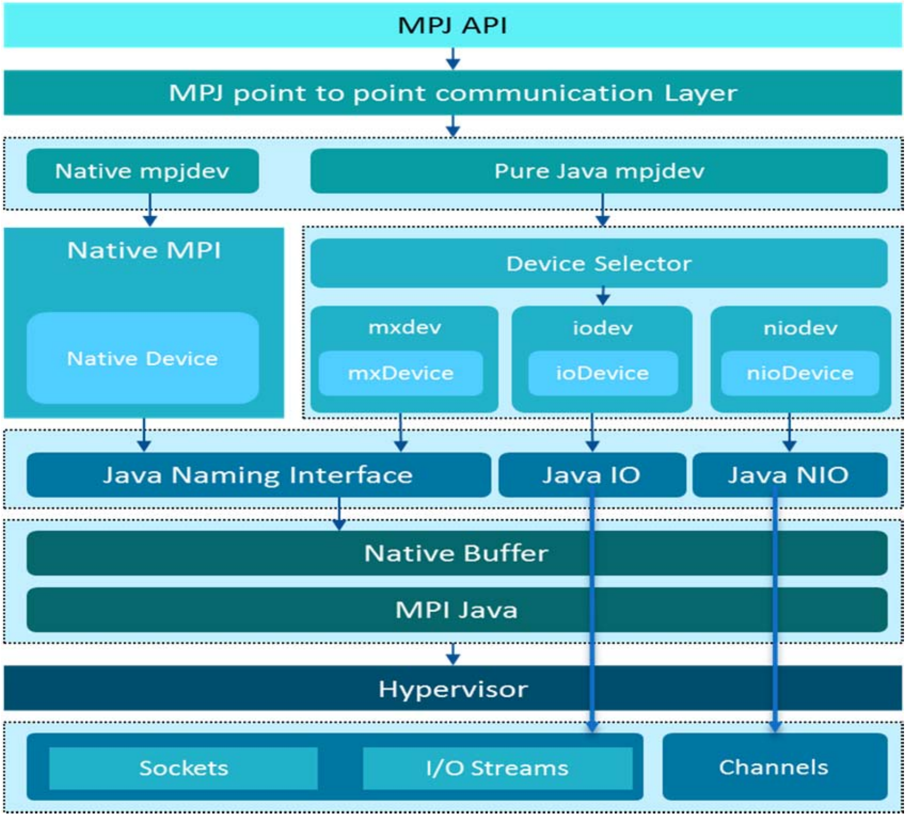


Figure 109 MPJ-Core Proposed Architecture Stack

- **Overall System Flowchart**

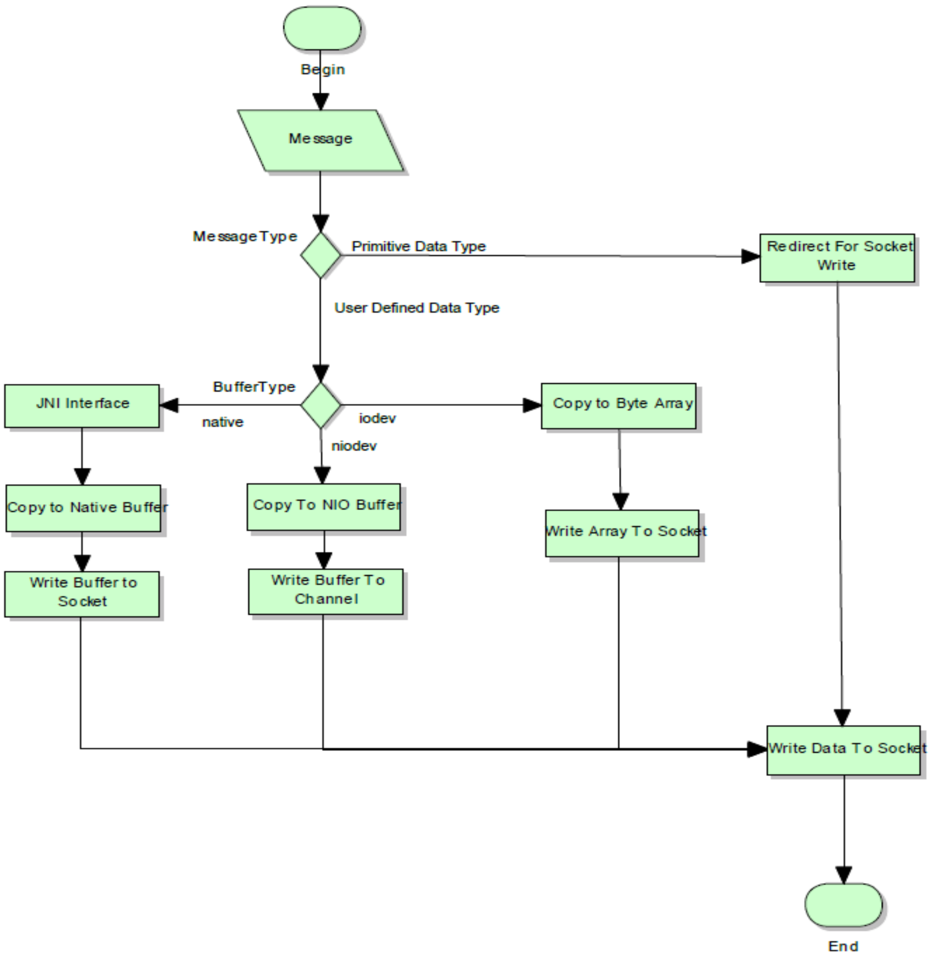


Figure 110 MPJ-Core Flow chart

5.4.2. MPJ-Runtime

- **Introduction**
  - MPJ can be configured in two ways i.e. Multicore Configuration and cluster configuration. The multicore configuration is used by developers who want to execute their parallel Java applications on multicore or shared memory machines (laptops and desktops). Cluster configuration is used by developers who want to execute their parallel Java applications on distributed memory platforms including clusters and network of computers.
  - The growth of cluster computing has played a significant role in high performance computing. It has led to comprehensive software stack for clusters which includes job scheduling, resource management etc. Message passing interface is a major programming paradigm in HPC. Its runtime system should be scalable, resilient to errors and process

crashes as well as process health monitoring. Scalability is to deploy the set of processes in logarithmic time . The commodity clusters also come with a number of challenges when scheduling MPI jobs on batch systems like PBS because they can't track all resource utilization and processes with conventional secure shell authentication [Castain2005].

- **Problems in Existing Systems**
  - For harnessing the power of thousands and millions of cores we need an optimized runtime infrastructure for the MPJ. MPJ has an experimental runtime which does not scale well to batch scheduling systems like PBS and SGE. There has been no fault monitoring and fault tolerance in MPJ Express which makes the application more resilient to errors. We aim to minimize the application startup and spawn application process scalable manner.
  - mpiJava [Baker99] is a Java messaging system that uses JNI to interact with the underlying native MPI library. Some conflicts between the JVM and the underlying MPI implementation were reported but the situation has improved with the evolution of the JVM. Initial versions of this software transferred only the primitive data-types, but the current version supports the transfer of Java objects using automatic JDK serialization

- **Proposed Solution**

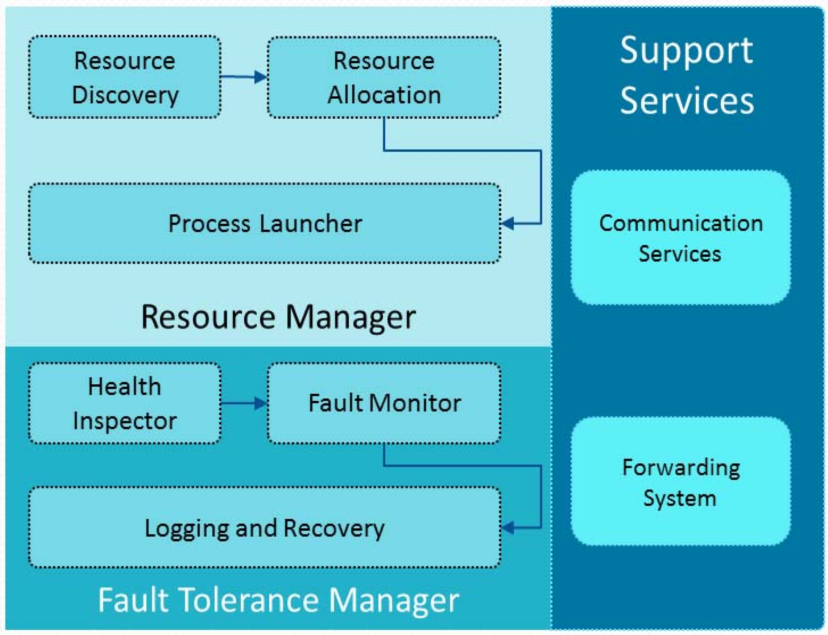


Figure 111 MPJ-Runtime Proposed Architecture Stack

- **Resource Manager**

- **Resource Discovery Module**: The Resource Discovery Module (RDM) is responsible for identifying the computational resources available to the MPJ runtime system and making that information available to other modules. The RDM currently reads host files which are typically generated by a specific user and contain information on machines that might be available to that user; and it also obtains its information from a system-level resource file containing an XML-based description of the cells known to the MPJ runtime system. Information from RDM is placed on a file for easy retrieval by other subsystems within that system.

- **Resource Allocation Module:** The Resource Allocation Module (RM) examines the environment and the command line to determine what, if any, resources have already been allocated to the specified application. If resources have not been previously allocated, the module will attempt to obtain an allocation from an appropriate cell based on information from the RDM. Once an allocation has been determined, the RM constructs two segments – a node segment containing information on the nodes allocated to the application, and a job segment that holds information on each process within the application (e.g., node name where the application is executing, communication sockets, etc.).

- **Process Launcher**: Finally, the Process Launcher (PL) utilizes the information provided by the prior subsystems to initiate execution of the application's processes. The PL starts by spawning a head node process (HNP) on the target cell's frontend machine. If an existing HNP is not available, then the new HNP identifies the launch environment supported by that cell and instantiates the core universe services for processes that will operate within it. The application processes are then launched accordingly.

- **Fault Tolerance Manager**
    - Health Inspector: The Health Inspector (HI) decides when to start monitoring depending on the application specifications and user input.
    - Fault Monitor: Error management within the runtime is performed at several levels. Wherever possible, the condition of each process in an application is continuously monitored by the Fault Monitor (FM). Fault monitor can monitor through both pull and push method.
    - Logging and recovery: Once an error has been detected, the Logging and recovery (LNR) subsystem is called to determine the proper response. The LNR can be called in two ways: locally, when an error is detected within a given process; or globally, when the FM detects that a process has abnormally terminated.

- **Support Services**
    - Forwarding System: The forwarding system (FS) is responsible for transporting standard input, output, and error communications between the remote processes and the user. Connections are established prior to executing the application to ensure the transport of all I/O from the beginning of execution
    - Communication Services: This module transports data on process state-of-health, inter-process contact information and serves as the conduit for communication around the FT manager and resource manager

- **Overall System Flowchart**
    - The MPJ Runtime starts with a simple script which will authenticate the cluster and indicate that the processes are ready to start and also get the resources available. The pro-

cesses are divided into the nodes to run and processes are launched.

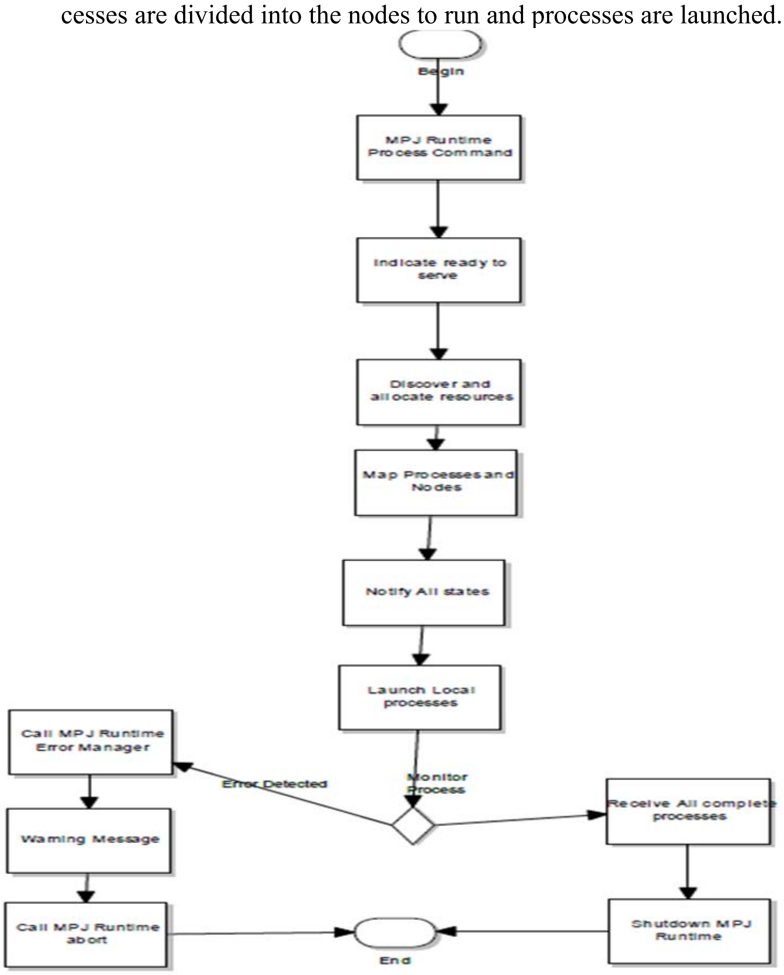


Figure 112 MPJ-Runtime Flow chart

5.4.3. Parallel I/O

- **Introduction**
  - MPI provides a rich collection of functions for transferring data between MPI processes. This type of transfer is called communication. On the other hand, when MPI processes move the data into and out of an application to or from files on a storage system, it is called I/O. Parallel I/O in general means any I/O that is performed by an MPI application.
  - The most important contribution for MPI I/O lies in the area of collective I/O, which is defined as the I/O functionality that takes place between multiple MPI processes accessing a single file.
  - POSIX provides the most widely used file system interface, but it cannot handle high

performance I/O because it lacks the capability to handle performance optimizations like grouping, collective buffering and disk directed I/O. This is because POSIX has no support for collective data operations and describing how the file data is partitioned among processes.
- MPI has the potential of providing a high performance interface for File I/O because it already has support for collective communication among processes. Based on the same principle of collective communication, MPI I/O provides collective file operations. It also provides greater control over physical file layout on storage devices like disks.
- MPI only specifies a programming interface (an API) for I/O rather than providing specifications for a file system. This allows MPI-2 I/O interface to work with a wide range of file systems.
- In order to partition the file data among processes one needs to define I/O modes for access patterns of processes to a shared file just like we have broadcast, reduction, scatter, gather. MPI takes an alternate approach and leverages the power of MPI derived data types to define partitioning of file data.

▪ **Problems in Existing Systems**
- There have been enormous efforts to improve I/O performance in high performance applications. A lot of research projects have addressed I/O performance in C, C++ and FORTRAN but very little research has been done for Java. Java has evolved over the period of time and new I/O APIs have been released incrementally.
- A very important evaluation exists in [Thakur2001]. This paper is very well written and compares different approaches to perform parallel I/O in Java. The paper discusses bulk I/O or array based file I/O operations in Java whereas Jaguar allows Java runtime system to be extended with new primitive operations that allow efficient access of the hardware resources available to Java programs. Jaguar also has a novel idea of pre-serialized objects ready for communication and I/O. Our project will address to develop and MPI I/O style File I/O API for Java and then provide a reference implementation of this project.
- AgentTeamwork [Phillips2007] is a grid-computing middleware system that coordinates parallel and fault-tolerant job execution with mobile agents. The software has its own parallel file I/O facility called AgentTeamwork's MPI-I/O-oriented random access file, which is implemented using GridTcp and GridFile. GridTcp implements error-recoverable TCP communication over multiple clusters. GridFile provides the same interface as Java files including RandomAccessFile and buffers file contents as serializable data in the wrapper. The API is not MPI-IO compliant and is severely limited in its functionality. The software is very inefficient in terms of performance because of the use of Grid protocols like GridTcp for communication over the network.

▪ **Requirements**
- Defining a parallel I/O API which is written for Java language by keeping in mind the specifics of an object oriented language.
- Implementing the MPJ-IO API and provide an easy to use interface for Java HPC programmers.
- Documenting the entire parallel I/O processes relevant to the JVAPIO library.
- Providing sufficient performance evaluations to prove a case for Java based parallel file access interface.

▪ **Proposed Solution**
- The parallel I/O API for Java, which we will develop in this project, does not exist at the

time of writing this document so the entire process of developing this API is unique to our team. Other parallel I/O APIs have only focused on C, C++ and FORTRAN languages whereas we will focus on object oriented programming languages such as Java.

- Most other APIs do not have virtualization aware or cloud aware implementations, but we plan to optimize our API and its implementation to be virtualization aware from the very start.
- Benchmarking the implementation of the API on upcoming and prevalent cloud platforms will enable us to provide the users tweaked and custom versions of our implementation for each cloud platform. This has not been done before by any implementer of the MPI like libraries.
- The overall architecture modules are shown in the Figure 113. The figure explains how the API is divided into different functional modules.

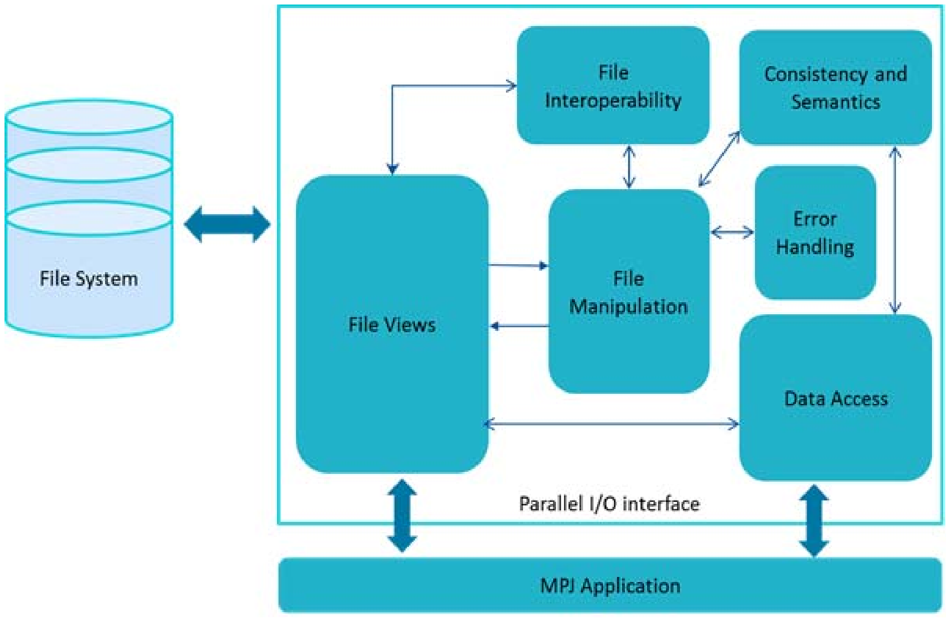


Figure 113 Proposed Architecture

- **Overall System Flowchart**
  - The overall flowchart for the parallel I/O system is in Figure 114.
  - The flowchart explains how the system will start for a shared memory machine as well as a distributed memory machine.

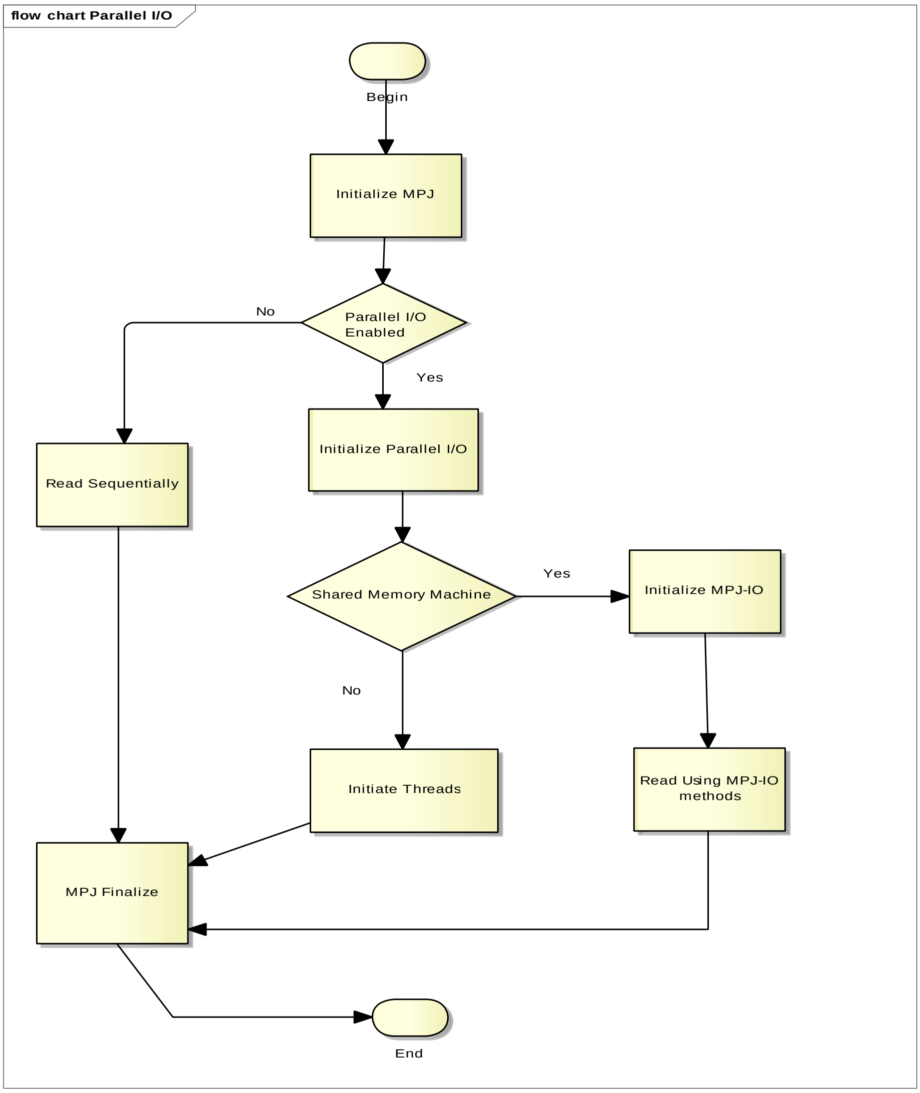


Figure 114 Overall System Flowchart

5.4.4. Anonymization and Cloud Security

- **Introduction**
  - Cloud computing is an epitome of on demand computing. It provides virtualized computing resources (i.e., processing power, storage facility, application services) over the internet. However, as cloud is owned and managed by a third party, the risk of privacy infringement escalates when confidential data is outsourced to an untrusted domain of cloud service provider. The most common approach to ensure data confidentiality is

through encryption, which conceals information such that no one can learn information except those having valid data encryption key.
- Although encryption ensures data confidentiality, however it restrains data processing (i.e., searching, processing, modification etc.) over encrypted data. To process data residing within the untrusted domain of a cloud service provider, data owner is left with two options. First, share data decryption key with the cloud service provider, once processing is completed data is re-encrypted with new encryption key. Second, download the entire encrypted data locally, decrypt it with valid data decryption key and then process it.

▪ **Problems in Existing Systems**
  - Both of the options discussed earlier are not feasible for data intensive applications, where huge amount of data is required to be processed. In order to ensure data privacy there is a need to anonymize the data. Anonymization ensures that data residing within the untrusted domain of a cloud service provider cannot be linked to actual users (i.e., diabetic patients). With anonymous data outsourced, there is no need to encrypt data in order to achieve data privacy. The Figure 115 on next page illustrates the conceptual model of our proposed anonymization service on hybrid cloud configuration.

▪ **Requirements**
  - Anonymization service enables cloud service subscribers to make most of the hybrid cloud.
  - It enables cloud service subscribers to process confidential data within un-trusted domain of a cloud service provider without privacy concerns.
  - It utilizes secure pseudo randomness generator to conceal confidential data.
  - It is a configurable service which can be adapted to process data according to the context

▪ **Proposed Solution**
  - Anonymization service de-identifies the data, by replying actual identities with pseudo identities. This ensures that diabetic patient cannot be linked with the data residing within in a cloud service. Anonymization service provides a mapping between the real identities and pseudo identities of diabetic patient. This mapping is stored on secure location in order to avoid risk of privacy infringement in any adverse event. Figure 116 shows the architecture of anonymization service, which provides a bridge between the mapping stored locally on secure server and data outsourced to public cloud for clinical processing.
  - Anonymization service is further composed of four cohesive components ensuring that data migrating to cloud service is properly de-identified and cloud service provider cannot exploit it to compromise privacy of a diabetic patient. Anonymization service is composed of Identity Management, Consent Management and Consent Validator and Anonymization Proxy.

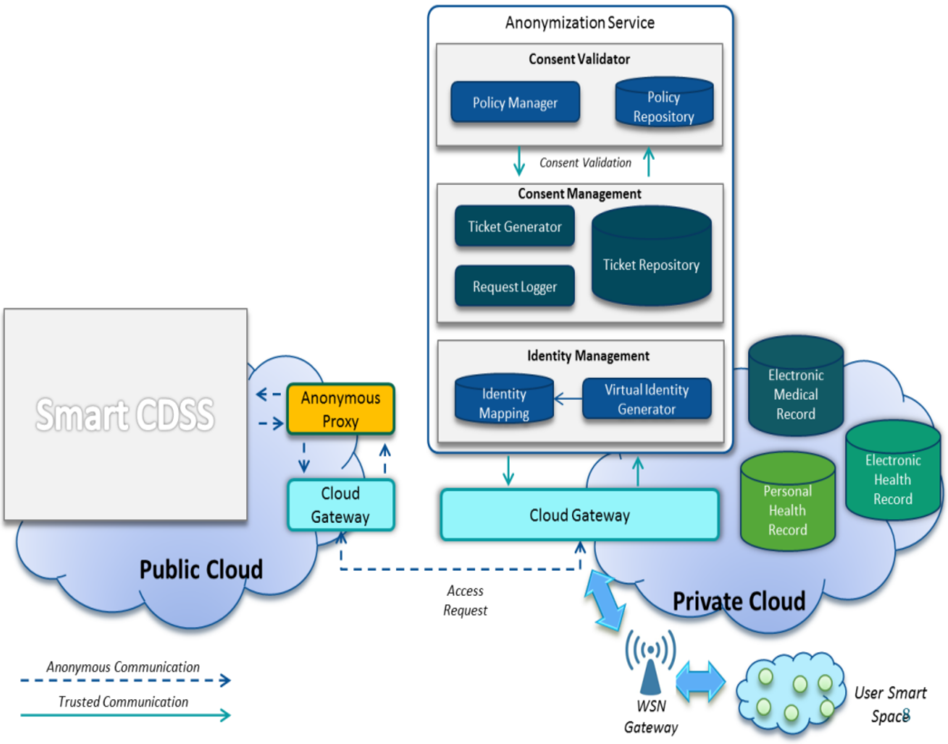


Figure 115 Proposed System

- **Identity Management**
  - Identity management is responsible for de-identifying the patient information. It generates pseudo identity (i.e., virtual identity) for each patient. It then securely stores the mapping between the virtual identities and real identities. These mapping are only know to identity management component. This ensures that whenever SmartCDSS requests data access its real identities can only be swapped with virtual identity by identity management component. Identity management is composed of following two sub components

- **Virtual Identity Generator**
  - It is a secure random string generator which generates randomized identity for each request. This ensures that for a single patient new pseudo random identity is generated on each request to SmartCDSS's services. This restrains the capability of malicious cloud service provider to link SmartCDSS request to a particular diabetic patient.

- **Identity Mapping**
  - Identity mapping is a repository that securely stores the mapping between the virtual identities and real identities of diabetic patients. These mapping are stored in encrypted form to ensure that even if anonymization service is compromised attacker cannot gain access to mapping repository.

- **Consent Management**
  - Consent management is responsible for enforcing access control polices during data access request to cloud storage. It not only enforces access control policies by also ensures that requests are generated by authorized subscriber and components

- **Ticket Generator**
  - Ticket generator is responsible for evaluating the access control policy and making decision on the outcome of access control policy evaluation. If policy is evaluated successfully a valid access ticket is generated which is digitally signed by Consent Management, otherwise access request is rejected restraining unauthorized data access to the outsourced data. Access ticket contains information about the data access request. It also encodes the validity of the access ticket to ensure that it cannot be used multiple times to prevent reply attack.

- **Request Logger**
  - Request Logger logs each data access request to ensure proper audit trail of each data access request. Request logger logs time stamp at which request is generated, which resource is requested in data access request, information of the requester (i.e., subscriber information) and outcome of access control policy evaluation. This detailed log of data access request ensures that cause of failure / malicious activity can be effortlessly traced by a system administrator, in case of any adverse event (i.e., security attack, system failure).

- **Ticket Repository**
  - Tickets generated by Ticket Generator are persisted in secure Ticket Repository. Persisted tickets are used to de-anonymized the data once request is processed by SmartCDSS in public cloud. This ensures that in trusted domain of private cloud processed data can be linked with patients in order to execute SmartCDSS's workflows properly.

- **Consent validator**
  - Cloud service provider is an untrusted entity. To ensure authorized data access and request execution rights, SmartCDSS cannot rely on cloud service provider – it can authorized malicious subscriber to access outsourced data and compromise privacy of patients. Consent validator is responsible for governing access to compute resources public cloud (i.e., SmartCDSS services) and outsourced data persisted in public cloud storage service.

- **Policy Manager**
  - Policy manager is responsible to evaluate access control policy on each data access request. This ensure that whenever subscriber seeks access to the outsourced data its access privileges are evaluated and access is granted according to its access right of the outsourced data.

- **Policy Repository**
  - Since, cloud service provider is not a trustable entity, access control policies need to be evaluated within trusted domain. For that access control policies are stored in

policy repository. This ensure that system administrator can properly manage them, and if there is a change in any access control policy it can be realized and enforced immediately.

- **Overall System Flowchart**

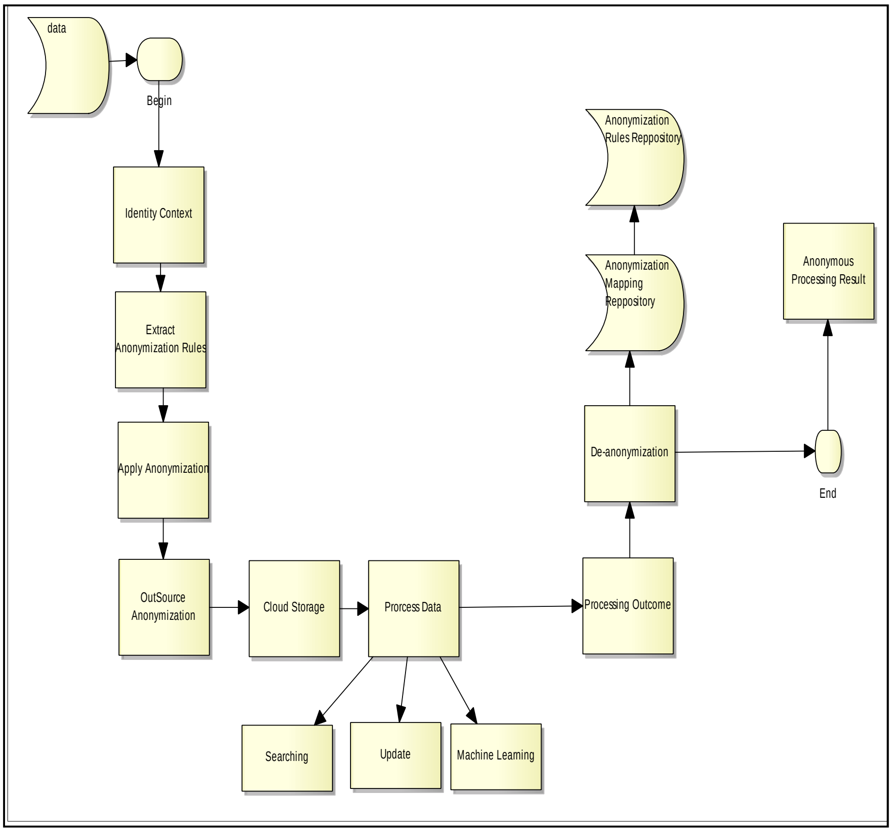


Figure 116 Anonymization service Flowchart

DETAIL DESIGN OF SC3 SERVICES

- In this chapter, the detailed design of SC3 services is explained.
- The design for services: 1) LDSS for Wellness, 2) LDSS for Chronic Disease, 3) Smart CDSS for Head and Neck Cancer, and 4) Cloud Computing and Security.
- The details design diagrams are given below with explanation.
- The following sections will present each service design in details.

6.1. LDSS for Wellness

- **Use Case Diagram**
  - Use case diagram is shown in Figure 117, where as its description is given below.
  - **Actors:**
    - **User:** User is the main entity of the LDSS for Wellness service that provide data using social media, sensors, and profile. Then user is the main entity of consuming the services LDSS for Wellness is providing.
    - **Wellness Service Manager:** Wellness service manager is responsible for brokering and providing different services to the user based on their requests.

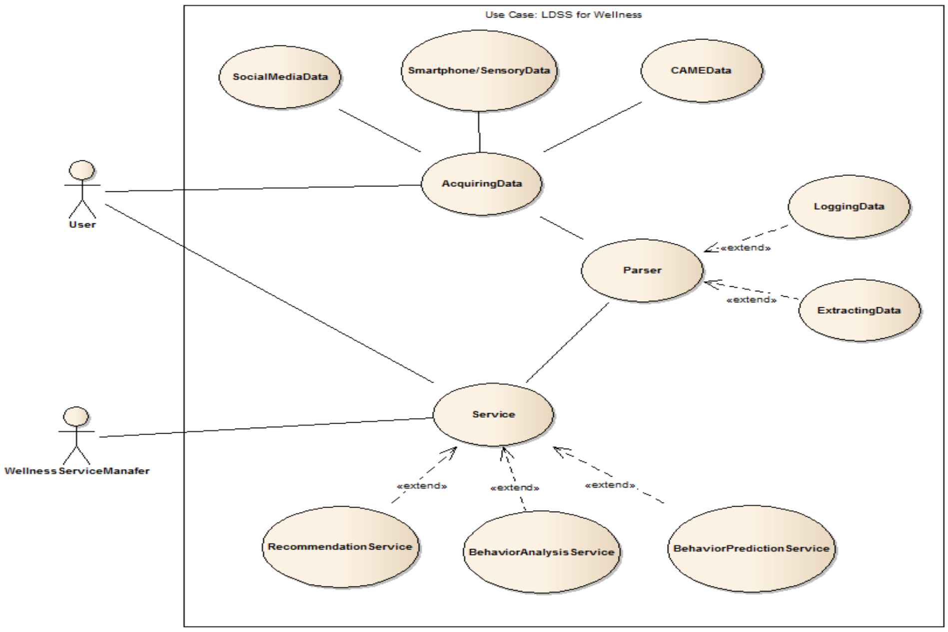


Figure 117 Use case diagram of LDSS for Wellness

  - **Use case Description**
    - **AcquiringData:** AcquiringData use case listen to the data coming from different sources. Then forward this data for logging to Parser.

- **Smartphone/SensoryData:** This use case acquires data coming from smartphone and sensors deployed in the environment and send this to AcquiringData use case.
- **SocialMediaData:** This use case acquires data from Social Media and forward it to AcquiringData use case.
- **CAMEData:** This use case acquires data from CAME (the high level activities) and forward it to AcquiringData use case.
- **Parser:** Parser is responsible to handle the incoming data from AcquiringData us case for logging in Life Log and outgoing data from life log to be used for different services using the Service use case.
- **LoggingData:** This use case is responsible for logging the incoming data in the life log.
- **ExtractingData:** This use case is responsible for extracting the required data from life log to be used by different services.
- **Service:** This use case interacts with both User and Service Manager. It is responsible for collecting the requests from user and activating the appropriate service based on user request. It is also responsible to provide appropriate data to the services which is extracted from life log.
- **RecommendationService:** This use case is responsible to provide user with recommendation services based on their daily life activities data.
- **BehaviorAnalysisService:** This use case provide user with behavior analysis services based on their daily life activities which are stored for a longer duration.
- **BehaviorPredictionService:** This use case is responsible to provide user with behavior prediction services based on user current situation which is result of user longer durations daily life activities and then project and predict appropriate behavior for user.

- **Sequence Diagram**
  - Sequence diagrams of LDSS for Wellness are described as follows. These diagrams show the interaction between different objects of the system. LDSS for Wellness overall sequence diagram is shown in Figure 118.
  - **LDSS for Wellness overall Sequence Diagram**
    - User performs different activities and social interactions which are provided as input to the system based on the nature of user interaction. This activate appropriate interaction component based on the interaction performed by the user.
    - HAR gets the activity information from user when user perform any activity using activity(). Which is processed by HAR and then forwarded to CAME for further processing as well as in logged in Life Log repository using logData().
    - CAME gets set of activities from HAR using activity(set) and use these for processing and high level activity recognition. After processing, the high level activities are then logged in Life Log using logData() method.
    - Social Media Interaction receives user interaction information on social media using interaction() method and process the information using developed algorithms. After processing, the meaningful information is logged in Life Log using logData() method.
    - User activate different services based on user interest/needs using serviceRequest() or recommendationRequest() method.

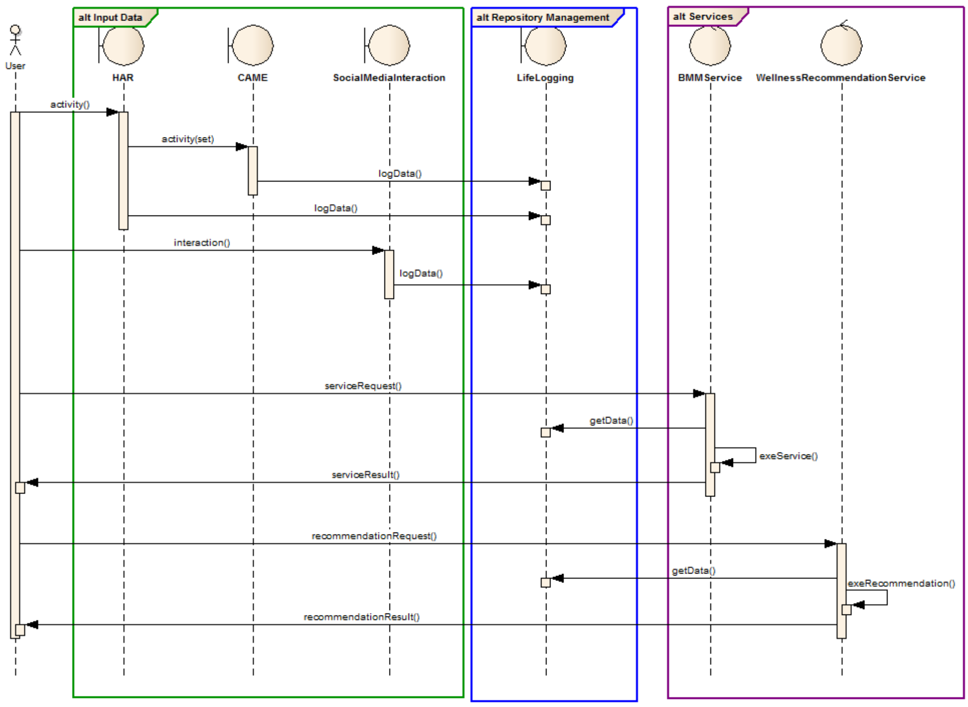


Figure 118 Overall sequence diagram of LDSS for Wellness

- **Class Diagram**
  - LDSS for Wellness class diagram (given in Figure 119) shows how the different classes are related with each other.
  - How the data is collected from different mediums and processed.
  - HAR,CAME, and SocialMediaInteraction classes process the incoming data and after processing they log the meaningful data in life Log.
  - LogData class of parser class is responsible for logging the incoming meaningful data in life log and the ExtractData class is responsible for extracting the data from Life Log based on the user request for service.
  - Service class is mainly responsible for requesting of extracting data from Life Log and activating appropriate service based on user request.
  - BMM service is activated for the purpose to analyze and predict user behavior.
  - Recommendation service is activated for the purpose to generate recommendations for user based on his/her current situation and existing data.

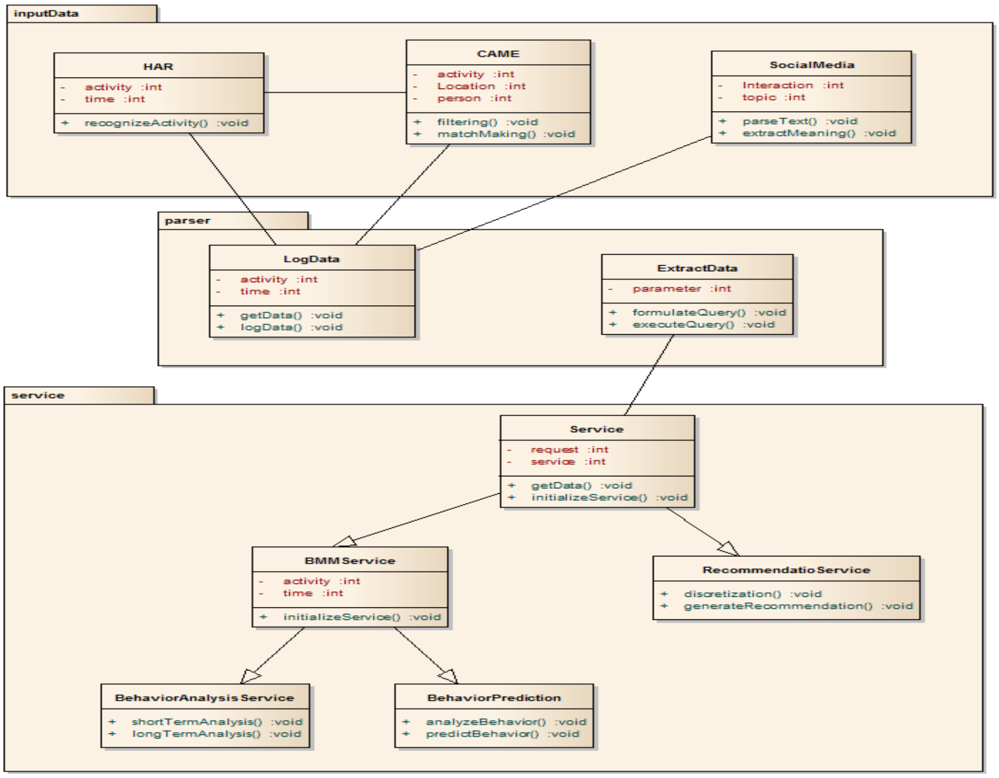


Figure 119 Class diagram of LDSS for Wellness

- **Component Diagram**
  - The component diagram of LDSS for Wellness shows the different components and their subcomponents interactions with each other. These are explained as follows:

  - **HAR**
  - This component is used to provide user daily life activities information, user diet information, and his actions. It consists of subcomponent to collect and recognize activities that are as follow:
    - Accelerometer based activity recognition.
    - Smartphone based activity recognition
    - Voice based activity recognition
    - Video based activity recognition
    - Diet information recognition

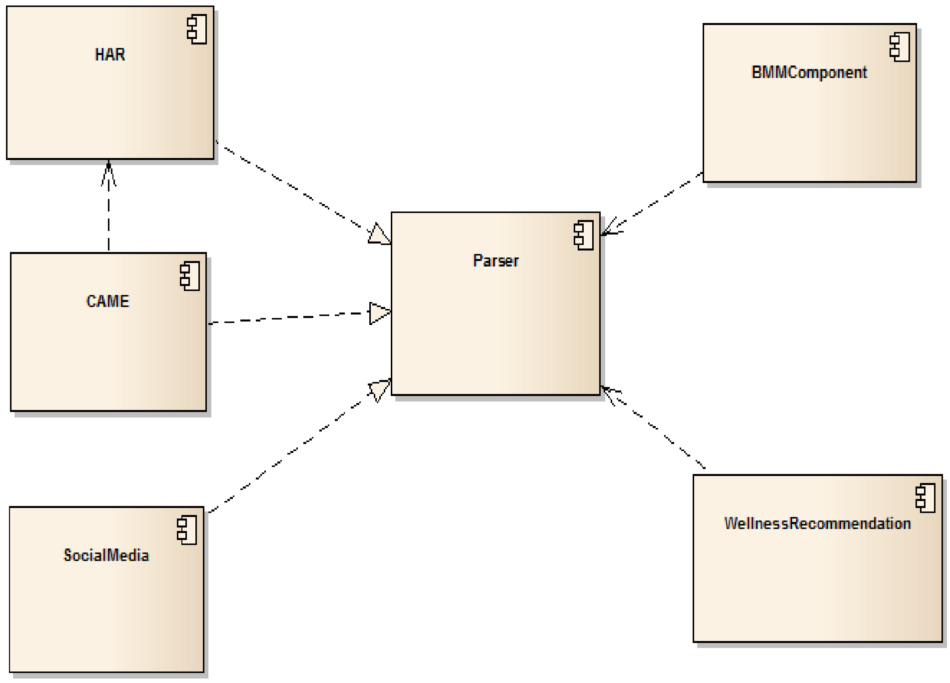


Figure 120 LDSSW4 Component diagram of LDSS for Wellness

- **CAME**
- This component is used to convert low level activity information coming from HAR into high level activities information which is usable for situation analysis and decision making. These activities are then stored in Life Log. This component consists of two subcomponents that are:
  - Context representation
  - Context Interpreter

- **Social Media Interaction**
- This component is used to process the data gathered from the social media. The information is then stored in Life Log which is later used by the LDSS for Wellness for the purpose of behavior analysis services and/or recommendation generation. This component also has different subcomponents:
  - Tweet Analysis
  - Trajectory Analysis
  - Interaction Analysis
  - Non Negative Matrix Factorization

- **BMM Component**
- This component provides the facility of behavior analysis and predictions based on the logged data in Life Log. This component also consists of two subcomponents:
  - Behavior Analysis
  - Behavior Prediction

- **Wellness Recommendation**
- This component is one of the main components of LDSS for Wellness and is used to provide wellness recommendations based on the logged data and current user situation. This component consist of three subcomponents:
  - Rough Set based recommendations
  - Case-based Recommendations
  - Bayesian Probability based Recommendations

6.1.1. Human Activity Recognition (HAR)

6.1.1.1. Audio Based Emotion Recognition

- **Use Case Diagram**

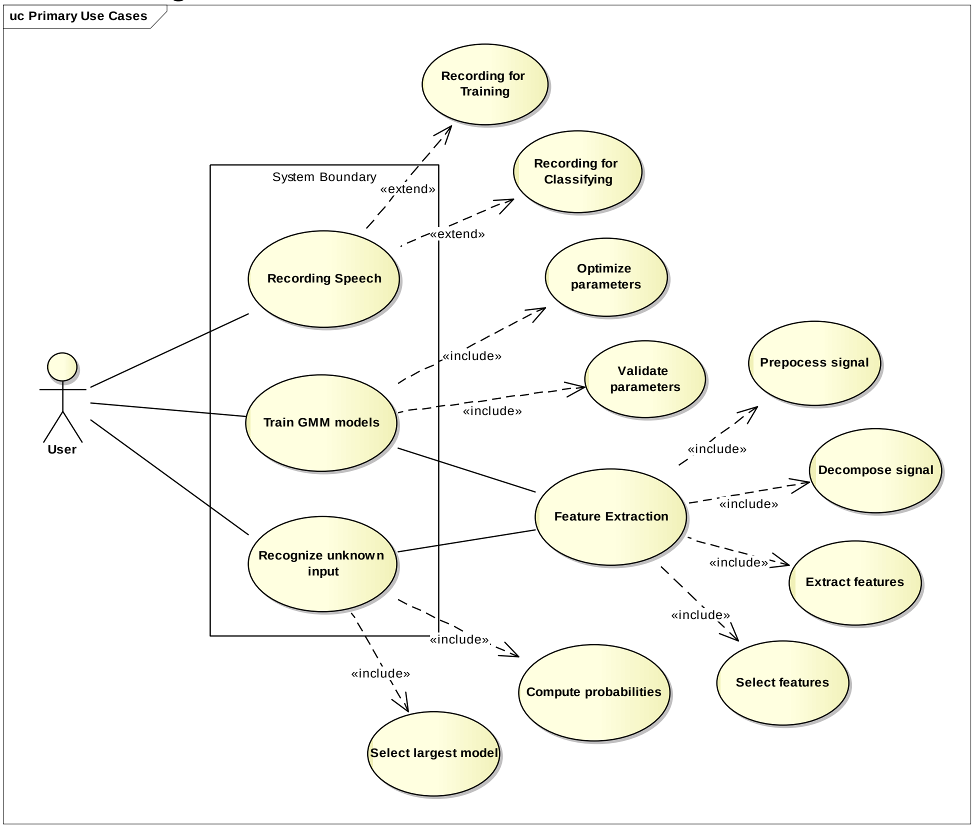


Figure 121 Use Case diagram of Audio-based Emotion Recognition

- Description: It interacts with the audio sensor and activates different components of the audio-based emotion recognition.
- Actors: Human beings interacting with the objects.

- Pre-conditions: Annotated data for training purpose
- Basic flow: The basic flow should be the events of the interaction with objects and everything is perfect; there are no errors, no exceptions. The exceptions will be handled to maintain the logs and alerts to the concern person.

- **Sequence Diagram**
  - Following training and testing phase sequence diagrams show the sequence of message between objects in an interaction.

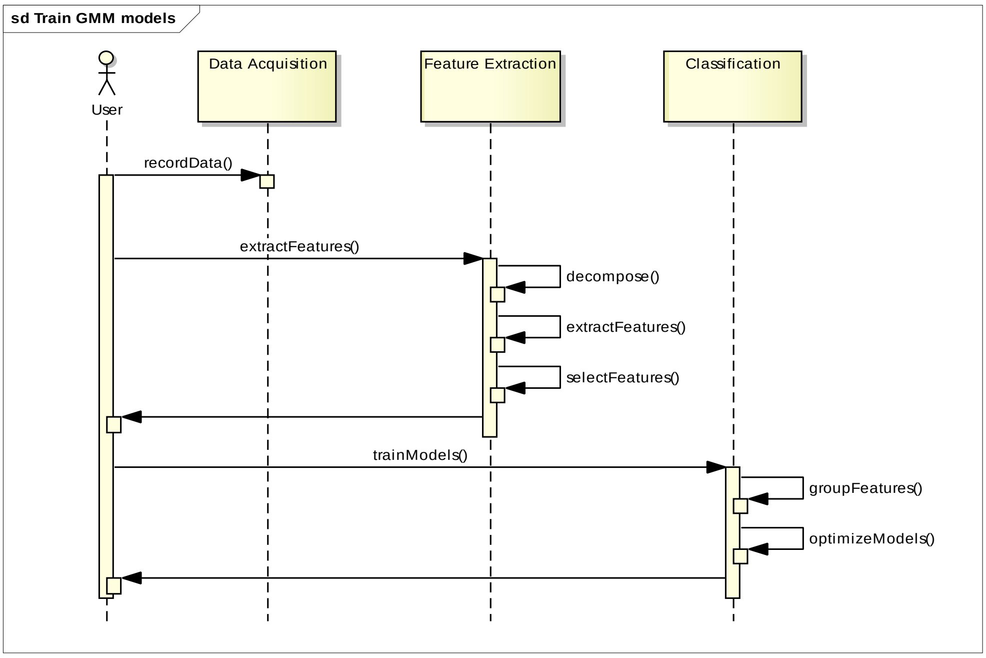


Figure 122 Sequence diagram of training phase of AER

- Initially, Data Acquisition is called to detect and record speech signal from audio stream.
- Main Process requests Feature Extraction to extract feature for all training speech signal. Inner processes are called to generate feature vector following Matching Pursuit algorithm.
- Finally, training stage of GMM algorithm is used to train models for all emotional classes.
- Following diagram show the recognizing phase sequence diagram.

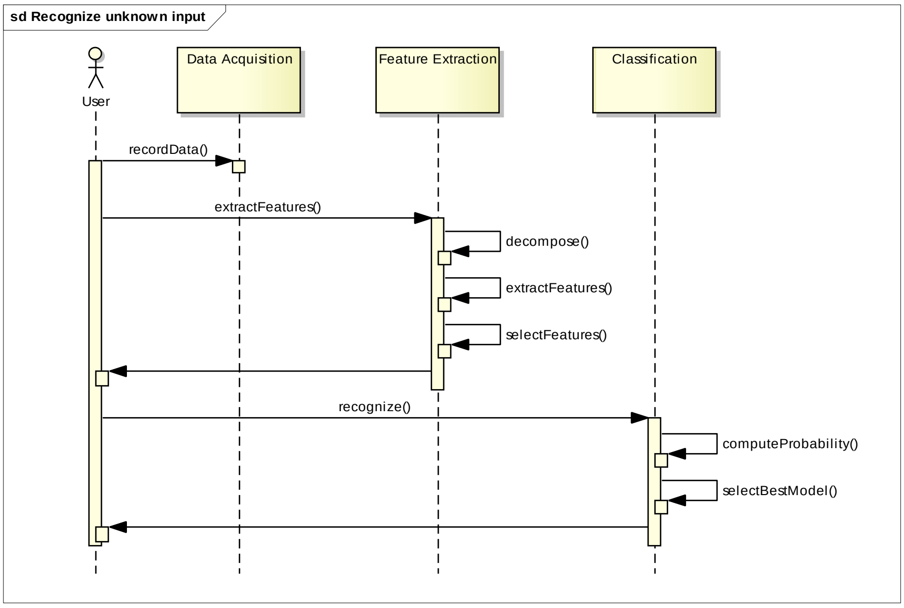


Figure 123 Sequence diagram of testing phase of AER

- Similar with training, Data Acquisition is called to record unknown signal.
- Feature Extraction is applied for unknown signal also.
- GMM classifier takes feature vector from Feature Extraction and computes probability with all models to find the model that has the largest probability value and then assigns according emotion as output label.

Class Diagram

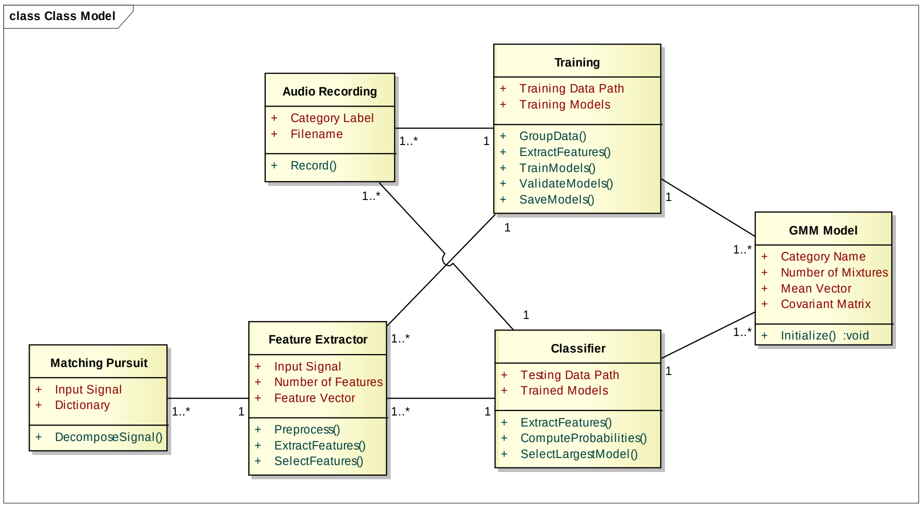


Figure 124 Class diagram of AER

- Similar with training, Data Acquisition is called to record unknown signal.
- Feature Extraction is applied for unknown signal also.
- GMM classifier takes feature vector from Feature Extraction and computes probability with all models to find the model that has the largest probability value and then assigns according emotion as output label.

▪ **Component Diagram**

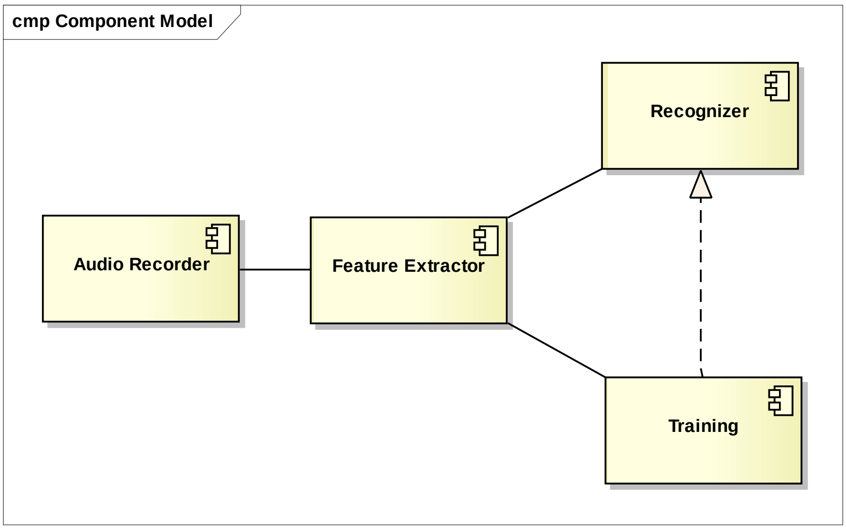


Figure 125 Component diagram of AER

- The component diagram shows different components and their relationships with each other. Mainly there are four main components and are explained below.
- Audio Recorder: record and preprocess audio, group training data into different emotion categories
- Feature Extractor: extract and select feature from training and testing data
- Training: use training data to generate Gaussian mixture model for each emotion category
- Recognizer: use trained model to predict emotion label from unknown input signal

6.1.1.2. Video Based Activity Recognition

▪ **Use Case Diagram**

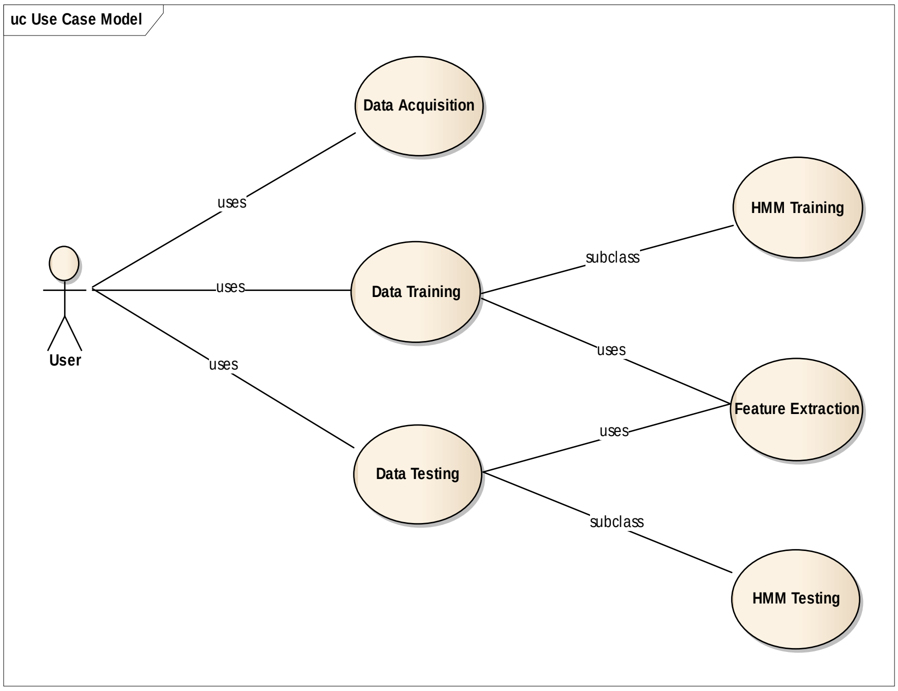


Figure 126 The use case diagram of human activity recognition system.

- **Description.** It communicates with the video camera and different component of activity recognition system are turned on.
- **Actors.** Human communicate with the 2D camera.
  - **Assumptions.** The common supposition is to employ machine learning techniques.
  - **Pre-conditions.** Processed and cleaned data for training purpose.
  - **Basic Flow.** The essential flow should be the incidents of the interaction with human, and there are no errors, no omissions that will be handled to preserve the logs and alerts to the corresponding person

- **Sequence Diagram**

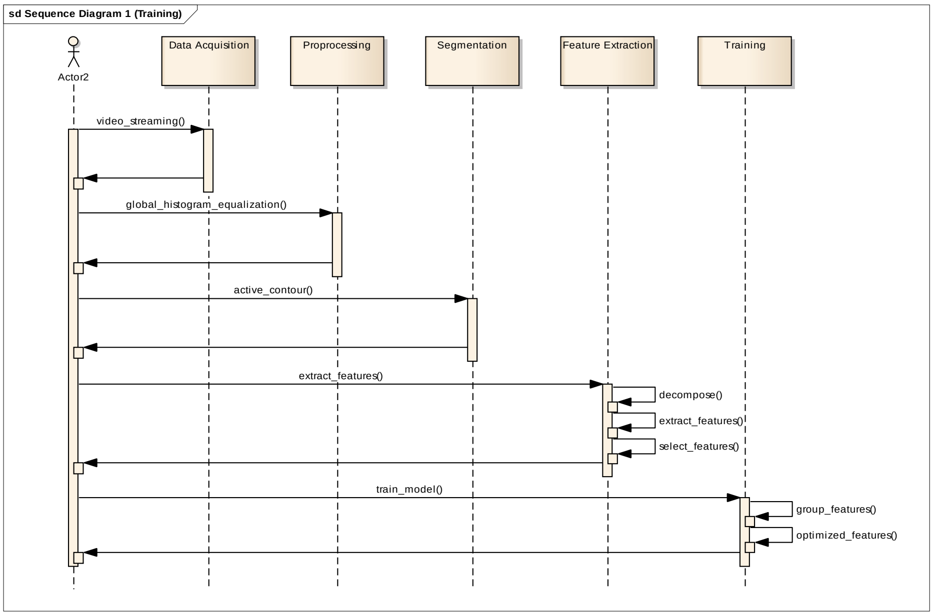


Figure 127 The sequence diagram of human activity recognition for training.

- At the beginning, in the main process the video of the human activity recognition is streamed and cleaned from noise at data acquisition stage.
- Some environmental effects such as lighting effect can be diminished at the preprocessing stage via Global Histogram Equalization (GHE) technique.
- An un-supervised technique such as active contour is employed to segment the human body automatically from the video frame at the segmentation stage.
- A robust technique called wavelet transform is employed to extract features from the segmented body.
- The system will be trained with suitable activity labels.

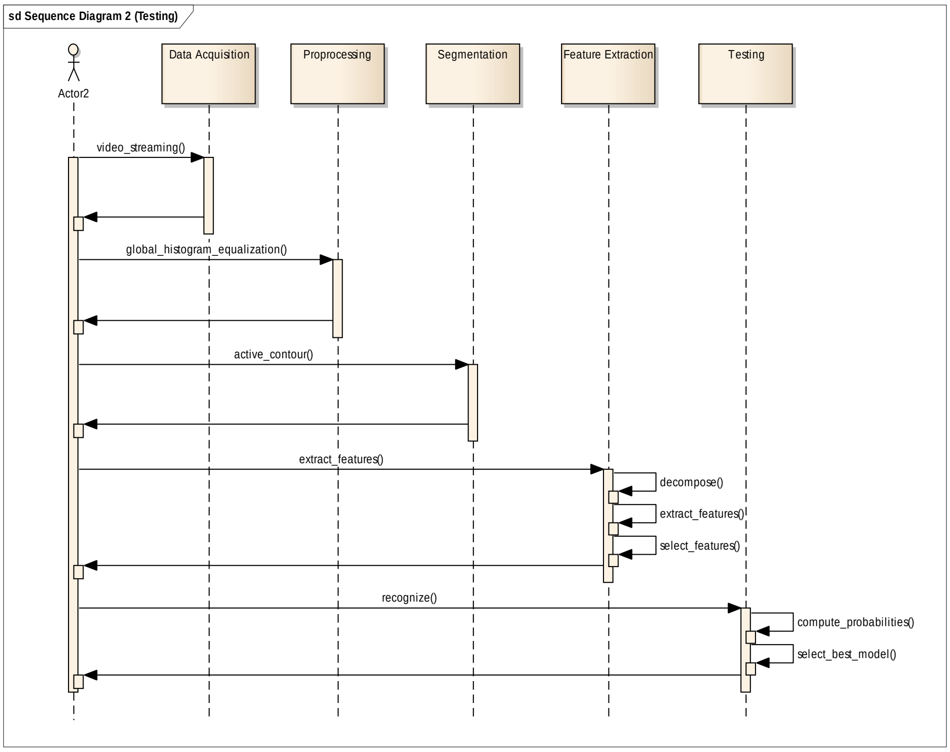


Figure 128 The sequence diagram of human activity recognition system for testing.

- In testing (recognition) phase, the streamed video frames are cleaned from environmental facts and then the human bodies have been segmented from individual activity frame.
- The features has been extracted and selected with the help of wavelet transform.
- The further activities have been recognized according to the trained labels

- **Class Diagram**

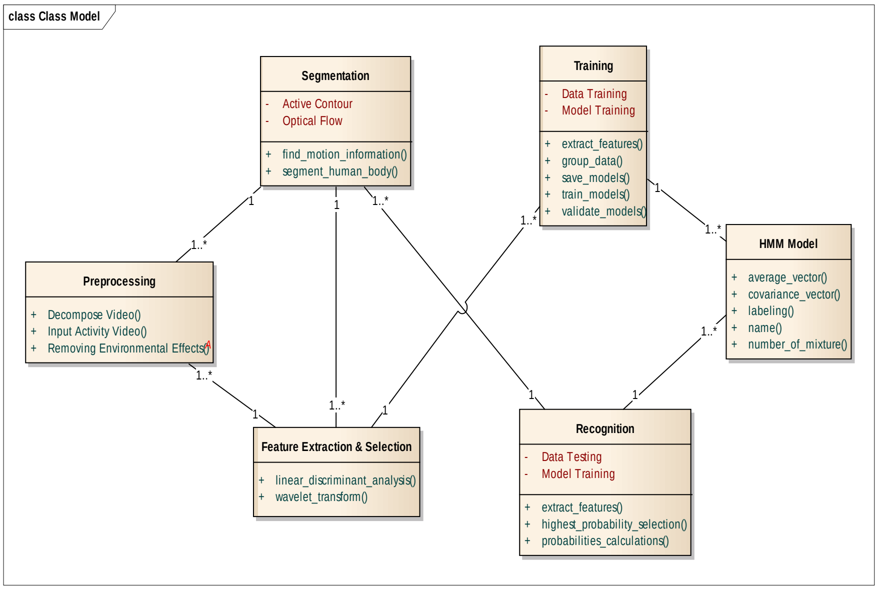


Figure 129 The class diagram for human activity recognition system.

- The class diagram shows how different classes related with each other to provide appropriate decision of activity recognition.
- In preprocessing class, the illumination and lighting effects are diminished to increase the recognition accuracy, using techniques like morphological filters, homomorphic filters, or median filters.
- In segmentation class an unsupervised segmentation technique such as active contour has been exploited for human body segmentation and the motion information between the two consecutive frames has been found by employing optical flow.
- The feature extraction module deals with extracting distinguishable features for each expression and quantizing each of them as a discrete symbol. Therefore, in feature extraction, wavelet transform has been used to extract the useful global and local features. Moreover, for feature selection a well-known statistical approach named linear discriminant analysis has been exploited.
- In recognition class, a classifier is used to train and to generate a label for the human activity recognition contained in the incoming video data. Among all of the classifiers, hidden Markov model (HMM) can frequently be employed for sequential data such as activity recognition. HMM is trained and tested to recognize incoming activity frames.

- **Component Diagram**

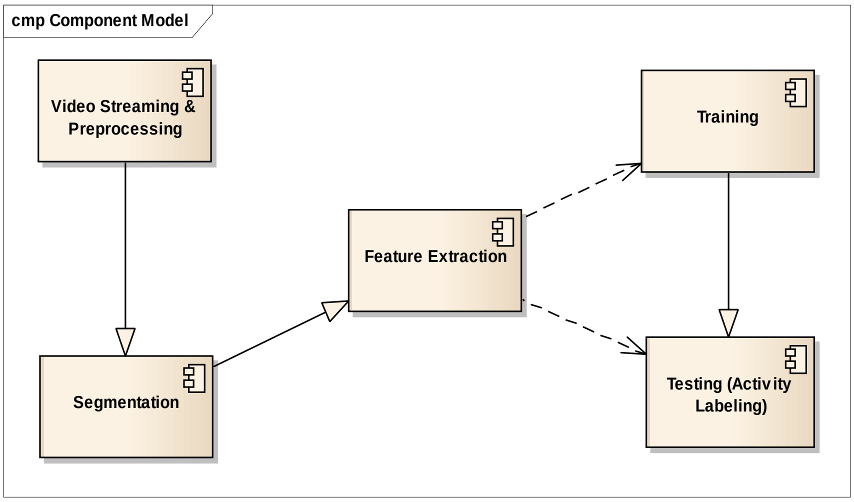


Figure 130 Component Diagram of Activity Recognition System.

- **Video Streaming & Preprocessing component**
  - This component loads the video activity and divides it into number of frames and diminishes some environmental effects such as lighting effects.

- **Segmentation Component**
  - Accurate human body segmentation techniques are required in order to achieve best accuracy of recognition; therefore in this component an unsupervised segmentation has been exploited to segment the human body automatically.

- **Feature Extraction Component**
  - The accuracy of recognition module is completely reliant on the extraction of good features that means, the effectiveness of the extracted features will also affect the accuracy of recognition module. Therefore, in this component some informative features have extracted in order to make the decision for classifier.

- **Training Component**
  - After feature extraction, in this component, a classifier is trained according to suitable activity labels that further can be used for activity classification.

- **Testing component**
  - In this component, future activities have been recognized based on the trained labels.

6.1.1.3. Multimodal Sensor Fusion

- **Use Case Diagram**
  - Following figure shows the interaction of the component to construct the activities and emotion profile.

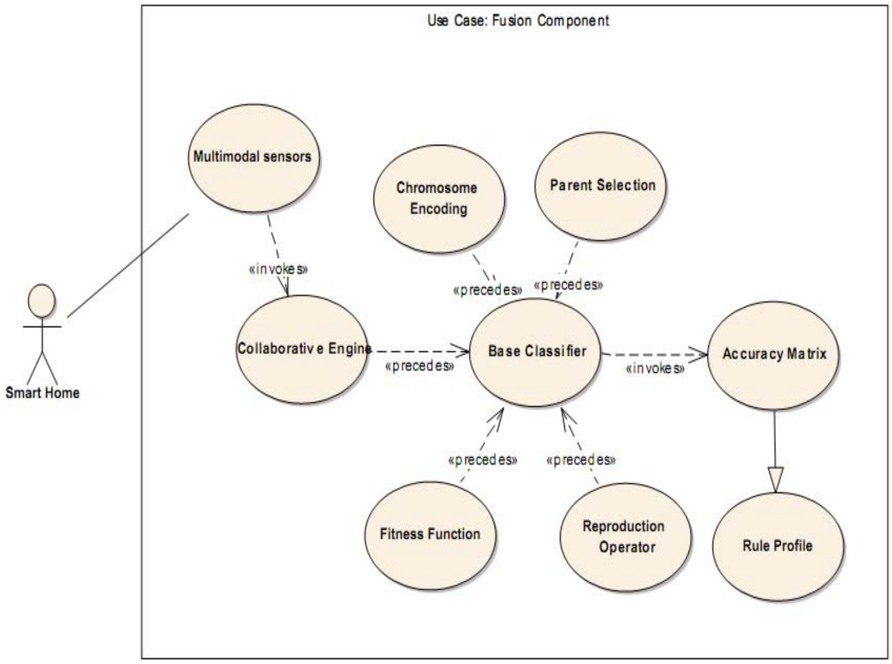


Figure 131 Use Case diagram of Multimodal Sensor Fusion

- **Use case Description**
  - **Multimodal Sensors:** This use case receives the multimodal sensory information from deployed sensors.
  - **Collaborative Engine:** This use case is responsible to transform the heterogeneous information into required format.
  - **Base Classifier:** This use case is used to train the model to obtained the optimized rule set.
  - **Chromosome Encoding:** It is responsible to formulate the activity recognition problem into genetic problem.
  - **Parent Selection:** The responsibility of this use case is to select the best parent for the crossover operation.
  - **Fitness Function:** It evaluate the fitness of each chromosome and assign score accordingly.
  - **Reproduction Operation:** This use case reproduces the solution by applying crossover and mutation operator with certain probability.
  - **Accuracy Matrix:** This use case assigns the importance to the rules for getting more accurate results.
  - **Rule Profile:** This use case maintains the rules profile for recognition phase.

▪ **Sequence Diagram**
- Following training and testing phase sequence diagrams illustrate the chronological sequence of messages between objects in an interaction:

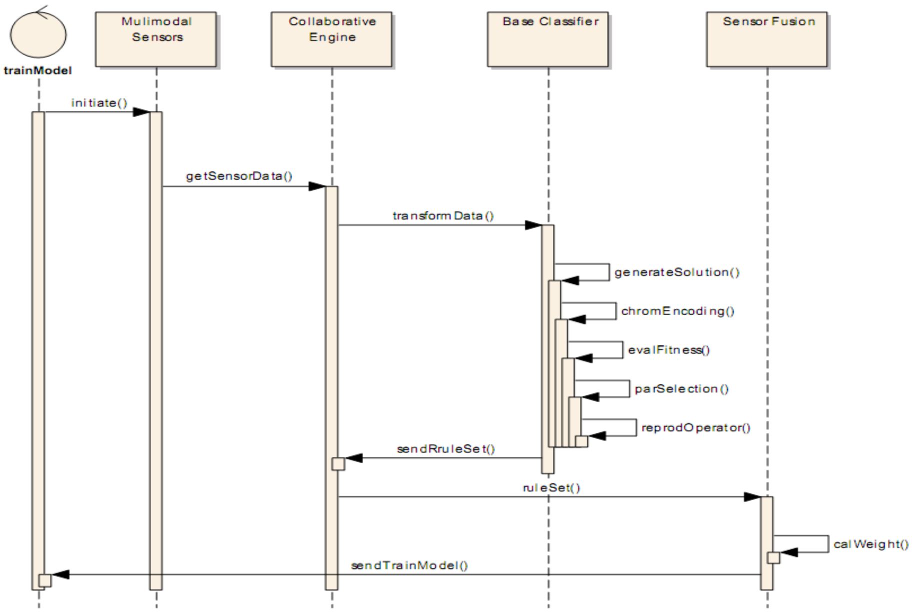


Figure 132 Training sequence diagram of the module

- Initially, collaborative engine receives the training data from the multimodal sensors and transform it into the specific input format for training the model.
- Base classifier is interacts with the collaborative engine and receive the transformed data and run the evolutionary technique Genetic Algorithm (GA) to extract the rule profile.
- Accuracy matrix assigns the importance to rule and help to make decision during recognition phase.

Following diagram shows the recognition phase sequence diagram

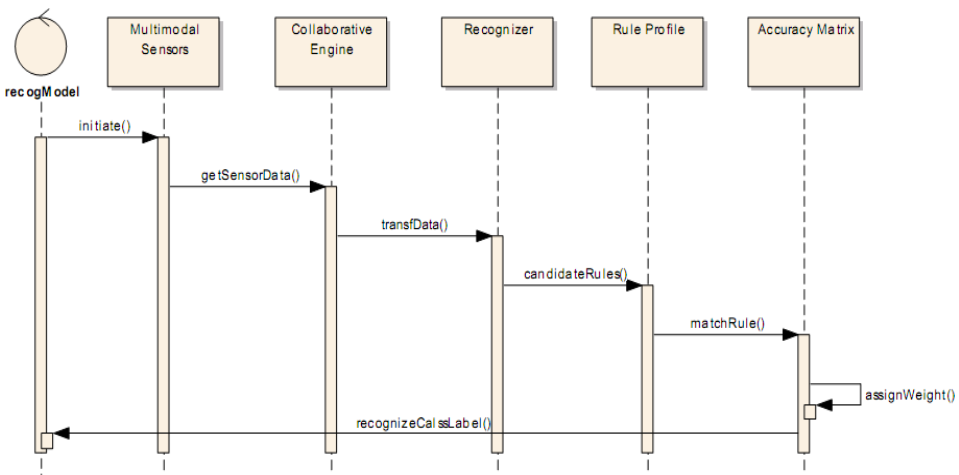


Figure 133 Recognizer sequence diagram of the module

- In recognition phase, data acquisition mode grabs the data and same process is repeated in the transformation mode.
- After transformation mode, rule matching is done over the rule profile and importance is figure out through accuracy matrix.
- Finally, emotions and activity labels are assigns on the current situation.

▪ **Class Diagram**

- It consists of the main class that interacts with the internal classes of the component.
- In order to decouple the code we made each module implementation separately and interact with through the proper interfaces.

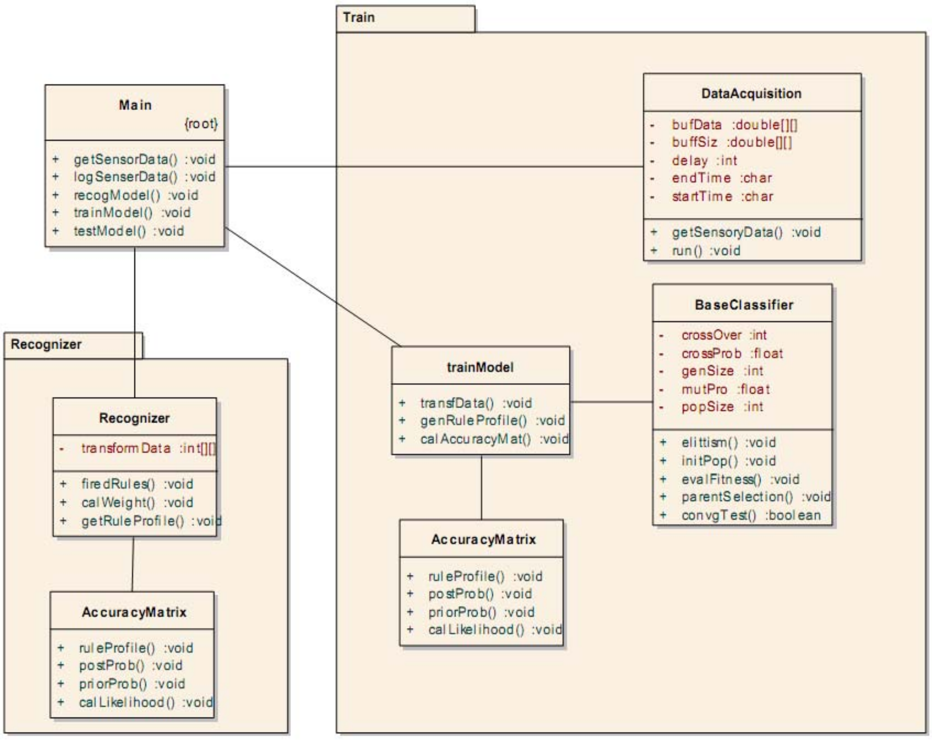


Figure 134 Class diagram of the overall module

▪ **Component Diagram**

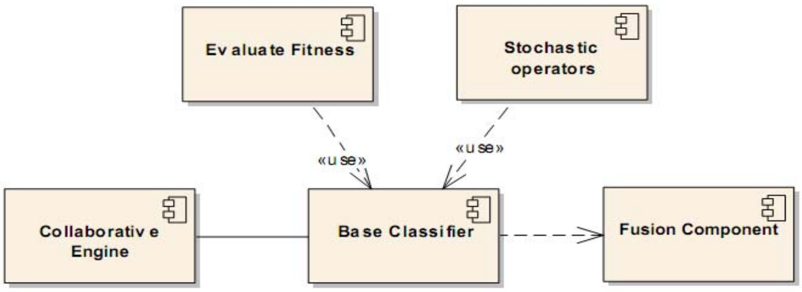


Figure 135 Component diagram of fusion engine

- The component diagram shows different components and their relationships with each other. Mainly there are three main components and are explained below.
- Collaborative Engine: transform the heterogeneous data formats into uniform format.
- Base Classifier: extracts the rule profile by using stochastic operators and fitness function.
- Fusion Component: calculate the accuracy matrix for assigning the weights to extracted rules.

6.1.1.4. Emotion Recognition

- **Use Case Diagram**

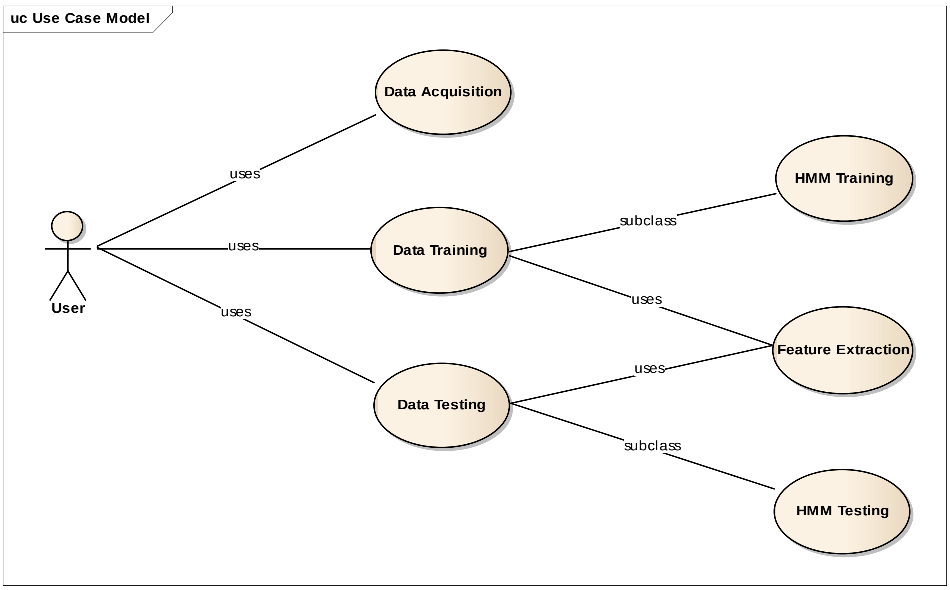


Figure 136 The use case diagram of automatic facial expression recognition system.

- **Description.** It communicates with the video camera and different component of FER system are turned on.
- **Actors.** Human communicate with the 2D camera.
  - **Assumptions.** The common supposition is to employ machine learning techniques.
  - **Pre-conditions.** Processed and cleaned data for training purpose.
  - **Basic Flow.** The essential flow should be the incidents of the interaction with human, and there are no errors, no omissions that will be handled to preserve the logs and alerts to the corresponding person

- **Sequence Diagram**

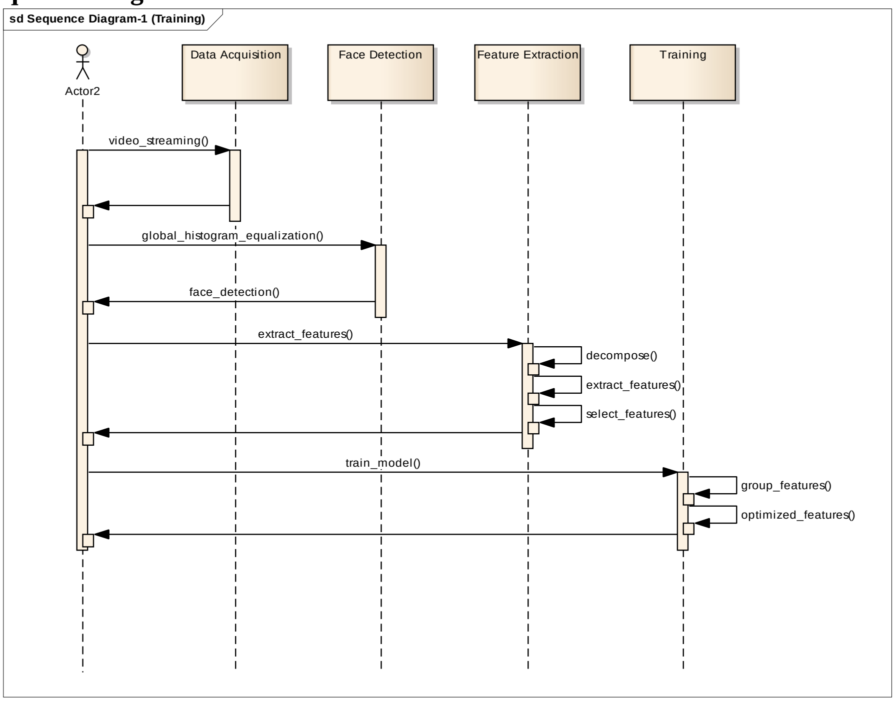


Figure 137 The sequence diagram of automatic facial expression recognition for training.

- At the beginning, in the main process the video of facial expression is streamed and some environmental effects such as lighting effect can be diminished at data acquisition stage.
- New techniques named Gray-level and Skin-toe are employed to detect the face from the video frame.
- A well-known statistical techniques; such as principle component analysis independent component analysis and discriminant analysis are employed to extract and select features.
- The system will be trained with suitable facial expression labels.

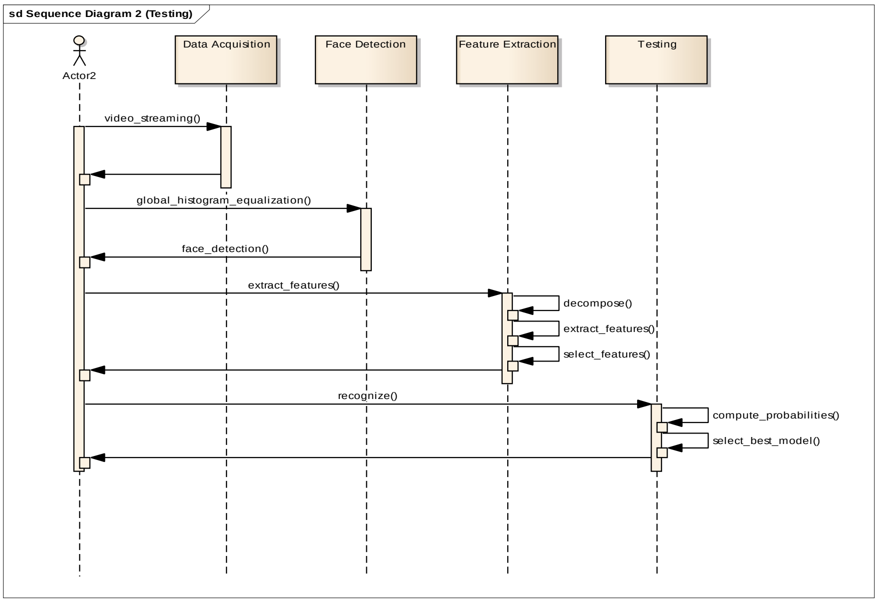


Figure 138 The sequence diagram of automatic facial expression recognition for testing.

- In testing (recognition) phase, the streamed video frames are cleaned from environmental facts and then the faces have been detected from each expressions frame.
- The features has been extracted and selected with the help of a linear classifier named principle component analysis and independent component analysis followed by linear discriminant analysis.
- The further expressions have been recognized according to the trained labels.

▪ Class Diagram

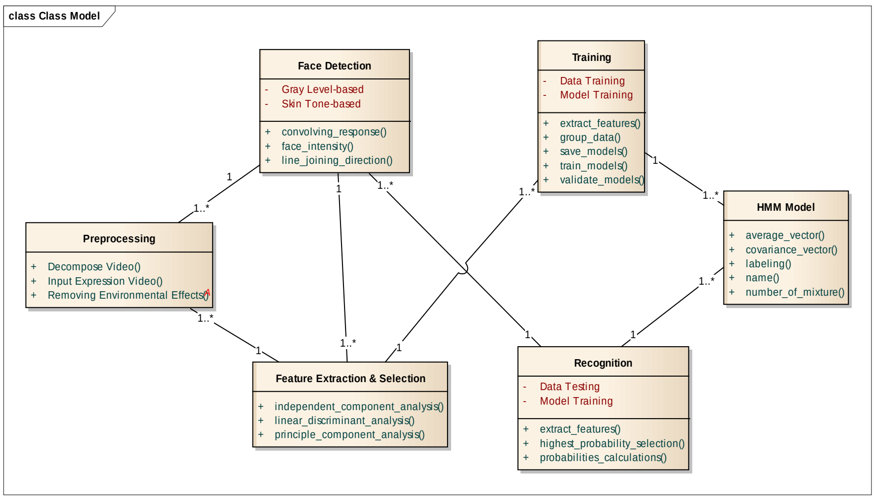


Figure 139 The class diagram for facial expression recognition system.

- The class diagram shows how different classes related with each other to provide appropriate decision of facial expression.
- In preprocessing class, the illumination and lighting effects are diminished to increase the recognition accuracy, using techniques like morphological filters, homomorphic filters, or median filters.
- Face contains most of the expression-related information, so, before investigation the expression, face must be detected first. An accurate facial expression recognition system requires automatic face detection which is considered the essential part of the facial expression system. So in face detection class the faces are detected and extracted first.
- The feature extraction module deals with extracting distinguishable features for each expression and quantizing each of them as a discrete symbol. Therefore, in feature extraction and selection some well-known statistical approaches are applied to extract and select informative features.
- In recognition class, a classifier is used to train and to generate a label for the human facial expression contained in the incoming video data. Among all of the classifiers, hidden Markov model (HMM) can frequently be employed for sequential data such as facial expressions. HMM is trained and tested to recognize incoming expression frames.

▪ **Component Diagram**

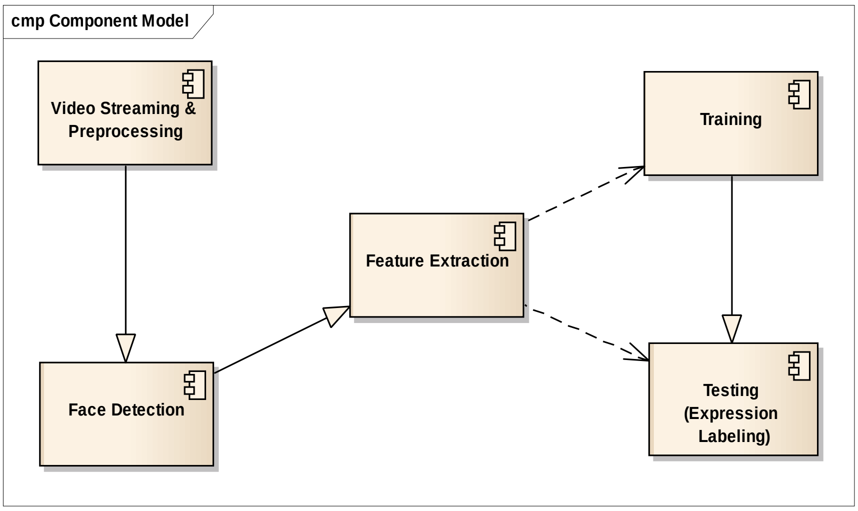


Figure 140 Component diagram of video based emotion recognition system.

- **Video Streaming & Preprocessing component**
  - This component loads the video of facial expression and divides it into number of frames and diminishes some environmental effects such as lighting effects.

- **Face Detection Component**
  - Face detection and extraction is an important step before investing the facial expres-

sion, therefore in this component a robust and an accurate has been employed to detect and extract the face from the video frames.

- **Feature Extraction Component**
  - The accuracy of recognition module is completely reliant on the extraction of good features that means, the effectiveness of the extracted features will also affect the accuracy of recognition module. Therefore, in this component some informative features have extracted in order to make the decision for classifier.

- **Training Component**
  - After feature extraction, in this component, a classifier is trained according to suitable expression labels that further can be employed for expression classification.

- **Testing component**
  - In this component, future facial expressions have been recognized based on the trained labels.

6.1.2. Context-aware Activity Manipulation Engine

- LDSS for Wellness takes the input form CAME engine as well.
- Context-aware Activity Manipulation Engine (CAME) is one of the main components of SC^3 and also an inessential component of LDSS for Wellness.
- CAME is the process of inferring high level activities from low level activities recognized by different sensors.
- The component based framework architecture diagram of CAME and the information flow is given in main architecture of LDSS foe Wellness.
- The Activity Extractor component extract activity related information from XML and Text files.
- Then with the help of context information available in Knowledgebase and customized rules, CAME infer high level/actual activity performed by human body.
- Based on the activities performed, CAME also gives suggestions and makes decisions in different environment with the help of context information available.
  - For example; we have all the context/profile information (i.e. professor name, designation, current courses, class room no, and class timings) about a professor in the knowledgebase (ontology).
  - Now if Professor enters a class room on his class time, then the body, motion, location and video sensors will recognize that Professor has entered the class room at a specified time.
  - Then CAME using these information from the sensors and information available in the Knowledgebase infers that its lecture time of Professor.
  - So the system starts issuing commands for turning off class room lights, turn on computer and turn on plus scroll down multimedia (projector) in class room

- **Use Case Diagram**
  - The use case diagram of CAME is shown in Figure 141, where as its description is given below.
  - **Actors:**

- **Admin:User:** Admin:User is the main user of the system that is using CAME for activity analysis and decision making.
- **ActivityRecognitionEngine:** Is the main source of input to CAME. The activity information coming in into CAME is manipulated and decision is propagated to the consumers.

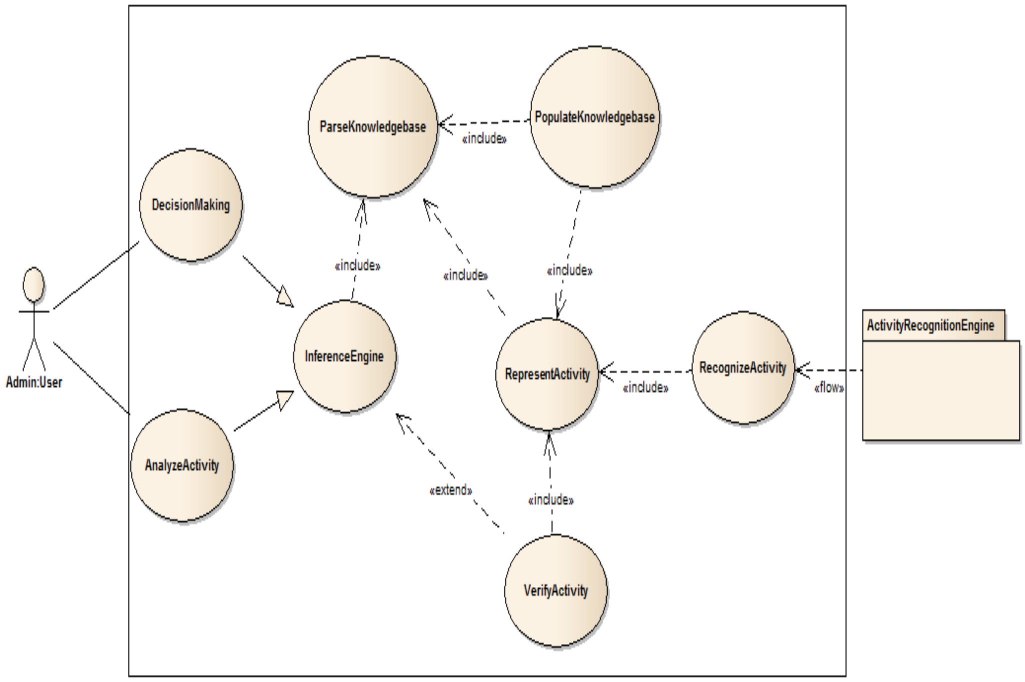


Figure 141 Use case diagram of CAME

- **Use case Description**
  - **RecognizeActivity:** This use case receives the incoming activities from the sensors and identifies its format.
  - **RepresentActivity:** This use case is responsible for representing the incoming activity from ActivityRecognizer in ontological structure while take help from VerifyActivity for verification of activity.
  - **VerifyActivity:** This use case is used to verify the incoming activity against the Knowledgebase for its consistency and existence.
  - **PopulateKnowledgebase:** It is responsible for logging the verified activity in the knowledgebase.
  - **ParseKnowledgebase: The** responsibility of this use case is to parse the knowledgebase for inference engine and its processing.
  - **InferenceEngine:** Inference engine involves the reasoning that include MatchMaking and filtering
  - **AnalyzeActivity:** This use case analyze the activity based on the reasoning and recognize the situation of user.

- **DecisionMaking:** This use case makes the decisions based on the reasoning which is performed in InferenceEngine.

▪ **Sequence Diagram**

- Sequence diagrams for CAME are described below that shows the interaction between different objects of the system for the achievement of objectives.

- **Activity Representation (Figure 142)**
  - When the activity is received in CAME then for its usage in Inference Engine and storage in Knowledgebase needs to be in Knowledgebase representational structure.
  - It calls the ActivityRecognizer using getActivityContents() and receive the activity.
  - It gets the representational structure from Knowledgebase by calling getRepresentation().
  - Which is executed by Parser and Knowledgebase is parsed for that. Then the activity is represented using representActivity().

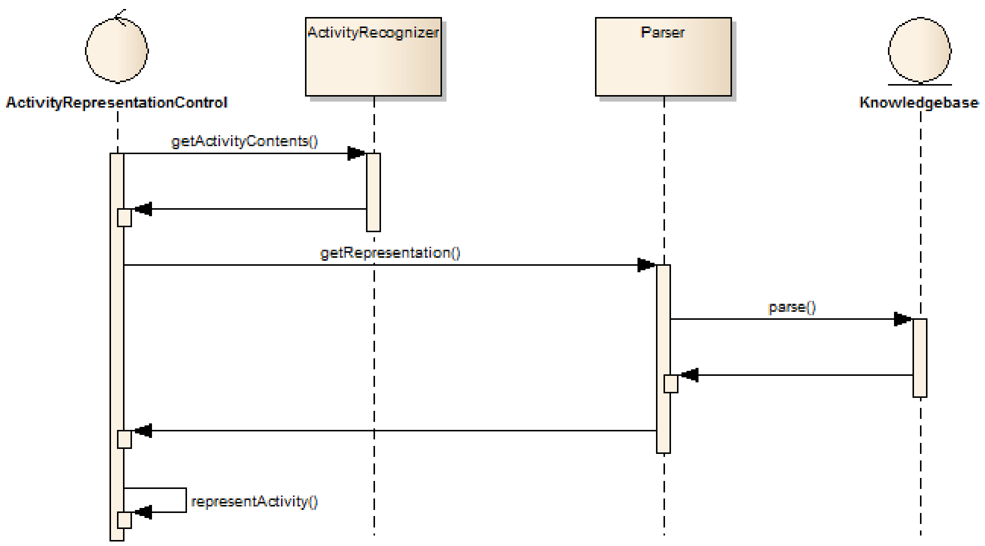


Figure 142 Sequence diagram of CAME Activity Representation

- **Analyze Activity (Figure 143)**
  - The activity is first extracted using extractActivity() from Knowledgebase and then forwarded to Inferencer using analyzeActivity().
  - Inferencer performs matchmaking and then apply rule using applyRules() extracted from Rules
  - Then the result is displayed using displayResult().

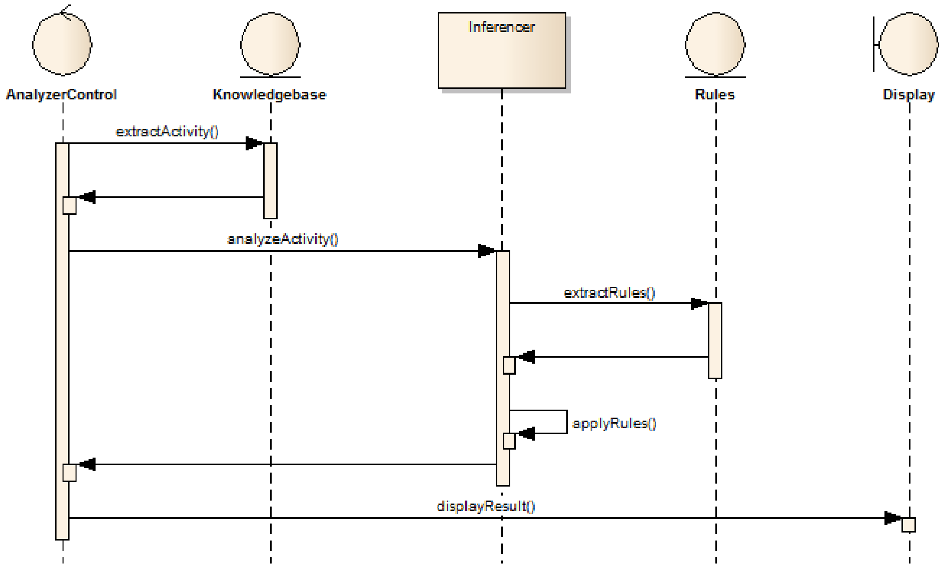


Figure 143 Sequence diagram of CAME Analyze Activity

- **Decision Making (Figure 144)**
  - The detected activity is first represented in the Knowledgebase representational structure and then the DecisionControl is activated.
  - The DecisionControl extract the relevant activities form Knowledgebase using matchmaking process.
  - Then the DecisionControl apply the rules extracted from Rules using applyRules().
  - The activity is then identified and decision is made. The decision is then displayed or propagated to DisplayDecision.

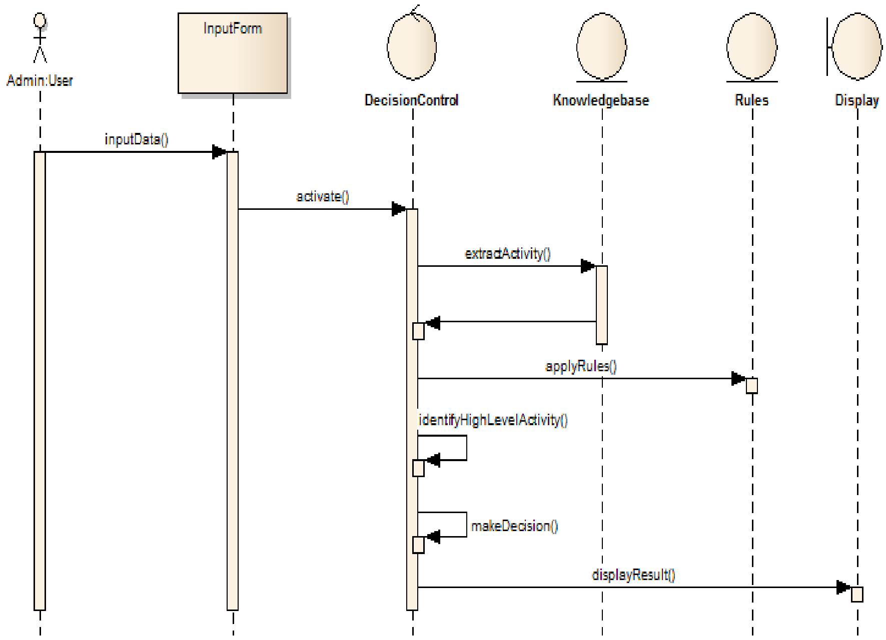


Figure 144 Sequence diagram of CAME Decision Making

- **Class Diagram**
  - Class diagram (Figure 145) of CAME shows the different classes and their relationships with each other.
  - Initially the ActivityRecognizer and ActivityRepresentation are loaded which work together for incoming activities.
  - Then the activity is Verification class is called for activity verification. The verified activity is then logged using PopulateKnowledgebase that uses Parser class for the job.
  - Inferencer class initiates the matchmaking and filtering for the new activity and makes the situation analysis. This is also responsible for decision making.

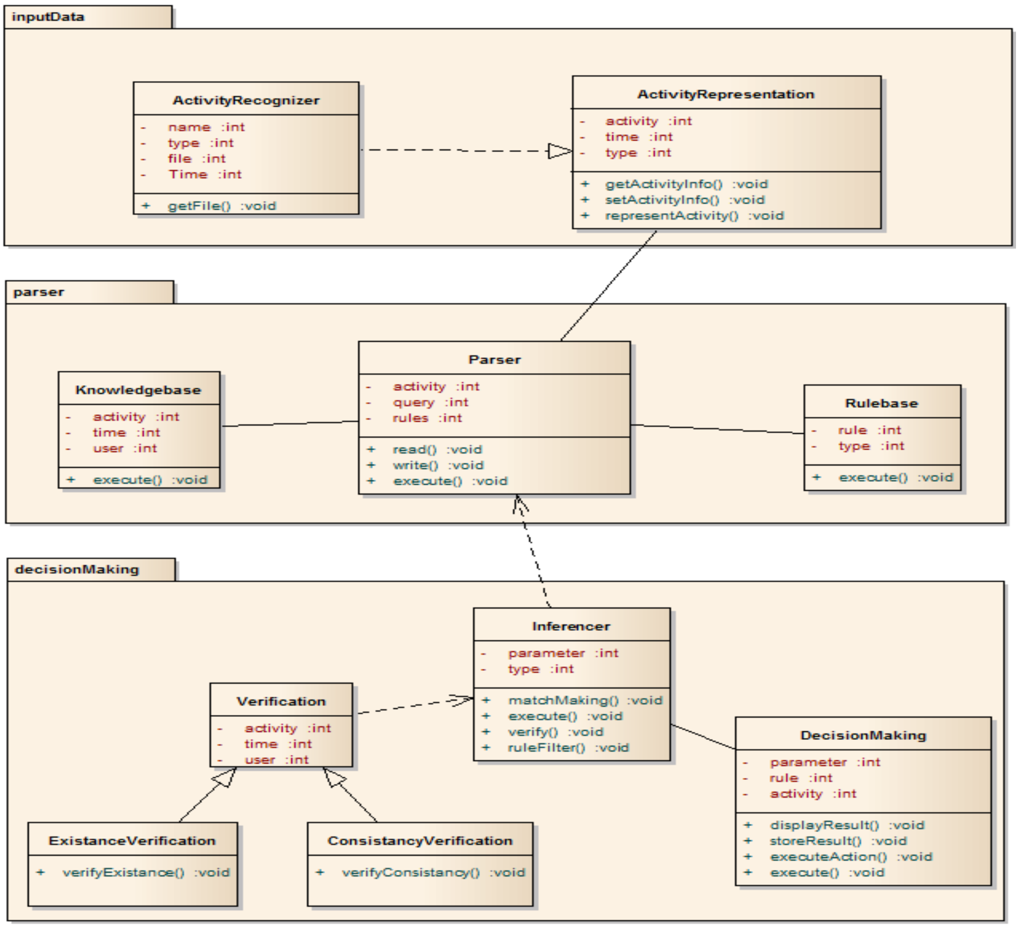


Figure 145 Class diagram of CAME

- **Component Diagram**
  - The component diagram of CAME (Figure 146) shows different components and their relationships with each other. Also it shows the subcomponents interaction, purpose, and their relationships with each other. Mainly there are four main components and are explained below.

  - **Activity Recognizer Component**
  - It is the main component of CAME that is responsible for collecting the activity information coming from different sources.

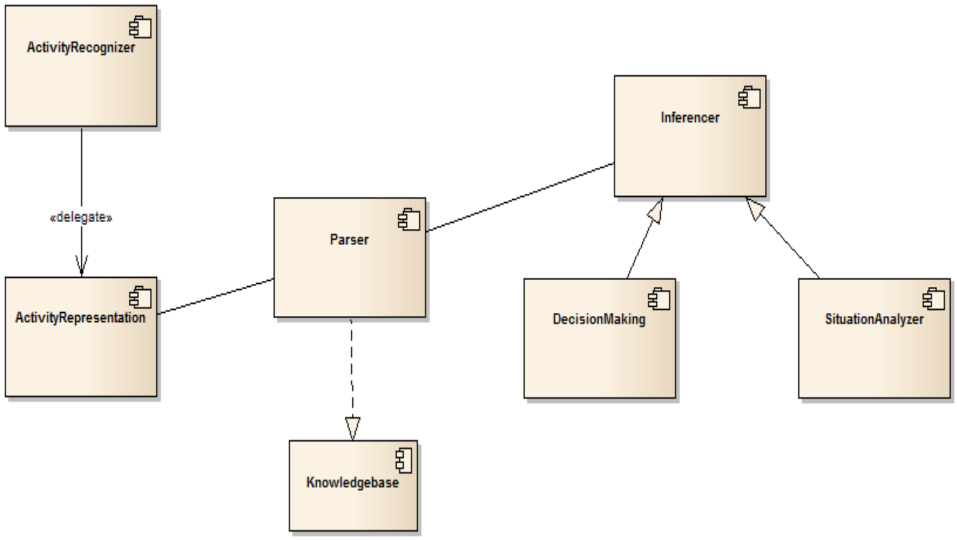


Figure 146 Component diagram of CAME

- **Activity Representation Component**
- It represents the incoming activities information in the ontological structure. It also uses the help of verification for verifying the activities consistency.

- **Parser Component**
- It is responsible for CAME components interaction with Knowledgebase for activity logging and extraction. It is the main component that is always connected with the Knowledgebase and fulfills the jobs coming for knowledgebase.

- **Inferencer Component**
- Inferencer contains the brain of CAME. It is the main component that performs the matchmaking and filtering. It uses the services of DecisionMaking and SituationAnalyzer for analysis and decisions.

6.1.3. Social Media Interaction

6.1.3.1. Tweet Analysis

- **Use Case Diagram**

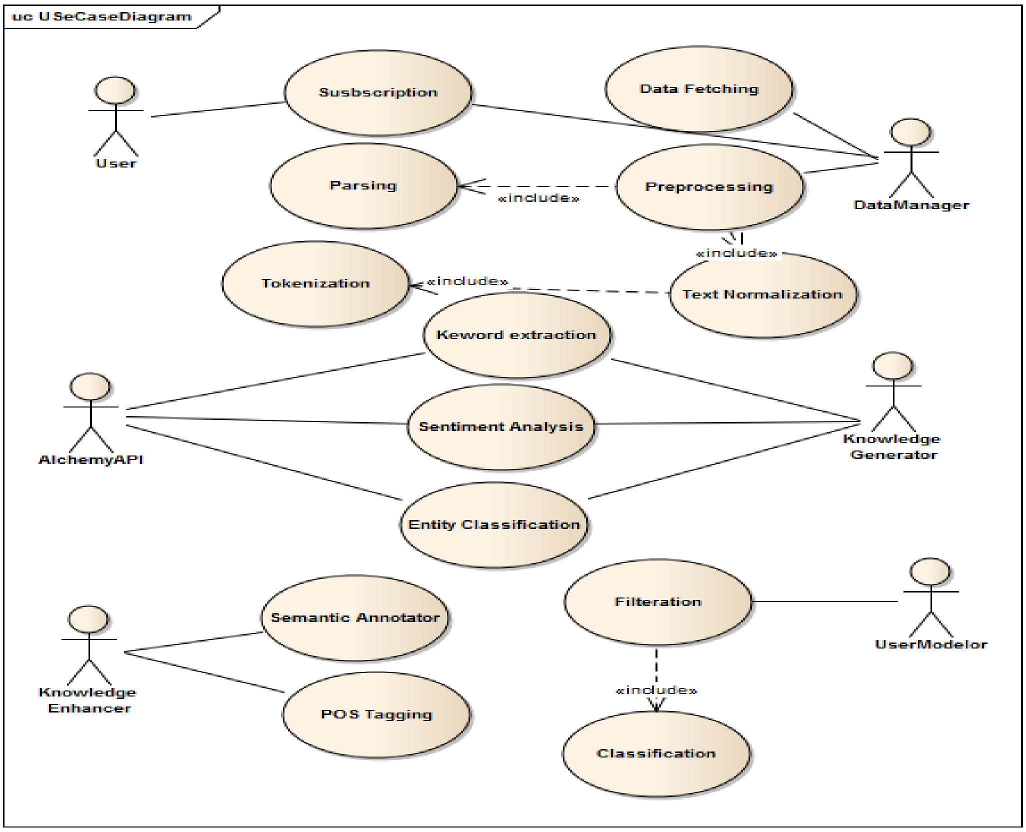


Figure 147 Use case diagram of Tweet Analysis for wellness

- **Actors:**
  - **Alchemy API**
    Third party component use to extract knowledge from tweets.
    Knowledge Enhancer
    Application component used to enhance knowledge of the system
  - **User Modeler**
    Responsible to generate personalized profile on the basis of information extracted from tweets.
  - **Data Manager**
    Handles data access from twitter and system's data store.
- **Brief Descriptions of Use Cases**
  - **Subscription:** Responsible for register user with the application. Application send request to user and user verify application to access his tweets.
  - **Data Fetching:** This use case is responsible to fetch user tweets from Twitter to generate user profile.
  - **Preprocessing:** Data preprocessor handle with data parsing and slang removal from tweets.
  - **Parsing:** Parser use XML parser to convert XML data from twitter to standardized

individual objects

- **Tokenization:** Tokenization converts sentences into separate word to lookup on each word individually.
- **Text Normalization:** This use case process tweets and search for slangs in tweets to map slangs with their respective original word.
- **Keyword extraction:** To extract meaningful information from huge text Keyword extraction process extracts important keywords. It accepts tweets and return known keywords present in those tweets.
- **Entity classification:** This use case is responsible for recognizing individual entities from text and classifies those entities into different groups.
- **Sentiment analysis:** Aim of this use case is to determine the attitude of a speaker or a writer with respect to entities and keywords. This helps to know user behavior towards specific entities.
- **Semantic annotator:** It is about attaching synonyms and definitions, to keywords and entities. It provides additional information about an existing piece of data.
- **POS tagging:** It is the process of marking up a word in a text as corresponding to a particular part of speech, i.e. relationship with adjacent and related words in a phrase sentences nouns, adjectives, adverbs, etc.
- **Classification:** Tweets may be classified into different domain specific categories on the basis of entities and keywords extracted from tweets.
- **Filtration:** A data filtration is a process that allows domain specific data to pass for personalized modeler and ignore other information.

▪ **Sequence Diagram**

- The objective of Interaction model of a system is to depict the process scenario of how different objects interact with each other. Life span and sequencing of objects are the prime components of any interaction diagram. Sequence diagrams are described in following section.

- **Stream Collection**
  - Data Fetcher sends request to Twitter to collect tweets. After tweet collection from Twitter it passes data to preprocessor.

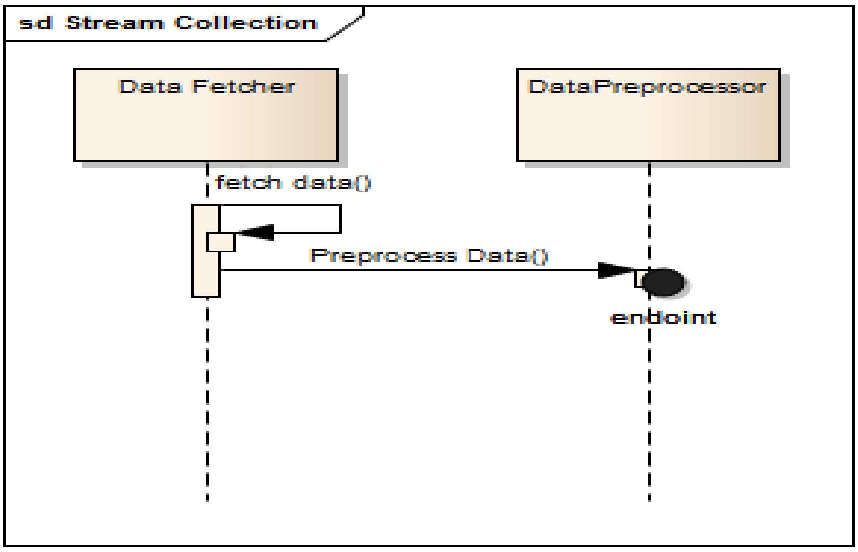


Figure 148 Stream Collection for wellness

- **Preprocessing**
  - Data preprocessor component processes to identify presence of any short hand notation and normalizes them to meaningful words. It passes data to XML parser. XML parser parses data using DOM parser and return parsed data. Preprocess then pass data to text normalizer. Text normalizes call tokenizer and passes data to apply tokenization on data. Tokenizer returns tokenized data to text normalize. Text normalizes replace slangs with original words.

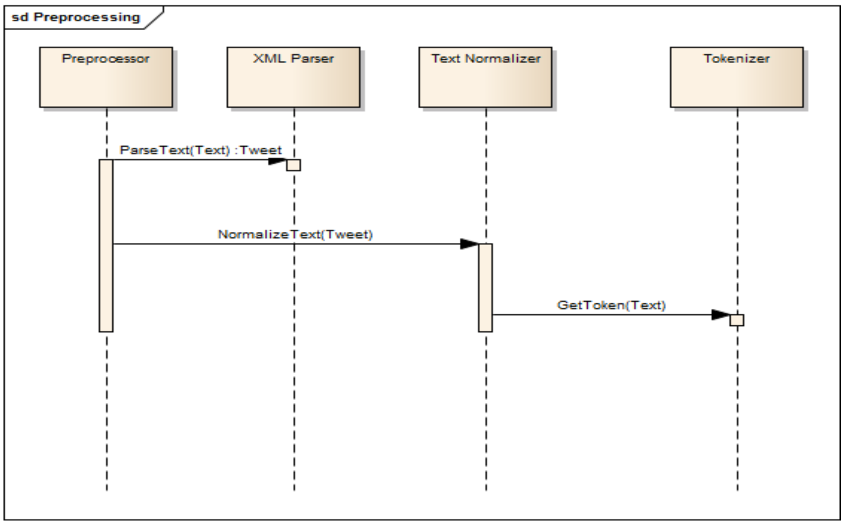


Figure 149 Data Preprocessing for wellness

- **Knowledge generation**
  - Knowledge generator pass tweets to keyword extractor which extracts known key phrases from tweets. Knowledge generator then pass tweet to entity classifier which identify different entities from tweets and return entities and their type. Sentiment analysis returns user sentiments towards those entities and keywords.

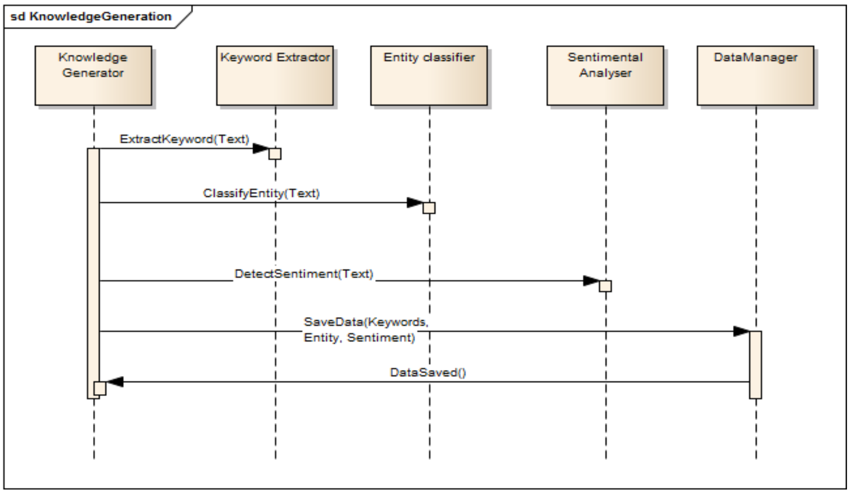


Figure 150 Knowledge Generation for wellness

- **Knowledge Enhancer**
  - Knowledge enhancer module passes tweet to part of speech tagger which split text into noun, verb, object, subject etc. Then knowledge enhancer pass keywords and entities to semantic annotator which use wordnet to increase knowledge by adding synonyms and definition.

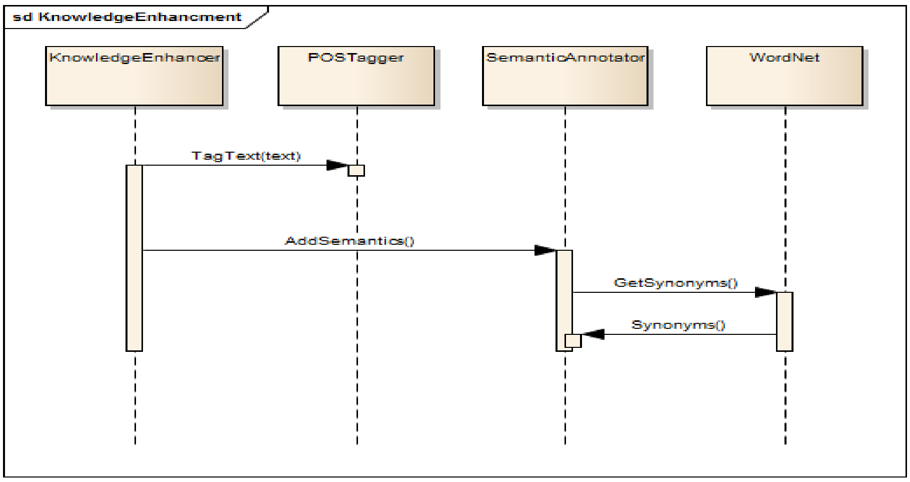


Figure 151 Knowledge enhancement for wellness

- **User modeling**
  - User modeler passes data to filter which filters data to make it domain specific like for health care it only passes information which is related to health care and ignore other information. Then information is passed to data manager which maintains user profile for future use.

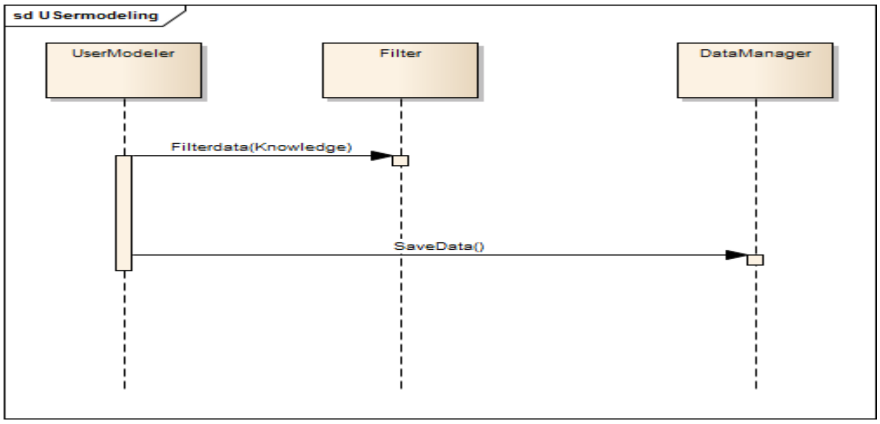


Figure 152 User Modeling for wellness

- **Class Diagram**

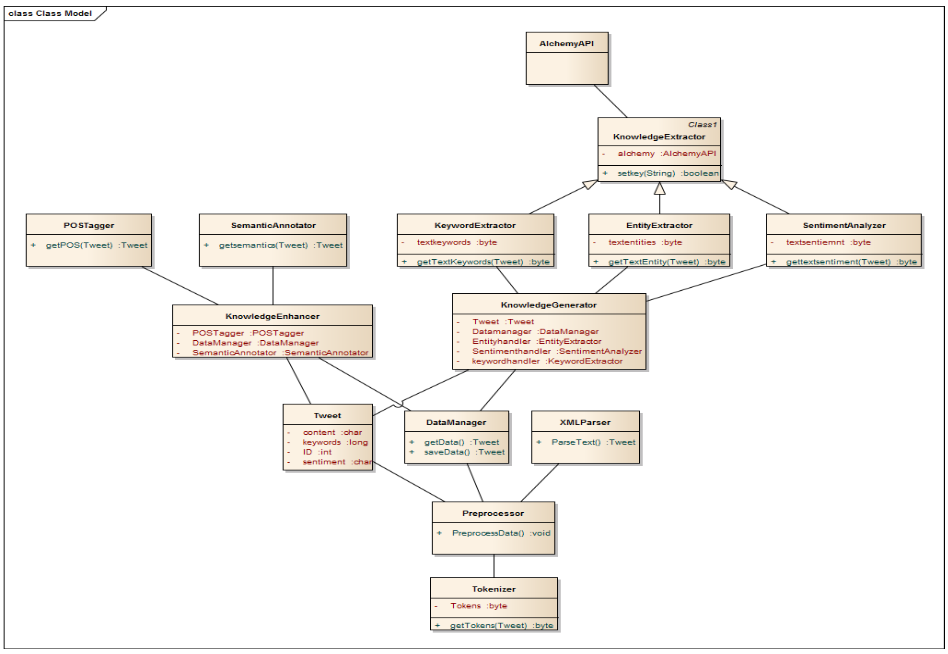


Figure 153 Class diagram of Tweet Analysis for wellness

- **Component Diagram**

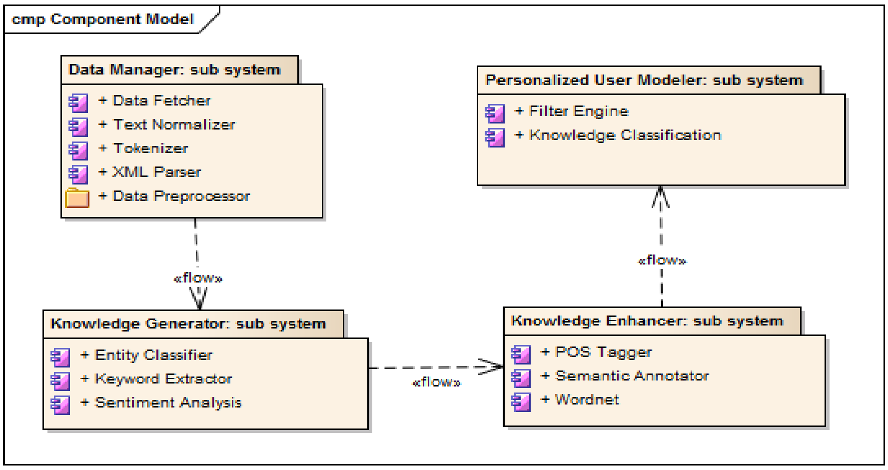


Figure 154 Component Model of Tweet Analysis for wellness

- **Data Manager**
  - This sub system is responsible for fetching data from social media and processing the fetched data. It has two parts.
- **Data Fetcher**
  - Data Fetcher sends request to social media for stream of user. The fetched data is in different format for each media.
- **Data Processor**
  - Fetched data requires some preprocessing before analyzing.
- **XML Parser**
  - XML parser uses Dom parser to convert XML data into required usable format and store data into different fields.
- **Text Normalizer**
  - Users use abbreviations to save time and space. Such kind of noise in data effects knowledge extracted from tweets. Therefore to remove such kind of noise, Text normalization removes slang and abbreviated word using slang lexicon
- **Tokenizer**
  - It split sentence into text based on the defined delimiter. Tokens are then used by text normalize to remove slang.
- **Knowledge Generator**
  - Knowledge Generator extracts user's interest by using Alchemy API. It obtains knowledge by exposing the semantic richness hidden in post.
- **Entity Classifier**
  - It extracts specific entities from tweets and type of those entities using alchemy API.
- **Keyword Extractor**

  - Keyword extractor use alchemy API to extract keywords from tweets.
- **Sentiment analysis**
  - It involves detection of user sentiment related to keywords and entities to now user attitude towards different entities and keywords.
- **Knowledge Enhancer**
  - Knowledge enhancer enhances knowledge of system using semantic annotator and part of speech tagging to make system context aware.
- **POS Tagging**
  - To know about relation of user with entity and to add more knowledge system extract part of speech and then add verb into information already extracted by knowledge generator. It also adds knowledge about relation of subject or object with user.
- **Semantic Annotator**
  - Use of word net makes it possible to add semantic of individual keywords and entities by addition of synonyms and definition of each entity and keyword.
- **Personalized user modeler**
  - Better services delivery requires maintenance of history of individual's interest and behaviors. Personalized User Modeler maintains user's data in Personalized Profile.
- **Classification**
  - To provide domain specific service based on user temporal patterns system classify data into different domains and store them into profile.
- **Filter Engine**
  - Filter engine activated and filter domain specific knowledge to provide better personalized services.

6.1.3.2. Trajectory Analysis

▪ **Use Case Diagram**
- Use case diagram of Trajectory Analysis is shown in Figure 155 Use Case Diagram.
- **Actors:**
  - **Social Network Trend Setter:** The trendsetters are the well-known personalities to whom people want to follow. Activities schedule of these personalities are acquired and used as a reference for the user. User can also add this schedule by himself.
  - **User:** The individual, whose activities and movement routines are tracked and monitored. User is responsible for adding the semantic tags of activity location.

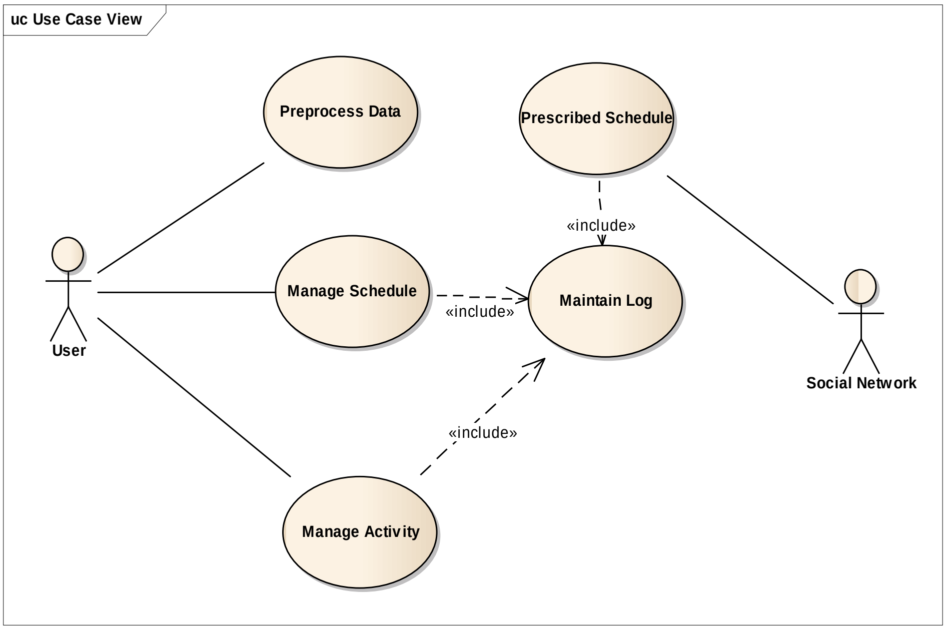


Figure 155 Use Case Diagram

- **Use Case Description**
  - **Manage Activity:** Manage Activity is main use case responsible for fetching the details of particular activity. As user changes its current activity location, system starts tracking the triggering of new frequent activity. Conversion of GPS coordinates, which are recorded by GPS enabled smart device of user into Geo tag is also a part of this use case.
  - **Manage Schedule:** In Manage Schedule user adds his preferred schedule if he/she want to add some recommended schedule. Schedule includes activities name, duration and number of occurrences in a defined period.
  - **Process Data:** Role of Preprocess data is to analyze the performed activities of user as compared to prescribed patterns by trendsetter or user recommended patterns. All the inconsistencies in performed activities are fetched and shown to the user in the form of a report.
  - **Add Prescribed Schedule:** Practitioner is required to add suggestions for a particular activity and also the complete schedule for the user. This prescribed schedule is taken as a reference and all the performed activities of user are evaluated against this schedule. This schedule is stored in repository using maintain log module.

▪ **Sequence Diagram**
- Sequence diagrams for Trajectory Analysis are described below that shows the interaction between different objects of the system for the achievement of objectives.
- **Analyze Performed Schedule:** Purpose of this interaction model is to track, monitor and evaluate user's schedule. On the change of activity location, movementTracker, fetches the required information of new activity using movementInformation(). GPS co-

ordinates information is passed from movementTracker and corresponding semantic tags of the location are acquired by the user. All the data are mapped with semantic tag and stored in the trajectoryRepositiory by the stroreActivity(). Activity Analyzer is the main function to analyze and compare the activity schedule. It fetches the performed activities of user of a particular period and compares it with the prescribed schedule of practitioner. Results of this module are sent to the network adopter. Below is the figure, Trajectory Analysis 2 Sequence Diagram of Analyze Performed Schedule describing this module.

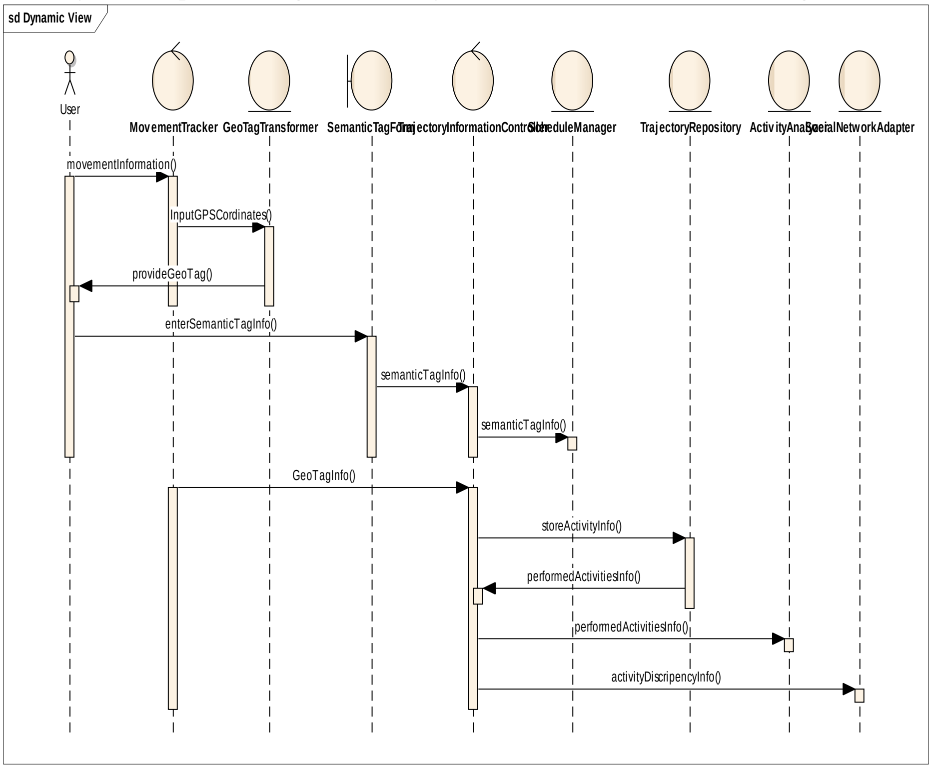


Figure 156 Sequence Diagram of Analyze Performed Schedule

- **Evaluate Performed Schedule:** Trendsetter module acquires the schedule from well-known personalities and also suggestions roe ah of activity is fetched. All of this information is stored in Trajectory Repository. Activity analyzer fetches this information and after comparing it with performed schedule of patient send it to social network adopter. Figure 156 Sequence Diagram of Evaluate Performed Schedule shows the sequence diagram of this module.

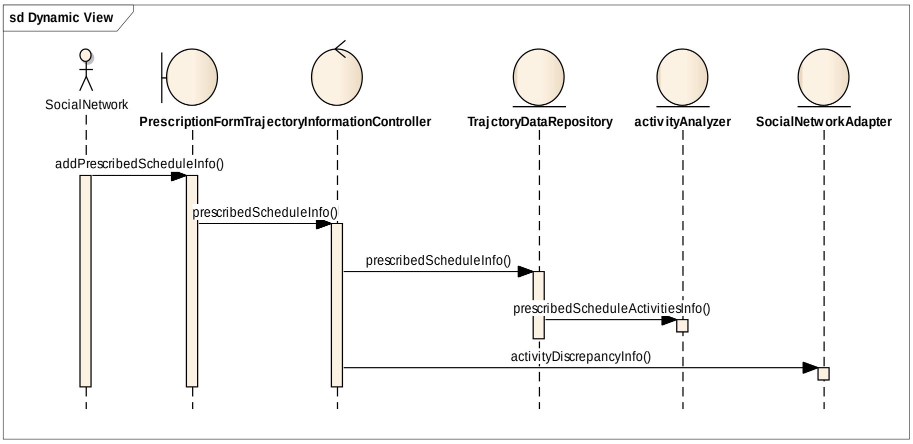


Figure 157 Sequence Diagram of Evaluate Performed Schedule

- **Class Diagram**
  - The purpose of class diagram is to show the relationships among the classes of the system. Additionally it covers the properties and operations that are visible at a higher level. Class diagram provides the high level overview of the system of how to be transformed into a detailed object model which ultimately helps to implement the system. Following section depicts the class diagram for Trajectory interaction for smartCDSS.
  - Task Assigned to the class Datapreprocessor is to fetch all the required information of trajectory and initial processing on it. ScheduleManager maps all the trajectory information with the semantic tag acquired from the user. Activity analyzer compares the user's schedule with the prescribed schedule and all the data stored in the repository by using class TrajectoryDataRepository. Figure 158 Class Diagram of Trajectory Analysis shows the class diagram of our system.

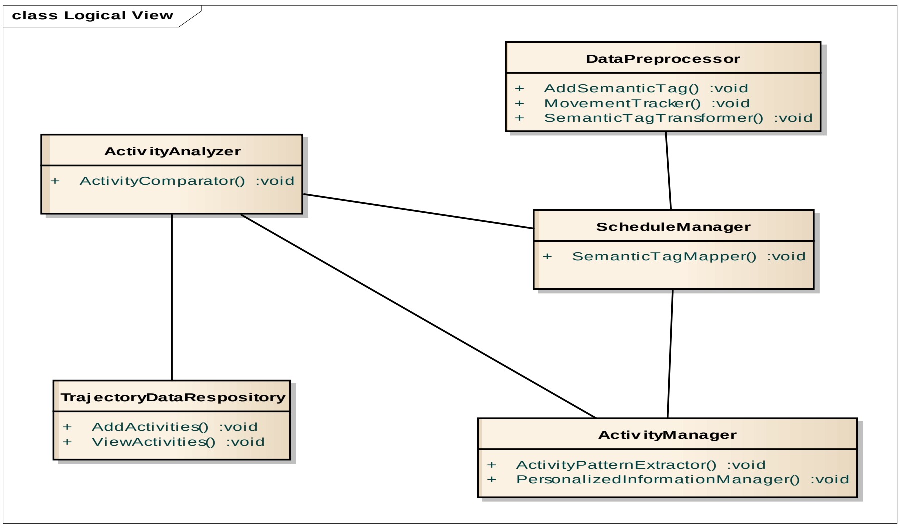


Figure 158 Class Diagram of Trajectory Analysis

- **Component Diagram**
  - A component is a software package, or a module, that encapsulates a set of related functions. All system processes are placed into separate components so that all of the data and functions inside each component are semantically related (just as with the contents of classes). Because of this principle, it is often said that components are modular and cohesive. By keeping above definition of software component, we divided our work into different components as shown in Figure below.
    - Data Preprocessor
    - Schedule Manager
    - Activity Manager
    - Activity Repository

  - **Activity Preprocessor**
    - Purpose of DataPreprocessor is to fetch all the data from the movement patterns of the patient. It further includes three parts. The imperative location finder is to detect that either particular location satisfy all the parameters of frequent patterns. Semantic tag acquires the context information of the particular activity location and task of Geo tag transformer is to convert GPS coordinates into Geo tag by using Google API.

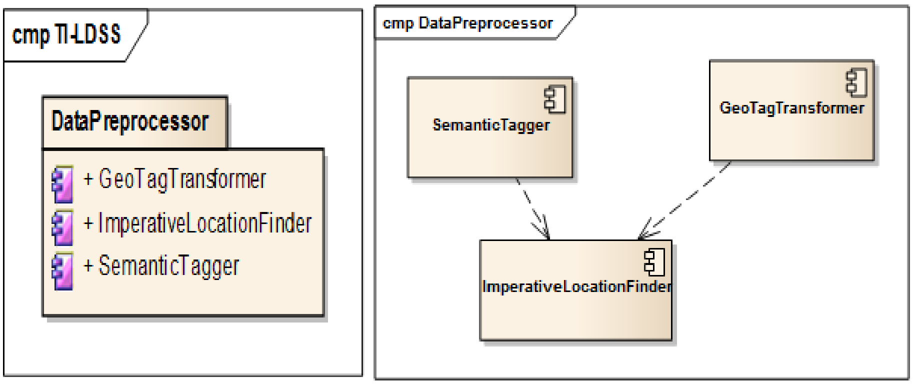


Figure 159 Component Diagram of Activity Preprocessor

  - **Schedule Manager**
    - Schedule Manager prepares and processes the data to find inconsistencies. It includes Semantic tag Mapper which links the semantic tag of location and other required information and pass it to Activity Manager and then to the trajectory repository. Purpose of second subpart, Activity analyzer is to compare the trendsetter schedule and user's performed schedule. Following is the component diagram of the schedule Manager.

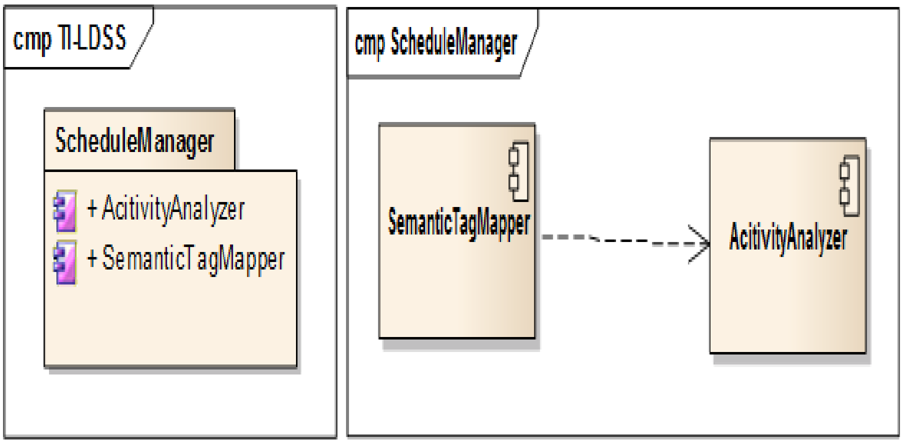


Figure 160 Component Diagram of Schedule Manager

- **Activity Manager**
  - Activity Manager is responsible for extracting the personalized information from the daily performed patterns of the user. For this first particular patterns are extracted from the activities and then passed into Personalized Information Manager.

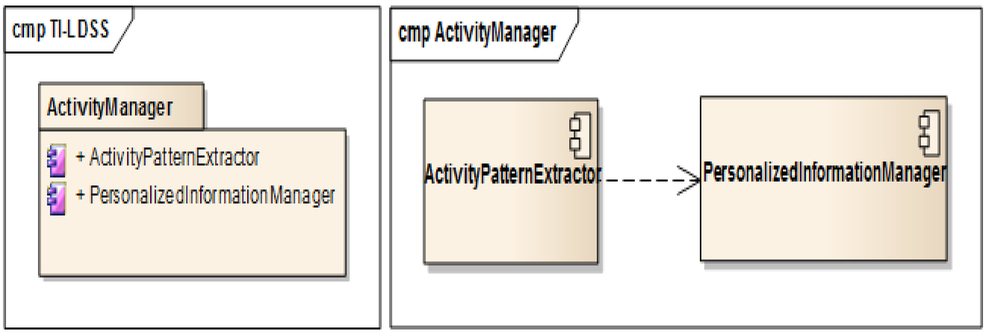


Figure 161 Component Diagram of Activity Manager

- **Trajectory Repository**
  - All the three kinds of data, 1) Performed patterns, the activities which are performed in daily life by the user. 2) Prescribed patterns, the schedule which is added by trendsetter and 3) Suggestions, the recommendations for each of the activity in pre-scribed schedule are stored in Trajectory Repository.

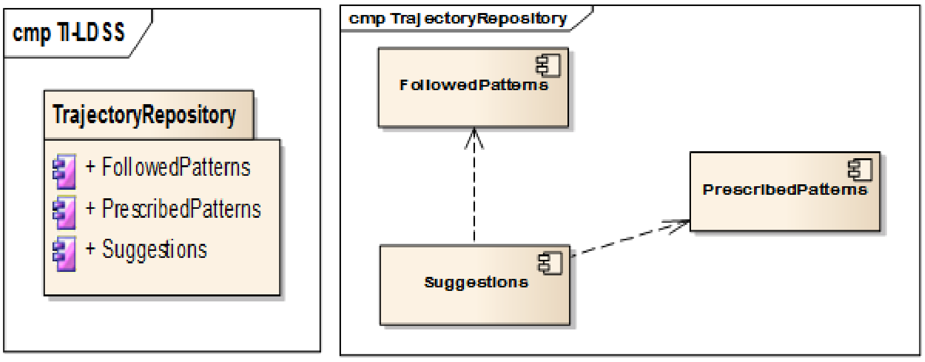


Figure 162 Component Diagram of Trajectory Repository

6.1.3.3. Interaction Analysis

- **Use Case Diagram**
  - The use case diagram of interaction analysis is shown in Figure 163, where as its description is given below
  - **Actors:**
    - **User:** The individual, whose emails are monitored and analyzed to find regular, frequent behavior.
    - **Keyphrases Extraction:** Application component used to extract keyphrases from the contents of interaction
    - **Data Manager:** Responsible to model graph on the basis of interaction between users with relevant keyphrases. Data Manager is also responsible to set the parameters for significance of the identified patterns
    - **Patterns identification:** Identify the candidate frequent and regular patterns and prune then according to the user defined parameters

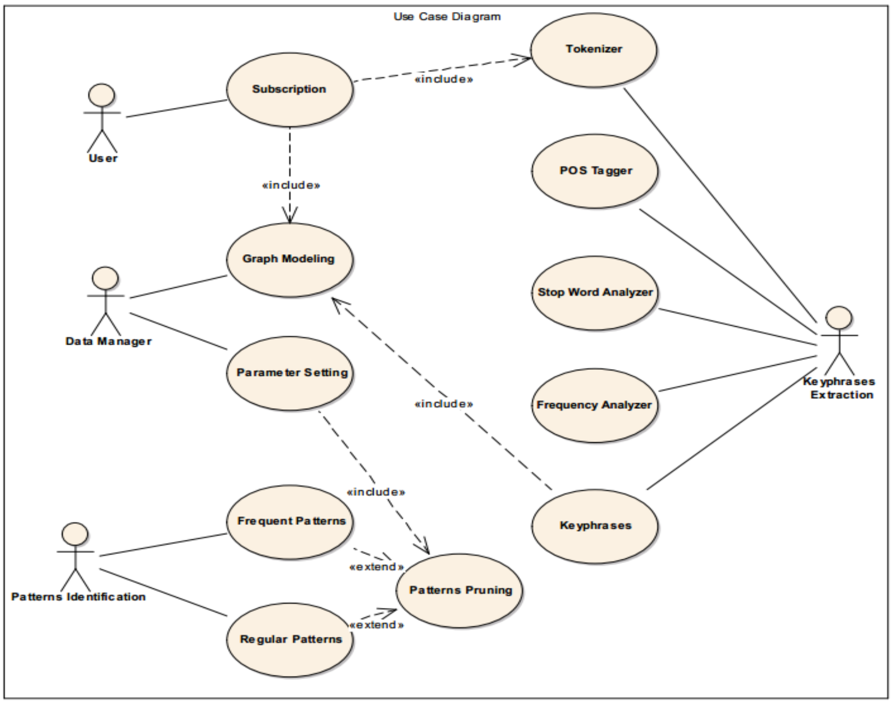


Figure 163 Overall system use case diagram

- **Brief Descriptions of Use Cases**
  - **Subscription**
    Responsible for register user with the application. Application send request to user and user verify application to access his emails.
  - **Tokenization**
    Tokenization converts sentences into separate word to lookup on each word individually.
  - **POS tagger**
    It is the process of marking up a word in a text as corresponding to a particular part of speech, i.e. relationship with adjacent and related words in a phrase or sentence as nouns, verbs, adjectives, adverbs, etc.
  - **Stop Word Analyzer**
    This use case process email contents and search for stops words after tagging into different parts. It removes the stop words from the contents of interaction.
  - **Frequency Analyzer**
    To identify the repetition of a particular word in the text, this use case calculates frequency of words which are semantically similar.
  - **Keyphrases**
    This use case collects the final keyphrases extracted at the end of NLP processing.
  - **Graph Modeling**
    Role of graph modeling is to model the graph from the user interaction on the basis of time and attach the relevant extracted keyphrases on each node which gives the semantic of interaction at a particular time.

- **Parameter Setting**
  Aim of this use case is to set the threshold parameters for the identification of regular and frequent patterns.
- **Frequent Patterns**
  This use case is responsible for mine the frequent patterns from the graph model of interactions.
- **Regular Patterns**
  This use case is responsible for mine the regular patterns from the graph model of interactions.
- **Patterns Pruning**
  It identifies the patterns of interest from the set of frequent and regular patterns after looking into the parameter settings of threshold.

▪ **Sequence Diagram**

- The objective of Interaction model of a system is to depict the process scenario of how different objects interact with each other. Life span and sequencing of objects are the prime components of any interaction diagram. Sequence diagrams are described in following section.

- **Keyphrases Extraction (Figure 164)**
  - Extraction of the semantic keyphrases is the essential requirement of the accurate data modeling with the user interaction. First email contents are tokenized by using POS tagger and stop words analyzer. The frequency of each word is counted in the email and then relevant keyphrases are returned.

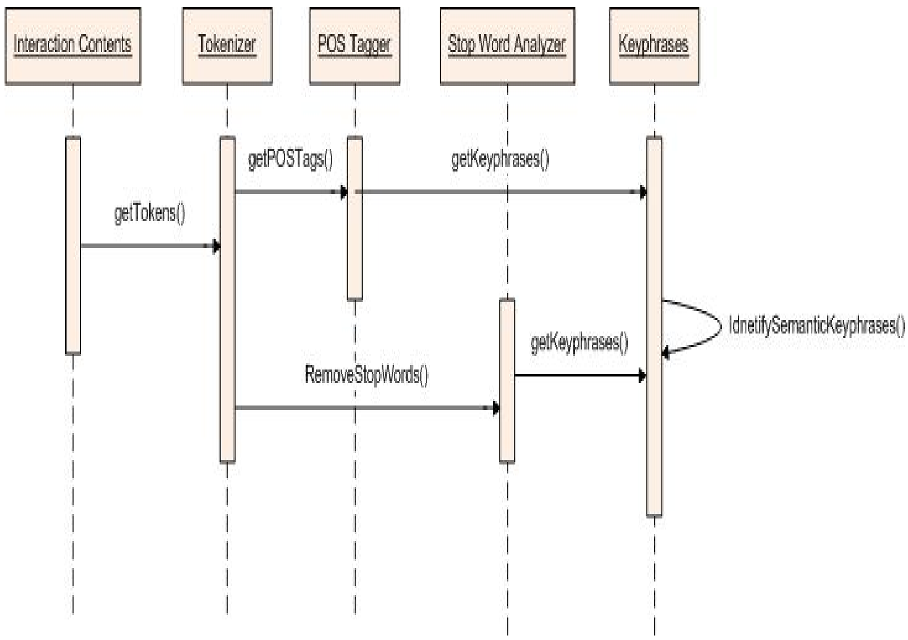


Figure 164 Sequence Diagram of Keyphrases Extraction

- **Data Manager (Figure 165)**
  - This module helps in data modeling and parameter settings before applying the mining algorithm. It extracts a population of interest from messy email interaction data by removing noise. The extracted information is modeled in graphs based on user defined interactions intervals and extracted keyphrases. In each graph nodes are the individuals with keypharses as node label and directed edge represents the interaction between them. Parameters set the thresholds of frequency and periodicity to identify the patterns of interest. For that, it is necessary to define a demanded minimum level (minimum confidence), so that all those sets of actions that have higher confidence level than the minimum confidence are considered as basic frequent periodic patterns.

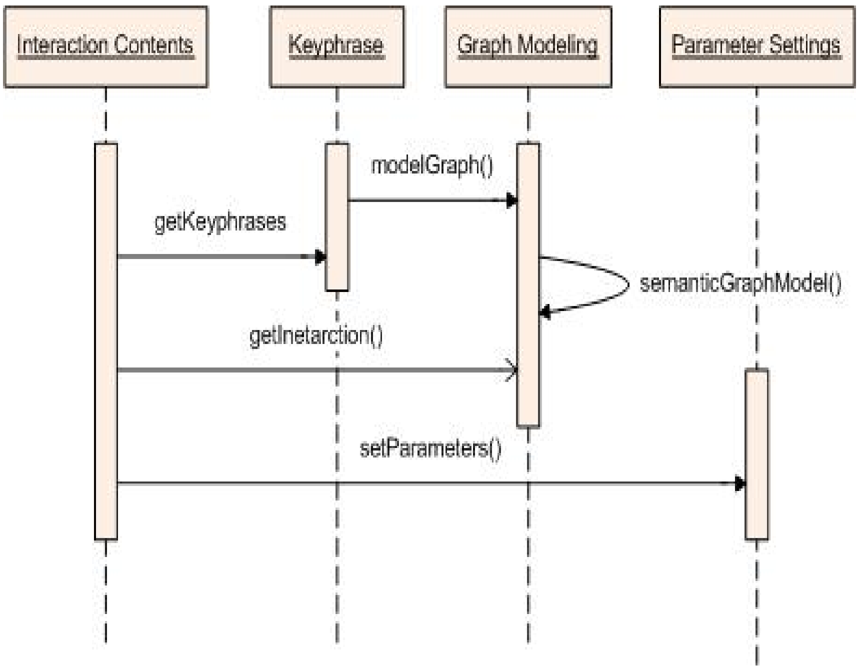


Figure 165 Sequence Diagram of Data Manager

- **Patterns Identification (Figure 166)**
  - This module identifies a set of frequent and periodic patterns from email interaction graphs. Frequent patterns emphasize the significance of patterns and periodic patterns consider the regularity between them. The objective is to identify the sets of actions that frequently and periodically occur together. Once basic frequent periodic patterns have been discovered, an aspect to consider is if there is any action which is not frequent taking into accounts all the periods, so that it is not discovered in the previous task, but it is frequent enough in those periods where the basic frequent periodic patterns occur. Patterns pruning reflects the common characteristics of a typical email interaction with some unusual association between patterns. For that, the starting point will be the candidate patterns are transformed into integrated set in order to make it useful comprehensively. Combining these two concepts allows us to define periodic patterns in a way that avoids any redundant information

.

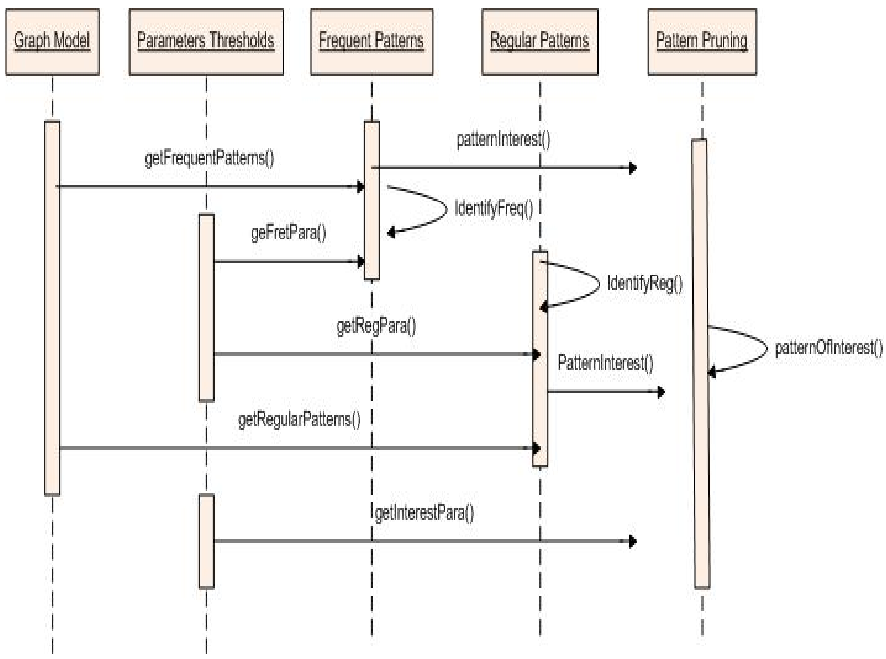


Figure 166 Sequence Diagram of Patterns Identification

- **Class Diagram**
  - The purpose of class diagram is to show the relationships among the classes of the system as shown in Figure 167. Additionally it covers the properties and operations that are visible at a higher level. Class diagram provides the high level overview of the system of how to be transformed into a detailed object model which ultimately helps to implement the system. Following section depicts the class diagram for interaction analysis for smart LDSS.
  - Task assigned to the class keyphrases is to extract keyphrases from the contents of interaction after applying the NLP techniques like POS tagging and stop words analyzer.

GraphModel maps the interaction and semantic keyphrases into the graphical format. FrequentPatterns and PeriodicPattens identify the candidate patterns while PatternsPruning identifies the patterns of interest after taking into account the parameter thresholds.

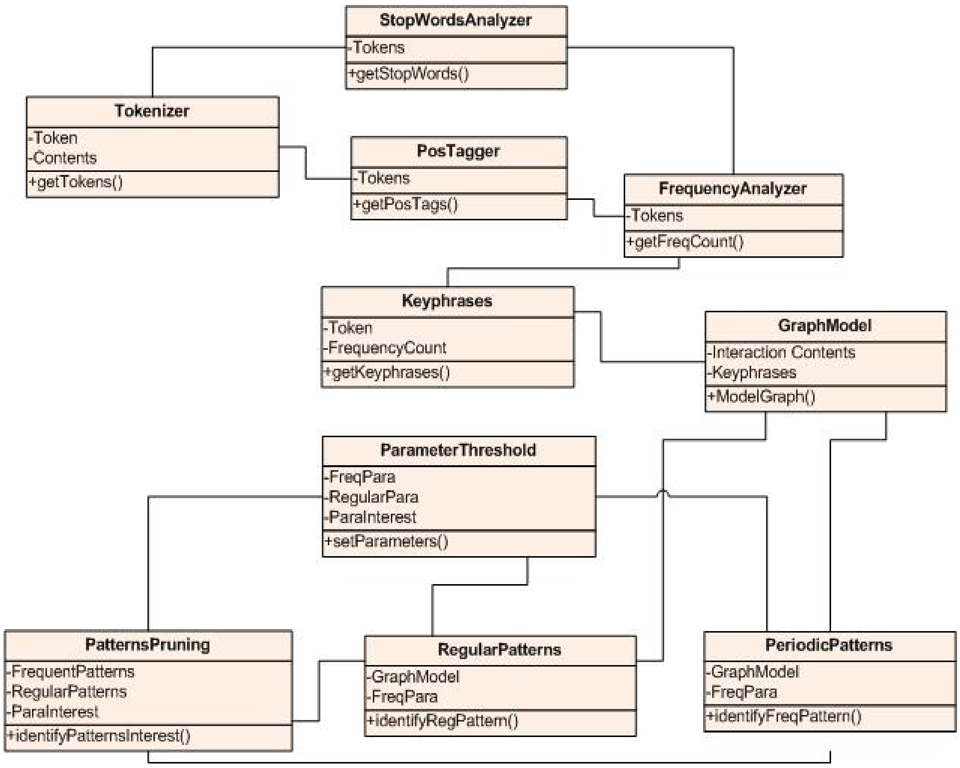


Figure 167 Class Diagram for Interaction Analysis

- **Component Diagram**

  - The component diagram of interaction analysis (Figure 168) shows different components and their relationships with each other. Also it shows the subcomponents interaction, purpose, and their relationships with each other. Mainly there are four main components and are explained below.

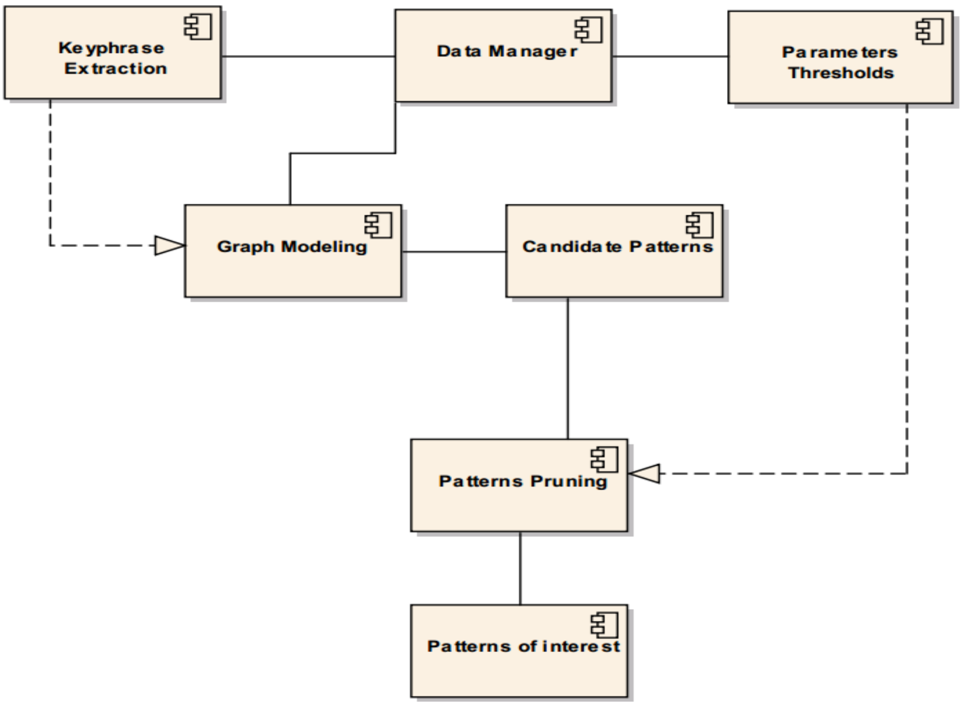


Figure 168 Component Diagram for Interaction Analysis

- **Keyphrases Extraction**
  - Extraction of the semantic keyphrases is the essential requirement of the accurate data modeling with the user interaction. The extracted keyphrases are used by graph modeling and data manager components for semantic annotation of interaction.

- **Data Manager**
  - This module helps in data modeling and parameter settings before applying the mining algorithm. It extracts a population of interest from messy email interaction data by removing noise. The extracted information is modeled in graphs based on user defined interactions intervals and extracted keyphrases. In each graph nodes are the individuals with keyphrases as node label and directed edge represents the interaction between them. Parameters set the thresholds of frequency and periodicity to identify the patterns of interest. For that, it is necessary to define a demanded minimum level (minimum confidence), so that all those sets of actions that have higher confidence level than the minimum confidence are considered as basic frequent periodic patterns.

- **Parameter Threshold**
  - This module manages the user defined threshold for graph modeling and for patterns in interest. Pattern pruning module used this information while processing the candidate patterns into patterns of interested.

- **Candidate Patterns**
  - This module identifies a set of frequent and periodic patterns from email interaction graphs. Frequent patterns emphasize the significance of patterns and periodic patterns consider the regularity between them. The objective is to identify the sets of actions that frequently and periodically occur together. Once basic frequent periodic patterns have been discovered, an aspect to consider is if there is any action which is not frequent taking into accounts all the periods, so that it is not discovered in the previous task, but it is frequent enough in those periods where the basic frequent periodic patterns occur.

- **Pattern Pruning**
  - This module applies one mining process to identify frequent and periodic patterns under the given parameter settings. Patterns pruning reflects the common characteristics of a typical email interaction with some unusual association between patterns. For that, the starting point will be the candidate patterns are transformed into integrated set in order to make it useful comprehensively. Briefly explained, it infers meaningful actions from the data collected by email data and then it splits the string of actions into periodic sequences based on some frequent support. Combining these two concepts allows us to define periodic patterns in a way that avoids any redundant information

- **Patterns of interest**
  - The patterns of interest after pattern pruning are converted are stored the extracted patterns in lifelog which is then passed to behavioral modeling module to analyze the lifelog with other information of user.

6.1.4. Behavior Modeling Module (BMM)

- BMM is one of the core components of LDSS for Wellness.
- It takes input form HAR, CAME, and Social Media and log them in life Log.
- Later it uses the Life Log information for service provisioning.
- The Life Log information is used for analyzing user behavior for shorter time interval and for longer time intervals which can be even years.
- On the other hand, BMM also provide the facility for behavior prediction in case the current behavior is matching with some existing (stored, learned) behaviors.
- The component based framework architecture diagram of BMM is given in main architecture of LDSS for Wellness.

- **Use Case Diagram**
  - The use case diagram of BMM is shown in Figure 169, whereas its description is given below.
  - **Actors:**
    - **User:** User is the main user of the system that is using BMM for behavior analysis and prediction which is based on the data user has performed and interacted.
    - **InputSources:** Input sources are the entities which are producing output which is considered as input for BMM.

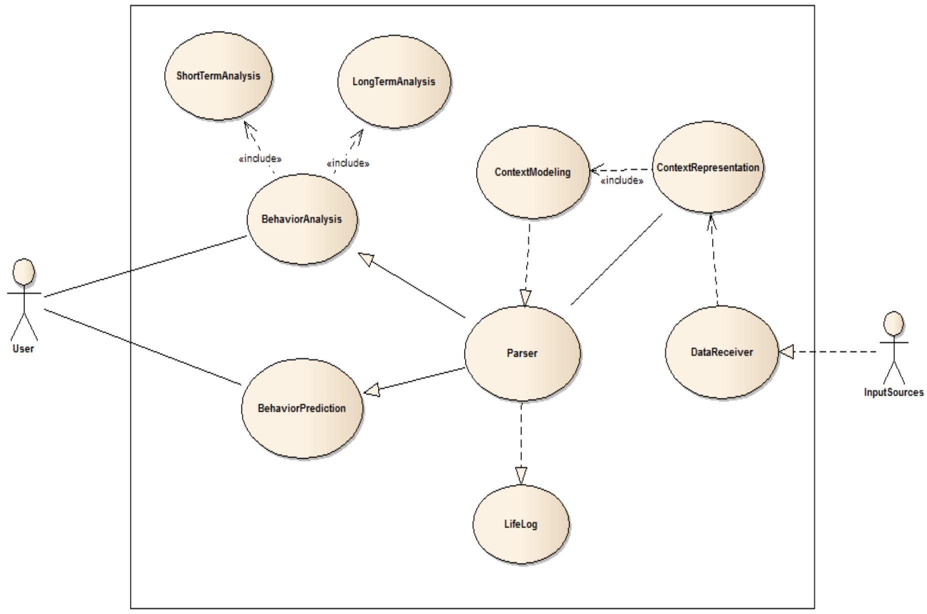


Figure 169 Use case diagram of BMM

- **Use case Description**
  - **DataReceiver:** This use case receives the incoming data from HAR, CAME, and Social Media in their respective formats.
  - **ContextRepresentation:** This use case is responsible for representing the incoming information from DataReceiver in ontological structure defined in Life Log while taking help from Parser.
  - **ContextModeling:** This use case is used to verify the represented information against the Life Log for its consistency and existence.
  - **Parser:** The responsibility of this use case is to parse the Life Log for the purpose of Life Logging as well as for assessing Life Log data to make the Behavior Analysis and Prediction.
  - **BehaviorAnalysis:** This use case uses different data mining techniques for analyzing user short term behavior analysis and long term behavior analysis.
  - **BehaviorPrediction:** This use case predicts user behavior based on user current interactions and already stored/learned behaviors.

▪ **Sequence Diagram**

- Sequence diagrams for BMM are described below that shows the interaction between different objects of the system for the achievement of defined objectives.

- **Context Modeling and Logging (Figure 170)**
  - When context is received in BMM then for its usage for behavior analysis and prediction and storage in Life Log needs to be in Life Log representational structure.
  - It calls the Controller using initiate(context) and start modeling the context received.

- It gets the representational structure from Life Log by calling getStructure() to Parser which retrieve the structure from Life Log.
- The represented context information is checked for consistency verification and if verified then is logged in Life Log using ContextModeling by calling logContext().

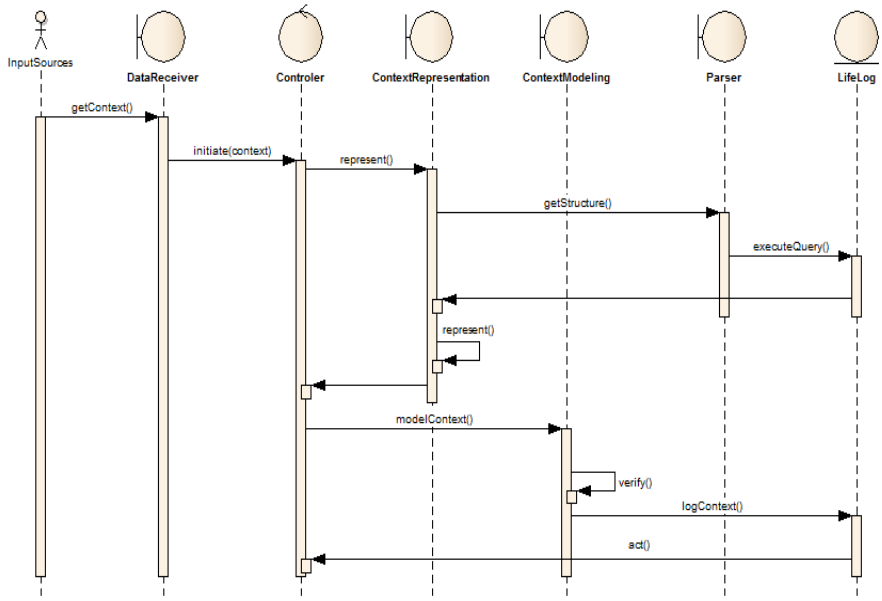


Figure 170 Sequence diagram of BMM Context Modeling and Logging

- **Behavior Analysis and Prediction (Figure 171)**
  - The Controller is activated for the purpose to initiate the behavior analysis and prediction
  - The logged data is extracted from LifeLog using executeQuery().
  - The BehaviorAnalysis and BehaviorPrediction are activated by Controller using initiateBehaviorAnalysis() and initiateBehaviorPrediction() respectively.
  - The results of BehaviorAnalysis and BehaviorPrediction are displayed on DisplayResult.

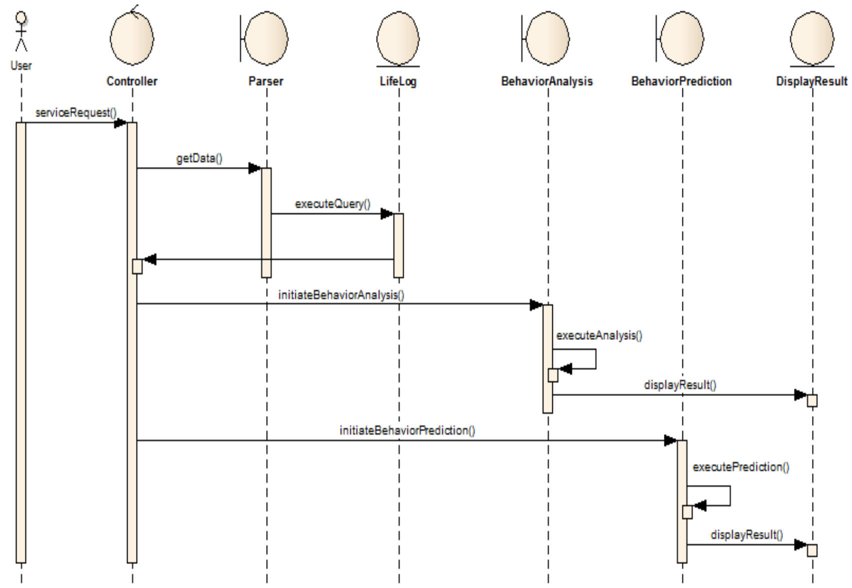


Figure 171 Sequence diagram of BMM Behavior Analysis and Prediction

- **Class Diagram**
  - Class diagram of BMM (Figure 172) shows the different classes and their relationships with each other.
  - Initially the DataReceiver and ContextRepresentation are loaded which work together for incoming context information representation.
  - Then the context information is verified in ContextModeling Class. The verified context is then logged using Parser Class in the Life Log.
  - DataExtractor extract the required data form the Life Log for user request service and provide the data to that particular service.
  - BehaviorAnalysis and/or BehaviorPrediction service is activated based on user request and requested services are returned to user.

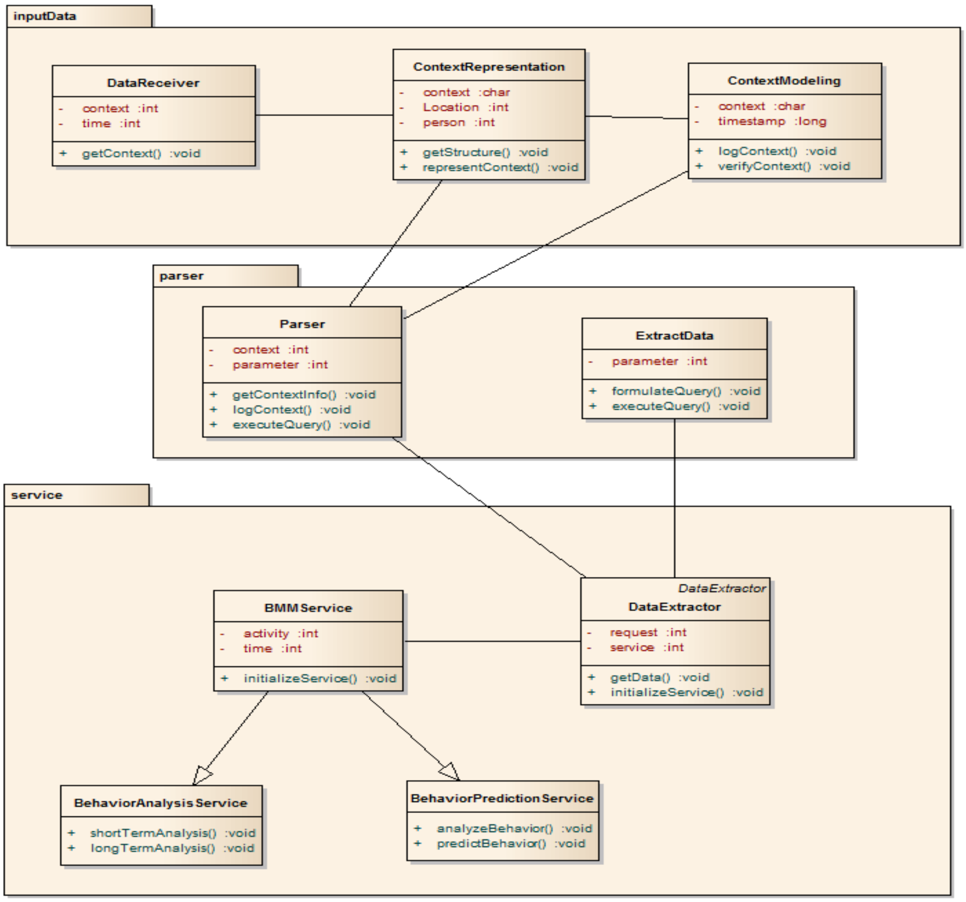


Figure 172 Class diagram of BMM

- **Component Diagram**
  - The component diagram (Figure 173) of BMM shows different components and their relationships with each other. Also it shows the subcomponents interaction, purpose, and their relationships with each other. Mainly there are five main components and are explained below.

  - **Context representation and Modeling Component**
  - It is responsible for representing the incoming context received from DataReceiver in the Life Log structure and then verify the context for its consistency.

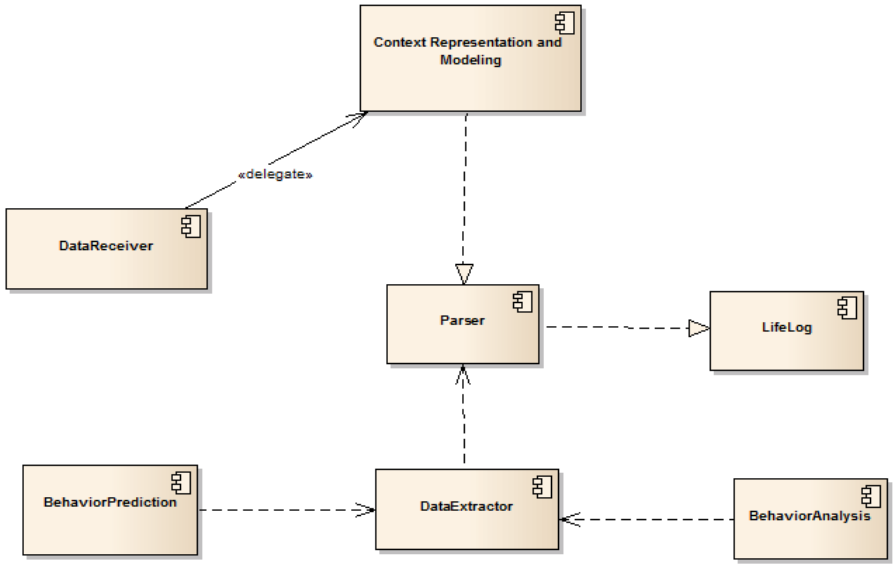


Figure 173 Component diagram of BMM

- **Parser Component**
- It is responsible for BMM components interaction with the Life Log for context logging and extraction. It is the main component that is always connected with the Life Log and fulfills the jobs coming for Life Log access.

- **Behavior Analysis Component**
- It is responsible for sending appropriate request for extracting data from Life Log based on user request. Then this data is used for User Behavior Analysis.

- **Behavior Prediction Component**
- This component is responsible to provide appropriate Behavior Prediction for user request based on the data extracted from Life Log.

6.1.5. Automatic Rules Generation and Inferencing using Rough Set Theory

- The purpose of this module is: 1- To automatically generate knowledge base form a huge volume of data with the help of Rough Set Theory and 2- To generate wellness recommendation for the users.
- We are proposing Rough set base inferencing, Case base inferencing and Bayesian based inferencing for generating recommendation (for the time being we are focusing on rough set theory).
- After analysis of the proposed system, use-case diagrams, sequence diagrams, class diagrams and component diagrams have been designed which are shown in the following subsequent sections.

- **Use Case Diagram**
  - Use-case Diagram for Rules Generation and Recommendation Using Rough Set Approach is shown below in Figure 174.

Figure 174 Use case diagram of Automatic Rules Generation and Inferencing using Rough Set Theory

- **Actors:**
  - **User**
    User is any person who wants to use the system.
    The actor is a trainer to train the system for the first time to build the knowledge base.
    Here the actor also represents a user who wants to make query for recommendation.
  - **System**
    The computer by itself is the System actor.

- **Use Case Description**
  - **Sets Policy:** Policy manager set the policy as a rough set technique for the generation of rules.
  - **Make Query:** Make Query use-case represents the way for querying Life Log for the data to train the system.
  - **Collects Training Data:** Query aggregator aggregates the data into a tabular form for training.
  - **Transform Data:** Data is transferred to the transformation module where it is discretized.

- **Handles Missing Data:** Discretized data is send to ‘handling missing data module’ of the reasoned to process the missing data.
- **Selects Features:** Most important features are selected.
- **Generates Rules:** Rules generation module is activated and the rules are generated.
- **Stores Rules to Knowledge Base:** Rules are stored in rough set knowledge-base.
- **Makes Query:** User makes query form the system for getting recommendations.
- **Generates Recommendations:** The system uses rough set reasoning mechanism and generates recommendations which are displayed to the users.

▪ **Sequence Diagram**

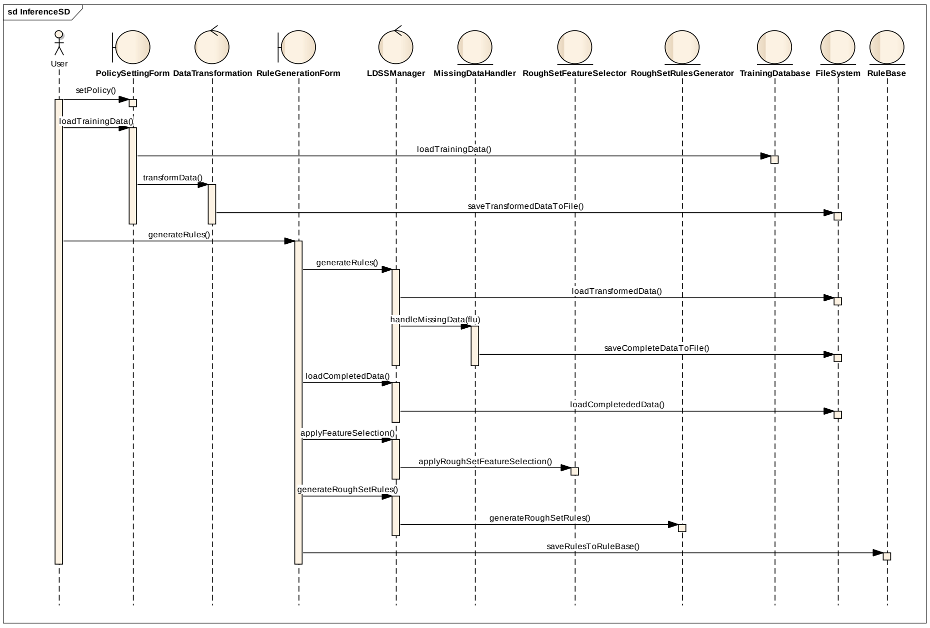


Figure 175 Sequence diagram for rules generation

- **Description of the messages for rules generation (Figure 175)**
  - **setsPolicy():** User calls the Policy Manager to set the policy for the selection of a particular repository from the Life Log and a desired reasoning method out of Rough set, Bayesian and Case-base.
  - **loadTrainingData():** User calls the Policy Manager to make a Query from the TrainingDatabase for the training data.
  - **transformData():** PolicyManager activates DataTransformation controller for the transformation of data.
  - **saveTransformedDataToFile():** DataTransformation controller transfers the transformed data to the FileSystem, a database, for storage.
  - **generateRules():** User activates the RulesGenerationForm which activates LDSS-MAnager controller in turn.
  - **loadTransformedData():** LDSSManager activates FileSystem entity for the retrieval of transformed data..

- **handleMissingData():** LDSSManager activates MIssinDataHAndler for handling missing data.
- **saveCompleteDataToFile():** MissingDataHandler save the completed data to the FileSystem database.
- **loadCompletedData():** RulesGenerationForm load completed data from the FileSystem database by first activating LDSSManager controller and then FileSystem entity.
- **applyFeatureSelection():** RuleGenerationForm activates LDSSManager for the selection of features.
- **applyRoughSetFeatureSelection()** : LDSSManager activates RoughSetFeatureSeletion entity for the selection of features using rough set approach.
- **generateRoughSetRules:** RulesGenerationForm activates LDSSMAnager for the generation of rules which in turn activates RoughSetRulesGenearator entity.
- **saveRulesToRuleBase:** RulesGenerationForm activates RuleBase database to store the generated rules.

- **Description of the messages for recommendation generation**
  - **inputQuery():** User makes query from the RecommendationForm boundary object.
  - **setPolicy()**: RecommendationForm activates DataManager controller for activating a particular reasoning method (rough set in this case).
  - **handleMissingData():** RecommendationForm activates LDSSManager to see whether something is missing in the query or not. LDSSMAnger in turn activates MissingDataHAndler entity to make the query complete.
  - **generateRecommendation():** RecommendationForm activated the InferenceEngine to start inferencing for the user query.
  - **searchForReleventRules():** InferenceEngine activates RuleBase database and search for the relevant decision which are returned to the user as recommendations.

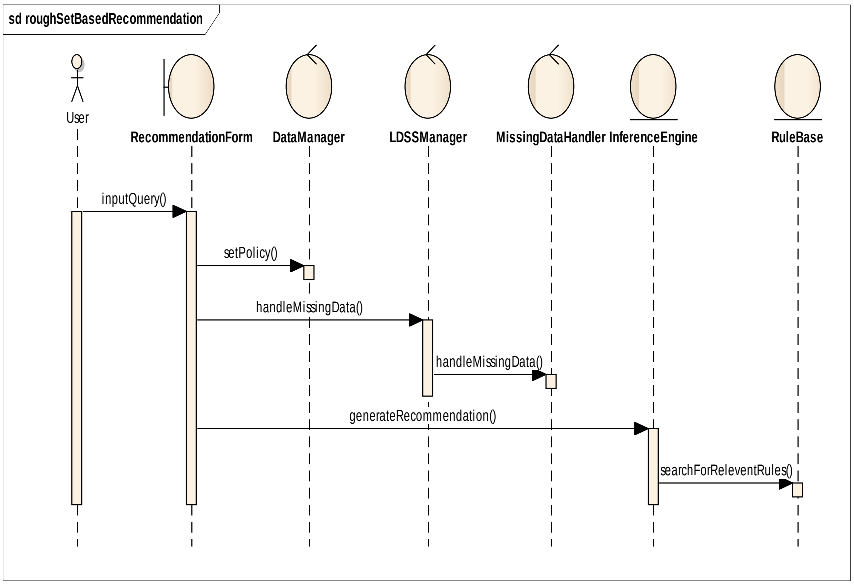


Figure 176 Sequence diagram for Recommendations generation

- **Class Diagram**
  - In total, we have seven classes for the proposed system to automatically generate rules using rough set theory from a huge data and then generate recommendations for user's queries.
  - The classes in the proposed system are: TrainingDatabase, MissingDataHandler, FileSystem, FeatureSelector , RuleGenerator , RuleBase, InferenceEngine.
  - The purpose (scope) and interaction of each class with other classes is given below.

  - **Description of the classes**
    - **TrainingDatabase:** This class represents the structured data in the form of database which can have missing filed, redundant field and uncertain records.
    - **MissingDataHandler:** This class is to take the data from the TrainingDatabase and perform different technique to find out the missing data which is necessary for further operations.
    - **FileSystem:** FileSystem is a text file system that is used for storing processed data.
    - **FeatureSelector:** The input for this class comes from FileSytem class. This class selects the best features out of all features using rough set technique. The selected features are send to the RuleGenerator class as an output.
    - **RuleGenerator:** This class takes selected features as input from and generates rules using rough set approach.
    - **RuleBase:** This class stores the rules generated by the RulesGenerator class.
    - **InferenceEngine:** This class takes rules from the RuleBase class as in input and perform inferencing to generate recommendations.

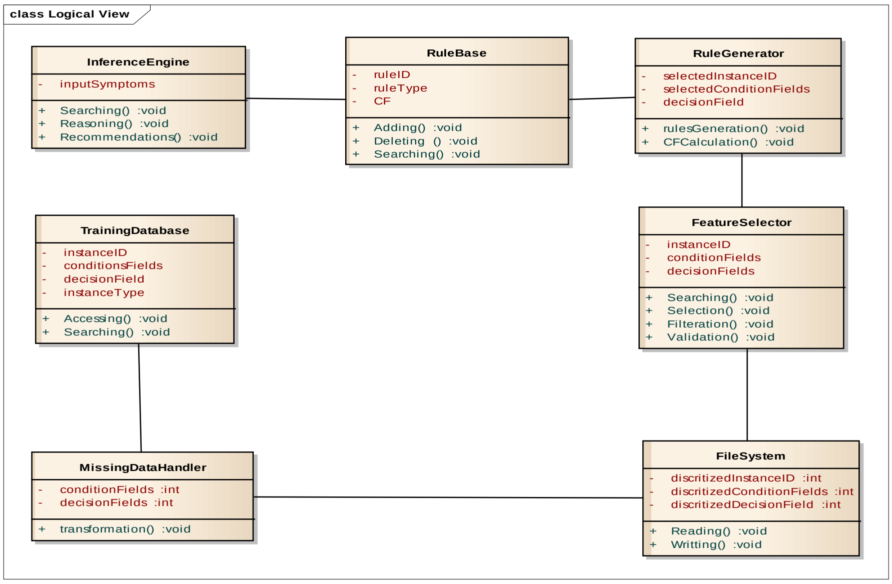


Figure 177 Class diagram

- **Component Diagram**
  - LDSS rules generation and recommendation framework consists of the following four main components.
    - DataManager Subsystem
    - InterfaceEngine Subsystem
    - DataAcquisitionEngine Subsystem
    - KnowledgeInferenceEngine Subsystem

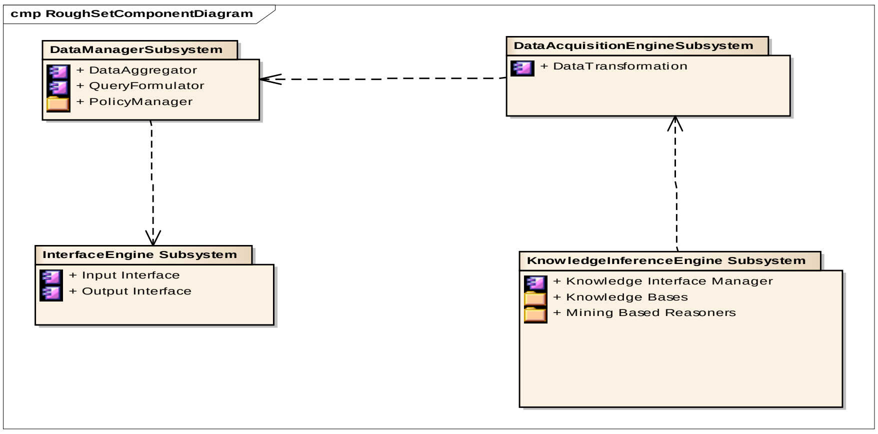


Figure 178 Component Diagram of the Whole System

- **DataManager Subsystem**
  - The purpose of this module is to retrieve data from the life log using a query, set some policy for transferring the data (or user query) to its respective modules and activating the corresponding reasoning technique for generating recommendation. Its sub-modules are shown below:
    a. Policy manager
    b. Query formulation
    c. Data aggregation

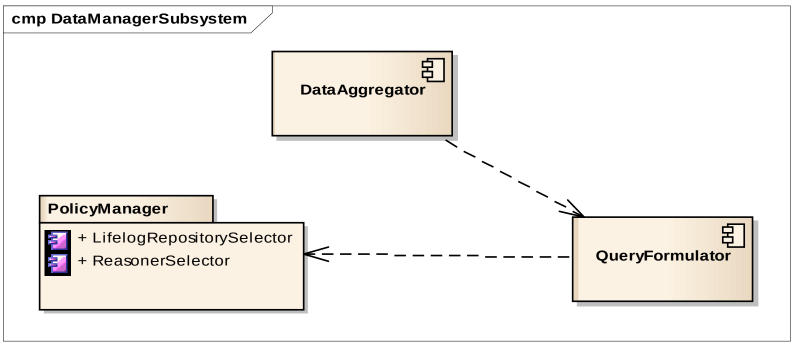


Figure 179 Component Diagram of Data Manager

- **Policy manager:** This module shown below performs two tasks: 1- Selection of a particular life log repository with the help of "*Life Log Repository Selection*" module and 2-Selection of a desired reasoning technique with the help of "*Reasoner Selection*" module.

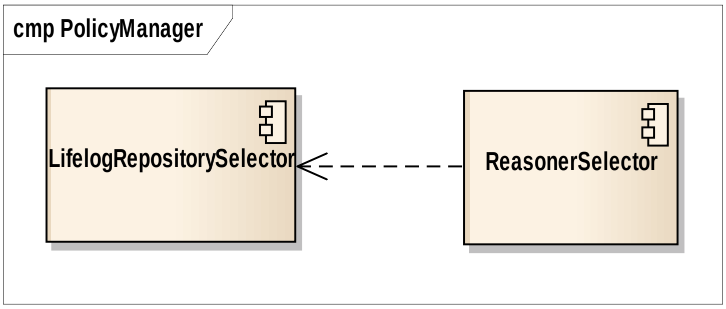


Figure 180 Component Diagram of Policy Manager

- **Query Formulation:** Query Formulation formulates a query over the Life Log to retrieve the required data for training purpose.

- **Data Aggregation:** If data is required from more than one source such as social media, activity recognition, context and some raw data then the Query Aggregator is activated.

- **InterfaceEngine Subsystem**
  - Interface engine provides a standard input interface and output interface to the users for inputting data to the system for training and producing output recommendation. It consists of two types of interfaces shown below.

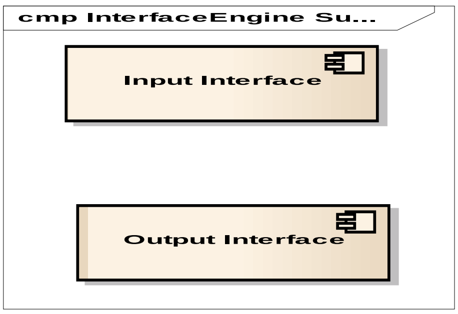


Figure 181 Component Diagram of Interface Engine

- **Input Interface:** This interface is used for two purposes: 1-User's input query and 2-Assigning values to features for discretization etc.

- **Output Interface:** Output interface is used to display the output of the system, generated in the form of recommendations, to the users.

- **DataAcquisitionEngine Subsystem**
  - In data acquisition module, transformation of data to the required format for a reasoner is converted. This component is shown below.

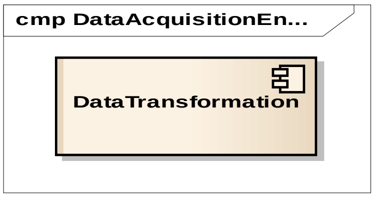


Figure 182 Component Diagram of Data Transformation Engine

- This engine performs differently based on the type of reasoning algorithm selected by the policy manager of the data manager module. For case-based reasoner, Bayesian-based reasoner and rough set base reasoner, this module has different roles.

- **KnowledgeInferenceEngine Subsystem**
  - This is the core module of the LDSS.
  - It consists of the following sub-components.
    a. Knowledge Interface Engine
    b. Mining Based Reasonor
    c. Knowledge Base

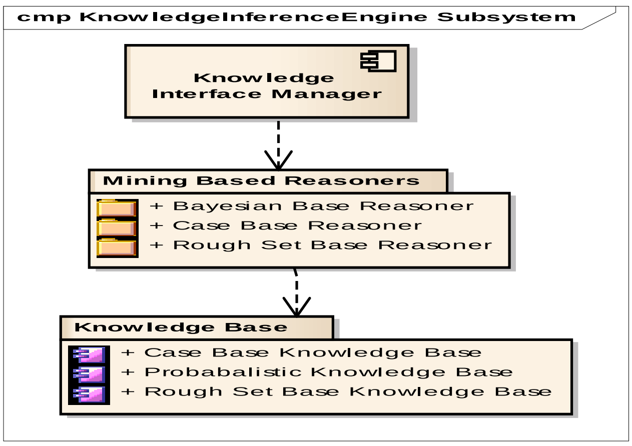


Figure 183 Component Diagram of Knowledge Inference Engine

- **Knowledge Interface Engine:** It is the interface through which the user query and recommendations are passed between the users and the reasoners.

- **Mining Based Reasonor (Rough Set):** Reasoner is the main module where inferencing take place for drawing conclusions (recommendations in this case). In our proposed architecture, we have three types of reasoners, case-base, Bayesian and rough set. Each one of them has its own format for input, processing and generating and displaying recommendations. Here, only rough set based approach is focused whose components are shown below.

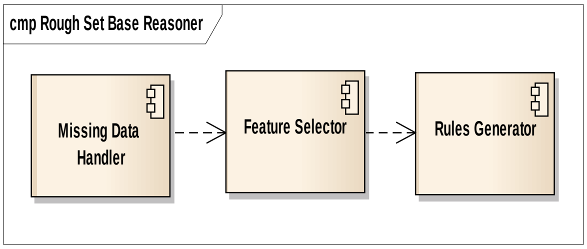


Figure 184 Component Diagram of Rough Set Reasoner

- **Knowledge base:** Knowledge base is the storage for the knowledge learnt during the first stage of processing the data. Based on the type of reasoning, our proposed architecture has three types of knowledge bases, case-base knowledge-base, probabilistic knowledge-base and rough set knowledge-base.

6.2. LDSS for Chronic Disease

- **Use Case Diagram**
  - Use case diagram is shown in Figure 185, where as its description is given below.
  - **Actors:**
    - **Patient:** Patient is responsible to provide the social media data, his/her data is also collected using sensors, his behavior data and clinical data.
    - **Physician:** Physician interacts with the system by entering the required data that is converted to HMIS compliant standard format.
    - **BMM Manager:** BMM Manager records the behavior related information of the patient.
    - **Social Media Manager:** Social Media Manager is responsible for managing the social media information of the patient.
    - **Sensor Application Patient Agent:** Sensor Application Patient Agent is responsible for collecting the sensor based low level and high level patient activities.

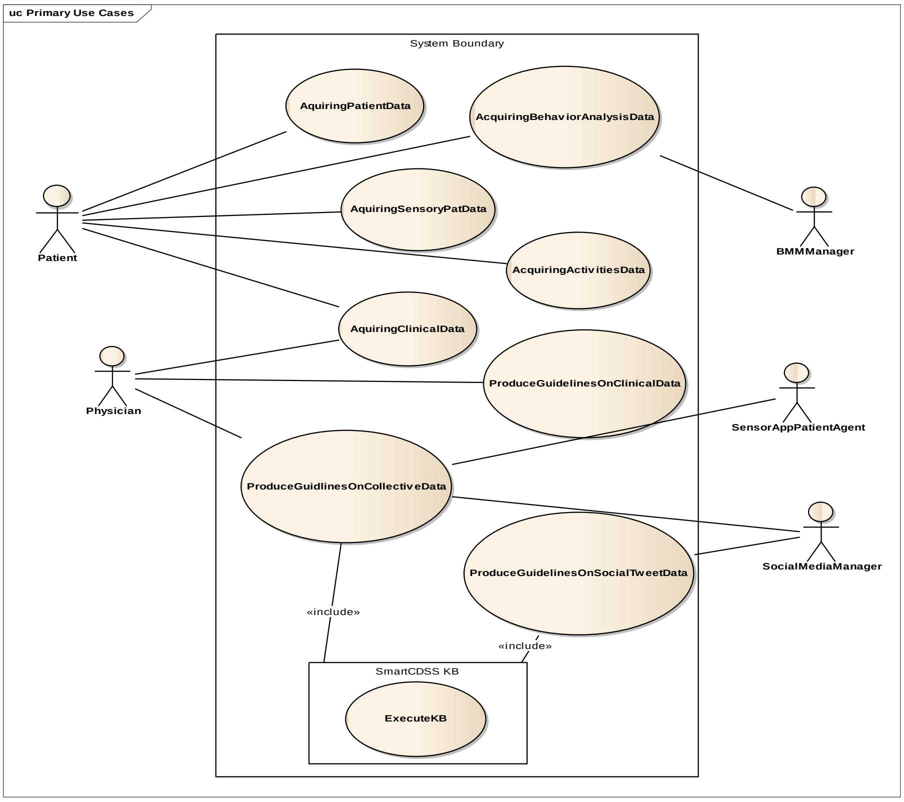


Figure 185 Use case diagram of LDSS for Chronic Disease

- **Use case Description**
  - **Acquiring Behavioral Analysis Data:** Acquiring Behavioral Analysis Data use case records the patient behavior analysis data. This includes patient daily routine activities.
  - **Acquiring Sensor Data:** Acquiring Sensor Data use case is used for collecting sensory data about the patient. This data is collected using different sensors.
  - **Acquiring Activities Data:** Acquiring Activities Data use case is responsible for acquiring the different activities data from sensors and camera and finally fusing the data.
  - **Acquiring Clinical Data:** Acquiring Clinical Data use case is used for obtaining the **clinical** data of the patient from the HMIS. This includes patient observations.
  - **Acquiring Patient Data:** Acquiring Patient Data use case collects patient data from clinical information and also from social media.
  - **Produce Guidelines on Clinical Data:** Produce Guidelines on Clinical Data use case is used for proving guidelines based on the clinical data that consists of clinical observations.
  - **Produce Guidelines on Collective Data:** Produce Guidelines on Collective Data use case provides the collective guidelines based on all the inputs to the LDSS system.
  - **Produce Guidelines on Social Tweet Data:** Produce Guidelines on Social Tweet

Data use case provides guidelines based on social media data.

- **Execute KB:** Execute KB use case stores all the rules in the knowledge base that needs to be fired when recommendations required to be generated.

- **Sequence Diagram**
  - Sequence diagrams are described as follows that shows the interaction between different objects. LDSS-Chronic Disease overall sequence diagram is given in Figure 186.
  - **LDSS for Chronic Disease Sequence Diagram**
    - Physician provides input in the form of observations using getGuidelines(patObsHL7CDA) method to LDSS. Adapter Interoperability Engine object coverts standard format of CDA to VMR using createLDSSInputvMR(patObsHL7CDA) method.
    - Social Media Manager provides input about patient's social media interaction using getGuidelines(tweets) method. Social Media Adapter object creates its VMR using createLDSSInputvMR(tweets) method.
    - Sensor Application Patient Agent provides input about patient's activities using getGuidelines(patActivitiesXML) method. These activities are low level sensory activities. AER Adapter object creates its VMR using createLDSSInputvMR(patActivitiesXML) method.
    - Sensor Application Patient Agent provides input about patient's activities using getGuidelines(patHighLvlActivitiesXML) method. These activities are high level activities. CAME Adapter object creates its VMR using createLDSSInputvMR(patHighLvlActivitiesXML) method.
    - BMM Manager takes data from life log repository and provides the behavior analysis data using getGuidelines(behavior) method. BMM Adapter object creates its VMR using createLDSSInputvMR(behavior) method.
    - The overall data from all the modules is collected and provided to the Fusion Adapter object. This object concatenates all the VMRs using mergeInput(inputHL7VMR) method.
    - The concatenated VMR is provided to the LDSS Service object for obtaining guidelines using getGuidelines(imoutHL7VMR) method.

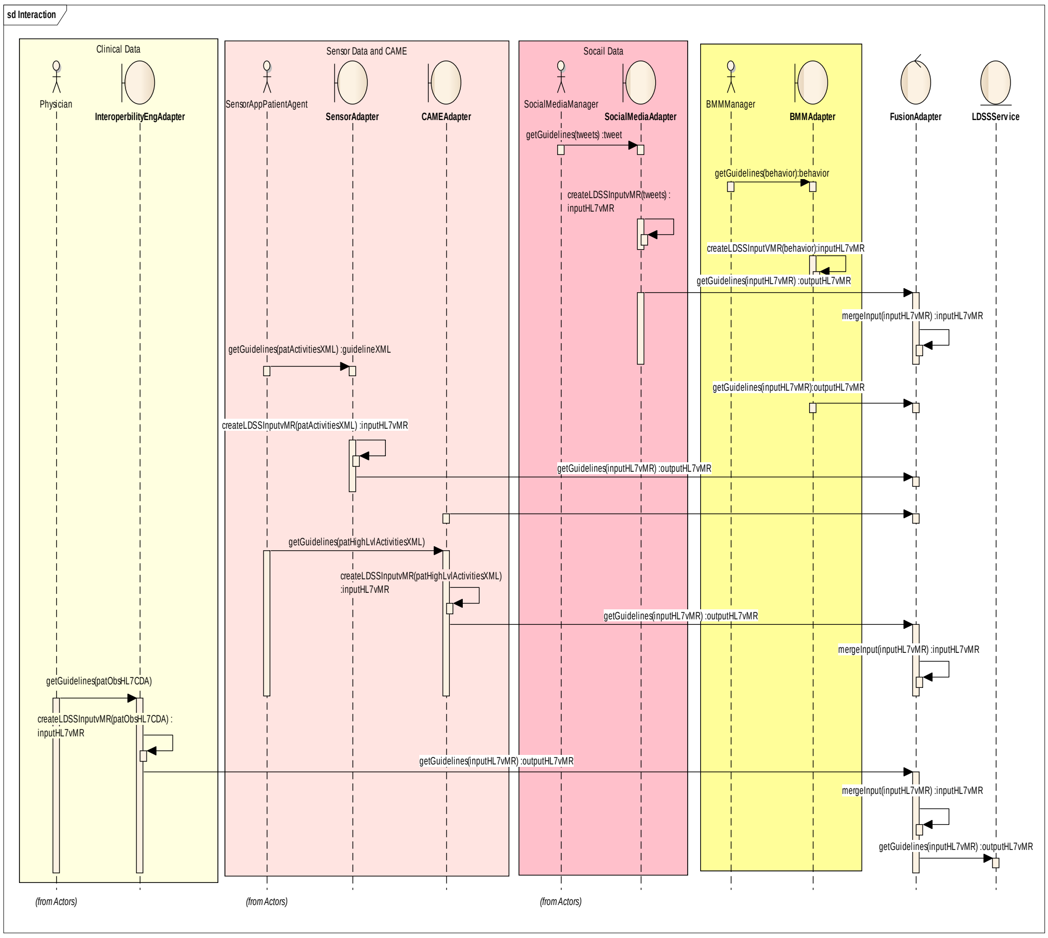


Figure 186 Overall sequence diagram of LDSS for Chronic Disease

- **Class Diagram**
  - LDSS-Chronic Disease class diagram shows how the different classes (see Figure 187) relates with each other to provide appropriate guidelines to the physicians.
  - Medical Logic Modules (MLM) are the standard format for generation and storage of rules in Arden Syntax.
  - These MLMs store the logic behind the rule to be fired. These are stored in the Knowledge Base. Therefore the classes are divided on the bases of MLM distribution in the knowledge base.
  - Factory Design pattern is shown in the class diagram that shows MLMExecutionFactory class.
  - This class is based on clinical and non-clinical information, to decide which reasoned needs to be invoked for recommendation generation. Clinical information is represented by ClinicalMLMExecutionFactory class while the non-clinical information is represented by NonClinicalKBExecutionFactory class.
  - ClinicalMLMExecutionFactory class is related with MLMBaseInterface abstract class. Different classes are inherited from MLMBaseInterface class based on the logic stored in

each MLM class.

- These includes DiabetesFindingMLM class that is used to find whether the patient has diabetes or not; the DiabetesCholestrolMLM class that is used to find about the problem in the cholesterol level of the patient; and finally DiabetesGlycaemiaMLM that is used to find whether the patient has glycaemia or not.
- Also there is MLMMetaData class related with MLMBaseInterface class that is used to store annotations about each MLM stored in the knowledge base for easy retrieval.

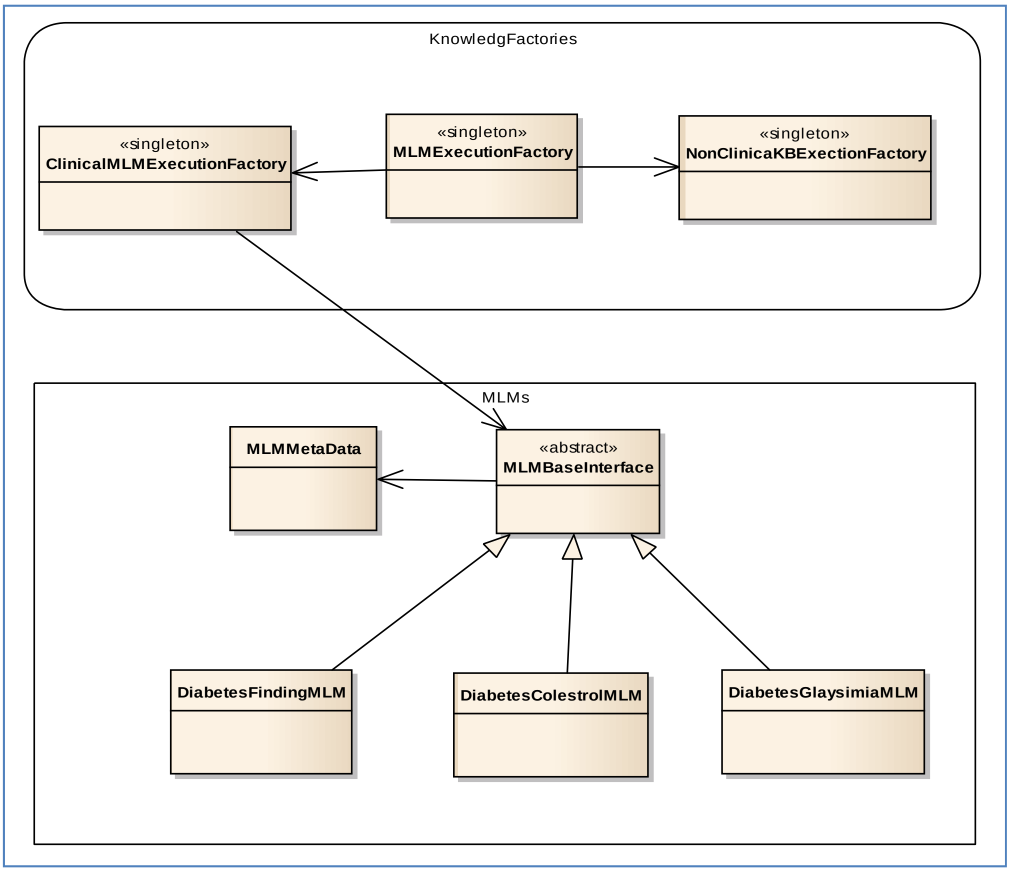


Figure 187 Class diagram of LDSS for Chronic Disease

- **Component Diagram**
  - The component diagram of LDSS-Chronic Disease (see Figure 188) shows the different components and their subcomponents interactions with each other. These are explained as follows

  - **HAR**
  - This component is used to provide user daily life activities information, user diet information, and his actions. It consists of subcomponent to collect and recognize activities that are as follow:

- Accelerometer based activity recognition.
- Smartphone based activity recognition
- Voice based activity recognition
- Video based activity recognition
- Diet information recognition

- **EMR**
- This component is used to provide clinical data in HMIS compliant standard format. It consists of subcomponent to generate the standard format.
  - CDA Generator is the subcomponent used for generation of CDA format instance of the patient clinical information.

- **CAME**
- This component is used to convert low level sensory information into high level activities. These activities are then provided to LDSS using CAME Adapter subcomponent. This component consists of two subcomponents that are:
  - Context Interpreter
  - Mapper & Transformer

- **Social Media Interaction Engine**
- This component is used to process the data gathered from the social media. This information is provided to the LDSS using Social Media Adapter subcomponent. The different subcomponents of this module are:
  - Tweet Analysis
  - Trajectory Analysis
  - Interaction Analysis
  - Non Negative Matrix Factorization

- **Adaptability Engine**
- This component is responsible for obtaining as input data about chronic disease from different heterogeneous modules. The modules outside LDSS can only interact with this system through Adaptability Engine. This engine includes the following subcomponents that are described individually as well in next sections:
  - Adapter AER
  - Adapter BMM
  - Adapter CAME
  - Adapter Interoperability Engine
  - Adapter Social Media
  - Fusion Adapter

- **Authoring Tool**
- This component provides the facility to physicians to enter their knowledge into the knowledge base that will becomes the rules for recommendations to be provided. These consist of subcomponents like:
  - Guideline Publisher
  - Knowledge Validator

- **Interface Engine**
- This component is used to behave as bridge between adaptability engine and knowledge inference engine. It takes input from the adaptability engine and provides to knowledge inference engine from processing. Finally it takes the recommendations from knowledge interface engine and provides it to adaptability engine. It also provides subscription facility to authorized users. All these functions are performed by subcomponents of these components that are:
    - Standard Input Interface
    - Standard Output Interface
    - Subscription Service

- **Knowledge Inference Engine**
- This component processes the input data to generate the output in the form of recommendations. It performs reasoning on the data and provides appropriate guidelines. It consists of knowledge broker subcomponent that is used for deciding to invoke reasoned that is based on standard ontologies or non-standard based reasoning engine. It consists of the following subcomponents:
    - Knowledge Broker
    - Mining Based Reasoning
    - Standard Ontologies
- All the main components of LDSS-Chronic Disease that are responsible for taking data in different formats are described in detail in the next sub-sections.

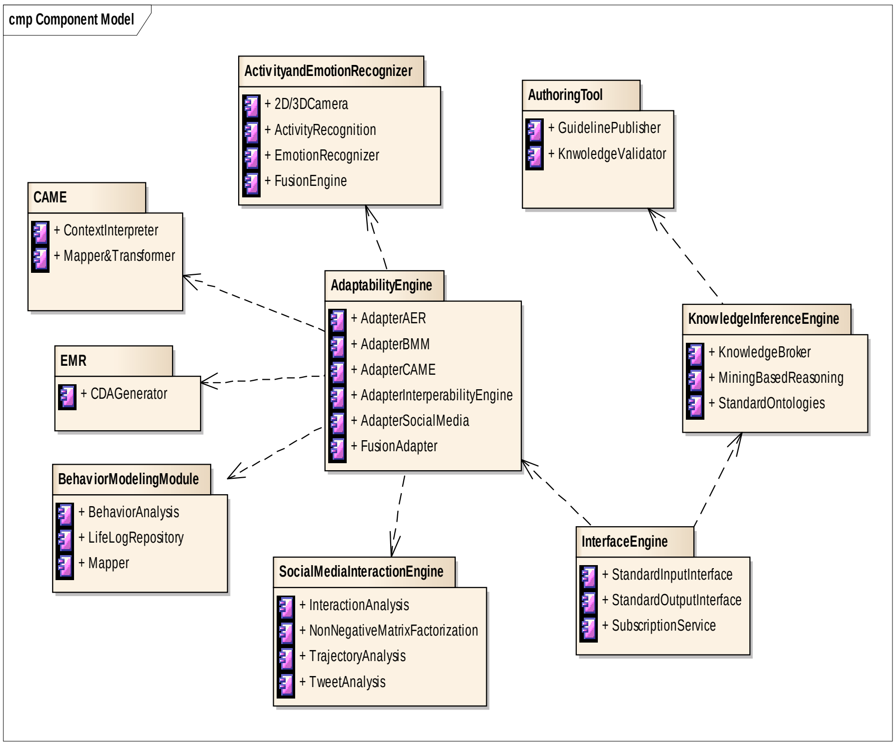


Figure 188 Component diagram of LDSS for Chronic Disease

6.2.1. Human Activity Recognition (HAR)

6.2.1.1. Audio Based Emotion Recognition

- **Use Case Diagram**

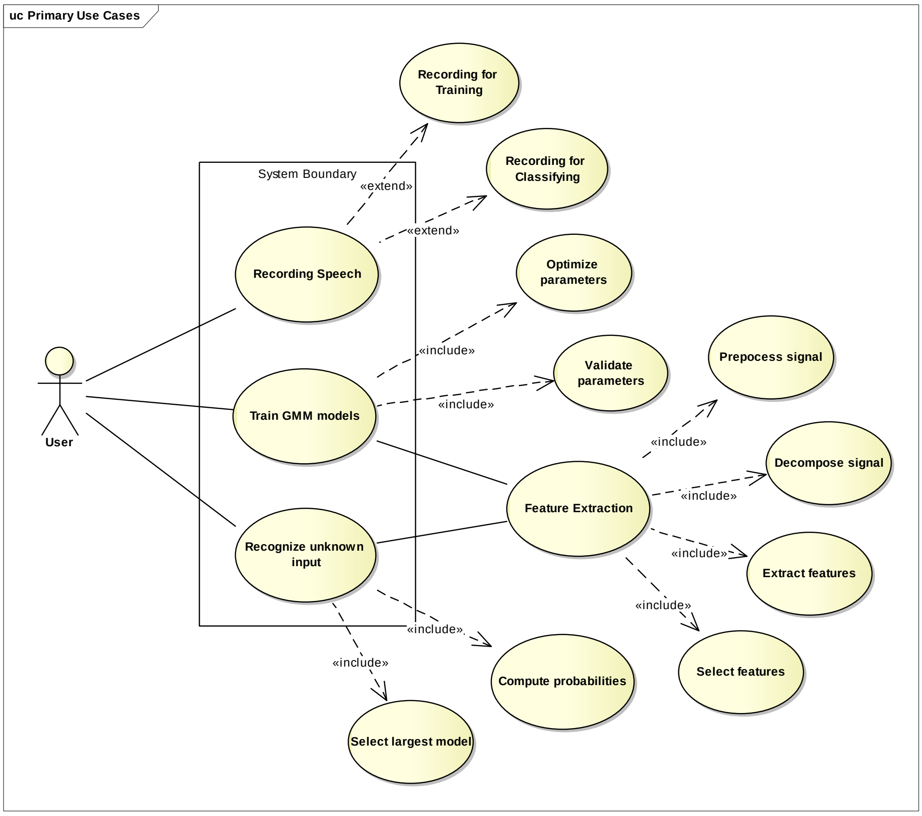


Figure 189 Use Case diagram of Audio-based Emotion Recognition

- Description: It interacts with the audio sensor and activates different components of the audio-based emotion recognition.
- Actors: Human beings interacting with the objects.
- Pre-conditions: Annotated data for training purpose
- Basic flow: The basic flow should be the events of the interaction with objects and everything is perfect; there are no errors, no exceptions. The exceptions will be handled to maintain the logs and alerts to the concern person.

- **Sequence Diagram**
  - Following training and testing phase sequence diagrams show the sequence of message between objects in an interaction.

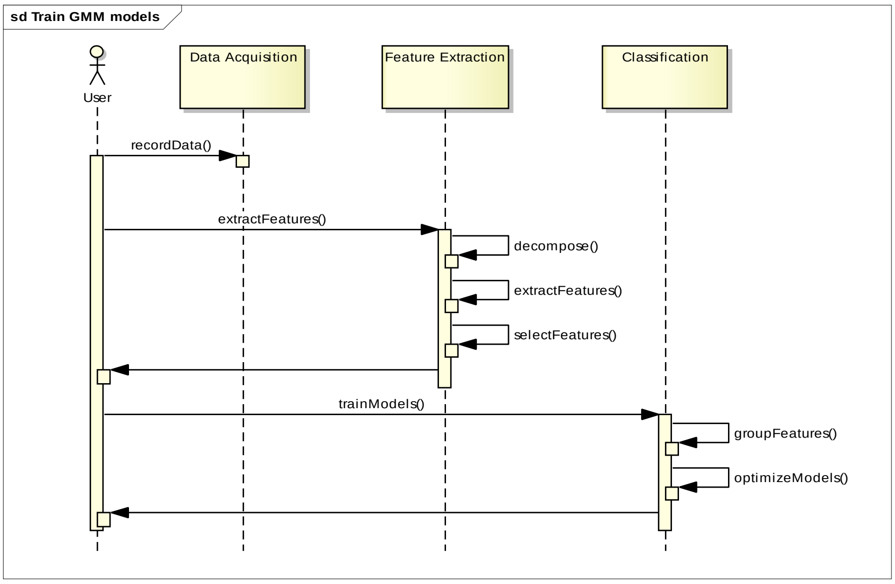


Figure 190 Sequence diagram of training phase of AER

- Initially, Data Acquisition is called to detect and record speech signal from audio stream.
- Main Process requests Feature Extraction to extract feature for all training speech signal. Inner processes are called to generate feature vector following Matching Pursuit algorithm.
- Finally, training stage of GMM algorithm is used to train models for all emotional classes.
- Following diagram show the recognizing phase sequence diagram.

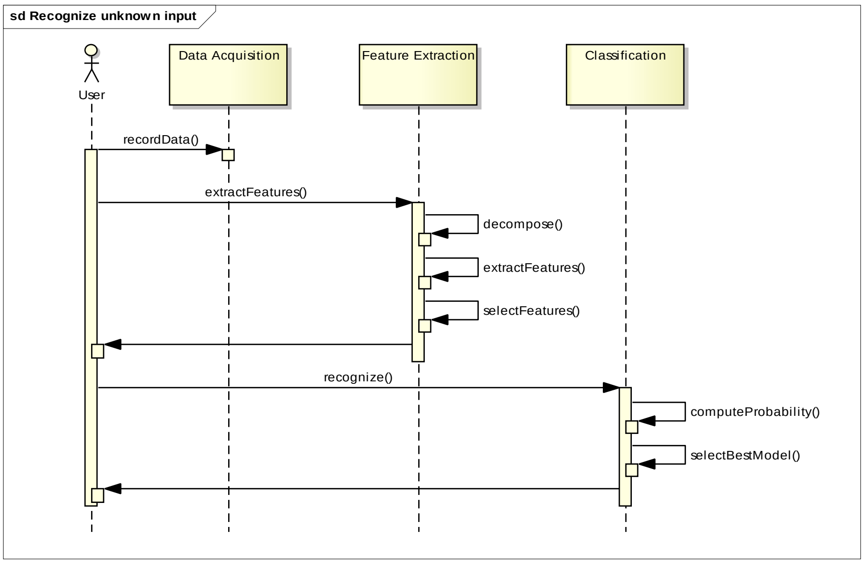


Figure 191 Sequence diagram of testing phase of AER

- Similar with training, Data Acquisition is called to record unknown signal.
- Feature Extraction is applied for unknown signal also.
- GMM classifier takes feature vector from Feature Extraction and computes probability with all models to find the model that has the largest probability value and then assigns according emotion as output label.

▪ Class Diagram

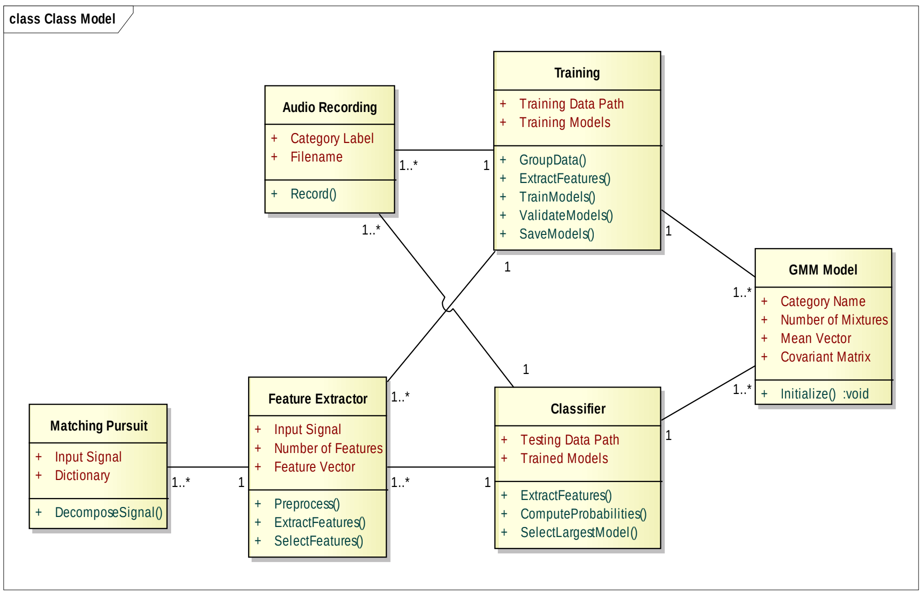


Figure 192 Class diagram of AER

- Similar with training, Data Acquisition is called to record unknown signal.
- Feature Extraction is applied for unknown signal also.
- GMM classifier takes feature vector from Feature Extraction and computes probability with all models to find the model that has the largest probability value and then assigns according emotion as output label.

▪ **Component Diagram**

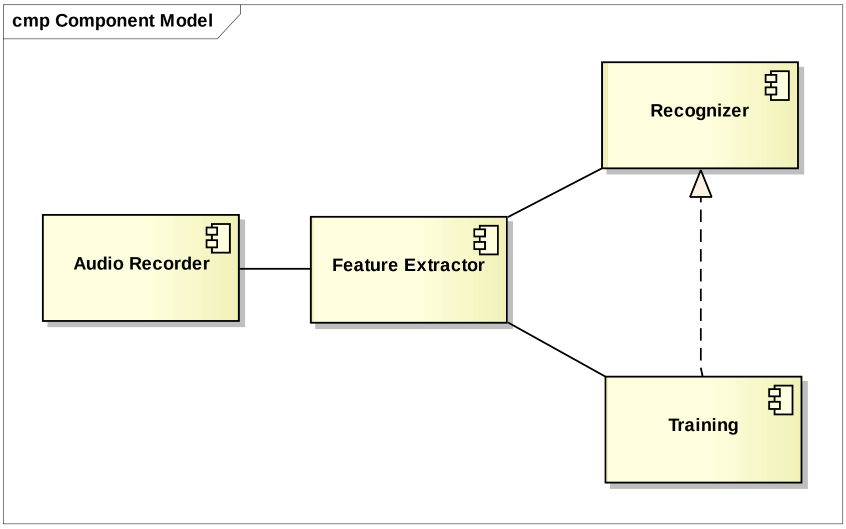


Figure 193 Component diagram of AER

- The component diagram shows different components and their relationships with each other. Mainly there are four main components and are explained below.
- Audio Recorder: record and preprocess audio, group training data into different emotion categories
- Feature Extractor: extract and select feature from training and testing data
- Training: use training data to generate Gaussian mixture model for each emotion category
- Recognizer: use trained model to predict emotion label from unknown input signal

6.2.1.2. Video Based Activity Recognition

▪ **Use Case Diagram**

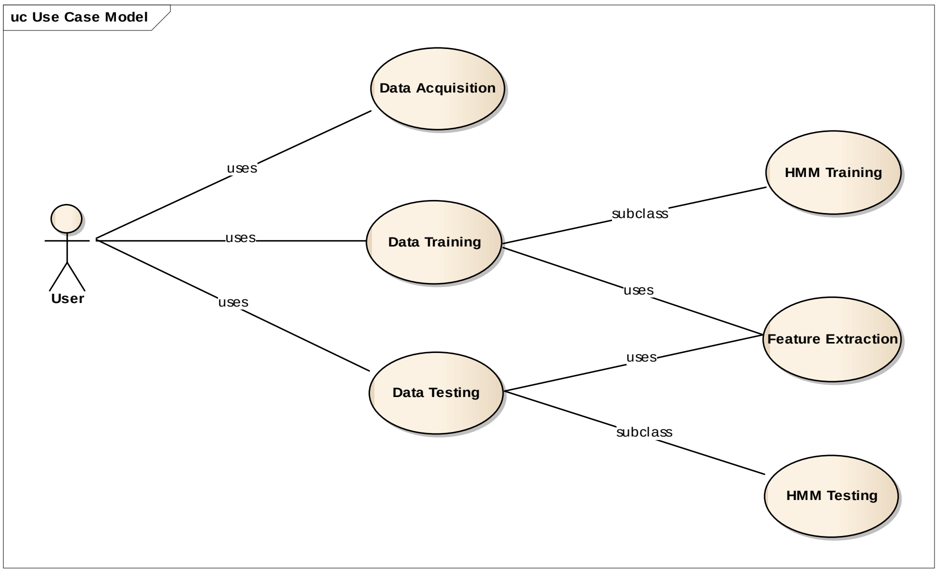


Figure 194 The use case diagram of human activity recognition system.

- **Description.** It communicates with the video camera and different component of activity recognition system are turned on.
- **Actors.** Human communicate with the 2D camera.
  - **Assumptions.** The common supposition is to employ machine learning techniques.
  - **Pre-conditions.** Processed and cleaned data for training purpose.
  - **Basic Flow.** The essential flow should be the incidents of the interaction with human, and there are no errors, no omissions that will be handled to preserve the logs and alerts to the corresponding person

- **Sequence Diagram**

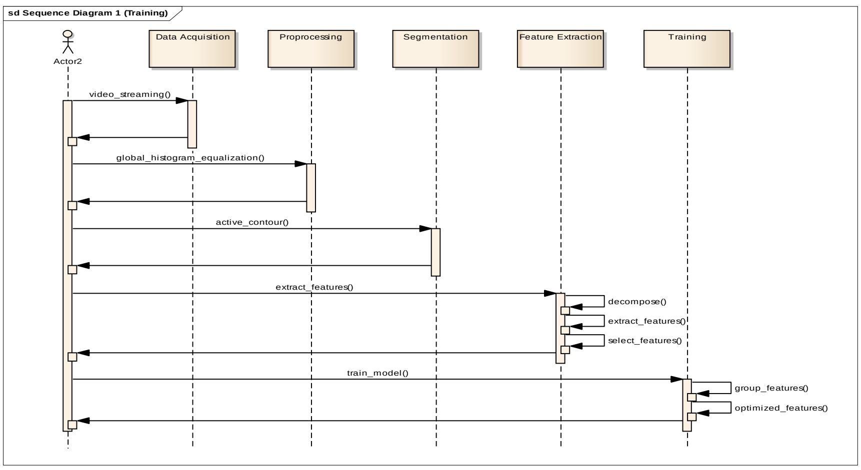


Figure 195 The sequence diagram of human activity recognition for training.

- At the beginning, in the main process the video of the human activity recognition is streamed and cleaned from noise at data acquisition stage.
- Some environmental effects such as lighting effect can be diminished at the preprocessing stage via Global Histogram Equalization (GHE) technique.
- An un-supervised technique such as active contour is employed to segment the human body automatically from the video frame at the segmentation stage.
- A robust technique called wavelet transform is employed to extract features from the segmented body.
- The system will be trained with suitable activity labels.

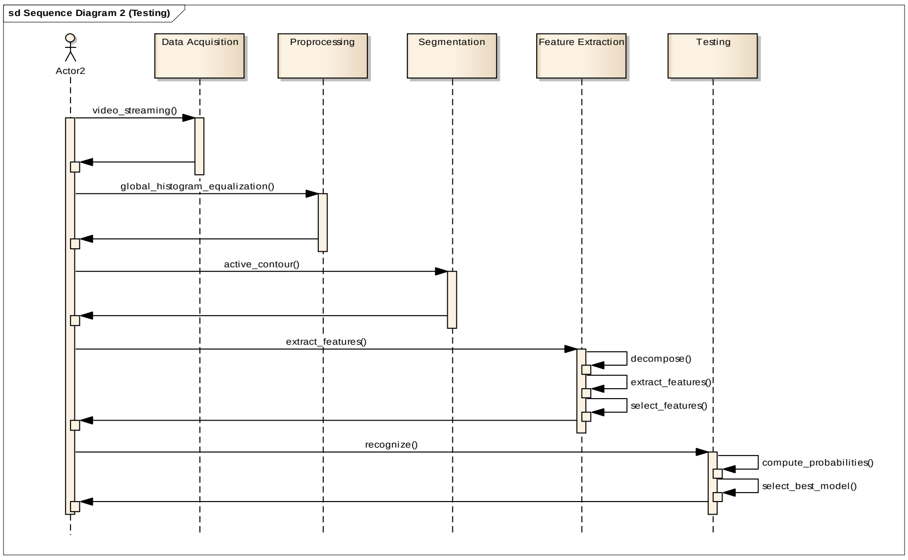


Figure 196 The sequence diagram of human activity recognition system for testing.

- In testing (recognition) phase, the streamed video frames are cleaned from environmental facts and then the human bodies have been segmented from individual activity frame.
- The features has been extracted and selected with the help of wavelet transform.
- The further activities have been recognized according to the trained labels

- **Class Diagram**

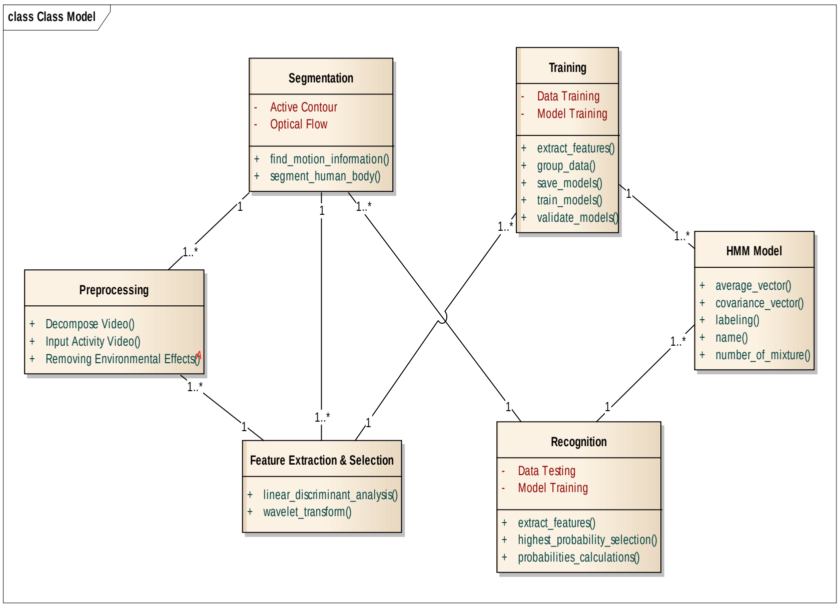


Figure 197 The class diagram for human activity recognition system.

- The class diagram shows how different classes related with each other to provide appropriate decision of activity recognition.
- In preprocessing class, the illumination and lighting effects are diminished to increase the recognition accuracy, using techniques like morphological filters, homomorphic filters, or median filters.
- In segmentation class an unsupervised segmentation technique such as active contour has been exploited for human body segmentation and the motion information between the two consecutive frames has been found by employing optical flow.
- The feature extraction module deals with extracting distinguishable features for each expression and quantizing each of them as a discrete symbol. Therefore, in feature extraction, wavelet transform has been used to extract the useful global and local features. Moreover, for feature selection a well-known statistical approach named linear discriminant analysis has been exploited.
- In recognition class, a classifier is used to train and to generate a label for the human activity recognition contained in the incoming video data. Among all of the classifiers, hidden Markov model (HMM) can frequently be employed for sequential data such as activity recognition. HMM is trained and tested to recognize incoming activity frames.

- **Component Diagram**

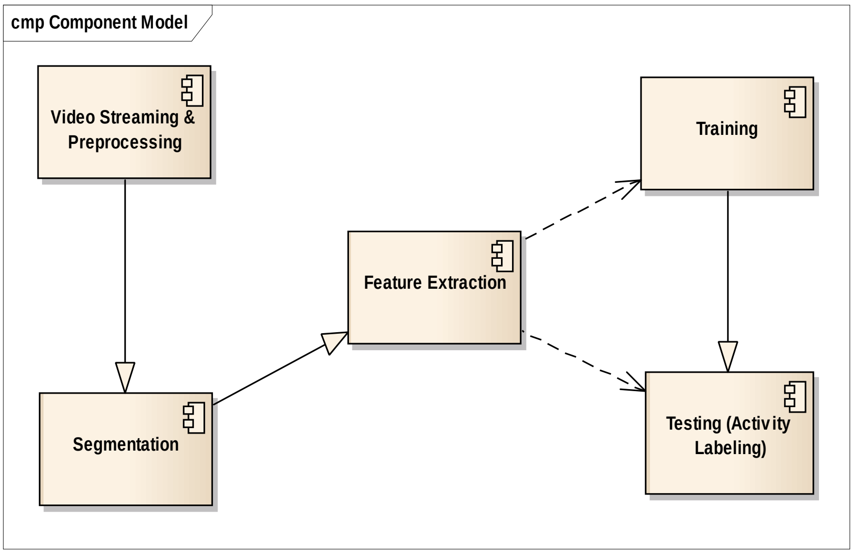


Figure 198 Component Diagram of Activity Recognition System.

- **Video Streaming & Preprocessing component**
  - This component loads the video activity and divides it into number of frames and diminishes some environmental effects such as lighting effects.

- **Segmentation Component**
  - Accurate human body segmentation techniques are required in order to achieve best accuracy of recognition; therefore in this component an unsupervised segmentation has been exploited to segment the human body automatically.

- **Feature Extraction Component**
  - The accuracy of recognition module is completely reliant on the extraction of good features that means, the effectiveness of the extracted features will also affect the accuracy of recognition module. Therefore, in this component some informative features have extracted in order to make the decision for classifier.

- **Training Component**
  - After feature extraction, in this component, a classifier is trained according to suitable activity labels that further can be used for activity classification.

- **Testing component**
  - In this component, future activities have been recognized based on the trained labels.

6.2.1.3. Multimodal Sensor Fusion

- **Use Case Diagram**
  - Following figure shows the interaction of the component to construct the activities and

emotion profile.

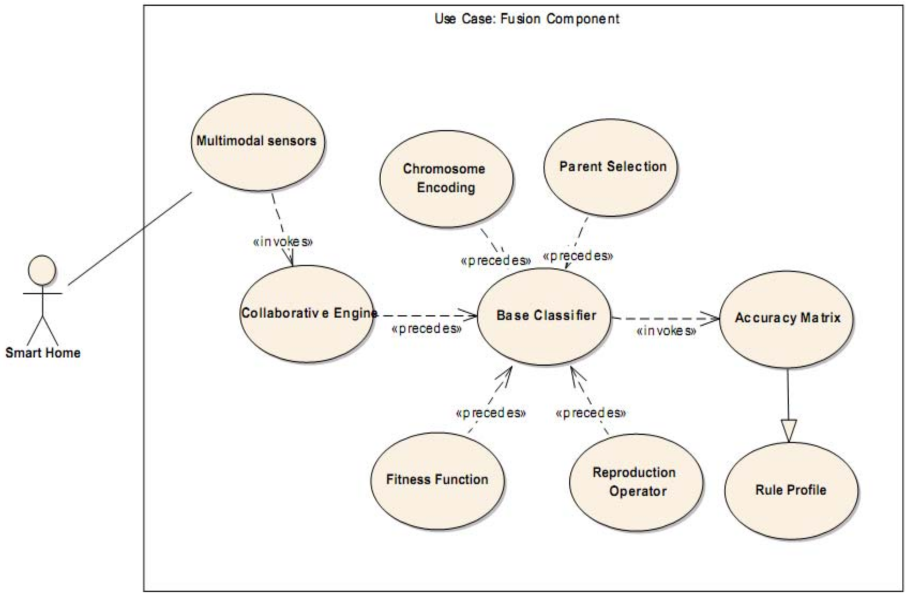


Figure 199 Use Case diagram of Multimodal Sensor Fusion

- **Use case Description**
  - **Multimodal Sensors:** This use case receives the multimodal sensory information from deployed sensors.
  - **Collaborative Engine:** This use case is responsible to transform the heterogeneous information into required format.
  - **Base Classifier:** This use case is used to train the model to obtained the optimized rule set.
  - **Chromosome Encoding:** It is responsible to formulate the activity recognition problem into genetic problem.
  - **Parent Selection:** The responsibility of this use case is to select the best parent for the crossover operation.
  - **Fitness Function:** It evaluate the fitness of each chromosome and assign score accordingly.
  - **Reproduction Operation:** This use case reproduces the solution by applying crossover and mutation operator with certain probability.
  - **Accuracy Matrix:** This use case assigns the importance to the rules for getting more accurate results.
  - **Rule Profile:** This use case maintains the rules profile for recognition phase.

▪ **Sequence Diagram**

- Following training and testing phase sequence diagrams illustrate the chronological sequence of messages between objects in an interaction:

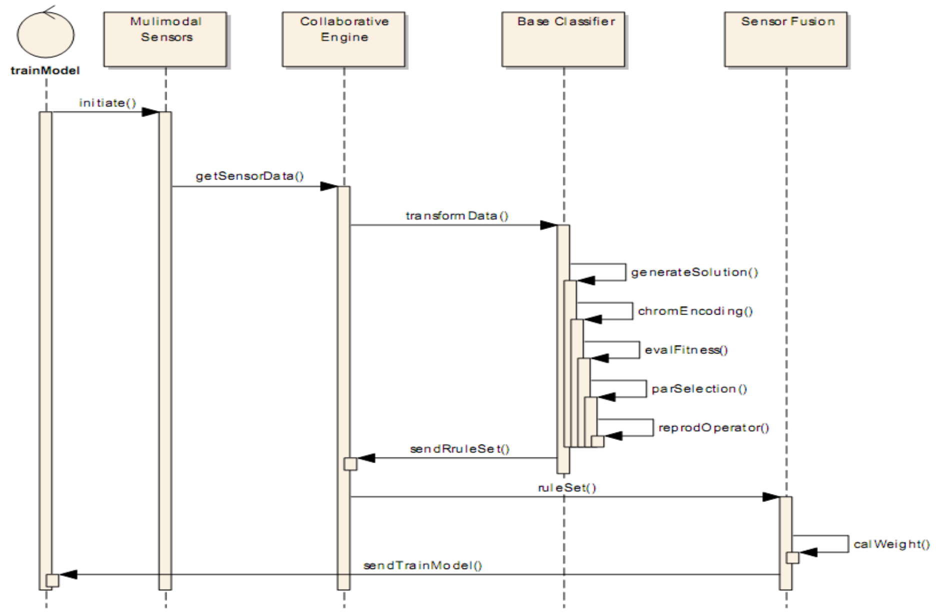


Figure 200 Training sequence diagram of the module

- Initially, collaborative engine receives the training data from the multimodal sensors and transform it into the specific input format for training the model.
- Base classifier is interacts with the collaborative engine and receive the transformed data and run the evolutionary technique Genetic Algorithm (GA) to extract the rule profile.
- Accuracy matrix assigns the importance to rule and help to make decision during recognition phase.

Following diagram shows the recognition phase sequence diagram

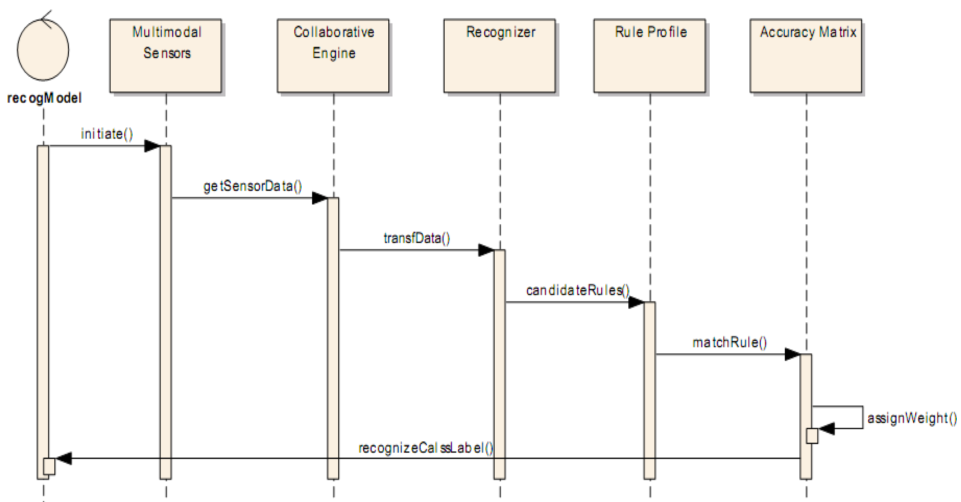


Figure 201 Recognizer sequence diagram of the module

- In recognition phase, data acquisition mode grabs the data and same process is repeated in the transformation mode.
- After transformation mode, rule matching is done over the rule profile and importance is figure out through accuracy matrix.
- Finally, emotions and activity labels are assigns on the current situation.

▪ **Class Diagram**

- It consists of the main class that interacts with the internal classes of the component.
- In order to decouple the code we made each module implementation separately and interact with through the proper interfaces.

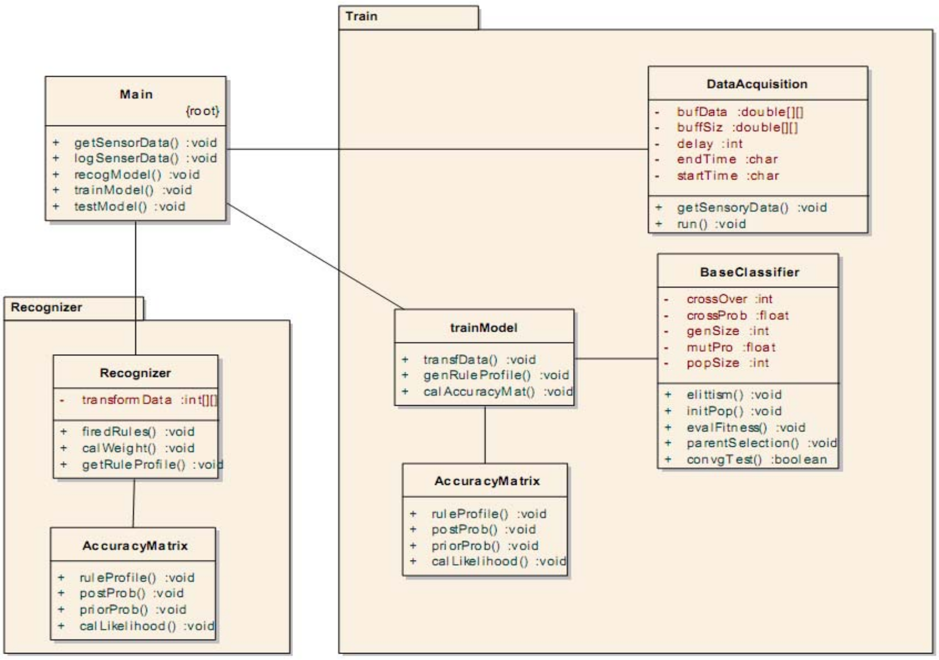


Figure 202 Class diagram of the overall module

▪ **Component Diagram**

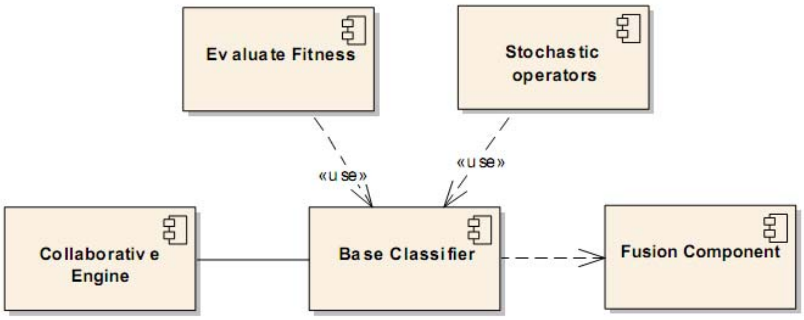


Figure 203 Component diagram of fusion engine

- The component diagram shows different components and their relationships with each other. Mainly there are three main components and are explained below.
- Collaborative Engine: transform the heterogeneous data formats into uniform format.
- Base Classifier: extracts the rule profile by using stochastic operators and fitness function.
- Fusion Component: calculate the accuracy matrix for assigning the weights to extracted rules.

6.2.1.4. Emotion Recognition

- **Use Case Diagram**

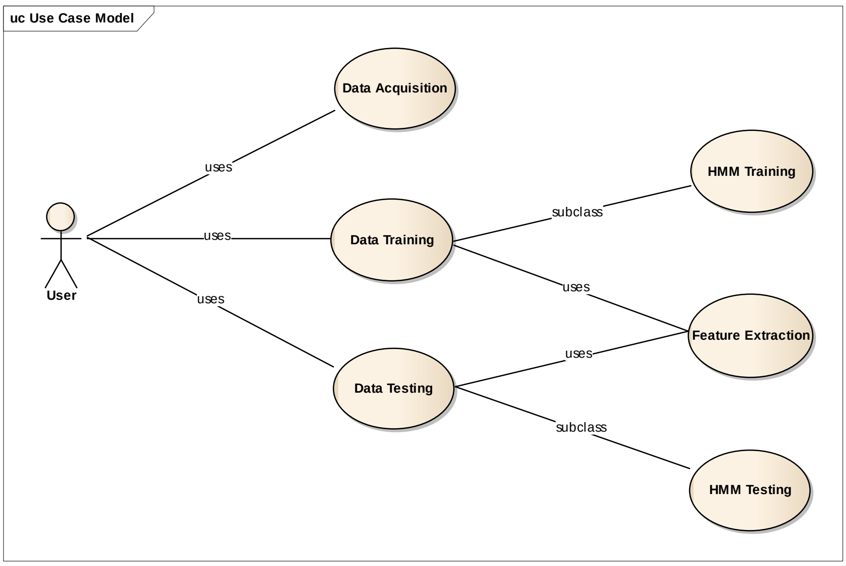


Figure 204 The use case diagram of automatic facial expression recognition system.

- **Description.** It communicates with the video camera and different component of FER system are turned on.
- **Actors.** Human communicate with the 2D camera.
  - **Assumptions.** The common supposition is to employ machine learning techniques.
  - **Pre-conditions.** Processed and cleaned data for training purpose.
  - **Basic Flow.** The essential flow should be the incidents of the interaction with human, and there are no errors, no omissions that will be handled to preserve the logs and alerts to the corresponding person

▪ **Sequence Diagram**

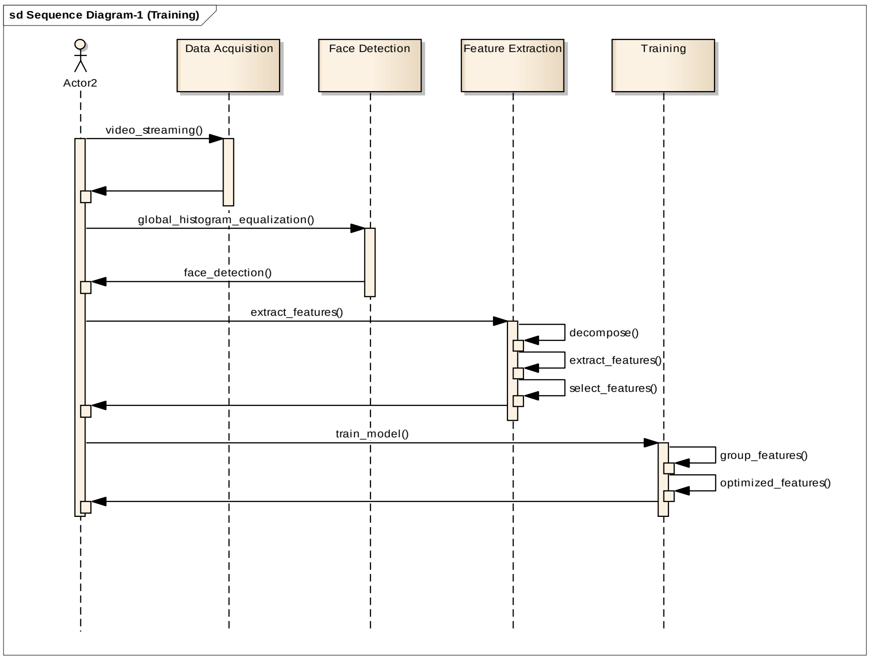


Figure 205 The sequence diagram of automatic facial expression recognition for training.

- At the beginning, in the main process the video of facial expression is streamed and some environmental effects such as lighting effect can be diminished at data acquisition stage.
- New techniques named Gray-level and Skin-toe are employed to detect the face from the video frame.
- A well-known statistical techniques; such as principle component analysis independent component analysis and discriminant analysis are employed to extract and select features.
- The system will be trained with suitable facial expression labels.

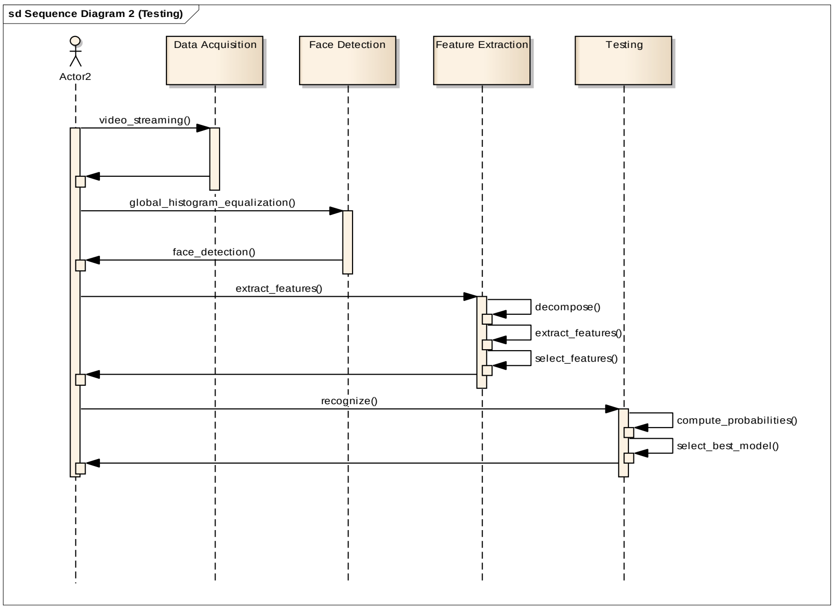


Figure 206 The sequence diagram of automatic facial expression recognition for testing.

- In testing (recognition) phase, the streamed video frames are cleaned from environmental facts and then the faces have been detected from each expressions frame.
- The features has been extracted and selected with the help of a linear classifier named principle component analysis and independent component analysis followed by linear discriminant analysis.
- The further expressions have been recognized according to the trained labels.

- **Class Diagram**

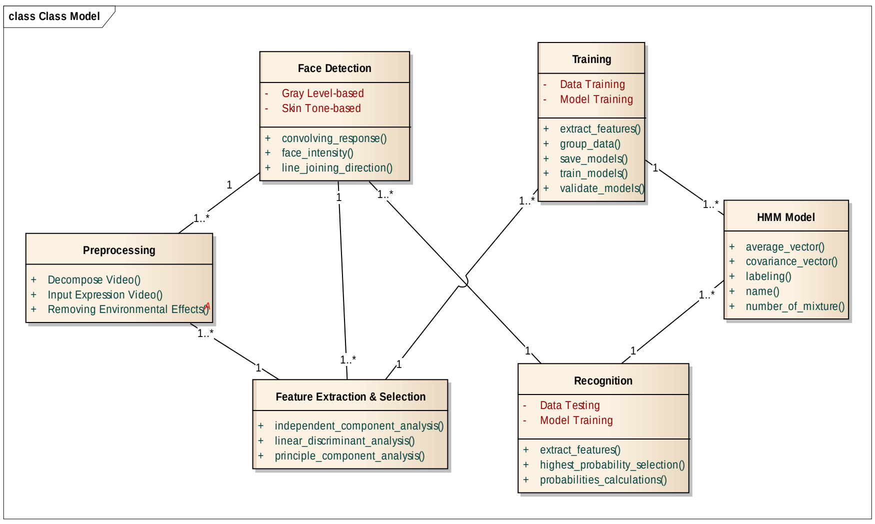


Figure 207 The class diagram for facial expression recognition system.

- The class diagram shows how different classes related with each other to provide appropriate decision of facial expression.
- In preprocessing class, the illumination and lighting effects are diminished to increase the recognition accuracy, using techniques like morphological filters, homomorphic filters, or median filters.
- Face contains most of the expression-related information, so, before investigation the expression, face must be detected first. An accurate facial expression recognition system requires automatic face detection which is considered the essential part of the facial expression system. So in face detection class the faces are detected and extracted first.
- The feature extraction module deals with extracting distinguishable features for each expression and quantizing each of them as a discrete symbol. Therefore, in feature extraction and selection some well-known statistical approaches are applied to extract and select informative features.
- In recognition class, a classifier is used to train and to generate a label for the human facial expression contained in the incoming video data. Among all of the classifiers, hidden Markov model (HMM) can frequently be employed for sequential data such as facial expressions. HMM is trained and tested to recognize incoming expression frames.

▪ **Component Diagram**

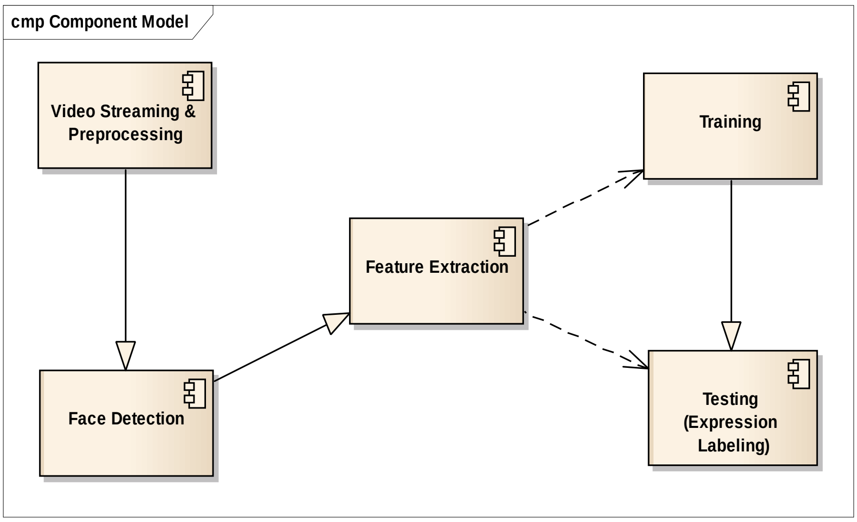


Figure 208 Component diagram of video based emotion recognition system.

- **Video Streaming & Preprocessing component**
  - This component loads the video of facial expression and divides it into number of frames and diminishes some environmental effects such as lighting effects.

- **Face Detection Component**
  - Face detection and extraction is an important step before investing the facial expres-

sion, therefore in this component a robust and an accurate has been employed to detect and extract the face from the video frames.

- **Feature Extraction Component**
  - The accuracy of recognition module is completely reliant on the extraction of good features that means, the effectiveness of the extracted features will also affect the accuracy of recognition module. Therefore, in this component some informative features have extracted in order to make the decision for classifier.

- **Training Component**
  - After feature extraction, in this component, a classifier is trained according to suitable expression labels that further can be employed for expression classification.

- **Testing component**
  - In this component, future facial expressions have been recognized based on the trained labels.

6.2.2. Context-aware Activity Manipulation Engine

- LDSS for Chronic Disease takes the input form CAME engine as well.
- Context-aware Activity Manipulation Engine (CAME) is one of the main components of SC^3 and also an inessential component of LDSS for Chronic Disease.
- CAME is the process of inferring high level activities from low level activities recognized by different sensors.
- The component based framework architecture diagram of CAME and the information flow is given in main architecture of LDSS for Chronic Disease.
- The Activity Extractor component extract activity related information from XML and Text files.
- Then with the help of context information available in Knowledgebase and customized rules, CAME infer high level/actual activity performed by human body.
- Based on the activities performed, CAME also gives suggestions and makes decisions in different environment with the help of context information available.
  - For example; we have all the context/profile information (i.e. professor name, designation, current courses, class room no, and class timings) about a professor in the knowledgebase (ontology).
  - Now if Professor enters a class room on his class time, then the body, motion, location and video sensors will recognize that Professor has entered the class room at a specified time.
  - Then CAME using these information from the sensors and information available in the Knowledgebase infers that its lecture time of Professor.
  - So the system starts issuing commands for turning off class room lights, turn on computer and turn on plus scroll down multimedia (projector) in class room

- **Use Case Diagram**
  - The use case diagram of CAME is shown in Figure 209, where as its description is given below.
  - **Actors:**

- **Admin:User:** Admin:User is the main user of the system that is using CAME for activity analysis and decision making.
- **ActivityRecognitionEngine:** Is the main source of input to CAME. The activity information coming in into CAME is manipulated and decision is propagated to the consumers.

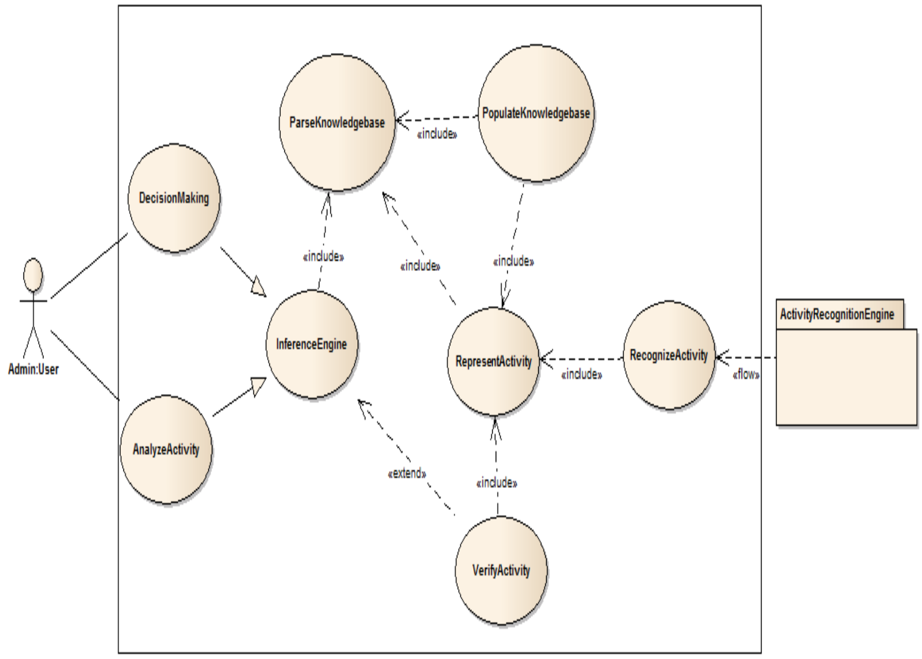


Figure 209 Use case diagram of CAME

- **Use case Description**
  - **RecognizeActivity:** This use case receives the incoming activities from the sensors and identifies its format.
  - **RepresentActivity:** This use case is responsible for representing the incoming activity from ActivityRecognizer in ontological structure while take help from VerifyActivity for verification of activity.
  - **VerifyActivity:** This use case is used to verify the incoming activity against the Knowledgebase for its consistency and existence.
  - **PopulateKnowledgebase:** It is responsible for logging the verified activity in the knowledgebase.
  - **ParseKnowledgebase: The** responsibility of this use case is to parse the knowledgebase for inference engine and its processing.
  - **InferenceEngine:** Inference engine involves the reasoning that include MatchMaking and filtering
  - **AnalyzeActivity:** This use case analyze the activity based on the reasoning and recognize the situation of user.
  - **DecisionMaking:** This use case makes the decisions based on the reasoning which

is performed in InferenceEngine.

- **Sequence Diagram**
  - Sequence diagrams for CAME are described below that shows the interaction between different objects of the system for the achievement of objectives.

  - **Activity Representation (Figure 210)**
    - When the activity is received in CAME then for its usage in Inference Engine and storage in Knowledgebase needs to be in Knowledgebase representational structure.
    - It calls the ActivityRecognizer using getActivityContents() and receive the activity.
    - It gets the representational structure from Knowledgebase by calling getRepresentation().
    - Which is executed by Parser and Knowledgebase is parsed for that. Then the activity is represented using representActivity().

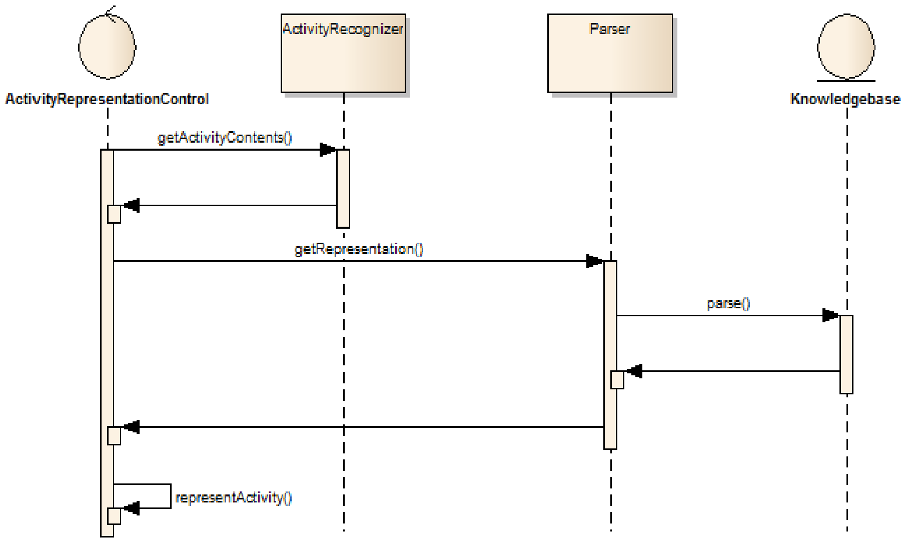


Figure 210 Sequence diagram of CAME Activity Representation

  - **Analyze Activity (Figure 211)**
    - The activity is first extracted using extractActivity() from Knowledgebase and then forwarded to Inferencer using analyzeActivity().
    - Inferencer performs matchmaking and then apply rule using applyRules() extracted from Rules
    - Then the result is displayed using displayResult().

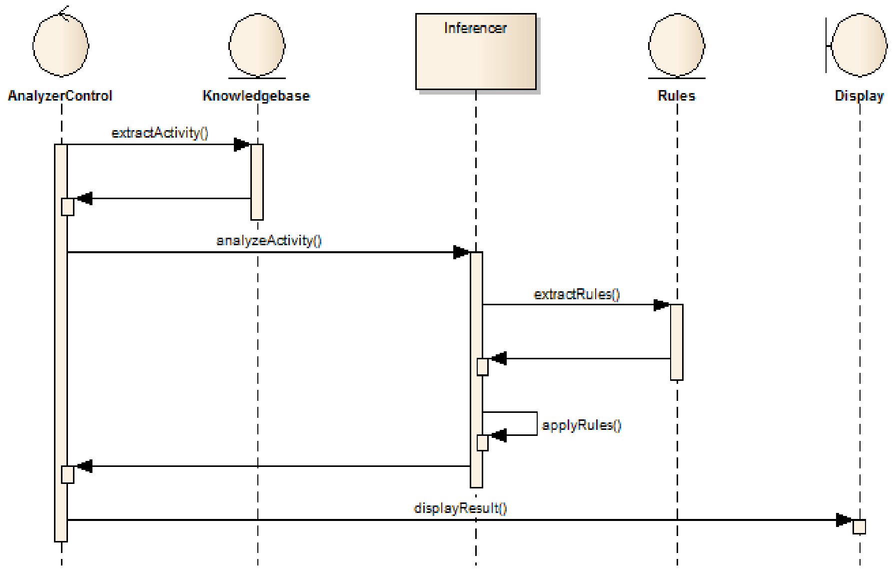


Figure 211 Sequence diagram of CAME Analyze Activity

- **Decision Making (Figure 212)**
  - The detected activity is first represented in the Knowledgebase representational structure and then the DecisionControl is activated.
  - The DecisionControl extract the relevant activities form Knowledgebase using matchmaking process.
  - Then the DecisionControl apply the rules extracted from Rules using applyRules().
  - The activity is then identified and decision is made. The decision is then displayed or propagated to DisplayDecision.

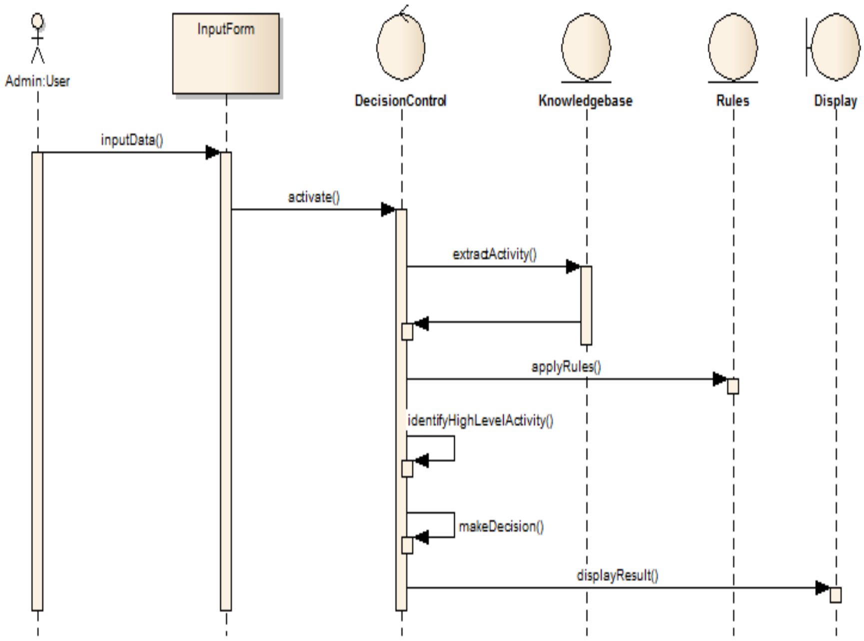


Figure 212 Sequence diagram of CAME Decision Making

- **Class Diagram**
  - Class diagram (Figure 213) of CAME shows the different classes and their relationships with each other.
  - Initially the ActivityRecognizer and ActivityRepresentation are loaded which work together for incoming activities.
  - Then the activity is Verification class is called for activity verification. The verified activity is then logged using PopulateKnowledgebase that uses Parser class for the job.
  - Inferencer class initiates the matchmaking and filtering for the new activity and makes the situation analysis. This is also responsible for decision making.

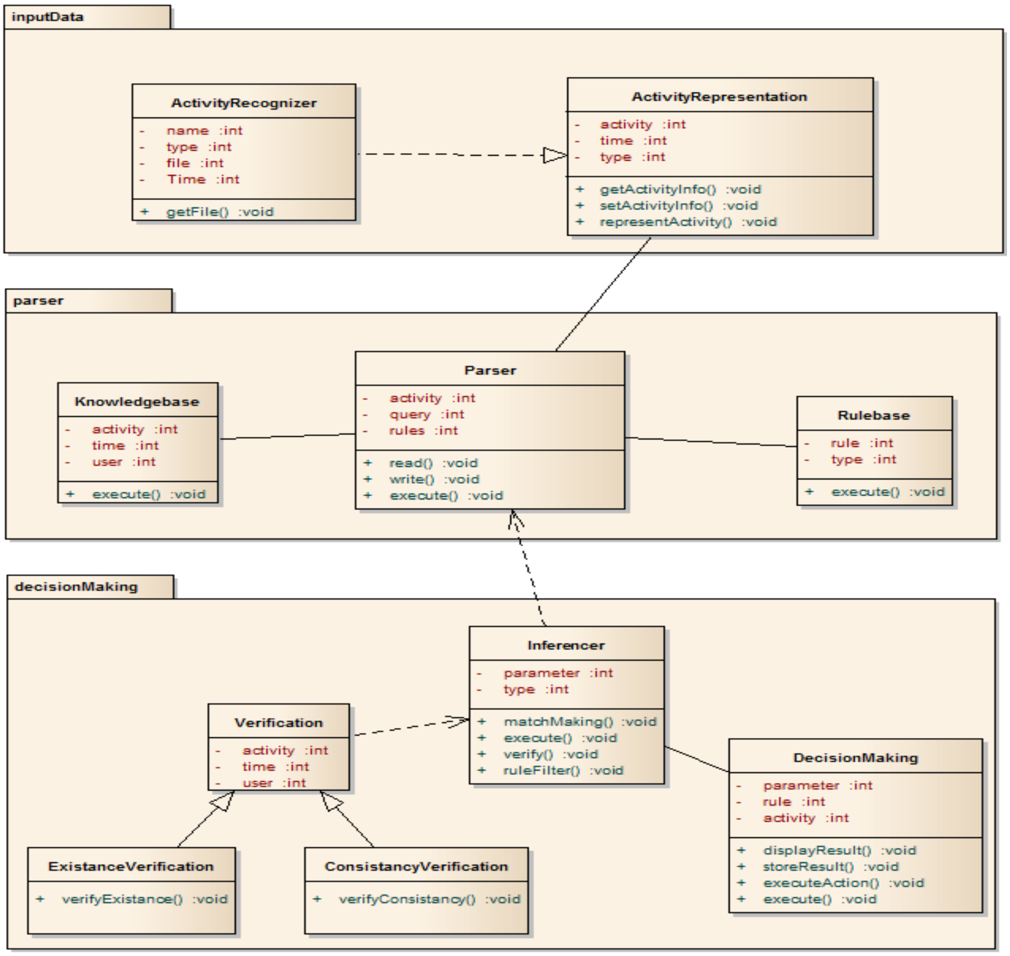


Figure 213 Class diagram of CAME

- **Component Diagram**
  - The component diagram of CAME (Figure 214) shows different components and their relationships with each other. Also it shows the subcomponents interaction, purpose, and their relationships with each other. Mainly there are four main components and are explained below.

  - **Activity Recognizer Component**
  - It is the main component of CAME that is responsible for collecting the activity information coming from different sources.

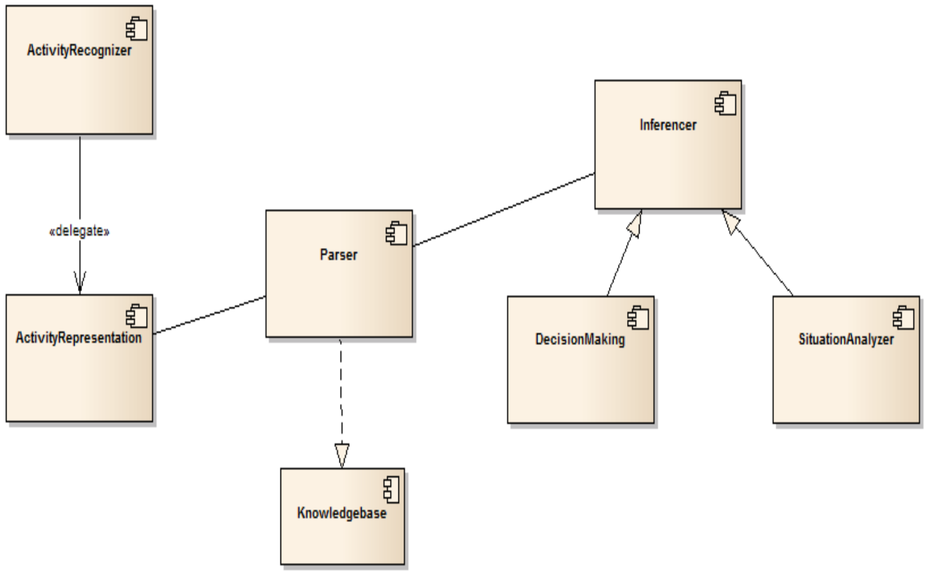


Figure 214 Component diagram of CAME

- **Activity Representation Component**
- It represents the incoming activities information in the ontological structure. It also uses the help of verification for verifying the activities consistency.

- **Parser Component**
- It is responsible for CAME components interaction with Knowledgebase for activity logging and extraction. It is the main component that is always connected with the Knowledgebase and fulfills the jobs coming for knowledgebase.

- **Inferencer Component**
- Inferencer contains the brain of CAME. It is the main component that performs the matchmaking and filtering. It uses the services of DecisionMaking and SituationAnalyzer for analysis and decisions.

6.2.3. Social Media Interaction

6.2.3.1. Tweet Analysis

- **Use Case Diagram**

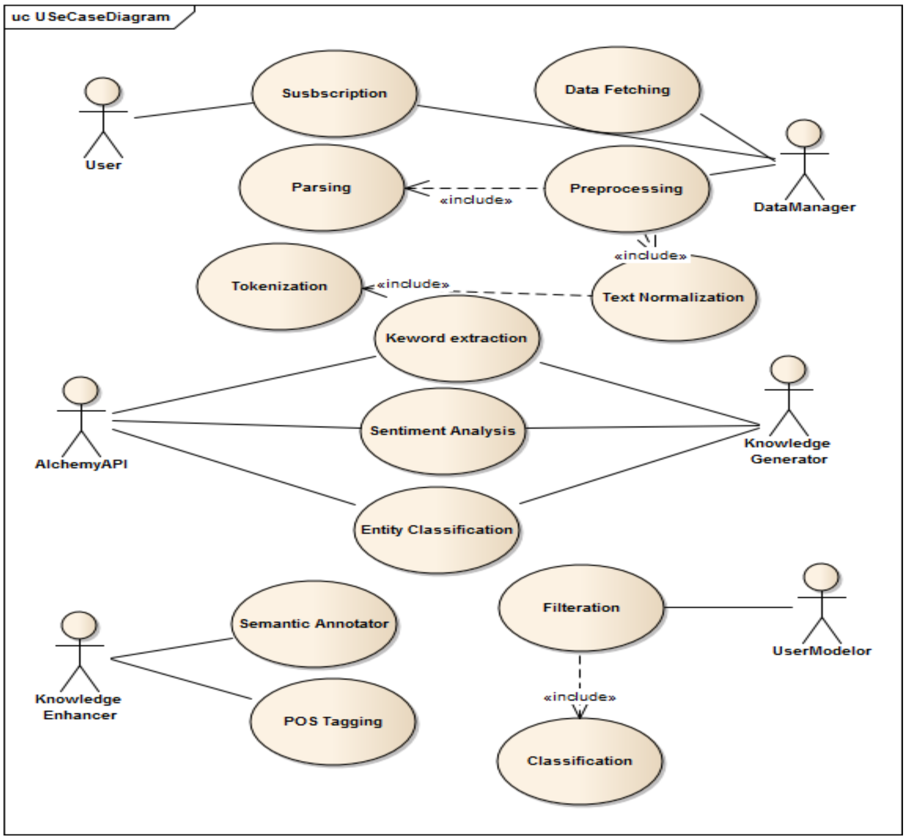


Figure 215 Use case diagram of Tweet Analysis for wellness

- **Actors:**
  - **Alchemy API**
    Third party component use to extract knowledge from tweets.
    Knowledge Enhancer
    Application component used to enhance knowledge of the system
  - **User Modeler**
    Responsible to generate personalized profile on the basis of information extracted from tweets.
  - **Data Manager**
    Handles data access from twitter and system's data store.
- **Brief Descriptions of Use Cases**
  - **Subscription:** Responsible for register user with the application . Application send request to user and user verify application to access his tweets.
  - **Data Fetching:** This use case is responsible to fetch user tweets from Twitter to generate user profile.
  - **Preprocessing:** Data preprocessor handle with data parsing and slang removal from tweets.
  - **Parsing:** Parser use XML parser to convert XML data from twitter to standardized individual objects

- **Tokenization:** Tokenization converts sentences into separate word to lookup on each word individually.
- **Text Normalization:** This use case process tweets and search for slangs in tweets to map slangs with their respective original word.
- **Keyword extraction:** To extract meaningful information from huge text Keyword extraction process extracts important keywords. It accepts tweets and return known keywords present in those tweets.
- **Entity classification:** This use case is responsible for recognizing individual entities from text and classifies those entities into different groups.
- **Sentiment analysis:** Aim of this use case is to determine the attitude of a speaker or a writer with respect to entities and keywords. This helps to know user behavior towards specific entities.
- **Semantic annotator:** It is about attaching synonyms and definitions, to keywords and entities. It provides additional information about an existing piece of data.
- **POS tagging:** It is the process of marking up a word in a text as corresponding to a particular part of speech, i.e. relationship with adjacent and related words in a phrase sentences nouns, adjectives, adverbs, etc.
- **Classification:** Tweets may be classified into different domain specific categories on the basis of entities and keywords extracted from tweets.
- **Filtration:** A data filtration is a process that allows domain specific data to pass for personalized modeler and ignore other information.

▪ **Sequence Diagram**

- The objective of Interaction model of a system is to depict the process scenario of how different objects interact with each other. Life span and sequencing of objects are the prime components of any interaction diagram. Sequence diagrams are described in following section.

- **Stream Collection**
  - Data Fetcher sends request to Twitter to collect tweets. After tweet collection from Twitter it passes data to preprocessor.

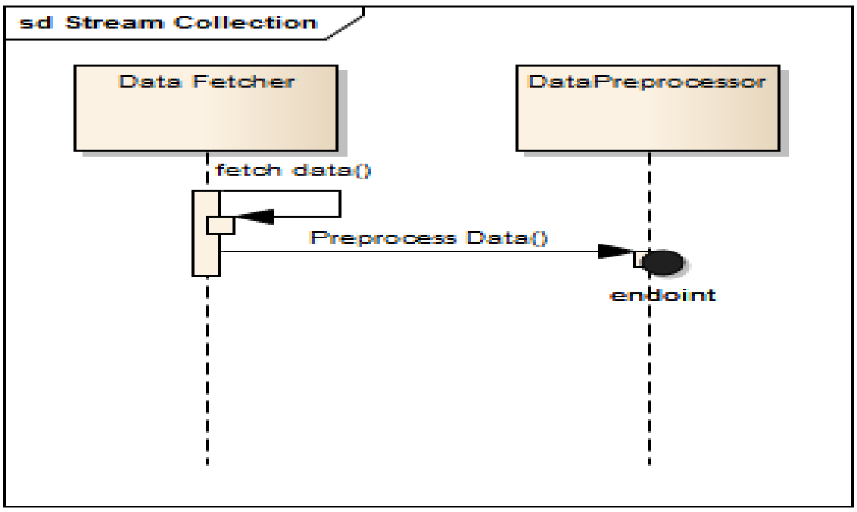


Figure 216 Stream Collection for wellness

- **Preprocessing**
  - Data preprocessor component processes to identify presence of any short hand notation and normalizes them to meaningful words. It passes data to XML parser. XML parser parses data using DOM parser and return parsed data. Preprocess then pass data to text normalizer. Text normalizes call tokenizer and passes data to apply tokenization on data. Tokenizer returns tokenized data to text normalize. Text normalizes replace slangs with original words.

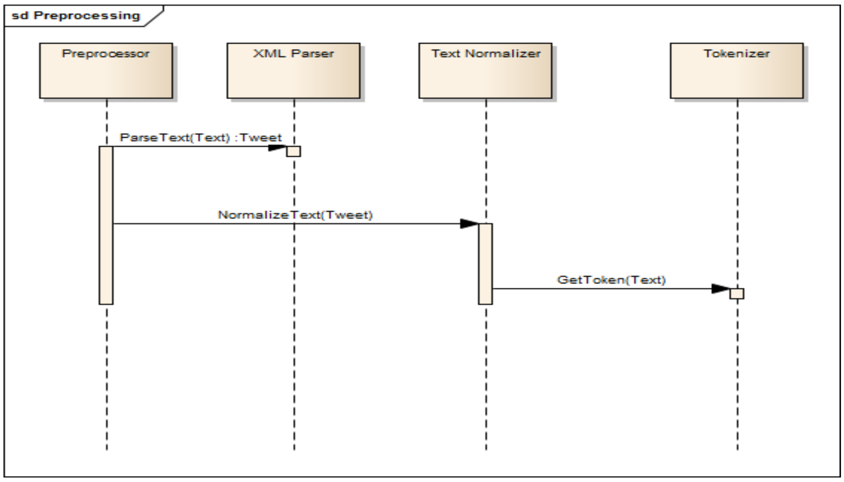


Figure 217 Data Preprocessing for wellness

- **Knowledge generation**
  - Knowledge generator pass tweets to keyword extractor which extracts known key phrases from tweets. Knowledge generator then pass tweet to entity classifier which identify different entities from tweets and return entities and their type. Sentiment analysis returns user sentiments towards those entities and keywords.

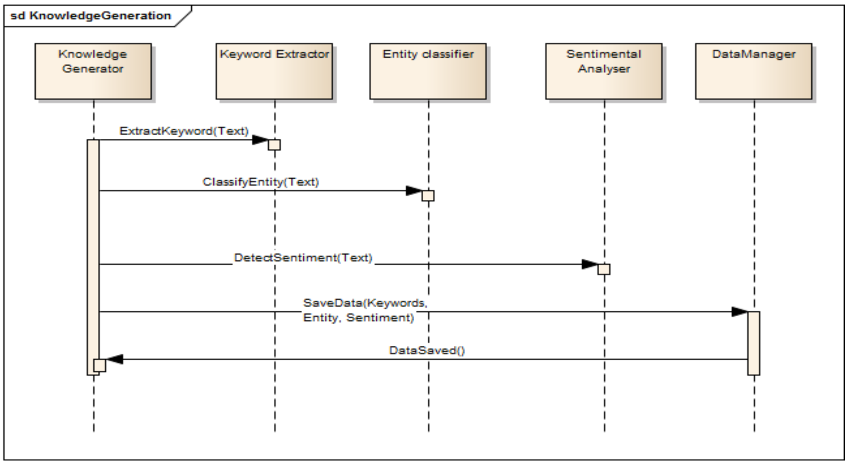


Figure 218 Knowledge Generation for wellness

- **Knowledge Enhancer**
  - Knowledge enhancer module passes tweet to part of speech tagger which split text into noun, verb, object, subject etc. Then knowledge enhancer pass keywords and entities to semantic annotator which use wordnet to increase knowledge by adding synonyms and definition.

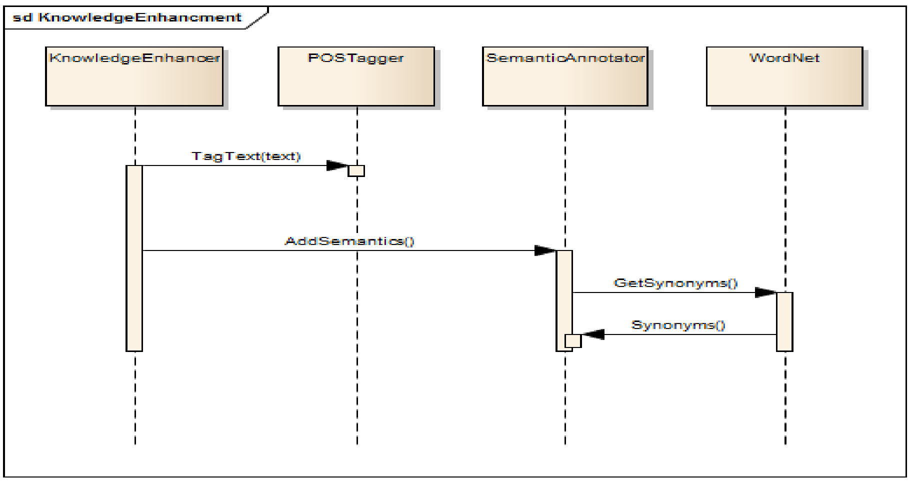


Figure 219 Knowledge enhancement for wellness

- **User modeling**
  - User modeler passes data to filter which filters data to make it domain specific like for health care it only passes information which is related to health care and ignore other information. Then information is passed to data manager which maintains user profile for future use.

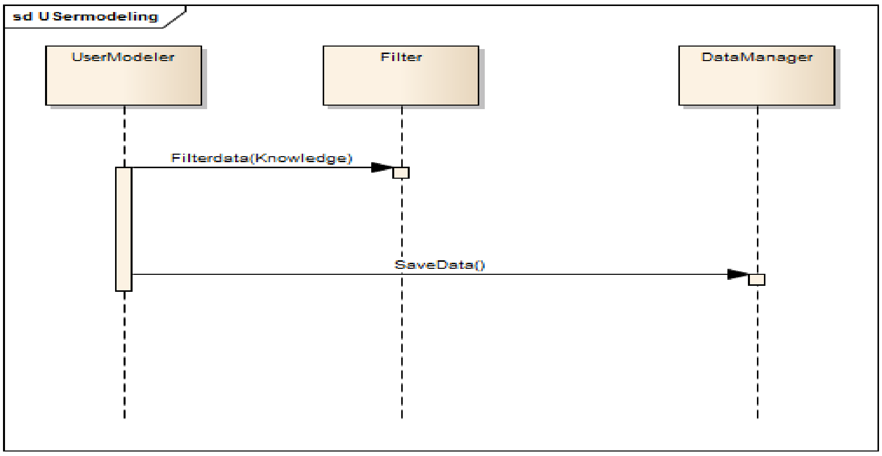


Figure 220 User Modeling for wellness

- **Class Diagram**

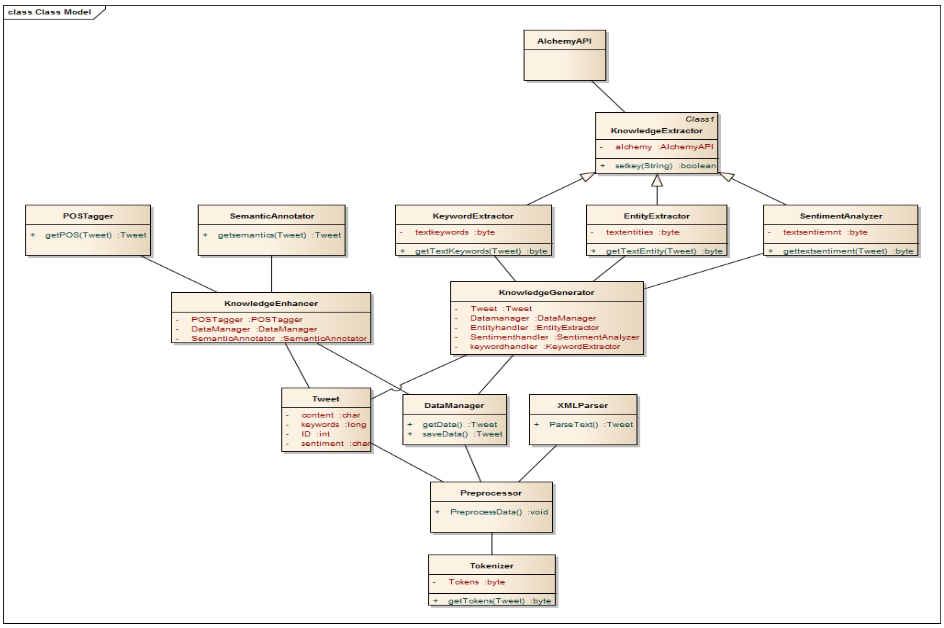


Figure 221 Class diagram of Tweet Analysis for wellness

- **Component Diagram**

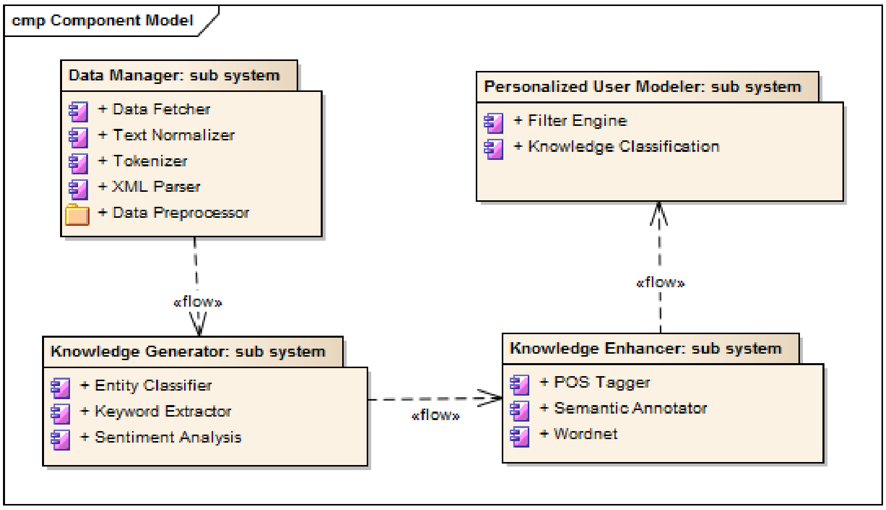


Figure 222 Component Model of Tweet Analysis for wellness

- **Data Manager**
  - This sub system is responsible for fetching data from social media and processing the fetched data. It has two parts.
- **Data Fetcher**
  - Data Fetcher sends request to social media for stream of user. The fetched data is in different format for each media.
- **Data Processor**
  - Fetched data requires some preprocessing before analyzing.
- **XML Parser**
  - XML parser uses Dom parser to convert XML data into required usable format and store data into different fields.
- **Text Normalizer**
  - Users use abbreviations to save time and space. Such kind of noise in data effects knowledge extracted from tweets. Therefore to remove such kind of noise, Text normalization removes slang and abbreviated word using slang lexicon
- **Tokenizer**
  - It split sentence into text based on the defined delimiter. Tokens are then used by text normalize to remove slang.
- **Knowledge Generator**
  - Knowledge Generator extracts user's interest by using Alchemy API. It obtains knowledge by exposing the semantic richness hidden in post.
- **Entity Classifier**
  - It extracts specific entities from tweets and type of those entities using alchemy API.
- **Keyword Extractor**
  - Keyword extractor use alchemy API to extract keywords from tweets.
- **Sentiment analysis**

- It involves detection of user sentiment related to keywords and entities to now user attitude towards different entities and keywords.

- **Knowledge Enhancer**
  - Knowledge enhancer enhances knowledge of system using semantic annotator and part of speech tagging to make system context aware.
- **POS Tagging**
  - To know about relation of user with entity and to add more knowledge system extract part of speech and then add verb into information already extracted by knowledge generator. It also adds knowledge about relation of subject or object with user.
- **Semantic Annotator**
  - Use of word net makes it possible to add semantic of individual keywords and entities by addition of synonyms and definition of each entity and keyword.
- **Personalized user modeler**
  - Better services delivery requires maintenance of history of individual's interest and behaviors. Personalized User Modeler maintains user's data in Personalized Profile.
- **Classification**
  - To provide domain specific service based on user temporal patterns system classify data into different domains and store them into profile.
- **Filter Engine**
  - Filter engine activated and filter domain specific knowledge to provide better personalized services.

6.2.3.2. Trajectory Analysis

- **Use Case Diagram**
  - Use case diagram of Trajectory Analysis is shown in Figure 223 Use Case Diagram.
  - **Actors:**
    - **Social Network Trend Setter:** The trendsetters are the well-known personalities to whom people want to follow. Activities schedule of these personalities are acquired and used as a reference for the user. User can also add this schedule by himself.
    - **User:** The individual, whose activities and movement routines are tracked and monitored. User is responsible for adding the semantic tags of activity location.

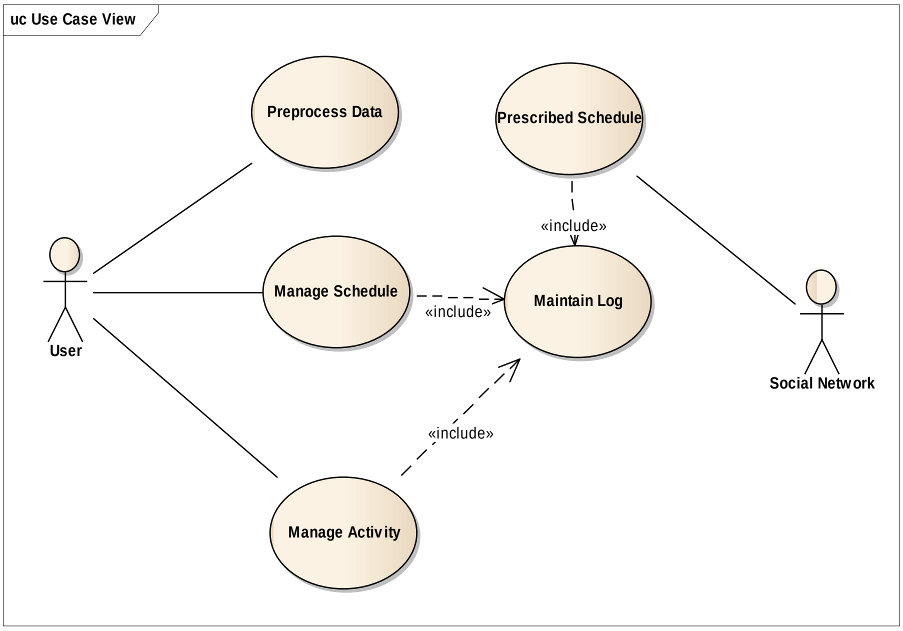


Figure 223 Use Case Diagram

- **Use Case Description**
  - **Manage Activity:** Manage Activity is main use case responsible for fetching the details of particular activity. As user changes its current activity location, system starts tracking the triggering of new frequent activity. Conversion of GPS coordinates, which are recorded by GPS enabled smart device of user into Geo tag is also a part of this use case.
  - **Manage Schedule:** In Manage Schedule user adds his preferred schedule if he/she want to add some recommended schedule. Schedule includes activities name, duration and number of occurrences in a defined period.
  - **Process Data:** Role of Preprocess data is to analyze the performed activities of user as compared to prescribed patterns by trendsetter or user recommended patterns. All the inconsistencies in performed activities are fetched and shown to the user in the form of a report.
  - **Add Prescribed Schedule:** Practitioner is required to add suggestions for a particular activity and also the complete schedule for the user. This prescribed schedule is taken as a reference and all the performed activities of user are evaluated against this schedule. This schedule is stored in repository using maintain log module.

▪ **Sequence Diagram**
- Sequence diagrams for Trajectory Analysis are described below that shows the interaction between different objects of the system for the achievement of objectives.
- **Analyze Performed Schedule:** Purpose of this interaction model is to track, monitor and evaluate user's schedule. On the change of activity location, movementTracker, fetches the required information of new activity using movementInformation(). GPS coordinates information is passed from movementTracker and corresponding semantic tags

of the location are acquired by the user. All the data are mapped with semantic tag and stored in the trajectoryRepositiory by the stroreActivity(). Activity Analyzer is the main function to analyze and compare the activity schedule. It fetches the performed activities of user of a particular period and compares it with the prescribed schedule of practitioner. Results of this module are sent to the network adopter. Below is the figure of Sequence Diagram of Analyze Performed Schedule describing this module.

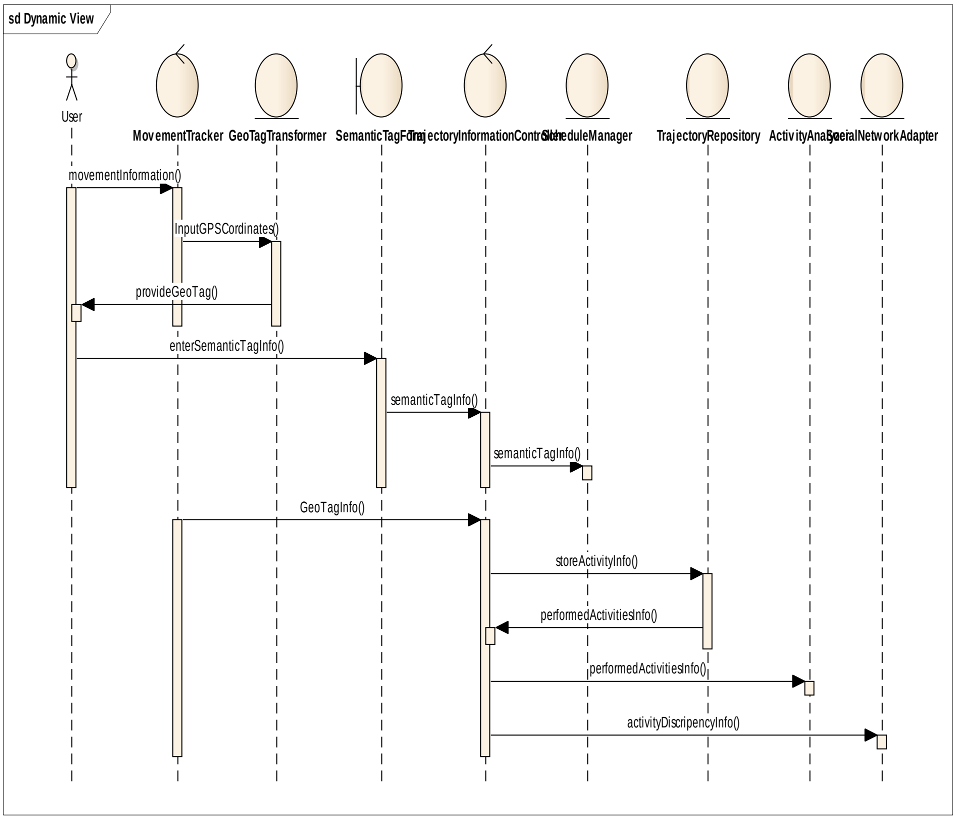


Figure 224 Sequence Diagram of Analyze Performed Schedule

- **Evaluate Performed Schedule:** Trendsetter module acquires the schedule from well-known personalities and also suggestions roe ah of activity is fetched. All of this information is stored in Trajectory Repository. Activity analyzer fetches this information and after comparing it with performed schedule of patient send it to social network adopter. Figure 225 Sequence Diagram of Evaluate Performed Schedule shows the sequence diagram of this module.

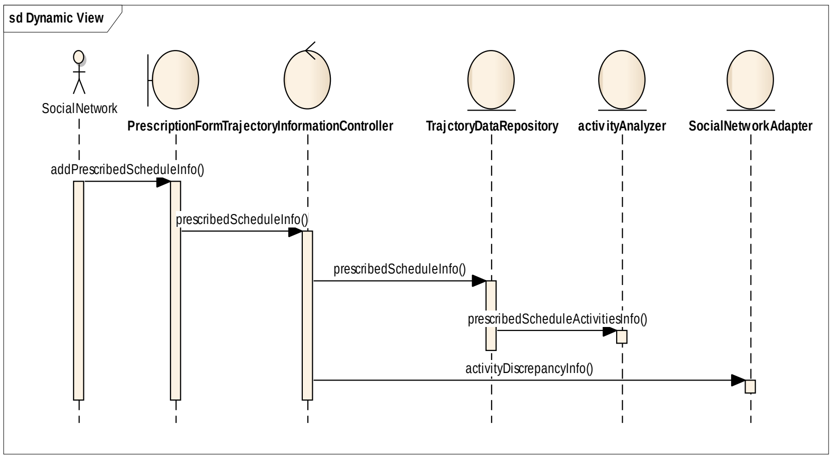


Figure 225 Sequence Diagram of Evaluate Performed Schedule

- **Class Diagram**
  - The purpose of class diagram is to show the relationships among the classes of the system. Additionally it covers the properties and operations that are visible at a higher level. Class diagram provides the high level overview of the system of how to be transformed into a detailed object model which ultimately helps to implement the system. Following section depicts the class diagram for Trajectory interaction for smartCDSS.
  - Task Assigned to the class Datapreprocessor is to fetch all the required information of trajectory and initial processing on it. ScheduleManager maps all the trajectory information with the semantic tag acquired from the user. Activity analyzer compares the user's schedule with the prescribed schedule and all the data stored in the repository by using class TrajectoryDataRepository. Figure 226 Class Diagram of Trajectory Analysis shows the class diagram of our system.

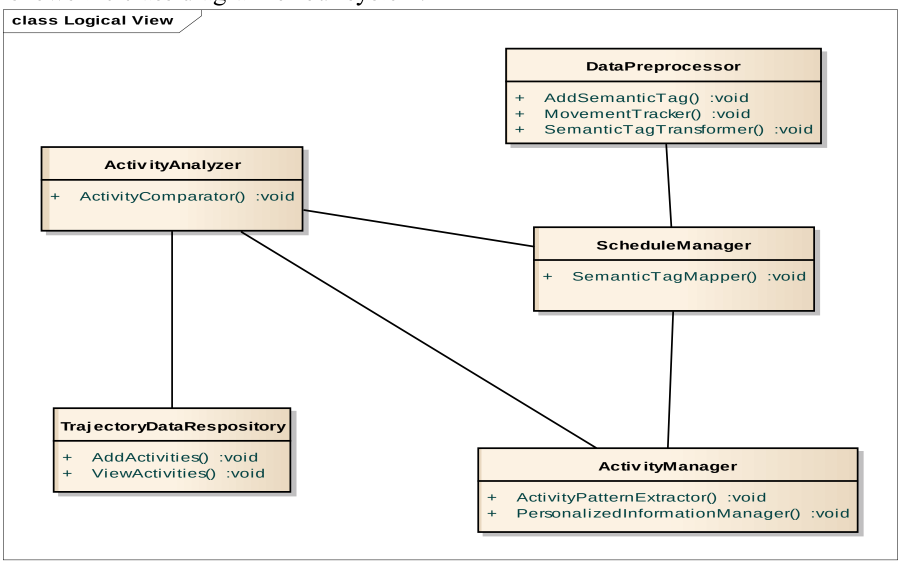


Figure 226 Class Diagram of Trajectory Analysis

- **Component Diagram**
  - A component is a software package, or a module, that encapsulates a set of related functions. All system processes are placed into separate components so that all of the data and functions inside each component are semantically related (just as with the contents of classes). Because of this principle, it is often said that components are modular and cohesive. By keeping above definition of software component, we divided our work into different components as shown in Figure below.
    - Data Preprocessor
    - Schedule Manager
    - Activity Manager
    - Activity Repository

  - **Activity Preprocessor**
    - Purpose of DataPreprocessor is to fetch all the data from the movement patterns of the patient. It further includes three parts. The imperative location finder is to detect that either particular location satisfy all the parameters of frequent patterns. Semantic tag acquires the context information of the particular activity location and task of Geo tag transformer is to convert GPS coordinates into Geo tag by using Google API.

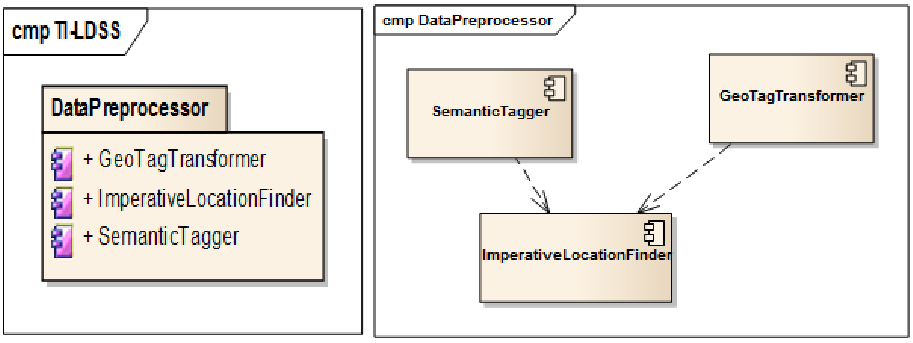


Figure 227 Component Diagram of Activity Preprocessor

  - **Schedule Manager**
    - Schedule Manager prepares and processes the data to find inconsistencies. It includes Semantic tag Mapper which links the semantic tag of location and other required information and pass it to Activity Manager and then to the trajectory repository. Purpose of second subpart, Activity analyzer is to compare the trendsetter schedule and user's performed schedule. Following is the component diagram of the schedule Manager.

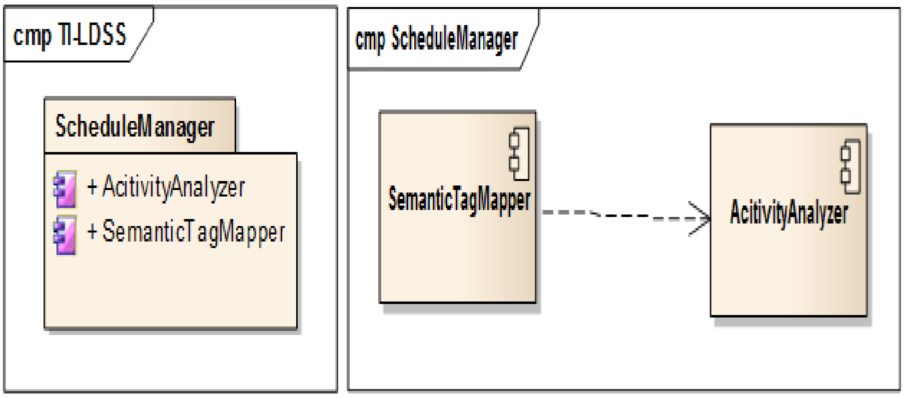


Figure 228 Component Diagram of Schedule Manager

- **Activity Manager**
    - Activity Manager is responsible for extracting the personalized information from the daily performed patterns of the user. For this first particular patterns are extracted from the activities and then passed into Personalized Information Manager.

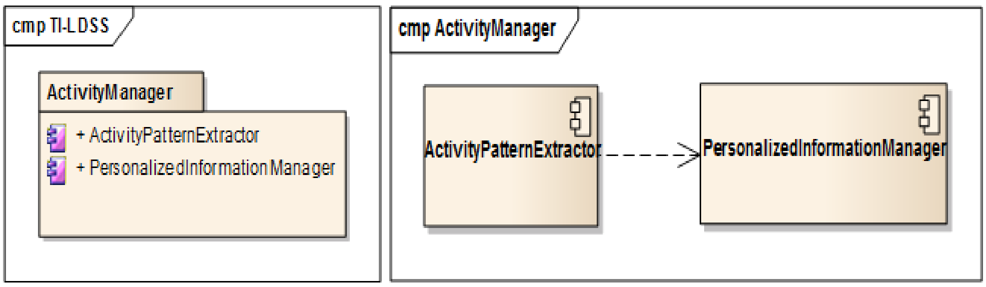


Figure 229 Component Diagram of Activity Manager

- **Trajectory Repository**
    - All the three kinds of data, 1) Performed patterns, the activities which are performed in daily life by the user. 2) Prescribed patterns, the schedule which is added by trendsetter and 3) Suggestions, the recommendations for each of the activity in pre-scribed schedule are stored in Trajectory Repository.

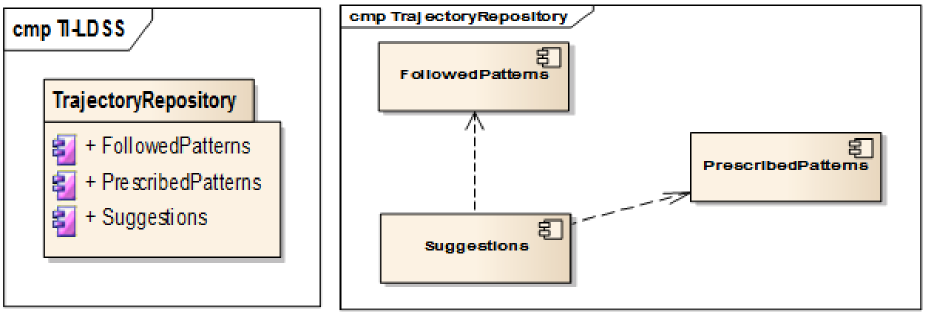


Figure 230 Component Diagram of Trajectory Repository

6.2.3.3. Interaction Analysis

- **Use Case Diagram**
  - The use case diagram of interaction analysis is shown in Figure 231, where as its description is given below
  - **Actors:**
    - **User:** The individual, whose emails are monitored and analyzed to find regular, frequent behavior.
    - **Keyphrases Extraction:** Application component used to extract keyphrases from the contents of interaction
    - **Data Manager:** Responsible to model graph on the basis of interaction between users with relevant keyphrases. Data Manager is also responsible to set the parameters for significance of the identified patterns
    - **Patterns identification:** Identify the candidate frequent and regular patterns and prune then according to the user defined parameters

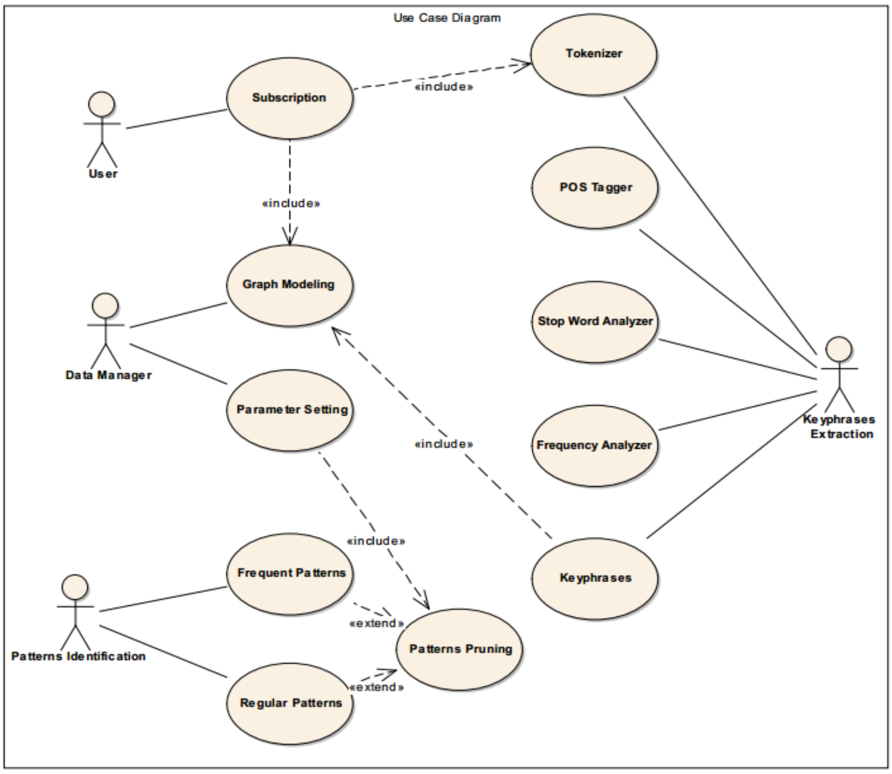


Figure 231 Overall system use case diagram

- **Brief Descriptions of Use Cases**
  - **Subscription**
    Responsible for register user with the application. Application send request to user and user verify application to access his emails.
  - **Tokenization**
    Tokenization converts sentences into separate word to lookup on each word individually.
  - **POS tagger**
    It is the process of marking up a word in a text as corresponding to a particular part of speech, i.e. relationship with adjacent and related words in a phrase or sentence as nouns, verbs, adjectives, adverbs, etc.
  - **Stop Word Analyzer**
    This use case process email contents and search for stops words after tagging into different parts. It removes the stop words from the contents of interaction.
  - **Frequency Analyzer**
    To identify the repetition of a particular word in the text, this use case calculates frequency of words which are semantically similar.
  - **Keyphrases**
    This use case collects the final keyphrases extracted at the end of NLP processing.
  - **Graph Modeling**

Role of graph modeling is to model the graph from the user interaction on the basis of time and attach the relevant extracted keyphrases on each node which gives the semantic of interaction at a particular time.

- **Parameter Setting**
  Aim of this use case is to set the threshold parameters for the identification of regular and frequent patterns.
- **Frequent Patterns**
  This use case is responsible for mine the frequent patterns from the graph model of interactions.
- **Regular Patterns**
  This use case is responsible for mine the regular patterns from the graph model of interactions.
- **Patterns Pruning**
  It identifies the patterns of interest from the set of frequent and regular patterns after looking into the parameter settings of threshold.

▪ **Sequence Diagram**

- The objective of Interaction model of a system is to depict the process scenario of how different objects interact with each other. Life span and sequencing of objects are the prime components of any interaction diagram. Sequence diagrams are described in following section.

- **Keyphrases Extraction (Figure 232)**
  - Extraction of the semantic keyphrases is the essential requirement of the accurate data modeling with the user interaction. First email contents are tokenized by using POS tagger and stop words analyzer. The frequency of each word is counted in the email and then relevant keyphrases are returned.

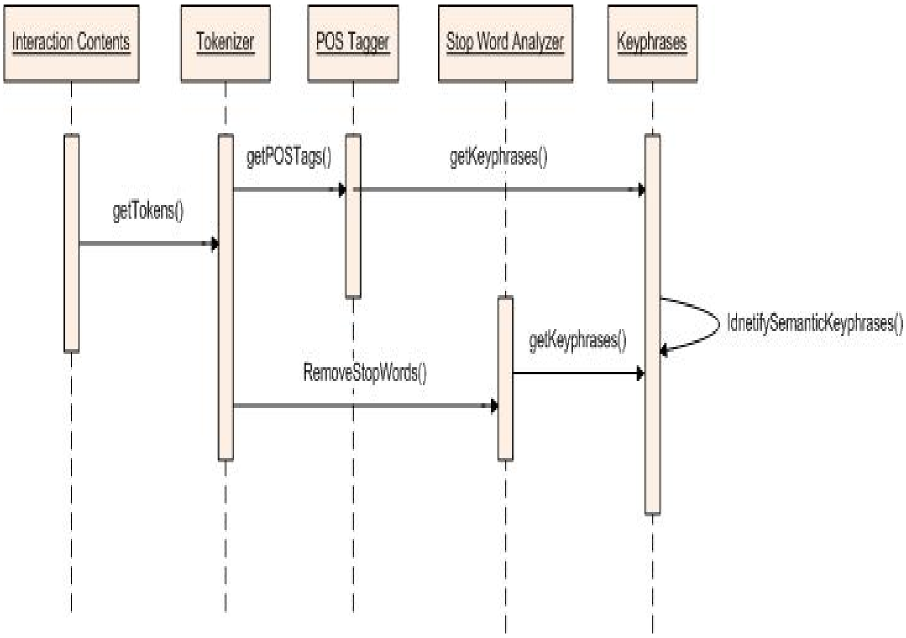


Figure 232 Sequence Diagram of Keyphrases Extraction

- **Data Manager (Figure 233)**
  - This module helps in data modeling and parameter settings before applying the mining algorithm. It extracts a population of interest from messy email interaction data by removing noise. The extracted information is modeled in graphs based on user defined interactions intervals and extracted keyphrases. In each graph nodes are the individuals with keypharses as node label and directed edge represents the interaction between them. Parameters set the thresholds of frequency and periodicity to identify the patterns of interest. For that, it is necessary to define a demanded minimum level (minimum confidence), so that all those sets of actions that have higher confidence level than the minimum confidence are considered as basic frequent periodic patterns.

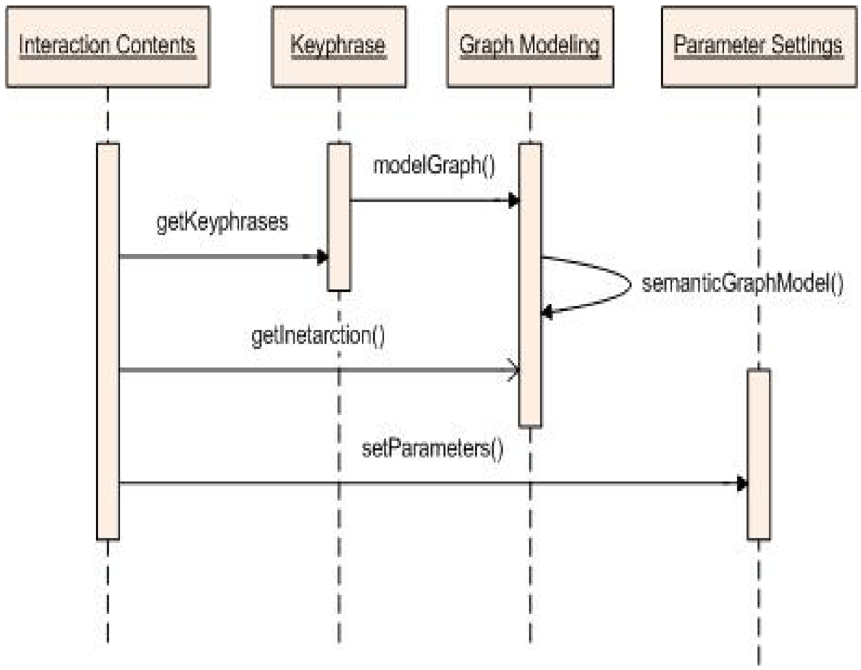


Figure 233 Sequence Diagram of Data Manager

- **Patterns Identification (Figure 234)**
  - This module identifies a set of frequent and periodic patterns from email interaction graphs. Frequent patterns emphasize the significance of patterns and periodic patterns consider the regularity between them. The objective is to identify the sets of actions that frequently and periodically occur together. Once basic frequent periodic patterns have been discovered, an aspect to consider is if there is any action which is not frequent taking into accounts all the periods, so that it is not discovered in the previous task, but it is frequent enough in those periods where the basic frequent periodic patterns occur. Patterns pruning reflects the common characteristics of a typical email interaction with some unusual association between patterns. For that, the starting point will be the candidate patterns are transformed into integrated set in order to make it useful comprehensively. Combining these two concepts allows us to define periodic patterns in a way that avoids any redundant information

.

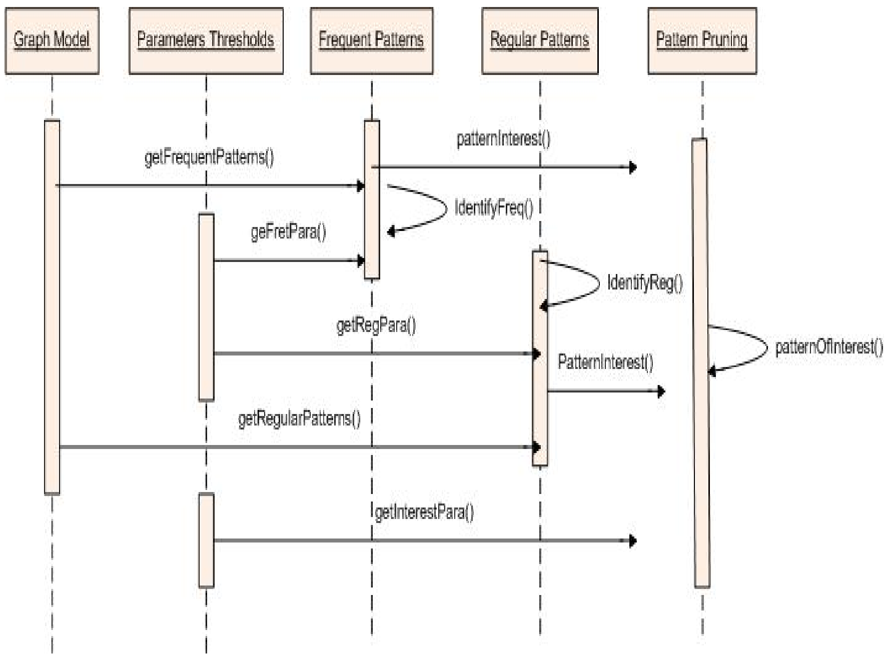


Figure 234 Sequence Diagram of Patterns Identification

- **Class Diagram**
  - The purpose of class diagram is to show the relationships among the classes of the system as shown in Figure 235. Additionally it covers the properties and operations that are visible at a higher level. Class diagram provides the high level overview of the system of how to be transformed into a detailed object model which ultimately helps to implement the system. Following section depicts the class diagram for interaction analysis for smart LDSS.
  - Task assigned to the class keyphrases is to extract keyphrases from the contents of interaction after applying the NLP techniques like POS tagging and stop words analyzer. GraphModel maps the interaction and semantic keyphrases into the graphical format.

FrequentPatterns and PeriodicPattens identify the candidate patterns while PatternsPruning identifies the patterns of interest after taking into account the parameter thresholds.

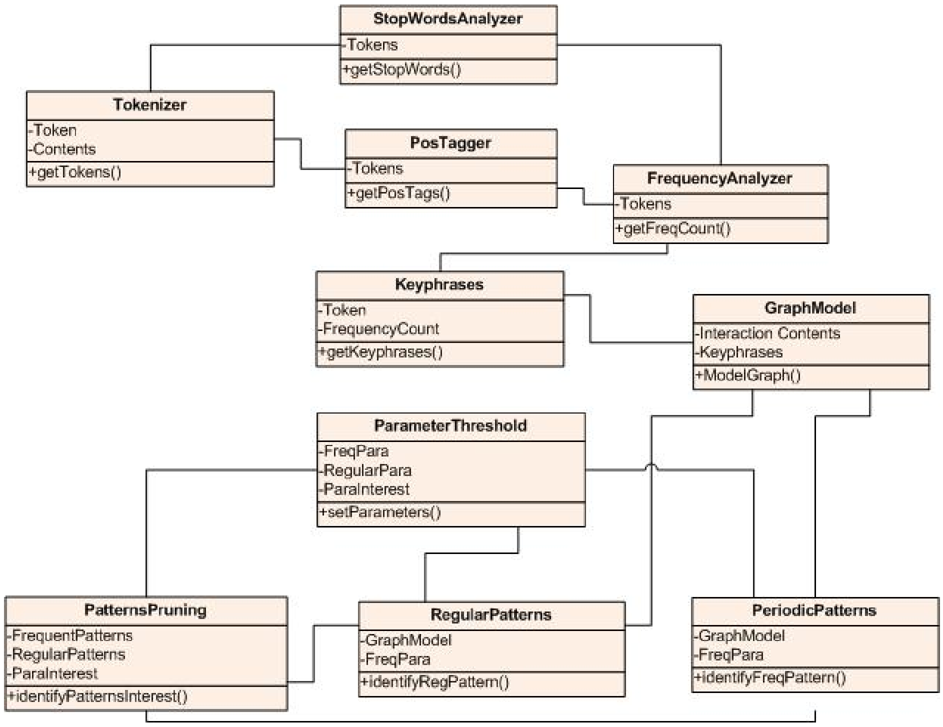


Figure 235 Class Diagram for Interaction Analysis

- **Component Diagram**

  - The component diagram of interaction analysis (Figure 236) shows different components and their relationships with each other. Also it shows the subcomponents interaction, purpose, and their relationships with each other. Mainly there are four main components and are explained below.

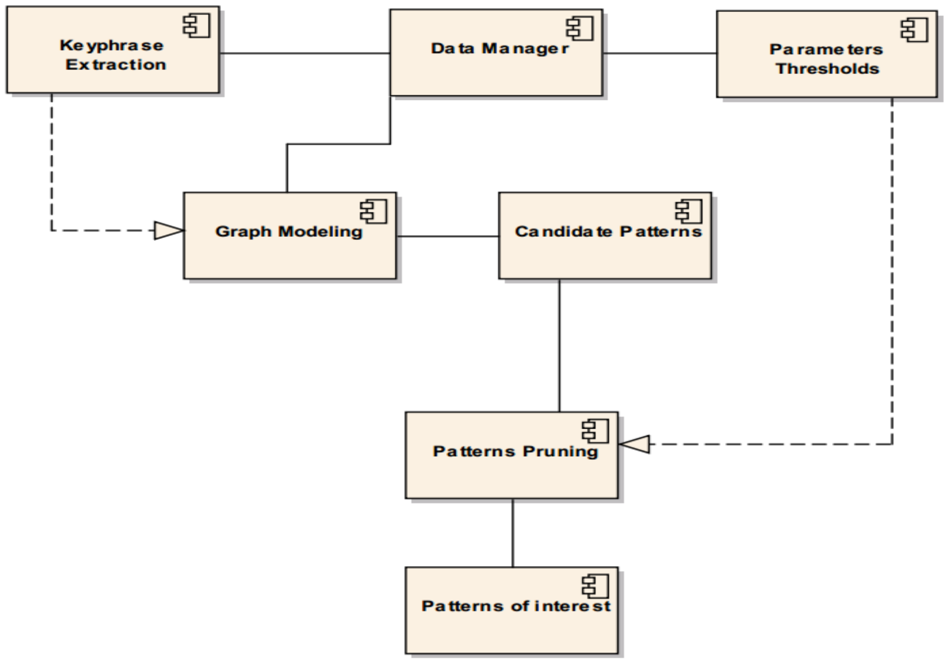


Figure 236 Component Diagram for Interaction Analysis

- **Keyphrases Extraction**
  - Extraction of the semantic keyphrases is the essential requirement of the accurate data modeling with the user interaction. The extracted keyphrases are used by graph modeling and data manager components for semantic annotation of interaction.

- **Data Manager**
  - This module helps in data modeling and parameter settings before applying the mining algorithm. It extracts a population of interest from messy email interaction data by removing noise. The extracted information is modeled in graphs based on user defined interactions intervals and extracted keyphrases. In each graph nodes are the individuals with keyphrases as node label and directed edge represents the interaction between them. Parameters set the thresholds of frequency and periodicity to identify the patterns of interest. For that, it is necessary to define a demanded minimum level (minimum confidence), so that all those sets of actions that have higher confidence level than the minimum confidence are considered as basic frequent periodic patterns.

- **Parameter Threshold**
  - This module manages the user defined threshold for graph modeling and for patterns in interest. Pattern pruning module used this information while processing the candidate patterns into patterns of interested.

- **Candidate Patterns**

- This module identifies a set of frequent and periodic patterns from email interaction graphs. Frequent patterns emphasize the significance of patterns and periodic patterns consider the regularity between them. The objective is to identify the sets of actions that frequently and periodically occur together. Once basic frequent periodic patterns have been discovered, an aspect to consider is if there is any action which is not frequent taking into accounts all the periods, so that it is not discovered in the previous task, but it is frequent enough in those periods where the basic frequent periodic patterns occur.

- **Pattern Pruning**
  - This module applies one mining process to identify frequent and periodic patterns under the given parameter settings. Patterns pruning reflects the common characteristics of a typical email interaction with some unusual association between patterns. For that, the starting point will be the candidate patterns are transformed into integrated set in order to make it useful comprehensively. Briefly explained, it infers meaningful actions from the data collected by email data and then it splits the string of actions into periodic sequences based on some frequent support. Combining these two concepts allows us to define periodic patterns in a way that avoids any redundant information

- **Patterns of interest**
  - The patterns of interest after pattern pruning are converted into vMR format that is passed to the CDSS system through adapter social network as shown in Figure 237.

```
<vmrInput>
    <templateId root="2.16.840.1.113883.3.795.11.1.1"/>
    <patient>
        <clinicalStatements><!--Health problem>
            <problems>
            <problem> <!--Muscle Fatigue-->
                <id root="107c2dc0-67a5-11db-bd13-0800200c9a66"/>
                <problemCode codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED
            CT" code = "56946009" />
            </problem>
            <problem> <!-- Depression -->
                <id root="10dca459-3249-29db-bd31-0800200c9a66"/>
                <problemCode codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED
            CT" code = " 291746001 "/>
            </problem>
        </problems>
        </clinicalStatement>
    </patient>
</vmrInput>
```

Figure 237 VMR for Interaction Analysis

6.2.4. Behavior Modeling Module

- Like LDSS for Wellness, BMM is also one of the core components of LDSS for Chronic Disease.
- It takes input form HAR, CAME, and Social Media and log them in life Log.
- Later it uses the Life Log information for service provisioning.
- The Life Log information is used for analyzing user behavior for shorter time interval and for longer time intervals which can be even years.
- On the other hand, BMM also provide the facility for behavior prediction in case the current behavior is matching with some existing (stored, learned) behaviors.
- The component based framework architecture diagram of BMM is given in main architecture of LDSS for Chronic Disease.

- **Use Case Diagram**
  - The use case diagram of BMM is shown in Figure 238, whereas its description is given below.
  - **Actors:**
    - **User:** User is the main user of the system that is using BMM for behavior analysis and prediction which is based on the data user has performed and interacted.
    - **InputSources:** Input sources are the entities which are producing output which is considered as input for BMM.

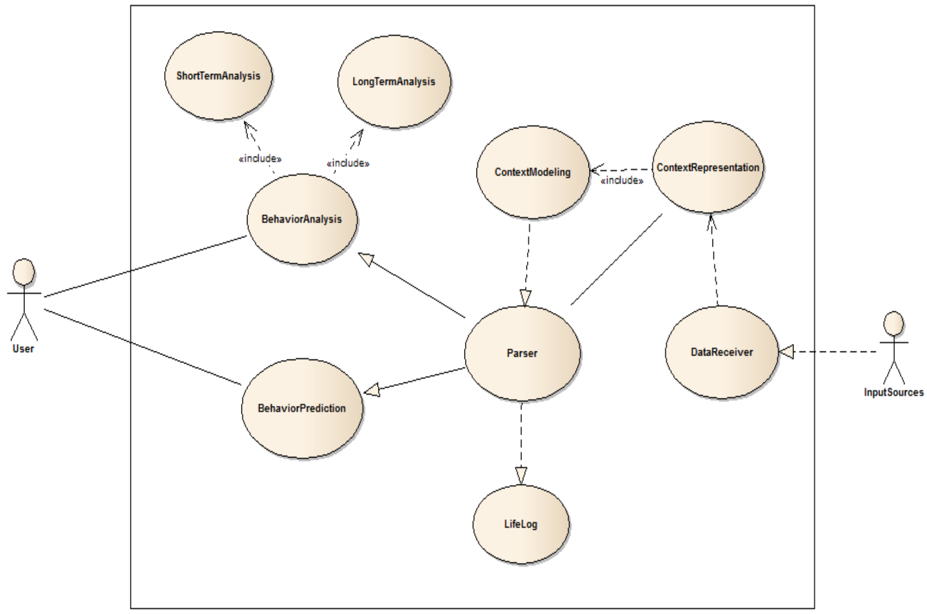


Figure 238 Use case diagram of BMM

  - **Use case Description**
    - **DataReceiver:** This use case receives the incoming data from HAR, CAME, and Social Media in their respective formats.

- **ContextRepresentation:** This use case is responsible for representing the incoming information from DataReceiver in ontological structure defined in Life Log while taking help from Parser.
- **ContextModeling:** This use case is used to verify the represented information against the Life Log for its consistency and existence.
- **Parser:** The responsibility of this use case is to parse the Life Log for the purpose of Life Logging as well as for assessing Life Log data to make the Behavior Analysis and Prediction.
- **BehaviorAnalysis:** This use case uses different data mining techniques for analyzing user short term behavior analysis and long term behavior analysis.
- **BehaviorPrediction:** This use case predicts user behavior based on user current interactions and already stored/learned behaviors.

▪ **Sequence Diagram**

- Sequence diagrams for BMM are described below that shows the interaction between different objects of the system for the achievement of defined objectives.

- **Context Modeling and Logging (Figure 239)**
  - When context is received in BMM then for its usage for behavior analysis and prediction and storage in Life Log needs to be in Life Log representational structure.
  - It calls the Controller using initiate(context) and start modeling the context received.
  - It gets the representational structure from Life Log by calling getStructure() to Parser which retrieve the structure from Life Log.
  - The represented context information is checked for consistency verification and if verified then is logged in Life Log using ContextModeling by calling logContext().

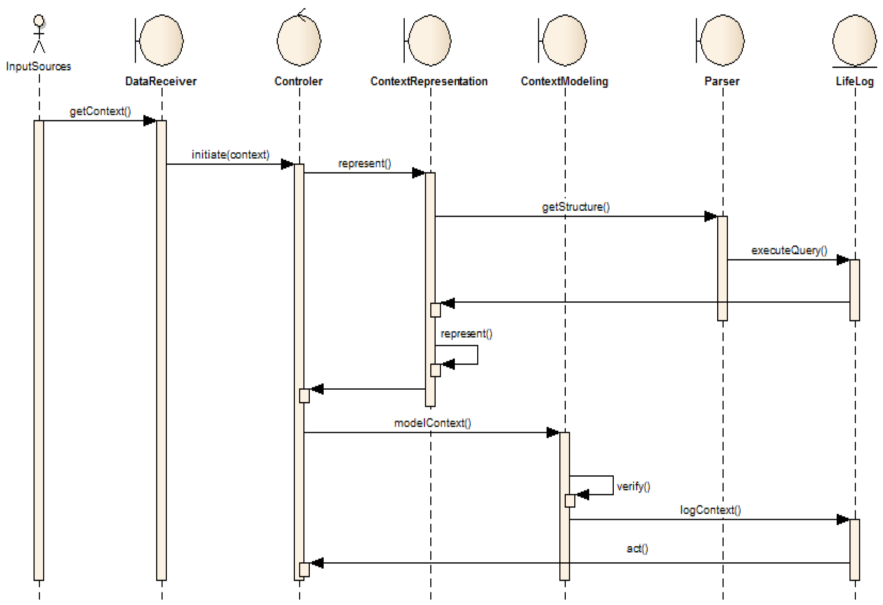


Figure 239 Sequence diagram of BMM Context Modeling and Logging

- **Behavior Analysis and Prediction (Figure 240)**
  - The Controller is activated for the purpose to initiate the behavior analysis and prediction
  - The logged data is extracted from LifeLog using executeQuery().
  - The BehaviorAnalysis and BehaviorPrediction are activated by Controller using initiateBehaviorAnalysis() and initiateBehaviorPrediction() respectively.
  - The results of BehaviorAnalysis and BehaviorPrediction are displayed on DisplayResult.

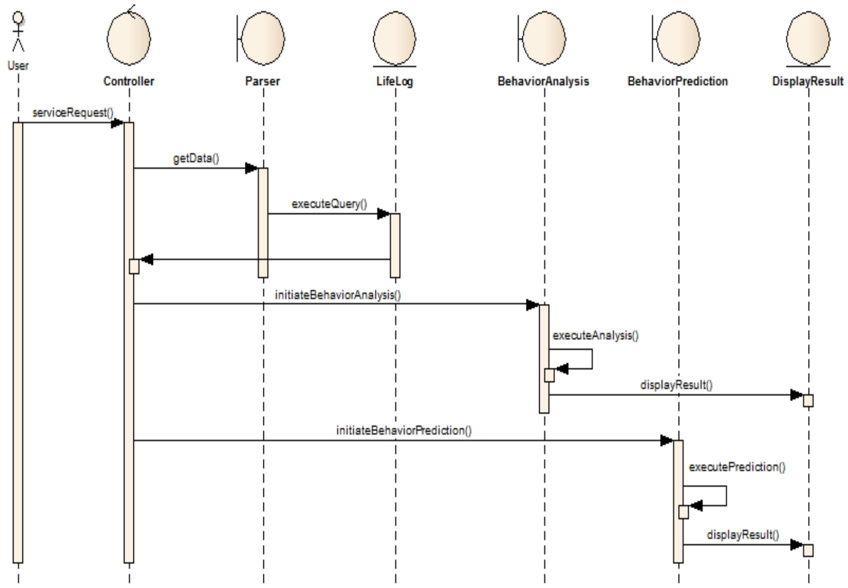


Figure 240 Sequence diagram of BMM Behavior Analysis and Prediction

- **Class Diagram**
  - Class diagram of BMM (Figure 241) shows the different classes and their relationships with each other.
  - Initially the DataReceiver and ContextRepresentation are loaded which work together for incoming context information representation.
  - Then the context information is verified in ContextModeling Class. The verified context is then logged using Parser Class in the Life Log.
  - DataExtractor extract the required data form the Life Log for user request service and provide the data to that particular service.
  - BehaviorAnalysis and/or BehaviorPrediction service is activated based on user request and requested services are returned to user.

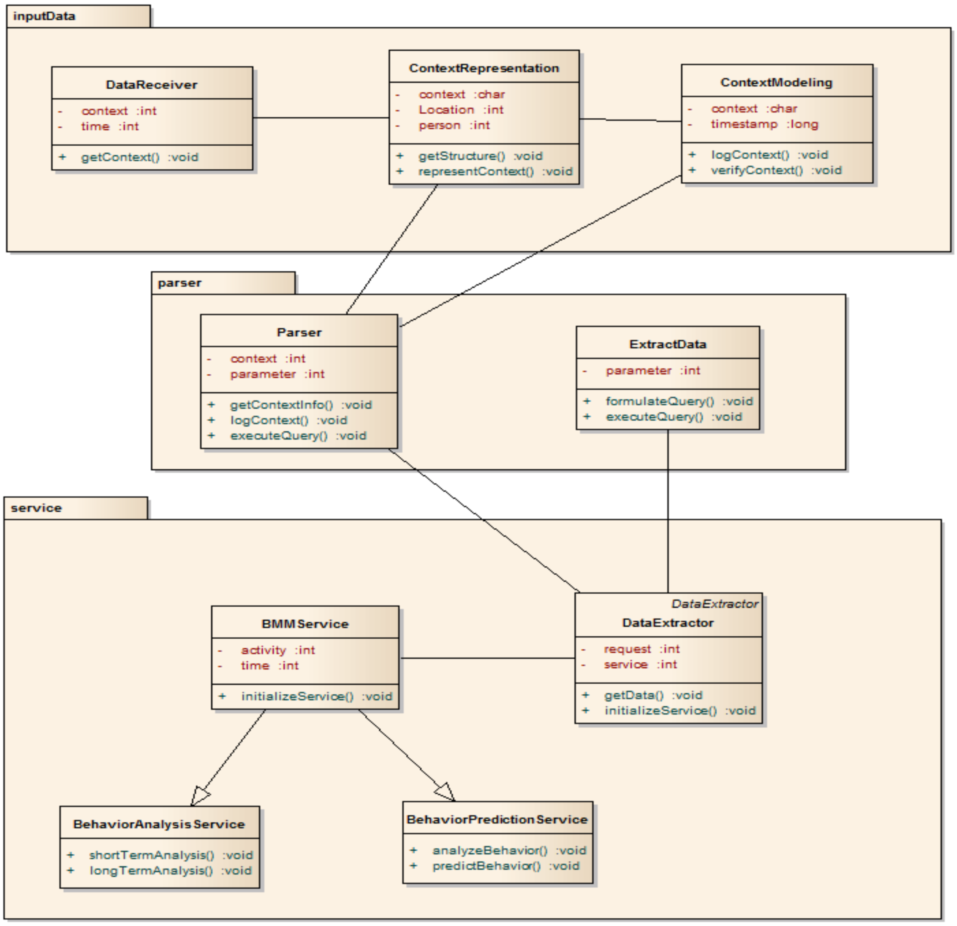


Figure 241 Class diagram of BMM

- **Component Diagram**
  - The component diagram (Figure 242) of BMM shows different components and their relationships with each other. Also it shows the subcomponents interaction, purpose, and their relationships with each other. Mainly there are five main components and are explained below.

  - **Context representation and Modeling Component**
  - It is responsible for representing the incoming context received from DataReceiver in the Life Log structure and then verify the context for its consistency.

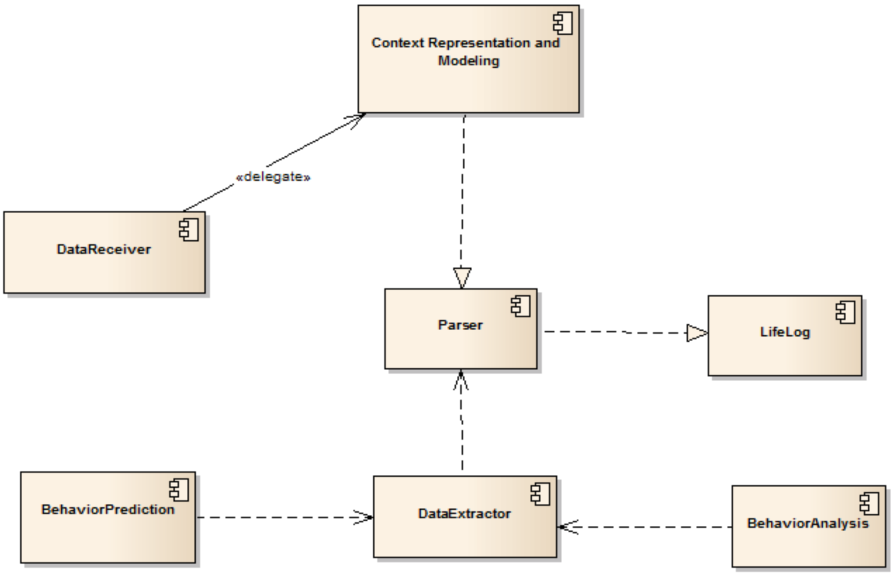


Figure 242 Component diagram of BMM

- **Parser Component**
- It is responsible for BMM components interaction with the Life Log for context logging and extraction. It is the main component that is always connected with the Life Log and fulfills the jobs coming for Life Log access.

- **Behavior Analysis Component**
- It is responsible for sending appropriate request for extracting data from Life Log based on user request. Then this data is used for User Behavior Analysis.

- **Behavior Prediction Component**
- This component is responsible to provide appropriate Behavior Prediction for user request based on the data extracted from Life Log.

6.2.5. Adapter Interoperability Engine

- **Use Case Diagram**
  - LDSS-Chronic Disease takes clinical data from HMIS for processing using Adapter Interoperability Engine (ARIEN) component. Heterogeneities among HMIS compliant standard and LDSS-Chronic Disease compatible standard exists. ARIEN resolves these heterogeneities for HMIS's to interact easily with LDSS-Chronic Disease to utilize its services. It behaves as mediator between the two systems. HMISs compliant to different healthcare standards understand only the standardized format such as: HL7 CDA, openEHR, CEN 13606, while LDSS-Chronic Disease can only process virtual Medical Record (vMR) format. ARIEN provides bridge services that use ontology matching techniques to generate mappings between heterogeneous healthcare standards for automatic

transformation of information to enable interoperable communication among healthcare systems.

- **Actors**
    - **Software Engineer:** Software Engineer is responsible for generating the ontology mappings using different ontologies and storing the generated mappings in a repository.
    - **Physician:** Physician interacts with the system by entering the required data that is converted to HMIS compliant standard format.
    - **HMIS:** HMIS is responsible for converting the physician entered information to standard format.

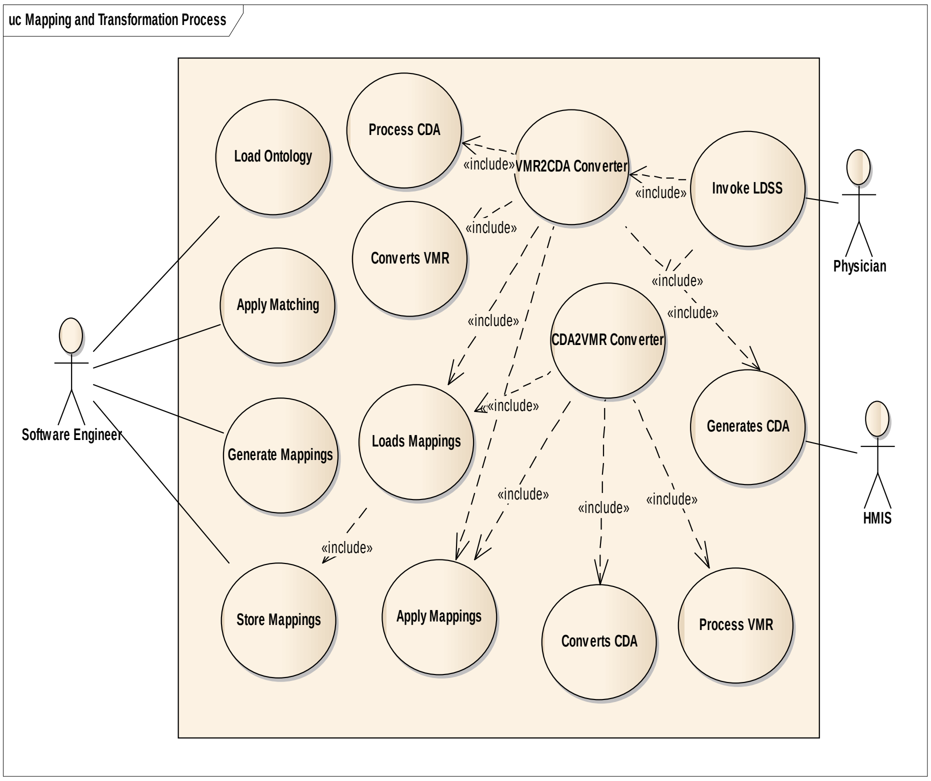


Figure 243 Interoperability Engine Use Case Diagram

- **Sequence Diagram**
    - **Generate Ontology Mappings**
        - User browsers the source and target ontologies using loadOntologies(source,target) method. The ontologies are loaded by ontologyLoader() method by OntologyLoader object. User selects the matcher using selectMatcher() method for the performMatching() method to be performed by OntologyMatching object. Finally the user store the mappings using saveMappings() method.

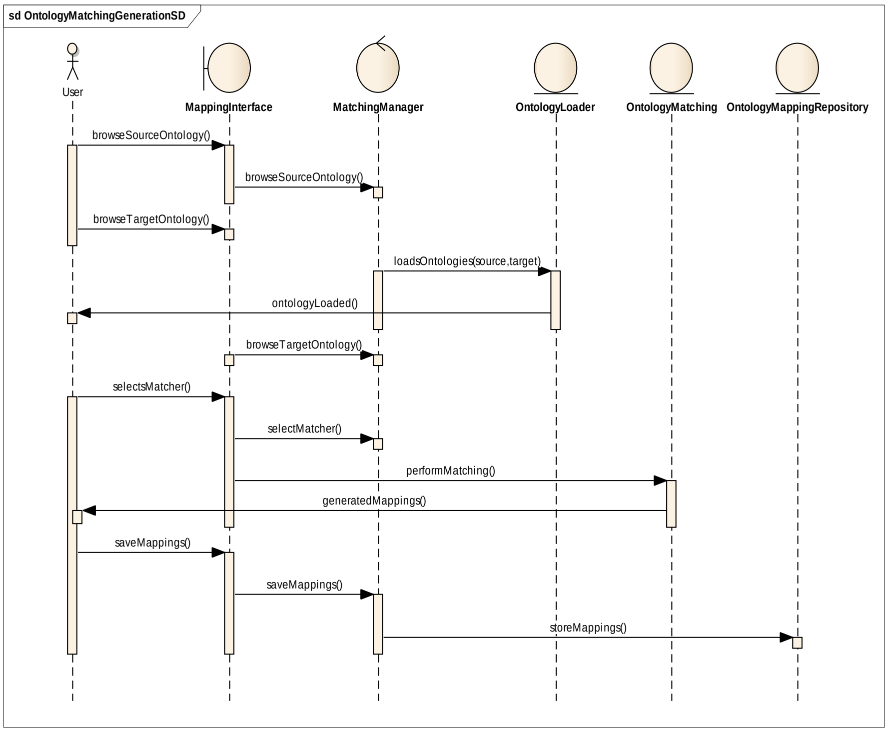


Figure 244 Generate Ontology Mappings Sequence Diagram

- **Transform Standard Format**
  - Physician invokes the LDSS and provides information in CDA format. CommunicationController object access the CDA format and calls convertCDA2VMR() method. The mappings are loaded using loadMappings() method by the CDA2VMRConverter object to apply mappings and generate the corresponding VMR using generateVMR() mappings. Finally the generated VMR is accessed by the InterfaceEngine object for final concatenated VMR creation. The concatenation process is responsibility of the Fusion Adapter module.

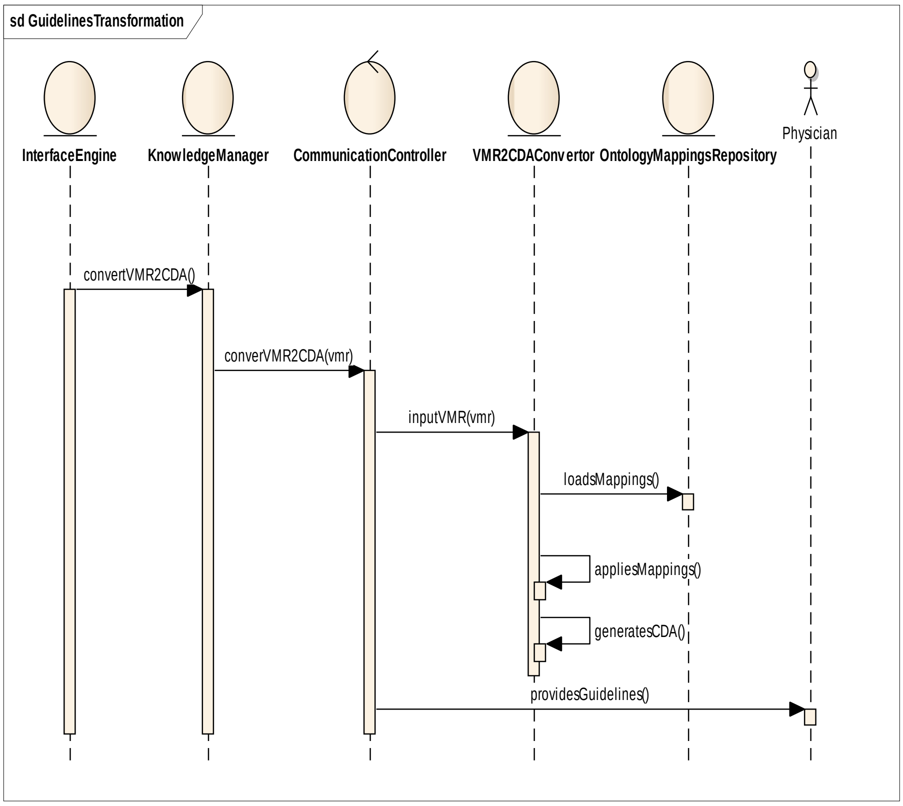


Figure 245 Transform Standard Format Sequence Diagram

- **Transform Guidelines**
  - The Knowledge Manager module generates the guidelines and provides to the InterfaceEngine in VMR format. InterfaceEngine forwards the generated guidelines in VMR format to CommunicationHandler object that provides it as input to VMR2CDAConverter object. VMR2CDAConverter object loads the mappings using loadMappings() method and further uses appliesMappings() and generatedCDA() functions to convert the VMR format to CDA format. Finally the generated CDA is provided to HMIS compliant to CDA that shows the guidelines to physician in a user friendly manner.

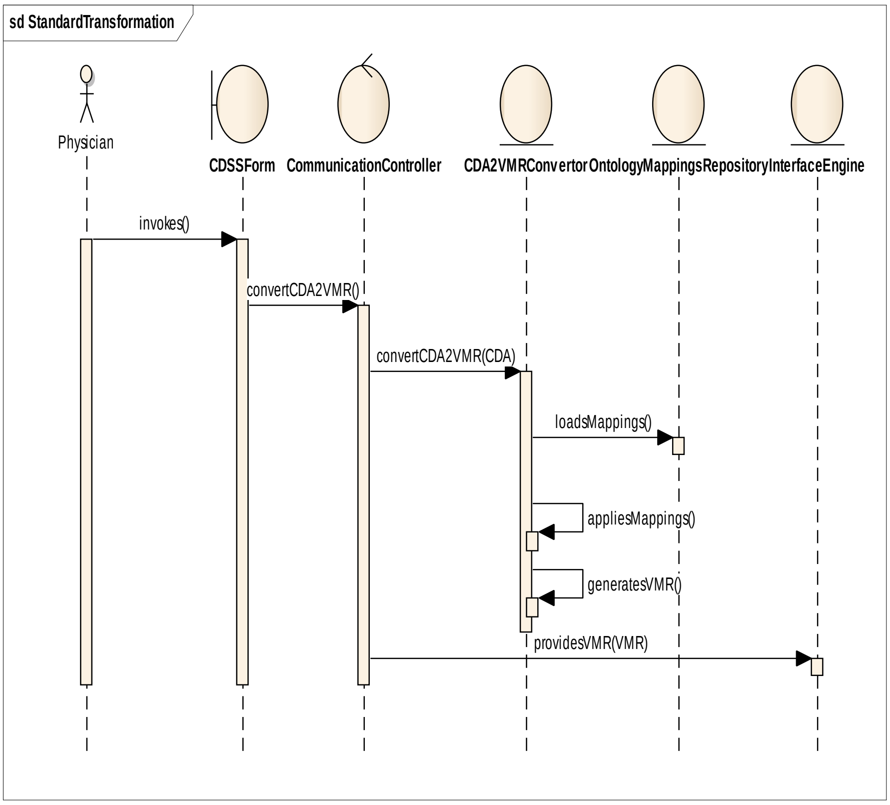


Figure 246 Transform Guidelines Sequence Diagram

- **Class Diagram**
  - Class diagram shows the different classes and their relationships with each other. The class diagram for ARIEN system shows classes and their dependencies with each other. Initially the OntologyLoader class loads the source and target ontologies. These ontologies are used for mappings generation among different standards and therefore are passed to OntologyMatcher class. OntologyMatcher class performs ontology matching techniques to generate mappings. These generated mappings are passed to OntologyMappingRepository class for storage purposes. The mappings are stored by OntologyMappingRepository class and it can edit the mapping by performing delete, store and edit functions. These mappings are then used for transformation of one standard to another. CommunicationHandler class access the CDA format from HMIS and provides this information to CDA2VMRConverter class. CDA2VMRConverter class is responsible for using the mappings stored to convert CDA format to VMR. In the same way VMR2CDAConverter performs the opposite function by converting VMR format to CDA format.

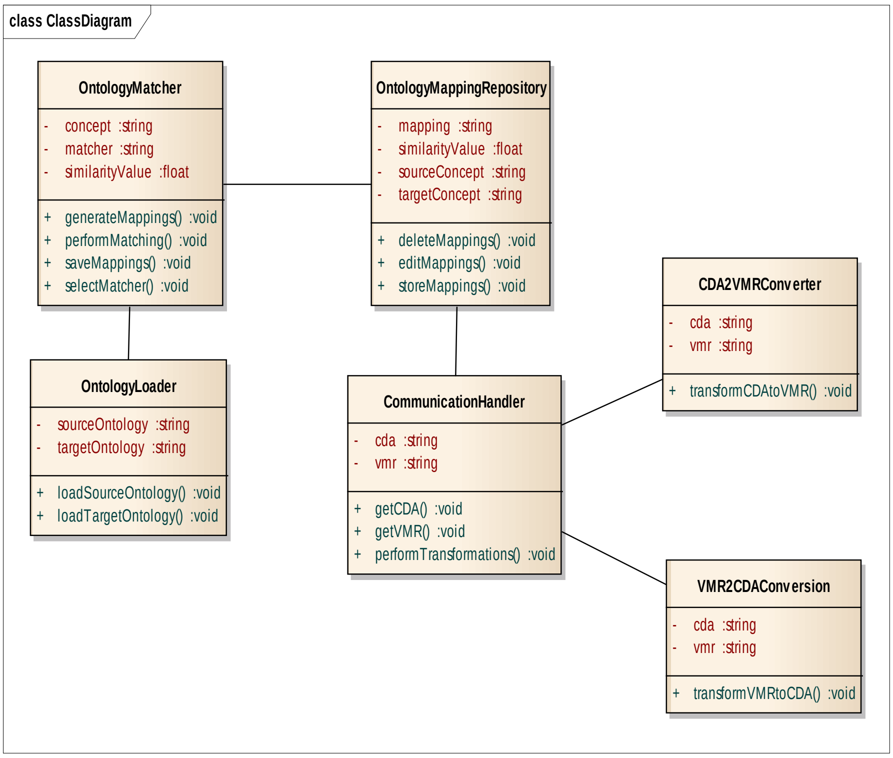


Figure 247 Interoperability Engine Class Diagram

- **Component Diagram**
  - The component diagram of ARIEN shows the different components and their relationships with each other. Also it shows the subcomponents of the main components and their relationships with each other. Mainly three components are included in the ARIEN module. These are explained as follows:
    - **Accuracy Mapping Engine Component**
      It is the main component of Adapter Interoperability Engine that is used for generating the mappings between different healthcare standards and then storing them. It is composed of subcomponents like OntologyLoader, OntologyMatcher and MappingVerifier.

    - **Mapping Repository Component**
      Another main component is the MappingRepository that is used to store the mappings in the form of Bridge ontology. This component also consists of two subcomponents MappingAccess and MappingStorage.

    - **Integration Module Component**
      IntegrationModule is another main component that is used to use the mappings generated by AccuarcyMappingEngine and stored in MappingRepository. These mappings are used for transformation purpose between different standard formats. It includes subcomponents like CommunicationHandler and ConversionEngine.

- **OntologyLoader Subcomponent**
  OntologyLoader is the subcomponent of AccuracyMappingEngine component and is used for loading the source and target ontologies for mappings generation.

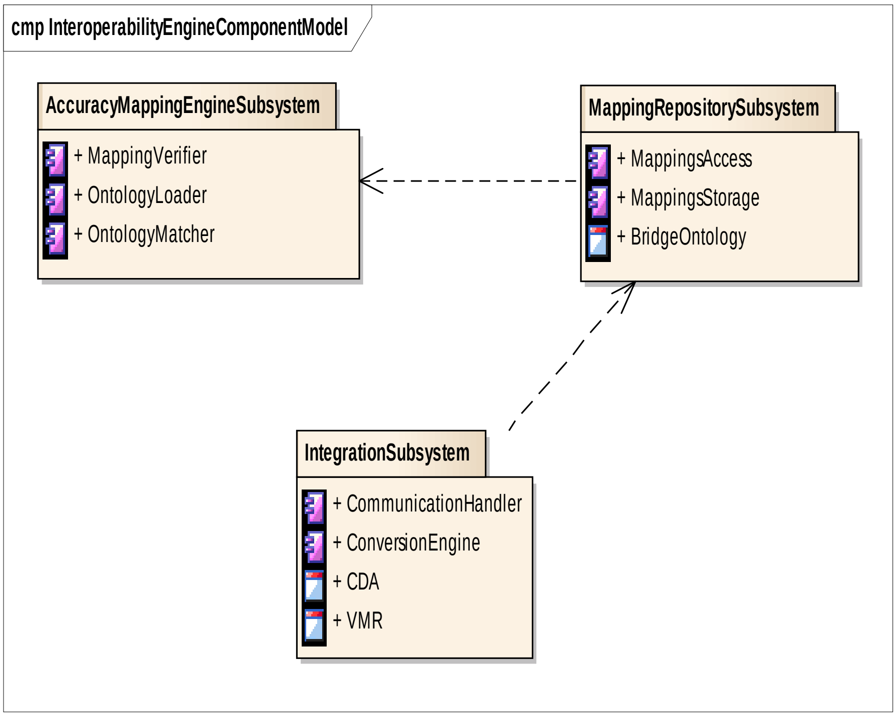


Figure 248 Interoperability Engine Overall Component Diagram

- **OntologyMatcher Subcomponent**
  OntologyMatcher is the subcomponent of AccuracyMappingEngine component and is used for generating the mappings between different standards using various ontology matching techniques.

- **MappingVerifier Subcomponent**
  Mappings generated by OntologyMatcher subcomponents require verification that is carried out by the MappingVerifier subcomponent.

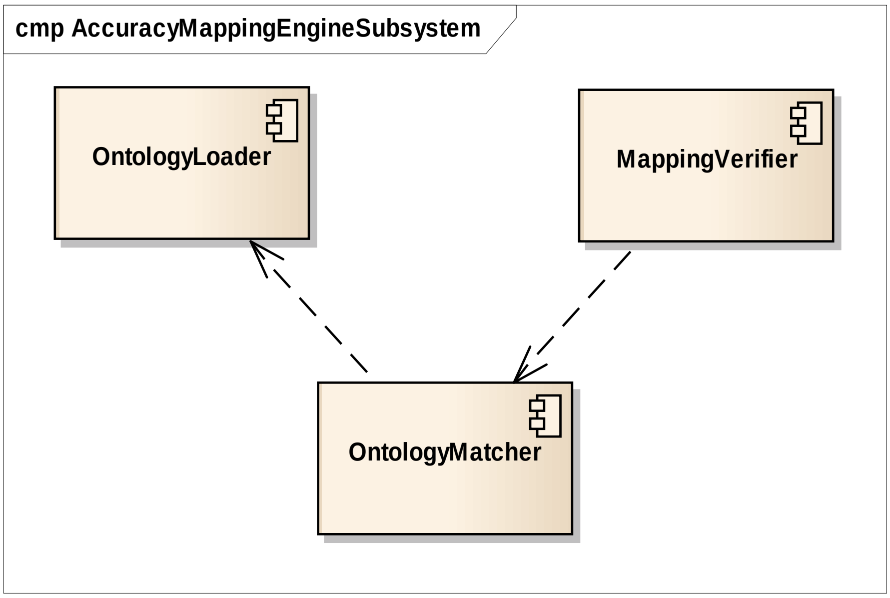


Figure 249 Accuracy Mapping Engine Subsystem Components

- **MappingStorage Subcomponent**
  MappingStorage is the subcomponent of MappingRepository component and is used for storing the mappings generated using ontology matching techniques. This component gives the mapping file a structure and stores it in RDF format.

- **MappingsAccess Subcomponent**
  MappingsAccess is the subcomponent of MappingRepository component and is used for providing the required mappings to the conversion engines for transformation from one standard format to another.

- **BridgeOntology Object**
  BridgeOntology object is created by the MappingStorage subcomponent and is used by the MappingAccess subcomponent for transformation purpose.

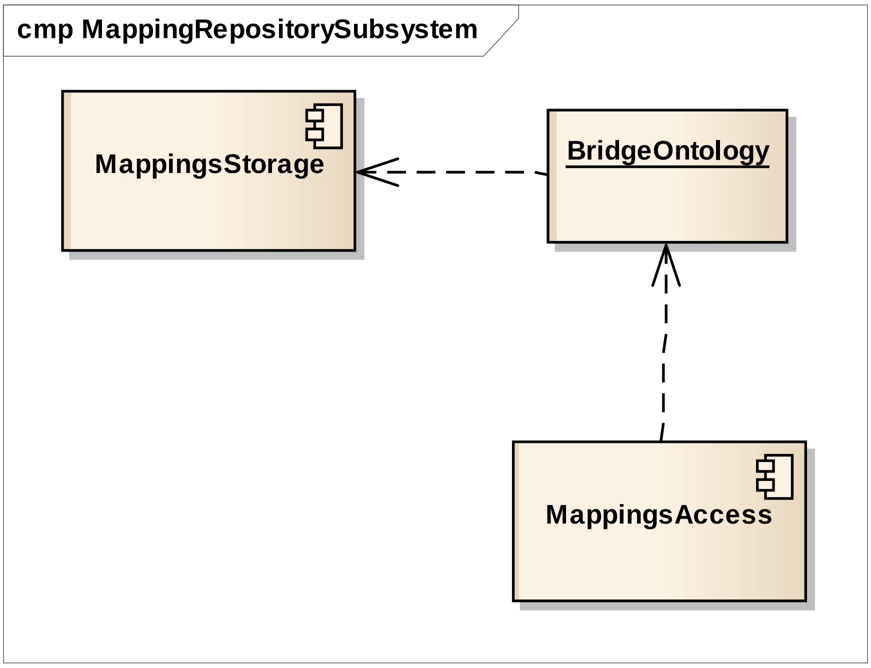


Figure 250 Mapping Repository Subsystem Components

- **CommunicationHandler Subcomponent**
  CommunicationHandler is the subcomponent of IntegrationModule component and its primary purpose if to obtain HMIS compliant standard format and provides it to ConversionEngine for applying mappings.

- **ConversionEngine Subcomponent**
  ConversionEngine is the subcomponent of IntegrationModule component and is used to transform standard format from HMIS compliant standard to LDSS compatible standard and vice versa.

- **CDA Object**
  CDA Object is the standard format that is required by the HMIS to understand. Initially the HMIS provides CDA instance to LDSS and later get guidelines in the same format.

- **VMR Object**
  VMR Object is the standard format that the LDSS understands for processing. Therefore before processing the CDA format is converted to VMR and the guidelines are also generated in the same format.

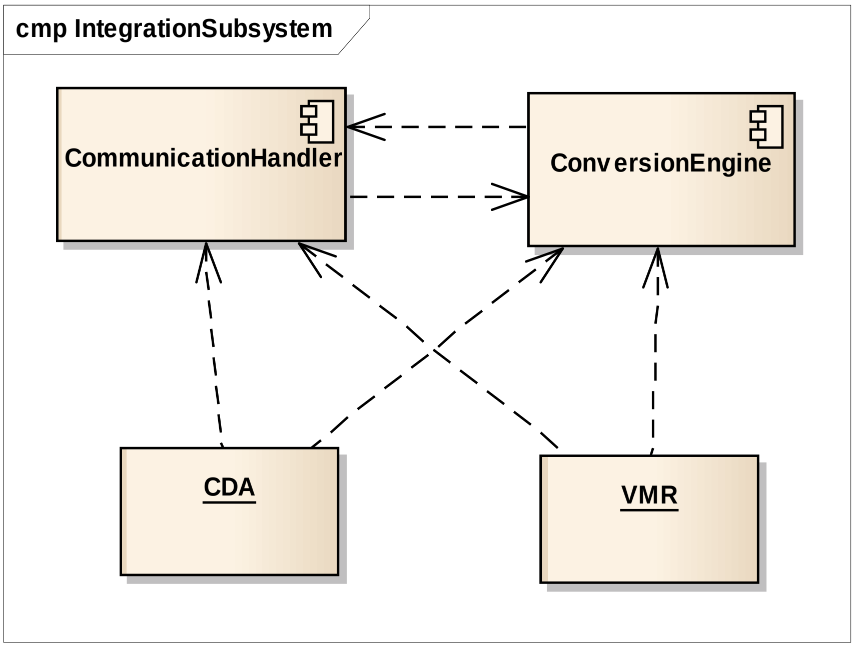


Figure 251 Integration Subsystem Components

6.3. SmartCDSS for Head & Neck Cancer

- **Use Case Diagram**
  - **Physician (Actor)**
    - Physician interacts with Smart CDSS system to get guidelines and recommendations during the course of HNC patient diagnosis and treatment.
  - **Consumer System (Actor)**
    - Smart CDSS is sometimes called by consumer system like EMR or HMIS automatically based on some event might occur.

  - **Use cases for Smart CDSS (Use Case Diagram)**
    - **Author Rule**
      Smart CDSS is able to generate guidelines based on the rules in its Knowledge Base. These rules are authored by expert physicians based on their knowledge and experience in the domain. AuthorRule use case is designed to capture the rule authoring activity. Note this use case is further elaborated and decomposed in several other use cases.
    - **Verify Rule**
      A knowledge rule once created requires dual verification from the expert physician in order to ensure the correctness and validity. VerifyRule is responsible to hold all those steps required for verification.
    - **Acquiring Clinical Data**

Smart CDSS works on the input data of a patient in order to generate the guidelines. This data comes from the health system where Smart CDSS is deployed. Physicians get the data for a particular patient and call to Smart CDSS service. Acquiring Clinical Data covers the function of data acquisition from within the workflows of hosting health system such as EMR and make a call to Smart CDSS.

- **Get Guidelines**
  The main objective of Smart CDSS is to produce guidelines for physicians to assist in clinical decisions. Get Guidelines use case generates guidelines from Smart CDSS for physician by performing two further activities; reasoning and rule execution.
- **Reasoning**
  The Reasoning use case reasons on the Knowledge Base to reach to a correct rule based on the input data sent by physicians.
- **Execute Rule**
  Based on the correct identification of rule by Reasoning use case, the ExecuteKB use case executes that rule and produces the resultant guidelines for the user.
- **Generate Alerts**
  Sometimes the system such as EMR or HMIS by itself called to Smart CDSS based on some event might occur.

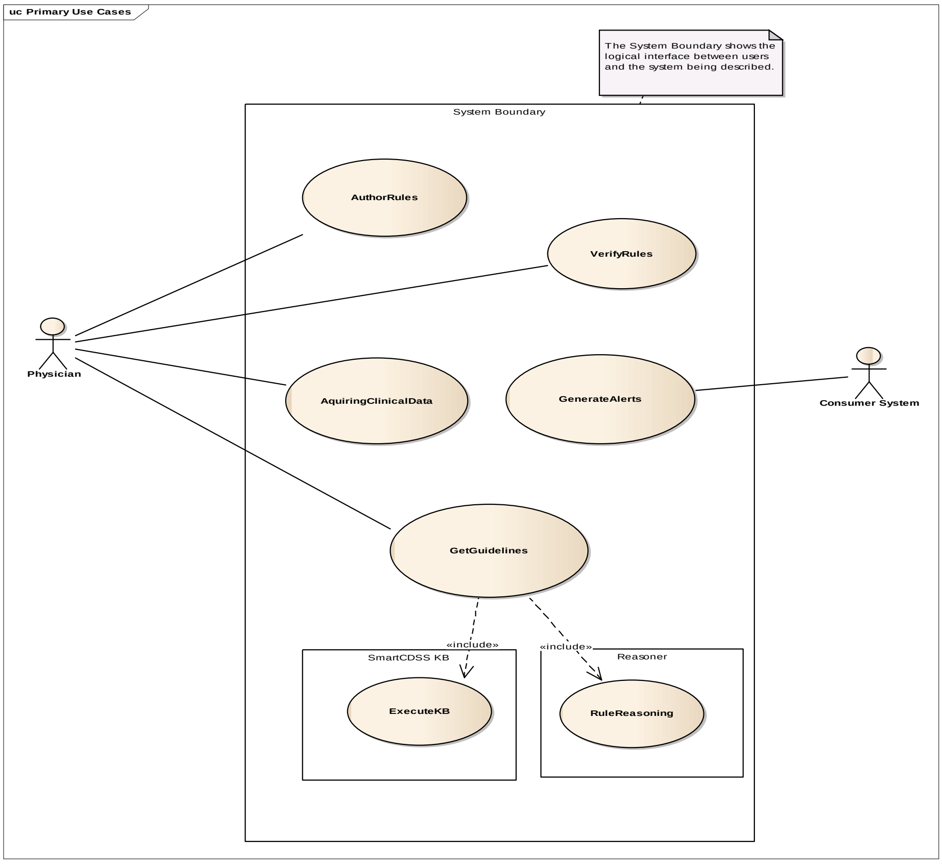


Figure 252 Smart CDSS Use Case Diagram

- **Use Cases for Smart CDSS Authoring Tool (Use Case Diagram)**
  - Authoring Tool is a sub component of smart CDSS to develop, manage and maintain the Knowledge Base of Smart CDSS. Smart CDSS assists the physician to provide guidelines and recommendations using the knowledge of Knowledge Base. So Authoring Tool provides the environment of expert physicians to share their knowledge with expert system Smart CDSS. The Authoring Tool component is further divided into four subcomponents like Rule Editor, MLM Validator, Compilation Module and Knowledge Base.

  - **Browse Clinical Rule Editor**
    To create rule the expert will browse the system rule editor. Rule editor will load the related ontology to the browser. This domain ontology will use in rule creation.
  - **Create Rule**
    In this use case the physician write the Rule on editor using the loaded ontology.
  - **Validate Rule**
    In this use case the physician validates his/her created rule with the standard format of MLM. If it is valid then pass store this MLM in MLM File Repository for sharing and reusable knowledge.
  - **Save Rule to Knowledge Base**
    After validation of rule, the physician saves the rule using rule editor. This use case includes three tasks to perform.
  - **Parse Rule**
    Before saving the rule to knowledge base the MLM is divided into small categories like Maintenance, Library and knowledge.
  - **Transform to executable**
    For the execution purpose the categories of MLM transformed to executable code here we have C# code.
  - **Compile Rule**
    After generation of executable classes, this use case will actually execute it and save this exe file into Knowledge Base for our decision making by smart CDSS.

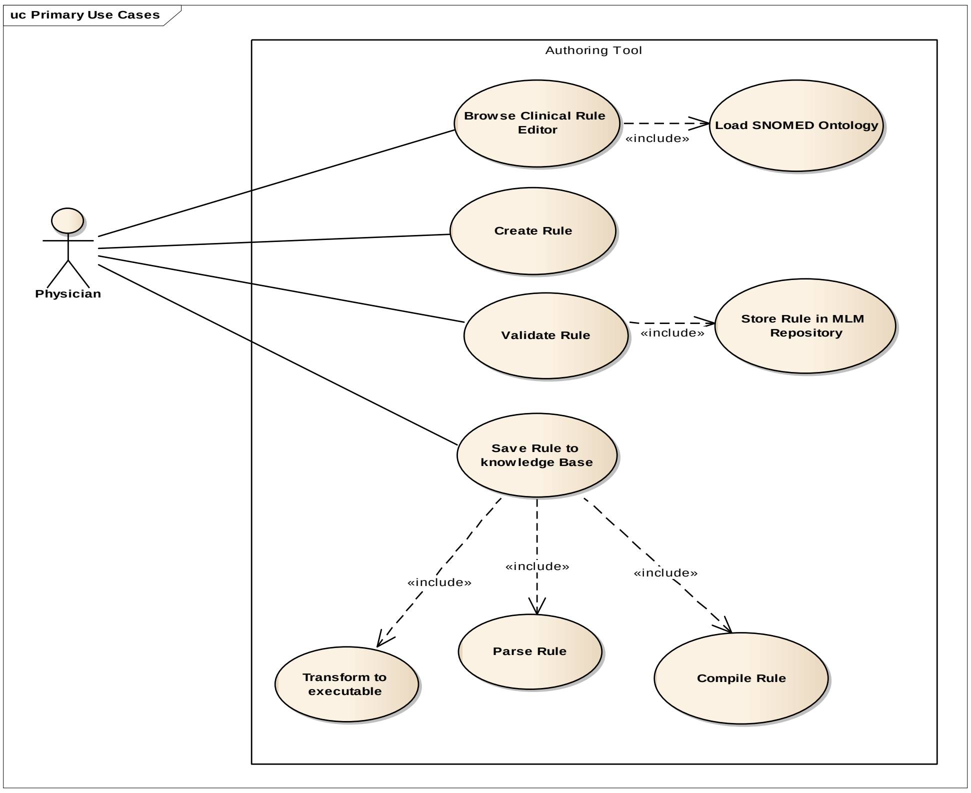


Figure 253 Authoring Tool Use case diagram

- **Sequence Diagram Smart CDSS Core Components and Authoring Tool**
  - The objective of Interaction model of a system is to depict the process scenario of how different objects interact with each other. Life span and sequencing of objects are the prime components of any interaction diagram. Sequence diagrams are described in following section.

  - **Acquiring Clinical Data**
    - Physician, a user of Consumer system such as EMR/HMIS, often requires support in clinical decision making process in the course of HNC patient diagnosis and treatment. The Physician calls to Smart CDSS from within the EMR/HMIS system to get the required guidelines. The main object involved in this process is SmartCDSSServiceInterface which called to SmartCDSSServiceController once the data has been acquired from consumer system.

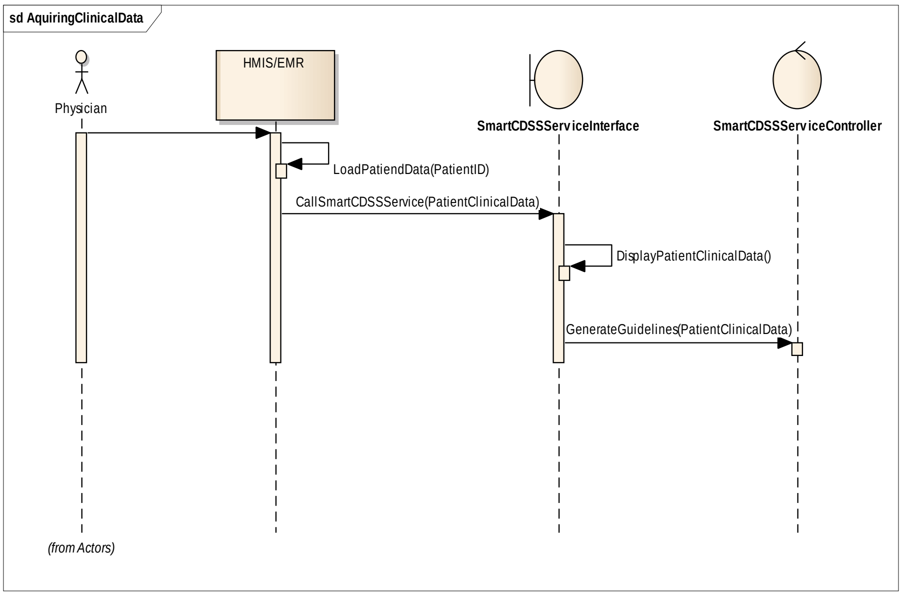


Figure 254 Sequence Diagram: Acquiring Clinical Data

- **Get Guidelines**
  - Based on the acquired data for a particular patient, physician needs assistance from Smart CDSS in order to reach to a final conclusion in diagnosis or treatment process. The patient data that is in HL7 CDA format is transformed to HL7 vMR standard in order to be consumed by Smart CDSS service. The Interoperability Adapter basically achieves this transformation and sends the vMR formatted message to Smart CDSS Service.

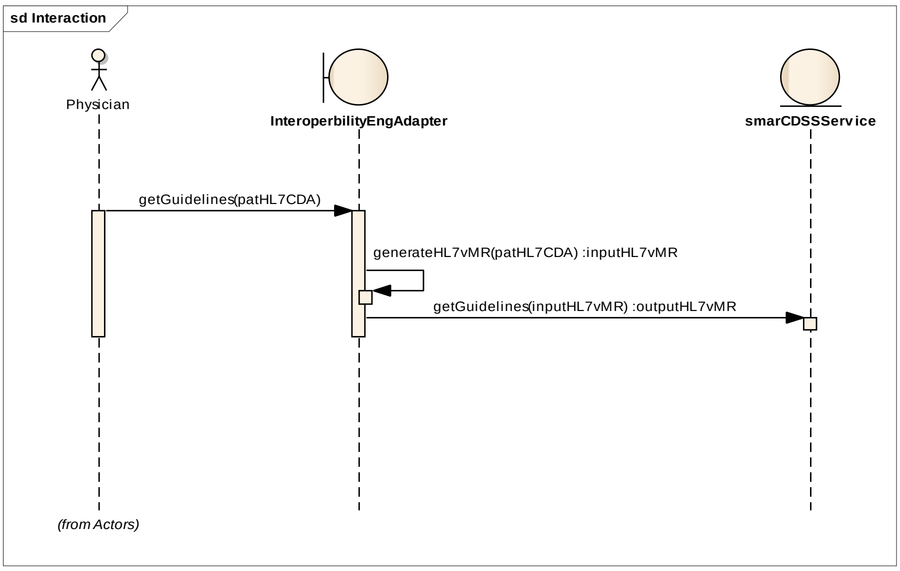


Figure 255 Sequence Diagram: Get Guidelines

- **Execute Knowledge Base**
  - Execution of the knowledge base occurs based on the reasoning done by Reasoner component in order to fire the appropriate rule in KB. There is a range of activities performed in this use case. Physicians from with the consumer application connect to Smart CDSS Interface object in order to get the final guidelines. The Smart CDSS Interface internally calls the MLMReasoner Object which starts reasoning on the MLMs in Knowledge base iteratively in order to find the correct MLM. Once the appropriate MLM is found, it is then executed. Based on the logic in fired MLM, final guidelines/recommendations are generated and returned to the user.

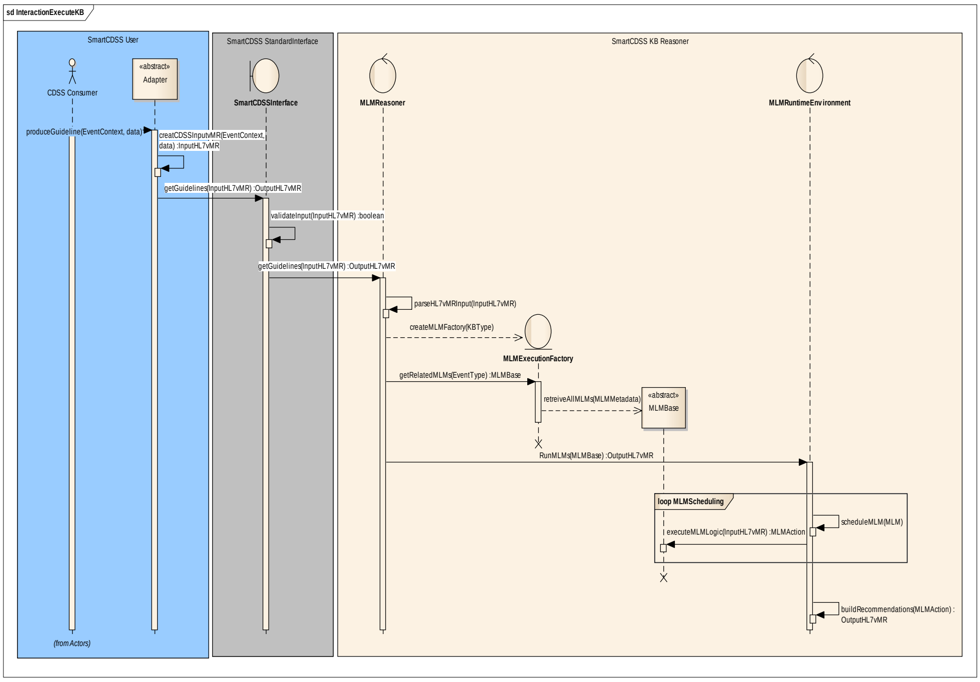


Figure 256 Sequence Diagram: Execute Knowledge Base

- **Create Clinical Rule**
  - When Physician wants to create new rule or update the exiting rule then he/she will open the Rule Editor by OpenRuleEditor(). On browsing the editor the SNOMED CT ontology will load automatically by LoadSNOMEDOntology(). All the related ontology will fetch to the user interface. Physician will create rule using the loaded ontology by CreateRule(). Following is the Sequence Diagram of Create Clinical Rule.

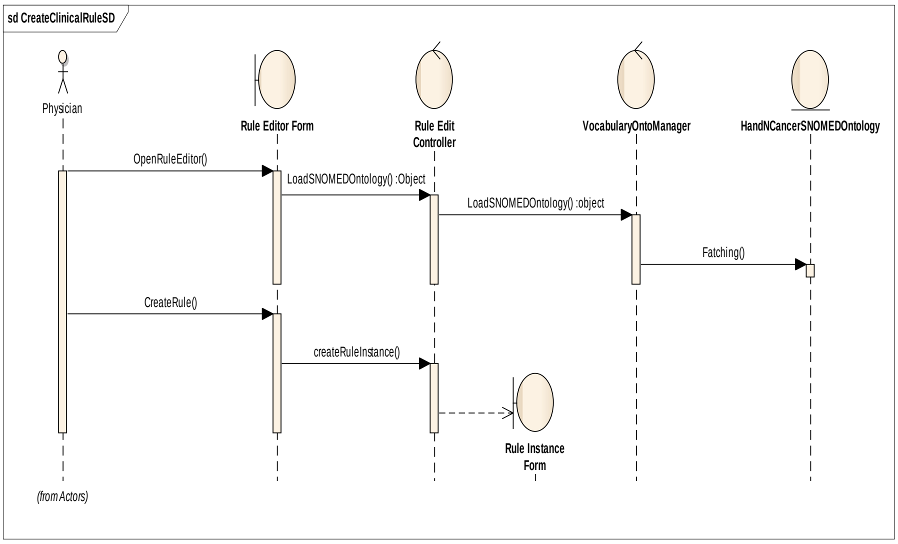


Figure 257 Sequence Diagram: Create Clinical Rule

- **Validate Rule**
  - Physician validate the created rule by CheckValidation() function, this function will perform by Rule Edit Controller using ValidateRule(), internally this function will check the created rule with standard format of MLM that store in Standard MLM Format Repository. If the created MLM (rule) is valid, then save it in MLM File Repository. Following is the sequence diagram.

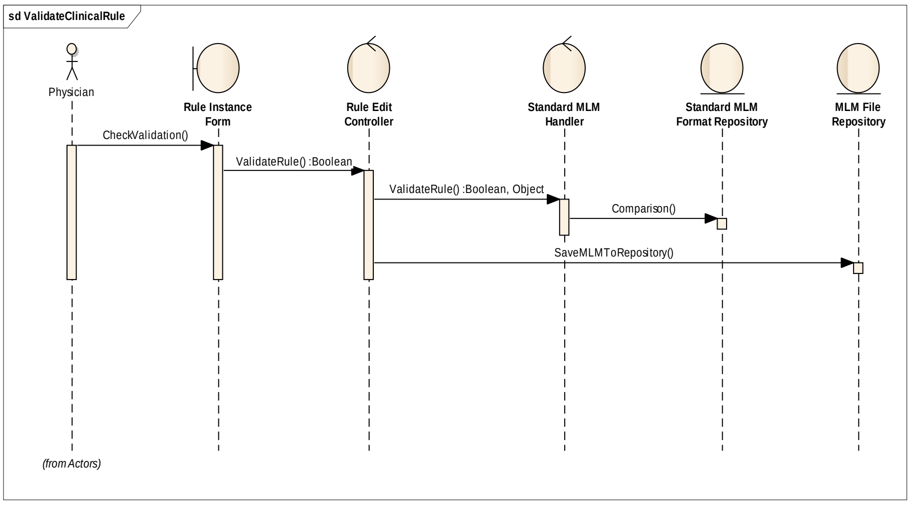


Figure 258 Sequence Diagram: Validate Rule

- **Compile Rule**
  - After checking validation, the physician will save the created rule, but before saving this rule the system will perform parsing and compiling this rule. Before saving in repository, a Parser will divide MLM in some categories like Maintenance, Library and Knowledge by CategorizeMLM(). After categorization, the executable file is generated by using GenerateExecutableCode(). Finally the executable code is compiled and an exe file has been created by CreateExeFile() and the exe file is saved in Executable Knowledge Base by SaveExeFileToKnowledgeBase(). Following is sequence diagram of Compile Rule.

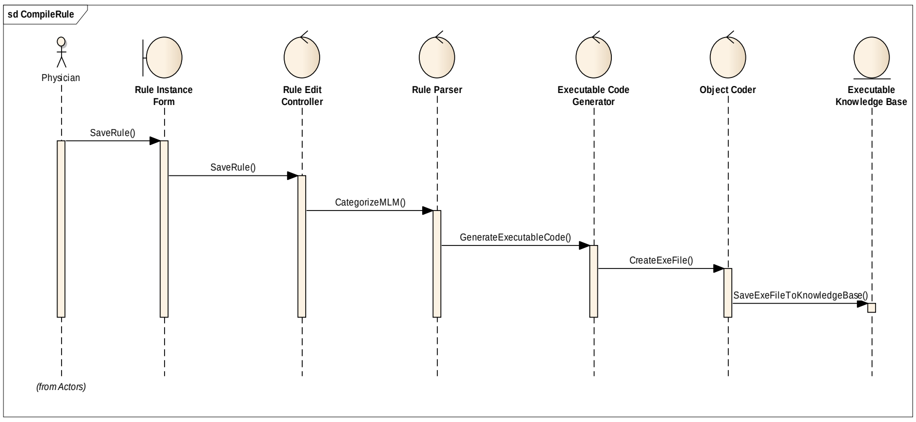


Figure 259 Sequence Diagram: Compile Rule

- **Class Diagram**
  - The purpose of class diagram is to show the relationships among the classes of the system. Additionally it covers the properties and operations that are visible at higher level. Class diagram provides the higher level overview of the system of how to be transformed into a detailed object model which ultimately helps to implement the system. Following section depicts the class diagram for Smart CDSS. The diagram is divided into three sub diagrams as shown in Figures 260, 261, and 262 respectively.

  - **Smart CDSS Reasoner and Execution Environment Class Diagram**

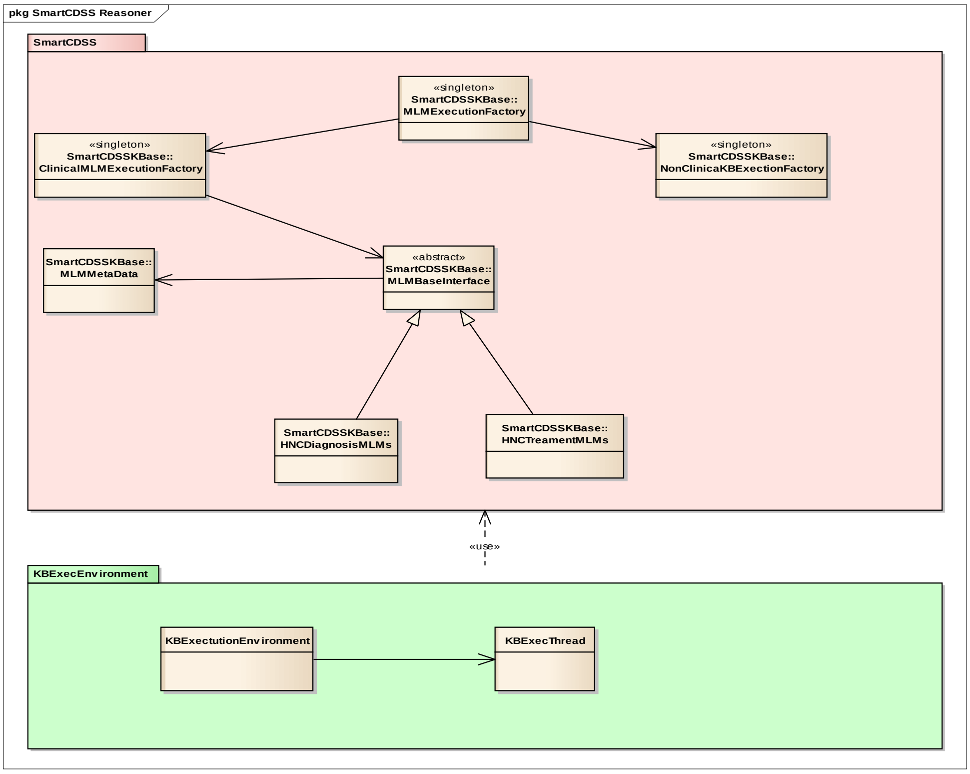


Figure 260 Partial Class Diagram: Reasoner and Execution Environment

- **Smart CDSS Knowledge Base Class Diagram**

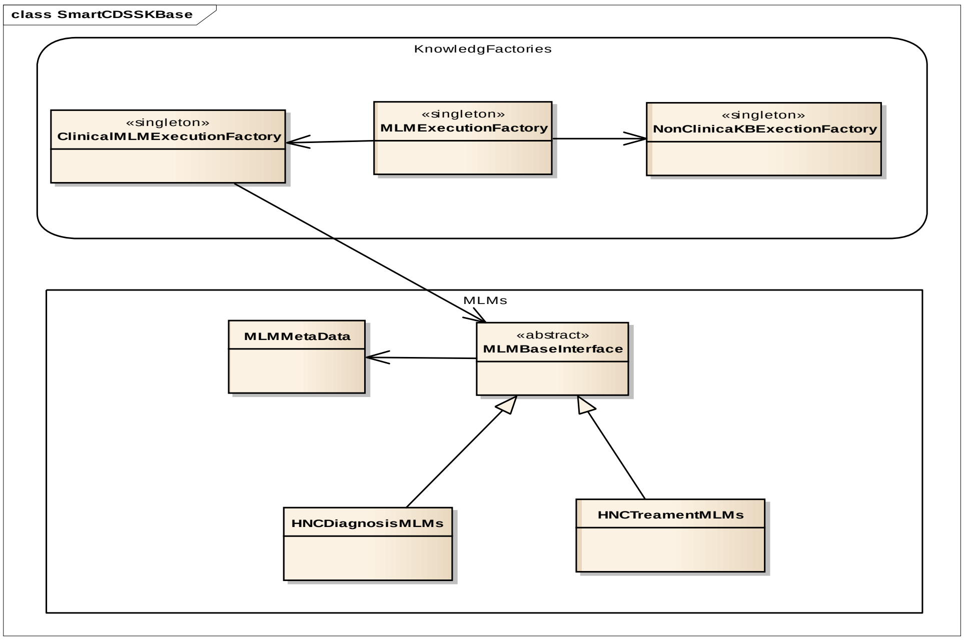


Figure 261 Partial Class Diagram: Knowledge Base

- Authoring Tool

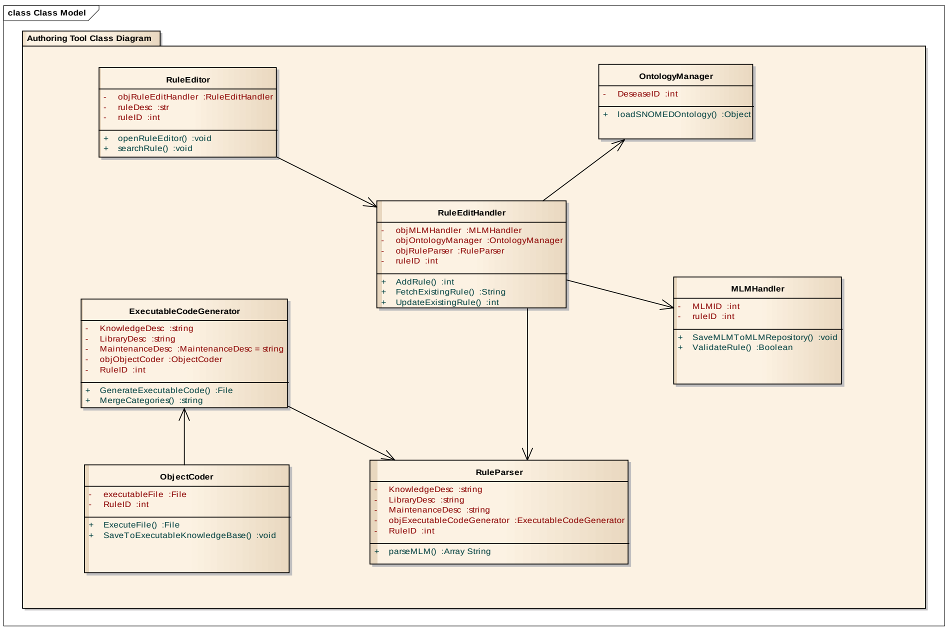


Figure 262 Partial Class Diagram: Authoring Tool

- **Component Diagram**
  - **Smart CDSS Core Components**
    - A component is a software package, or a module, that encapsulates a set of related functions. All system processes are placed into separate components so that all of the data and functions inside each component are semantically related (just as with the contents of classes). Because of this principle, it is often said that components are modular and cohesive. By keeping above definition of software component, we divided our work into different components like Authoring tool, Interface, Reasoner, Knowledge Base and Alert Generator.

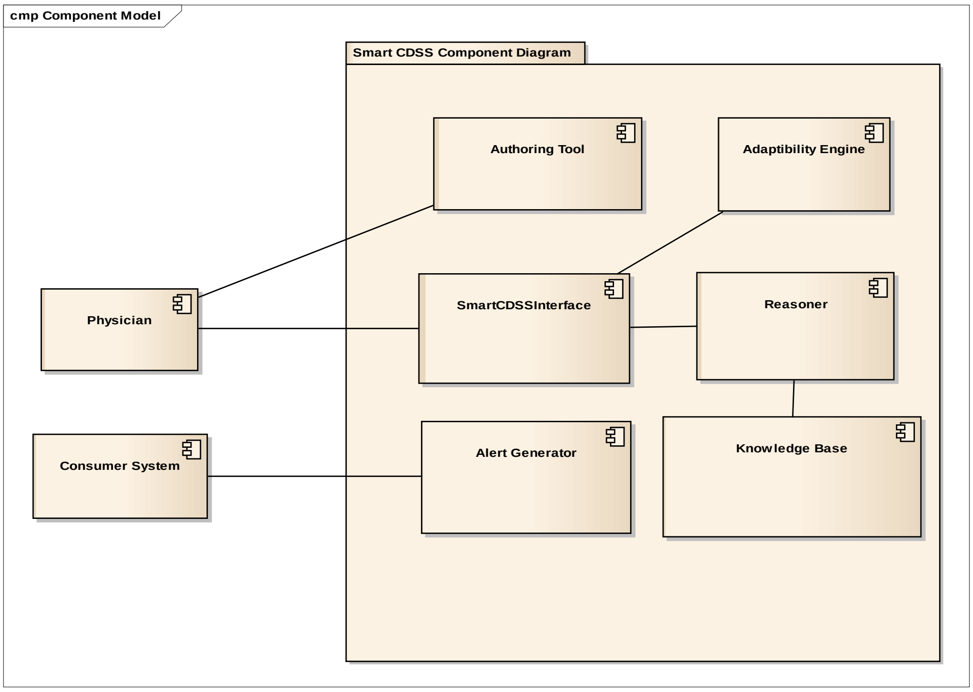


Figure 263 Component Diagram: Smart CDSS Core Components

  - **Component Diagram for Authoring Tool**
    - A component is a software package, or a module, that encapsulates a set of related functions. All system processes are placed into separate components so that all of the data and functions inside each component are semantically related (just as with the contents of classes). Because of this principle, it is often said that components are modular and cohesive. By keeping above definition of software component, we divided our work into different components like Rules Editor, MLM Validator, Compilation Module and Knowledge .In detail level there are some sub components of main component like Rule Editor has subcomponents of Arden Syntax Editor and SNOMED Ontology.

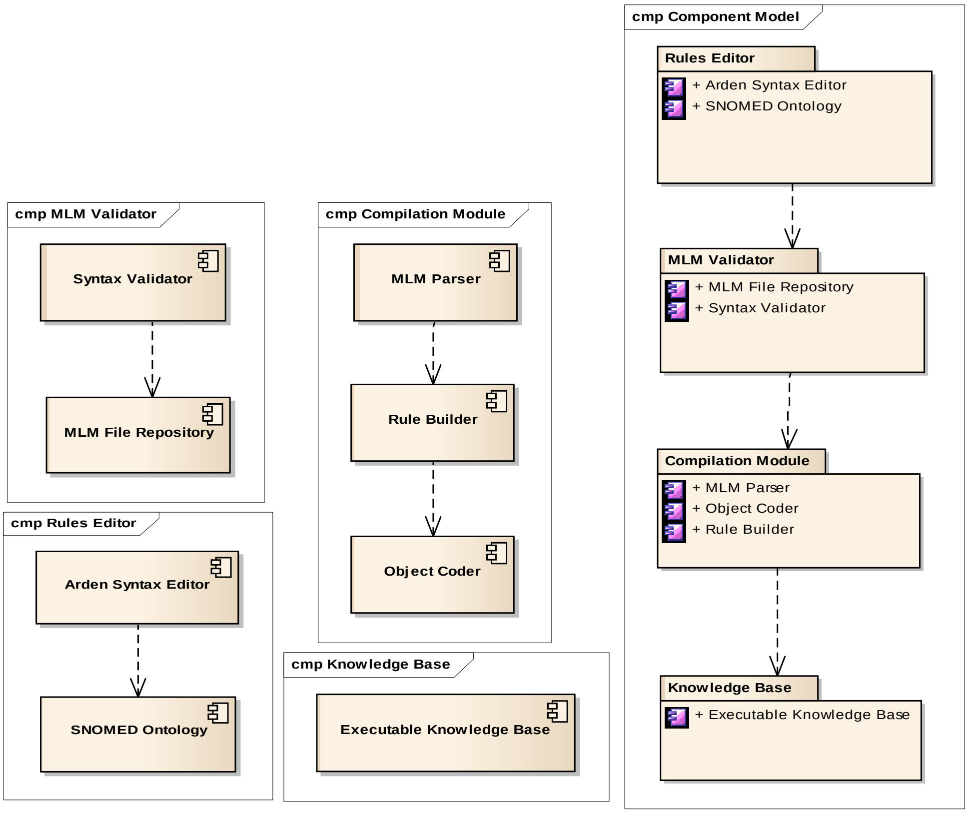


Figure 264 Component Diagram: Authoring Tool

6.4. Cloud Computing and Security

6.4.1. MPJ-Core

- **Use Case Diagram**
  - The use case diagram of MPJ-Core is shown in Figure 265, where as its description is given below.
  - **Actors:**
    - **MPJ-Daemon:** An agent installed over every participating cloud node, involved in node handshake by sharing socket object among nodes.
    - **MPJ-Runtime:** Package that connects the communication device layer of MPJ-Core to Hypervisor.

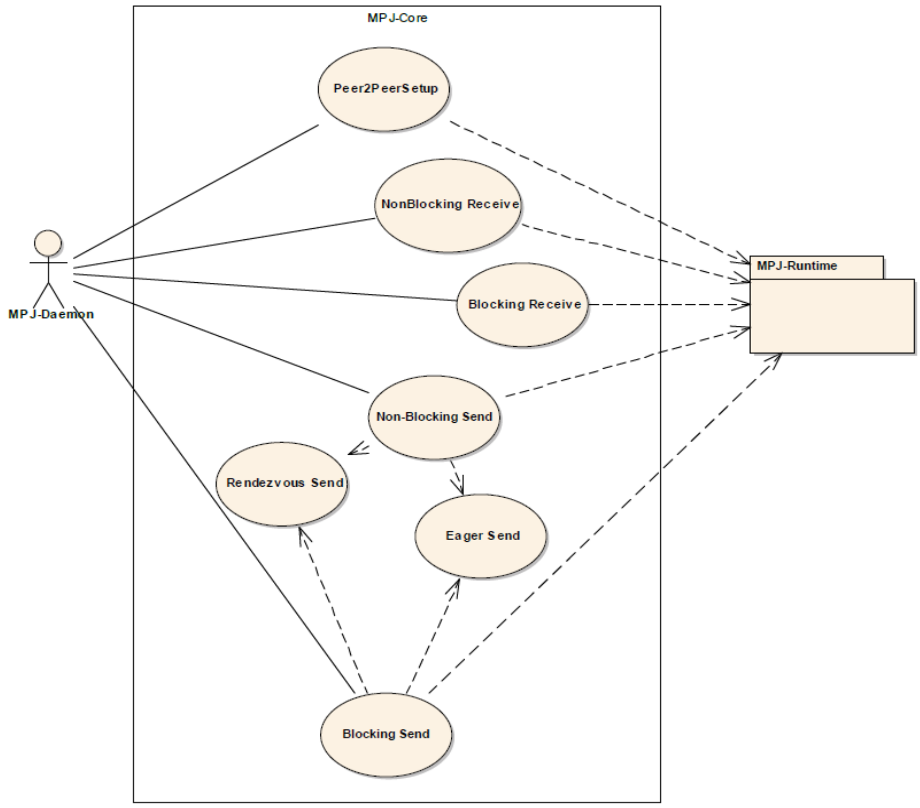


Figure 265 MPJ-Core Use Case Diagram

- **Sequence Diagram**
  - Sequence diagrams for MPJ-Core are described below that shows the interaction between different objects of the system for the achievement of objectives.
  - **MPJ-Core Peer2Peer Setup**
    - MPJ-Daemon reads the local port and ip via machine, conf and sends it over to p2p Initializer for Ip Table creation.
    - P2p Initializer passes the ip and port number to Message package where the socket table with uuid's for the node is created.
    - P2p Initializer connects to the socket and sends uuid to its peers.

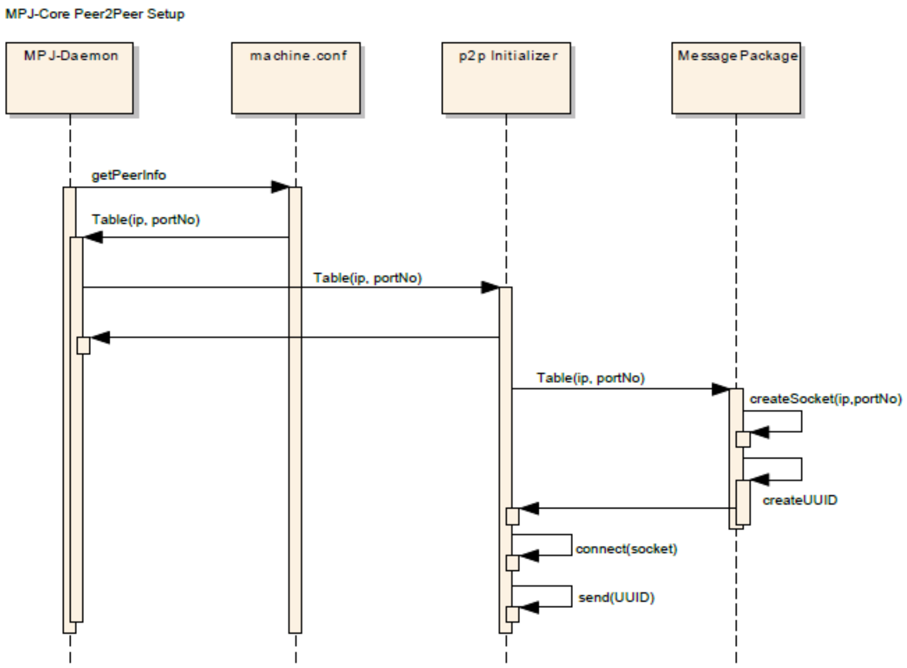


Figure 266 MPJ-Core Peer2Peer Setup Sequence Diagram

- **Blocking Send**
  - Message package gets the socket from socket table by providing the destination uuid.
  - Message package packs the header with message and send it over to Message Type Interface.
  - Message type interface treats the message as request and invokes blocking send object for communication.
  - Blocking send initializes the comm device and sends the request.

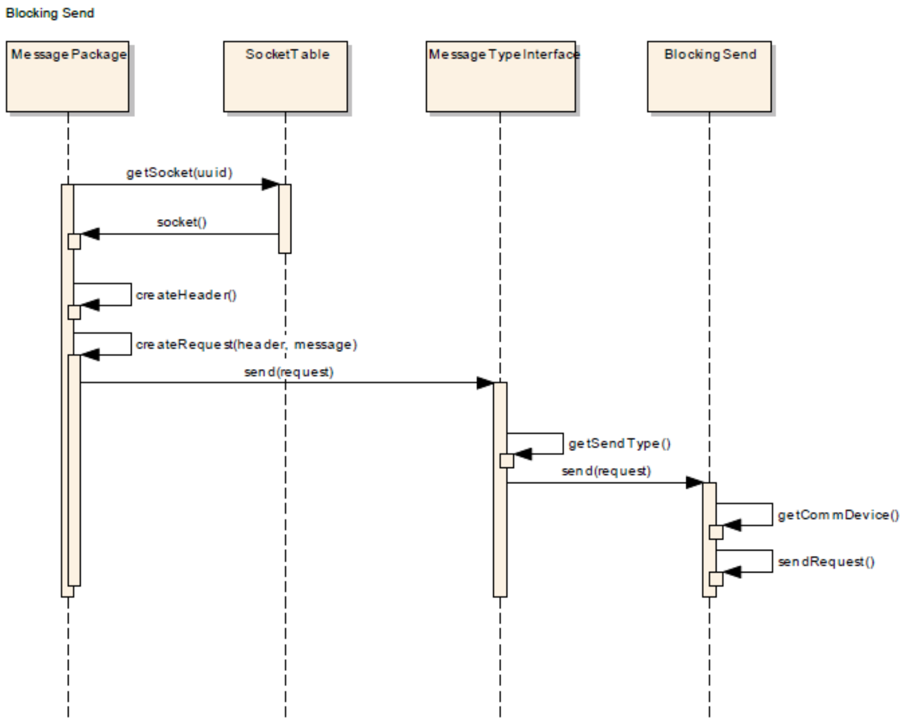


Figure 267 MPJ-Core Blocking Send Sequence Diagram

- **Non-Blocking Send**
  - Message package gets the socket from socket table by providing the destination uuid.
  - Message package packs the header with message and send it over to Message Type Interface.
  - Message type interface treats the message as request and invokes blocking send object for communication.
  - Non-Blocking send initializes the comm device and sends the request.

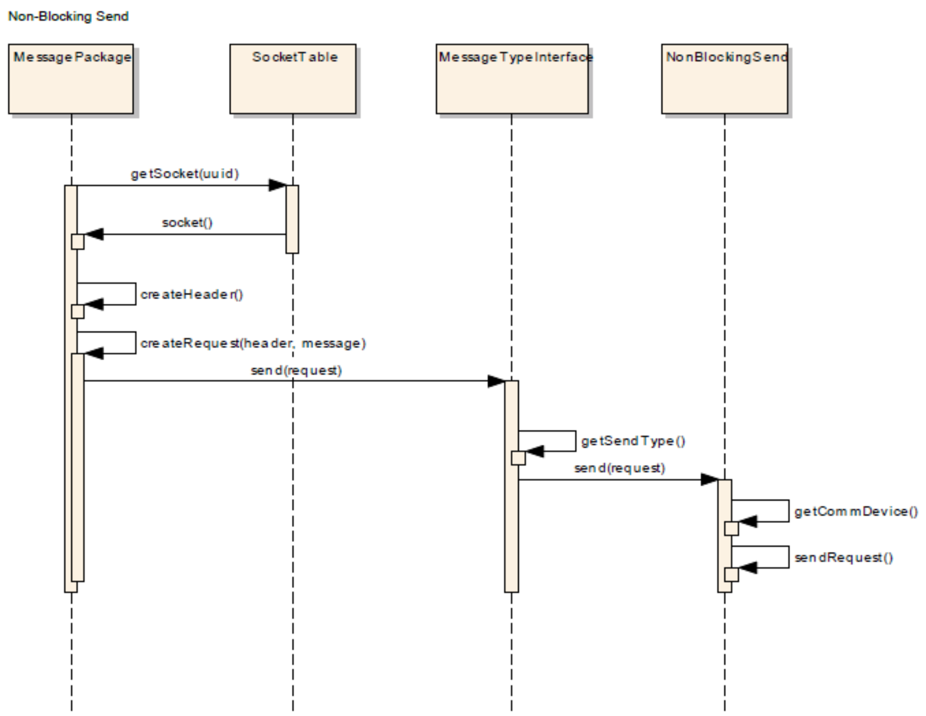


Figure 268 MPJ-Core Non-Blocking Send Sequence Diagram

- **Eager Send Protocol**
  - Sender sends control message to the receiver and sends actual request object without acknowledgement.

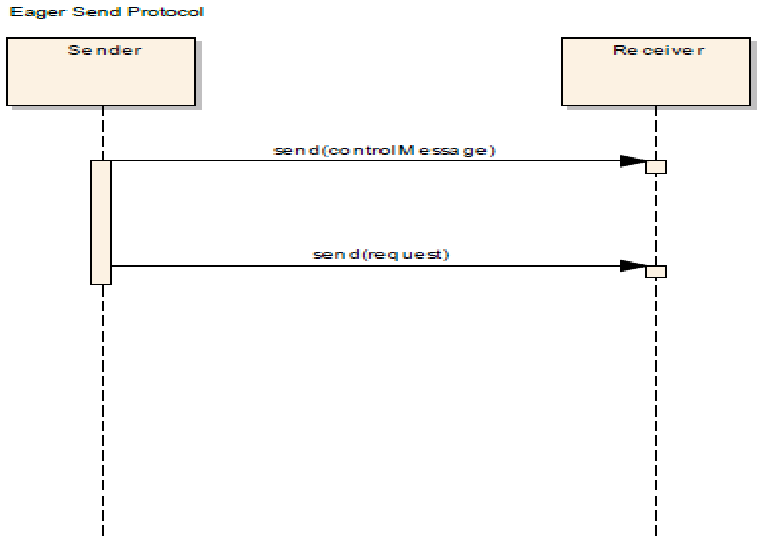


Figure 269 MPJ-Core Eager Send Protocol Sequence Diagram

- **Rendezvous Send Protocol**
  - Sender sends control message to the receiver and sends actual request object after acknowledgement.

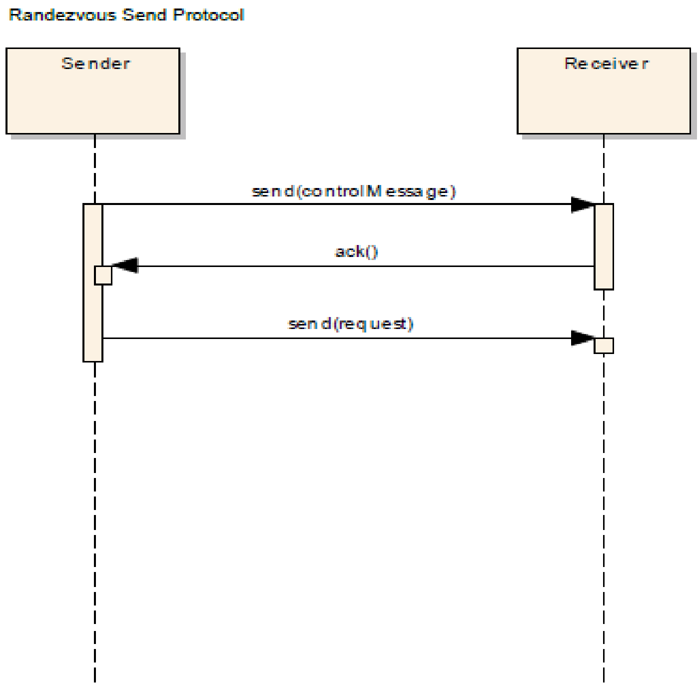


Figure 270 MPJ-Core Rendezvous Send Protocol Sequence Diagram

- **Eager Send Buffered Protocol**
  - Sender copies the request object over buffer and sends control message to the receiver and sends actual request object without acknowledgement.

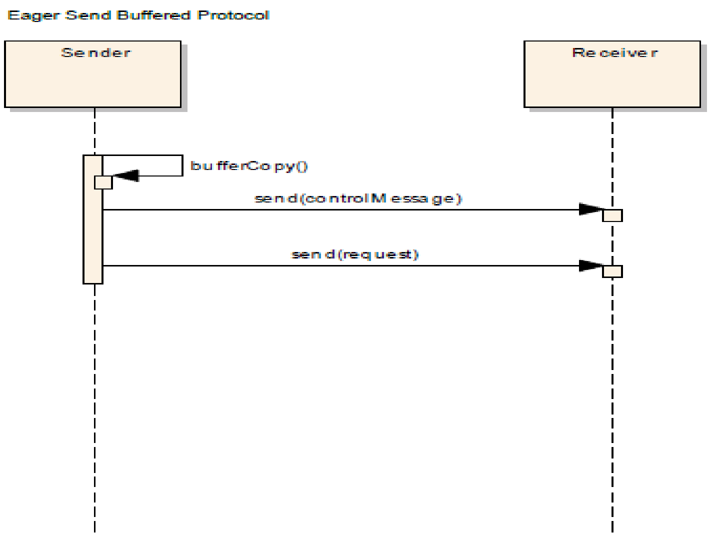


Figure 271 MPJ-Core Eager Buffered Send Protocol Sequence Diagram

- **Rendezvous Buffered Send Protocol**
  - Sender copies the request object over buffer and sends control message to the receiver and sends actual request object after acknowledgement.

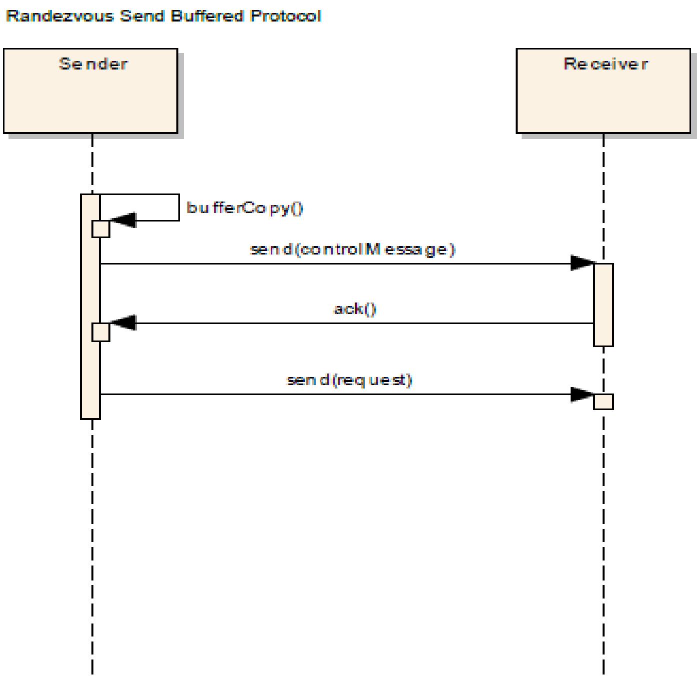


Figure 272 MPJ-Core Rendezvous Buffered Send Protocol Sequence Diagram

- **Class Diagram**
  - **Class** diagram of MPJ-Core shows the different classes and their relationships with each other.
  - Device interface has been implemented by IODevice and NativeDevice, providing device itself as a polymorphic instance.
  - IODevice passes the socket object to IORequest to invoke send and receive over iodev.
  - NativeDevice passes the message object to NativeRequest which invokes send and receive over native mpi device.

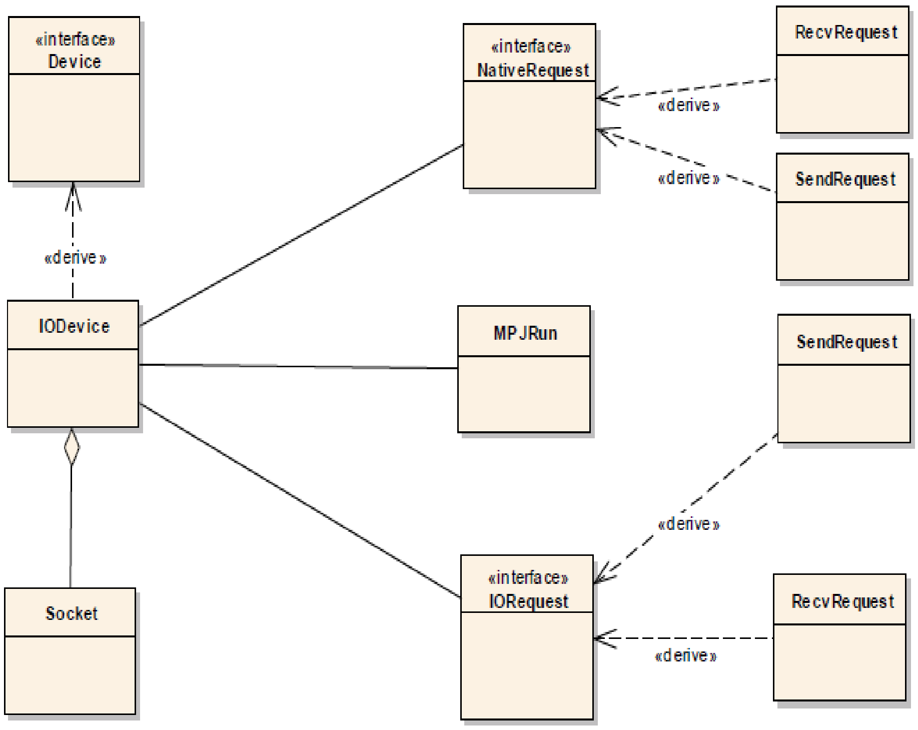


Figure 273 MPJ-Core Class Diagram

- **Component Diagram**
  - Component diagram of MPJ-Core shows the different components and their relationships with each other.
  - Buffer component provides implementation of buffer classes including IO and NIO based buffers.
  - Devx is an interface component providing standardization over comm device implementation.
  - DevX is implemented in IODev, NIODev and NativeDev components as IODevice, NIODevice and NativeDevice respectively.

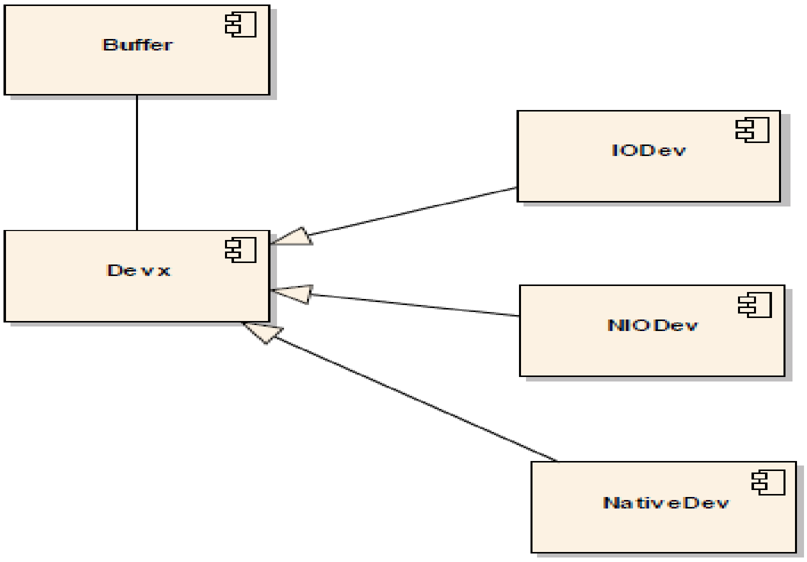


Figure 274 MPJ-Core Component Diagram

6.4.2. MPJ-Runtime

- **Use Case Diagram**
  - The use case diagram of MPJ-runtime is shown in Figure 275, where as its description is given below.
  - **Actors:**
    - **MPJ-Daemon:** MPJ daemon is a trigger which will help in deploying processes among different virtual machines.

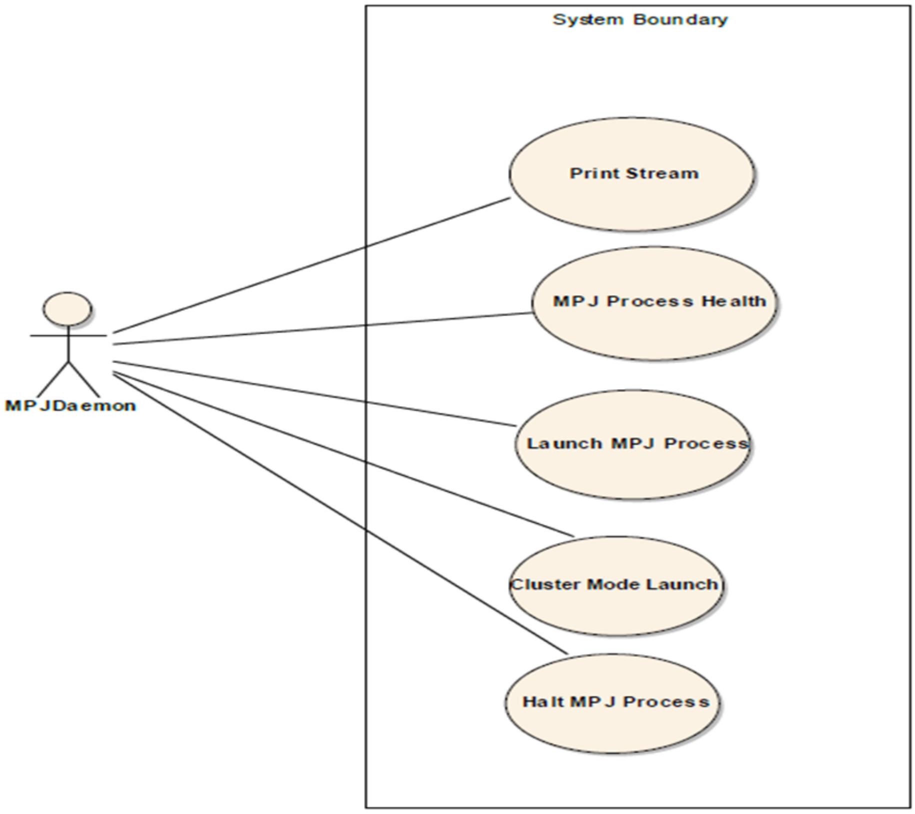


Figure 275 MPJ-Runtime Use Case Diagram

- **Sequence Diagram**
  - Sequence diagrams for MPJ-runtime are described below that shows the interaction between different objects of the system for the achievement of objectives.
  - **MPJ-runtime Process Launch**
    - The script is executed for running the processes and relevant arguments are sent to the wrapper class.
    - It filters the arguments and MPJDaemon then launches the processes via process builder class and log the stream generated.

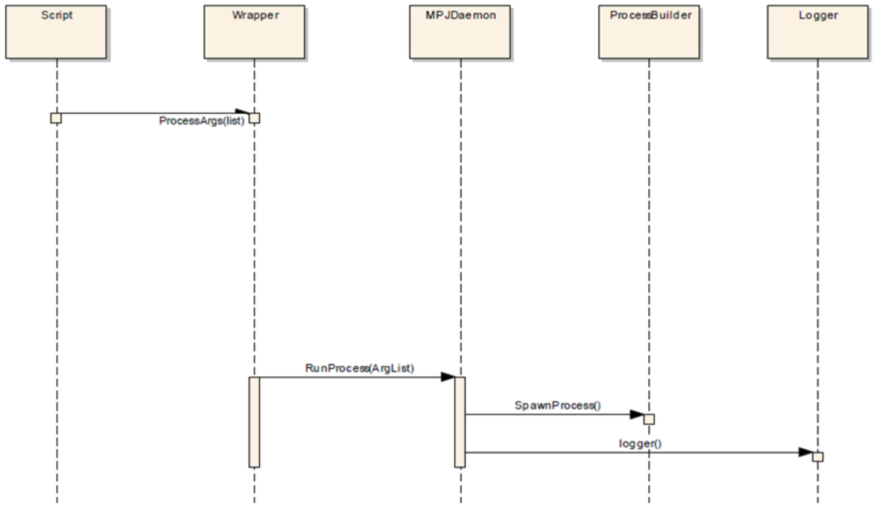


Figure 276 MPJ-Runtime Process Launch Sequence Diagram

- **MPJ-runtime Process Health**
    - The inquire class calls MPJHealth for a process information on a particular node or all the nodes. The MPJDaemon than acquires the status via an acknowledgment.

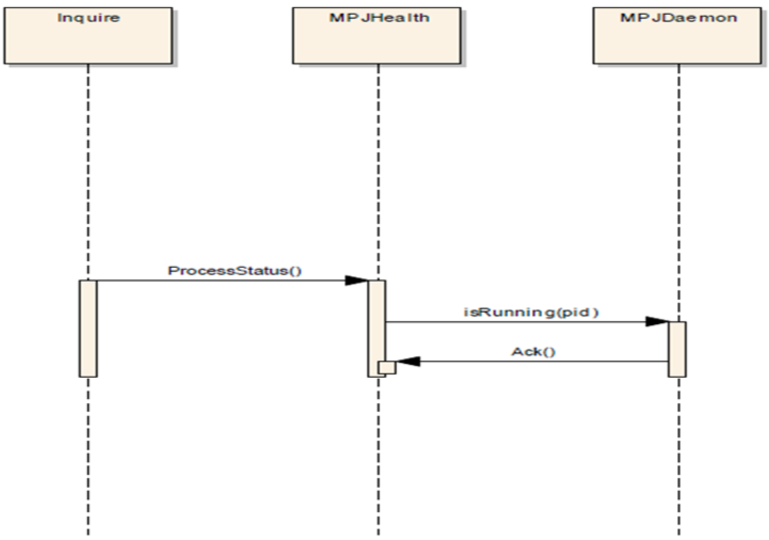


Figure 277 MPJ-Runtime Process Health Sequence Diagram

- **MPJ-runtime Halt Process**
    - The inquire class calls MPJHalt for a process information on a particular node or all the nodes. The MPJDaemon kills the required process with the given process id.

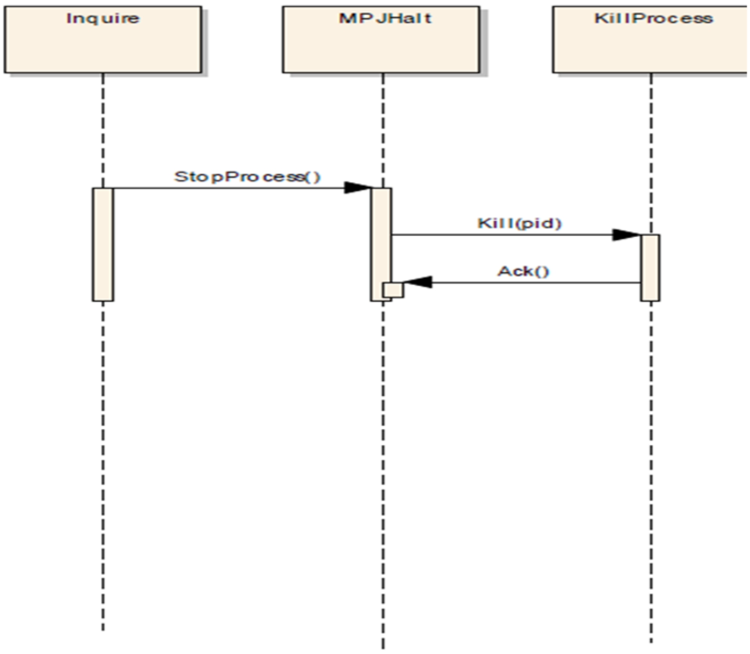


Figure 278 MPJ-Runtime Halt Process Sequence Diagram

- **Class Diagram**
  - Class diagram of MPJ-runtime shows the different classes and their relationships with each other.
  - MPJDaemon is the main class which will launch the processes and create logger which will invoke a print stream for all processes.
  - Multicore Daemon runs the processes on a multicore architecture.
  - HaltProcess takes the process id and kills the process.
  - MPJHealth notifies if a process is running or crashed.
  - Info class tells us the information about the architecture of cluster etc.

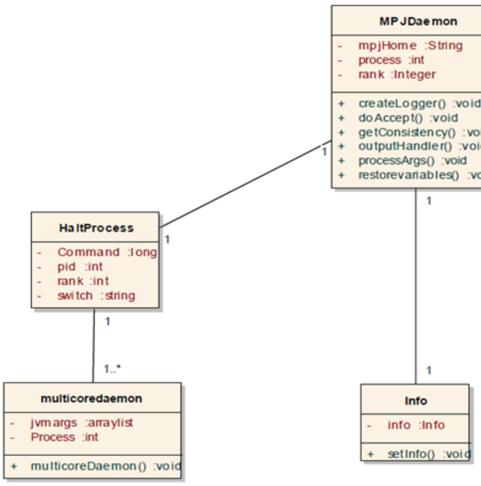


Figure 279 MPJ-Runtime Class Diagram

- **MPJ-Runtime Component Diagram**
  - Component diagram of MPJ-Runtime shows the different components and their relationships with each other.
  - Buffer component provides implementation of buffer classes including IO and NIO based buffers.
  - xDev is an interface component providing standardization over comm device implementation.
  - Runtime Starter handles the multicore configuration for the process launch.
  - The daemon class is the main class which handles the daemon launch and process launch as well as process monitoring.

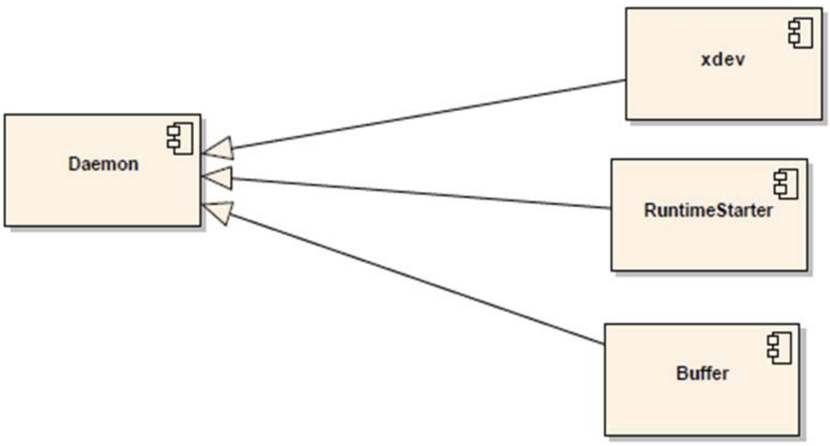


Figure 280 MPJ-Runtime component Diagram

6.4.3. Parallel I/O

- **Use Case Diagram**
  - The use case diagram of Parallel I/O is shown in Figure 281, where as its description is given below.
  - **Actors:**
    - **MPJ Application: Program:** MPJ Application: Program is the main user of the system that is using Parallel I/O library to read and write files in parallel.
    - **mpiJava library:** Is the main component which is invoked by the application using the JNI layer.

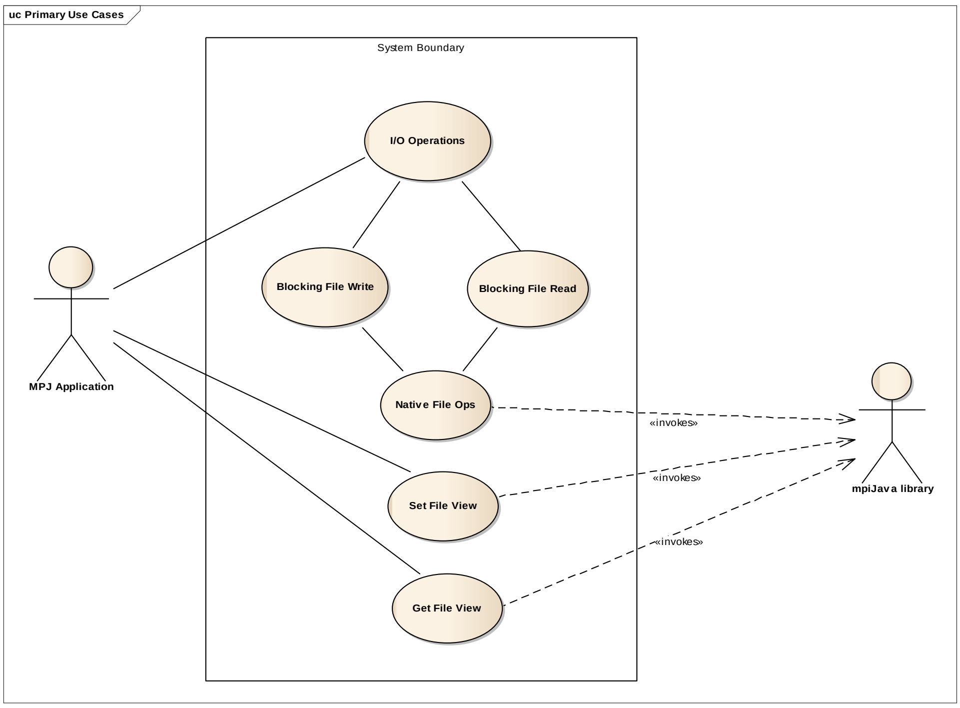


Figure 281 Use case diagram of Parallel I/O

- **Sequence Diagram**
  - The sequence diagram in Figure 282 shows the sequence for setting or getting file view operations. Since, our system has very specific standard way of accessing, all of the file related operations will follow the same set of sequence diagrams for “get and set” operations like getAtomicity, setAtomicity etc.

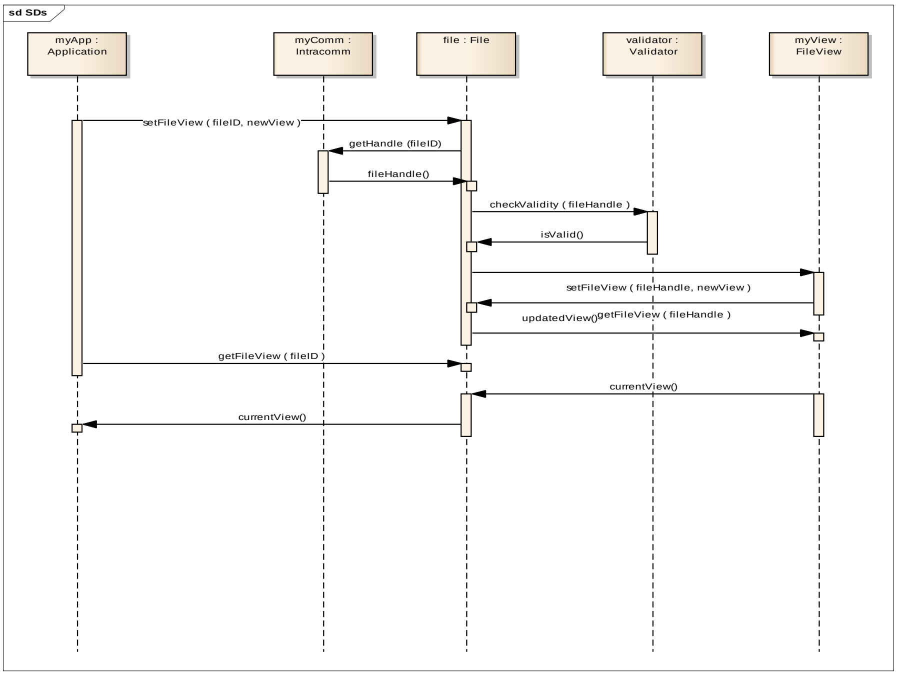


Figure 282 Sequence diagram of Parallel I/O

- **Class Diagram**
  - Class Diagram shows the different classes that we will develop. Some of the classes will be used from the existing system.

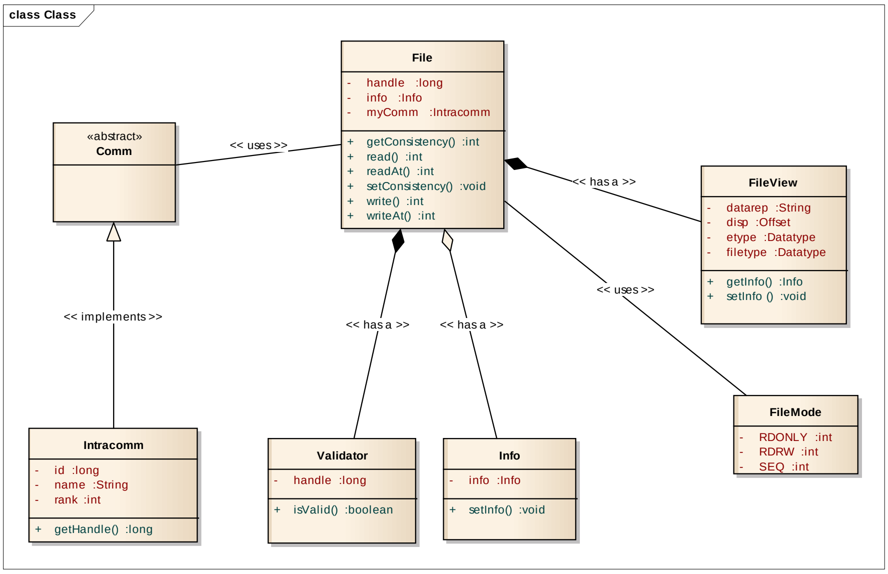


Figure 283 Class Diagram of Parallel I/O

- **Component Diagram**
  - The component diagram explains and shows the components of our system in Figure 284.

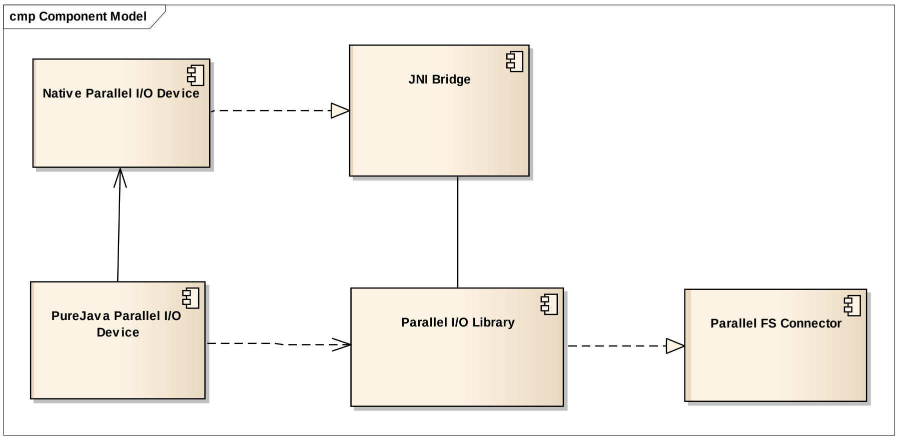


Figure 284 Component Diagram of Parallel I/O

6.4.4. Anonymization and Cloud Security

- **Use Case Diagram**
  - The use case diagram for anonymization is shown in the following figure. Here a user or agent who is interested for the anonymization service is shown as an actor. The anonymization service starts with the original data and transforms it into the desired output.

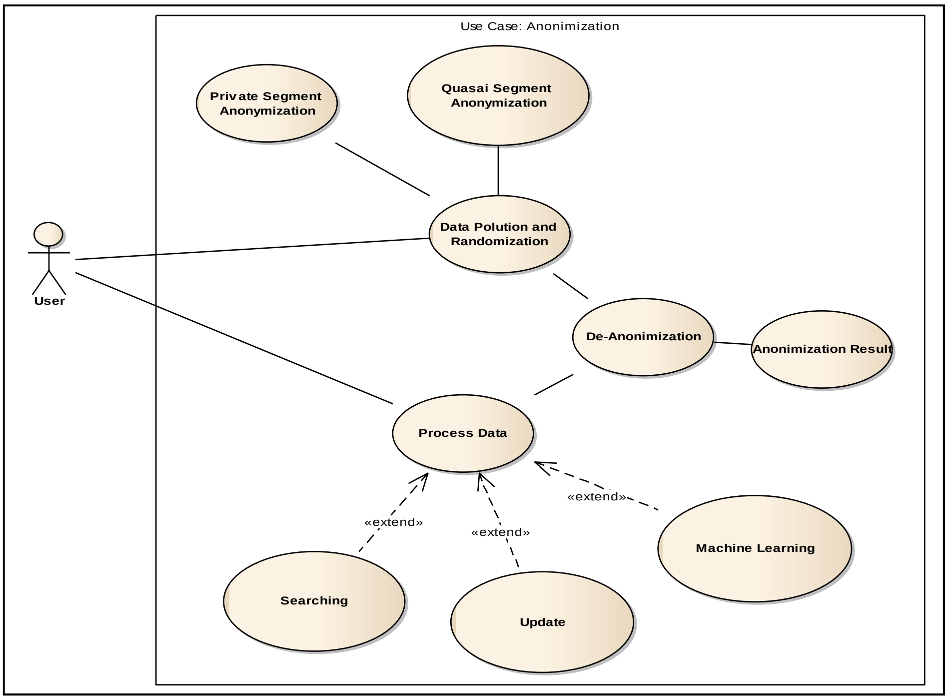


Figure 285 Anonymization Service: Use Case Diagram

- **Sequence Diagram**
  - The sequence diagram is shown below with the interaction of different objects on the timeline during the anonymization.

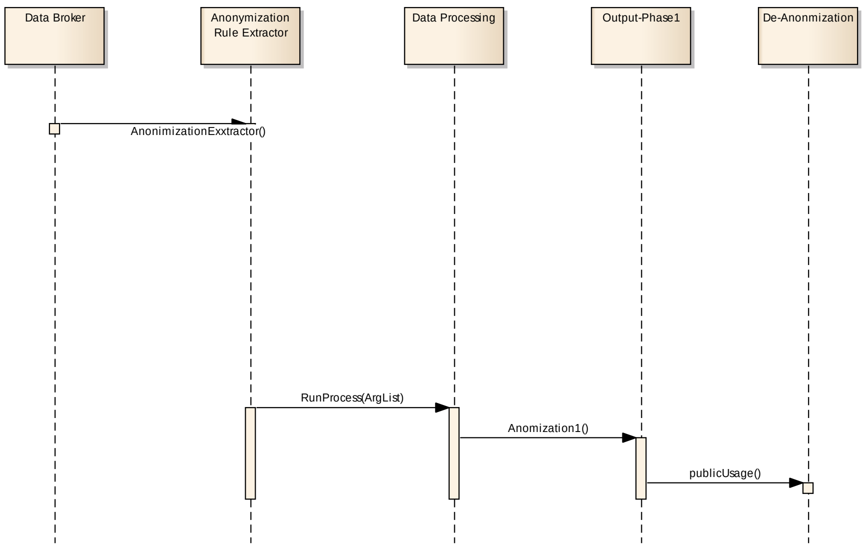


Figure 286 Anonymization: Sequence Diagram

- **Class Diagram**
  - Interaction of various classes in anonymization service is shown in following class diagram.

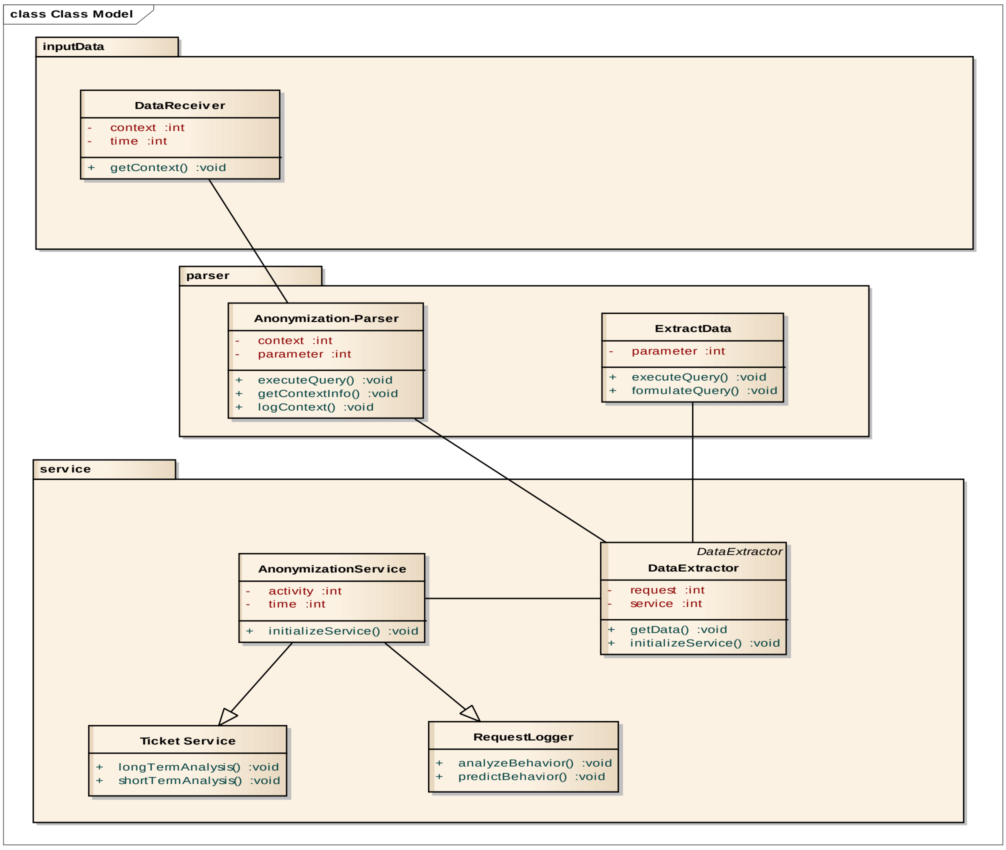


Figure 287 Anonymization: Class Diagram

- **Component Diagram**
  - The components diagram of anonymization is given below

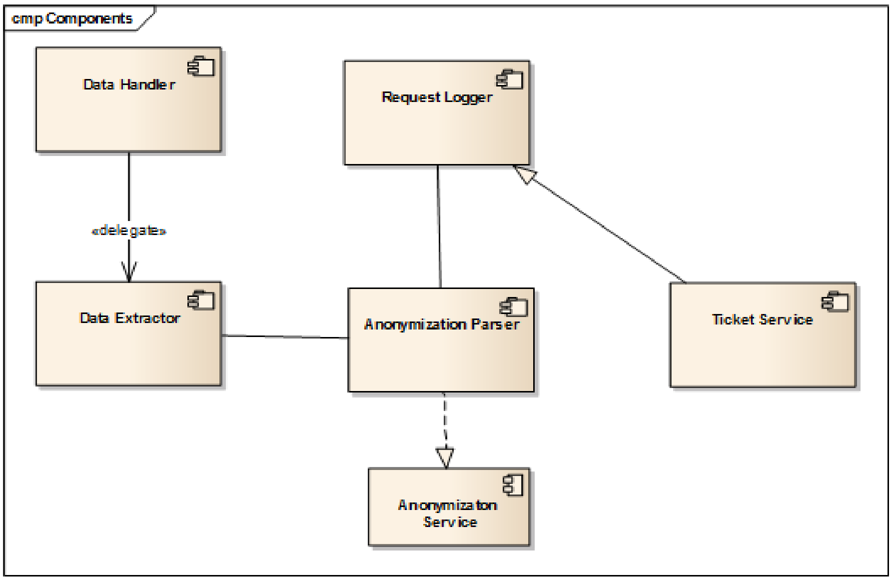


Figure 288 Anonymization: Component Diagram

APPLICATION SENARIOS OF SC^3 SERVICES

- In this chapter we discuss four different application scenarios of SC^3 that we have developed and deployed.
  - First application scenario is CAME for Alzheimer's disease Patient
  - Second is Wellness services for normal user of LDSS for Wellness
  - Third is the scenario of LDSS for Chronic Disease (Diabetic Patient)
  - Fourth scenario is the SmartCDSS development for Head & Neck Cancer
- All these scenarios are discussed in the subsequent sections.

7.1. Application Scenario: CAME

- Our general system deployment is shown in Figure 289. The patient's house includes a kitchen, a bed-room, and a living room.
- Several sensors and cameras are deployed in the patient's house to collect sensory data and images.
- We deploy a cloud gateway in the living room to collect data from all sensors and cameras. It connects to the Cloud via Internet high speed router.
- Doctors, nurses, and patient's relatives (e.g., his daughter) can access easily via Web2.0 interface.

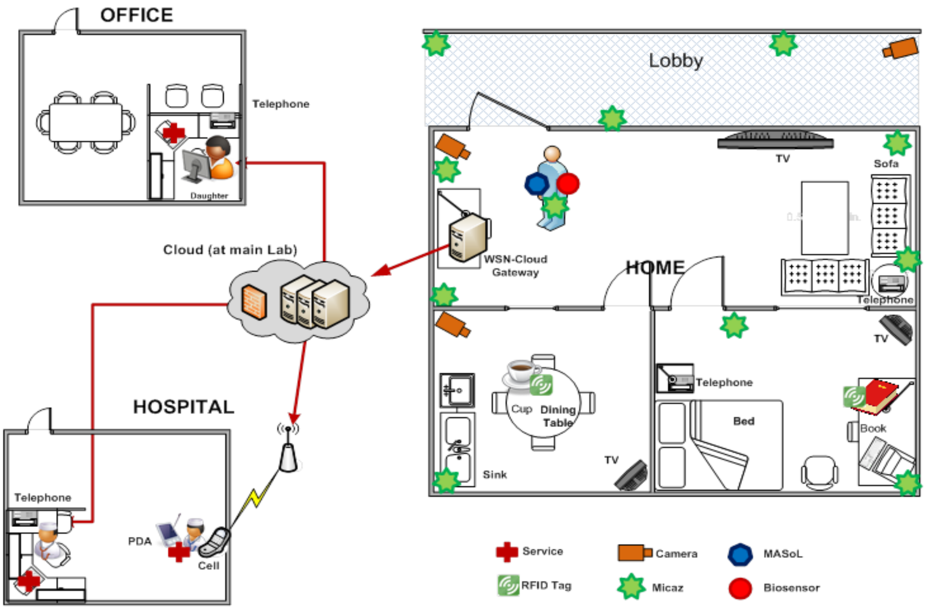


Figure 289 Our ITRC Test bed where sensors are deployed to collect user activities

- The following figures (Figure 290, 291, 292, and 293) show how we collect data from sensors and cameras and deploy to the Cloud.

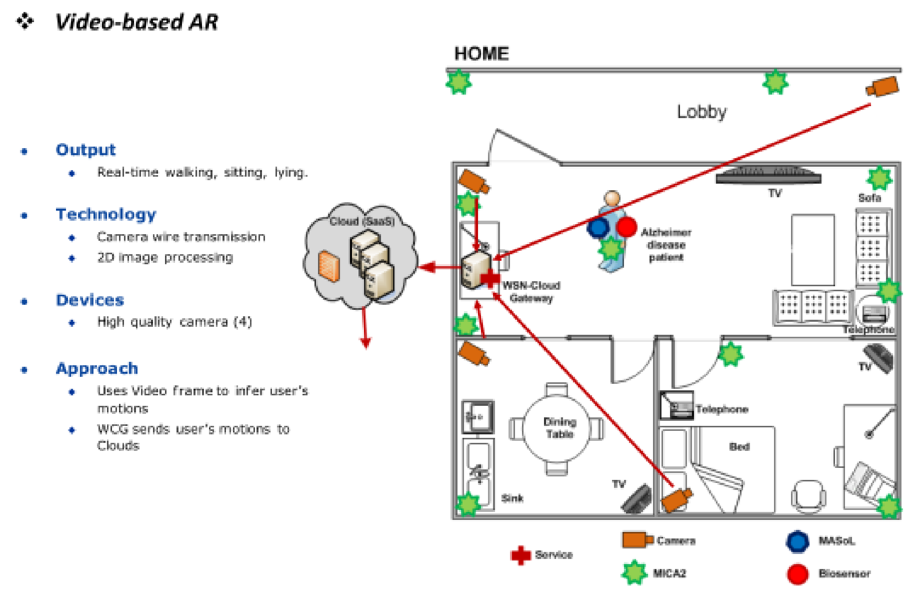


Figure 290 Video Based Activity Recognition

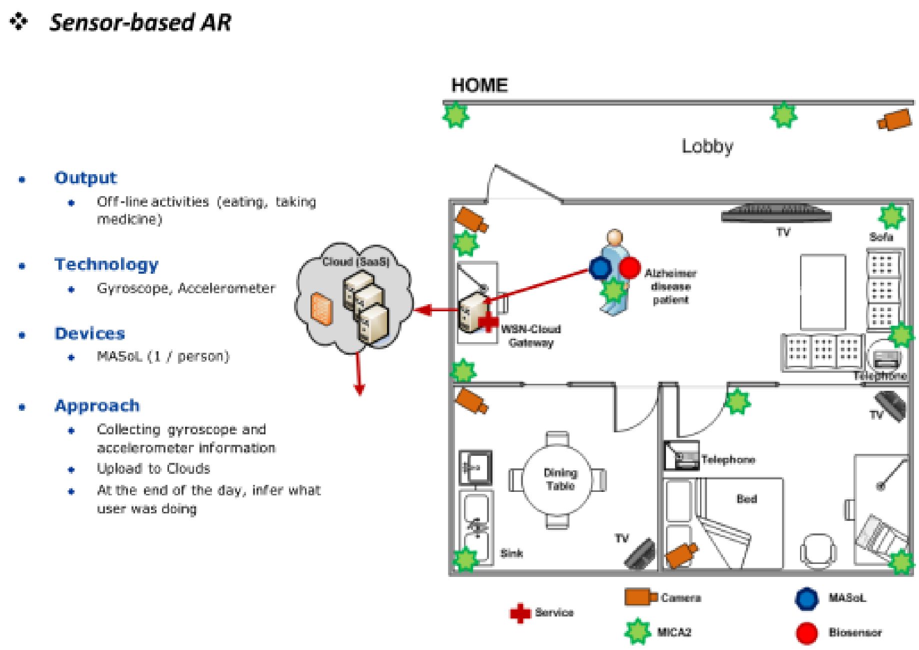


Figure 291 Sensor Based Activity Recognition

❖ *Location Tracking*

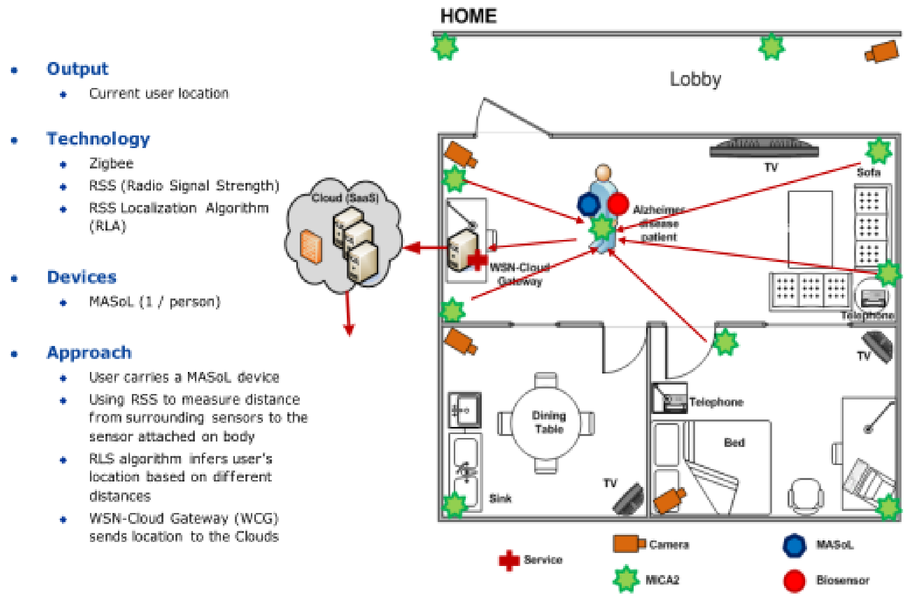


Figure 292 Location Tracking

❖ *Medical Data Collection*

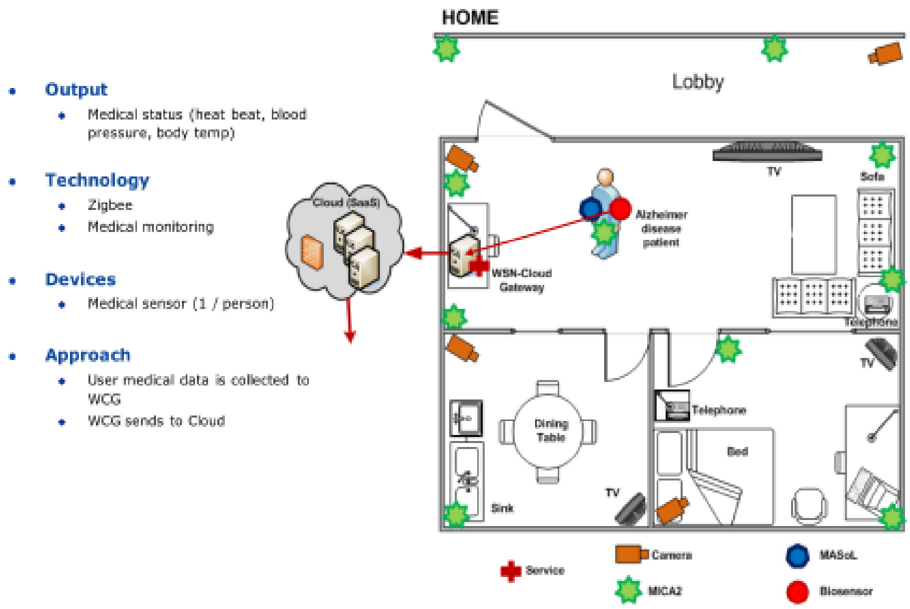


Figure 293 Medical Data Collection

- A home network is deployed with sensors and cameras to detect user's activity as illustrated in Figure 294.
- We customized our ITRC (Information Technology Research Center) test-bed room as a patient's home environment with a living room, a kitchen, and a bedroom.
- We use a WiTilt V3 sensor supported accelerometer and gyroscope attaching on the patient's right hand to detect his activity such as taking medicine, reading book, eating, teeth brushing. In each room, we deploy a TinyOSMall PIR motion sensor to detect if the patient is in the room.
- A Logitech wide-angle web camera is attached on the wall of the living room and the kitchen to detect his movement such as watching TV, doing exercise.
- The home gateway is deployed at the patient's home to collect and transmit raw data from sensors and cameras to the Cloud.
- We installed a free source code Enomaly ECP in 4 PCs Pentium IV dual-core 2.5GHz, 3GB RAM to serve as a Cloud server.

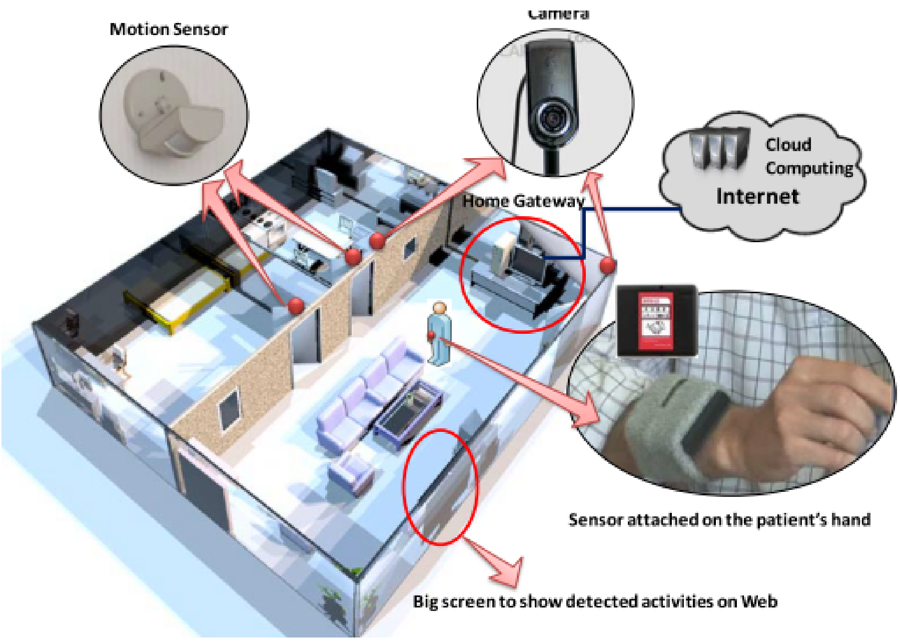


Figure 294 CAME deployment in ITRC Test Bed

- A sample scenario is implemented in order to show how CAME supports an Alzheimer's disease patient. It works as follows.
  - At 7 o'clock in the morning, the patient enters the kitchen and has breakfast.
  - When he enters, the motion sensor sends a sensed signal to the CAME.
  - CAME detects he is in the kitchen, so it sends a command to turn on the light.
  - While he is waiting for breakfast, he sits on the chair and looks at the TV. CAME detects his posture by collecting image data from the camera and inferring the activity. So it sends a command to turn on the TV.
  - Then, CAME collects gyroscope and accelerometer signal from the embodied sensor and infers his eating and teeth brushing action (Figure 295).

Figure 295 CAME recognizes the patient is watching TV, so it turns on the TV. It also detects he is eating, and teeth brushing

- After breakfast, the patient reads a book in the bedroom. Detecting that the patient is reading, CAME turns off the TV so that he can focus on the book.
- A while later, CAME recognizes that he did not take medicine and do exercise for today by checking the activity database which it has recorded.
- So it sends a sound reminder “Take medicine please!”, and then “Take exercise please!” to him.
- When the patient performs those two actions, it updates to the database so that it will not reminder him later on (Figure 296).

Figure 296 CAME detects he is reading, and reminds him to take medicine and do exercise. It then records his actions to the database and not remind later on

- After doing exercise, the patient wants to read the book that he was reading before.
- However, he forgets where he left it. So he looks at the big screen on the wall which displays all activities he has been performing.
- It is a website showing Time, Activity, and Location (Figure 297).
- He then finds down the last time he was reading the book is in the bedroom, so he can easily gets the book.
- The patient also checks his health condition and sends a report to Cloud through a Web 2.0 interface.

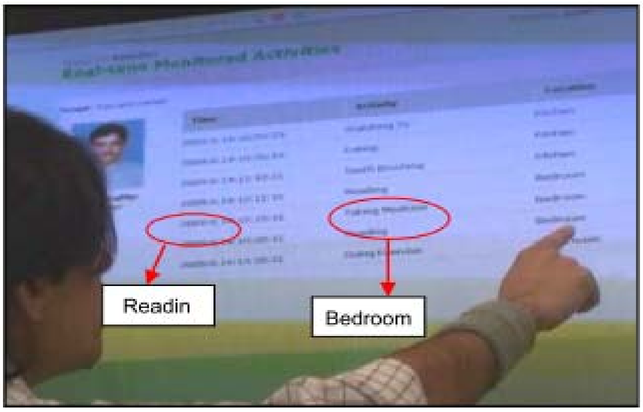


Figure 297 The big screen is showing all the activities

- At the hospital, the nurse accesses to the Cloud and checks the patient's health condition.
- She also can see whether he forgot to do something, such as taking medicine, doing exercise.
- As she concludes the patient is not getting better, she comes to the doctor and they have a short discussion.
- After that, the doctor adds a new medication and lets the nurse brings to the patient (Figure 298).

Figure 298 At the hospital, nurse and doctor check the patient condition via Cloud. A new medication is added and brought to the patient

7.2. Application Scenario: LDSS for Wellness

- **Need for Wellness**
  - Approximately 60% of emergency visits are clinically inappropriate.
  - Approximately 30% of physician office visits are not clinically necessary.
  - Approximately 25% of prescribed medications are not necessary.
  - Only 45% of patients comply with the medical advice they receive.
  - Healthcare cost for poor lifestyle (in USA) as shown in Figure 299.

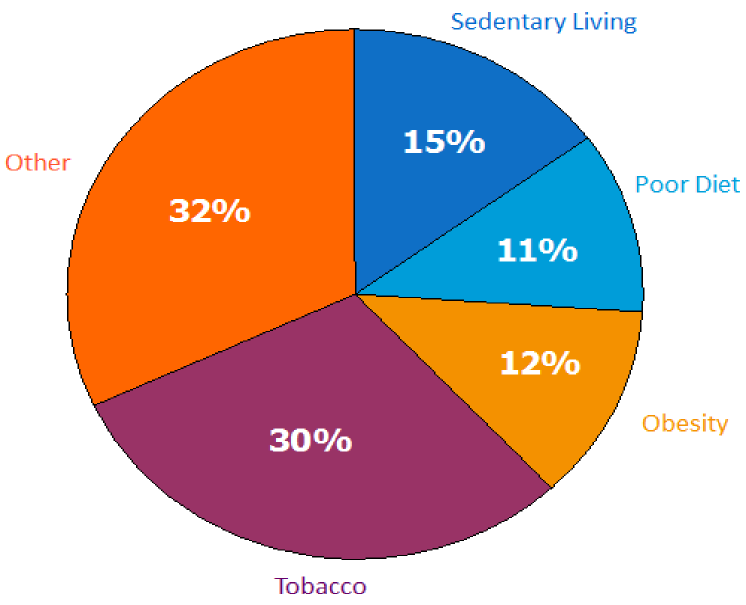


Figure 299 Healthcare cost because of poor lifestyle in USA

- **Principles for better Lifestyle**
  - To overcome this cost these simple principles are required to be followed.
    - Know your condition
    - Have active Involvement in decision making with the caregiver
    - Follow the Care plan that is agreed upon with the caregiver
    - Monitor symptoms associated with the condition(s) and Respond to, manage and cope with the symptoms
    - Manage the physical, emotional and social Impact of the conditions on your life
    - Live a healthy Lifestyle

- **A Case Study for Wellness**
  - We assume that all the developed components and apps of SC3 are deployed and in use of the user.
  - Based on the information collected from these components, LDSS for Wellness will generate recommendations to the user.

Activity: Good Sleep
Time: 07:15:34 hr

Figure 300 Good sleep activity detected

- **Activity 1**
  - The user wake up in the morning and the smartphone app detect that user had a good sleep.
  - The app forwards the information (Shown in 300) to LDSS for Wellness where this information is stored in Life Log.
  - based To overcome this cost these simple principles are required to be followed.

- **Activity 2**
  - The profile information of the user indicate that the user is healthy user as shown in Figure 301.

Profile Information
Health: Good

Figure 301 Healthy User

- **Activity 3**
  - The smartphone app has detected that the user is walking as shown in Figure 302.

Activity: Walking
Time: 08:45:34 hr

Figure 302 User walking

- **Activity 4**
  - The user is in a coffee shop as shown in Figure 303.

Diet: coffee
Time: 08:46:04 hr

Figure 303 User is having coffee

- **Activity 5**
  - The trajectory information is conveying message that user is continuously moving as shown in Figure 304.

Trajectory
Action: moving
Location: City Hall
Time: 10:54:25 hr

Figure 304 User is moving around

- **Activity 6**
  - Smartphone app also triggered that user is walking as shown in Figure 305. During all these activities detected at different time, it is obvious that user did not has any meal.

Activity: Walking
Time: 12:33:34 hr

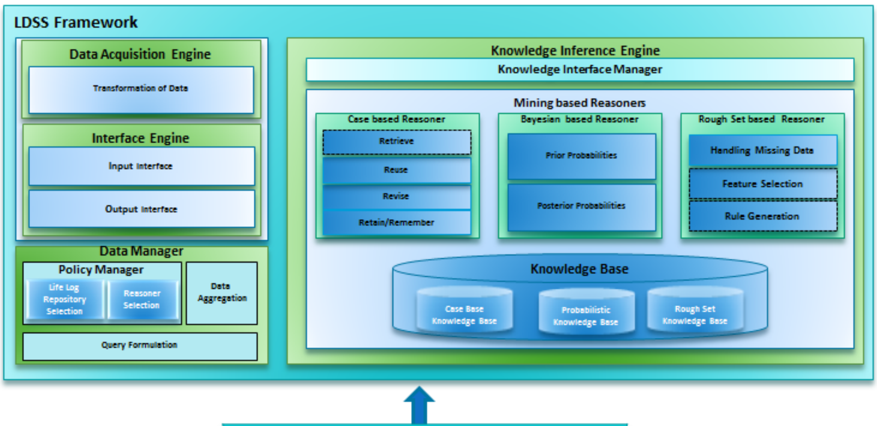
Figure 305 User walking

- **Activity 7**
  - User tweet about his current situation as shown in Figure 306. This is the processed information.

Tweet
Topic: Moving
Sentiments: Negative
Feelings: Tired
Time: 10:54:25 hr

Figure 306 User is moving and feeling tired using tweet

- Now all the collected information is retrieved from Life Log and used for recommendation generation by LDSS for Wellness Recommendation Generation Module as shown in Figure 307.

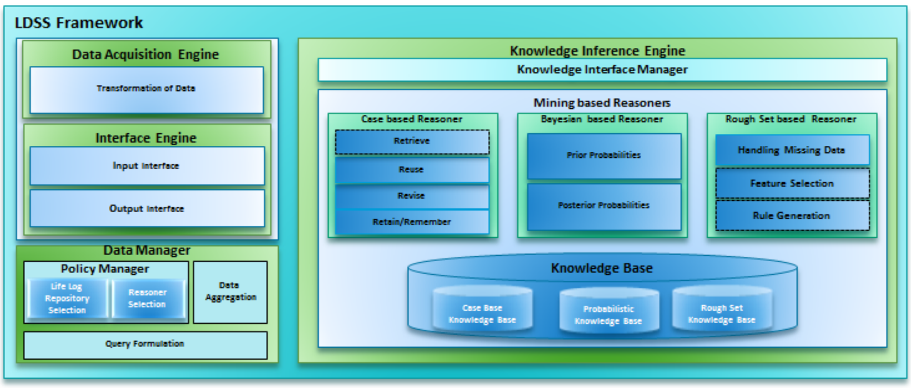


Complete Decision Table for Activities

User	Activity	Diet	Disease	Sleep	Health Condition
U1	Walking	null	No	Good	Good
U1	Moving	null	No	Good	Good
U3	Eating	Extensive	Yes	Good	Bad
U1	Walking	null	No	Good	Good
U5	Bus	..	Yes	Normal	Bad
U6	Eating	Normal	No	Normal	Normal

Figure 307 Process initiation for recommendation generation

- After inferencing, the system understand that the user is not having any meal and also feeling tired.
- So the LDSS for Wellness generate below given (in Figure 308) recommendation for the user.



LDSS Recommendations

Recommendation	Take Lunch and Drinks
POI	Nearest Restaurants are: •Mr. Pizza •I am Fine Chicken •Korean Restaurant

Figure 308 Recommendations for the user including map

7.3. Application Scenario: LDSS for Chronic Disease

- **VMR Data from CAME**
  - CAME finds the high level activities of the chronic disease patient and adaptability engine converts it into VMR. The Figure below shows chronic disease patient doing exercise activity in living room at a given time. This information is converted to VMR format.

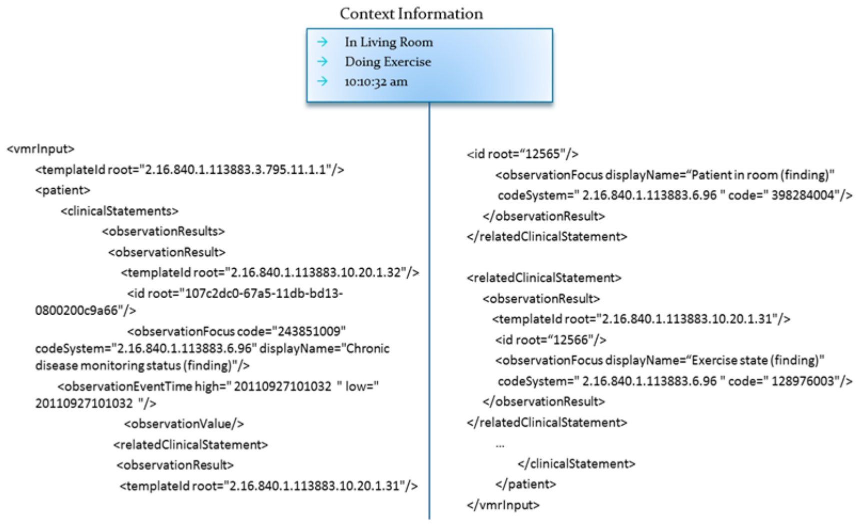


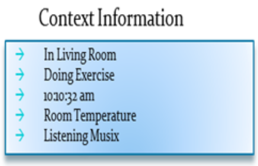
Figure 309 CAME VMR Format

- **VMR Data from Social Media**
  - The information from the social media shows the social activities that chronic disease patient is performing. The trajectory analysis shows that he has not gone outside for past few days. Twitter analysis shows that he is tweeting depression related tweets. Email interaction shows that he is not regularly in contact with his friends that depict a depression mode of the chronic disease patient.

```xml
<vmrInput>
        <templateId root="2.16.840.1.113883.3.795.11.1.1"/>
        <patient>
            <clinicalStatements><!--Health problem>
                        <problem><!--Depression -->
                            <id root="TW_107c2dc0-67a5-11db-bd13-0800200c9a66"/>
                            <problemCode codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED CT" code = "291746001"  />
                        </problem>
                        <problem><!-- diabetes-->
                            <id root="TW_10dca459-3249-29db-bd31-0800200c9a66"/>
                            <problemCode codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED CT" code = " 73211009 "/>
                        </problem>
                  <problem><!-- Abdominal weakness- ->
                            <id root="TR_107c2dc0-67a5-11db-bd13-0800200c9a66"/>
                            <problemCode codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED CT" code = "162239000  " />
                        </problem>
                        <problem><!-- Exercise assessment not completed- -->
                            <id root="TR_10dca459-3249-29db-bd31-0800200c9a66"/>
                            <problemCode codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED CT" code = " 413161009  "/>
                        </problem>
                  <problem><!--Muscle Fatigue-->
                            <id root="IN_107f2dc0-67a5-11db-bd13-0800200c9a66"/>
                            <problemCode codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED CT" code = "56946009"  />
                        </problem>
                        <problem><!-- Depression-->
                            <id root="IN_10fga459-3249-29db-bd31-0800200c9a66"/>
                            <problemCode codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED CT" code = " 291746001  "/>
                        </problem>
                  <problem> <!– Diabestes Type 2->
                            <id root="TDS_107c2dc0-67a5-11db-bd13-0800200c9a66"/>
                            <problemCode codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED  CT" code = "1626938751 " />
                        </problem>
                        <problem> <!– Take medicine Metformin -->
                            <id root="TDS_10dca459-3249-29db-bd31-0800200c9a66"/>
                            <problemCode codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED  CT" code = " 4131598685 "/>
                        </problem
  </clinicalStatement>
        </patient>
</vmrInput>
```


Figure 310 Social Media VMR Format

- **VMR Data from Behavioral Modeling Module**
  - In addition to the came engine, BMM also shows that behavior analysis of the chronic disease patient. The patient while doing exercise was also listening music and room temperature was normal.




```
<vmrInput>
  <templateId root="2.16.840.1.113883.3.795.11.1.1"/>
  <patient>
    <clinicalStatements>
      <observationResults>
        <observationResult>
          <templateId root="2.16.840.1.113883.10.20.1.32"/>
          <id root="107c2dc0-67a5-11db-bd13-
  0800200c9a66"/>....
          <observationFocus code="243851009"
  codeSystem="2.16.840.1.113883.6.96" displayName="Chronic
  disease monitoring status (finding)"/>
      <observationEventTime high=" 20110927101032 " low="
  20110927101032 "/>
          <observationValue/>
        <relatedClinicalStatement>
        <observationResult>
          <templateId root="2.16.840.1.113883.10.20.1.31"/>
<id root="12565"/>
    <observationFocus displayName="Room Temperature"
    codeSystem=" " code=" 1"/>
  </observationResult>
</relatedClinicalStatement>

<relatedClinicalStatement>
  <observationResult>
   <templateId root="2.16.840.1.113883.10.20.1.31"/>
    <id root="12566"/>
    <observationFocus displayName="Listening Music"
    codeSystem=" " code=" 2"/>
  </observationResult>
</relatedClinicalStatement>
    ...
      </clinicalStatement>
    </patient>
</vmrInput>
```


Figure 311 BMM VMR Format

- **VMR Data from Interoperability Engine**
  - Clinical information about the chronic disease patient is obtained through interoperability engine. This show the different symptoms of the patient when he visited hospital last time.

```
<vmrInput>
  <templateId root="2.16.840.1.113883.3.795.11.1.1"/>
  <patient>
    <clinicalStatements>
      ...
       <observationResult>
      <templateId root="2.16.840.1.113883.10.20.1.31"/>
      <id root="12564"/>
      <observationFocus displayName="FBS"
      codeSystem=" 2.16.840.1.113883.6.96 "
      code=" 18256789"/>
      <observationEventTime high=" 20110927101300+0300
      " low=" 20110927101300+0300 "/>
      <observationValue>
      <physicalQuantity value=" 150 " unit="mg/dl"/>
      </observationValue>
      </observationResult>
      <observationResult>
      <templateId root="2.16.840.1.113883.10.20.1.31"/>
      <id root="12564"/>
      <observationFocus displayName="pp2h"
      codeSystem=" 2.16.840.1.113883.6.96 "
      code=" 18256789"/>
      <observationEventTime high=" 20110927101300+0300
      " low=" 20110927101300+0300 "/>
      <observationValue>
      <physicalQuantity value=" 210" unit="mg/dl"/>
      </observationValue>
      </observationResult>

      <observationResult>
      <templateId root="2.16.840.1.113883.10.20.1.31"/>
      <id root="12564"/>
      <observationFocus displayName="Cholestrol (finding)"
      codeSystem=" 2.16.840.1.113883.6.96 "
      code=" 168077005"/>
      <observationEventTime high=" 20110927101300+0300
      " low=" 20110927101300+0300 "/>
      <observationValue>
      <physicalQuantity value=" 210 " unit="mg/dl"/>
      </observationValue>
      </observationResult>
      ...
    </clinicalStatement>
  </patient>
</vmrInput>
```


Figure 312 Interoperability Engine VMR Format

- **VMR Data from Activity and Emotion Recognition**
  - AER Engine identifies by monitoring the chronic disease patient sleep patterns and finds that the patient is not getting proper sleep.

```
<vmrInput>
    <templateId root="2.16.840.1.113883.3.795.11.1.1"/>
    <patient>
        <clinicalStatements>
            <observationResults>
             <observationResult>
               <templateId root="2.16.840.1.113883.10.20.1.32"/>
                <id root="107c2dc0-67a5-11db-bd13-
0800200c9a66"/>....
                <observationFocus code="243851009"
codeSystem="2.16.840.1.113883.6.96" displayName="Chronic
disease monitoring status (finding)"/>
      <observationEventTime high=" 20110927101032 " low="
20110927101032 "/>
                <observationValue/>
              <relatedClinicalStatement>
              <observationResult>
               <templateId root="2.16.840.1.113883.10.20.1.31"/>
<id root="12565"/>
      <observationFocus displayName="Sleep Monitoring"
      codeSystem=" " code=" 1"/>
<observationValue> Not Proper <observationValue>
   </observationResult>
</relatedClinicalStatement>

      ...
         </clinicalStatement>
      </patient>
</vmrInput>
```


Figure 313 AER VMR Format

- **Concatenated VMR**
  - Finally the VMR's are concatenated by the Fusion Adapter engine, to generate comprehensive VMR for Knowledge inference engine to process for recommendation.

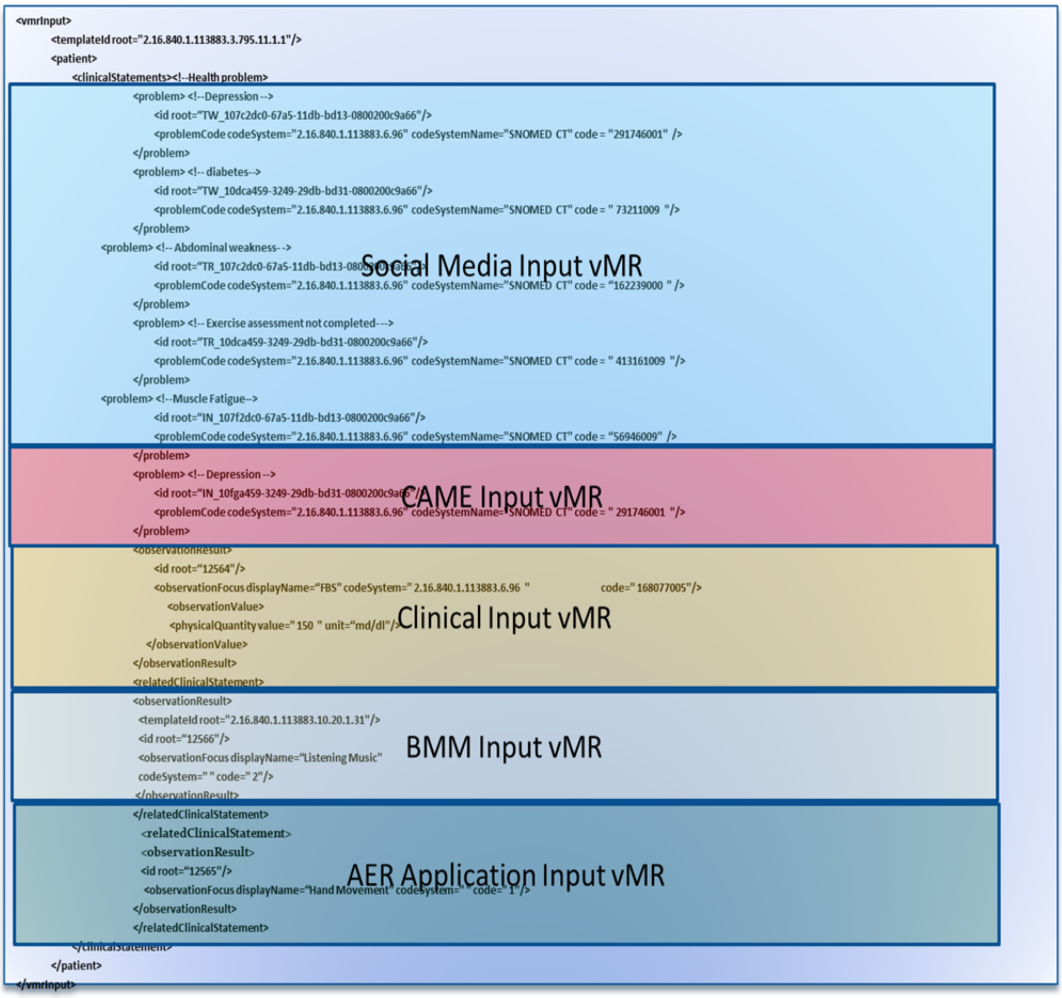


Figure 314 Concatenated VMR

- **Final Recommendation**
  - Based on the concatenated VMR, a final recommendation is generated and provided to the user. This recommendation shows that patient is Diabetes Mellitus patient, having problem in cholesterol level, suffering from depression nowadays and sleep pattern is not proper.

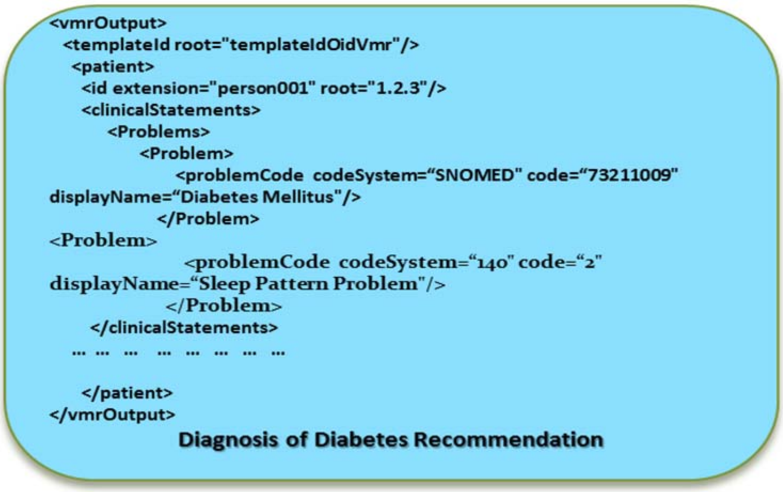


Figure 315 Final Recommendation

7.4. Application Scenario: SmartCDSS for Head & Neck Cancer

- Smart CDSS for Head & Neck cancer targeting two areas of treatments; Finding appropriate staging by incorporating TNM staging guidelines and providing appropriate treatment plan based on staging and patient conditions. Treatment plan recommendations come from NCCN guidelines and oncologist's experiences published to knowledge base.
- Smart CDSS interventions are incorporating during staging, treatment plan and re-evaluation of the patient for further treatment. Following figure shows, the encounter of Head & Neck cancer patient during diagnosis and treatment management.
- In this application scenario, staging intervention is described in detail through invocation of various components of Smart CDSS.

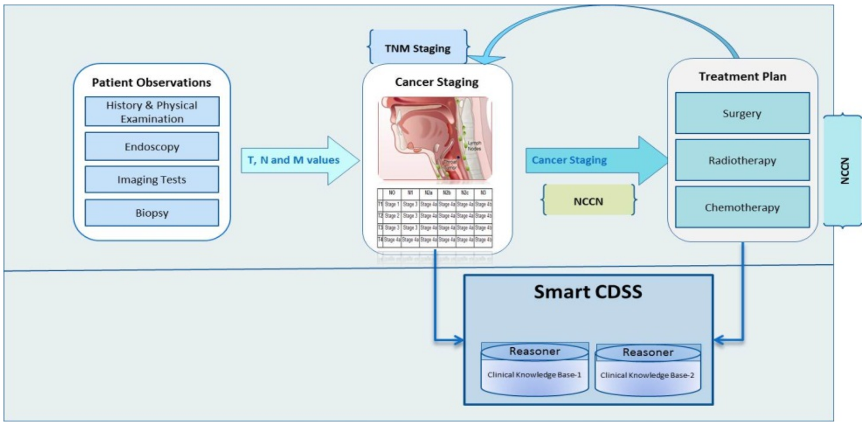


Figure 316 Head & Neck Cancer Diagnosis and Smart CDSS Interventions

- **Staging Intervention event at HMIS**
  - During patient encounter for diagnosing tumor, staging is needed to find out for further treatment plan. At Hospital Management Information System (HMIS), the staging event cause to invoke Smart CDSS. The HMIS interact with Smart CDSS via Adaptability Engine by providing patient symptoms in HL7 CDA format. Considering patient symptoms for T=2, N=0 and M=0, the HMIS pass following CDA to Smart CDSS.

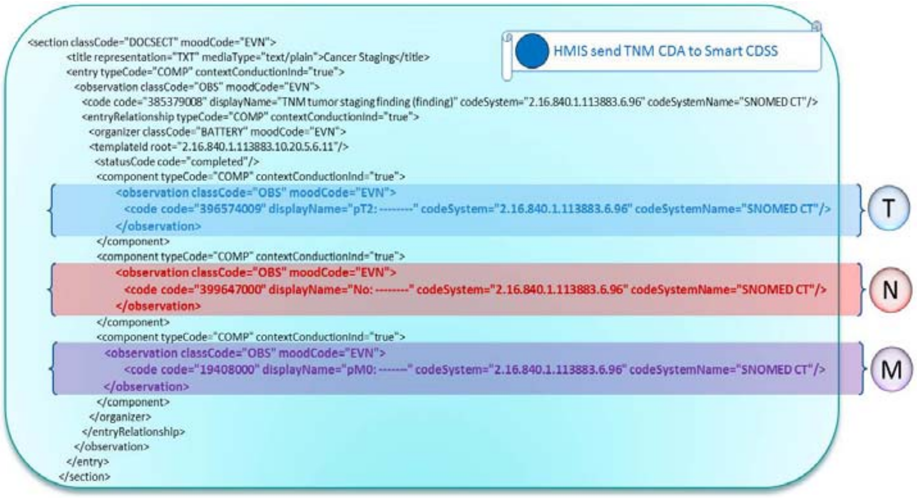


Figure 317 Standard CDA for TNM Observations

- **HMIS Input Transformation into Smart CDSS Standard Input**
  - The TNM CDA coming from HMIS is transformed into Smart CDSS standard HL7 vMR format by Adapter Interoperability Engine. Following diagram shows the transformed HL7 vMR format of TNM CDA.

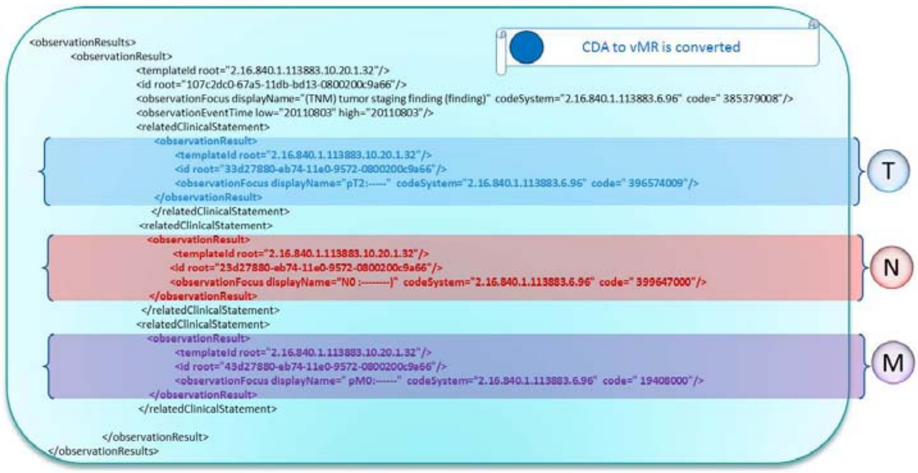


Figure 318 Standard vMR Input for TNM Staging

- **Knowledge base execution**
  - Knowledge broker select knowledge base upon event type and reasoner schedule appropriate rules (which is called MLM: Medical Logic Module) to execute on input data set. For TNM staging event, following MLM will be scheduled to execute.

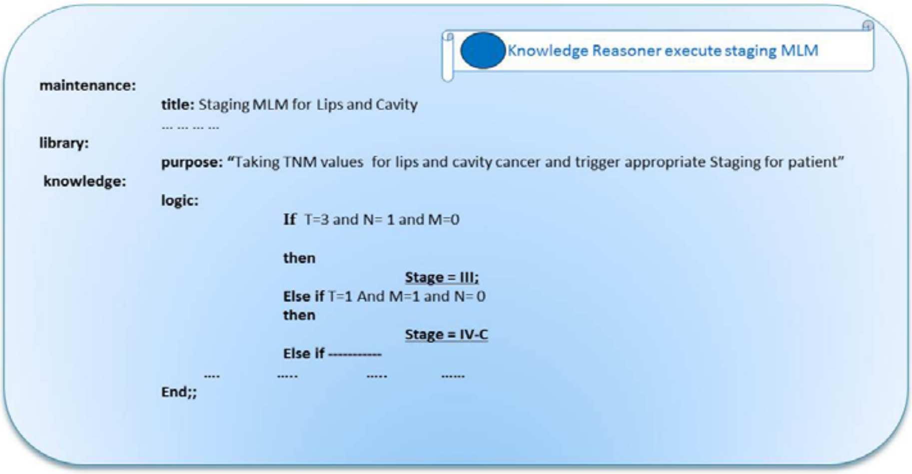


Figure 319 Sample of Arden Syntax MLM for TNM Staging

- **Staging Recommendation**
  - The TNM MLM provides stage II for lips and cavity for TNM values provided in input. The corresponding output is converted into standard vMR output format and sends back to caller application.

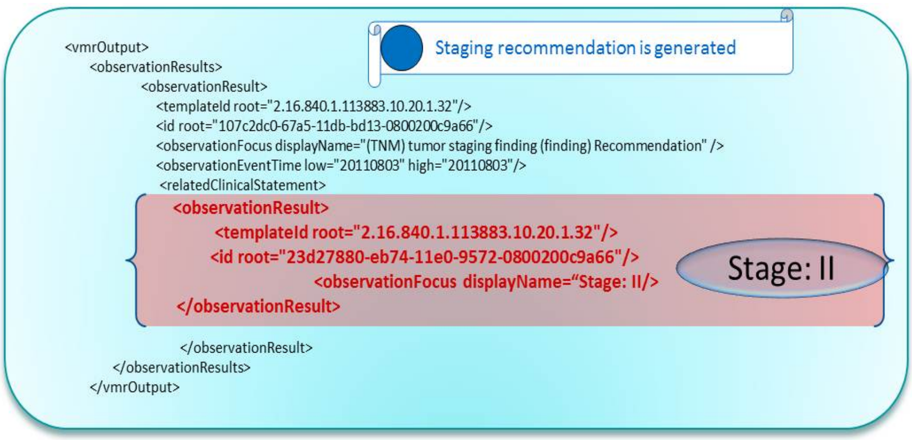


Figure 320 TNM Recommendation: Standard vMR output

REFERENCES

8.1. References

[Abel2011a] F. Abel, Q. Gao, G. Houben, and K. Tao, “Analyzing user modeling on twitter for personalized news recommendations,” User Modeling, Adaption and Personalization, pp. 1–12, 2011.

[Abel2011b] F. Abel, Q. Gao, G.-J. Houben, and K. Tao., “Semantic enrichment of twitter posts for user profile construction on the social web,” The Semantic Web: Research and Applications, pp. 375–389, 2011.

[Abel2011c] F. Abel, Q. Gao, G. Houben, and K. Tao, “Analyzing temporal dynamics in twitter profiles for personalized recommendations in the social web,” in Proceedings of ACM WebSci ’11, 3rd International Conference on Web Science. ACM, 2011.

[Ahtinen2009] Ahtinen, A., et al., Designing social features for mobile and ubiquitous wellness applications. Proceedings of the 8th International Conference on Mobile and Ubiquitous Multimedia. Cambridge, United Kingdom, ACM: 1-10, 2009.

[Amazon] http://www.informationweek.com/news/services/storage/showArticle.jhtml?articleID=212201778&subSection=Security

[Archivist2012] “Archivist,” (Last visited in May 2012). [Online]. Available: http://archivist.visitmix.com/

[Ayadi2011] Ayadi, M.E., Kamel, M.S., Karray, F.: Survey on speech emotion recognition: Features, classification schemes, and databases. Pattern Recognition 44 (3), 572 - 587 (2011).

[Baker1999] Mark Baker, Bryan Carpenter, Geoffrey Fox, Sung Hoon Ko, and Sang Lim. An Object-Oriented Java interface to MPI. In Proceedings of the International Work-shop on Java for Parallel and Distributed Computing, San Juan, Puerto Rico, April 1999.

[Bamse1996] Banse R. Scherer, K.R.: Acoustic profiles in vocal emotion expression. Journal of Personality and Social Psychology 70(3), 614 - 636 (1996).

[Barabási2002] Barabási AL, Jeong H, Neda Z, Ravasz E, Schubert A, Vicsek T., Evolution of the social network of scientific collaborations. Physica A: Stat Mech Appl 311(3–4):590–614, 2002.

[Bartlett1999] M.S. Bartlett, J.C. Hager, P. Ekman, and T.J. Sejnowski, “Measuring facial expressions by computer image analysis,” *Psychophysiology*, vol. 36, pp. 253 – 263, 1999.

[Bartlett2002] M. S. Bartlett, J. R. Movellan, and T. J. Sejnowski, “Face Recognition by Independent Component Analysis," *IEEE Trans. Neural Networks*, vol. 13, no. 6, pp. 1450 – 1464, 2002.

[Bettadapura2009] V.K. Bettadapura, "Face expression recognition and analysis : The state of the art," *Emotion*, pp. 1–27, 2009.

[Bitouk2010] Bitouk, D., Verma, R., Nenkova, A.: Class-level spectral features for emotion recognition. Speech Communication 52(7-8), 613 - 625 (2010).

[Buciu2003] I. Buciu, C. Kotropoulos, and I. Pitas, "ICA and Gabor Representation for Facial Expression Recognition," *Proc. Int. Conference on Image Processing*, pp. 855 – 858, 2003.

[Buciu2009] I. Buciu., C. Kotropoulos., I. Pitas. "Comparison of ICA approaches for facial expression recognition." *SIViP*, vol. 3, no. 4, pp.345 – 361, 2009.

[Carpenter1999] Bryan Carpenter, Geoffery Fox, Sung-Hoon Ko, and Sang Lim. mpiJava 1.2: API Specification. Technical report, Northeast Parallel Architectures Center, Syracuse University, October 1999.

[Carpenter2000] Bryan Carpenter, Vladimir Getov, Glenn Judd, Tony Skjellum, and Geoffrey Fox. MPJ: MPI-like Message Passing for Java. Concurrency: Practice and Experience, 12(11), 2000.

[Castain2005]. R. H. Castain, T. S. Woodall, D. J. Daniel. "The Open Run-Time Environment (OpenRTE): A Transparent Multi-Cluster Environment for High-Performance ting". Euro PVM/MPI 2005, September, 2005.

[Celik2011] I. Celik, F. Abel, and G. Houben, "Learning semantic relationships between entities in twitter," Web Engineering, pp. 167–181, 2011.

[Chan2001] T. Chan., L. Vese. "Active contours without edges." *IEEE Trans. Image Proc*., vol. 10, pp. 266 – 77, 2001.

[Chapanond2005] Chapanond A, Krishnamoorthy MS, Yener B, Graph theoretic and spectral analysis of Enron email data. Comput Math Organ Theory 11(3):265–281,2005.

[Chen2010] J. Chen, R. Nairn, L. Nelson, M. Bernstein, and E. Chi, "Short and tweet: experiments on recommending content from information streams," in Proceedings of the 28th international conference on Human factors in computing systems. ACM, pp. 1185–1194, 2010.

[Chen2012] L. Chen, C. Nugent, H. Wang, "A Knowledge-Driven Approach to Activity Recognition in Smart Homes," Knowledge and Data Engineering, IEEE Transactions on, vol. 24, no 6, pp. 961-972, 2012.

[Chuang2006] C. F. Chuang, and F. Y. Shih, "Recognizing facial action units using independent component analysis and support vector machine," *Pattern Recognition*., vol.39, no. 9, pp. 1795 – 1798, 2006.

[Chu2009] Chu, S., Narayanan, S., Kuo, C.-C.: Environmental sound recognition with time-frequency audio features. Audio, Speech, and Language Processing, IEEE Transactions on 17 (6), 1142-1158 (2009).

[Clark2008] J. Clark, “Twitter Venn,” (Last visited in May 2012). [Online]. Available: http://www.neoformix.com/2008/TwitterVenn.html

[Clark2011] E. Clark and K. Araki, “Text normalization in social media: Progress, problems and applications for a pre-processing system of casual English,” Procedia-Social and Behavioral Sciences, vol. 27, pp. 2–11, 2011.

[Coll2011] Navig. Coll., Dalian Maritime Univ., Dalian, China “2nd IEEE International Conference on Emergency Management and Management Sciences (ICEMMS)”, 2011.

[Cowie2001] Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Votsis, G., Kollias, S., Fellenz, W., Taylor, J.: Emotion recognition in human-computer interaction. IEEE Signal Processing Magazine 18(1), 32 - 80 (2001).

[Dante2010] I. Dante, A. Tapia, A. Fraile, and B. Javier, "Healthcare Information Fusion Using Context-Aware Agents", 5th International Conference on Hybrid Artificial Intelligence Systems. 2010.

[Diesner2005] Diesner J, Carley KM, Exploration of communication networks from the Enron Email corpus. In: Proceedings of the 2005 SIAM workshop on link analysis, counterterrorism and security, pp 3–14, 2005.

[Dogac2006] A. Dogac, G. Laleci, S. Kirbas, Y. Kabak, S. Sinir, A. Yildiz, and Y. Gurcan. Artemis: deploying semantically enriched web services in the healthcare domain. Journal of Information Systems, 31(4-5):321{339, 2006.

[Doulamis1999] A. D. Doulamis., N. D. Doulamis., K. S. Ntalianis., S. D. Kollian. “Unsupervised Semantic Object Segmentation of Stereoscopic Video Sequences.” *Proc. of IEEE Intl. Conf. on Intelligence, Information and Systems*, Washington D.C, USA, 1999.

[Du2008] K. Du, D. Zhang, X. Zhou, M. Mokhtari, M. Hariz, and W. Qin. “HYCARE: A hybrid context-aware reminding framework for elders with mild dementia”, ICOST 2008.

[Ducrou2009] A. Ducrou. Complete interoperability in healthcare: technical, semantic and process interoperability through ontology mapping and distributed enterprise integration techniques. Doctor of Philosophy Thesis, 2009.

[Eagle2006] Eagle N, Pentland A., Reality mining: sensing complex social systems. Personal and Ubiquitous Comput 10(4):255–268, 2006.

[Elger1998] G. Elger, B. Furugren, “SmartBo an ICT and computer based demonstration home for disabled, in improving the quality of life for European citizen,” Technology for inclusive Design and Equality Assistive Technology Research Series, vol. 4, pp. 392 – 395, 1998.

[Faloutsos1999] Faloutsos M, Faloutsos P, Faloutsos C On power-law relationships of the internet topology. In: Proceedings of the conference on applications, technologies, architectures, and protocols for computer communication ACM, pp 251–262, 1999.

[Fatima2011] I. Fatima, M. Fahim, D. Guan, Y. K. Lee, and S. Y. Lee, "Socially Interactive CDSS for u-Life Care", The 5th ACM International Conference on Ubiquitous Information Management and Communication (ACM ICUIMC 2011), Seoul, Korea, February, 2011.

[Fischhoff2007] Fischhoff IR, Sundaresan SR, Cordingley J, Larkin HM, Sellier M-J, Rubenstein DI Social relationships and reproductive state influence leadership roles in movements of Plains zebra, Equus burchellii. Anim Behav 73(5):825–831, 2007.

[Gao2008] T. Gao, C. Pesto, L. Selavo, Y. Chen, J. Ko, J. Lim, A. Terzis, A. Watt, J. Jeng, B. Chen, K. Lorincz, M. Welsh, “Wireless Medical Sensor Networks in Emergency Response: Implementation and Pilot Results”, The 2008 IEEE International Conference on Technologies for Homeland Security, May, 2008.

[Georis2004] B. Georis, M. Maziere, F. Bremond, and M. Thonnat, “A Video Interpretation Platform Applied to Bank Agency Monitoring,” *Proc. of 2nd Workshop on Intelligent Distributed Surveillance Systems (IDSS)*, 2004.

[Gobl2003] Gobl, C., Chasaide, A.N.: The role of voice quality in communicating emotion, mood and attitude. Speech Communication 40, 189 - 212 (2003).

[Godbole2007] N. Godbole, M. Srinivasaiah, and S. Skiena, “Large-scale sentiment analysis for news and blogs,” in Proceedings of the International Conference on Weblogs and Social Media (ICWSM), pp. 219–222, 2007.

[Google] Google Health https://www.google.com/health

[Gu2009] T. Gu., Z. Wu., X. Tao., H. K. Pung., J. Lu. “epSICAR: An Emerging Patterns based approach to sequential, interleaved and Concurrent Activity Recognition.” *Proc. of IEEE Intl. Conference on Pervasive Computing and Communications*, 2009.

[Hayley2011] Hayley Hung, Yan Huang, Gerald Friedland, Daniel Gatica-Perez: Estimating Dominance in Multi-Party Meetings Using Speaker Diarization. IEEE Transactions on Audio, Speech & Language Processing 19(4): 847-860, 2011.

[Henr2004] K. Henricksen, and J. Indulska, “Modelling and Using Imperfect Context Information”. In Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, Washington DC, March 14 - 17, 2004.

[Huang1997] C.-L. Huang, and Y.-M. Huang, “Facial expression recognition using model-based feature extraction and action parameters classification,” *J. Vis. Comm. Image Repress.*, vol. 8, no. 3, pp. 278 – 290, 1997.

[Huang2007] W. Huang., J. Zhang., Z. Liu, Z. “Activity recognition based on hidden markov models.” *Proc. of 2nd Int. Conf. Knowledge Science, Engineering Management*, Melbourne, pp. 532 – 537, 2007.

[Huang2011] D. Huang, C. Shan, M. Ardabilian, Y. Wang, and L. Chen, “Local binary patterns and its application to facial image analysis: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vo. 41, no. 6, pp. 765 – 781, 2011.

[Hussain2010] Husain, W. and P. Lim Thean, The development of personalized wellness therapy recommender system using hybrid case-based reasoning. Computer Technology and Development (ICCTD), 2nd International Conference on, 2010.

[Iliev2010] Iliev, A.I., Scordilis, M.S., Papa, J.P., Falco, A.X.: Spoken emotion recognition through optimum-path forest classification using glottal features. Computer Speech and Language 24(3), 445 - 460 (2010).

[Jabeed2010] T. Jabeed, M. H. Kabir, and O. Chae, "Robust Facial Expression Recognition Based on Local Directional Pattern," *ETRI Journal*, vol. 32, no. 5, pp. 784 – 794, 2010.

[Jabid2010] T. Jabid, H. Kabir, and O. Chae, "Local Directional Pattern for Face Recognition," *Proc. IEEE International Conference on Consumer Electronics*, NV, pp. 329 – 330, 2010.

[Jansen2009] B. Jansen, M. Zhang, K. Sobel, and A. Chowdury, "Twitter power: Tweets as electronic word of mouth," Journal of the American society for information science and technology, vol. 60, no. 11, pp. 2169–2188, 2009.

[Jiao2006] L. Jiao, Liu J., Zhong W.: An organizational co evolutionary algorithm for classification. IEEE Transactions on Evolutionary Computation, 2006.

[Juang2002] Juang P, Oki H, Wang Y, Martonosi M, Peh LS, Rubenstein DI., Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet. ACM SIGPLAN Notices 37(10):96–107, 2002.

[KEA] http://www.nzdl.org/Kea/

[Kass1988] M. Kass., A. Witkin., D. Terzopoulos. "Snake: Active Contour Models." *Intl. J. Comput. Vision*, vol. 1, no. 14, pp. 321 – 331, 1988.

[Khan2009] W. A. Khan, M. Hussain, M. Afzal, K. Latif, H. F. Ahmad, and A. Khattak. Towards semantic process interoperability. In 10th International HL7 Interoperability Conference (IHIC), pages 88{95, Kyoto, Japan, May 2009.

[Khattak2010] A. M. Khattak, L. T. Vinh, D. V. Hung, P. T. H. Truc, L. X. Hung, D. Guan, Z. Pervez, M. Han, S. Y. Lee and Y. K. Lee, "Context-aware Human Activity Recognition and Decision Making", 12th International Conference on e-Health Networking, Application Services(IEEE HealthCom 2010), Lyon, France, July 1-3, 2010.

[Khattak2011] A. M. Khattak, Z. Pervez, S. Y. Lee, and Y. K. Lee. "Intelligent Healthcare Service Provisioning Using Ontology with Low-Level Sensory Data". KSII Transaction on Internet and Information Systems, vol. 5, no. 11, November 2011.

[Kim2009] T.-S. Kim, and J. J. Lee, "Human Facial Expression Recognition using Fisher Independent Component Analysis and Hidden Markov Model," *Biomedical Engineering*, pp. 627 – 644. Carlos Alexandre Barros de Mello (Ed.), ISBN: 978-953-307-013-1, InTech, 2009.

[Kim2010] T.-S. Kim., M. Z. Uddin. “Silhouette-based Human Activity Recognition Using Independent Component Analysis, Linear Discriminant Analysis and Hidden Markov Model.” *New Developments in Biomedical Engineering*, ISBN: 978-953-7619-57-2. Edited by: Domenico Campolo. Published by InTech, 2010.

[Klein2002] J. Klein, Y. Moon, and R. Picard. This computer responds to user frustration: Theory, design, and results. Interacting with Computers, 14:119-140, February 2002.

[Korea2008] Korea u-Life care system (to be updated in Korean version)

[Kulkarni2011] R. V. Kulkarni, A.F Orster and G. K. Venaya, "Computational Intelligence in Wireless Sensor Networks: A Survey", IEEE Communications Surveys & Tutorial, vol. 13. 2011.

[Lahiri2010] M. Lahiri and T.Y. Berger-Wolf. Periodic subgraph mining in dynamic networks. Knowledge and Information Systems, Volume 24, Issue 3, p. 467, 2010.

[Lee2005] Lee, C.M., Narayanan, S.: Towards detecting emotions in spoken dialogs. IEEE Transactions on Speech and Audio Processing 13, 293 - 303 (2005).

[Lionel2011] N. Lionel, “Internet of Healthcare Things” Proceedings of the 2nd SMCEW international workshop, Korea, October, 2011.

[Little2008] L. Little, and P. Briggs, “Ubiquitous Healthcare: Do we want it?” Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction-Volume 2, 2008.

[Liwc2012] “Liwc,” (Last visited in May 2012). [Online]. Available: http://www.liwc.net/

[Lo2011] Chi-Chun Lo et. al. Ubiquitous Healthcare Service System with Context-awareness Capability: Design and Implementation. Elsevier journal of Expert Systems with Applications vol. 38 pp. 4416-443, 2011.

[Magni2007] A. Magni, R. de Oliveira Albuquerque, R. de Sousa Jr, M. Hans, and F. Magni. Solving incompatibilities between electronic records for orthodontic patients. American Journal of Orthodontics and Dentofacial Orthopedics, 132(1):116{121, 2007.

[Mama2000] N. Marmasse, C. Schmandt, “Location-aware information delivery with commotion”. The 2nd International symposium on Handheld and Ubiquitous Computing, Bristol, England, pp. 157–171 (2000).

[Malika2009] Malika Meghjani: http://www.cim.mcgill.ca/~malika/Publications/Meghjani09_Masters_Thesis.pdf

[Mallat1993] Mallat, S. G., Zhang, Z.: Matching pursuit with time-frequency dictionaries. IEEE Transactions on Signal Processing 41 (12), 3397 – 3415 (1993).

[Mehrabian1968] A. Mehrabian, “Communication without Words,” Psychology Today, vo1.2, no.4, pp. 53 – 56, 1968.

[Microsoft] Microsoft HealthVault http://healthvault.com

[Mohktar2010] Mohktar, M. S., et al.., A guideline-based decision support system for generating referral recommendations from routinely recorded home telehealth measurement data. Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE, 2010.

[Nanavati2006] Nanavati AA, Gurumurthy S, Das G, Chakraborty D, Dasgupta K, Mukherjea S, Joshi A., On the structural properties of massive telecom call graphs: findings and implications. In: Proceedings of the 15th ACM international conference on Information and knowledge management (New York, NY, USA, 2006). ACM, pp 435–444, 2006.

[Nasukawa2003] T. Nasukawa and J. Yi, "Sentiment analysis: Capturing favorability using natural language processing," in Proceedings of the 2nd international conference on Knowledge capture. ACM, pp. 70–77, 2003.

[Newman2001] Newman MEJ (2001) From the cover: the structure of scientific collaboration networks. Proc Natl Acad Sci 98:404–409, 2001.

[Nwe2003] Nwe, T.L., Foo, S.W., Silva, L.C.D.: Speech emotion recognition using Hidden Markov Models. Speech Communication 41(4), 603 - 623 (2003).

[Paparo2005] Paparo, M., Automated Wellness System Managing, Personalized Fitness Programs, United States Patent, US, 2005.

[Pradeep2010] K. A. Pradeep, M. H. Anwar, A.E. Saddik, S.K. Mohan, "Multimodal fusion for multimedia analysis: a survey", Journal of Multimedia System, pp. 345-379, 2010.

[Prendinger2006] H. Prendinger, C. Becker, and M. Ishizuka. A study in users' physiological response to an empathic interface agent. International Journal of Humanoid Robotics, 3(3):371-391, September 2006.

[Pugh2003] W. Pugh, and J. Spacoo, MPJava: High-Performance Message Passing in Java using Java.nio, Workshops on Languages and Compilers for Parallel Computing, Dept. of Computer Science, University of Texas at Austin, October 2003.

[Puig2008] A. O. Puig, Mansilla E. B. Evolutionary rule-based systems for imbalanced data sets. In Journal of Soft Computing - A Fusion of Foundations, Methodologies and Applications. 2008.

[Robertson2006] N. Robertson., I. Reid. "A general method for human activity recognition in video." *Comput. Vis. Image Und.*, vol. 104, no, 2, pp. 232 – 248, 2006.

[Rogova2004] G. L. Rogova & V. Nimier, "Reliability in Information Fusion: Literature Survey", The 7th International Conference on Information Fusion. 2004.

[Sahay2008] R. Sahay, W. Akhtar, and R. Fox. Ppepr: plug and play electronic patient records. In

Proceedings of the 2008 ACM symposium on Applied computing, pages 2298{2304. ACM, 2008.

[Schels2010] M. Schels, and F. Schwenker, “A multiple classifier system approach for facial expressions in image sequences utilizing GMM supervectors,” *Proc. International Conference on Pattern Recognition (ICPR)*. pp. 4251 – 4254, 2010.

[Schuller2004] Schuller, B., Rigoll, G., Lang, M.: Speech emotion recognition combining acoustic features and linguistic information in a hybrid support vector machine-belief network architecture. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 1, pp. 577 - 580 (2004).

[SecureCloud] http://cloudsecurity.org/

[Shan2009] C. Shan, S. Gong, and P. McOwan, “Facial expression recognition based on local binary patterns: A comprehensive study,” *Image and Vision Computing*, vol. 27, no. 6, pp. 803 – 816, 2009.

[Siddiqi2010] M. H. Siddiqi., M. Fahim., S. Y. Lee., Y.-K. Lee. “Human Activity Recognition Based on Morphological Dilation followed by Watershed Transformation Method.” *Proc. of International Conference on Electronics and Information Engineering (ICEIE)*, pp. V2 433 – 437, 2010.

[Sohn2005] T. Sohn, K. Li, G. Lee, I. Smith, J. Scott, W. Griswold, “Place-its: A study of location-based reminders on mobile phones”. In: Ubicomp, pp. 232–250, 2005.

[Sundaresan2007] Sundaresan SR, Fischhoff IR, Dushoff J, Rubenstein DI, Network metrics reveal differences in social organization between two fission–fusion species, Grevys zebra and onager. Oecologia 151(1):140–14, 2007.

[Tacconi2008] Tacconi, D., Mayora, O., Lukowicz, P., Arnrich, B., Setz, C., Troster, G., Haring, C.: Activity and emotion recognition to support early diagnosis of psychiatric diseases. In: Proceedings of the Second International Conference on Pervasive Computing Technologies for Healthcare, pp. 100 - 102 (2008).

[Tian2005] Y. L. Tian, T. Kanade, and J. F. Cohn, “Facial Expression Analysis,” *Handbook of Face Recognition, S.Z. Li and A.K. Jain*, eds. pp. 247 – 276, 2005.

[Tumasjan2010] A. Tumasjan, T. Sprenger, P. Sandner, and I. Welpe, “Predicting elections with twitter: What 140 characters reveal about political sentiment,” in Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media, pp. 178–185, 2010.

[Turaga2008] P. Turaga, R. Chellappa, V. S. Subrahmanian, O. Udrea, “Machine Recognition of Human Activities: A Survey”. In IEEE Transaction on Circuits and Systems for Video Technology, vol.18, no.11, 2008.

[Uddin2008] M. Z. Uddin., J .J. Lee., T.-S. Kim. “Shape-Based Human Activity Recognition Using Independent Component Analysis and Hidden Markov Model.” *Proc. of 2nd International Conference on Industrial, Engineering, and other Applications of Applied Intelligent*

Systems, pp.245 – 254, 2008.

[Uddin2009] M.Z. Uddin, J.J. Lee, and T.-S. Kim, "An enhanced independent component-based human facial expression recognition from video," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 4, pp. 2216 – 2224, 2009.

[Uddin2010] M. Z. Uddin., J. J. Lee., T.-S. Kim. "Independent shape component-based human activity recognition via Hidden Markov Model." *Appl. Intell*, vol. 33, no, 2, pp. 193 – 206, 2010.

[Umapathy2005a] Umapathy, K., Krishnan, S., Jimaa, S.: Multigroup classification of audio signals using time-frequency parameters. Multimedia, IEEE Transactions on 7 (2), 308 – 315 (2005).

[Umapathy2005b] Umapathy, K., Krishnan, S., Parsa, V., Jamieson, D., march 2005b. Discrimination of pathological voices using a time-frequency approach. Biomedical Engineering, IEEE Transactions on 52 (3), 421-430 (2005).

[UnifiedCloud] Unified Cloud Interface Standardization http://code.google.com/p/unifiedcloud/

[WKhan2012] Khan W, Khattak A, Lee S, Hussain M, Amin B, Latif K: Achieving interoperability among healthcare standards: building semantic mappings at models level. In Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication, ACM 101, 2012.

[Wang2009] F. Wang and K. J. Turner. "An Ontology-Based Actuator Discovery and Invocation Framework in Home Care Systems," 7th International Conference on Smart Homes and Health Telematics, pp. 66-73, LNCS 5597, Springer, Berlin, June 2009.

[Wu2010] X. Wu, and J. Zhao, J. "Curvelet feature extraction for facerecognition and facial expression recognition," *Proc.6th International Conference on Natural Computation*, pp. 1212 – 1216, 2010.

[Xue2010] Z. Xue., D. Ming., W. Song., B. Wan., S. Jin. "Infrared gait recognition based on wavelet transform and support vector machine. "*Pattern Recognition*, vol. 43, no. 8, pp. 2904 – 2910, 2010.

[Yi2003] J. Yi, T. Nasukawa, R. Bunescu, and W. Niblack, "Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques," in Data Mining, 2003. ICDM 2003. Third IEEE International Conference on. IEEE, pp. 427–434, 2003.

[Yue2009] Yang Yue,Yan Zhuang,Qingquan Li,Qingzhou Mao "17th International Conference on Geoinformatics", 2009.

APPENDIX

9.1. Appendix

All the slides for SC^3 and its services are listed below.