Smart CDSS for Head and Neck Cancer

Technical Report V (2.0)

By

E-Health and Standardization Team

Ubiquitous Computing Lab

Kyung Hee University

Korea

Authors
Maqbool Hussain (maqbool.hussain@oslab.khu.ac.kr)
Muhammad Afzal (muhammad.afzal@oslab.khu.ac.kr)
Taqdir Ali (taqdir.ali@oslab.khu.ac.kr)
Wajahat Ali Khan (wajahat.alikhan@oslab.khu.ac.kr)

Supervised by
Professor Sungyoung Lee (sylee@oslab.khu.ac.kr)

Professor Byeong Ho Kang (byeong.kang@utas.edu.au)

Document History	
Technical Report v (2.0)	December, 2013
Technical Report v (1.0)	December, 2012

Table of Contents

CHAPTER 1: INTRODUCTION - Smart CDSS for Head and Neck Cancer

1.1 Research Motivation

The rise in living standards of people with the advancement of new technology has also raised the importance of sophisticated standard based healthcare applications and services anytime, anywhere, having low cost. To achieve this objective we have designed and developed a system called Smart Clinical Decision Support System (Smart CDSS) that takes input from clinical knowledge base, with medical experts knowledge authoring to generate standards based personalized recommendations. Initially, Smart CDSS covers to capture the knowledge of Head and Neck Cancer (HNC) as we have the support of clinicians available in this area. However, the design of the system is flexible to extend for other domains with minimal efforts.

1.2 Introduction

The rise in living standards that has occurred with the advancement of new technology has increased the demand for sophisticated health-care applications and services. This has led to the emergence of information and communication technology (ICT)–based clinical decision support systems (CDSS) and online health-care (e-health) applications, systems, and services. To decrease their associated costs, the US federal government is investing $27 billion in health information technology (HIT) under the American Recovery and Reinvestment Act of 2009 [Blumenthal2010]. This huge investment was targeted to adopt electronic health records (EHR) at each level of care with meaningful use of HIT—the "meaningful use" criteria of EHR. It has been revealed that health-care costs could be reduced if HIT reduces expensive adverse events. In recent studies, researchers have determined that CDSS support in EHR produces the best return on investment for providers, as most of the meaningful-use criteria target these functionalities [Kern2012]. Moreover, various experiments have shown that EHR can improve patient care, reduce errors, and reduce time, if properly equipped with clinical decision support services [Wright2009].

With the evolution of architectures and rapidly changing requirements, CDSS has also evolved from a stand-alone to a service-based architecture [Wright2008]. Although CDSS reduces costs of health care, improves patient care, and reduces errors and omissions, it remains a nightmare to widely implement the clinical decision support capabilities. According to a study of commercially available CDSS systems, inconsistent capabilities have been observed, even though each claims to fulfill the CCHIT criteria. Additionally, the clinical information system has significant limitations in allowing pluggable CDS services within complex workflows [Wright2009].

The most prominent challenges for adoption of CDSS capabilities are improvement of existing CDS interventions, creation of new CDS interventions, and dissemination of CDS knowledge and interventions [Sittig2008]. At a more granular level, these challenges can be tackled by building a shareable knowledge base. Additionally, pluggable interfaces are required for seamless integration and unified interfaces through the Internet are needed for merging the various knowledge bases of different domains. Further needed refinements include prioritizing and avoiding duplications of alerts and recommendations and improving human computer interfaces (HCI) [Sittig2008].

To overcome the shortcomings of existing approaches for CDSS, we propose Smart CDSS architecture, which provides a hybrid approach combining some existing systems. To enable Smart CDSS to have a sharable knowledge base, HL7 Arden syntax is used, and to cope with the heterogeneity of existing clinical systems, HL7 vMR is incorporated in conjunction with HL7 GELLO. Therefore, reducing cost and making the system accessible via the Internet while maintaining the privacy of patient data, Smart CDSS is deployed as a Microsoft WCF service on a public cloud infrastructure Microsoft Azure, with fabricated anonymization service.

1.3 Problem Statement

In clinical domain, most of the times physicians face situations where they need more help to reach to a final clinical decision. They find answers using multiple ways such as books, online resources or contact with expert physicians in the domain. In cancer domain, such kind of support is often required due to its fast evolving nature. A well designed system with appropriate integration of associated components is required to help in solving the issue of support in clinical decisions with reduced costs and minimized time.

1.4 Necessity of Research

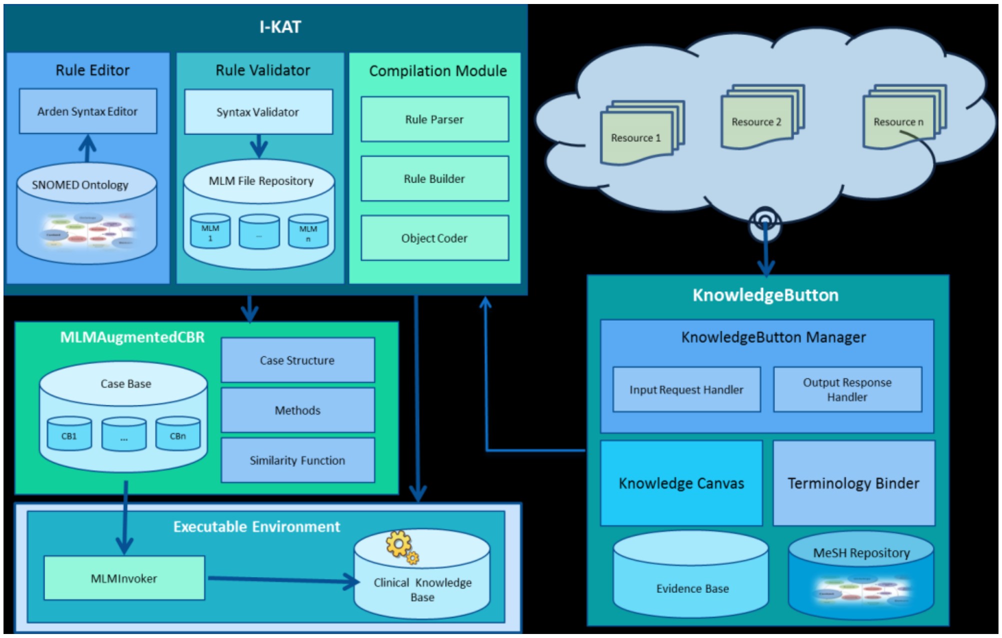
1.4.1 Motivation of Smart CDSS

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. Intelligent Knowledge Authoring Tool (I-KAT) is another component that facilitates the physicians to write the clinical knowledge rules by providing easy to use graphical user interfaces. Furthermore, we are providing evidence base support to clinical knowledge from online resoureces. KnowledgeButton is state of the art tool that facilitates proper support for attaching evidences from online resoureces.

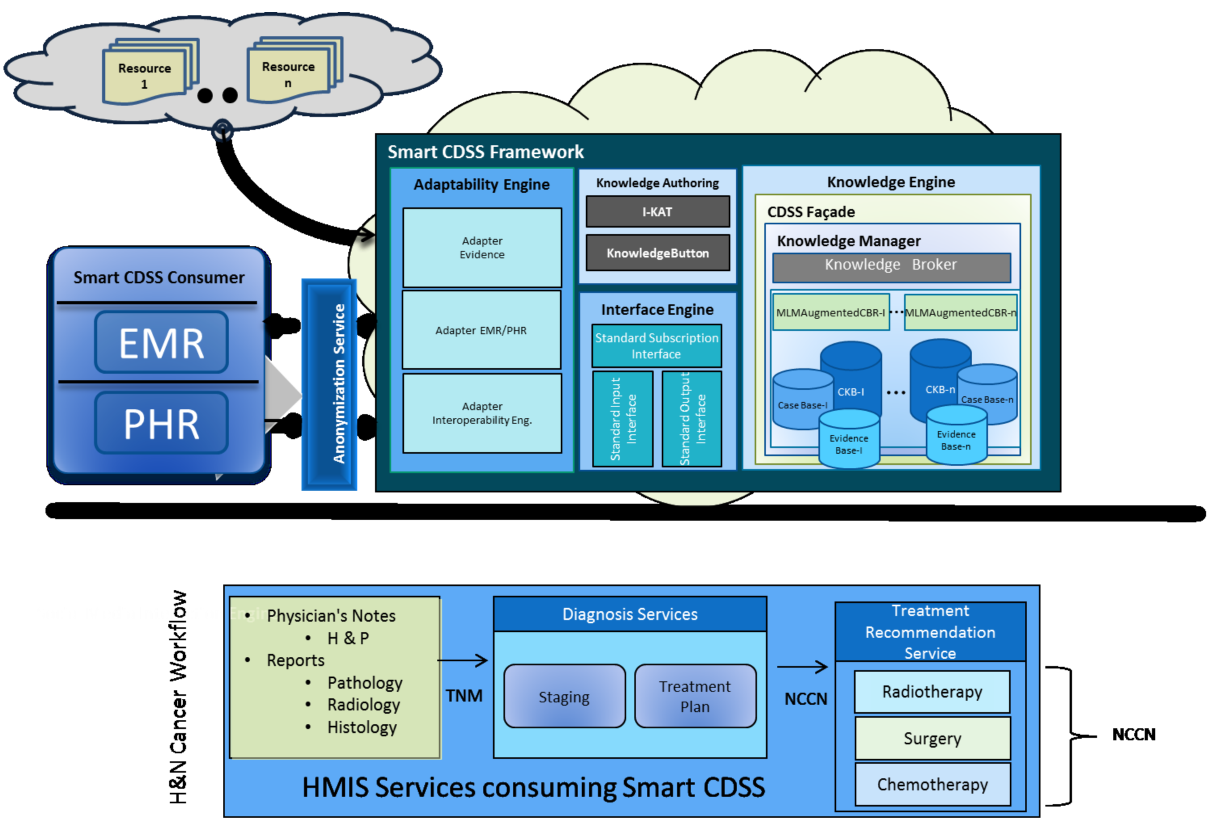
1.4.2 Uniqueness of Smart CDSS

- Collaborative Environment and Sharable Knowledge for seamless integration with healthcare systems
    - Use of standards like HL7 Arden Syntax provides the knowledge sharing facility.
- Easy to User Interface for the physicians to create and maintain the knowledge of their own
    - Design of an authoring tool provides the environment for creation knowledge rules.
    - Authoring Tool provides the environment of expert physicians to share their knowledge with expert system smart CDSS
- Dynamic availability of relevant information from online knowledge resources at run time
    - Provide access to online resources from within the context of clinical care.
    - Using support of online knowledge resources new evidences are added to the knowledge base. In this way, the CDSS becomes self-evolutionary Evidence Adaptive Clinical Decision Support System

1.4.3 Block Diagram



1.4.4 Detailed Architecture



CHAPTER 2: STANDARD BASED REASONING

2.1 Motivation

Standard base reasoning provides mechanism to execute knowledge rules persisted in form of standard MLMs. It allows maintaining shareable knowledge base and executing domain workflows provided by domain experts. The reasoner has the capability to handle event and data driven intervention provided by domain expert. It gives physicians to enhance knowledge base according to their requirements with minimal efforts.

2.2 Goals

- Constructing standard base domain knowledge for sharing among diverse organization.
- Standard base reasoner allows unify interfaces to manipulate knowledge rules representing domain expertise of physicians.
- Provide capability to fetch medical data required for decision making from diverse data source using standard vMR interfaces and generate standard base guidelines.

2.3 Related Work

The clinical decision support system (CDSS) has a strong history, starting in 1960 with stand-alone environments. With the advancement of architectural approaches and new requirements, CDSS has evolved from a stand-alone to a service-oriented architecture (SOA) environment. Moreover, for seamless integration of CDSS with existing health-care systems (EHRs, EMRs, PHRs, and CPOEs) to allow sharing of medical knowledge, various standards have emerged to achieve the desired goals. The most prominent knowledge representation language in the clinical domain is HL7 Arden syntax. Therefore, we will discuss the CDSS supporting Arden syntax as the main standard for the knowledge base.

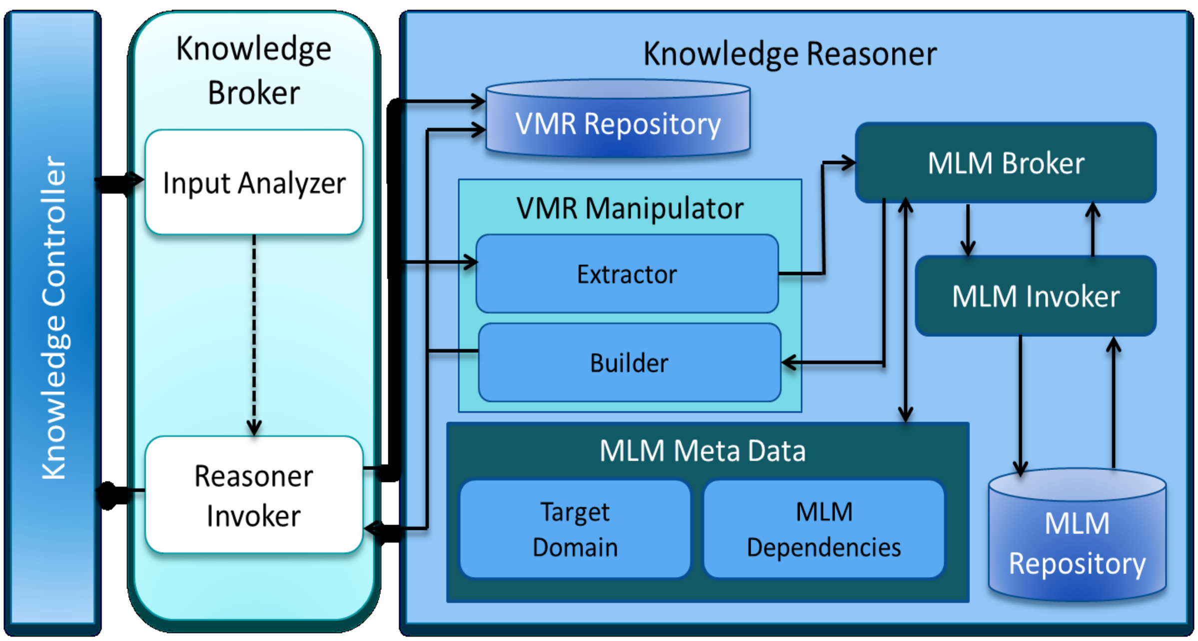
Moni-ICU detects and continuously monitors nosocomial (i.e., hospital-acquired) infections. Moni-ICU uses a distinct approach by invoking a number of MLMs and implementing different rules that are controlled from one central MLM. The Moni-ICU application works in the ICU connected to a microbiology lab and a patient management system. It monitors all patients on a daily basis in each of the normal intensive care units, which comprise around 100 beds in total [Samwald2012].

Arden2ByteCode, a newly developed open-source compiler, runs on Java virtual machines

(JVM) and translates Arden syntax directly to Java byte code (JBC) executable. This complier is integrated with SOA-based environments called open services gateway initiative (OSGi) platforms. The compiler has the capability to support all operators of Arden syntax and compile production Java bytecode in minimal time. Due to this direct byte code, the execution time of MLM is considerably reduced [Gietzelt2011].

Arden/J is a Java-based MLM execution environment that provides integration with XML-based and EMR systems and produces recommendations by executing MLM compiled to Java code. Arden/J supports a runtime environment that allows integration with other systems by implementing mapper interfaces. The authors claim good performance of the compiler and have tested it with XMLbased EMR [Karadimas2002].

2.4 Architecture



2.5 Uniqueness

- Extraction of data from diverse data source.
- Allow brokering service to decouple diverse domain knowledge.
- Capable of generating recommendation based on datasets fused from different data sources.

- Provide ownership on domain knowledge.
- Potentials to evolve knowledge base for futuristic domain rules.

2.6 UML Diagrams

2.6.1 User Classes and Characteristics

Actors

1. Patient

   Patient is responsible to provide the social media data, his/her data is also collected using sensors, his behavior data and clinical data.
2. Physician

   Physician interacts with the system by entering the required data that is converted to HMIS compliant standard format.
3. BMM Manager

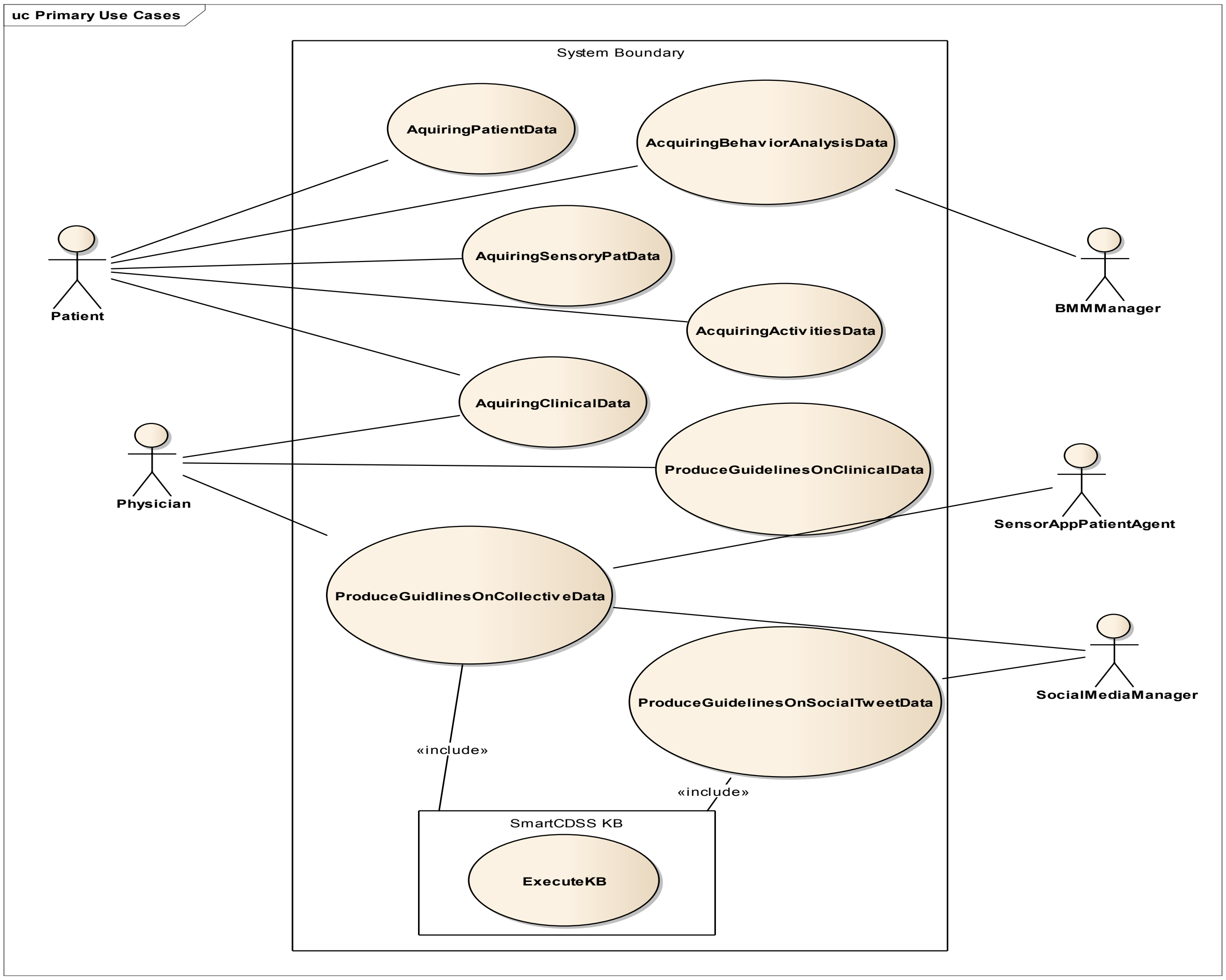
   BMM Manager records the behavior related information of the patient.
4. Social Media Manager

   Social Media Manager is responsible for managing the social media information of the patient.
5. Sensor Application Patient Agent

   Sensor Application Patient Agent is responsible for collecting the sensor based low level and high level patient activities.

2.7 Smart CDSS Primary Use Case Model

2.7.1 Use Case Description



2.7.2 Brief Descriptions of Use Cases

- **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 SMART CDSS 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.

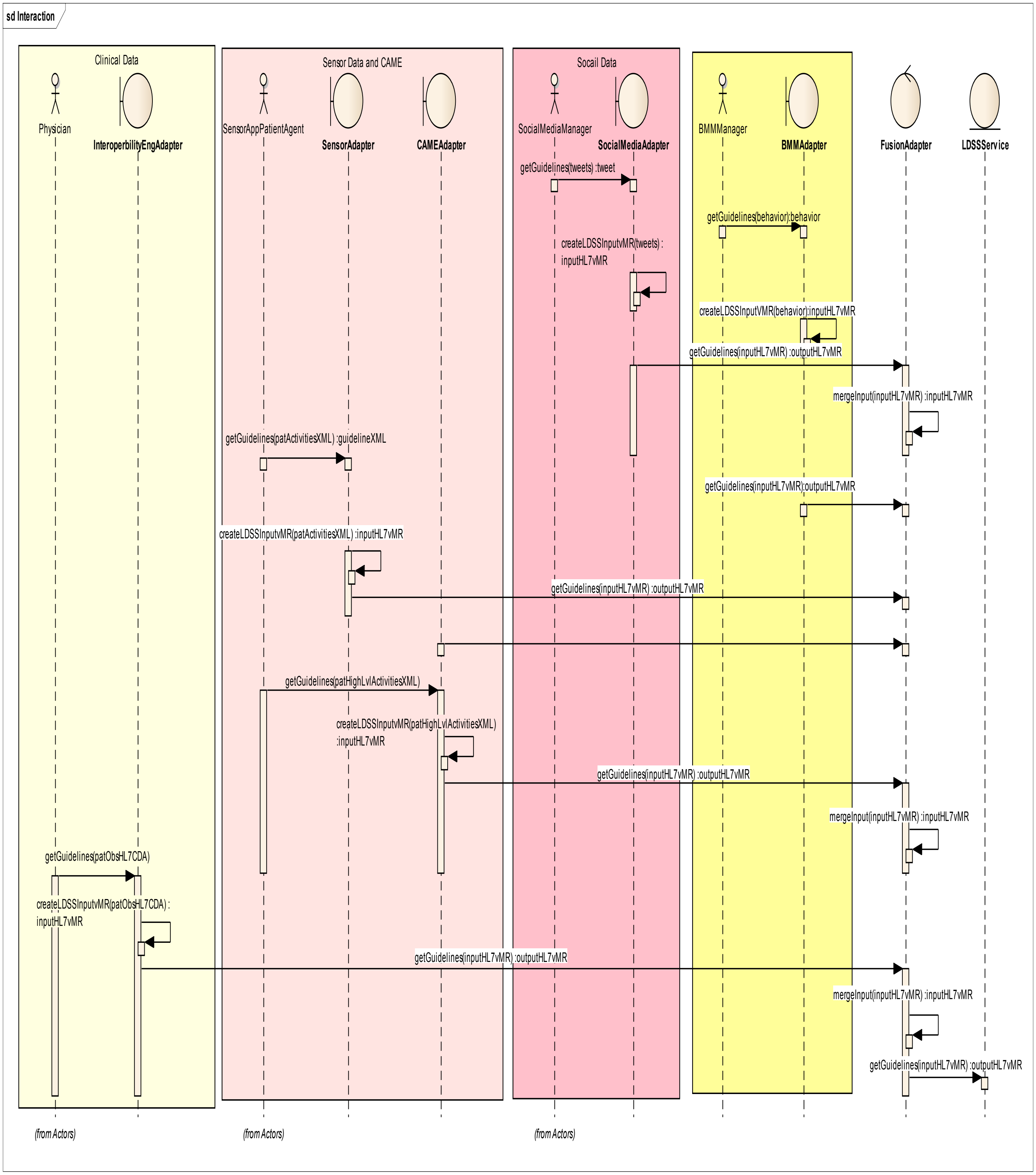
2.8 Interaction Model

Sequence diagrams are described as follows that shows the interaction between different objects. Smart CDSS provides flexible architecture for supporting multiple knowledge bases derived from same basic infrastructure. Following are example, interaction diagrams that helps in interaction of Smart CDSS in diverse domain such as in combination of diverse data sources and in solo with clinical domain such as cancer.

Smart CDSS overall sequence diagram is as follows:

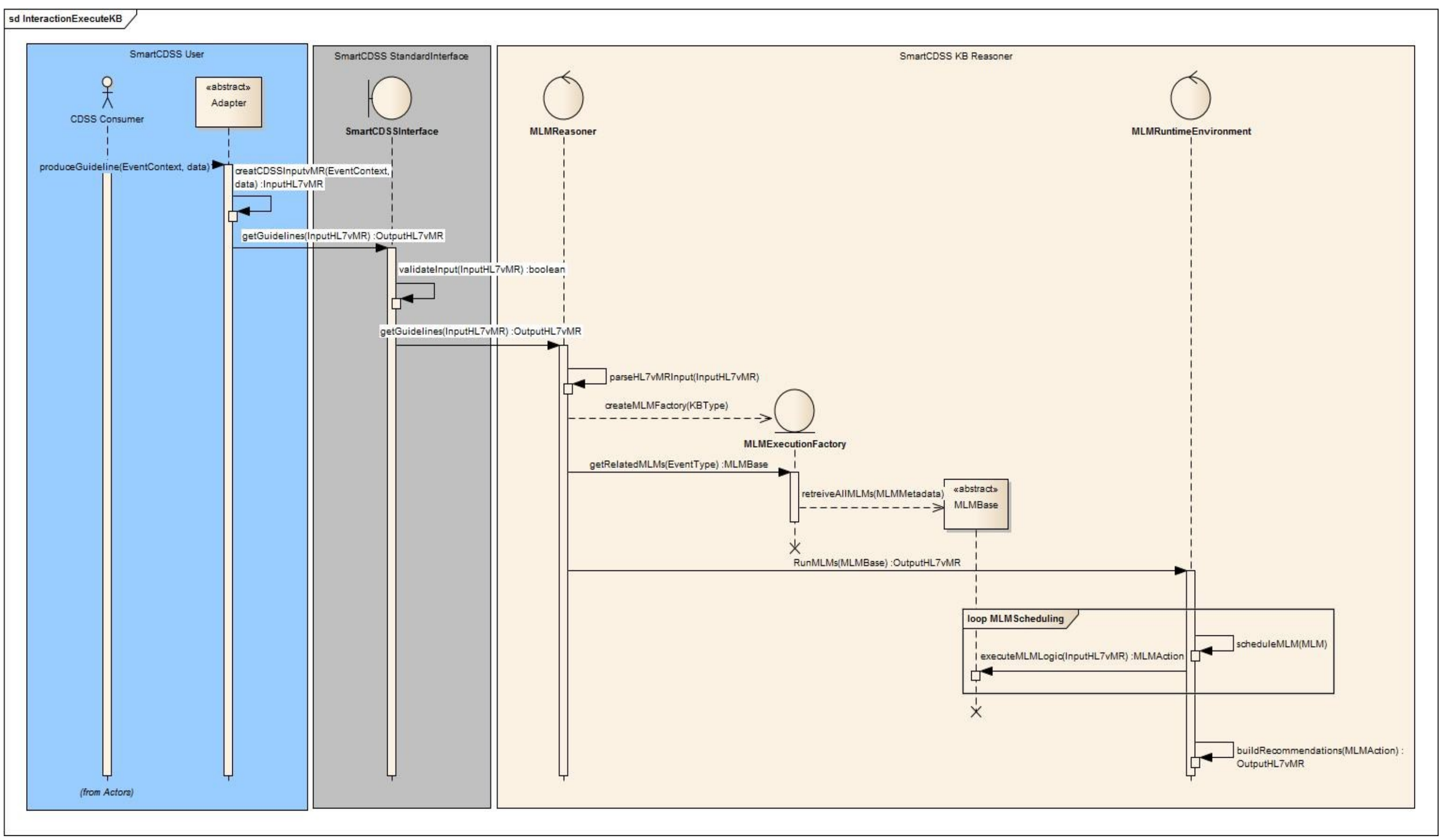
2.8.1 Smart CDSS Overall Sequence Diagram

- Physician provides input in the form of observations using getGuidelines(patObsHL7CDA) method to Smart CDSS. 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 Smart CDSS Service object for obtaining guidelines using getGuidelines(imoutHL7VMR) method.


sd Interaction
Clinical Data
Sensor Data and CAME
Socail Data
Physician
InteroperbilityEngAdapter
SensorAppPatientAgent
SensorAdapter
CAMEAdapter
SocialMediaManager
SocialMediaAdapter
BMMManager
BMMAdapter
FusionAdapter
LDSSService
getGuidelines(tweets) :tweet
getGuidelines(behavior):behavior
createLDSSInputvMR(tweets) : inputHL7vMR
createLDSSInputVMR(behavior):inputHL7vMR
getGuidelines(inputHL7vMR) :outputHL7vMR
mergeInput(inputHL7vMR) :inputHL7vMR
getGuidelines(patActivitiesXML) :guidelineXML
getGuidelines(inputHL7vMR):outputHL7vMR
createLDSSInputvMR(patActivitiesXML) :inputHL7vMR
getGuidelines(inputHL7vMR) :outputHL7vMR
getGuidelines(patHighLvlActivitiesXML)
createLDSSInputvMR(patHighLvlActivitiesXML) :inputHL7vMR
getGuidelines(inputHL7vMR) :outputHL7vMR
mergeInput(inputHL7vMR) :inputHL7vMR
getGuidelines(patObsHL7CDA)
createLDSSInputvMR(patObsHL7CDA) : inputHL7vMR
getGuidelines(inputHL7vMR) :outputHL7vMR
mergeInput(inputHL7vMR) :inputHL7vMR
getGuidelines(inputHL7vMR) :outputHL7vMR
(from Actors)
(from Actors)
(from Actors)

2.8.2 Smart CDSS Knowledge Execution Sequence Diagram

- Smart CDSS consumer applications such as HMIS generate an event and submit it to Smart CDSS for possible recommendations.
- Smart CDSS Adapter interface allows converting the consumer application data (such CDA) format into HL7 vMR format and forward to Smart CDSS via standard interfaces.
- Smart CDSS standard interfaces validate the consumer input and pass the valid request with data to reasoner.
- MLM reasoner in Smart CDSS interprets the consumer application data and schedules appropriate MLMs with MLM runtime environment.
- The scheduled MLM are executed and produce corresponding recommendations in vMR output format and passed to consumer application.



2.9 Class Diagram

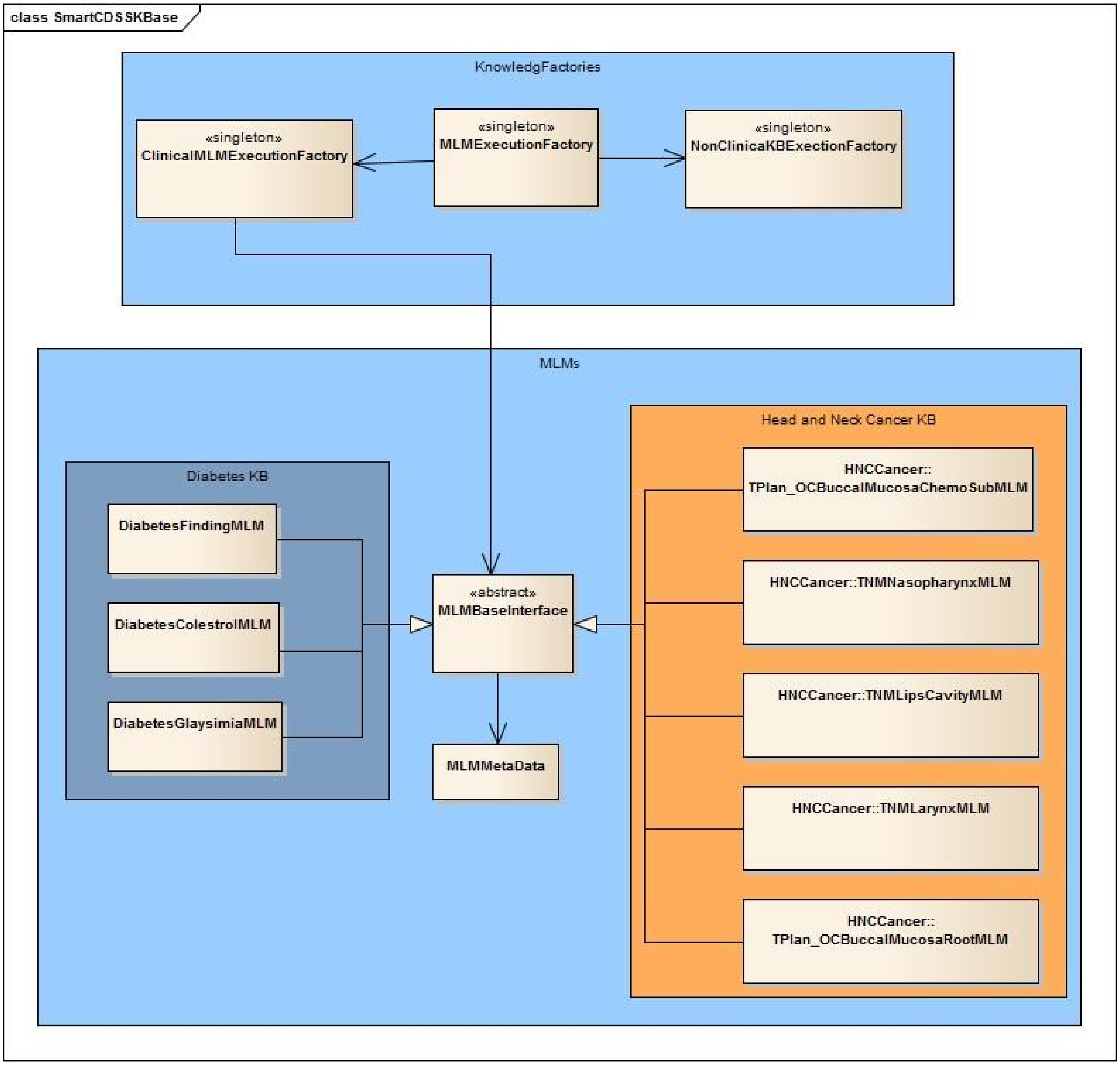
Smart CDSS class diagram shows how the different classes relate with each other to provide appropriate guidelines to the physicians. Medical Logic Modules (MLM) is the standard format for generation and storage of rules in Arden Syntax. These MLMs store the logic behind the rule to be fired. These are 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. This abstraction allows creating and maintaining multiple knowledge bases for different diseases. For example, in current work, Smart CDSS support basic recommendations for diabetes and head and neck cancer.

Various diabetes guidelines have been implemented through specialized MLM classes. DiabetesFindingMLM class 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.

Similarly, recent work is focusing on cancer domain. Guidelines related to head and cancer staging from TNM staging system and treatment plans from NCCN guidelines have been implemented. For staging, it includes MLMs for lips cavity, larynx and nasopharynx such as TNMLipscavityMLM, TNMlarynxMLM and TNMNasopharynxMLM. For treatment plan, TPlan_OCBuccalMucosaChemoSubMLM, TPlan_OCBuccalMucosaRootMLM and TPlan_OCBuccalMucosaSurgarySubMLM for chemo and surgery treatments of buccal cancer.



2.10 Description of Components

The component diagram of Smart CDSS shows the different components and their subcomponents interactions with each other. These are explained as follows:

2.10.1 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.

2.10.2 Adaptability Engine

This component is responsible for obtaining as input data about HNC from different heterogeneous modules. The modules outside Smart CDSS 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 Interoperability Engine
- EMR/EHR Engine

2.10.3 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

2.10.4 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:

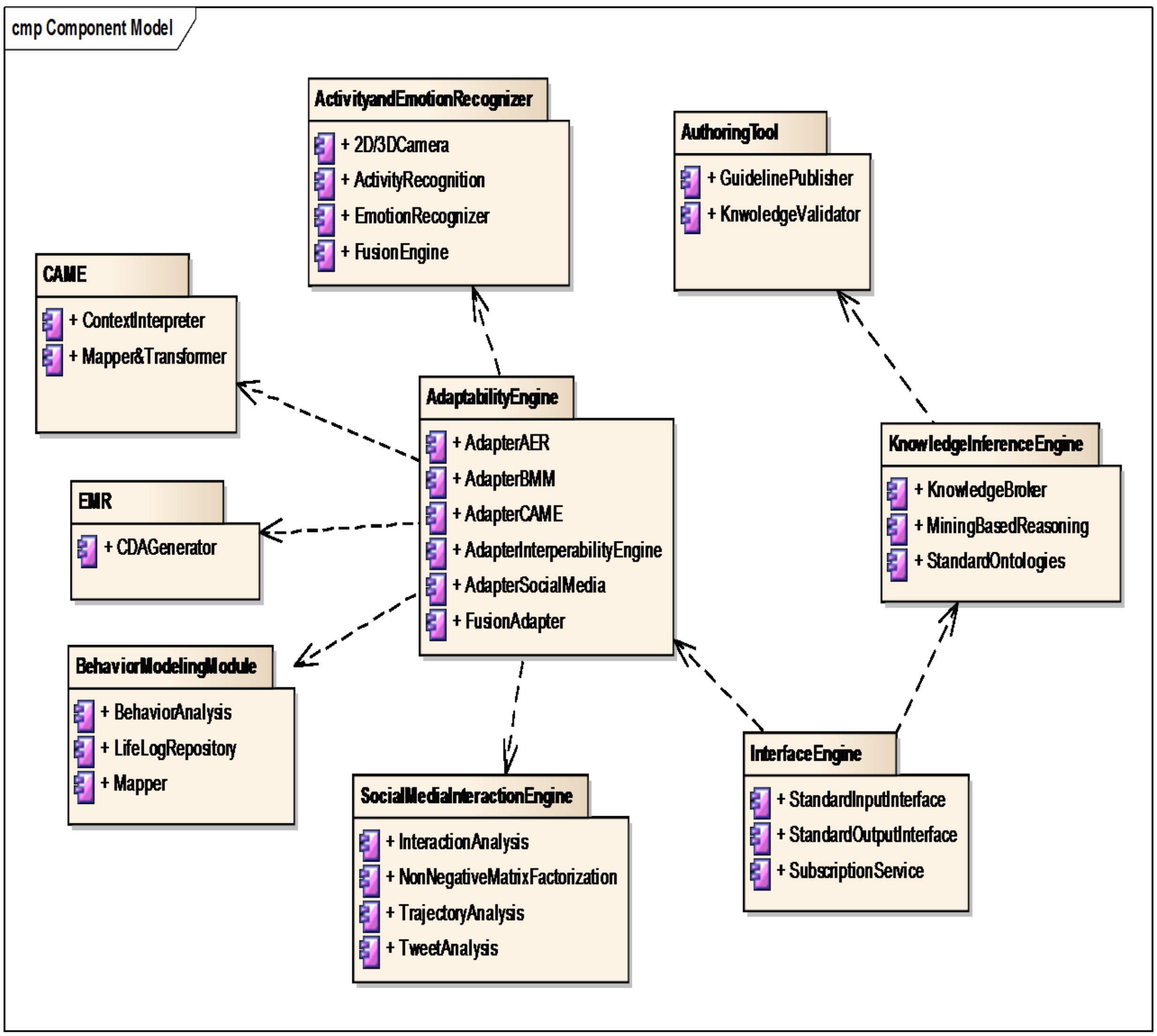
- Standard Input Interface
- Standard Output Interface
- Subscription Service

2.10.5 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 case base reasoning mechanism. It consists of the following subcomponents:

- Knowledge Broker
- MLMAugmentedReasoner
- Standard Clinical Knowledge base

All the main components of Smart CDSS that are responsible for taking data in different formats are described in detail in the next sub-sections.



CHAPTER 3: ADAPTER INTEROPERABILITY ENGINE (ARIEN)

3.1 Motivation

Smart CDSS takes clinical data from HMIS for processing using Adapter Interoperability Engine (ARIEN) component. Heterogeneities among HMIS compliant standard and Smart CDSS compatible standard exists. ARIEN resolves these heterogeneities for HMIS's to interact easily with Smart CDSS 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 Smart CDSS 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.

3.2 Goals

- Processing clinical data to build up clinical knowledge base
- Providing interoperability among medical systems and DSS
- Mapping service for conversion between medical systems compliant standards and DSS compliant standard
- Accuracy and continuity of mappings between healthcare standards with vMR
- Integrating legacy EMR/EHR/PHR/CPOE systems with DSS

3.3 Related Work

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 is an ontology mapping is 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.

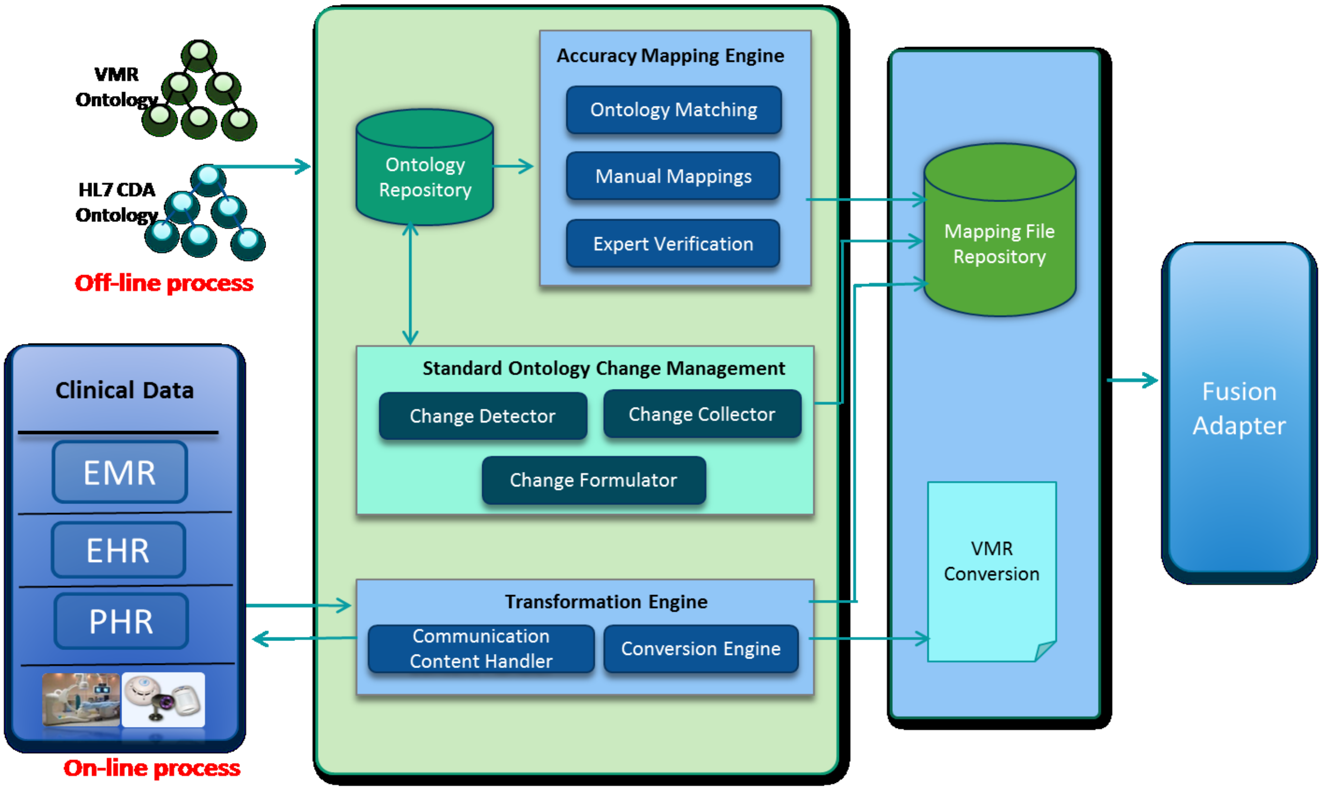
PPEPR [sahay2008] project is an integration platform that focuses on resolving the heterogeneity problem between two version of the same standard HL7 (V2 and V3). It is based on semantic SOA concepts and solves the problem of interoperability at the semantic level. It used Web Service Modeling Ontology (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 thats 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.

3.4 Architecture



3.5 Uniqueness

- Interoperability Adapter uses mapping service for conversion of clinical data into DSS standard format and vice versa
- Mapping Service is based on storage of mappings in Bridge Ontology (a mappings storage representation ontology)
- Accuracy of mappings is maintained for less data loss while conversion
- Changes in medical ontologies is catered by ensuring continuity of mappings

3.6 UML Diagrams

3.6.1 User Classes and Characteristics

Actors

- Software Engineer

Software Engineer is responsible for generating the ontology mappings using different ontologies and storing the generated mappings in a repository.

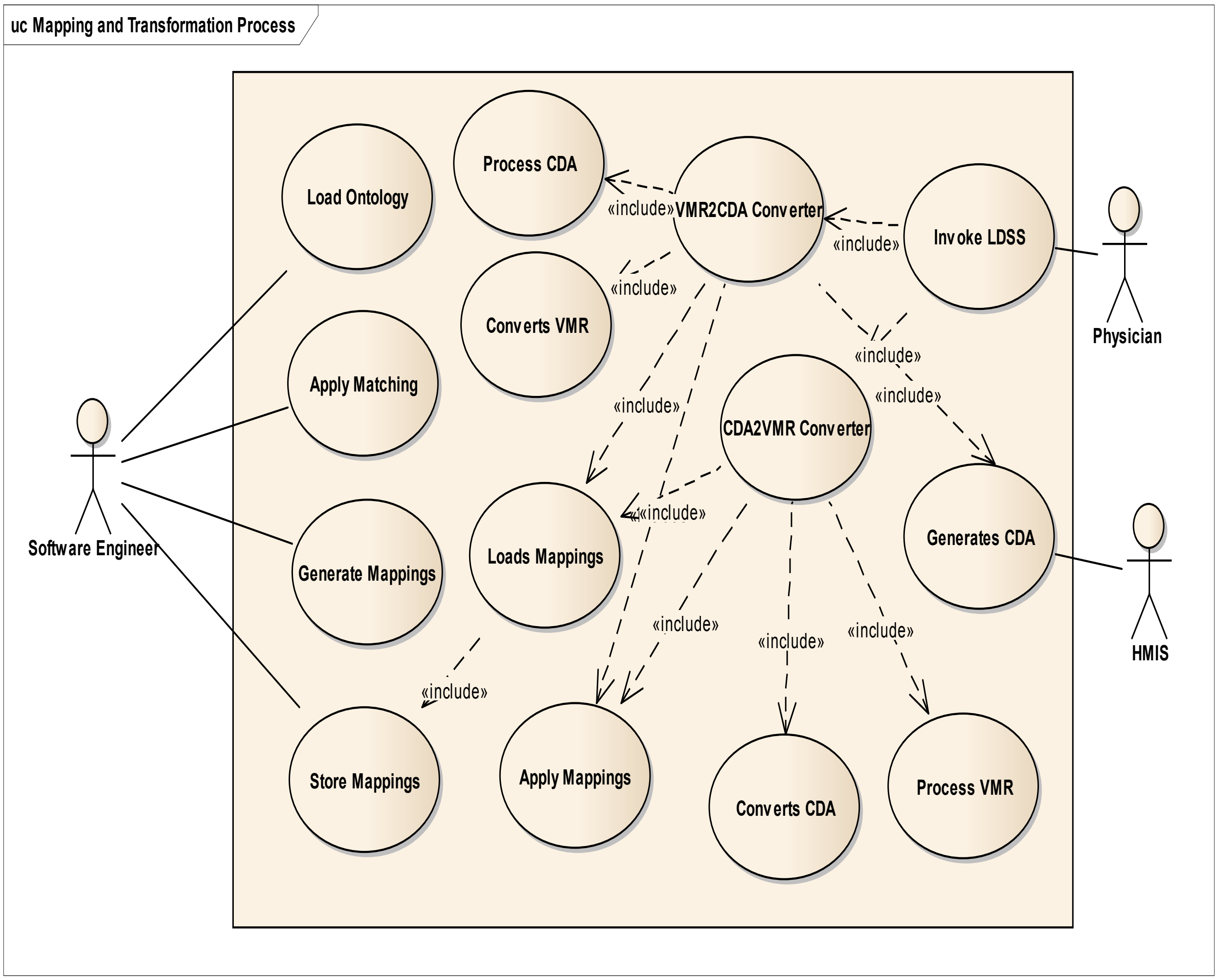
- Physician

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.

3.7 Use Case Model for Mappings Generation and Transformation



3.7.1 Use Case Description

Brief Descriptions of Use Cases

- **Load Ontology**

  Load Ontology use case loads the ontologies on which ontology matching needs to be performed. It loads the source and target ontologies for matching purpose.

- **Apply Matching**

Applies Matching is the use case that is responsible for selecting the matching techniques that needs to applied for ontology matching process. It applies the ontology matching techniques to generate mappings.

- **Generate Mappings**

  Generate Mappings use case applies ontology matching process and generates the mappings between source and target ontologies.

- **Store Mappings**

  The mappings generated are then stored in repository by Store Mappings use case. The repository contains many mapping files.

- **Invokes LDSS**

  Invokes LDSS use case is responsible for connecting the HMIS with the SMART CDSS. It includes converters for different standards.

- **Generates CDA**

  Generates CDA use case is used for creation of CDA instance from the data entered by the physician.

- **VMR2CDA Converter**

  VMR2CDA Converter use case coverts SMART CDSS standard format that is VMR to HMIS standard format which is CDA.

- **CDA2VMR Converter**

  CDA2VMR Converter use case coverts HMIS standard format which is CDA to SMART CDSS standard format that is VMR.

- **Process CDA**

  Process CDA use case processes CDA instance information and its constructs are accessed for applying the mappings stored in the repository.

- **Converts VMR**

  Converts VMR use case performs conversion from CDA format to VMR format using specific mappings generated from VMR and CDA models.

- **Process VMR**

  Process VMR use case processes VMR instance information and its constructs are accessed for applying the mappings stored in the repository.

- **Converts CDA**

Converts CDA use case performs conversion from VMR format to CDA format using specific mappings generated from VMR and CDA models.

- **Load Mappings**

  Load Mappings use case access the mapping repository and loads the required mappings to be used for conversion purpose.

- **Apply Mappings**

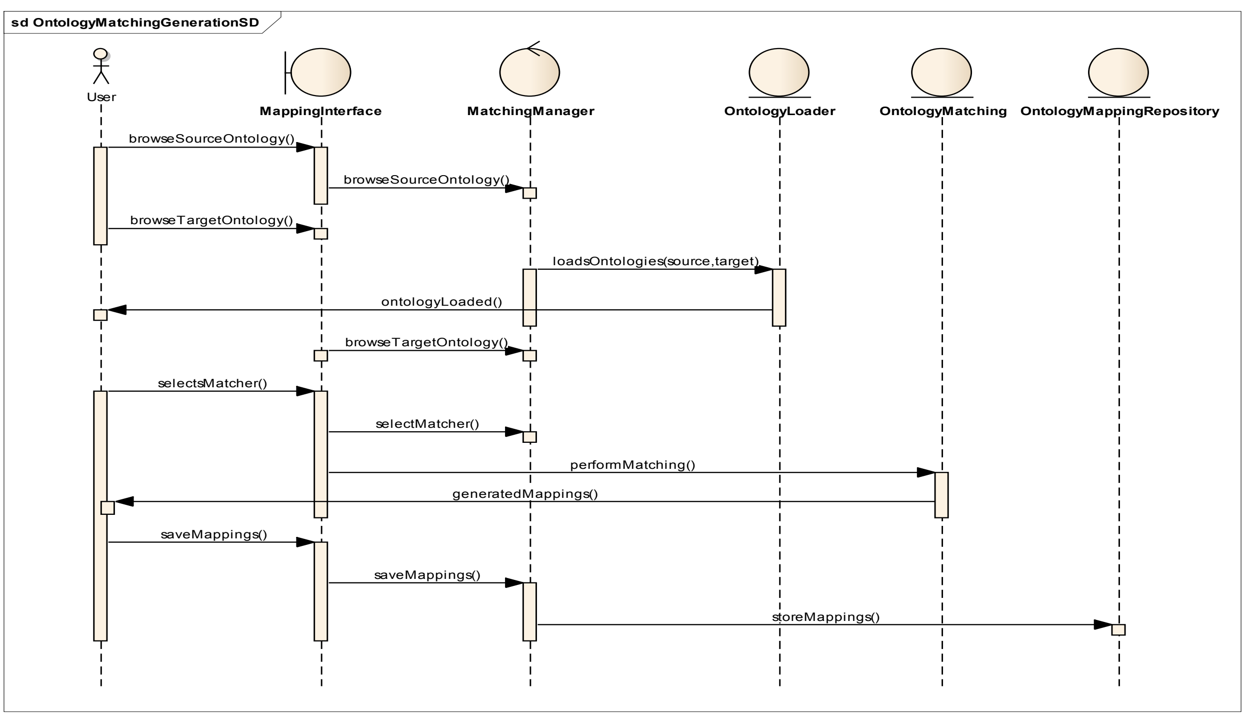
  Apply Mappings use case applied the loaded mappings for conversion from one standard format to another standard format.

3.8 Interaction Model

Sequence diagrams are described as follows that shows the interaction between different objects.

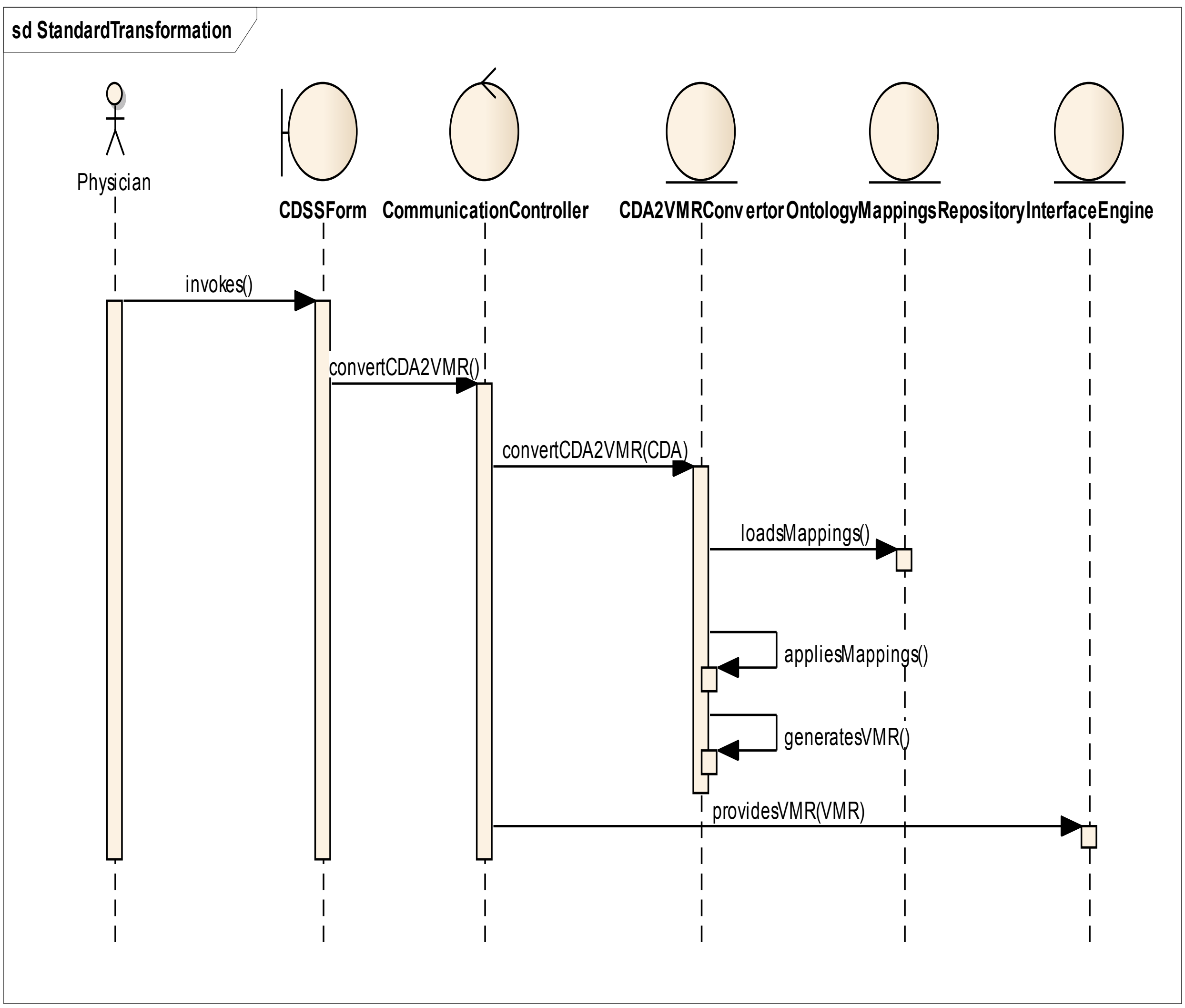
3.8.1 Generate Ontology Mappings

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.



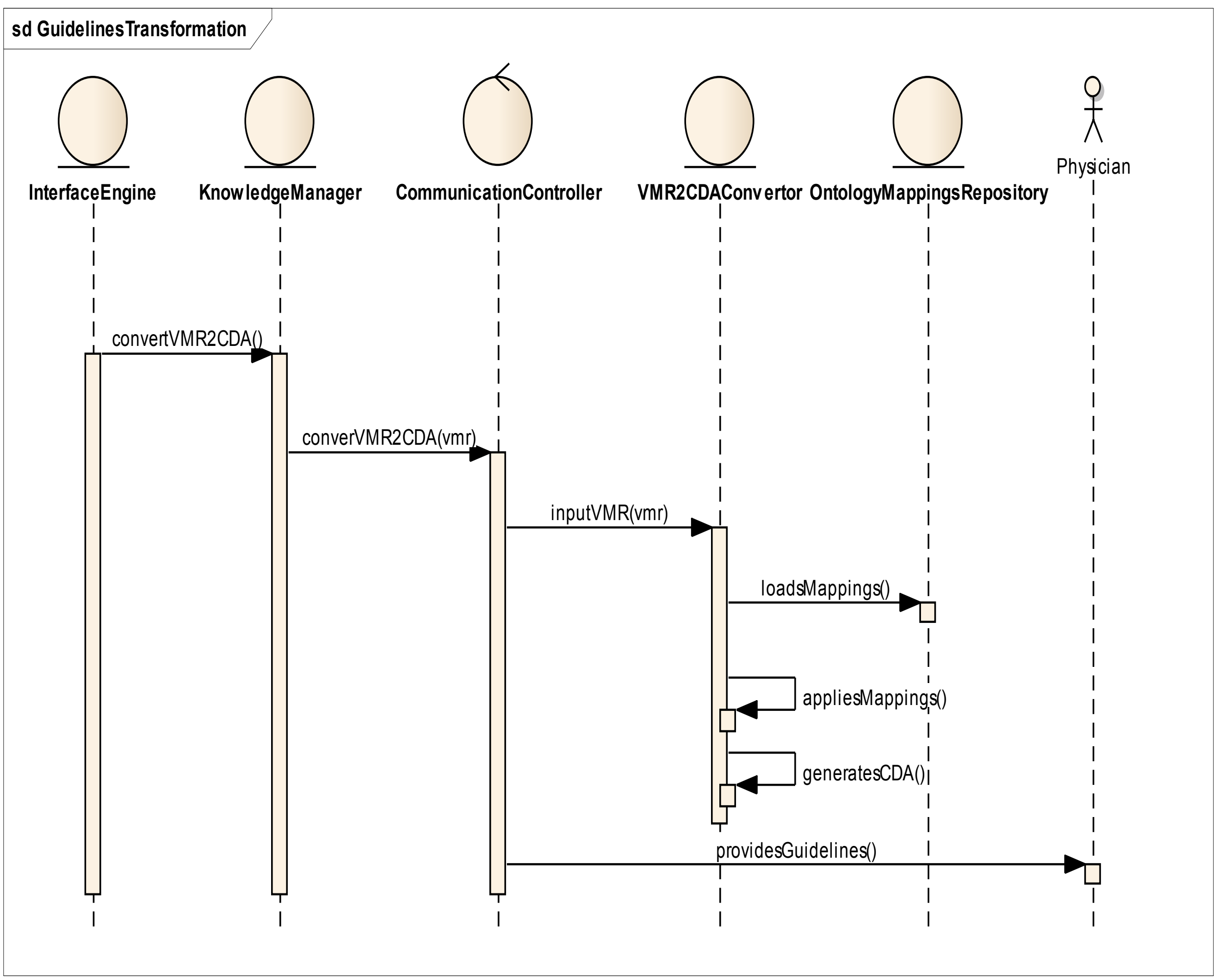
3.8.2 Transform Standard Format

Physician invokes the SMART CDSS 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.



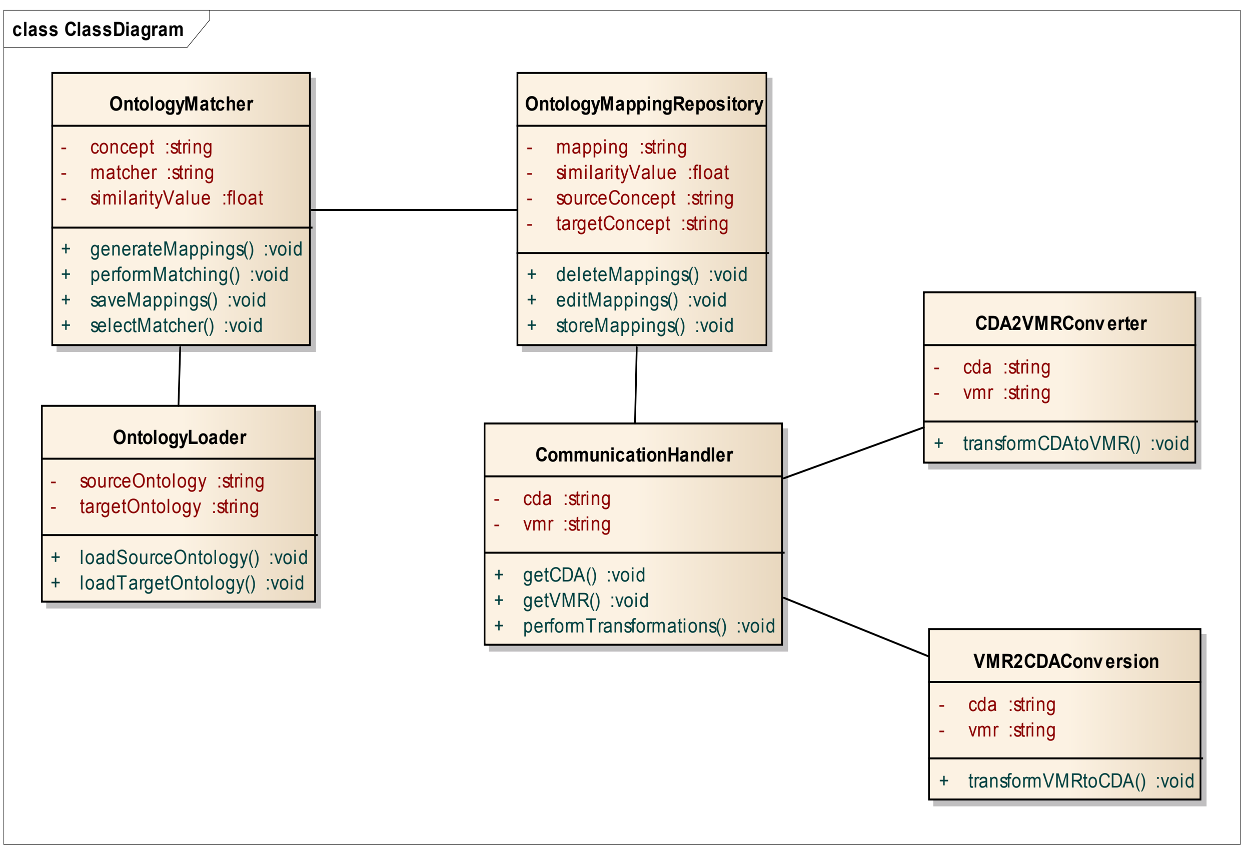
3.8.3 Transform Guidelines

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.



3.9 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.



3.10 Detailed Description of Components

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:

3.10.1 AccuracyMappingEngine Component

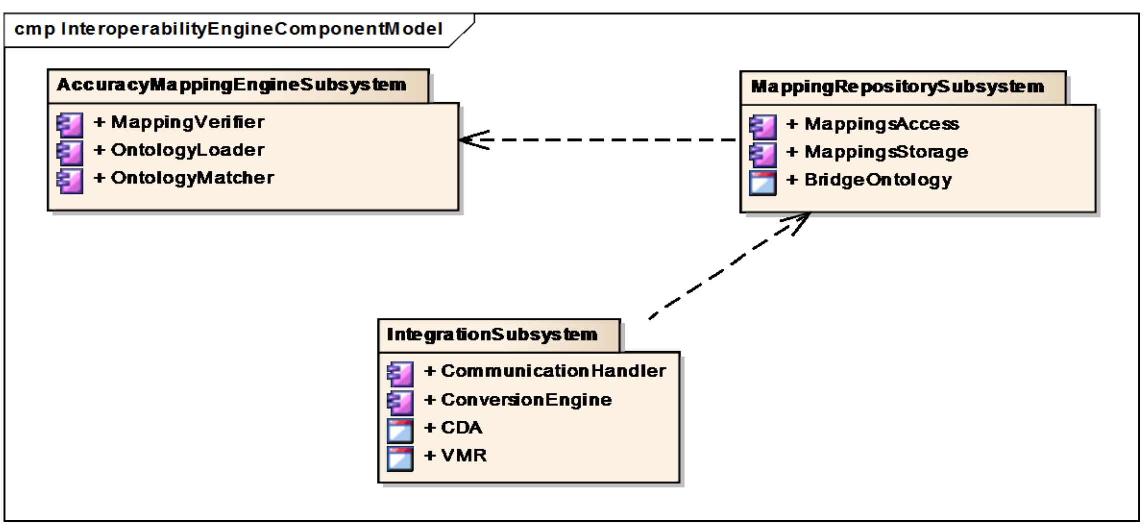
It is the main component of Adapter Interoperability Engine that is used for generating the mappings between differnet healthcare standards and then storing them. It is composed of subcomponents like OntologyLoader, OntologyMatcher and MappingVerifier.

3.10.2 MappingRepository 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.

3.10.3 IntegrationModule Component

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



3.10.4 OntologyLoader Subcomponent

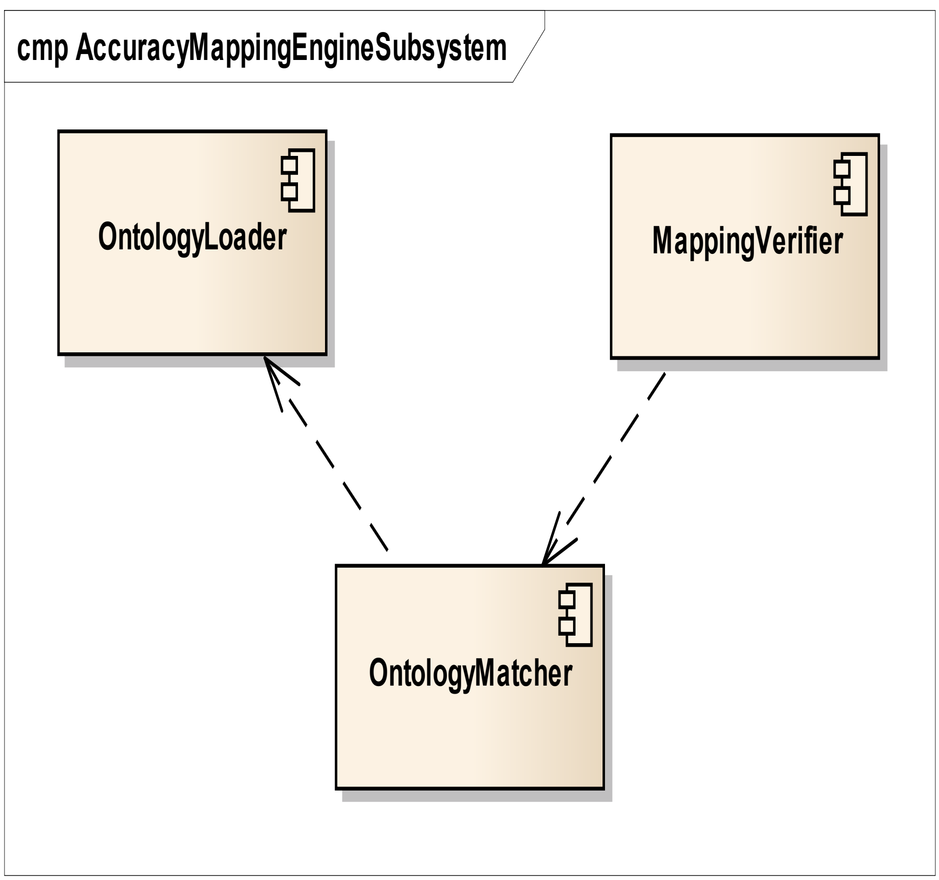
OntologyLoader is the subcomponent of AccuracyMappingEngine component and is used for loading the source and target ontologies for mappings generation.

3.10.5 OntologyMatcher Subcomponent

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

3.10.6 MappingVerifier Subcomponent

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



3.10.7 MappingStorage Subcomponent

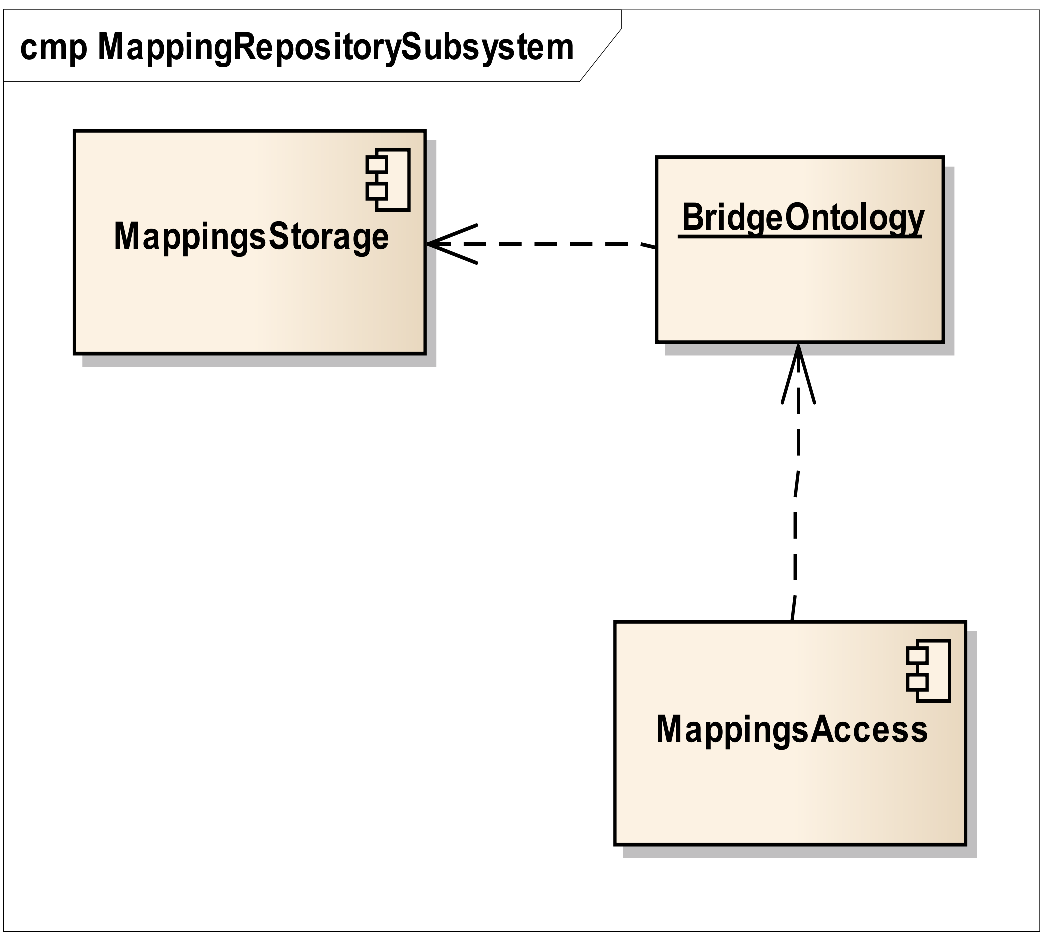
MappingStorage is the subcomponent of MappingRepository component and is used for storing the mappings generated using ontology matching techniques. This component gives the mapping file a structure and stores it in RDF format.

3.10.8 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.

3.10.9 BridgeOntology Object

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



3.10.10 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.

3.10.11 ConversionEngine Subcomponent

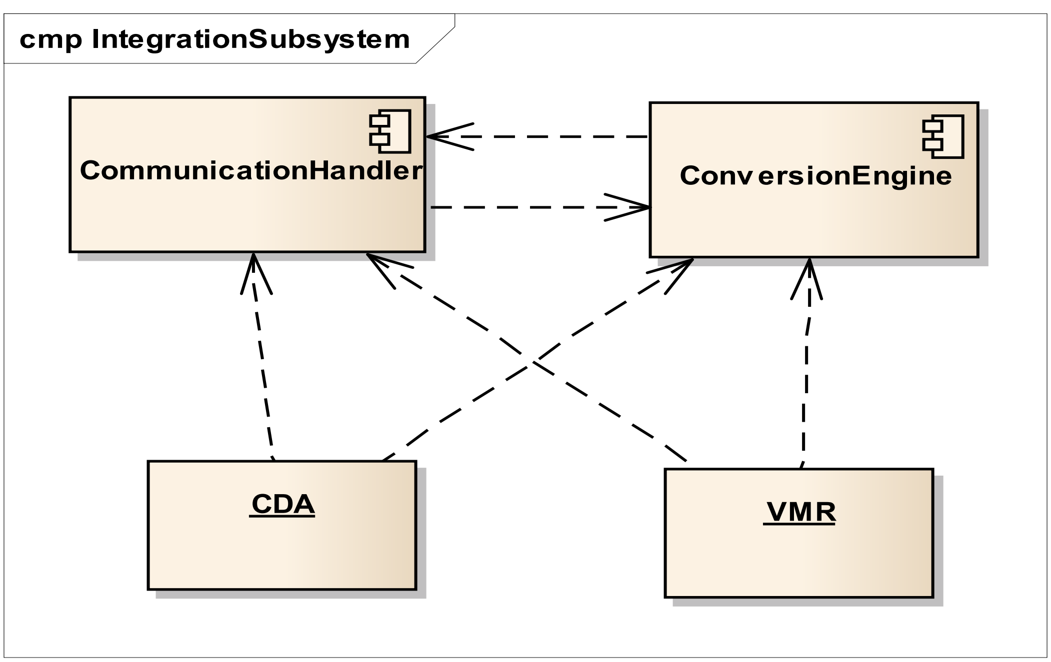
ConversionEngine is the subcomponent of IntegrationModule component and is used to transform standard format from HMIS compliant standard to SMART CDSS compatible standard and vice versa.

3.10.12 CDA Object

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

3.10.13 VMR Object

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



CHAPTER 4: INTELLIGENT KNOWLEDGE AUTHORING TOOL (I-KAT)

4.1 Motivation

Clinical Decision Support Systems (CDSS) assist clinicians in making clinical decisions by using experts' knowledge stored in the knowledge base. However, sharing and reusing the knowledge is a challenging task. Many systems are developed to facilitate sharing of medical knowledge and allow its reusability. These systems are compliant to standard approaches such as HL7 Arden Syntax and HL7 CDA (Clinical Document Architecture) to incorporate medical logic in standard format. The main drawback with these systems is the complicated procedure in the development of clinical knowledge by ordinary clinicians. The proposed research work is focusing on developing authoring tool that creates sharable clinical knowledge base using standards such as HL7 Arden Syntax, HL7 vMR and HL7 CDA. To achieve the goal of shareability and interoperability, along with the HL7 standards vMR and Arden Syntax, some standard terminologies are also required; in this case we have chosen the SNOMED CT medical terminologies. SNOMED CT is international standard terminologies, has multilingual support and contains more than 311,000 active concepts, in this system only the domain concepts will use in creating rules. The domain ontology will be extracted by the physician only once for a single domain. Moreover, the authoring tool provides user friendly GUI to facilitate clinicians in creating standard based executable clinical knowledge base. We are closely working with oncologists and clinicians of a prominent cancer hospital to deploy the tool for Head and Neck Cancer diagnosis and treatment recommendations.

4.2 Goals

- To develop, manage and maintain the Knowledge Base for Smart CDSS which assists the physicians to provide guidelines and recommendations.
- To provide the environment (GUI) where physicians can share their knowledge with other systems.
- To create shareable knowledge in form of Arden Syntax MLM behind the creation of knowledge rule by physician.
- To transform the shareable Arden Syntax MLM to executable form to make it usable compiled rules for Smart CDSS to generate recommendation.
- To provide interface to end user for extracting the domain ontology from overall SNOMED CT ontology

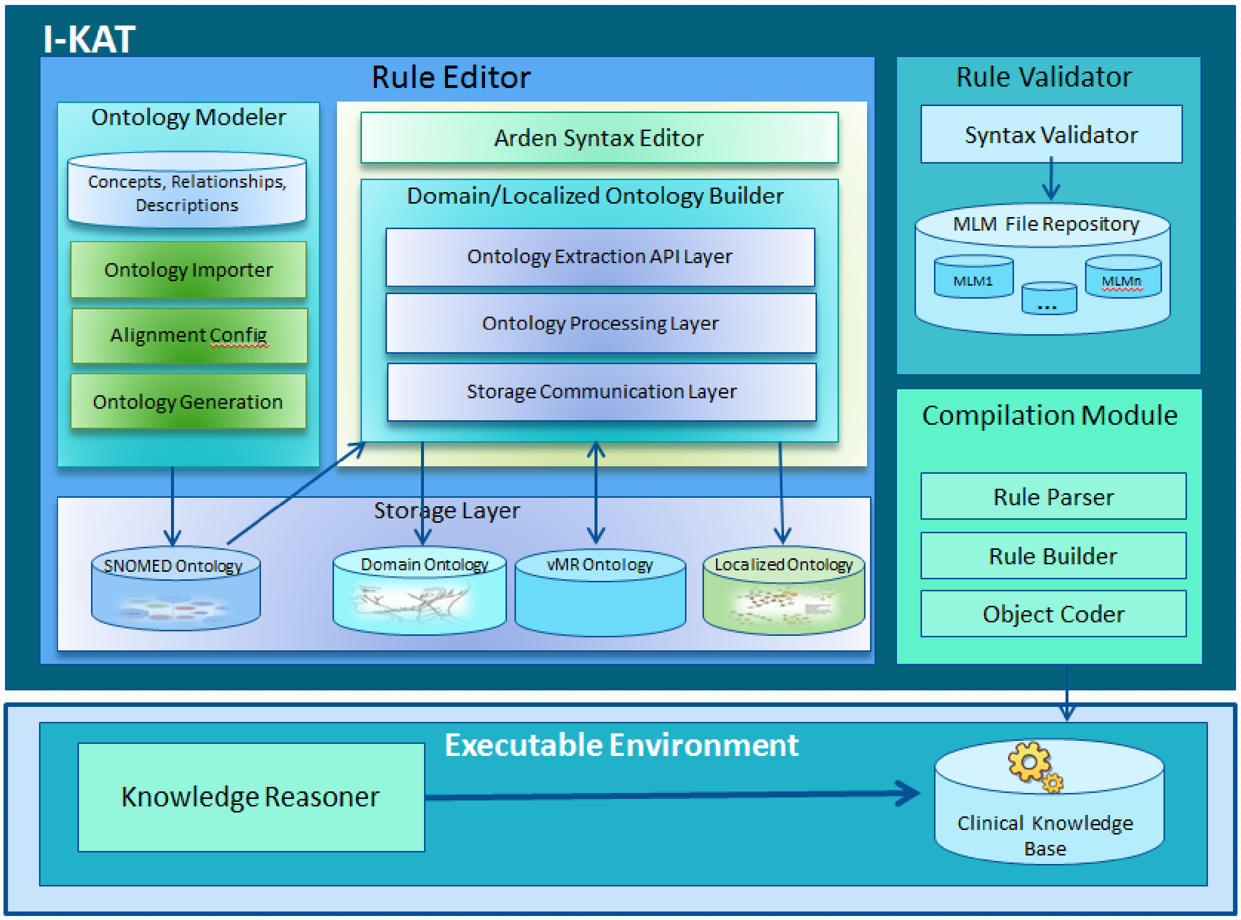
4.3 Related Work

There are many existing systems in literature focusing on development of authoring tools in the area of CDSS. A. Soumeya et al. [Soumeya2012] presented a UMLS based knowledge acquisition tool for CDSS. This tool allows the clinical experts to build and maintain clinical knowledge base in systematic manner. Although the tool provides benefits to clinical experts but it is not an easy to use tool for the clinicians due to more focus on intermediate phases to create a single rule. The phases include clinician to create domain ontology using UMLS browser initially, then selecting all related concepts for creating rule, and finally organizing an ascending order list to create actual rule. These phases put a lot of burden on the already burdened clinicians.

Nathan C. Hulse et al. [Nathan2005] proposed Knowledge Authoring Tool architecture and uses CDA standard for sharable and reusable information documents. The complexity of the system is increased because the executable information is also embedded in CDA format. Relationship of multiple CDA for a single patient makes processing more complex. CDA is a preferable approach for sharing data only, but is not a suitable choice for sharing knowledge rules. Dustin Dunsmuir et al. [Dustin2008], developed a Knowledge Authoring Tool by using a valuable Pattern And Outcome approach. This authoring tool is used in CDSS for anesthesia but its scope is limited and difficult to extend to other diseases. Also clinicians are directly exposed to work in XML file, requiring extra tedious task of XML training. Robert A. Jenders et al. [Robert2002] implemented standard based knowledge editor architecture which is used for knowledge dissemination and knowledge base sharing. Understandability of the system is difficult for clinicians and common users due to selection of appropriate object from the vMR standard data model. The end user can only use the system if they have fair knowledge about vMR standard and MLM structure.

Rachel Regier et al. [Rachel2009] proposed a clinical rule editor for EMR, developed for knowledge and rule management. Also, its purpose is to remove difficulty of editing hard coded rules. To create a single reminder needs more than one primitives, and the creation of these primitives are fully dependent on software and knowledge engineers. The existing systems discussed in this section worked on rule creation using authoring tool. Although these initiatives reflect considerable efforts in the area of authoring tools development but still lack in facilitating clinicians in easy to use system. Our proposed system maintains higher priority on easy to use system for clinicians by hiding unnecessary and complex technical details of standards specifications.

4.4 Architecture



4.5 Uniqueness

- Provide user friendly and manageable interface to create rules for knowledge base
- Reusability: Create reusable and extendable clinical rules
- Sharable: Create sharable rule, that can easily share with different clinical communities as standard MLM
- Incorporating interoperability standards to communicate
- Automatic extraction of domain ontology using SNOMED CT ontology

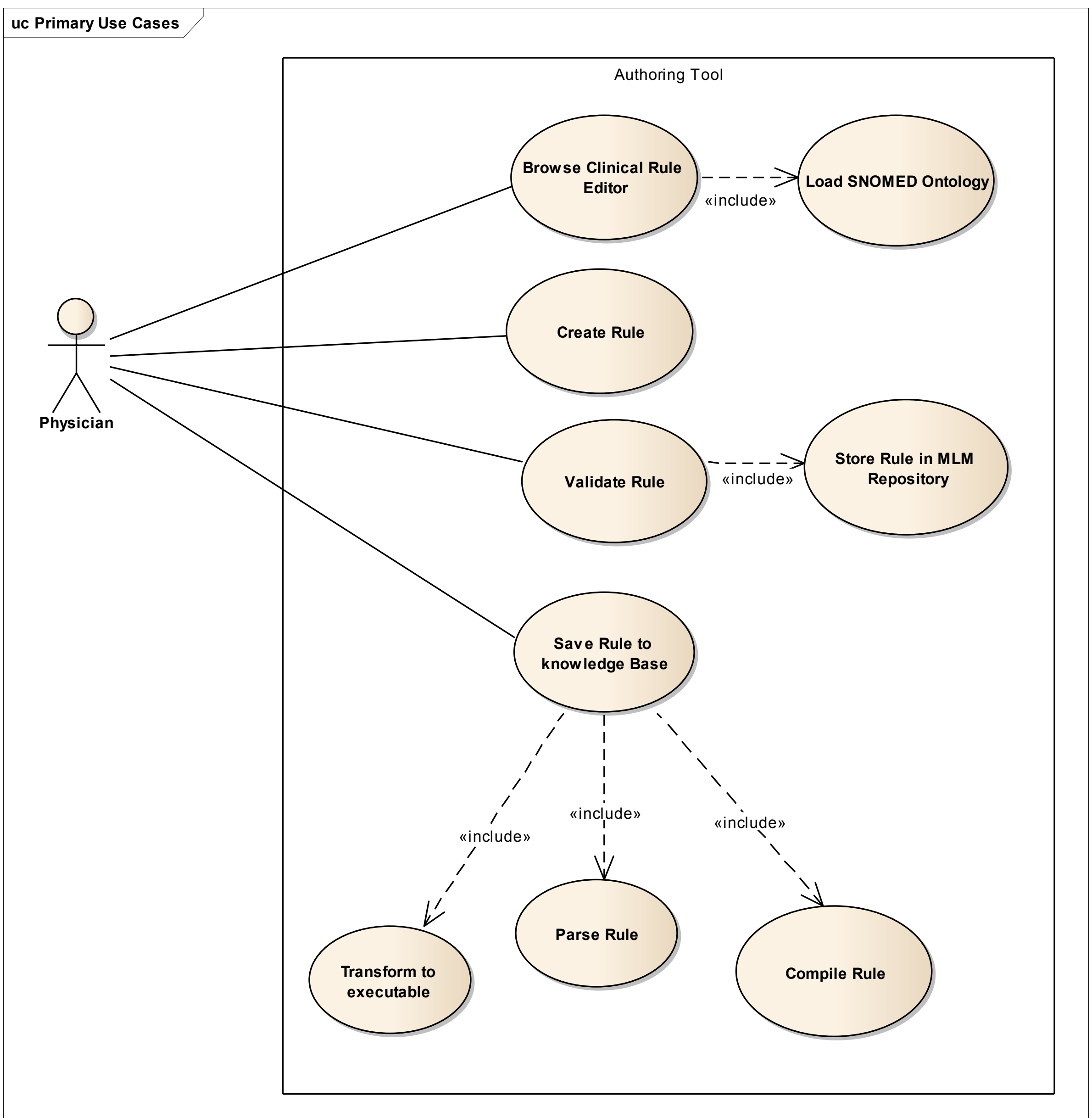
- Provides Intelli-sense to make the concepts available automatically during rule creation/modification.

4.6 UML Diagrams

4.6.1 Use Case Model of Authoring Tool

Actors

There is only one actor Physician for Authoring Tool. Physician interacts with system to create maintain rules and enhance the knowledge of Knowledge Base.



4.7 Authoring Tool Use Cases Brief Description

4.7.1 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.

4.7.2 Create Rule

In this use case the physician write the Rule on editor using the loaded ontology.

4.7.3 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.

4.7.4 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

4.7.5 Parse Rule

Before saving the rule to knowledge base the MLM is divided into small categories like Maintenance, Library and knowledge.

4.7.6 Transform to executable

For the execution purpose the categories of MLM transformed to executable code here we have C# code.

4.7.7 Compile Rule

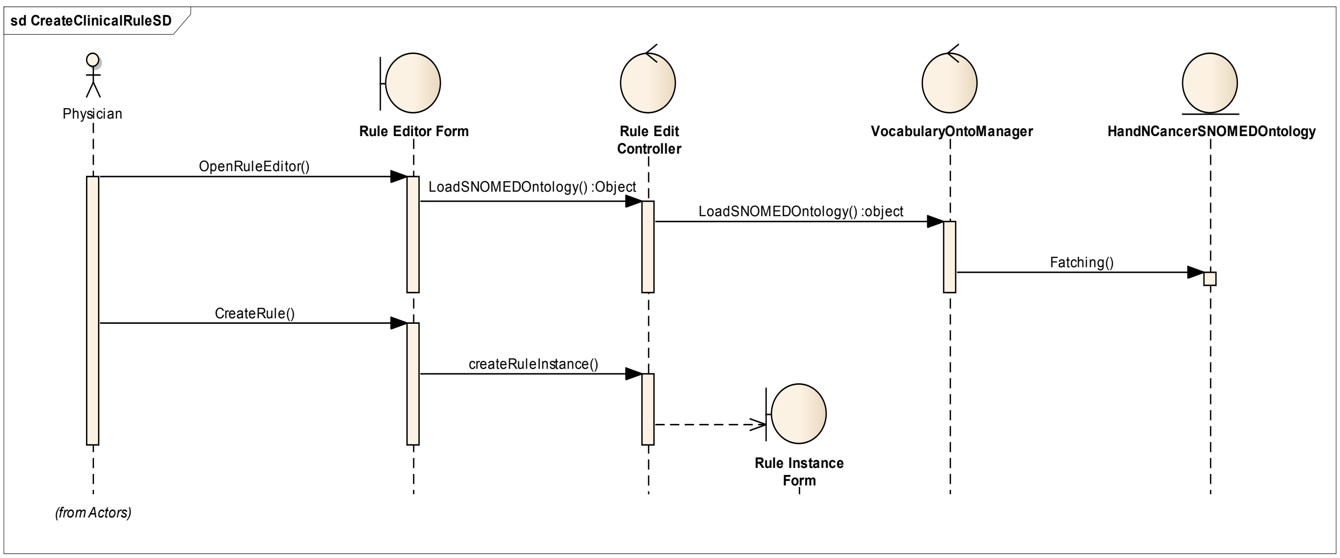
After generation of executable classes, this use case will actually execute it and save this exe file into Knowledge Base for our decision making by smart CDSS.

4.8 Authoring Tool Interaction Model

The objective of Interaction Model of Authoring Tool System is to show the process flow of objects that interacts with each other. Following are three basic interaction models;

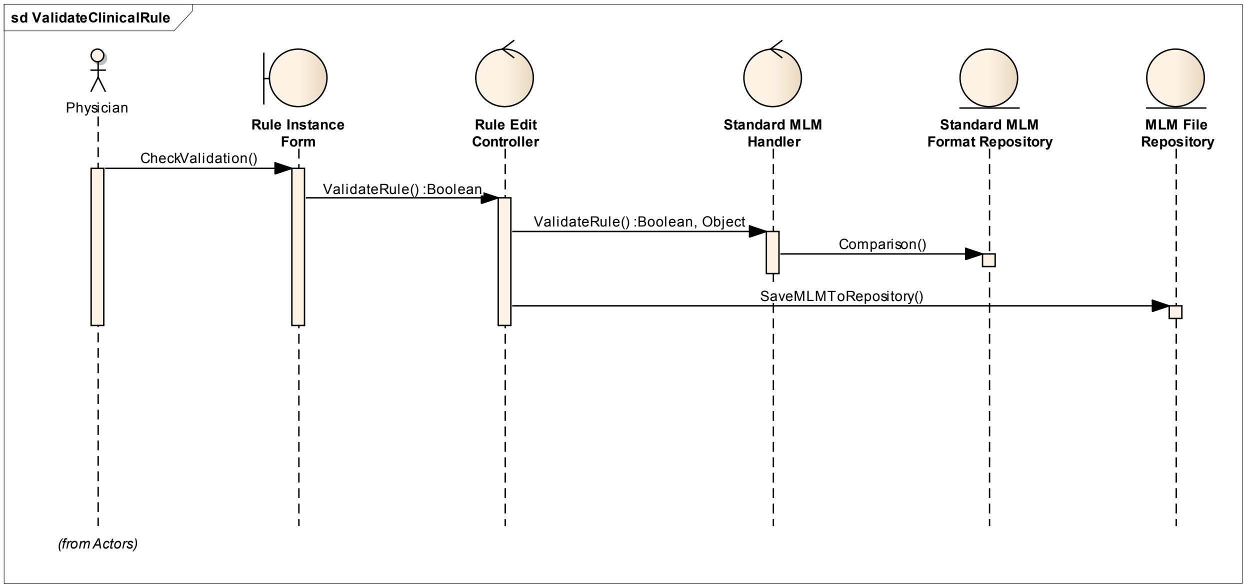
4.8.1 Create Clinical Rule

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.



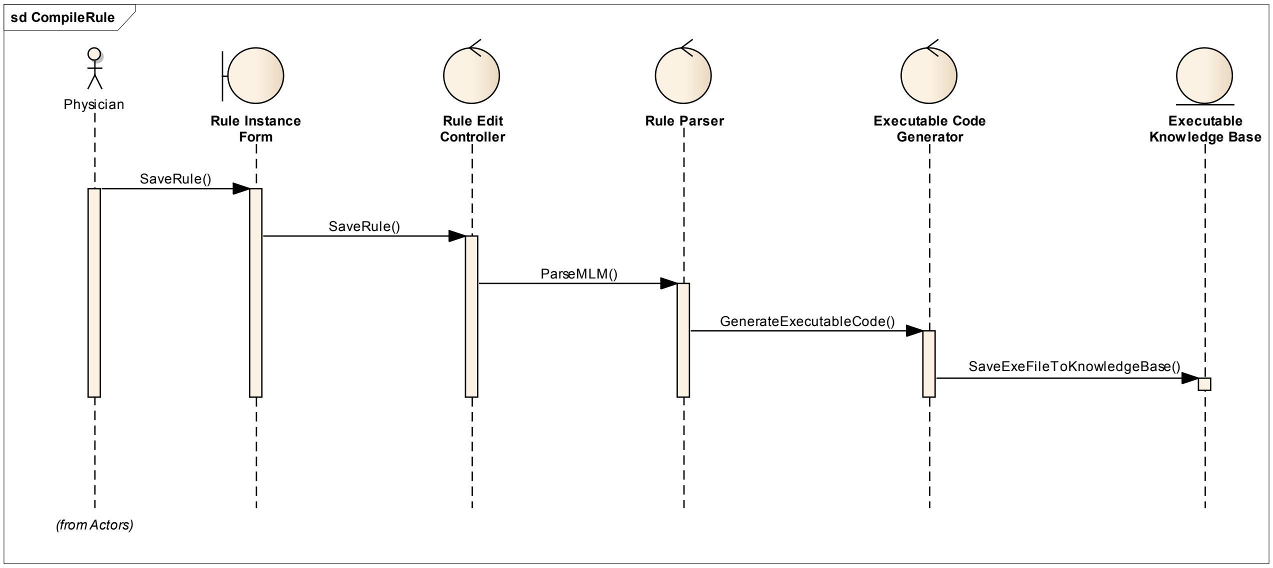
4.8.2 Validate Rule

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;



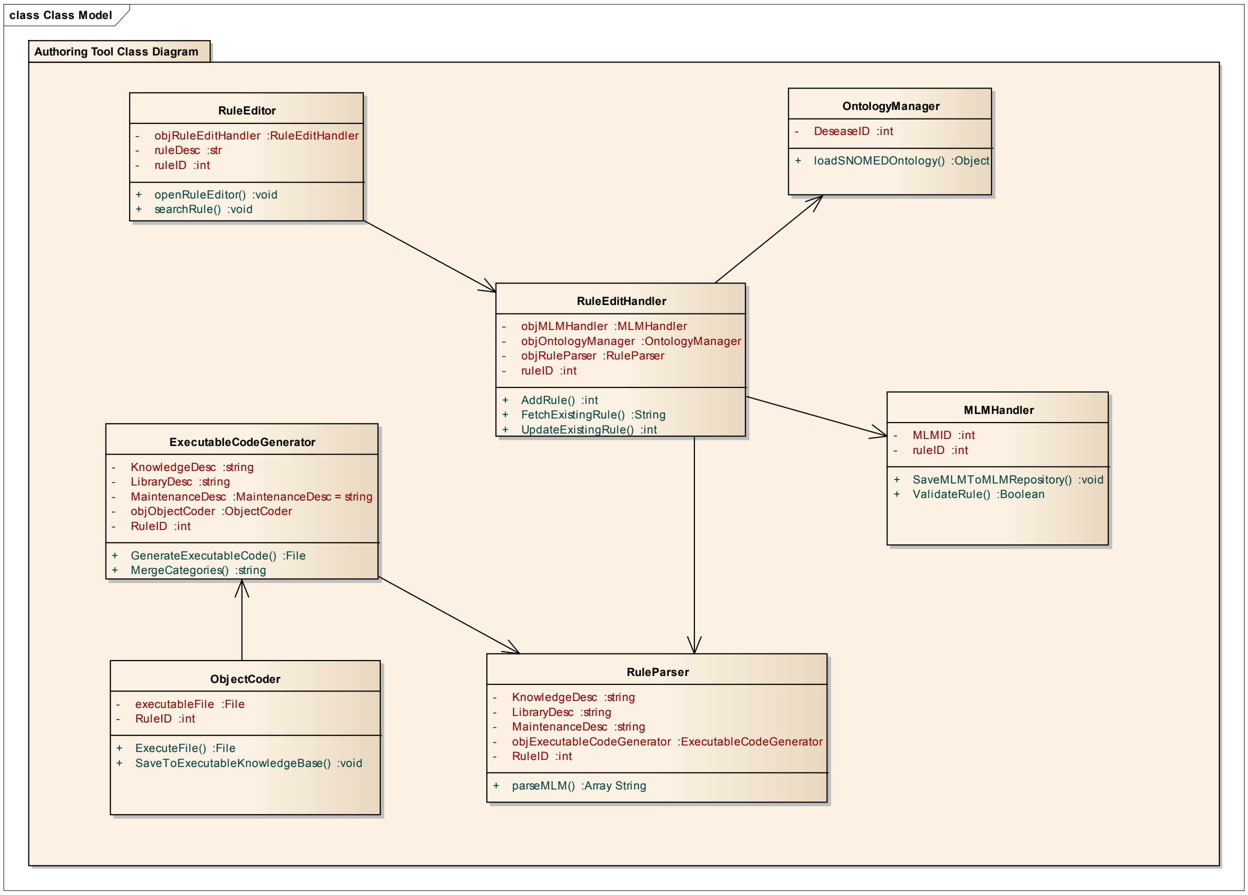
4.8.3 Compile Rule

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.



4.9 Class Diagram

The purpose of class diagram is to show the relationship among the classes of the system. Here the class diagram will show all the classes with attributes and operations. Following diagram shows the higher level of class diagram of Authoring Tool. This diagram contains seven different classes with their attributes and operations, the arrows show the association among these classes.



4.10 Component Model

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 Base as shown in following Figure.

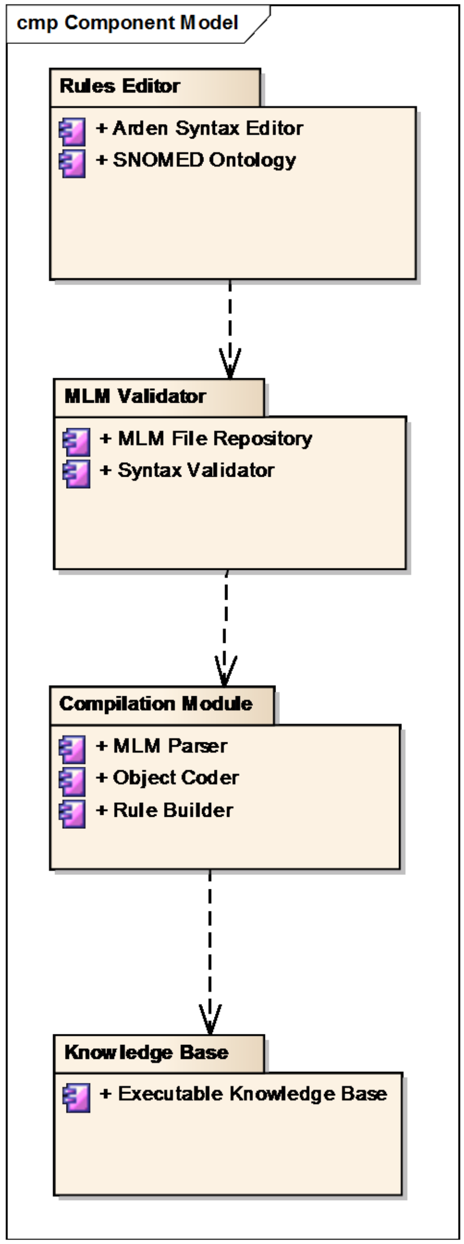
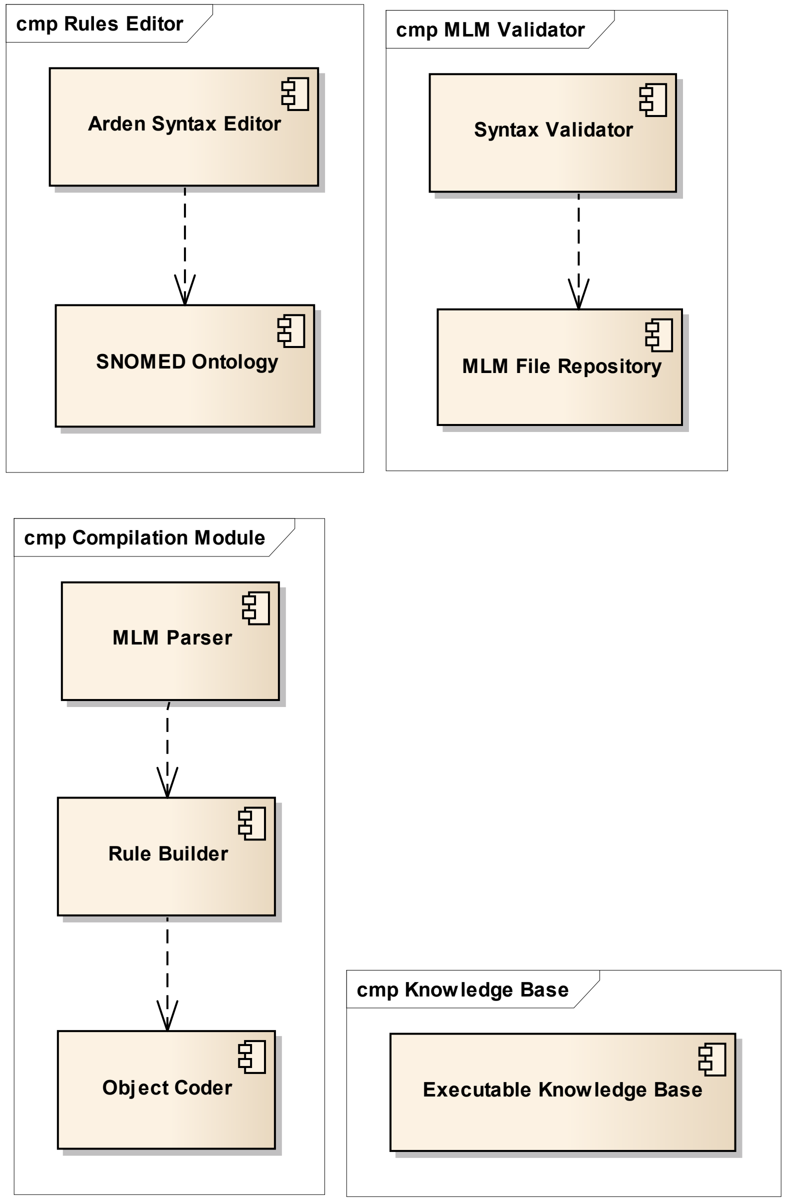
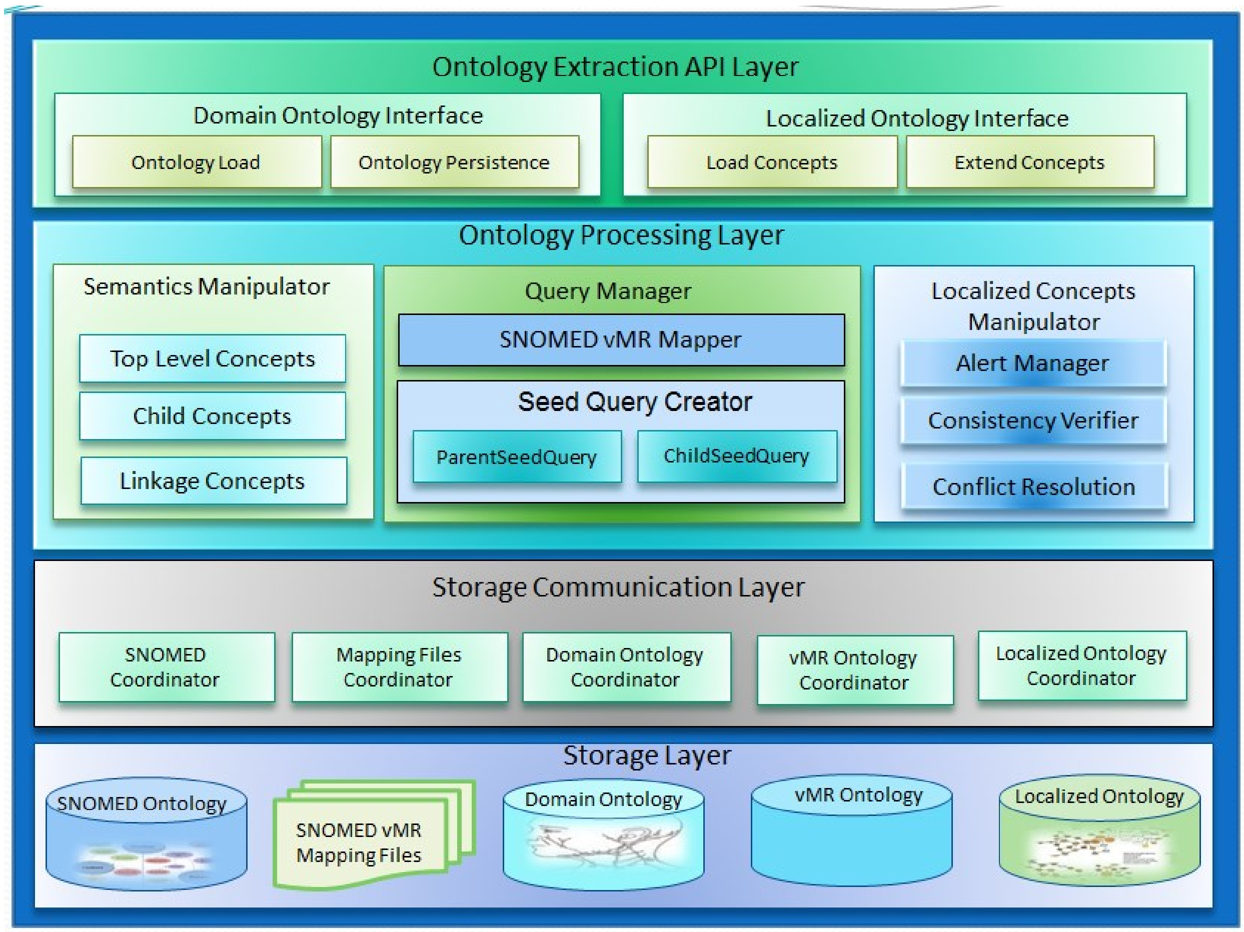


Figure depicted above, shows the four main components of Authoring Tool, but in detail level there are some sub components of main component like Rule Editor has subcomponents of Arden Syntax Editor and SNOMED Ontology. Following are the diagrams of subcomponents of the main components.


cmp Rules Editor
Arden Syntax Editor
SNOMED Ontology
cmp MLM Validator
Syntax Validator
MLM File Repository
cmp Compilation Module
MLM Parser
Rule Builder
Object Coder
cmp Knowledge Base
Executable Knowledge Base

4.11 Domain Ontology Extraction Module



The overall SNOMED CT ontology contains more 0.3 million concepts and each domain will use only 10 – 15 % of concepts of all. We have provided an interface to extract domain ontology. The above system architecture is using to extract domain on as well as manage the localized ontology.

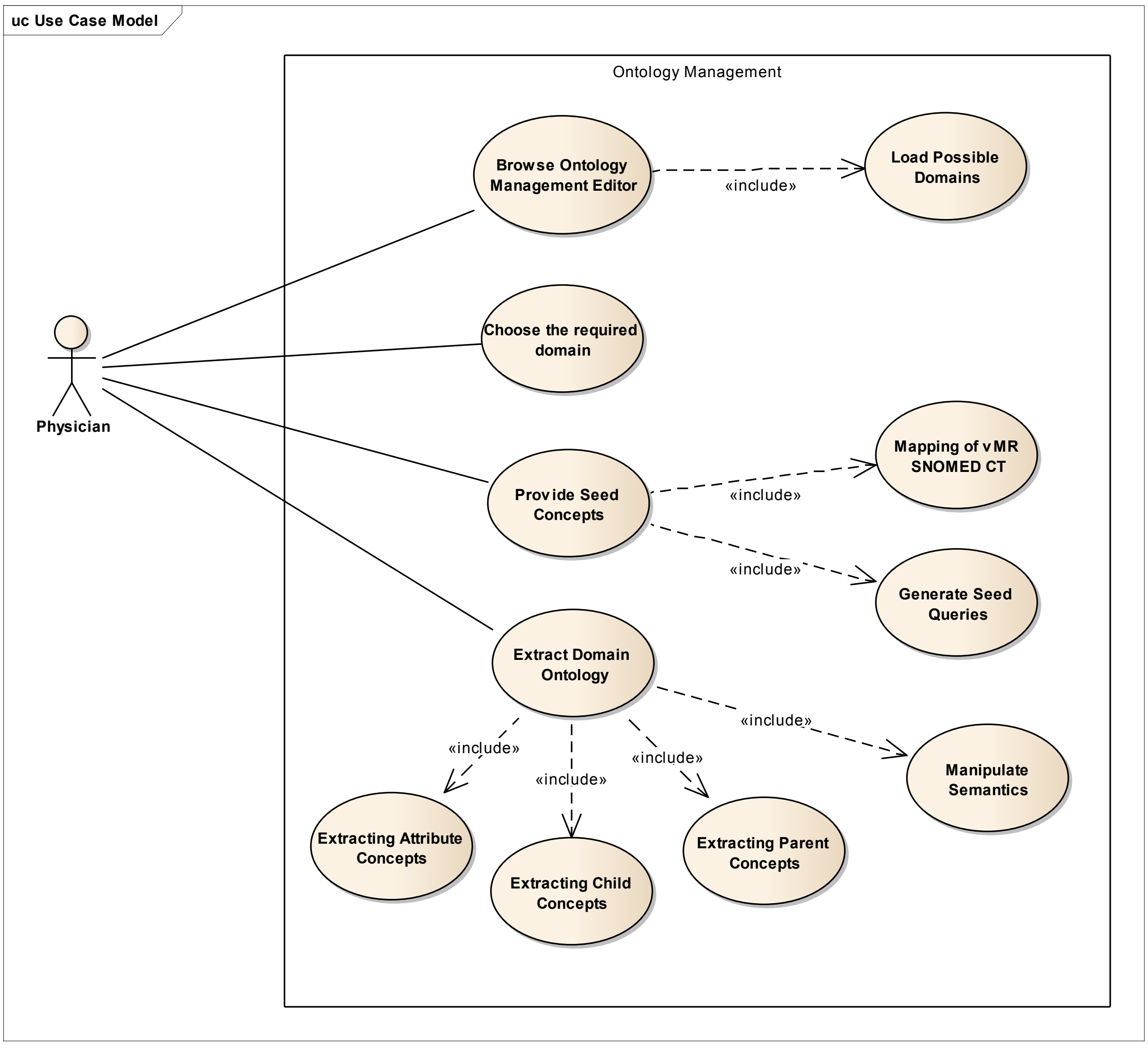
4.12 Uniqueness

- Automatic extraction of key as well as the relevant concepts to domain ontology
- Boundary detection using vMR Schema, with help of mapping between vMR schema classes and top level concepts of SNOMED CT
- Maintaining localized ontology when concepts do not exist in the domain ontology.

4.13 Use Case Model of Domain Ontology extraction

Actors

There is only one actor Physician or any other domain expert. The physician will interact as main actor to extract the domain ontology.



4.14 Domain ontology Use Cases Brief Description

4.14.1 Browse Ontology Management Editor

To extract domain ontology the physician or any other domain experts opens the ontology management editor. This editor will provide the user friendly environment to experts internally it shows all the possible domains

4.14.2 Choose the Required Domain

The physician will select a specific domain for that he/she wants to extract domain ontology. Suppose in this case the physician will select Head and Neck Cancer domain.

4.14.3 Provide Seed Concepts

In this case the physician will provide the seed concepts for extracting the domain ontology. The provided concepts are the base concepts of the domain. The system will extract all the parent, child and attribute concepts of the concepts. It internally uses two more cases.

4.14.4 Mapping of vMR and SNOMED CT

This module provides the mapping of vMR schema classes and top level concepts of the SNOMED CT. This mapper maps the attributes of each vMR class (49) classes with the top level concepts of SNOMED CT (19 concepts).

4.14.5 Generate Seed Queries

This module generates two types of seed queries, ParentSeedQuery and ChildSeedQuery according to seed concepts provided by the expert physicians. ParentSeedQuery identifies the top concept of hierarchy of seed concept. And ChildSeedQuery helps in extracting the child concepts of the provided seeds.

4.14.6 Extract the Domain Ontology

When the seed concepts will finalized by the physician then this module will extract the domain concepts using provided seeds and generated seed queries. This module internally uses four different cases.

4.14.7 Extracting Parent Concepts

This module extracts the parent concepts, start from child seed concept to parent seed concepts. And provides is-A links in between these concepts.

4.14.8 Extracting Child Concepts

This module extracts the child concepts of the provided seed concepts. It travers the ontology downward manner and extract only the child concepts of the seed concept. This process extract the child concepts up to the leaf concepts of that seed concept.

4.14.9 Extracting Attribute Concepts

Some concepts have also attribute concepts in SNOMED CT ontology. To manage and extract the attribute concepts of the provided seed concepts will extract using this module. This algorithm only extracts the attributes and childs of that attribute concepts.

4.14.10 Manipulate Semantics

After successful extraction of Child, Parent and Attribute Concepts; this modules create and generate the semantics and taxonomy in between these extracted concepts. After completion of this taxonomy the extract ontology stores to domain ontology of the system.

CHAPTER 5: KNOWLEDGEBUTTON: AN EVIDENCE SUPPORT TOOL FOR CDSS

5.1 Motivation

Healthcare domain is continuously growing with new knowledge emerged at different levels of clinical interest. Clinicians access the online resources on frequent basis for unmet questions during the course of patient care. The approach followed for resource utilization in the context of healthcare workflow is not well organized and integrated. The clinicians usually follow a dis-integrated approach to search for their required information from resources of their interest. Additionally they have no defined mechanism to reuse the searched information that is relevant and important. To overcome the limitation of inexplicit and disintegrated approach, we introduces the concept of "KnowledgeButton: An Evidence Support Tool for CDSS" in a well-defined and established manner. KnowledgeButton saves countable time of clinicians spent unnecessary in searching for evidences in the literature. The clinicians are interested to put an evidence for their knowledge in order to get confidence for the decisions. Not only this, but also the evidences can help in generating new knowledge of which the clinicians were not known before. In this way, the CDSS knowledge evolves with the passage of time and also the evidence support transforms the scope of knowledge from local to global.

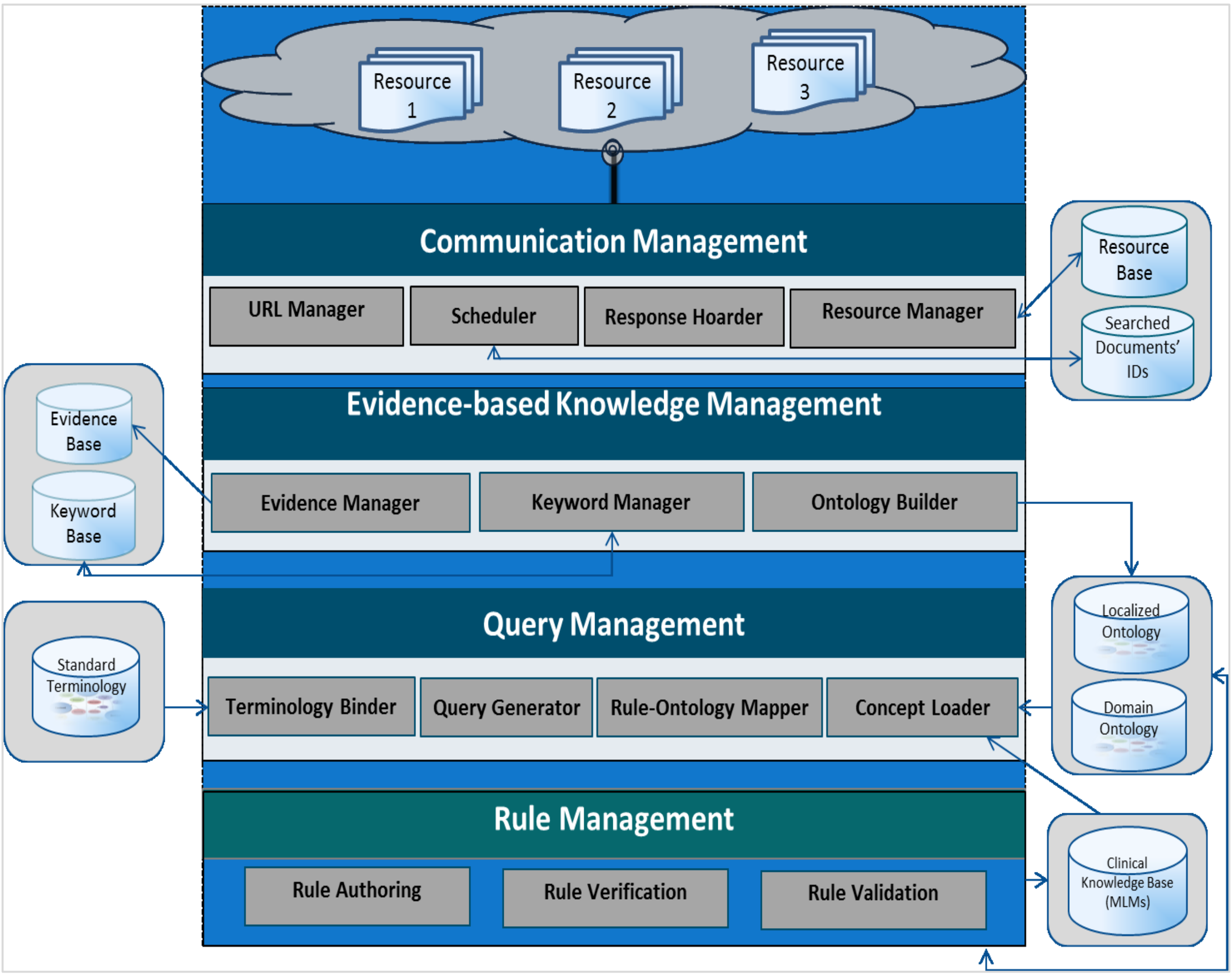
5.2 Goals

- To provide evidence support for Knowledge Rules of CDSS
- To keep physicians up-to-date with new research published
- To create an environment for evidence presentation and verification
- To create context aware interface to medical researchers for research analysis

5.3 Related Work

Generally, CDSS is an interactive computer-based information system that assists physicians and other healthcare professionals in the process of clinical decision making. Its goal is to reduce the time of physicians, spent on the clinical tasks that could otherwise be achieved through the use of CDSS. Any typical CDSS has three major components; KB, Inference Engine and User Interface [LFrenzel1986]. Among these three components, KB is the most important and is the success determiner for any contemporary CDSS [NHulse2005]. However there are several other desirable features that any contemporary CDSS should possess to satisfy the user requirements. Maintenance of the KB is one of these desirable features [LFrenzel1986] and the same is included in [DBates2003] as one of the Ten Commandments. Also this desirable feature is considered as a technical challenge and a barrier for implementation because of the ongoing research development in that domain. So most of the time, physicians referred to online resources in order to get the evidence for a specific health issue. To determine the need of online knowledge resources in diagnosis and treatment of HNC patient care, we asked several questions from the oncologists. Seeing the answers to these questions, we realized how much it is needed to have an easy but contextual access to the knowledge available online. However, utilization of online knowledge resources in a context rises to a number of research and implementation questions. The nature of input data and the workflows' complexity reflect the fact that as a researcher we need a deep understanding of the requirements as well as the objectives. Sometimes a CDSS system is strong from its capabilities but due to misfit into the workflow of a system, the chance of acceptance by the physicians is let downed. The reasons behind this lack of success included inadequate integration of CDSS into clinical workflow and inadequate integration of CDSS with EMR [AHolbrook2003]. Since, we are implementing this system into a real environment to link Health Information Management System (HMIS) of SKMCH&RC so we are confident to break this barrier of inadequate integration. HMIS of SKMCH&RC has very well matured patient workflows to manage cancer patients for their diagnosis and treatments. The system is lacking to integrate CDSS results in order to cover the time spent on tasks that would otherwise be done by the computer. From technical perspective we can use either standard based or non-standard based approach for this integration. Standard based approach is preferable over non-standard due the factor of eliminating the need for developing custom APIs with significant reduction in integration costs [GFiol2012]. HL7 developed Context-Aware Knowledge Retrieval Standard (Infobutton) [MGuilherme1997] is becoming popular and is considered for implementation by different entities around the world. There are two implementation guides provided by HL7 so far; URL based Implementation and Service Oriented based Implementation. In our approach we will prefer to use the later one. Infobutton only, however cannot satisfy the overall requirements. Our goal is not only to retrieve knowledge from the online resources and present them to the physicians in as-it-is format. Rather, we want to transform the retrieved information in a more easy to use manner and arrange them into a more logical format for the physicians to validate and generate rule from them.

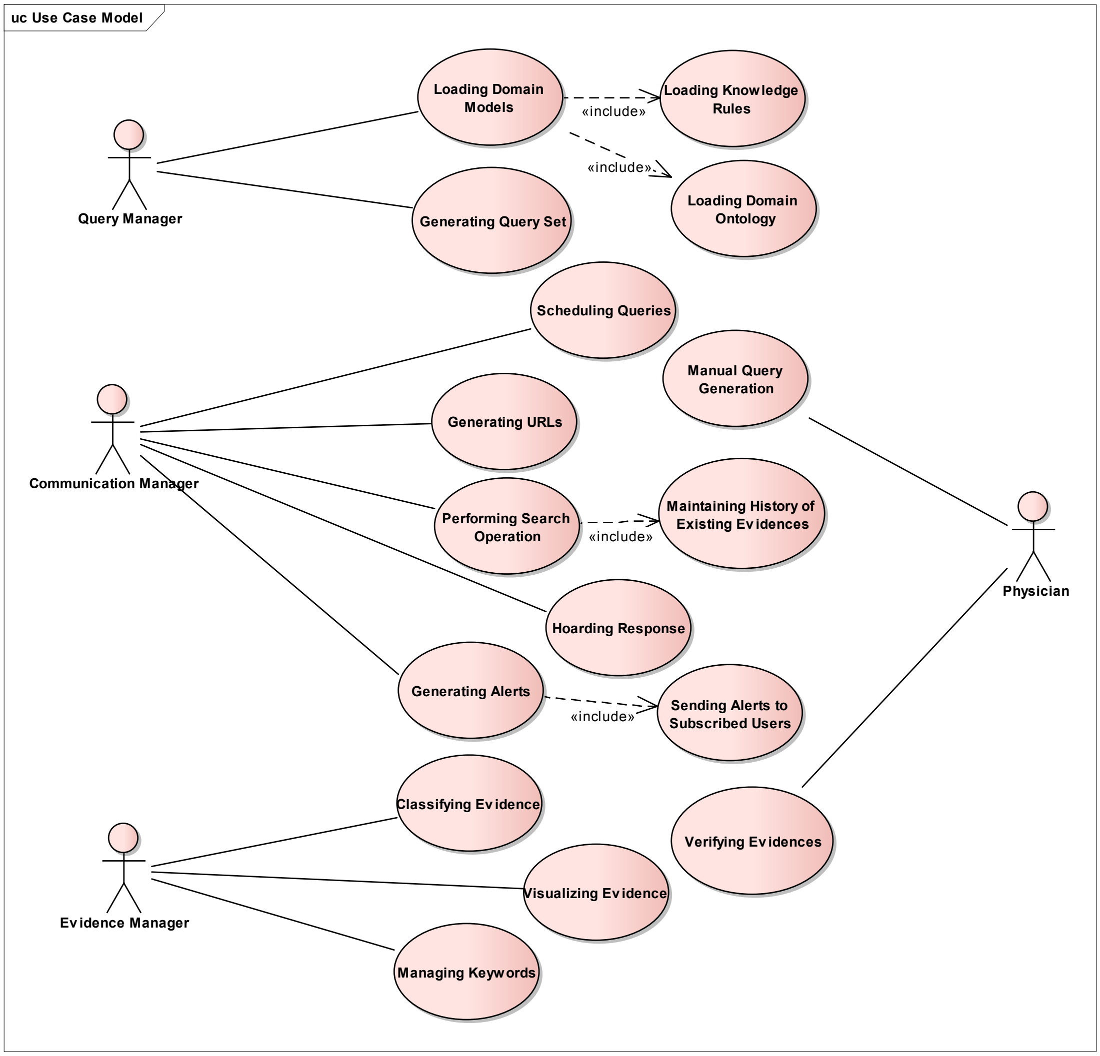
5.4 Architecture



5.5 Uniqueness

- Contemporary CDSS system are based on knowledge rules or case bases and evidence support is missing
- Online resource are utilized only up to the degree of information links in the context
- KnowledgeButton model is unique in its approach to collect and present evidences to the user to result in improvements of existing rules, helps in crafting new rules and keep the physicians up-to-date with current research

5.6 Use Case Model



Actors

There are four actors; three non-humans and one human.

- Query Manager: all the functions related to query generation are initiated by query manager.
- Communication Manager: functions such as scheduling queries, url generations, response hoarding and alert generation/sending comes under the responsibility of communication manager actor to initiate.
- Evidence Manager: this actor take care of evidence management related functions such as evidence classification and presentation.
- Physician: as a human actor, physician involves in manual query generation and evidence verification.

5.7 Use cases brief description

5.7.1 Loading Domain Models

Domain models include domain ontology and knowledge rules. This use case is required to load both of the models for query generation purpose. Domain ontology covers the domain concepts and relationships while knowledge rules represent the rules in knowledge base of a CDSS.

5.7.2 Generating Query Set

Generation of the queries from the set of knowledge rules and domain ontology is the responsibility of this use case. The queries are generated in three possible ways; rules based, ontology based and hybrid. The final query set is selected based on the evaluation as which approach generates queries with better results. This whole process of query generation is fully automated without human interventions.

5.7.3 Scheduling Queries

All queries to run at once can create a large set of results which can aggravate the management of response. This use case schedule the query set by dividing the large set of queries into smaller set. The division is based on topics in a domain and time selection. In head and neck cancer, the topics refer to the sites (oral cavity, larynx, pharynx etc.).

5.7.4 Generating URLs

The generated queries need to be transformed into proper URL format to be accepted by the target search engine. For PubMed searching, an Entrez Utility program (eUtils) provides URL in a specific format. The base URL is given as: *http://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=pubmed*.

5.7.5 Performing Search Operations

The URL are passed to search functions provided by eUtils server programs. There are seven functions eUtils provides:

- eSearch: use to perform basic search which returns the list of Document IDs matching the query
- eSummary: takes the list of IDs and returns the document summaries
- eFetch: take the list of IDs and returns formatted data records as specified
- eLink: take the list of IDs and returns the linked IDs from source and destination databases
- eInfo: provides statistics on input database
- ePost: accepts a list of UIDs, stores the set on the History Server, and responds with the corresponding query key and Web environment.
- EGQuery: responds to a text query with the number of records matching the query in each Entrez database.

5.7.6 Hoarding Response

The response returned by running any of the searched options needs to be formatted for the further manipulation. The IDs of the returned articles are saved for track recording. Next time when queries run, the saved IDs will be used input request to filter out the already searched articles.

5.7.7 Generating Alerts

Since the process of generating queries and getting response is automated and there is not direct involvement from the users. Users don’t know unless the system notifies them. This use case creates alerts on the results and notifies the subscribed users.

5.7.8 Classifying evidences

The retuned set of articles is considered as evidences. They need to be classified with respect on the basis of importance and relevance. The most important and relevant evidence is displayed on the top to catch the user attention in the first attempt.

5.7.9 Visualizing evidence

The classified evidences are presented to the users in relationship to knowledge rules. For example, if classified evidence set has 10 evidences so these 10 evidences are presented to the user along with relevant rules so that the physician can look both rules and evidences at one place.

5.7.10 Verifying evidences

It is not necessary that all the evidences classified by the system are meaningful for the users. It is on physicians' discretion to approve or dis-approve evidence based on its importance and relevancy. The approved evidences are saved to the repository as research evidences for knowledge rules.

5.7.11 Managing keywords

New words and terminologies are introduced from the research articles which need to be managed for future queries enrichment. We named these words and terminologies as keywords extracted from the articles on the basis of frequency of occurrence.

5.7.12 Generating Manual Queries

Sometimes, users want to write their own queries rather to rely on system generated queries. In that case, system provides contextual information of the domain for the users to select information items in order to generate manual queries. The system also tracks record of previously written queries by the same user or different users.

5.8 Interaction Model Diagram

The objective of interaction model is to provide process flow of the system through involved objects. In KnowlegeButton system, there are two scenarios; push model scenario and pull model scenario.

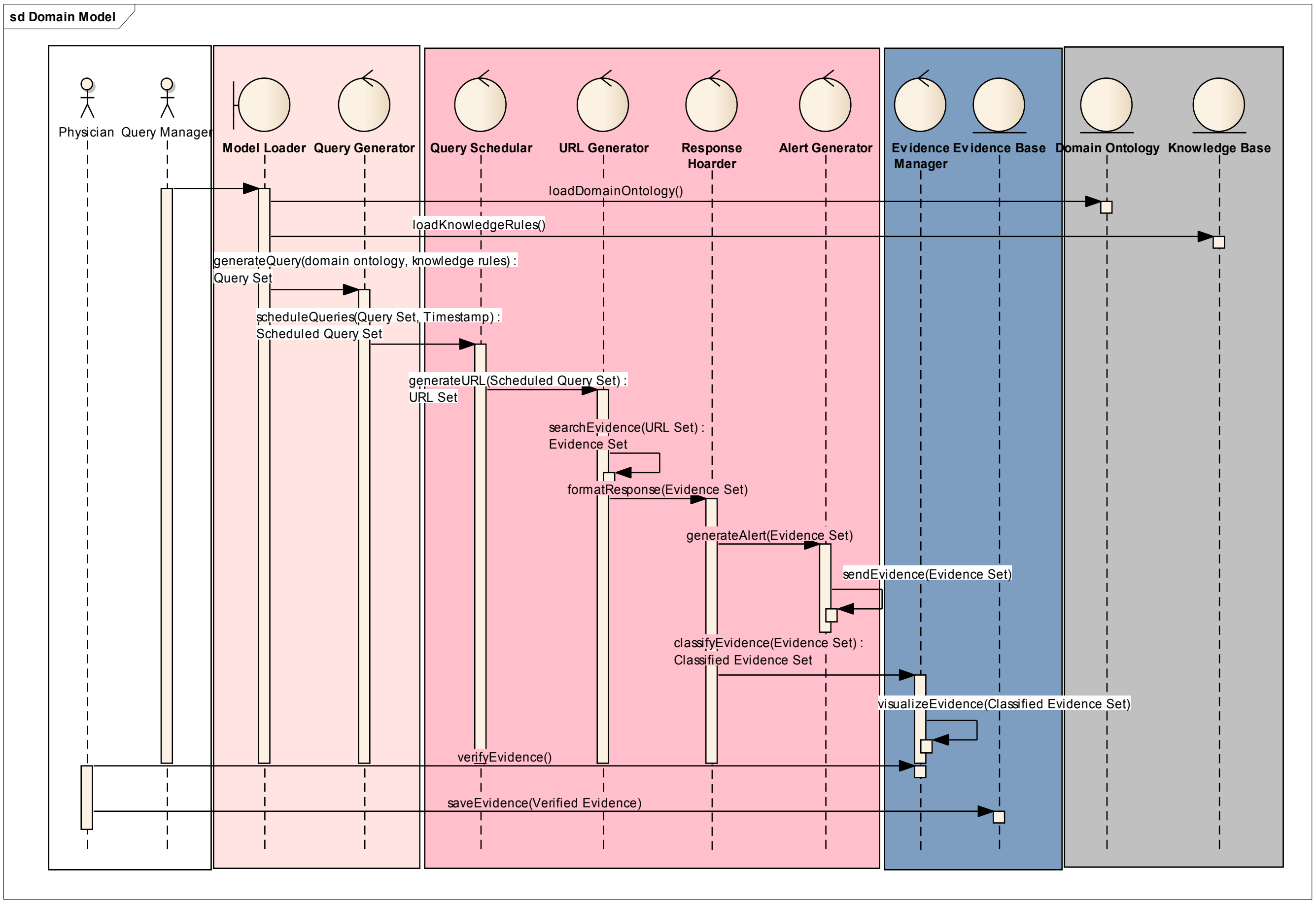
5.8.1 KnowledgeButton overall push and pull sequence diagram

In push model, the system initiates the process by itself with the help of Query Manager Actor and interacts in following fashion:

- Query Manager activates the Model Loader object in order to load the two models; domain ontology and knowledge base rules
- Model Loader passes control to the Query Generator for query set generation on the bases of models loaded in previous step.
- The control then passes to the Query Scheduler object to schedule the queries passed through scheduleQueries() method.

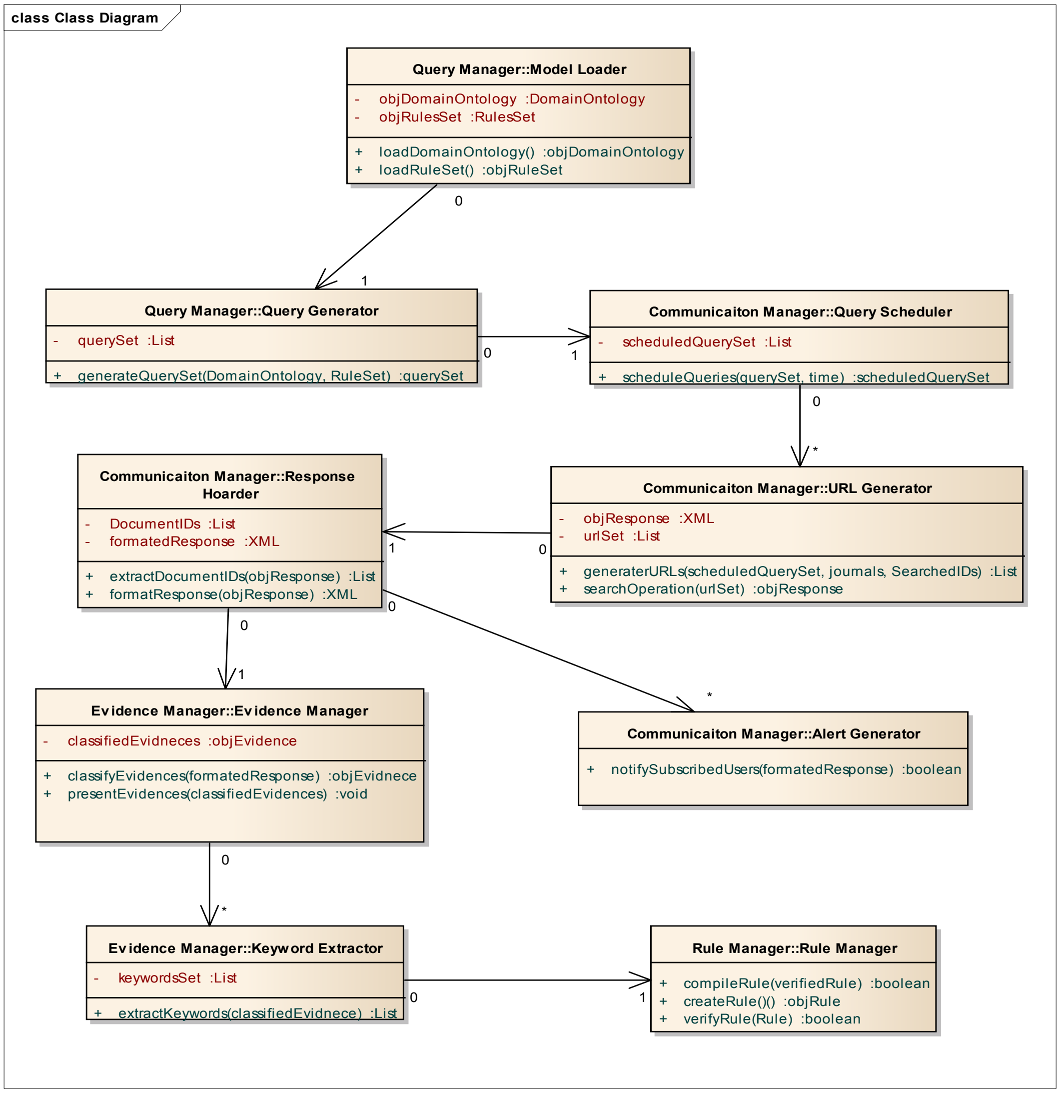
- The scheduled query set is passed to URL Generator object in order to generate the urls for the target resource. On the bases of urls different search operations (provided by Entrez eUtils Programs) are performed.
- The search operation response is held by Response Hoarder object to format it for further manipulations.
- At the same time, Alert Manager Object generates the alerts for subscribed users to notify them about the new results.
- Response Hoarder Object passes the control to Evidence Manager Object to classify and present the evidence to the user. The presentation object also takes care of the relevant rules to present in association to the evidence in order to provide coherent view of both evidence and knowledge rules. In the same step, Keyword Manager extracts the most frequent and important keywords from the research article for future us in query enrichment.

In pull model, the process starts by the physicians themselves to create queries manually. Query Editor Component provides the environment for query creation by showing domain information for easy to use selection of query terms. After query creation, the rest of the steps remain the same as that of push model except the two objects i.e. Query Scheduler and Alert Generator which are not required in pull model.



5.9 Class Diagram

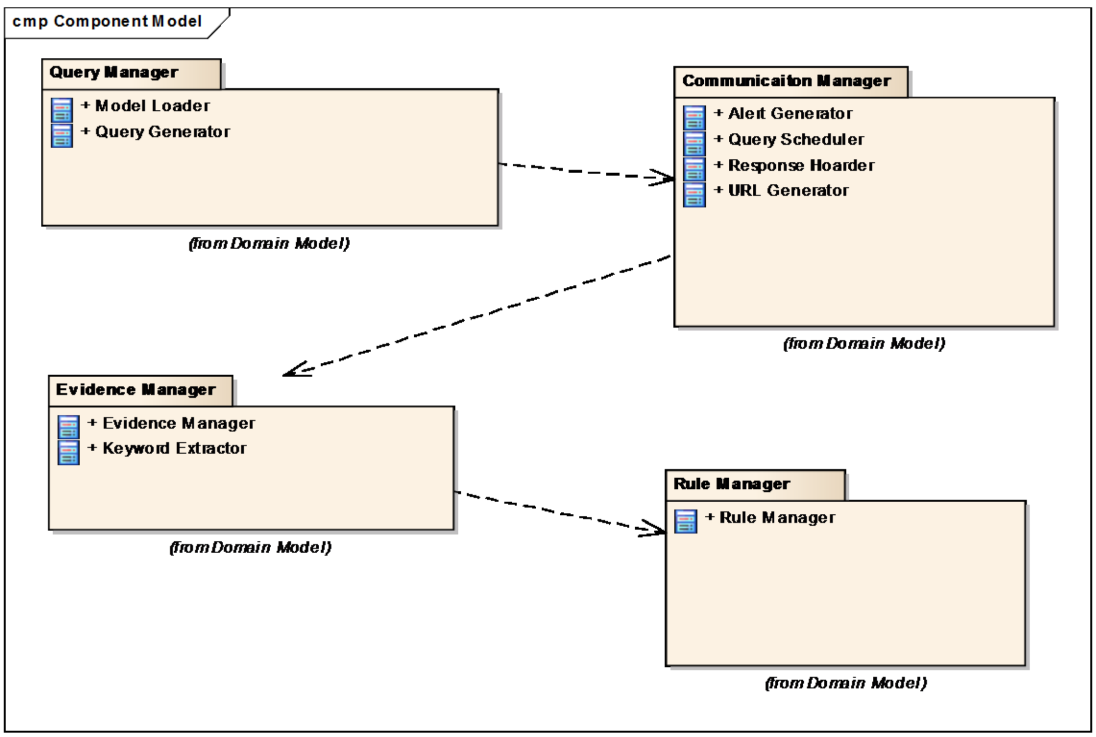
Class diagram represents the structured behavior of the system. Each class in the model has properties represented with attributes and behavior represented with operations. In Model Loader class we have two operations; loadDomainOntology() which loads the domain ontology and assigns to the objDomainOntology object and loadRuleSet() which loads rules from the knowledge base. Query Generator class generate querySet using its operation generateQuerySet(). The Query Scheduler schedules query with operation scheduleQuerySet and assigns it to the scheduledQuerySet list attribute. URL Generator utilizes the set of scheduled queries and generate a set of urls with the help of generateURL() operation. Response Hoarder class has the operation as formatResponse() which formats the received response from search operation and extract the document ids for future use. Alert Generator generates the alerts and notify the subscribed users via notifySubscribedUsers() operation. Evidence Manager uses two operations as classifyEvidences() which classify the retrieved set of evidneces according to the topics in the domain and presentEvidences() which presents the evidneces to the users to view in association with the knowledge rules. Keyword Manager extracts the most frequent and important keywords from the classified set of evidences using extractKeywords() operation.



5.10 Component Diagram

A component diagram packages related classes into components. KnowledgeButton system has four components:

- Query Manager: It has two classes i.e. Model Loader and Query Generator
- Communication Manager: It has four classes; Query Scheduler, URL Generator, Response Hoarder and Alert Generator.
- Evidence Manager: It has two classes; Evidence Manager and Keyword Extractor.
- Rule Manager: Only one class belongs to this component i.e. Rule Manager.



CHAPTER 6: CASE STUDY: HEAD AND NECK CANCER TREATMENT

6.1 Overview

The case study covers the treatment recommendation for oral cavity and salivary gland. Furthermore, we also incorporated the evidence support for generated recommendations.

6.2 Clinical Guidelines

6.2.1 NCCN Guideline for Oral Cavity

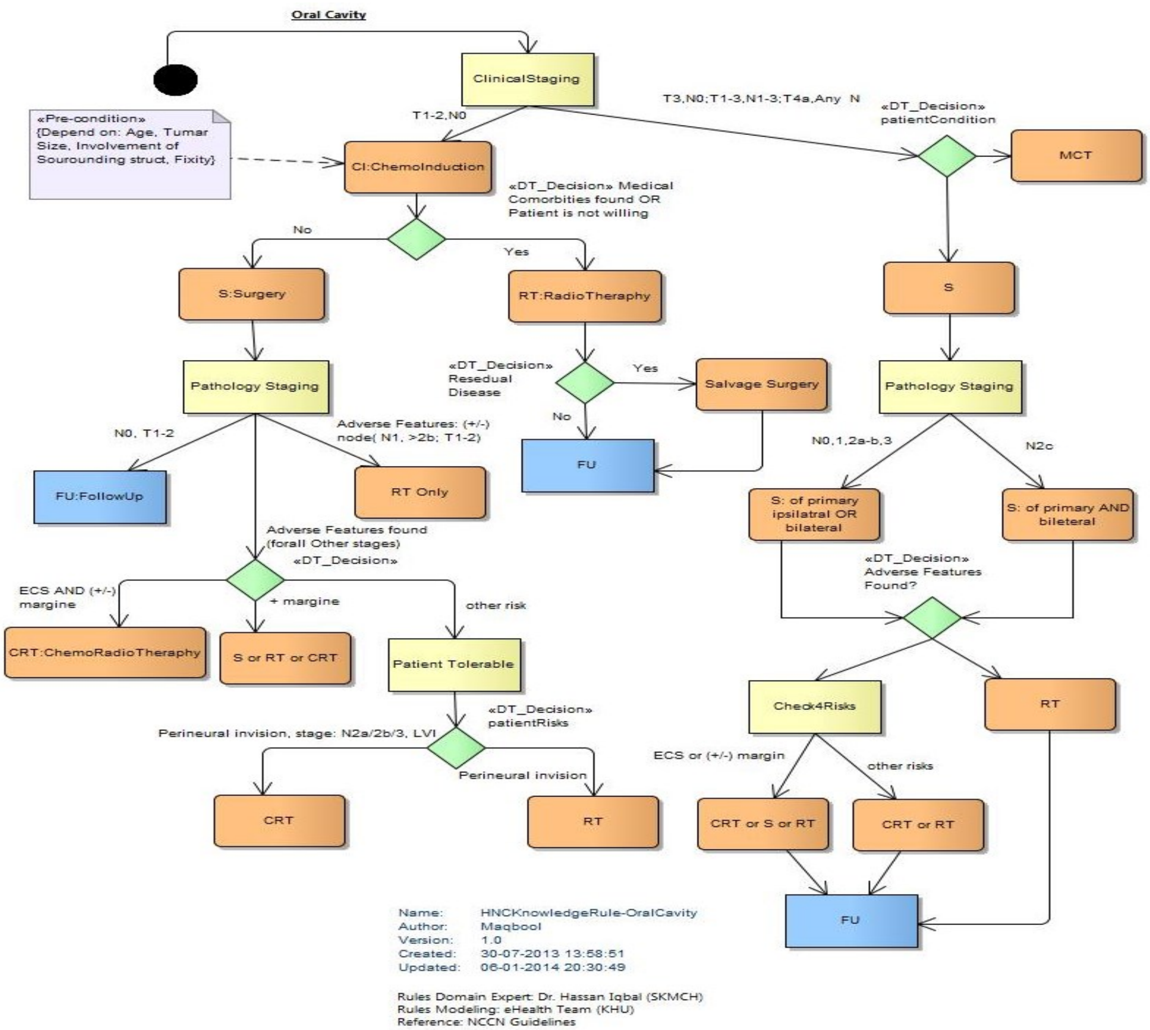


Figure 6.1: Oral Cavity NCCN Guideline

6.2.2 NCCN Guidelines for Salivary Gland

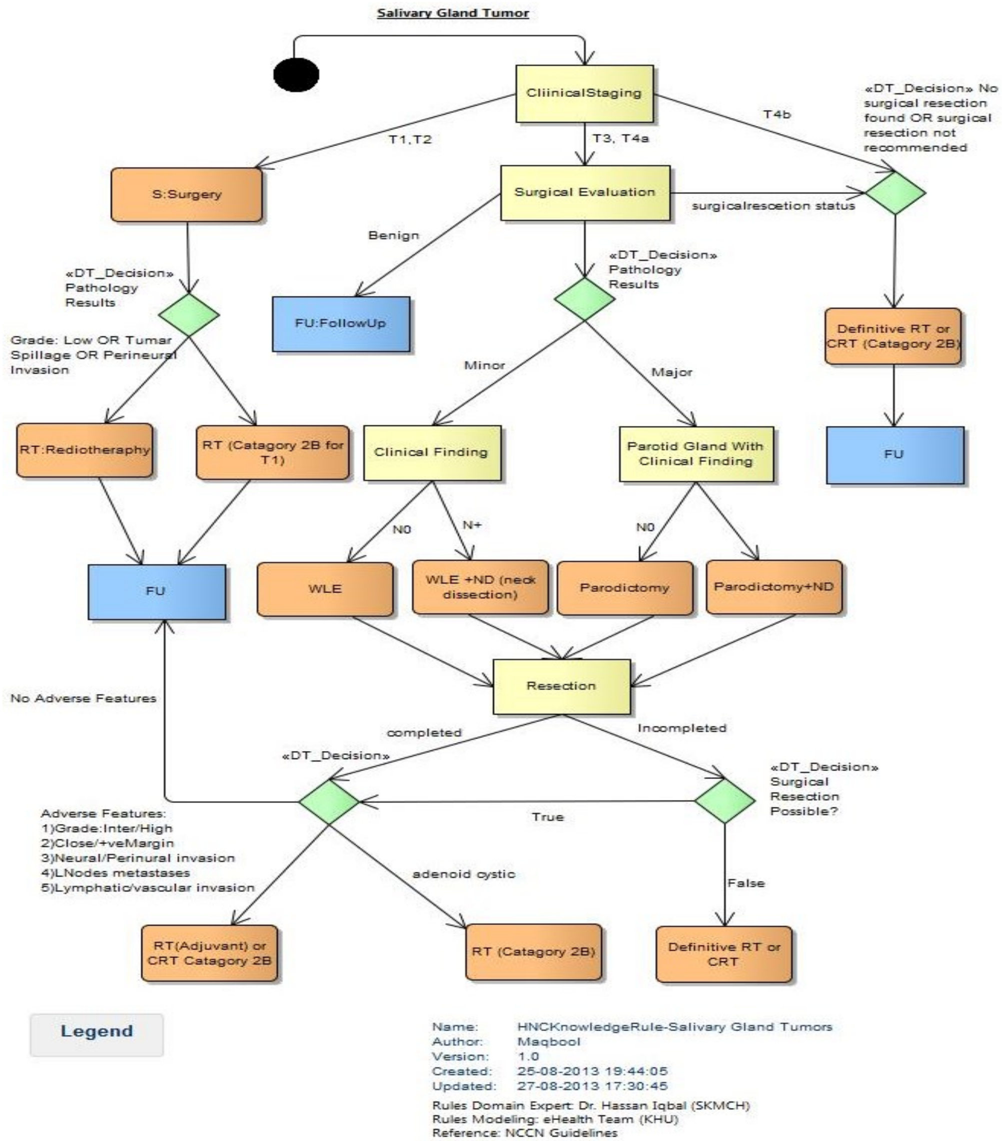


Figure 6.2: Salivary Gland NCCN Guideline

6.3 Recommendation Scenario for Oral Cavity

Oral Cavity treatment recommendations are based on patient clinical staging, treatment already done, patient observations, pathology staging and adverse features.

For patient with clinical staging T1 and T0, radiotherapy is done in first encounter; the Smart CDSS intervention will provide recommendations for next follow up. Figure 6.3, depicts recommendation for above symptoms.

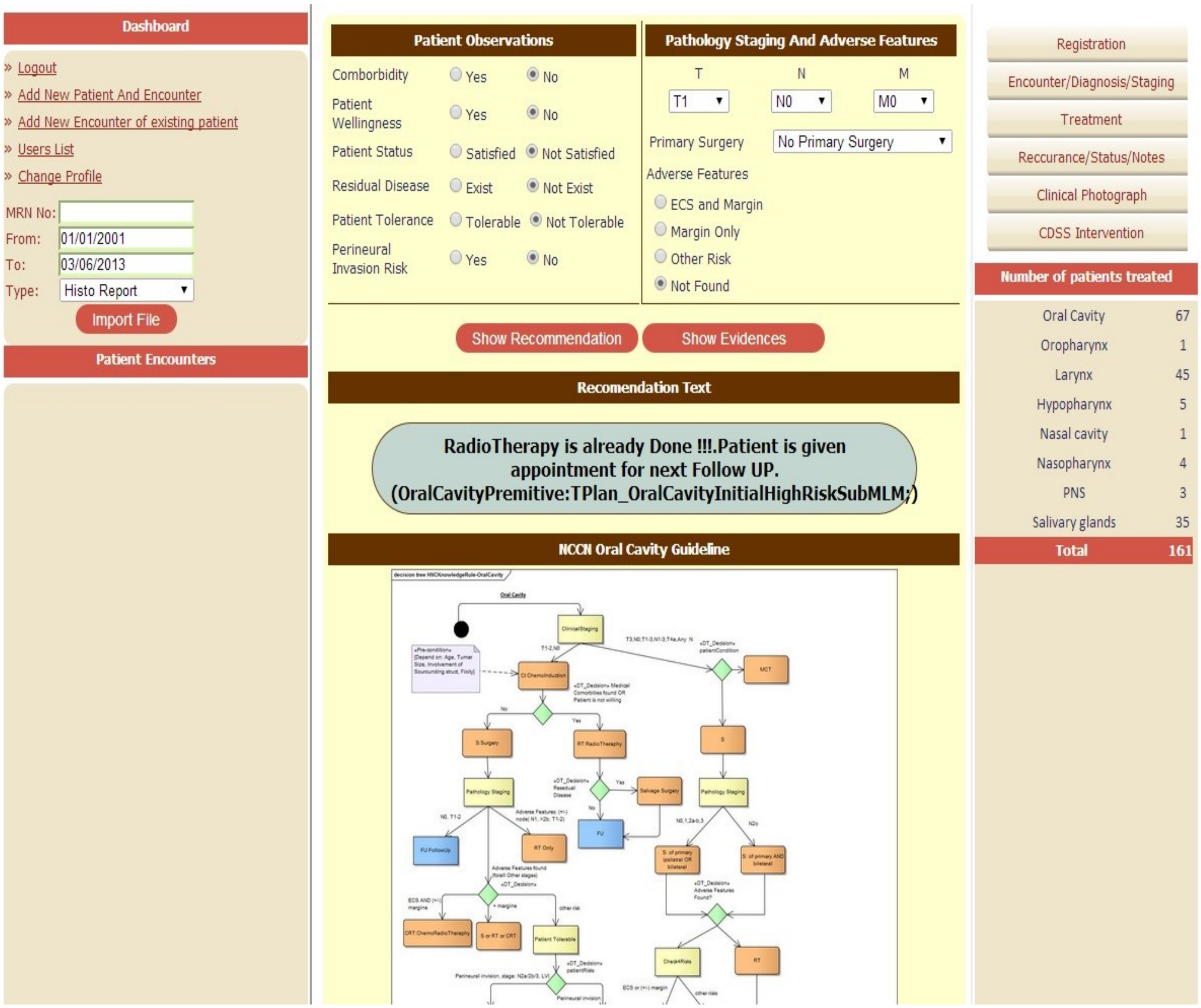


Figure 6.3: Smart CDSS Treatment Recommendation (Oral Cavity)

6.4 Evidences for Oral Cavity Recommendation

Oral Cavity treatment recommendations are provided with associated research published recently. For given recommendation as shown in Figure 6.4, the KnowledgeButton tool search PubMed to retrieve all related research.

For given recommendations, with associated MLMs of recommendations are parsed and query is generated based on relevant concepts used and submitted to online PubMed. The resultant online published resources that can be made as evidence are shown in Figure 6.4.

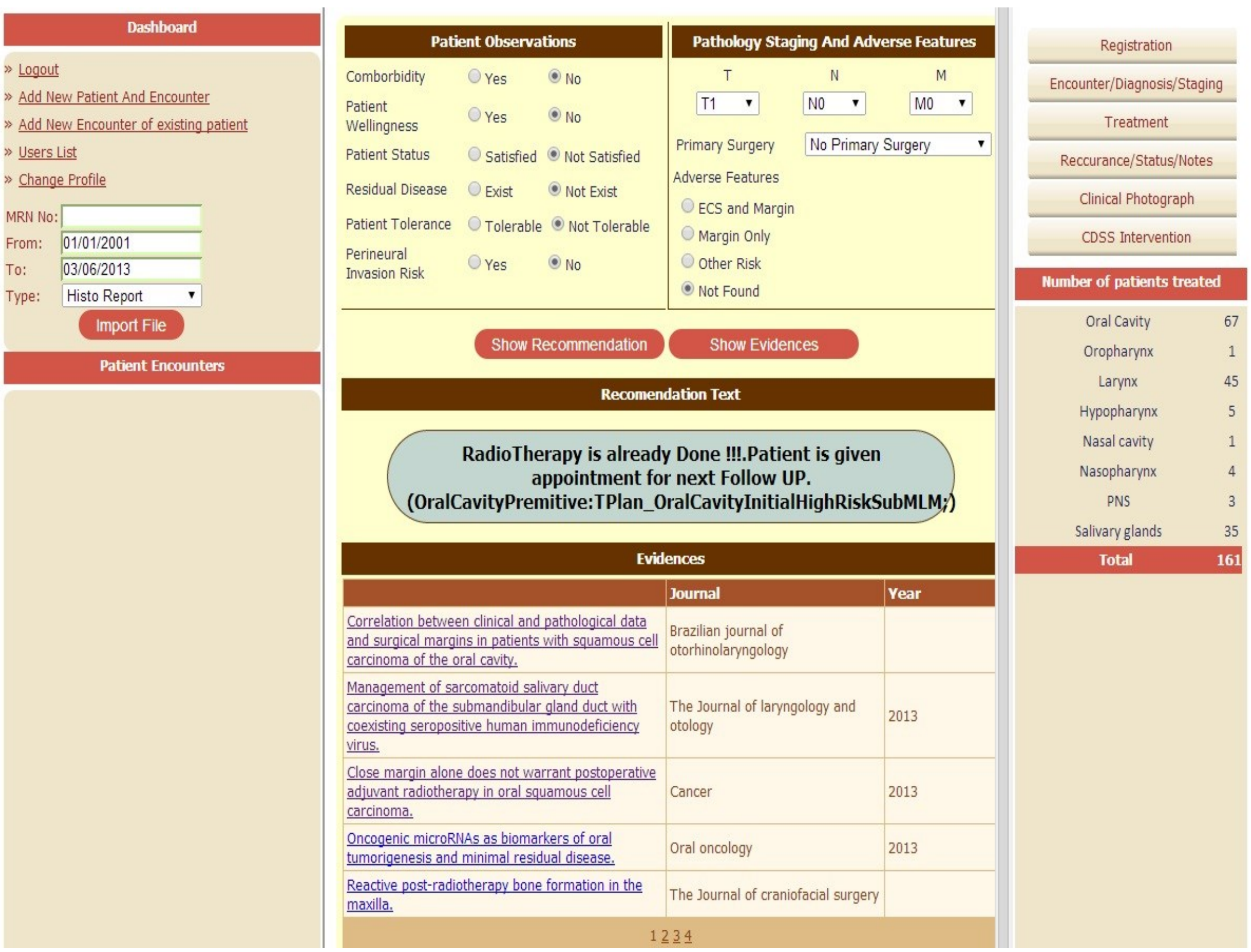


Figure 6.4: Smart CDSS Evidence Support for Oral Cavity recommendations Using KnowledgeButton

6.5 Recommendation Scenario for Salivary Gland

Salivary Gland treatment recommendations are based on patient clinical staging, treatment already done, patient observations, pathology staging and adverse features.

For patient with clinical staging T1 and T0, with Surgery already done and pathology results are major; the Smart CDSS intervention will provide recommendations for Radiotherapy. Figure 6.5, depicts recommendation for above symptoms.

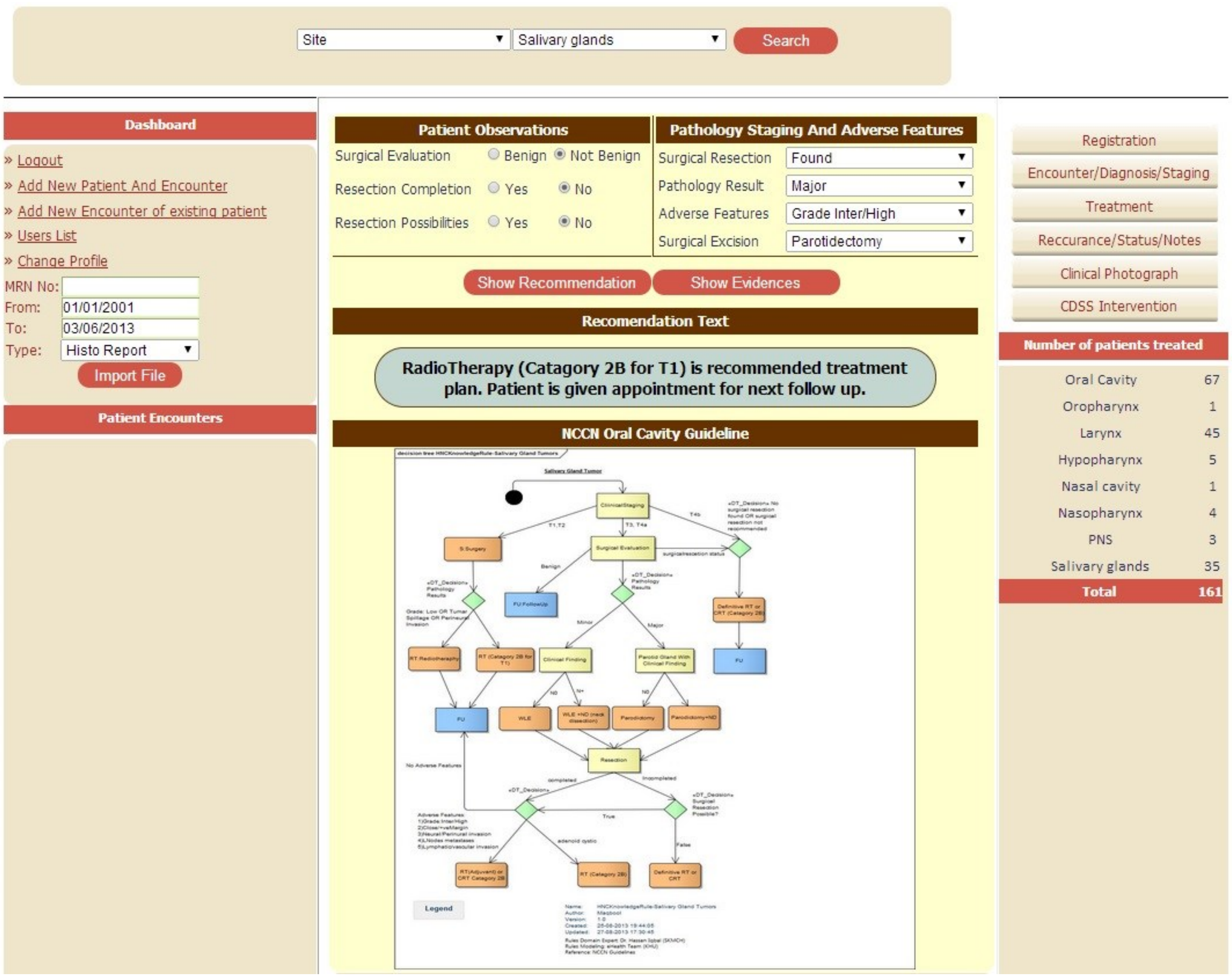


Figure 6.5: Smart CDSS Treatment Recommendation (Salivary Gland)

6.6 Evidences for Salivary Gland Recommendation

In order to associate relevant resources for recommendations of Salivary Gland, the KnowledgeButton tool is used to search PubMed to retrieve all related research.

For given recommendations, with associated MLMs of recommendations are parsed and query is generated based on relevant concepts used and submitted to online PubMed. The resultant online published resources that can be made as evidence are shown in Figure 6.6.

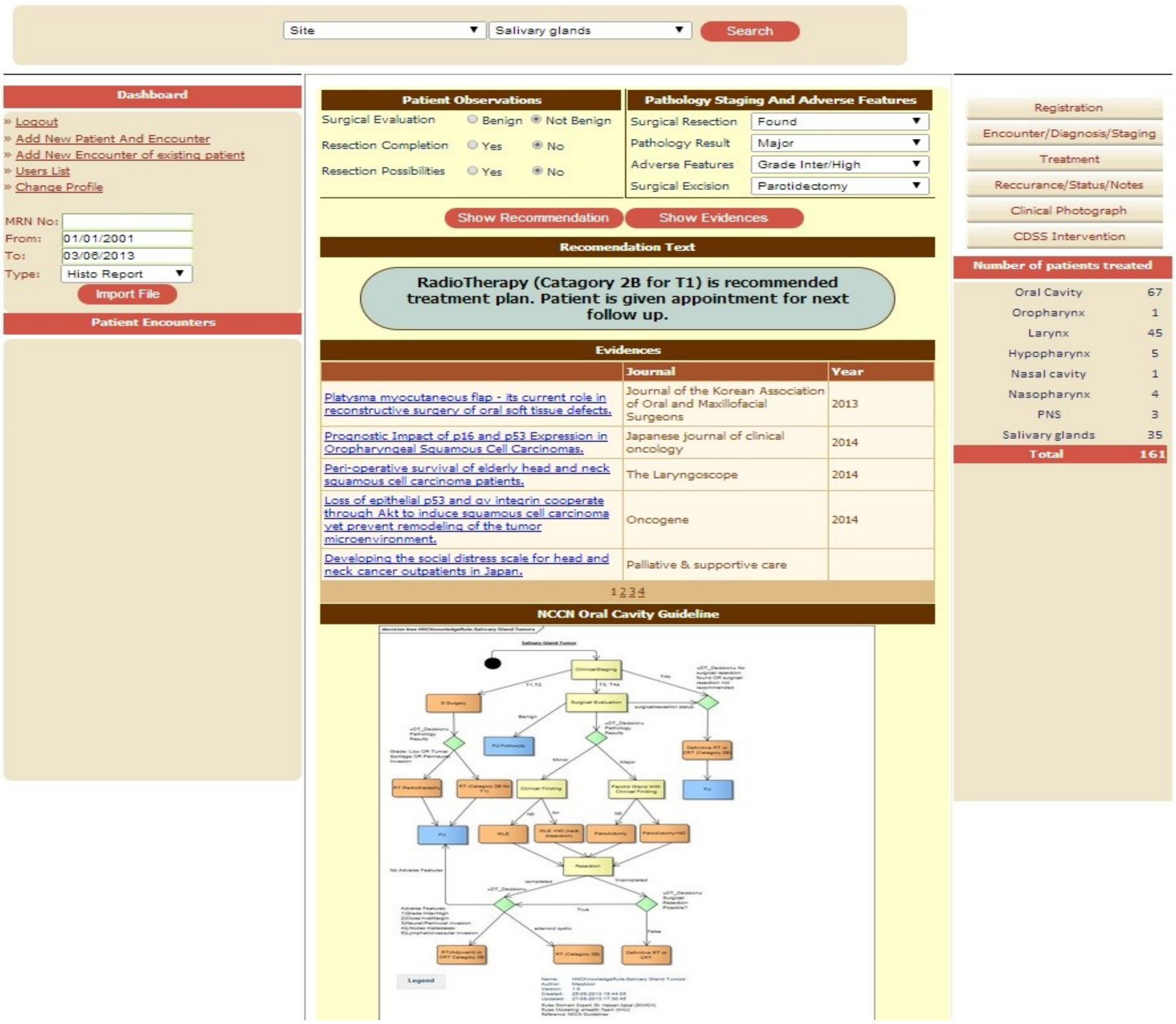


Figure 6.6: Smart CDSS Evidence Support for Salivary Gland recommendations Using KnowledgeButton

CHAPTER 7: CONCLUSION

The proposed decision support system provides clinical recommendation service by constructing intelligent knowledge base of HNC disease. Also, it utilizes the cloud infrastructure to reduce healthcare cost and facilitates interoperability among different heterogeneous healthcare services. Security and privacy is also taken into consideration and is provided by the system ensuring secured sensitive healthcare data exchange. The whole communication is carried out using healthcare standards such as HL7, Arden Syntax, and MLM. The system is extensible, flexible, and scalable and also provides automation in recommendation generation process. Intelligent Authoring tool is provided as an associated component of Smart CDSS to provide easy to use graphical user interface for knowledge rule creation. New evidence from online resources is also part of the system which makes Smart CDSS as an evidence adaptive CDSS system.

REFERENCES

[Blumenthal2010] Blumenthal D, Tavenner M. The "meaningful use" regulation for electronic health records. N Engl J Med. 2010;363(6):501–4.

[Kern2012] Kern LM, Wilcox A, Shapiro J, Dhopeshwarkar RV, Kaushal R. Which components of health information technology will drive financial value? Am J Manage Care. 2012;18(8):438–48.

[Wright2009] Wright A, SittigDF, Ash JS, Sharma S, Pang JE,Middleton B. Clinical decision support capabilities of commercially-available clinical information systems. J Am Med Inform Assoc. 2009;16(5):437–644.

[Wright2008] Wright A, Sittig DF. A four-phase model of the evolution of clinical decision support architectures. Int JMed Inform. 2008;77(10):641–9.

[Sittig2008] Sittig DF, Wright A, Osheroff JA, Middleton B, Teich JM, Ash JS, et al. Grand challenges in clinical decision support. J Biomed Inform. 2008;41(2):387–92.

[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.

[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.

[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.

[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.

[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.

[Soumeya2012] A.Soumeya, D.Michel, R.Claire, B.Philippe and L.Eric. "A UMLS-based knowledge acquisition tool for rule-based clinical decision support system development." Journal of the American Medical Informatics Association , vol 8, no. 4, pp. 351-360, 2012.

[Nathan2005] H. Nathan C, R. Roberto A, F. Guilherme Del, B. RICHARD L, H. Timothy P and R. Lorrie K, " KAT: A Flexible XML-based Knowledge Authoring Environment", Journal of the American Medical Informatics Association, vol 12, no. 4, pp. 418-430, 2005.

[Dustin2008] D. Dustin, D. Jemery, B. Christopher, F. Simon and A. J. Mark. "A Knowledge Authoring Tool for Clinical Decision Support" Journal of Clinical Monitoring and Computing, vol 22, no. 3, pp. 189-198, 2008.

[Robert2002] J. Robert A and D. Balendu, "Challenges in Implementing a Knowledge Editor for the Arden Syntax: Knowledge Base Maintenance and Standardization of Database Linkages", Proceedings of the AMIA Symposium. American Medical Informatics Association, pp. 355–359, 2002.

[Rachel2009] R. Rachel, G. Rupali and R. Roberto A, "A Clinical Rule Editor in an Electronic Medical Record setting: Development, Design, and Implementation", Proceedings of the AMIA Annual Symposium. American Medical Informatics Association, pp. 537–541, 2009.

[Samwald2012] Samwald M, De Fehre K, Bruin J, Adlassnig KP. The Arden
syntax standard for clinical decision support: experiences and directions.
J Biomed Inform. 2012. doi:10.1016/j.jbi.2012.02.001.

[Gietzelt2011] Gietzelt M, Goltz U, Grunwald D, Lochau M, Marschollek M, Song B, et al. Arden2ByteCode: a one-pass Arden syntax compiler for service-oriented decision support systems based on the OSGi platform. Comput Methods Prog Biomed. 2011.

[Karadimas2002] Karadimas HC, Chailloleau C, Hemery F, Simonnet J, Lepage E. Arden/J: an architecture for MLM execution on the Java platform. J Am Med Inform Assoc. 2002;9(4):359–68.

[DBates2003] D. Bates, G. Kuperman, S. Wang, T. Gandhi, A. Kittler, L. Volk, C. Spurr, R. Khorasani, M. Tanasijevic, and B. Middleton. Ten Commandments for effective clinical decision support: making the practice of evidence-based medicine a reality. Journal of the American Medical Informatics Association, 10(6):523{530, 2003.

[GFiol2012] G. Fiol, V. Huser, H. Strasberg, S. Maviglia, C. Curtis, and J. Cimino. Implementations of the hl7 context-aware knowledge retrieval (SinfobuttonT) standard: Challenges, strengths, limitations, and uptake. Journal of Biomedical Informatics, 2012.

[LFrenzel1986] L. Frenzel. Crash course in articial intelligence and expert systems.Sams, 1986.

[MGuilherme1997] M. Guilherme Del Fiol, K. Kawamoto, J. Cimino, N. Maviglia, P. Barr, T. Reuters, S. Bolte, G. Healthcare, and D. Ballot. Context-aware knowledge retrieval (infobutton) decision support service implementation guide.

[AHolbrook2003] A. Holbrook, S. Xu, and J. Banting. What factors determine the success of clinical decision support systems? In AMIA Annual Symposium Proceedings, volume 2003, page 862. American Medical Informatics Association, 2003.

[NHulse2005] N. Hulse, R. Rocha, G. Del Fiol, R. Bradshaw, T. Hanna, and L. Roemer. Kat: a flexible xml-based knowledge authoring environment. Journal of the American Medical Informatics Association, 12(4):418{430, 2005.