Computer Methods and Programs in Biomedicine 150 (2017) 41–72

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

Multi-model-based interactive authoring environment for creating shareable medical knowledge

Taqdir Ali[a], Maqbool Hussain[a,d], Wajahat Ali Khan[a], Muhammad Afzal[a,d], Jamil Hussain[a], Rahman Ali[a], Waseem Hassan[a], Arif Jamshed[b], Byeong Ho Kang[c], Sungyoung Lee[a,*]

[a] Department of Computer Science and Engineering, Kyung Hee University, Seocheon-dong, Giheung-gu, Yongin-si 446-701, Gyeonggi-do, Republic of Korea
[b] Department of Radiation Oncology, Shaukat Khanum Memorial Cancer Hospital and Research Centre, 7A Block R-3, M.A. Johar Town, Lahore 54782, Pakistan
[c] Computing and Information Systems, University of Tasmania, Hobart 7001, Tasmania, Australia
[d] College of Electronics and Information Engineering, Sejong University, Seoul, South Korea

ARTICLE INFO

Article history:
Received 18 July 2016
Revised 18 July 2017
Accepted 20 July 2017

Keywords:
Arden Syntax
Medical Logic Module (MLM)
HL7 vMR
SNOMED CT
Knowledge authoring tool
Shareable knowledge
Interoperable knowledge

ABSTRACT

Objective: Technologically integrated healthcare environments can be realized if physicians are encouraged to use smart systems for the creation and sharing of knowledge used in clinical decision support systems (CDSS). While CDSSs are heading toward smart environments, they lack support for abstraction of technology-oriented knowledge from physicians. Therefore, abstraction in the form of a user-friendly and flexible authoring environment is required in order for physicians to create shareable and interoperable knowledge for CDSS workflows. Our proposed system provides a user-friendly authoring environment to create Arden Syntax MLM (Medical Logic Module) as shareable knowledge rules for intelligent decision-making by CDSS.
Methods and materials: Existing systems are not physician friendly and lack interoperability and shareability of knowledge. In this paper, we proposed *Intelligent-Knowledge Authoring Tool* (I-KAT), a knowledge authoring environment that overcomes the above mentioned limitations. Shareability is achieved by creating a knowledge base from MLMs using Arden Syntax. Interoperability is enhanced using standard data models and terminologies. However, creation of shareable and interoperable knowledge using Arden Syntax without abstraction increases complexity, which ultimately makes it difficult for physicians to use the authoring environment. Therefore, physician friendliness is provided by abstraction at the application layer to reduce complexity. This abstraction is regulated by mappings created between legacy system concepts, which are modeled as domain clinical model (DCM) and decision support standards such as virtual medical record (vMR) and Systematized Nomenclature of Medicine - Clinical Terms (SNOMED CT). We represent these mappings with a *semantic reconciliation model* (SRM).
Results: The objective of the study is the creation of shareable and interoperable knowledge using a user-friendly and flexible I-KAT. Therefore we evaluated our system using completeness and user satisfaction criteria, which we assessed through the system- and user-centric evaluation processes. For system-centric evaluation, we compared the implementation of clinical information modelling system requirements in our proposed system and in existing systems. The results suggested that 82.05% of the requirements were fully supported, 7.69% were partially supported, and 10.25% were not supported by our system. In the existing systems, 35.89% of requirements were fully supported, 28.20% were partially supported, and 35.89% were not supported. For user-centric evaluation, the assessment criterion was 'ease of use'. Our proposed system showed 15 times better results with respect to MLM creation time than the existing

* Corresponding author.
E-mail addresses: taqdir.ali@oslab.khu.ac.kr (T. Ali), maqbool.hussain@oslab.khu.ac.kr (M. Hussain), wajahat.alikhan@oslab.khu.ac.kr (W. Ali Khan), muhammad.afzal@oslab.khu.ac.kr (M. Afzal), jamil.hussain@oslab.khu.ac.kr (J. Hussain), rahmanali@oslab.khu.ac.kr (R. Ali), waseem.h@khu.ac.kr (W. Hassan), arifj@skm.org.pk (A. Jamshed), byeong.Kang@utas.edu.au (B.H. Kang), sylee@oslab.khu.ac.kr (S. Lee).

http://dx.doi.org/10.1016/j.cmpb.2017.07.010

systems. Moreover, on average, the participants made only one error in MLM creation using our proposed system, but 13 errors per MLM using the existing systems.

Conclusion: We provide a user-friendly authoring environment for creation of shareable and interoperable knowledge for CDSS to overcome knowledge acquisition complexity. The authoring environment uses state-of-the-art decision support-related clinical standards with increased ease of use.

1. Introduction

1.1. Background

A clinical decision support system (CDSS) provides an effective and proficient service model in the healthcare domain to enhance service quality and cost effectiveness of patient care [1]. Physician adaptation to CDSS depends on the ease of using the system. CDSS can support and assist physicians in making decisions during patient care [1,2]. CDSS creates a coordination path between patients and physicians by providing effective recommendations, alerts, and reminders at the proper time [3]. A number of methods for automatic diagnosis, treatment, and medication administration are proposed in order to support the process of clinical decision-making [4–6]. For effective recommendations and alerts, CDSS requires a knowledge base to regulate the system's information flow. The key factors for success of a CDSS are knowledge quality and continuous improvement of that knowledge [7]. Each CDSS is comprised of three main components: knowledge base, inference engine, and communication mechanism [8]. The most important and frequently reported issues with CDSSs are integration into clinical workflows and dissemination of successful interventions from one system to another [9]. However, the fundamental barriers to successful utilization of CDSSs are creating, enhancing, and managing the knowledge base [10] and disseminating the created knowledge [11]. One solution for removing these barriers is to create a shareable and interoperable knowledge base. While this eliminates barriers on one hand, it incurs knowledge acquisition complexity on the other hand, which again drives physicians away from using CDSSs.

1.2. Motivation

Existing systems have strong dependency on the knowledge engineers to create knowledge with the input from physicians. The dependency on knowledge engineers resulted in lack of complete knowledge transfer, understanding dilemma due to communication gap, and additional cost of hiring knowledge engineers. Existing systems started developing by providing training to the physicians about some specialized languages like Arden Syntax to create knowledge. This resulted in less dependency on knowledge engineers, but difficulty level of these specialized languages prompted physicians to limit relying on these systems. Physicians require a simple, understandable, and manageable knowledge acquisition methodology in the form of knowledge authoring tools. However, if the complexity of the authoring tool increases, its adoption rate decreases drastically. Therefore, an easy-to-use interface, with maximum abstraction of technical aspects and closeness of contents to localized terms, is the most suitable product for physicians. Additionally, the system should have aspects of knowledge shareability and interoperability, which would attract physicians to adopt the CDSS and to utilize the knowledge acquisition process for shareable and interoperable knowledge creation and maintenance.

1.3. Barriers to CDSS adoption

1.3.1. Shareable knowledge base

Knowledge shareability can be realized through standardized representation of the knowledge. The Health Level-7 (HL7) community has designed Arden Syntax-based Medical Logic Module (MLM) as a standard unit of medical knowledge for healthcare [12]. HL7 Arden Syntax is an ANSI standard and provides a comprehensive structure for representing clinical knowledge [13,14]. The main intention of Arden Syntax is to enable physicians to easily transform their clinical experiences and practices into sharable knowledge [14].

Challenges: Physicians experience problems transforming their clinical knowledge into MLMs [15]. The existing systems [16] have complicated interfaces for creating MLMs in addition to complex structure and syntax, which create a significant barrier for knowledge acquisition.

1.3.2. Interoperable knowledge base

To determine the effectiveness of knowledge shareability, we need standard data models for knowledge representation. Knowledge acquisition tools have not been successfully adopted due to the minimal support for standard clinical data models [7]. Consequently, the existing knowledge acquisition tools fail to resolve the heterogeneity problem of clinical information models [17]. Another intrinsic barrier for creating MLMs is the curly brace problem of querying the required input data from the medical systems. The CDSS community has recommended a standard information model, virtual medical record (vMR), to resolve the issue of heterogeneous data models for CDSS systems [18] as well as the curly brace problem [19]. The standard data model vMR meets the scalability and interoperability objectives for a knowledge acquisition tool [18,20]. Knowledge that is created via standard CDSS input and output specified in the vMR data model is easily integrated among different CDSS systems. However, the use of the standard data model vMR needs to link with a standard terminology to maximize system interoperability. For example, Systematized Nomenclature of Medicine - Clinical Terms (SNOMED CT) supports comprehensive terminology in the clinical domain used by physicians worldwide [21,22].

Challenges: Incorporating the standard data model vMR into Arden Syntax MLM increases the complexity of clinical rules for domain experts who will need to learn the technical specifications of the standard. Additionally, physicians are comfortable with local terminologies and prefer creating rules using localized concepts instead of standard terminologies due to maximum recall of concepts.

1.4. Proposed solution: I-KAT overview

We propose a knowledge authoring environment called *Intelligent-Knowledge Authoring Tool* (I-KAT), which overcomes the limitations of the state-of-the-art systems. We introduce a *semantic reconciliation model (SRM)* to normalize the knowledge acquisition complexity that adopts HL7 MLM in combination with vMR and SNOMED CT. We incorporate SNOMED CT codes into the MLMs to avoid the intrinsic vocabulary binding issue of Arden Syntax [23,24]. Using SNOMED CT and vMR in MLM creation enhances shareability and interoperability on one hand, but increase system complexity with respect to knowledge creation. Based on our experience with physicians at the collaborative hospital, Shaukat Khanum Memorial Cancer Hospital and Research Centre (SKMCH),[1] we incorporated hospital management and information systems (HMIS) concepts in the form of the *domain clinical model (DCM)*. Physicians use the DCM concepts instead of directly using the vMR schema (described in our previous work [10,25]). Physicians feel comfortable creating rules using the familiar HMIS concepts compared to the technical aspects of HL7 vMR and other Arden Syntax artifacts [26]. As a result, the proposed system the complexity of knowledge creation is not presented to physicians. It is also equipped with IntelliSense to make rule creation easier. This functionality facilitates experts to select the desired value of a key from all possible values during rule creation. The proposed system achieves this functionality using DCM-vMR and vMR-SNOMED CT mappings established in the SRM.

We evaluated the results of our system through system-centric and user-centric aspects. We compared the performance of our system with the ArdenSuite tool of Medexter [27,28]. In the system-centric evaluation, we used state-of-the-art recommendations [16] and observed that our system exhibited higher implementation support for all three requirement categories (i.e., *Essential, Recommended*, and *Optional*) [16]. In the user-centric evaluation, we focused on the ease of use and reduction of MLM syntactic and logical errors. In the evaluation process, physicians and knowledge engineers were asked to create MLMs for treatment plan recommendations for an oral cavity site in the domain of head and neck cancer. On average, our system improved participant performance by a factor of 15 over Arden-Suite. The average error rate for MLM creation using our system decreased from 13 errors to just one error. We also validated our system on 1314 real cases of SKMCH oral cavity patients and verified the execution environment with the created knowledge base.

The contents of this paper are organized as follows. Section 2 describes the related work and Section 3 presents the details of our proposed methodology. Section 4 describes the results, it evaluates and compares the various aspects of the proposed system with existing systems. Section 5 presents the discussion on the system, and Section 6 concludes the paper and discusses future directions.

2. Related work

In the existing Arden Syntax-based CDSS systems, MLMs are generated either manually or semi-automatically. Samwald et al. [27] used MLMs generated by a commercialized tool in an integrated Arden Syntax development and test environment (IDE) in the CDSS system. Their MLM development tool provides a simple interface to create and test Arden Syntax MLMs. They are also working on solve the curly brace problem with the help of GELLO (a loose acronym for Guideline Expression Language Object-Oriented) and vMR standards. However, physicians are responsible for creating the MLM using its structural slots and need to be familiar with the MLM's syntax and structure, which is a tedious task. Therefore, domain experts greatly depend on medical informatics experts to transform knowledge into knowledge base. Similarly, there is no automation in the utilization of standard terminologies during MLM creation; physicians use standard concepts based on their own knowledge about standard terminologies.

A number of commercial vendors such as Agfa, AllScript, McKesson, Medexter, and Siemens incorporated Arden Syntax in CDSS integrated health information system softwares [29]. The initial versions of Arden Syntax provided easy to use environment to experts, but its evolution made the syntax difficult to ready, understand, and compose. This resulted in organizations need of hiring knowledge engineers to translate expert knowledge into Arden Syntax. Arden2ByteCode [30] is one such open source system that uses Eclipse framework for authoring environment to create and edit Arden Syntax MLM. Experts faced difficulty in understanding the complex environment of the system, therefore, they depended heavily on knowledge engineers to create knowledge.

Child Health Improvement through Computer Automated system (CHICA) [31] uses Arden Syntax MLM to generate encounter documents for patient visits. The system transformed the guidelines into algorithm to author MLM and required intense involvement of the knowledge engineers.

Anand et al. [31] proposed a rule editor for clinicians creating Arden Syntax MLMs, which has complex interfaces with multi-phase selection for the ontology domain; however, it lacks interoperability to integrate with external databases. Jung et al. [3] propose a system that executes MLMs represented in ArdenML. A third-party open source production rule system, Drool, has been used for MLMs execution. Therefore, the system needs two transformation phases: first, expert knowledge transformed into Arden Syntax; second, Arden Syntax transformed into ArdenML. In the second transformation, physicians need additional expertise in Extensible Markup Language (XML) and Extensible Stylesheet Language Transformations (XSLT).

Dunsmuir et al. [32] proposed pattern and outcome-based approach to enable clinicians to create rules without dependency on knowledge engineers. However, this system was designed for the anesthesia domain only and is not scalable to other domains. Physicians also require training in XML language to create patterns and outcomes to utilize in rule creation. Seitinger et al. [33] implemented an Arden Syntax-based CDSS to generate guidelines for Lyme disease. This system uses fuzzy Arden Syntax and manually transforms text-based medical guidelines into Arden Syntax MLMs. Generating MLMs manually is very tedious and error-prone. In addition, this system does not take into account any standard clinical data models or terminologies in the Arden Syntax MLM creation.

In light of the aforementioned literature, CDSSs without knowledge acquisition tools are not adaptive in the real environment. In general, the existing systems evolve the knowledge bases by creating MLMs manually or in a semi-automatic manner. However, those systems that automatically generate MLMs lack standard terminologies and standard data model practices, which hinders knowledge shareability. The use of standard terminologies and data models may make a system difficult to use; therefore, the existing systems lack user-friendly interfaces for acquiring knowledge. The syntax and other artifacts of Arden Syntax are tedious and difficult for clinical experts to memorize, but the existing systems do not provide any facility to hide these complexities from the clinical experts. In addition, the existing knowledge acquisition tools focus on a specific medical domain to create knowledge instead of a scalable system design that can be easily extended to other domains. As a result, our proposed system provides a user-friendly authoring environment to create Arden Syntax MLM as shareable knowledge rules for intelligent decision-making by CDSS.

3. Methods and materials

In the proposed system, we targeted head and neck cancer and focused on domain concepts for knowledge creation. Based on physicians' feedback regarding our previous work [10,25], we encapsulated and provided abstraction of the vMR data model and

[1] SKMCH: https://www.shaukatkhanum.org.pk/.

the complex structure and syntax of the MLM from the user interface and provided the most usable and ordinary HMIS concepts for rule creation. This abstraction is achieved by the *domain clinical model (DCM)* and the *semantics reconciliation model (SRM)* to handle the dependencies of HMIS, vMR, and SNOMED CT concepts. Therefore, we provided understandable DCM and SNOMED CT concepts on the user interface for rule writing, while shareable and interoperable MLMs were generated with standard vMR classes, attributes, and codes of SNOMED CT concepts in a back-end process.

3.1. Business process model and architecture

The system's business process model for rule creation and mappings of DCM, vMR, and SNOMED CT terminologies is shown in Fig. 1. Workflow is represented in standard Business Process Model and Notation (BPMN) format [34]. It resembles in domain analysis, formal notation selection, conceptual modeling, and logical modeling with the existing workflow process model [35]. The set of activities, processes, gateways, and messages is represented in pools with standard notations using Enterprise Architect [36]. High level system requirements in the form of the business process model are implemented and shown as system's architecture in Fig. 2. The system's workflow comprises two pools: *Physician Activity* and *Multi-Model Mapping*.

3.1.1. Physician activity pool

The physician is the main actor in the system creating knowledge rules. In rule creation, the physician can use the desired concept values either from DCM or SNOMED CT terminologies. This enhances concept recall and reduces the chance of errors while enabling IntelliSense functionality. These concept models are selected by *Choose concepts model* from inputs of data model objects *Domain Clinical Model* and *Domain SNOMED CT*, respectively. The business process *Create rule* includes two parallel activities of concept selection (i.e., *Show IntelliSense* and *Select concepts from DCM*) to provide input for creating rule activity, *Create final rule*. The proposed system implements *Show IntelliSense* and emphSelect concepts from DCM on the *User Interface* module in architecture. In *Select concepts from DCM*, the physician can easily select the desired concepts for use in the rule facts. In architecture, these concepts are fetched to the *User Interface* using *DCM Concepts Controller* and *DCM Query Manager*, which creates and runs the appropriate query on DCM Ontology.

In *Show IntelliSense*, the physician is presented with a window that shows all possible values of the selected concept and allows selection of the correct desired value. The value list comes from the *Domain Ontology* using DCM-vMR and vMR-SNOMED mappings. The vMR schema classes and attributes bridge the selected DCM concept and the values list of SNOMED CT concepts. The *IntelliSense Controller* is the component responsible for performing *Show IntelliSense* activity using *DCM-vMR Mapper* and *vMR-SNOMED Mapper*. Both mappings are queried by three corresponding query managers (i.e., *DCM Query Manager, vMR Query Manager*, and *SNOMED Query Manager*). The final rule creation activity, *Create final rule*, is invoked using *Show IntelliSense* and *Select concepts from DCM* as parallel activities.

After successful *Create final rule*, the proposed system transforms the rule into Arden Syntax MLM by *Transform to MLM*. In summary, the rule is created by the physician using understandable DCM and SNOMED CT concepts, which the system then transforms into Arden Syntax MLM with amalgamation of vMR classes and attributes along with SNOMED CT codes. The *MLM Creator*, in system's architecture Fig. 2, is responsible for performing *Transform to MLM*. The *Transform to MLM* involves three types of mappings generated by the *Multi-Model Mapping* pool using corresponding controllers and query managers. MLM has its own standard artifacts and syntax based on HL7 standard Arden Syntax specification. All artifacts are fetched by *Arden Artifacts Controller* using *Arden Query Manager*. The created MLM is stored in the knowledge base.

3.1.2. Multi-model mapping pool

In the business process model, the *Multi-model mapping* pool mainly focuses on generation of multi-model mappings among DCM, vMR schema, and SNOMED CT concepts. This mapping activity is performed once as a prerequisite for rule creation. Three parallel activities of *Analyze vMR Specifications, Analyze SNOMED CT Specifications*, and *Analyze Domain Clinical Model* are performed to analyze vMR, SNOMED CT, and DCM, respectively. The outcome of vMR and SNOMED CT analysis is utilized by *Generate vMR-SNOMED mappings* to generate *vMR-SNOMED mappings*. Similarly, the outcome of vMR and domain clinical model analysis is utilized by *Generate DCM vMR mappings* to generate *DCM-vMR mappings*, while the output of SNOMED CT and DCM analysis is used by *Generate DCM SNOMED mappings* to generate *DCM-SNOMED mappings*. These mappings belong to our proposed *semantic reconciliation model* (SRM). The subsequent sections explain the *DCM* and *SRM* in detail.

3.2. Domain clinical model (DCM)

The HMIS and electronic medical record (EMR) systems play a critical role in healthcare [37] to solve logistical organizational problems to improve experts' clinical decisions and reduce the cost of managing clinical information [38]. The HMIS and EMR systems are mostly used to maintain patients' active and inactive problems, allergy information, surgical, family and social histories, current medications, nicotine and alcohol use, symptoms, vital signs, and laboratory and radiology reports [39]. In general, physicians are familiar with these and other related clinical concepts. Therefore, the system facilitates creation of knowledge rules using understandable clinical concepts. The DCM provides a model to manage and organize the HMIS clinical concepts. We used a proper Clinical Information Modelling Process (CIMP) [16], based on investigating concept semantics. According to the standard requirements and recommended methods of the CIMP process, we collected clinical concepts from the HMIS system, analyzed and specified the clinical context among contents, and structured the DCM.

We structured the DCM concepts using a well-known and popular clinical notes protocol, SOAP (subjective, objective, assessment, plan). SOAP notes were developed by Weed [40] to provide a logical and reproducible framework for generating medical records [41]. A SOAP-based model improves the quality of client services by easy communication among healthcare professionals and by enabling physicians to identify, prioritize, and track patients' problems in a timely and systematic manner [42]. Therefore, we designed a SOAP-based structure for DCM, which allows the clinical concepts to model in a scalable and manageable manner. We derived and aligned DCM with the HL7 standard data model vMR to maintain semantics among different concepts, as partially shown in Fig. 3, as unified modeling language (UML) class diagram [43]. We transformed the DCM model into an ontology format, which was semantically verified by SKMCH physicians.

The information related to symptoms, past medical history, family history, social history, and current medication that exist in legacy systems are modeled under the *Subjective* category of the SOAP model. The *Objective* category includes vital sign and observable symptoms that can be easily measured through different physical tests, laboratory tests, and imaging tests. In the *Assessment* category, we organized all information about diagnoses, health status, and lifestyle changes of the patients. In the *Plan* category, we

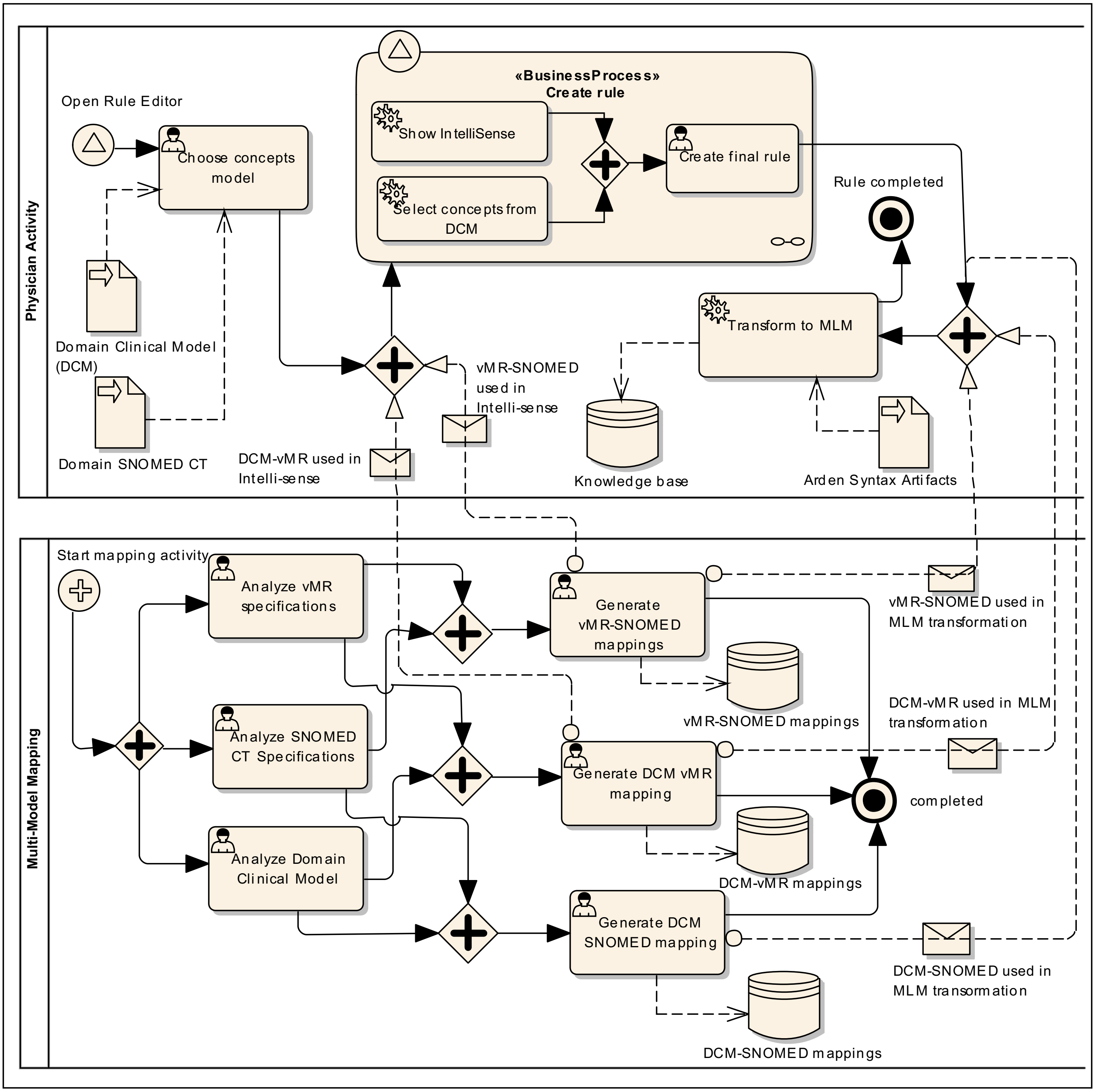

Fig. 1. Rule creation and mapping generation and use in rule creation workflow.

modeled all recommended treatment plans such as proposed medications, chemotherapy, radiotherapy, and surgeries. The individual DCM models are shown in Appendix A: Figs. A.12–A.15 for the *Subjective, Objective, Assessment*, and *Plan*, respectively, with attributes of the vMR data model.

3.3. Semantics reconciliation model (SRM)

SRM is a reconciliation model that unifies concepts from three different models (i.e., SNOMED CT, vMR, and DCM) and reconciles it with high level abstraction. SRM achieves the objectives of interoperability, shareability, and user friendliness. While the system creates rules in MLMs using the standard vMR data model and SNOMED CT codes to achieve shareability and interoperability, this increases the complexity of MLM creation for physicians. Therefore, the system hides this complexity and achieves user friendliness by providing a selectable tree of DCM concepts. It also provides SNOMED CT and DCM concepts in an IntelliSense window that allows physicians to select the desired concept. Consequently, the MLMs complex structure and syntax with consolidation of vMR and SNOMED CT are hidden from physicians. The SRM model, as shown in Fig. 4, provides three types of mappings: DCM concepts to SNOMED CT concepts (DCM-SNOMED), vMR data model to

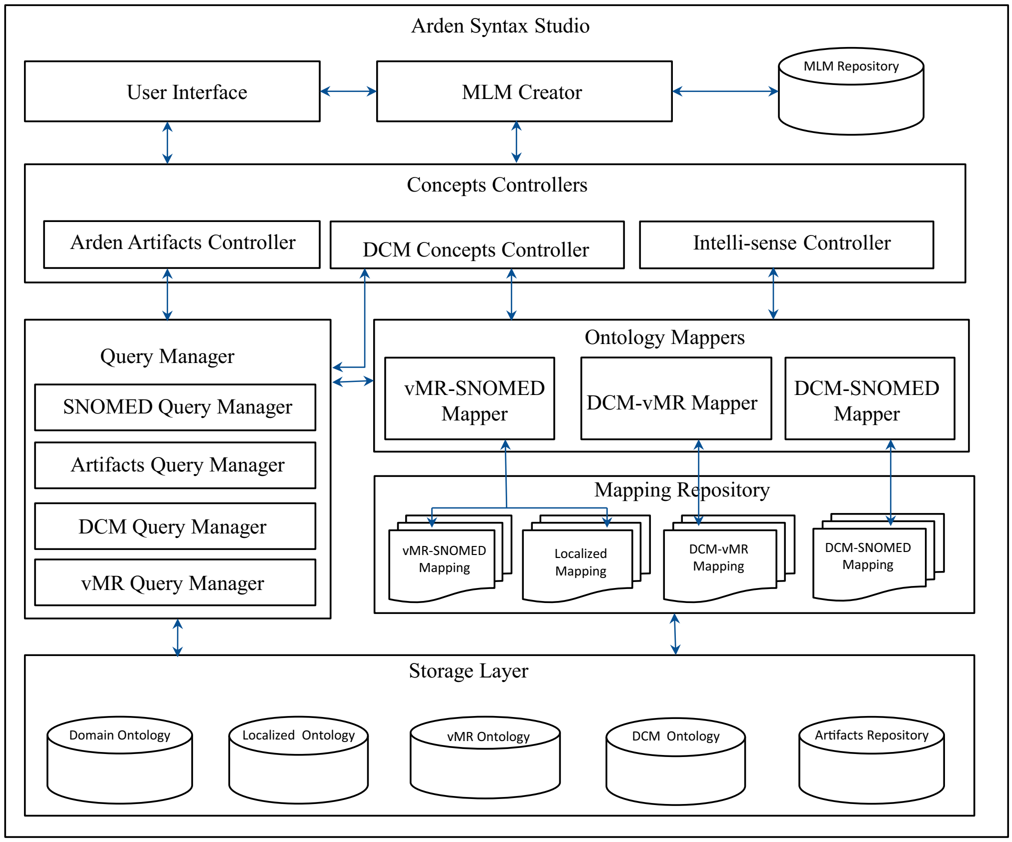

Fig. 2. I-KAT architecture.

SNOMED CT concepts (vMR-SNOMED), and DCM concepts to vMR data model (DCM-vMR). These mappings are described in the subsequent sub-sections.

3.3.1. Types of mappings

DCM-SNOMED mapping. Using standard terminologies enhances the interoperability and shareability of knowledge acquisition tools [20]. In the SRM model, we mapped DCM concepts to SNOMED CT concepts to achieve shareability and user friendliness goals as physicians are more familiar with DCM concepts than vMR and SNOMED CT concepts. In the user interface, the physician selects the desired DCM concept during rule creation, while in the back-end process, the selected DCM concept is represented with vMR and SNOMED CT code in the automatically generated MLM. The system also provides physicians with choice of SNOMED CT concepts for achieving flexibility in the DCM concept selection.

vMR-SNOMED mapping. Standard data models and terminologies help in interoperable and shareable knowledge acquisition [18,20]. These mappings are needed to achieve shareability and interoperability of the knowledge base. The system automatically transforms the created rule into Arden Syntax MLM with consolidation of the vMR and SNOMED CT codes of the correspondingly used DCM concepts in the rule.

Additionally, the vMR-SNOMED mapping helps concept selection in the IntelliSense window during rule creation. DCM concept scope may be compromised in some situations, requiring the physician to select “missing value” from the SNOMED CT. In this case, our system provides IntelliSense functionality from SNOMED CT concepts instead of DCM concepts, which increases flexibility for concept selection. In these mappings, vMR schema classes and their attributes are mapped to the corresponding top hierarchy concepts of SNOMED CT. These mappings are verified by physicians and domain experts in the HL7 community.

DCM-vMR mapping. Understanding and memorizing all vMR schema classes and their attributes is a tedious task for physicians. Therefore, in the user interface, our system provides physicians with DCM concepts instead of vMR schema classes and attributes. These mappings offer user friendliness for knowledge creation and interoperability of the knowledge base. The DCM concepts are mapped to the corresponding vMR classes based on the DCM-SNOMED and vMR-SNOMED mapping output.

3.3.2. SRM mapping methodology: example

In SRM, we focused on three types of mappings, i.e., DCM-SNOMED mappings, vMR-SNOMED mappings, and DCM-vMR mappings. The DCM-SNOMED mappings are generated using our previously developed ontology matching system (SPHeRe) [44]. SPHeRes matching algorithms include string, synonym, label, child, and property matching [17]. These algorithms are suitable for mapping SNOMED CT and DCM due to their ontological nature. We achieved 83.6% accurate mappings using SPHeRe. The remaining ambiguous and un-mapped concepts were mapped with the help of SKMCH physicians using the inspection method [45].

The vMR-SNOMED mappings were generated using the inspection method [45] involving different physicians. The vMR data model contains some specific and limited classes and attributes, which are mostly usable in CDSS systems. We selected the inspection method because vMR class attributes require coded values from particular SNOMED CT top-level hierarchical concepts. In the inspection method, the physicians role is essential because

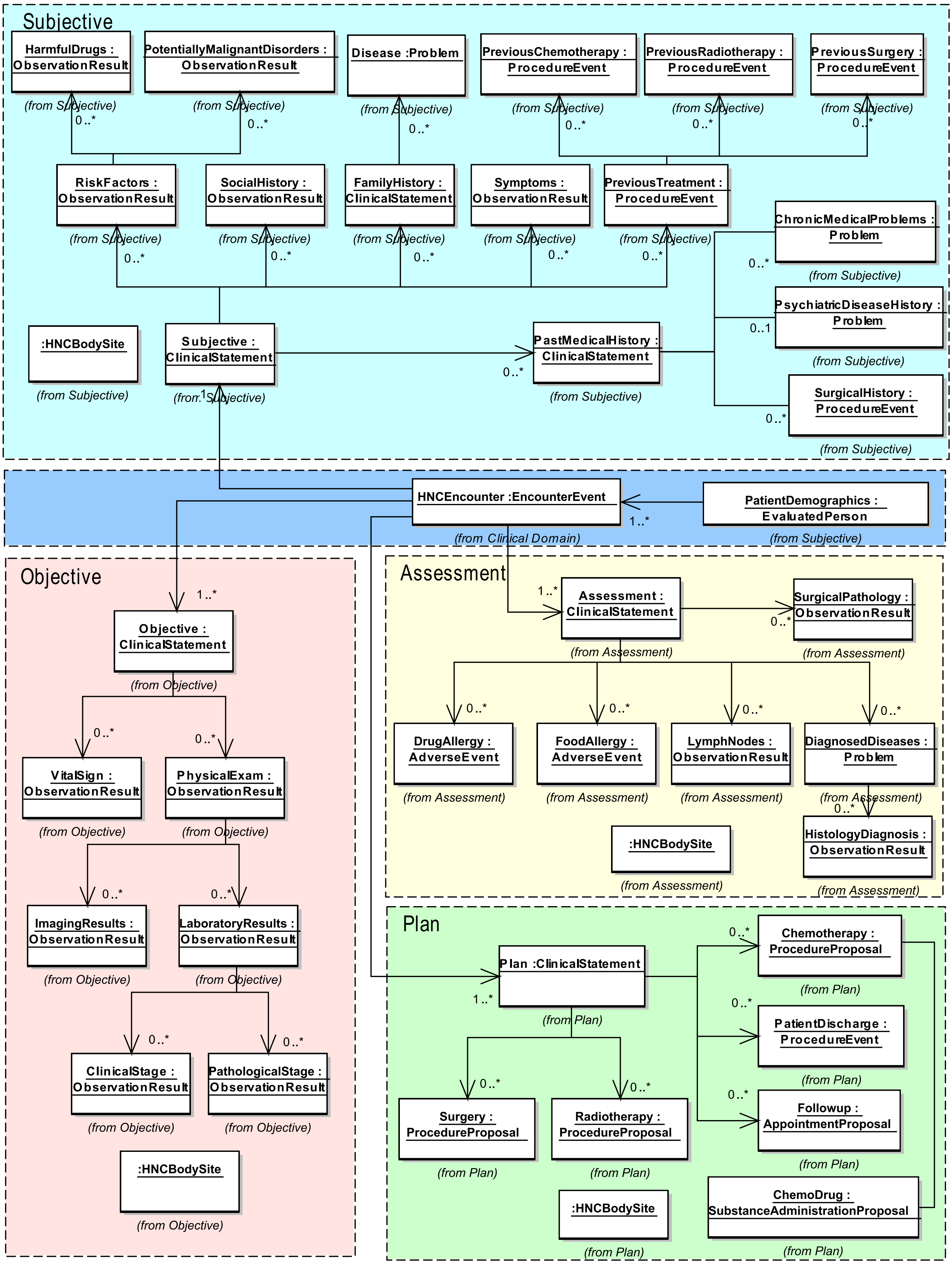

Fig. 3. UML class diagram of domain clinical model (DCM) using a SOAP-based (Subjective, Objective, Assessment, Plan) protocol and vMR data model.

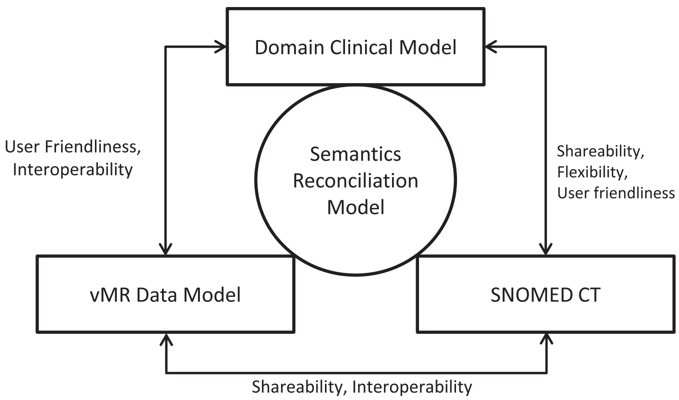

Fig. 4. Semantics reconciliation model (SRM).

the vMR-SNOMED mappings depend on the semantics described in notes, definitions, descriptions, and purposes of each class as well as attributes of vMR and top-level hierarchical concepts of SNOMED CT. The methodology followed by physicians for mapping vMR schema classes to SNOMED CT top-level concepts is illustrated in Fig. 5. The vMR-SNOMED mapping process contains two phases.

Phase 1: Class-level mapping The vMR classes are mapped with corresponding top-level concepts in SNOMED CT. We used the vMR specification document *HL7 Virtual Medical Record for Clinical Decision Support (vMR-CDS) Logical Model, Release 2* [46] and SNOMED CT specification document *SNOMED CT Starter Guide* [47]. The vMR classes and attributes are described in the notes and definitions in its specification document. First, the notes and definitions of each vMR class are found and analyzed to understand the semantics of that particular class. Similarly, all SNOMED CT top-level hierarchical concepts are examined against the selected vMR class. The definition and short description parts of each SNOMED CT top-level concept are iteratively explored and analyzed. Physicians recognize the semantics (i.e., formal, lexical, and conceptual semantics) of the top-level concept. Based on the physicians' analyses of both the vMR class and the SNOMED CT top-level concept semantics, they compared the similarity and retained semantically matched concepts. For example, the vMR class *ProcedureEvent* is mapped to the SNOMED top-level concept *Procedure (71388002)*, as shown in Fig. 6.

Phase 2: Attribute-level mapping We mapped the attributes of vMR class with a SNOMED CT top-level hierarchical concept and its child concepts with specified domain constraints. Primitive attributes proceed with the normal mapping process; however, for associative attributes, the association class is first retrieved, and its attributes are mapped with SNOMED CT concepts using a recursion process. The associative attributes of the selected class are linked with its association class. Furthermore, the phase I steps for finding, analyzing, and identifying semantics are repeated. The additional phase II steps include finding attribute types, processing the association classes, and controlling the domain constraints on child concepts to reduce the size. Finally, vMR class attribute semantics are compared with SNOMED CT concept semantics to assess similarity. Similar semantics are considered as mapped concepts and are stored in the mapping repository.

The third mapping category is DCM-SNOMED mappings, which we achieved using the law of transitive relation, as represented by Eq. (1).

$$\forall C_{DCM}, C_{snomed}, C_{vMR} \in X : (C_{DCM} R C_{snomed} \wedge C_{snomed} R C_{vMR}) \Rightarrow C_{DCM} R C_{vMR} \quad (1)$$

Fig. 6 illustrates an example of DCM-SNOMED, vMR-SNOMED, and DCM-vMR partial mappings for three DCM concepts (i.e., *surgical history, potentially malignant disorders*, and *psychiatric disease history*) that belong to the *Subjective* part of DCM with some partial mappings shown in Table B.6, Appendix B.

4. Results and evaluation

4.1. Case study: treatment plans for oral cavity lesions

We selected the formally extracted refined-clinical knowledge model (RCKM) from our previous work [48]. In this work, we used data-driven knowledge acquisition for real SKMCH patient data to generate a predictive model (PM). The PM was attained using a decision tree algorithm, chi-square automatic interaction detection (CHAID), on the dataset of 1229 patients. Simultaneously, a team of physicians created a clinical knowledge model (CKM) for the oral cavity site of head and neck cancer from a well-known online resource, National Comprehensive Cancer Network (NCCN) guidelines [49]. Finally, the R-CKM was created by a rigorous validation process of conforming the PM as a final model to the CKM. In this study, the created R-CKM specifically focuses on treatment plans for head and neck cancer with emphasis on the oral cavity, as shown in Fig. 7. For a given R-CKM tree, a set of eight rules can be created based on decision nodes for recommended treatment plans, as shown in Table 1. We created a single MLM for each corresponding rule and integrated the compiled version into the HMIS system. In this scenario, we focus on a single MLM for Rule 5, with the following steps performed in creating this rule.

Step 1: We display all required information about the MLM on the *Rule Editor* screen such as *Rule Title, MLM Name, Citation, Purpose*, and *Explanation. Author's name, Institution*, and *Created Date* appear by default from the author's profile information. The detailed implementation of the authoring environment is provided in Appendix C with complete features of the system.

Step 2: In the *Condition* box, we write the condition part of the rule. The *Treatment Intent* concept is reflected in the *Condition* box when we select the *Treatment Intent* from the DCM concepts tree, as shown in the highlighted Area 3 of Fig. C.17. We write the 'equal to'sign (=), and the IntelliSense window appears with the possible values set for *Treatment Intent*. Here we select the value *Radical*. We write all other inputs in the same manner such as *"Treatment Plan Given = Chemo induction"* with the help of the IntelliSense window and DCM Tree. The condition part in the *Condition* box with IntelliSense functionality is shown in Fig. C.18.

Step 3: We follow the procedure in Step 2 for the action part of the rule in the *Action* box of the *Rule Editor*. The *Condition* and *Action* parts after rule completion are shown in Fig. 8.

Step 4: The rule is saved by pressing the *Save Rule* button.

Step 5: The created MLM can be seen on the *Rule Viewer* screen by clicking the *Show Created MLM* button. The application view of the created MLM is shown in Fig. C.19. Moreover, the MLM details can be found in Listing 1 in Appendix D; while the DCM, vMR, and SNOMED CT mappings used to create the MLM are listed in Table B.6, Appendix B.

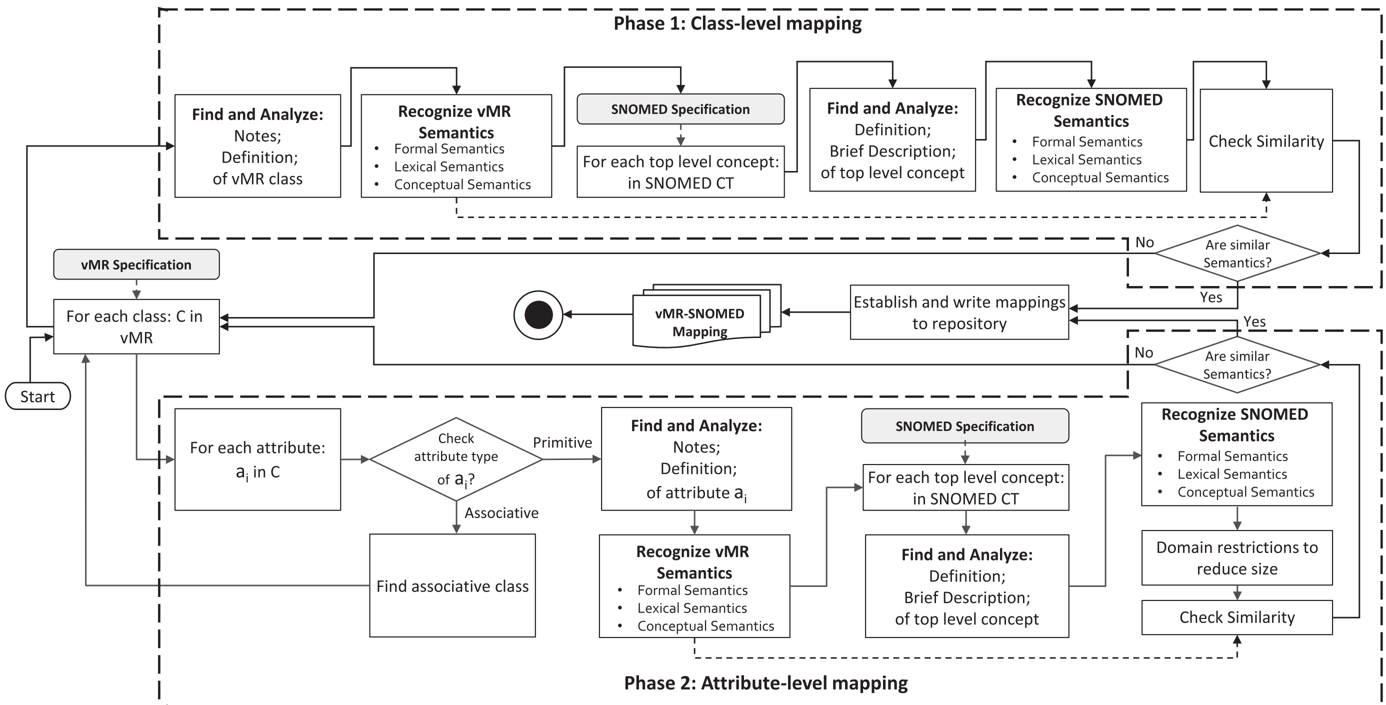

Fig. 5. vMR-SNOMED mapping process.

Table 1
Rules for treatment plan guidelines.

Rule ID	Rule Conditions	Rule Conclusion
Rule 1	Treatment Intent = Palliative	Treatment Plan = Radiotherapy (RT)
Rule 2	Treatment Intent = Radical	Treatment Plan = Chemoinduction
Rule 3	Treatment Intent = Radical and Treatment Plan Given = Chemoinduction and (T = T1 or T = T2) and N = N0	Treatment Plan = Surgery
Rule 4	Treatment Intent = Radical and Treatment Plan Given = Chemoinduction (T = T1 or T = T2) and N = N0 and Treatment Plan Given = Surgery and S = I	Treatment Plan = RT or Next Followup
Rule 5	Treatment Intent = Radical and Treatment Plan Given = Chemoinduction and (T = T1 or T = T2) and N = N0 and Treatment Plan Given = Surgery and S = II	Treatment Plan = RT
Rule 6	Treatment Intent = Radical and Treatment Plan Given = Chemoinduction and(T = T1 or T = T2) and N = N1	Treatment Plan = Surgery Followed by RT or Chemo-radiotherapy (CRT)
Rule 7	Treatment Intent = Radical and Treatment Plan Given = Chemoinduction and ((T = T3 and N = N0) or ((T = T1 or T2) and (N = N2 or N3)) or (T = T3 and (N = N1 or N2 or N3)) or (T = T4 and N = Any N)) and (Histology = 1 or 2 or 3)	Treatment Plan = CRT
Rule 8	Treatment Intent = Radical and Treatment Plan Given = Chemoinduction and ((T = T3 and N = N0) or ((T = T1 or T2) and (N = N2 or N3)) or (T = T3 and (N = N1 or N2 or N3)) or (T = T4 and N = Any N)) and (Histology = 4 or 5 or 6 or 7 or 8 or 9 or 10 or 11 or 12 or 13 or 14 or 15 or 16 or 17 or 18)	Treatment Plan = Surgery Followed by RT

4.2. MLMs validation using real patient cases

We implemented and validated the proposed system using a real practice dataset from SKMCH. The experimental setup and implementation are as follows.

- We created MLMs from eight rules, shown in Table 1, which are modeled from R-CKM as described in Section 4.1. This model was initially validated on a real practice dataset of 739 SKMCH patients with model accuracy of 53% [48]. We re-evaluated the R-CKM on recently generated data from 1783 patients with model accuracy of 73.7%. The R-CKM accuracy ($R-CKM_{acc}$) based on the newly created MLMs is a weighted mean accuracy of disjoint MLMs calculated by Eq. (2).

$$R-CKM_{acc} = \frac{\sum_{i=1}^{n}(pat_{MLM_i} \times A_{MLM_i})}{pat_c} \quad (2)$$

Where pat_{MLM_i} and A_{MLM_i} represent the number of patient cases assigned to MLM_i and its accuracy, respectively. pat_c represents total patient cases assigned to MLMs: MLM_1 to MLM_8.

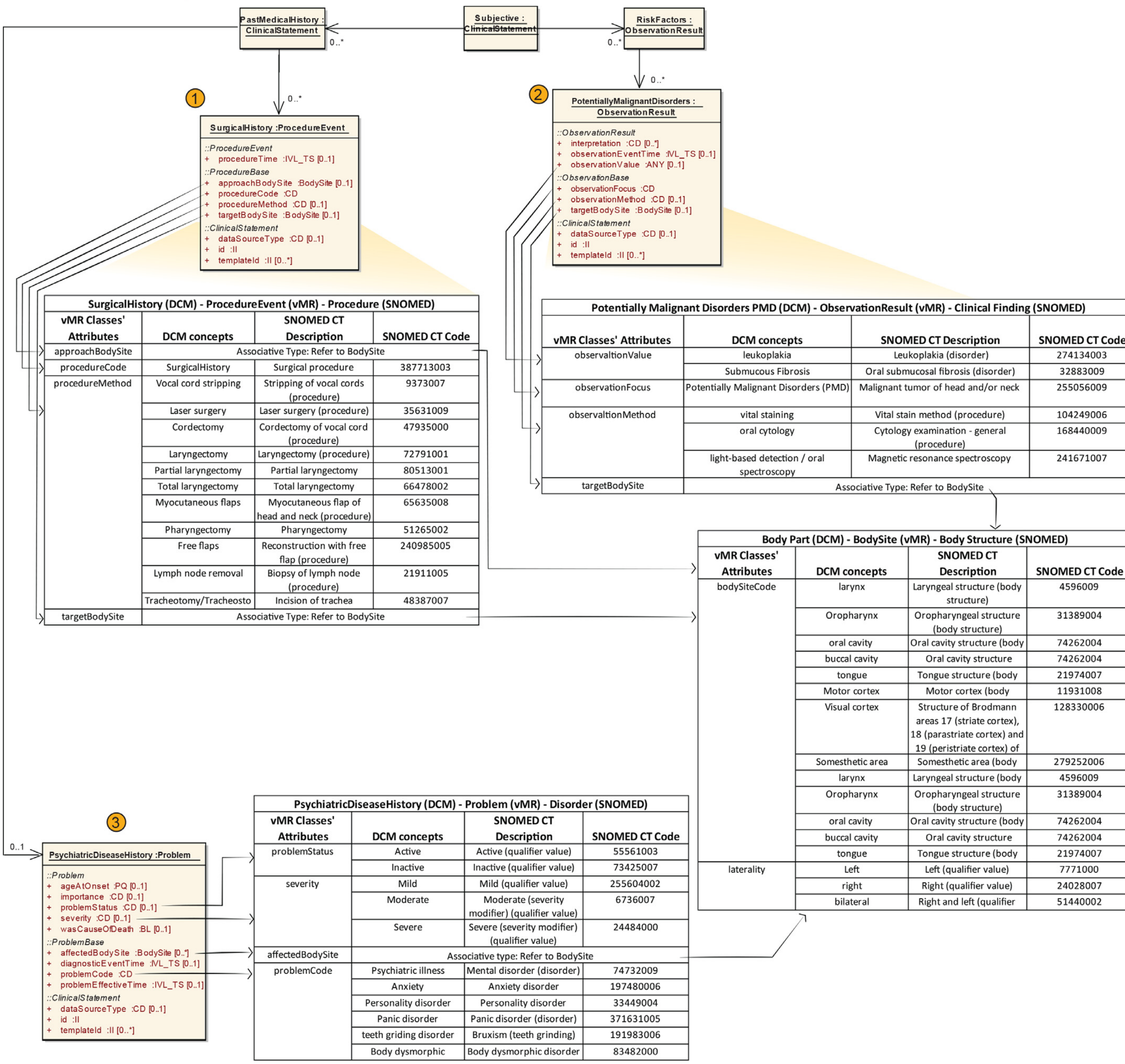

Fig. 6. SRM Example: Attribute-level mappings of vMR, DCM, and SNOMED CT.

Individual MLM accuracy A_{MLM} is calculated as in Eq. (3).

$$A_{MLM} = \frac{pat_{MLM_c} \times 100}{pat_{MLM}} \quad (3)$$

where pat_{MLM_c} and pat_{MLM} represent the number of correctly classified patient cases by MLM and total patient cases assigned to given MLM, respectively.

- The MLMs generated by our proposed system are developed and deployed as Smart CDSS XML-based web service, which is designed according to the framework mentioned in [2].
- We developed a client application in the .NET environment using C# language that extracts oral cavity cancer patient data from the SKMCH database. The client interacts with the Smart CDSS Service and iteratively launches individual patient data for recommendation. Individual patient cases with associated recommendations are saved in a CSV (comma separated values) file for MLM result verification.
- We tested and validated the MLMs on 1314 patient cases with 100% correct recommendations for all patients. This evaluation shows that the MLMs generated by our system are error free and do not affect R-CKM accuracy. Table 2 describes the distribution of patient cases over individual MLMs.

4.3. System comparison and evaluation

We evaluated our system by applying system-centric and user-centric evaluations [50,51]. In the system-centric evaluation, the system was evaluated against a predefined ground truth dataset of opinions. In the user-centric evaluation, the system was evaluated

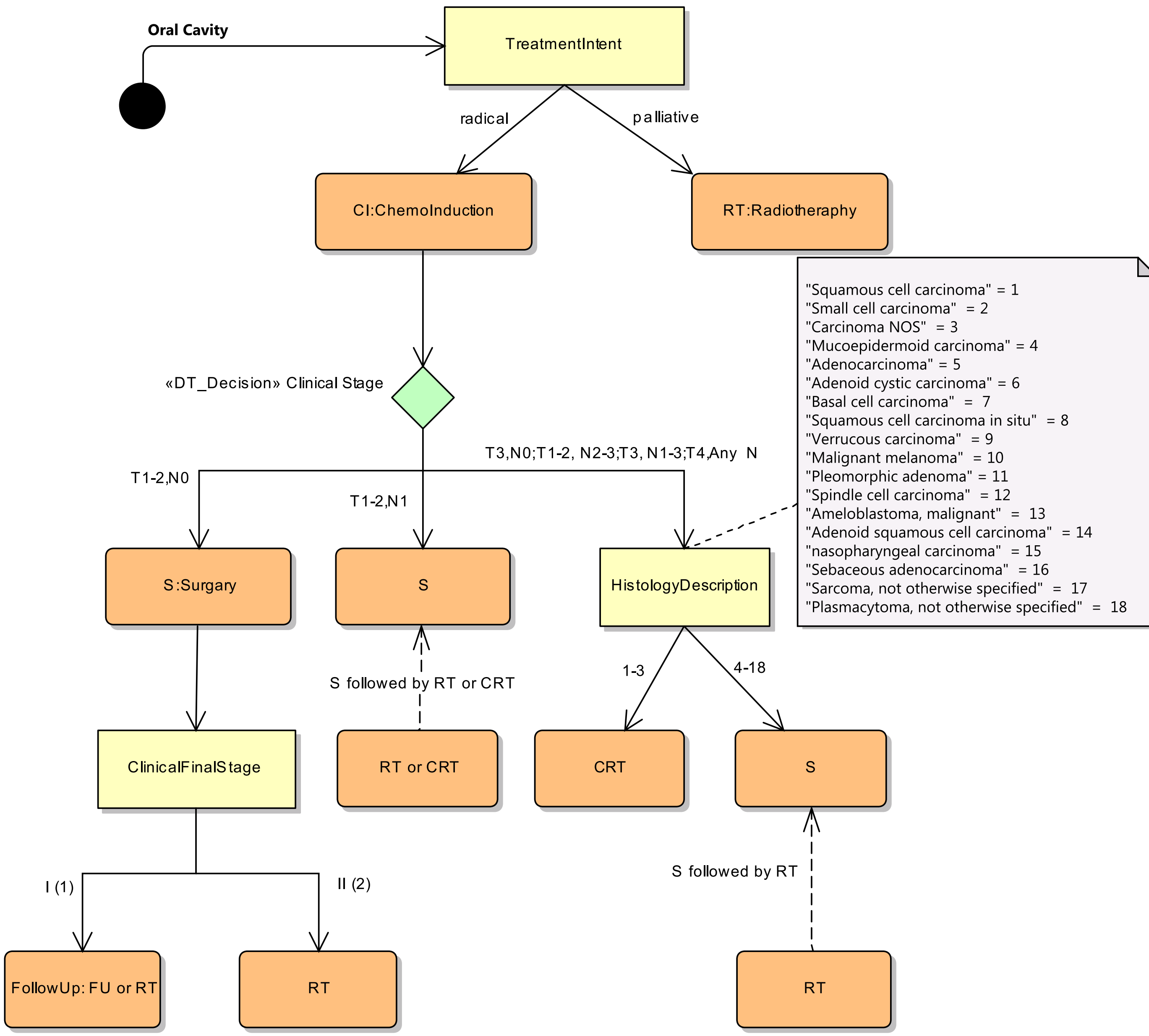

Fig. 7. Refined clinical knowledge model of a treatment plan for an oral cavity lesion [48].

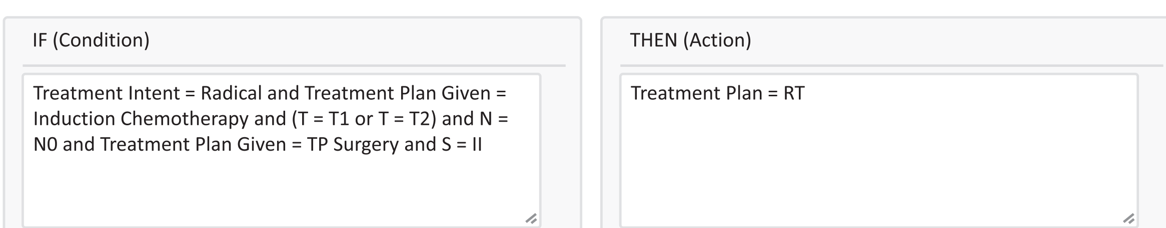

Fig. 8. Logic component of Rule 5.

by user interaction with the system based on performance with respect to MLM creation time of MLM.

4.3.1. System-centric evaluation

In the system-centric evaluation, we formulated the results based on the set of requirements for clinical information modeling tools developed by Moreno-Conde et al. [16]. These requirements were produced after rigorous and intensive surveys and interviews with experts and were categorized into *Essential*, *Recommended*, and *Optional* categories. The total requirements in *Essential*, *Recommended*, and *Optional* categories are 20, 21, and 15, respectively.

Table 2
Distribution of patient cases.

MLM ID	Associated Rule ID	Contents/Logic Complexity [No. of attributes, {No. of logical operators}]	Candidate patient cases
MLM1	Rule 1	[1, {And (0), Or (0)}]	241
MLM2	Rule 2	[1, {And (0), Or (0)}]	39
MLM3	Rule 3	[5, {And (3), Or (1)}]	121
MLM4	Rule 4	[7, {And (5), Or (1) }]	128
MLM5	Rule 5	[7, {And (5), Or (1)}]	158
MLM6	Rule 6	[5, {And (3), Or (1)}]	99
MLM7	Rule 7	[17, {And (7), Or (9)}]	427
MLM8	Rule 8	[29, {And (7), Or (21)}]	31
		Total	**1314**

Table 3
Comparison table of I-KAT and ArdenSuite with respect to implementation category [Essential: E, Recommended: R, Optional: 0].

Priority	Req. Number	I-KAT			ArdenSuite		
		NS	PS	FS	NS	PS	FS
E	IR1			✓		✓	
E	IR2			✓		✓	
E	IR3			✓	✓		
E	IR4			✓			✓
E	IR5			✓			✓
E	IR6			✓			✓
E	IR8			✓			✓
E	IR9			✓		✓	
E	IR10			✓	✓		
E	IR11			✓			✓
E	IR13			✓			✓
E	IR14			✓			✓
E	IR16			✓	✓		
E	IR17	✓			✓		
E	IR18			✓		✓	
E	IR19			✓			✓
R	IR23			✓			✓
R	IR25			✓		✓	
R	IR26			✓	✓		
R	IR27	✓					✓
R	IR29		✓		✓		
R	IR30			✓		✓	
R	IR31		✓				✓
R	IR32			✓	✓		
R	IR33			✓			✓
R	IR34		✓		✓		
R	IR37			✓	✓		
R	IR39			✓		✓	
R	IR40			✓		✓	
R	IR41	✓					✓
R	IR42			✓		✓	
R	IR43			✓	✓		
R	ER57			✓	✓		
R	ER58			✓	✓		
O	IR21			✓		✓	
O	IR45			✓	✓		
O	IR48			✓	✓		
O	IR51			✓			✓
O	IR52	✓				✓	

A team of knowledge engineers and domain experts was created to review and select the candidate requirements of the Clinical Information Modeling Tool (CIMT) [16] for the knowledge acquisition tools. The team formalized a four-phase model process: *reduction phase*, *enhancement phase*, *interpretation phase*, and *evaluation phase*. The objective of this process was to remove all requirements not directly applicable to the knowledge acquisition tools, to incorporate new appropriate requirements, and to interpret the CIMT-based requirements for the knowledge acquisition tools for evaluation. Finally, our proposed system was evaluated with the existing ArdenSuite tool of Medexter [27,28] based on the selected requirements.

Reduction phase: We reduced the total number of requirements by removing the requirements that specifically belonged to CIMT [16] and technology-oriented requirements that were not applicable to knowledge acquisition tools. *Essential* requirements (R) were reduced from 20 to 16 by removing R7, R12, R15, and R20; *Recommended* requirements were reduced from 21 to 16 by removing R24, R28, R35, R36, and R38; and *Optional* requirements were reduced from 15 to 5 by removing R22, R44, R46, R47, R49, R50, and R53-R56 (Fig. 9).

Enhancement phase: We added two new requirements to the *Recommended* category based on our experiences and observations from our previous work [10] with SKMCH physicians. The first new requirement, "*Provide Domain Clinical Model in hierarchical form for easy selection of required concepts during knowledge creation,*" was added as extended requirement ER57. The second requirement, "*Knowledge editor should provide the facility of contextual selection of required value of a concept from the values set using the IntelliSense window,*" was added as extended requirement ER58. Both requirements help experts recall domain concepts during knowledge creation. They also reduce the chance of errors in the knowledge base rules by minimizing the likelihood of wrong concept usage. In total, the *enhancement* phase increased the number of *Recommended* requirements from 16 to 18, as shown in Fig. 9.

Interpretation phase: We interpreted the consensus requirements of CIMT for knowledge authoring tools that are closely related to CIMP [16]. All clinical knowledge management tools and repositories are highly recommended to follow these requirements in the corresponding tools. We interpreted each requirement R in the final requirement set produced in the *enhancement* phase as the corresponding interpreted requirement (IR), as shown in the column for *Interpretation for Knowledge Authoring Tool* in Appendix E. The final requirements list after performing the four-phase process is shown in Appendix E.

Evaluation phase: We compared our system with the commercially available ArdenSuite [27,28] based on the final requirements produced in the *interpretation* phase listed in Appendix E. A detailed comparison based on the final requirements list is shown in Table 3. We classified the implementation of requirements into three categories: *fully supported*(FS), *partially supported* (PS), and *not-supported* (NS). A requirement is *fully supported* when the system has implemented it; if the system has not implemented it, then it is considered as *not-supported*. If some part of the requirement is implemented or has partial functionality, then it is considered as *partially supported*. For instance in IR9, ArdenSuite [27,28] validates the knowledge rule syntax according to the standard MLM syntax, while rule creation semantics depend on expert knowledge. The comparison list in Table 3 shows the priority, requirement number, and implementation status of the requirements with a tick mark (✓) in the corresponding implementation category for each tool.

The comparison performed in the *evaluation* phase is graphically depicted in Fig. 10. The graph shows that I-KAT provides full support to 32 out of 39 (82.05%) requirements, partial support to 3 out of 39 (7.69%) requirements, and no support to the remain-

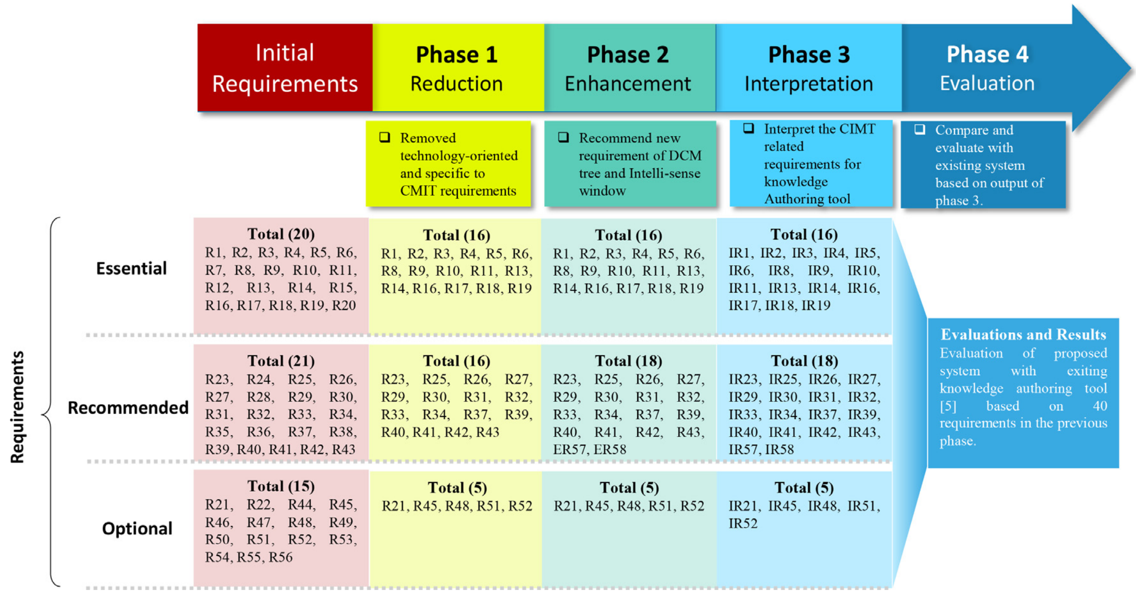

Fig. 9. Phases for evaluation of I-KAT.

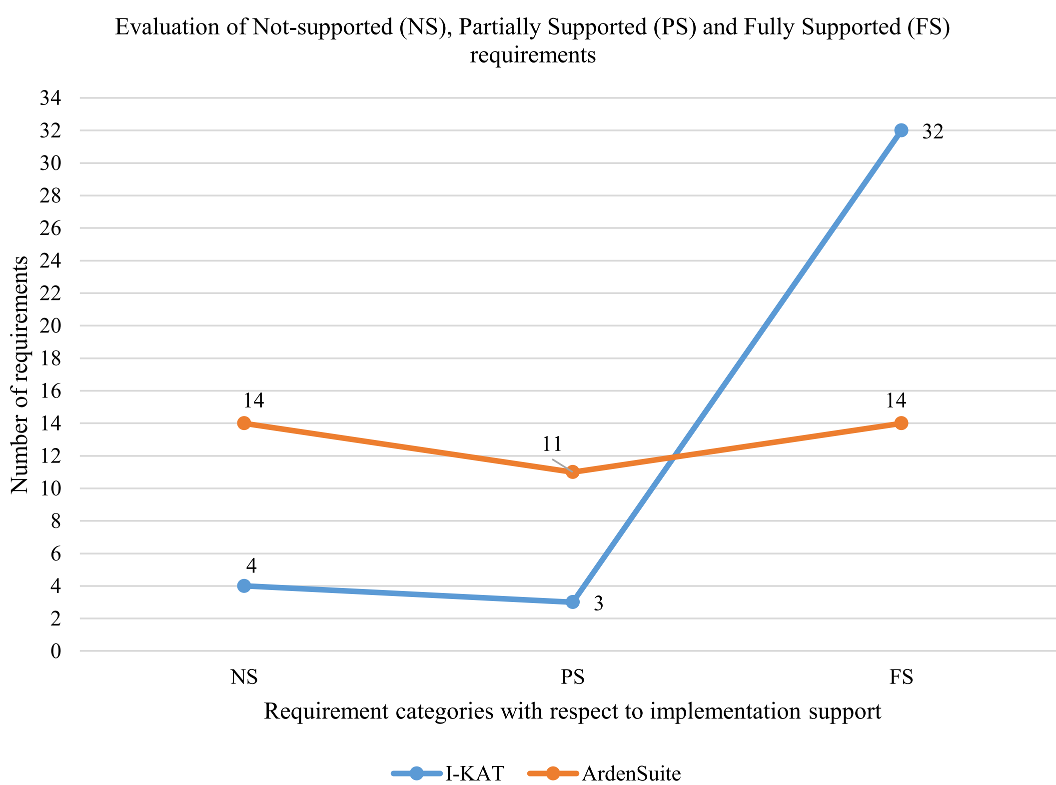

Fig. 10. Accumulative comparison of I-KAT and ArdenSuite with respect to NS, PS, and FS implementation requirements.

ing 4 out of 39 (10.25%) requirements. In contrast, ArdenSuite provides full support to 14 (35.89%) requirements, partial support to 11 (28.20%) requirements, and no support to 14 (35.89%) requirements. This shows that I-KAT offers higher implementation support for the requirements than ArdenSuite.

All three implementation categories (i.e., FS, PS, and NS) are inversely proportional to each other. Therefore, I-KAT has a higher percentage of implemented requirements in FS and a relatively low percentage in PS and NS compared to ArdenSuite. Fig. 11 shows the detailed individual graphs of the comparison between I-KAT and ArdenSuite with respect to implementation categories for all requirement categories. Fig. 11(a) depicts that I-KAT provides no support for one *Essential*, two *Recommended*, and one *Optional* requirements, while ArdenSuite provides no support for four, eight, and two requirements, respectively. I-KAT provides *partial* support for three *Recommended* requirements only, while ArdenSuite has par-

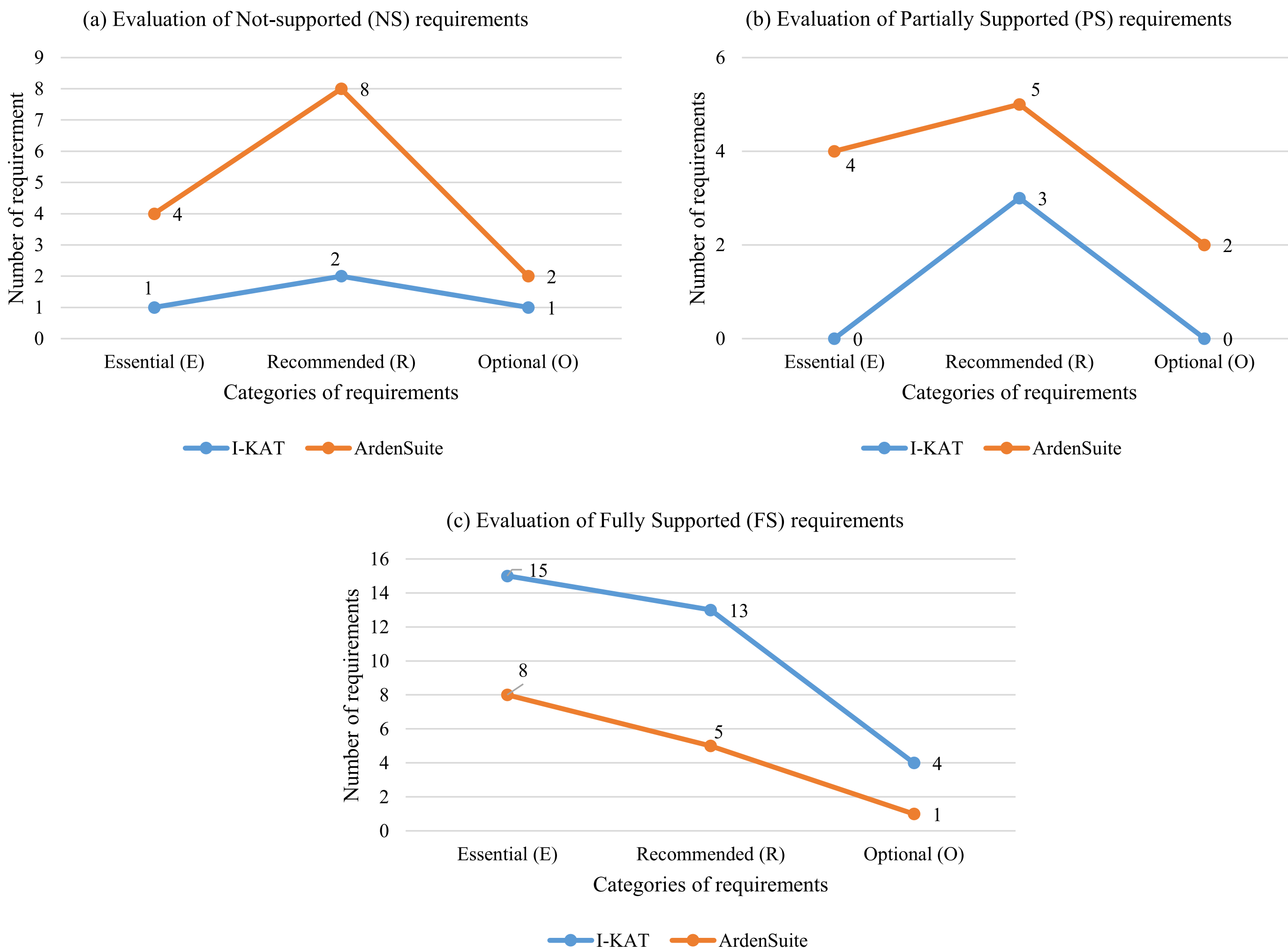

Fig. 11. Individual comparison of I-KAT and ArdenSuite with respect to requirement categories: Essential (E), Recommended (R), and Optional (O).

tial support for four *Essential*, five *Recommended*, and two *Optional* requirements, as shown in Fig. 11(b). I-KAT provides full support for 15 out of 16 Essential requirements, 13 out of 18 Recommended, and 4 out of 5 Optional requirements, as shown in Fig. 11(c). On the other hand, ArdenSuite supports 8 out of 16 *Essential*, 5 out of 18 *Recommended*, and 1 out of 5 *Optional* requirements. This overall evaluation shows that I-KAT exhibits higher implementation support for the requirements in all three categories.

4.3.2. User-centric evaluation

The main focus of our proposed system was to create an easy-to-use interface for creating shareable MLMs. A system with an easy-to-use interface is more time efficient than complex systems that require a great deal of time to produce the required results. Therefore, we considered time when evaluating the user friendliness of our system. Our second objective was to generate sharable knowledge with minimal complexity for physicians; therefore, we selected MLM validation as the second criterion for evaluation. In MLM validation with respect to errors, we focused on syntax and structure complexity as well as the accuracy of vMR classes, attributes, and SNOMED CT codes incorporated in the created MLM.

We evaluated our system with a knowledge engineer, physicians with Arden Syntax experience, and a physician with no such experience. In our experiment, three physicians and one knowledge engineer (total = 4) participated with the following expertise levels in Arden Syntax.

- Physician 1: Experienced
- Physician 2: Intermediate
- Physician 3: Novice
- Knowledge Engineer: Experienced

As a prerequisite, we provided the complete mappings discussed in the *Semantic Reconciliation Model* (SRM) section and trained the participants using basic artifacts of HL7 Arden Syntax for MLM creation in ArdenSuite [27,28] using these mappings. We also provided the Arden Syntax specification to participants. In the experiment, participants created MLMs for each rule described in Table 1 built from the guidelines of the *Treatment Plans* for *Oral Cavity*, as discussed in the previous scenario. In the first session, each participant created MLMs for Rule 1, Rule 3, Rule 5, and Rule 7 based on contents and logic complexity, as shown in Table 2. Each participant used ArdenSuite per our instructions to create an MLM and then created the same MLM using our proposed system. In the second session, we switched the sequence of rule editors and rules based on content and logic complexity, as shown in Table 2. Participants created MLMs for Rule 2, Rule 4, Rule 6, and Rule 8 to avoid bias when using our proposed system initially and then creating the same MLM using ArdenSuite. The experiments showed the following results.

Ease-of-use evaluation. Our proposed system enhanced participants average performance by a factor of 34 for the simplest MLM for Rule 1 creation and by 5 for the complex MLM for Rule 8 creation. The overall average performance showed a 15-fold improve-

Table 4
Ease-of-use evaluation with respect to time.

MLM No	MLM Creation Time		User Involved
	Using ArdenSuite	Using I-KAT	
MLM1	18 min 20 s	22 s	Physician 1
	21 min 15 s	46 s	Physician 2
	Not Applicable	66 s	Physician 3
	8 min 30 s	20 s	Knowledge Engineer
MLM2	18 min 22 s	23 s	Physician 1
	21 min 10 s	40 s	Physician 2
	Not Applicable	69 s	Physician 3
	8 min 34 s	23 s	Knowledge Engineer
MLM3	32 min 20 s	2 min and 47 s	Physician 1
	34 min 30 s	3 min and 5 s	Physician 2
	Not Applicable	2 min and 40 s	Physician 3
	19 min 15 s	2 min and 18 s	Knowledge Engineer
MLM4	33 min 25 s	3 min and 49 s	Physician 1
	35 min 39 s	3 min and 7 s	Physician 2
	Not Applicable	3 min and 45 s	Physician 3
	18 min 21 s	2 min and 19 s	Knowledge Engineer
MLM5	33 min 25 s	3 min and 49 s	Physician 1
	35 min 39 s	3 min and 7 s	Physician 2
	Not Applicable	4 min and 47 s	Physician 3
	21 min 21 s	3 min and 19 s	Knowledge Engineer
MLM6	32 min 20 s	2 min and 47 s	Physician 1
	34 min 30 s	3 min and 5 s	Physician 2
	Not Applicable	2 min and 40 s	Physician 3
	19 min 15 s	2 min and 18 s	Knowledge Engineer
MLM7	34 min 45 s	4 min and 53 s	Physician 1
	36 min 51 s	5 min and 51 s	Physician 2
	Not Applicable	8 min and 27 s	Physician 3
	21 min 34 s	4 min and 10 s	Knowledge Engineer
MLM8	35 min 58 s	5 min and 23 s	Physician 1
	37 min 51 s	6 min and 19 s	Physician 2
	Not Applicable	9 min and 47 s	Physician 3
	22 min 46 s	5 min and 13 s	Knowledge Engineer

ment. Table 4 lists the time in which participants performed the tasks.

MLM validation comparison with respect to errors. In the MLM validation, we recorded the logical and syntactic errors that occurred during MLM creation. For syntactic errors, we considered errors like missing semicolons, missing variable declaration, and missing colon and equal signs in the assignment operator. For logical errors, we considered incorrect vMR concepts, logical IF constructions, and incorrect use of logical operators. Using ArdenSuite, the participants made on average of 4, 3, 15, 16, 15, 14, 17, and 17 errors (syntactic or/and logical) for MLM1 to MLM8, respectively. The average number of errors made during MLM creation using ArdenSuite was 13. Using our proposed system, the average number of errors made during MLM creation was 1. There were no syntax errors in MLMs created by our system because the syntax complexity is hidden from the physicians. The logical errors made by participants when using our system occurred due to incorrect selection from the DCM concepts tree or IntelliSense window. The syntactic and logical errors made during the experiment are shown in Table 5.

5. Discussion

Arden Syntax is close to natural language, making it easier for physicians to understand and utilize it for knowledge rule creation. However, a number of complex artifacts in the Arden Syntax specification increase its complexity. Therefore, our proposed system provides simplified interfaces to hide the Arden Syntax complexity to some extent. Moreover, our system automatically generates MLMs using the maximum number of Arden Syntax artifacts. These artifacts include ":= object," ":= read," "EXTRACT ATTRIBUTE NAME," "IF THEN," and others, as shown in MLM Listing 1. However, some artifacts are not supported by our system, such as loops and some aggregate functions.

The existing legacy HMIS has a diverse format of schemas to represent the system's internal data models. This diversity reduces data interoperability and increases the complexity for integrating CDSS with legacy HMIS systems. The HL7 community recommended the vMR standard data model as an appropriate solution. Existing systems define the input parameters of an MLM using curly braces to represent a query from an external system database, but the designed data models in databases are different. Therefore, the use of a standard data model, HL7 vMR, helps to remove the curly brace problem during integration of CDSS with legacy HMIS. The proposed system provides direction towards the objective of automatic compilation of Arden Syntax to executable format. Arden2ByteCode [30] and ArdenSuite [27] systems incorporate automatic compilation of Arden Syntax to executable format. Arden2ByteCode require physician's expertise in Eclipse framework, while ArdenSuite is a commercial product. Physician feels burden in understanding Eclipse environment, therefore we intend as our future work, development of automatic compilation of Arden Syntax to executable format with fully integrated Arden Syntax MLM creation and testing environment. In the comparison evaluation, we evaluated our system with ArdenSuite, which is a commercially available system with mature compilation functionality. However, in the comparison, we only focused on the creation of shareable and interoperable knowledge in the form of MLMs.

SRM provides a flexible concept modelling environment to accommodate new concepts that can easily evolve using SOAP representation of DCM and data model vMR. We designed the DCM based on the well-known SOAP protocol, which provides a structured system for a comprehensive analysis of problems, diagnosis, treatment plans, demographics, and patient history [41]. Therefore, the DCM can easily adjust new concepts under one of its categories. Similarly, the data model vMR is envisioned to model CDSS-related clinical concepts and attributes with high scalability [46]. The data model vMR is designed and developed as a comprehensive and scalable representative set of data elements after a rigorous multi-national and multi-institutional analysis of CDSS systems [18].

We created *Domain Ontology* to provide a related value set in an IntelliSense window for user selection of the desired concept. Searching the related value set in the entire SNOMED CT versus in the *Domain Ontology* represents a tradeoff between performance efficiency and concept coverage. Searching the entire SNOMED CT for a concept improves coverage but slows performance at the interface level. Likewise, searching only the *Domain Ontology* decreases the concept coverage, but increases efficiency.

The current developed system validates new MLMs by comparing title, name, and purpose with previously created MLMs to find duplicates. We are conducting ongoing research for the maintenance and validation of MLMs; in the future, the system will examine the logic of new rules to determine whether they already exist in the MLM repository. Our experiments show that even novice users were able to create MLMs using our system compared to ArdenSuite. This demonstrates that our system provides a very user-friendly environment that enables physicians with minimal Arden Syntax experience to share their knowledge.

6. Conclusions and future work

The proposed system provides a user-friendly interface to create a shareable and interoperable knowledge base for CDSS. Using these interfaces, physicians can share their practices and experiences in the form of HL7 standard Arden Syntax without needing extensive knowledge of the syntax. Arden Syntax MLM is a standard representation of clinical knowledge that helps achieve

Table 5
MLM syntactic and semantic evaluation.

MLM No	MLM Errors Recorded(L: Logical errors, S: Syntax error		User Involved
	Using ArdenSuite	Using I-KAT	
MLM1	S:2, L:2	S:0, L:0	Physician 1
	S:3, L:5	S:0, L:0	Physician 2
	Not Applicable	S:0, L:0	Physician 3
	S:0, L:0	S:0, L:0	Knowledge Engineer
MLM2	S:2, L:1	S:0, L:0	Physician 1
	S:3, L:4	S:0, L:0	Physician 2
	Not Applicable	S:0, L:1	Physician 3
	S:0, L:0	S:0, L:0	Knowledge Engineer
MLM3	S:10, L:11	S:0, L:1	Physician 1
	S:5, L:18	S:0, L:1	Physician 2
	Not Applicable	S:0, L:2	Physician 3
	S:2, L:0	S:0, L:0	Knowledge Engineer
MLM4	S:9, L:13	S:0, L:1	Physician 1
	S:6, L:17	S:0, L:2	Physician 2
	Not Applicable	S:0, L:2	Physician 3
	S:3, L:0	S:0, L:1	Knowledge Engineer
MLM5	S:7, L:12	S:0, L:1	Physician 1
	S:6, L:16	S:0, L:1	Physician 2
	Not Applicable	S:0, L:3	Physician 3
	S:3, L:1	S:0, L:0	Knowledge Engineer
MLM6	S:8, L:9	S:0, L:1	Physician 1
	S:6, L:17	S:0, L:2	Physician 2
	Not Applicable	S:0, L:1	Physician 3
	S:1, L:0	S:0, L:0	Knowledge Engineer
MLM7	S:8, L:13	S:0, L:1	Physician 1
	S:8, L:17	S:0, L:2	Physician 2
	Not Applicable	S:0, L:4	Physician 3
	S:3, L:2	S:0, L:0	Knowledge Engineer
MLM8	S:9, L:15	S:0, L:2	Physician 1
	S:6, L:16	S:0, L:1	Physician 2
	Not Applicable	S:0, L:3	Physician 3
	S:2, L:3	S:0, L:0	Knowledge Engineer

shareability of the knowledge base. The proposed system uses vMR schema classes as a standard data model with standard SNOMED CT terminologies to enhance shareability. To increase user friendliness, the system provides a high-level abstraction of vMR schema classes in the form of a DCM. The use of different models and terminologies to represent clinical knowledge requires mapping among them. Therefore, the proposed system provides an SRM to map among these models. The system creates the Arden Syntax MLM at the back-end when physicians enter rules via our user-friendly interface.

In the future, we plan to extend the system to support complex Arden Syntax artifacts such as loops and aggregate functions. We also aim to extend the system with further mappings between vMR and DCM concepts to support a higher number of concepts. Additionally, we endeavor to integrate our ongoing research on maintenance and validation of MLMs into the current system.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT and Future Planning (2011-0030079). This research was also supported by Korea Research Fellowship program funded by the Ministry of Science, ICT and Future Planning through the National Research Foundation of Korea (NRF-2016H1D3A1938039).

Appendix A. Individual models of DCM for Subjective, Objective, Assessment, and Plan

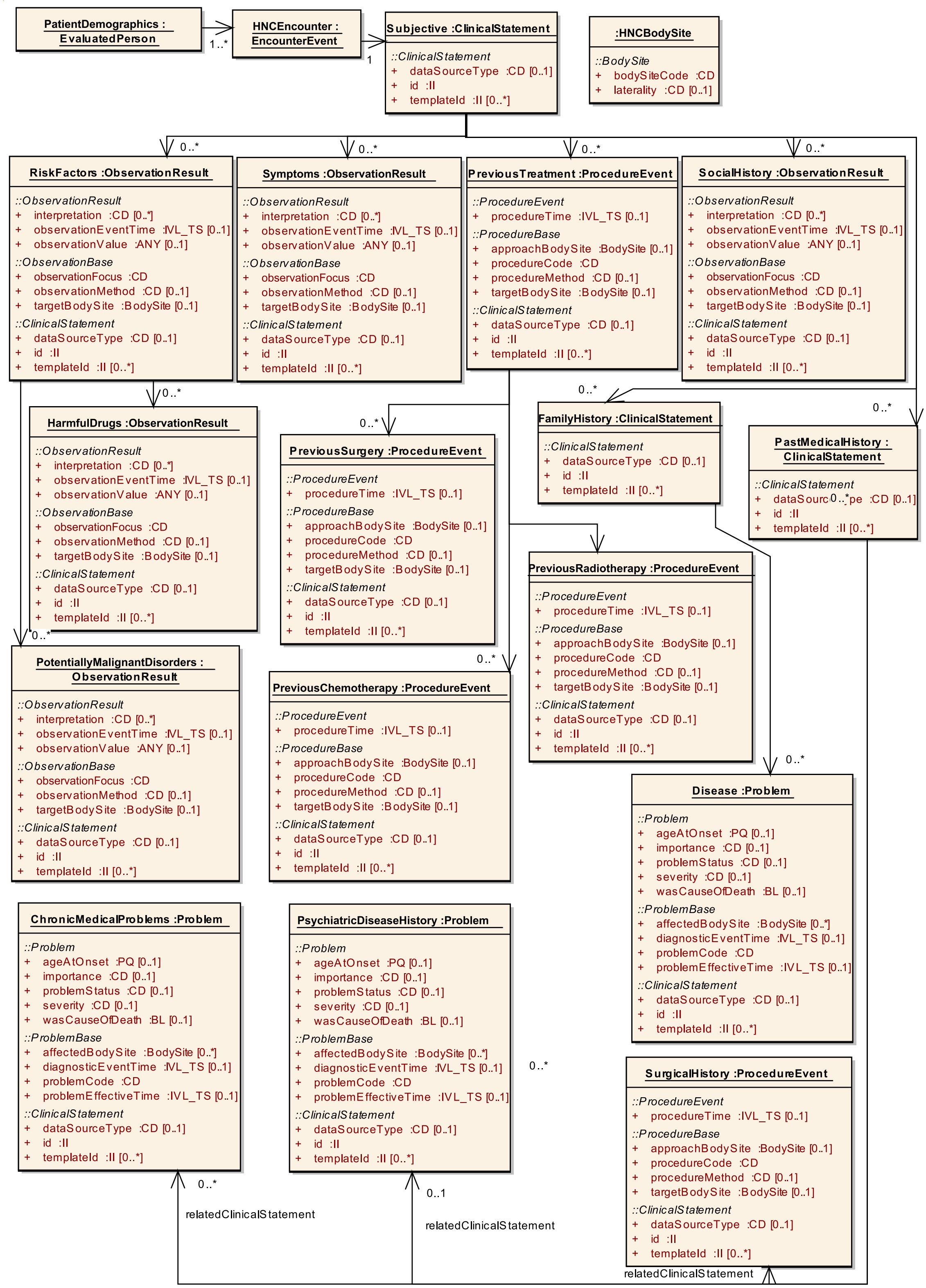

Fig. A.12. Subjective model of DCM.

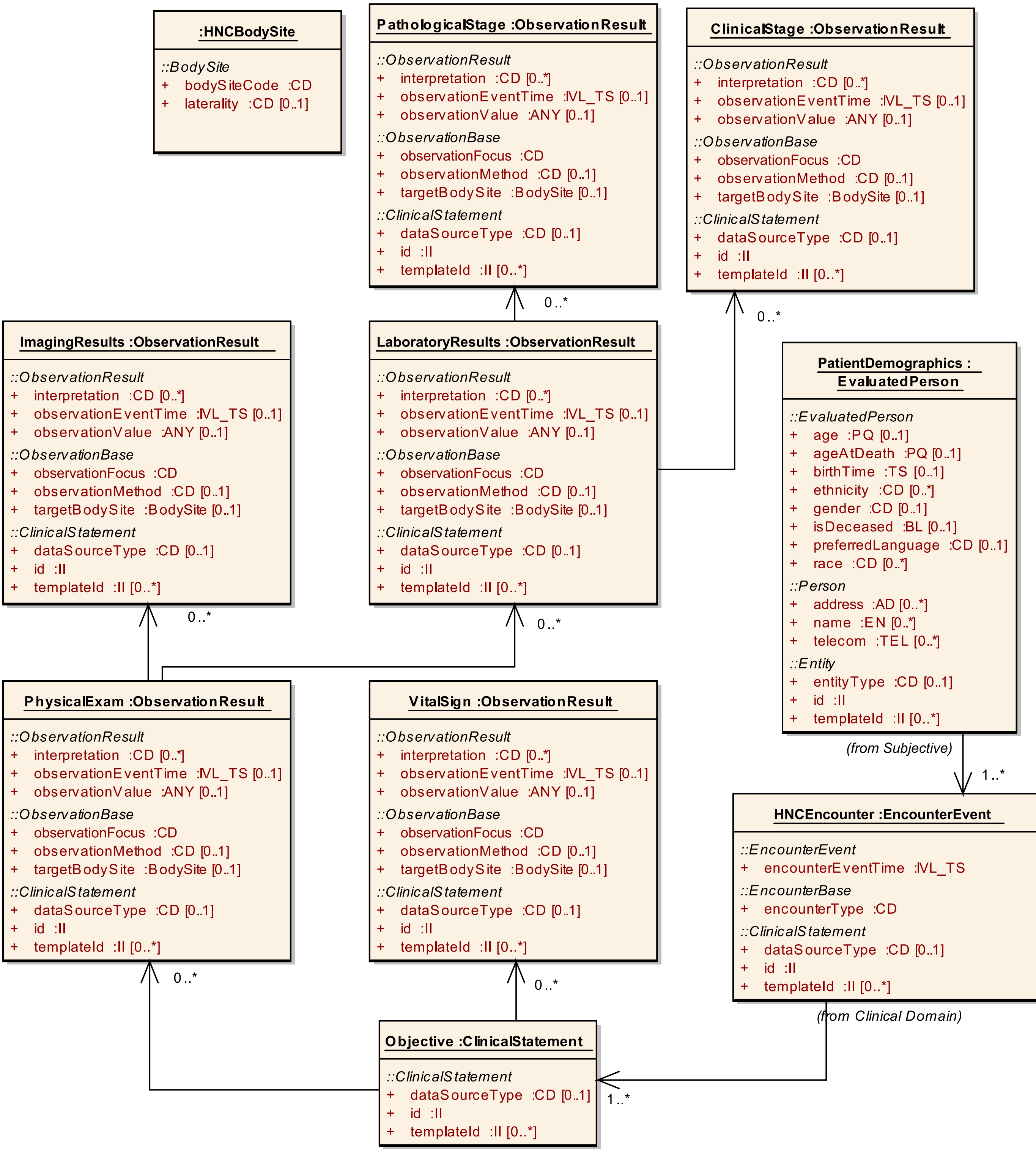

Fig. A.13. Objective model of DCM.

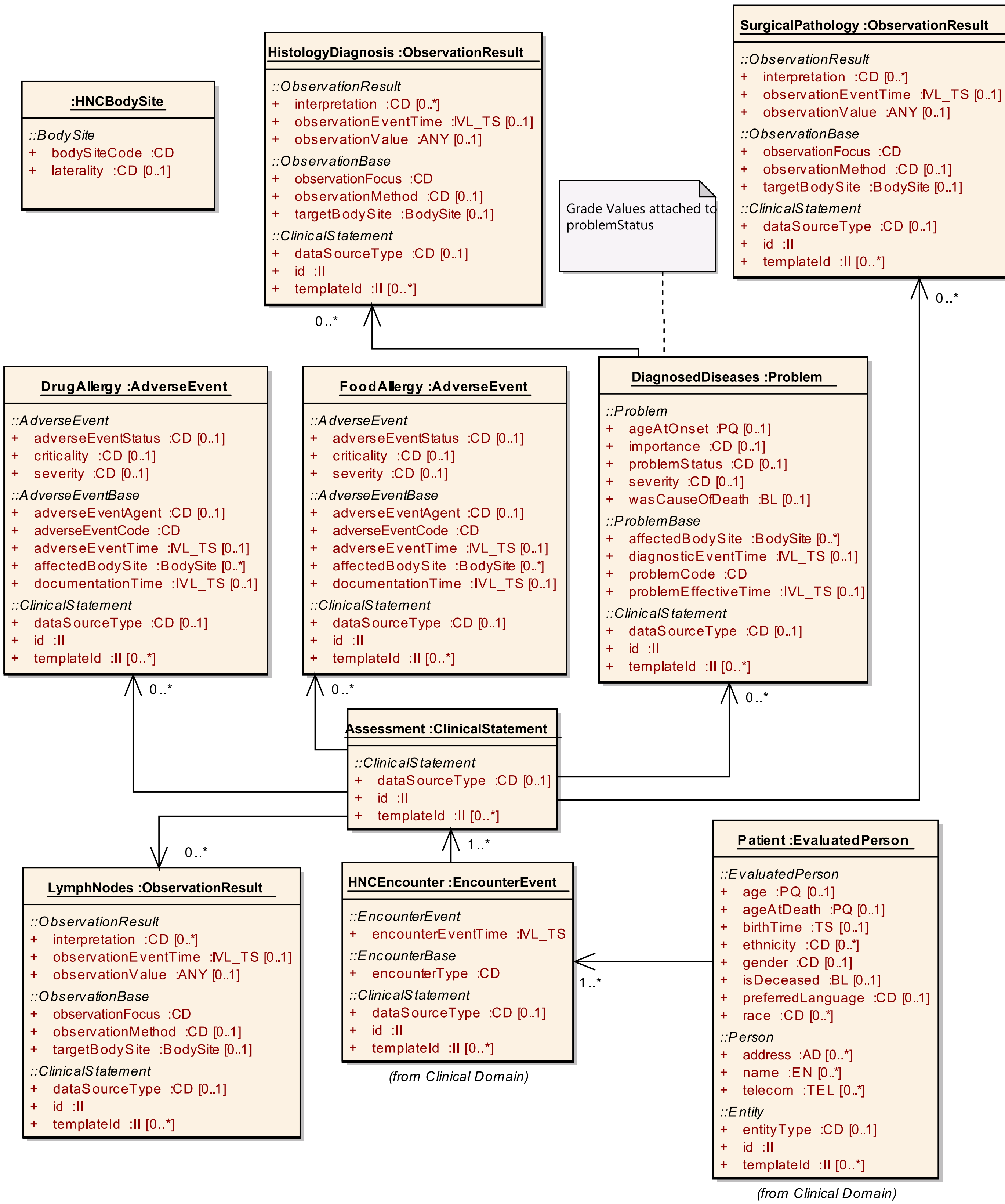

Fig. A.14. Assessment model of DCM.

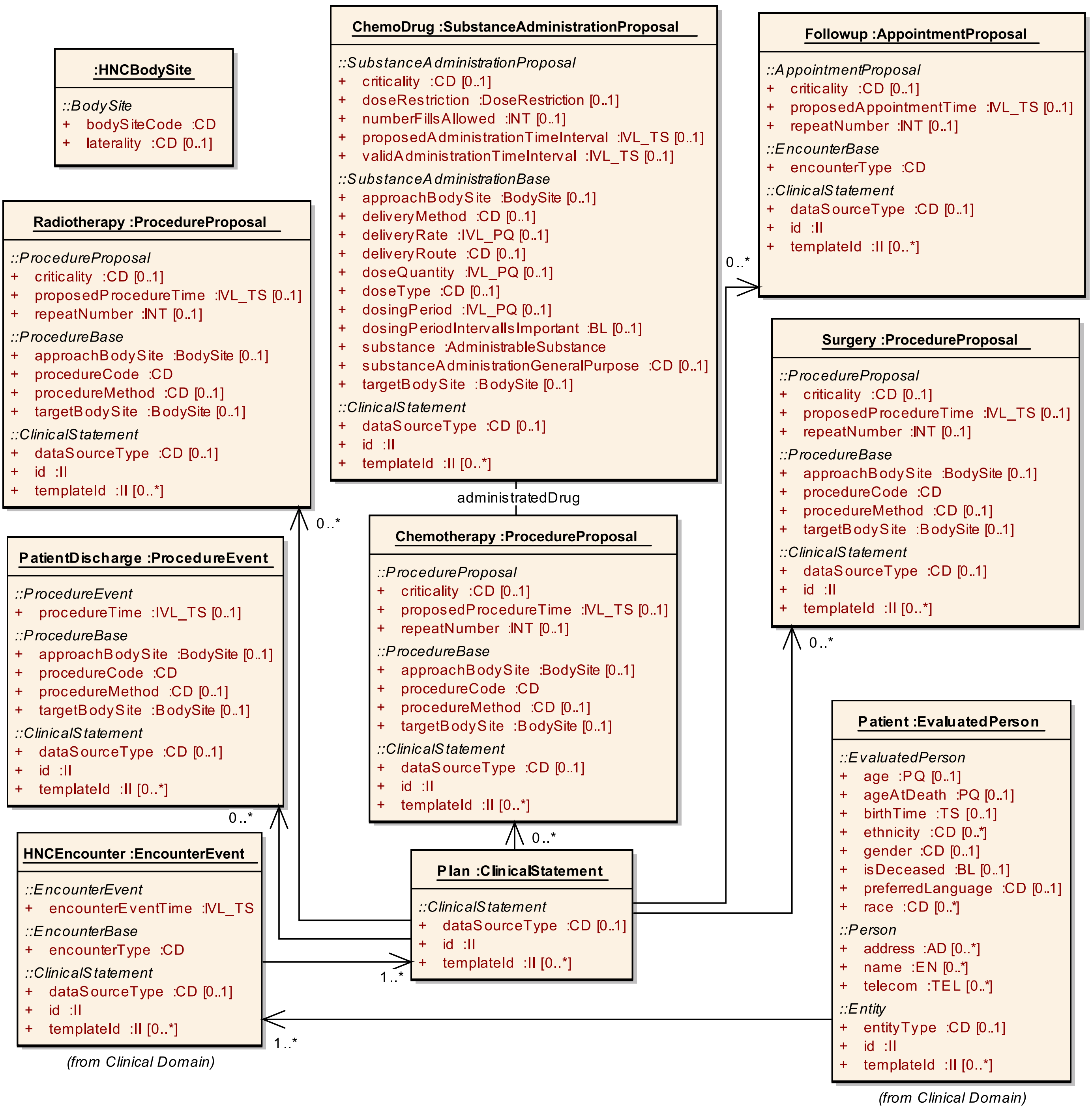

Fig. A.15. Plan model of DCM.

Appendix B. Semantic reconciliation model detail

In Table B.6, columns 1, 2, and 3 show some of the mappings between DCM, SNOMED CT, and vMR concepts, while columns 4, 5, and 6 (*DCM Values Set, SNOMED CT Code (for Values set)*, and *vMR Concept Attributes (for value set)*) show the mapping of corresponding values sets of DCM, SNOMED CT, and vMR concepts, respectively. Table B.6 partially depicts the three types of mappings in SRM. The total number of DCM concepts used in the HMIS system of SKMCH is 269: 258 were mapped correctly to SNOMED CT codes, while 11 were localized concepts missing from mappings. Table B.6 lists only the concepts that were used in the case study for system realization.

Table B.6
DCM concepts mapping to corresponding vMR and SNOMED CT concepts.

DCM Concept	SNOMED CT Concept	vMR Concepts	DCM Values Set	SNOMED CT Code (for Values Set)	vMR Concepts Attributes (for Values Set)
Clinical Stage T	385356007 Tumor stage finding (finding)	ObservationResult (observationFocus)	T0	58790005	observationValue
			T1	23351008	
			T2	67673008	
			T3	14410001	
			T4/T4a	65565005	
			T4b	396731008	
			Tx	43189003	
Clinical Stage N	385382003 Node (category finding (finding), N stage finding, Node category finding, Node stage finding)	ObservationResult (observationFocus)	N0	62455006	observationValue
			N1	53623008	
			N2	46059003	
			N3	5856006	
			N2a	261967001	
			N2b	370008004	
			N2b	370010002	
			Nx	79420006	
Clinical Stage S	80631005 Clinical stage finding (finding), Clinical stage finding	ObservationResult (observationFocus)	I	13104003	observationValue
			II	60333009	
			III	50283003	
			IV/IV A	2640006	
			IV B	1523005	
			IV C	33177002	
Treatment Intent	395077000 Treatment intent (situation)	ProcedureEvent (procedureCode)	Radical	27762005	procedureMethod
			Palliative	363676003	
			Consultation	11429006	
Histology Description	250537006 Histopathology finding	Problem (problemCode)	Squamous cell carcinoma	402815007	problemCode
			Small cell carcinoma	74364000	
			Carcinoma NOS	68453008	
			Adenocarcinoma	35917007	
			Adenoid cystic carcinoma	11671000	
			Adenoid cystic carcinoma	1338007	
			Squamous cell carcinoma in situ	59529006	
			Verrucous carcinoma	89906000	
			Malignant melanoma	2092003	
			Pleomorphic adenoma	8360001	
			Spindle cell carcinoma	65692009	
			Ameloblastoma, malignant	88253001	
			Adenoid squamous cell carcinoma	85956000	
			nasopharyngeal carcinoma	449248000	
			Sebaceous adenocarcinoma	54734006	
			Sarcoma, not otherwise specified	397355008	
			Plasmacytoma, not otherwise specified	415112005	
			Mucoepidermoid carcinoma	4079000	
Treatment Plan	1. 413737006 Cancer hospital treatment completed (situation) 2. 225292002 Developing a treatment plan (procedure)	ProcedureEvent (procedureCode)	Chemotherapy	367336001	procedureMethod
			CRT (Chemoradiotherapy)	703423002	
			RT (Radiotherapy)	108290001	
			Surgery	387713003	
			Induction Chemotherapy	450827009	

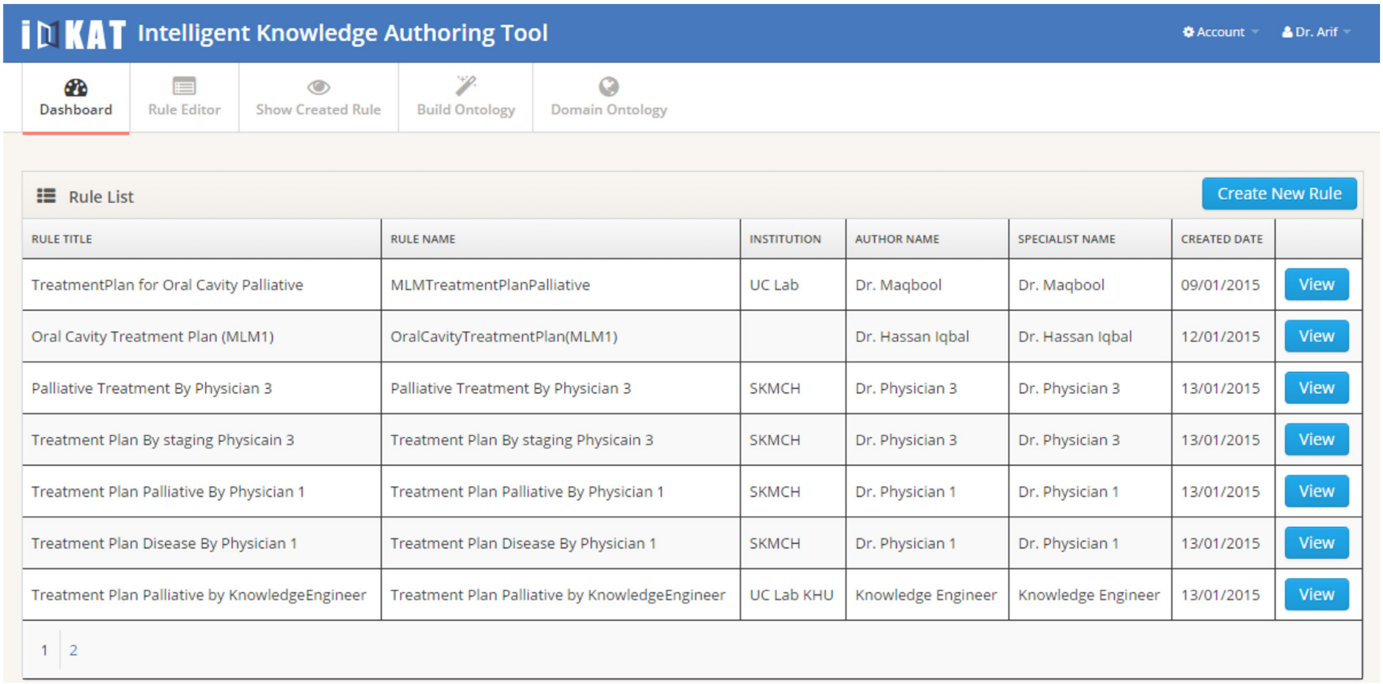

Fig. C.16. Dashboard for existing MLM.

Appendix C. System implementation and realization

We developed a web-based system and deployed it in a testing environment. In the user interface, we provided different screens such as *Rules List, Rule Editor*, and *Rule Viewer*.[2] The system provides a list of previously created MLMs with abstract information about the MLM, as shown in Fig. C.16. This interface allows a physician to view and update the complete details of previously created MLMs. When a physician clicks on the *View* button, the corresponding MLM is shown in the *Rule Editor* in an editable form. The list screen also provides the functionality of adding a new rule through the *Create New* Rule button.

The main interface for rule creation is *Rule Editor*, as shown in Fig. C.17. Highlighted Area 1 is used to capture metadata about the MLM such as *Rule Title, Author's name, MLM name, Institution*, and *Created date*. Similarly, the physician can use the citation button to attach online resources as evidence of the MLM [52]. The *Purpose* and *Explanation* boxes allow the physician to enter the rule purpose and provide explanation.

Highlighted Area 2 handles the main logic of the rule. It contains two boxes: *Condition* and *Action*. The *Condition* box allows the physician to write the facts involved in the condition part of the rule. The *Action* box is used to write the conclusion of the rule. This interface alleviates the physician from knowing the technical details of SNOMED CT, HL7 vMR, and the complex artifacts of HL7 Arden Syntax.

Physicians can select the DCM concepts from the DCM concepts tree shown in highlighted Area 3 by double-clicking on the required concept. The *Domain Clinical Model Concepts* option allows the domain concept to be brought from DCM, and *SNOMED CT Concepts* provides an enhanced search on the SNOMED ontology to obtain the domain concepts. While writing a condition or action statement, the physician can use either the tree model or the IntelliSense feature, as shown in Fig. C.18.

When the physician wants to save a created rule by clicking the *Save Rule* button, the corresponding MLM is generated in the back-end process. The generated MLM is represented in standard data model vMR concepts and SNOMED CT codes, instead of concepts in the understandable rule format on the user interface. After successful generation of the MLM, it is stored in the MLM knowledge base as text files and in the database repository in structured format. Physicians can see the newly created MLM by clicking the *Show Created MLM* button, and the result is displayed on the *Rule Viewer* page, as shown in Fig. C.19.

[2] Video Demo for review process: Please download video of rule creation using I-KAT https://goo.gl/Y8eHeu.

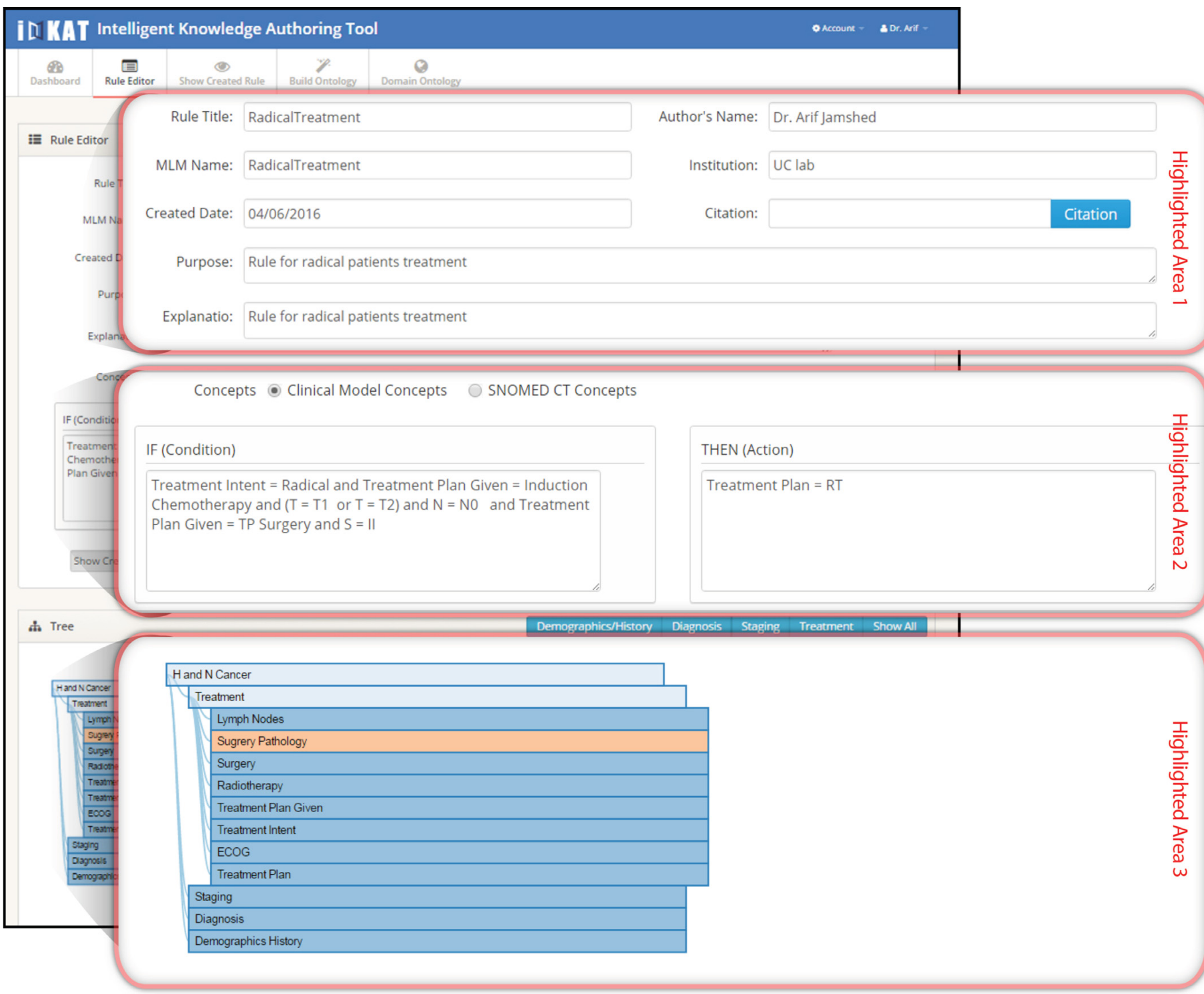

Fig. C.17. Rule editor for MLM creation.

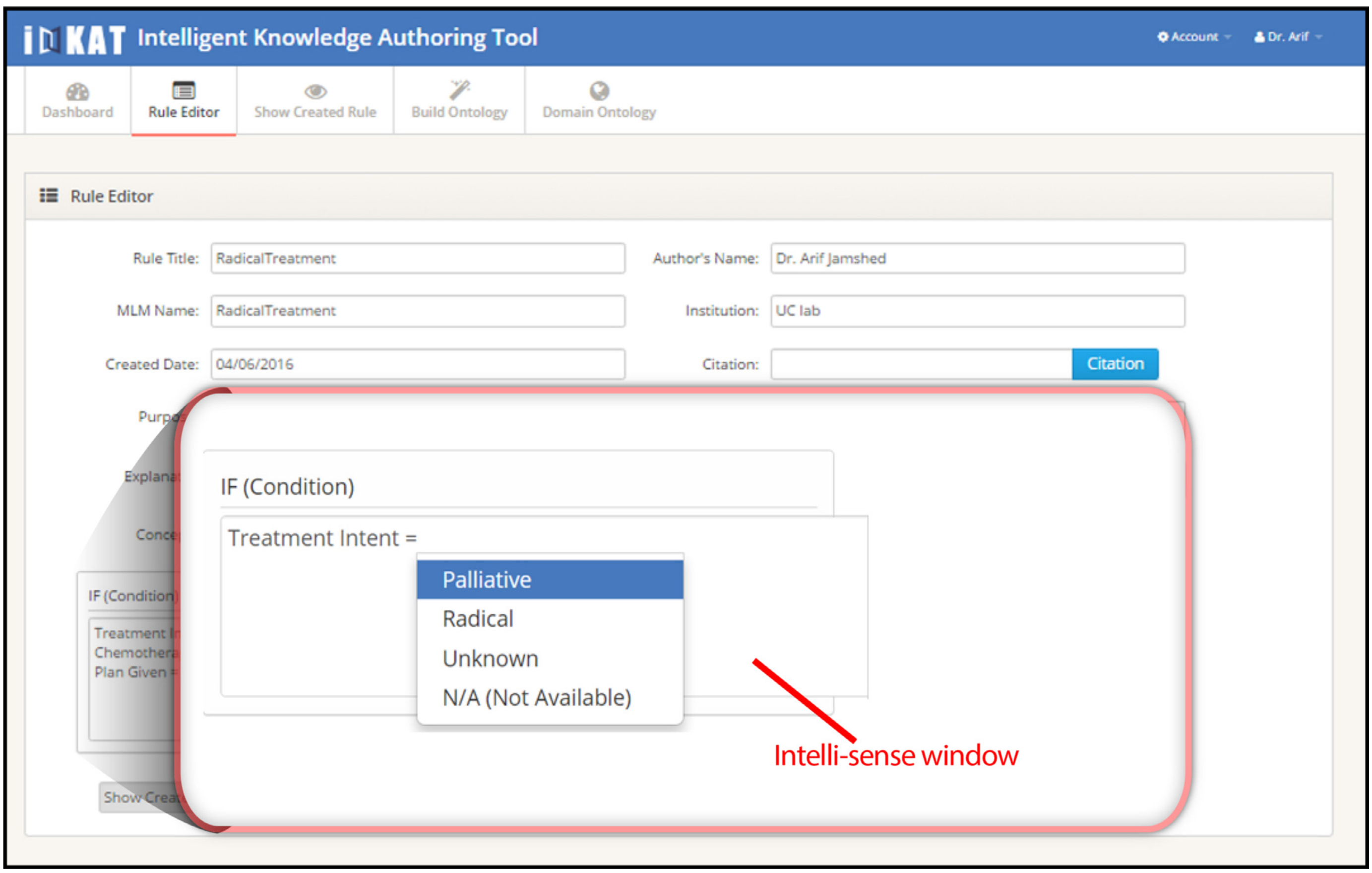

Fig. C.18. Detailed view of Rule 1.

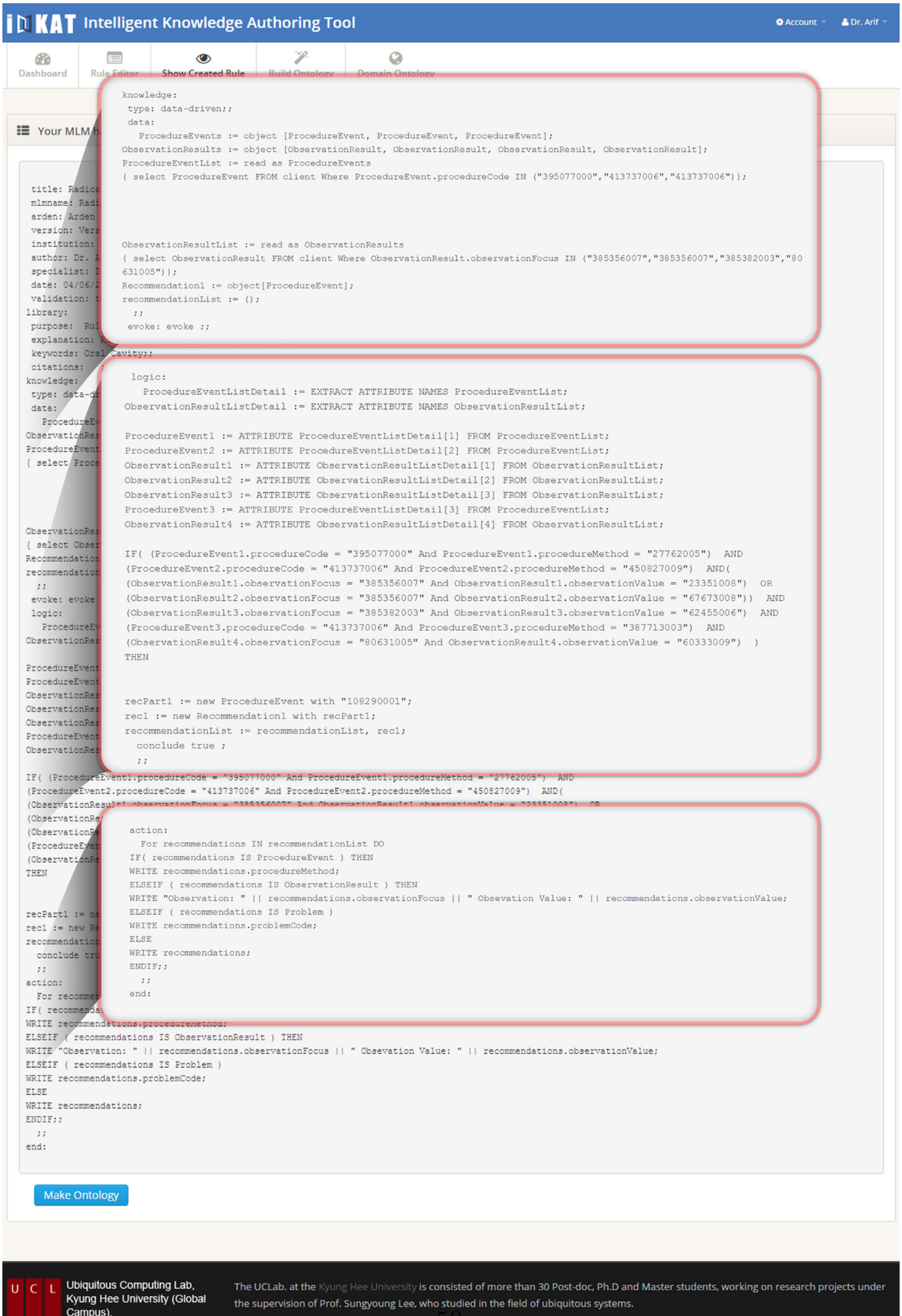

Fig. C.19. MLM view for Rule 1.

Appendix D. Detailed explanation of the generated MLM

In the newly created MLM, all information is mapped and shown in the correct slots of the MLM. This information consists of *title, mlmname, institution, purpose*, and *explanation*. The *data* and *logic* slots are saved with integration of vMR concepts and SNOMED CT codes. In *data*, the system creates input objects of the vMR classes. These objects include three for *ProcedureEvent* for the DCM concept *Treatment Intent*, two for *Treatment Plan Given (Chemoinduction and Surgery)*, and four for *ObservationResult* for DCM concepts of *T (T1 and T2)*, one concept for *N*, and one concept for *Clinical Stage (S)*. The system requests input values for objects from the client as shown in lines 21–25 in MLM Listing 1. Similarly, the system inputs the SNOMED CT codes (e.g., *395077000* for *Treatment Intent, 413737006* for concept *Treatment Plan Given, 413737006* for concept *T, 385382003* for concept *N*, and *80631005* for concept *S*) in lines 22 and 25. Lines 31–32 show the declaration of two output recommendations; both of these recommendations belong to the vMR class *ProcedureEvent* for the *Treatment Plan*.

In *logic*, the system extracts values of *Treatment Intent, Treatment Plan Given, T, N*, and *S* from the lists of *ProcedureEvent* and *ObservationResult* shown in lines 34–40. Lines 42–54 show the *IF, THEN* part of the MLM. In *IF*, the *ProcedureEvent1.procedureCode = "395077000"* shows the *Treatment Intent* and *ProcedureEvent1. ProcedureMethod = "27762005"* shows the *Radical*. Likewise, *ObservationResult1.observationFocus = "385356007"* is used for clinical stage *T*, while *ObservationResult1.observationValue = "23351008"* shows the value *T1*. In the same manner, the key and values of facts *"T = T2," "N = N1," "Treatment Plan Given = Chemoinduction,"* and *"Treatment Plan Given = Surgery"* are generated.

In the *Then* part of *logic*, the system creates output object *"ProcedureEvent"* with SNOMED CT codes *"108290001"* for *Radiotherapy* in line 51. In the *action* slot of MLM, the system writes all generated output objects in lines 57–66 that were created for recommendation. All DCM, vMR, and SNOMED CT mappings are listed in Table B.6, Appendix B.

```
maintenance:
 title: RadicalTreatment;;
 mlmname: RadicalTreatment;;
 arden: Arden Syntax V2.7;;
 version: Version 2.7;;
 institution: UC lab;;
 author: Dr. Arif Jamshed;;
 specialist: Dr. Arif Jamshed;;
 date: 04/06/2015;;
 validation: testing;;
library:
 purpose:  Rule for radical patients treatment;;
 explanation: Rule for radical patients treatment;;
 keywords: Oral Cavity;;
 citations:   ;;
knowledge:
 type: data-driven;;
 data:
   ProcedureEvents := object [ProcedureEvent, ProcedureEvent, ProcedureEvent ];
ObservationResults := object [ObservationResult, ObservationResult, ObservationResult, ObservationResult];
ProcedureEventList := read as ProcedureEvents
{ select ProcedureEvent FROM client Where ProcedureEvent.procedureCode IN ("395077000","413737006","413737006")};

ObservationResultList := read as ObservationResults
{ select ObservationResult FROM client Where ObservationResult.observationFocus IN ("385356007","385356007","385382003","80631005")};
Recommendation1 := object[ProcedureEvent];
recommendationList := ();
  ;;
 evoke: evoke ;;
 logic:
   ProcedureEventListDetail := EXTRACT ATTRIBUTE NAMES ProcedureEventList;
ObservationResultListDetail := EXTRACT ATTRIBUTE NAMES ObservationResultList;

ProcedureEvent1 := ATTRIBUTE ProcedureEventListDetail[1] FROM ProcedureEventList;
ProcedureEvent2 := ATTRIBUTE ProcedureEventListDetail[2] FROM ProcedureEventList;
ObservationResult1 := ATTRIBUTE ObservationResultListDetail[1] FROM ObservationResultList;
ObservationResult2 := ATTRIBUTE ObservationResultListDetail[2] FROM ObservationResultList;
ObservationResult3 := ATTRIBUTE ObservationResultListDetail[3] FROM ObservationResultList;
ProcedureEvent3 := ATTRIBUTE ProcedureEventListDetail[3] FROM ProcedureEventList;
ObservationResult4 := ATTRIBUTE ObservationResultListDetail[4] FROM ObservationResultList;

IF( (ProcedureEvent1.procedureCode = "395077000" And ProcedureEvent1.procedureMethod = "27762005")  AND
(ProcedureEvent2.procedureCode = "413737006" And ProcedureEvent2.procedureMethod = "450827009")  AND(
(ObservationResult1.observationFocus = "385356007" And ObservationResult1.observationValue = "23351008")  OR
(ObservationResult2.observationFocus = "385356007" And ObservationResult2.observationValue = "67673008"))  AND
(ObservationResult3.observationFocus = "385382003" And ObservationResult3.observationValue = "62455006")  AND
(ProcedureEvent3.procedureCode = "413737006" And ProcedureEvent3.procedureMethod = "387713003")  AND
(ObservationResult4.observationFocus = "80631005" And ObservationResult4.observationValue = "60333009")  )
THEN

recPart1 := new ProcedureEvent with "108290001";
rec1 := new Recommendation1 with recPart1;
recommendationList := recommendationList, rec1;
  conclude true ;
  ;;
action:
  For recommendations IN recommendationList DO
IF( recommendations IS ProcedureEvent ) THEN
WRITE recommendations.procedureMethod;
ELSEIF ( recommendations IS ObservationResult ) THEN
WRITE "Observation: " || recommendations.observationFocus || " Obsevation Value: " || recommendations.observationValue;
ELSEIF ( recommendations IS Problem )
WRITE recommendations.problemCode;
ELSE
WRITE recommendations;
ENDIF;;
  ;;
end:
```

Listing 1. Created MLM for oral cavity cancer treatment plan of Rule 5.

Appendix E. Interpreted requirements for the authoring tool

The following table shows the final list of selected and interpreted requirements according to the knowledge acquisition tool. The *Priority* column shows the requirement category (i.e., E for *Essential*, O for *Optional*, and R for *Recommended*). *Req. Number* column shows the requirement number. *Description* shows the requirement statement for CIMT. The *Interpretation for Knowledge Authoring Tool* column describes the requirement statement with respect to knowledge acquisition tools. The column *Implementation Methodology* shows the corresponding methodology to implement the requirement in our system.

The following table shows the final list of selected and interpreted requirements according to knowledge acquisition tools. The *Priority* column shows the category of requirement i.e. E for *Essential*, O for *Optional*, and R for *Recommended. Req. Number* column shows number of requirement, *Description* shows the statement of requirement for CIMT. *Interpretation for Knowledge Authoring Tool* column describes the statement of requirement with respect to knowledge acquisition tools. The column *Implementation Methodology* shows the corresponding methodology to implement the requirement in our system.

Table E.7
Classification and interpretation of requirements for CIMT to the Intelligent-Knowledge Authoring Tool.

Priority	Req. Number	Description	Interpretation for Knowledge Authoring Tool	Implementation Methodology
E	R1	Be able to define clinical information models accord-ing to a defined technical specification for structur-ing clinical information in EHR systems.	**IR1**: Knowledge Authoring Tool able to create knowledge that is aligned with technical specifications for structuring the clinical information in EHR.	Our proposed system generates the MLMs using HL7 standard data mod-el vMR for structuring and easy integrating with EHR systems.
E	R2	Support the semantic interoperability of EHR systems (Data Model, Std terminology).	**IR2**: Create knowledge support for easy integration with EHR workflows.	We are using standard the data model vMR and standard terminologies of SNOMED CT to enha-nce interoperability.
E	R3	Ensure consistency of information collected by enabling the definition of clinical information models generic enough to be compatible in multiple scenarios through specialization mechanisms for the additional constraints of each local scenario.	**IR3**: KAT should rely on and bind to local and standard clinical models and vocabulary for easy rule construction for localized recommendation interventions.	For MLM generation, the SRM maps DCM concepts with SNOMED CT and vMR. When new local concepts are added to the model, it maps them with corresponding vMR class and then with corresponding top-level concepts of SNOMED CT, which allows localized rule creation under the standard constraints.
E	R4	Definition and validation of the clinical information models according to a formal syntax.	**IR4**: KAT should support the validation process to ensure the validity of clinical rules.	We validate the created MLM with the structure and syntax of standard HL7 MLM.
E	R5	Import and export clinicalinformation models acco-rding to the following formal syntaxes: XML and ADL.	**IR5**: KAT allows transfor-mation of rules into multiple formats specified by the knowledge representation scheme.	Currently, our system presents knowledge in MLM format; in compilation module, we will represent it in ArdenML.
E	R6	Represent data types according an accepted data type standard (e.g. ISO 21090 standard or a subset of this).	**IR6**: KAT should support standard data types according to the used standard data model for rule creation.	We are using standard datatypes of MLM that are standard for HL7 community.
E	R8	Provide an automatic parser for the defined clinical information model.	**IR8**: KAT includes parsers for different supported knowledge representation schemes.	Our system has parsers to re-read the created rules from MLM text files as well as from the knowledge base.
E	R9	Tools will verify that clinical information model and their instances are semantically and syntactically consistent.	**IR9**: KAT should semantically and syntactically validate rules according to the representation scheme.	According to SRM, our system validates the used concepts with vMR and SNOMED CT hierarchy, while validating the structure according to the HL7 MLM standard.
E	R10	The tool allows the author to create term bindings by connecting with Terminology Servers using (e.g. using CTS2) or another suitable terminology server commu-nication specification.	**IR10**: KAT needs to bind concepts to standard terminologies to enhance shareability and simplify integration.	Our system binds the rule editor with standard terminologies of SNOMED CT for easy selection of the desired concepts; internally, the MLM is generated with SNOMED CT codes of the corresponding concepts as well.

(*continued on next page*)

Table E.7 (*continued*)

Priority	Req. Number	Description	Interpretation for Knowledge Authoring Tool	Implementation Methodology
E	R11	Should include an intuit-ive graphical user interface for navigating large taxonomies.	**IR11**: KAT should manage large number of rules and their dependencies in easy and understandable GUIs.	Our system provides DCM concepts in tree form for easy navigation and selection of concepts while large number of rules can be navigated through the provided dashboard.
E	R13	Should include mechan-isms that enable users and find a clinical infor-mation models in the repository by searching on any of its structured information properties.	**IR13**: KAT allows easy interface for searching the large number of rules within the knowledge base.	Our system provides facility to search the existing rules in the knowledge base using the dashboard, while searchin the desired concept in the DCM tree using the defined category panels.
E	R14	Should export its clinical information model in at least one format that conforms to a published international standard or specification.	**IR14**: KAT should support at least one standard knowledge representation format.	Our system generates HL7 standard Arden Syntax MLM to share with other organizations.
E	R16	Should allow collabora-tive authoring of clinical information models accor-ding to the established roles. As well as recor-ding experts and organiza-tion participating in this process.	**IR16**: KAT should support roles to identify and manage the ownership of the created knowledge rules.	Our system provides facility for role management, i.e., each user has access to the knowledge base according to their roles.
E	R17	Should provide mechan-isms to support multiple language translations of a clinical information model.	**IR17**: KAT should prov-ide multilingual support for knowledge creation to cover maximum regions.	Currently, our system has no functionality to create MLMs in different languages.
E	R18	Should enable the formal definition of clinical content by domain experts without the need for technical understanding.	**IR18**: KAT allows abstraction to use localized concepts and enables automatic transformation of the underlying knowledge representation scheme while hiding the underlying technical complexity of concepts and syntax.	Our system provides abstraction to users for writing complex structure and syntax of MLM. The experts do not deal with complex structure of MLM and data model vMR.
E	R19	Should ensure the defin-ition of purpose, approp-riate description of usage, and precise mention of clinical information model domain.	**IR19**: KAT should ensure the meta information of each rule from the expert in self-explanatory manner.	The users can enter information about pur-pose, functionality, and other rule details using an easy-to-use interface. The system saves information in the maintenance slot of MLM.
O	R21	Facilitate the implement-ation of EHR systems that meet clinical requirements.	**IR21**: KAT should have standard conceptual models that enable easy integration of knowledge base with EHR workflows.	Implementation methodology for R1.
O	R45	Import/select the Reference Model that will lead underpin the definition.	**IR45**: The conceptual model used in KAT should be validated using standard reference model.	We selected vMR as the reference model, and it leads underpinning of the definition.
O	R48	Tools should suggest clinical information modelers with candidate terminology/ontology terms based on their semantic underlying model.	**IR48**: KAT should suggest candidate standard termin-ologies when experts write knowledge rules.	We provide SNOMED CT terminologies as standard.
O	R51	Should integrate or link to educational material to teach clinicians how to participate either in core and validation domain expert group.	**IR51**: Should integrate or link to educational material to teach clinicians how to participate in core and validation domain expert group.	Our system facilitates experts to link some educational material to the rules as evidence.

(*continued on next page*)

Table E.7 (*continued*)

Priority	Req. Number	Description	Interpretation for Knowledge Authoring Tool	Implementation Methodology
O	R52	Should allow to assign or edit the GUI presentation capabilities for local purposes, making possible that clinician/administrator edit the local presentation.	**IR52**: KAT should allow editsto the GUI presentation and domain model according to the interest of experts.	We only change the GUI regarding the DCM tree with category-based selection.
R	R23	Support the organizati-onal needs relating to the definition process, with coordination capabi-lities among clinical information modelling experts and clinical teams to provide a common or consensus agreed definition of the clinical information model.	**IR23**: KAT should support the organizational needs to create domain knowledge with the help and cons-ensus of domain experts.	Our system provides a DCM concept as local concepts mapping with vMR and SNOMED CT in SRM. This model needs consensus and collaboration of clinical experts and knowledge engineers.
R	R25	Promote the clinician adoption with a simplified and guided view well understood by them that guide their participation in the modelling process.	**IR25**: KAT should provide simpl-ified and guided views to the experts and should hide all complexity when writing knowledge rules.	Our system provides an easy to understand and well-organized editor to create knowledge that hides the complex syntax and structure of the MLM.
R	R26	Define semantic and syntactic patterns in the form of constraints to on the selected Reference Model.	**IR26**: KAT should bind rule authoring to the standard data models and vocabulary to fulfill the constraints of the reference model.	We provide abstraction to MLM with the vMR data model; therefore, the expert is restricted with regard to wrong classes or attributes of vMR.
R	R27	Provide an automatic testing environment for systems using the defined clinical inform-ation model.	**IR27**: Provide a testing environment to test the behavior of newly created knowledge before production.	Our system validates the MLM at runtime, either during testing or production. In the future, a testing environment will be provided.
R	R29	Should include visuali-zation components for viewing complex term relationships.	**IR29**: Should include understandable and manag-eable components and views for domain experts to create knowledge in an easy manner.	Implementation methodology for R25.
R	R30	Should facilitate the use of the clinical infor-mation model to transform/map from existing data.	**IR30**: KAT should create knowledge rules with standard models and vocabulary to support the existing data of organizations.	Implementation methodology for R2.
R	R31	Should allow to define transformations of the clinical information models to/from other specifications.	**IR31**: KAT should allow transformation of the knowledge rules into different formats of knowle-dge representation schemes.	Implementation methodology for R5.
R	R32	A repository service should provide a noti-fication service to experts and systems about clinical inform-ation model updates, additions and backwa-rds compatibility.	**IR32**: The system should provide a notification service to experts and administrators about rules updates, additions, and backward compatibility.	According to role man-agement, whenever model or knowledge rules are changed, it will notify the persons of concern.
R	R33	Where more than one format is supported, requester user or sys-tem will be able to nominate the preferred retrieval format.	**IR33**: The tool allows the transformation of knowledge rules into multiple representation formats for retrieval of knowledge according to expert interest.	Our system facilitates retrieval of the rules in the desired format.
R	R34	Requesters of obsolete versions of an clinical information model shall be provided with a notification that an update (or updates) exist and be able to nominate the version(s) to be returned.	**IR34**: According to role management, the experts should be notified about updates in the knowledge rules and be able to nominate the correct updated version of the knowledge rule.	Implementation methodology for R32.

(*continued on next page*)

Table E.7 (*continued*)

Priority	Req. Number	Description	Interpretation for Knowledge Authoring Tool	Implementation Methodology
R	R37	Should provide mecha-nisms to assign the foll-owing roles to experts participating in the clinical information modelling process and document this informa-tion in the final clinical information model produced: editor, author and reviewer.	**IR37**: In KAT, the three main roles of editor, author, and reviewer should exist, each of which should be able to process the knowledge rules.	Implementation Methodology for R32.
R	R39	Should provide the means to define the clinical and usage scope of the clinical information model in a structured and coded format, in order to be able to check for possible scope overlap with other clinical informat-ion model.	**IR39**: The system should provide a mechanism based on standard data model and vocabulary to resolve merging conflicts between two knowledge bases.	Our system gives an immediate prompt to the expert when the logic part of a rule overlaps with existing knowledge rules during creation of rules and merging with other knowledge bases.
R	R40	Should implement clin-ician understandable mechanisms for a guided process for local specia-lization and validation purposes.	**IR40**: KAT should implement understandable and guided mechanisms for the clini-cians to adapt localized rules according to the standard data model.	Implementation methodology for R39.
R	R41	Should be able to cre-ate prototype screens for domain expert vali-dation of the defined clinical information model auto-generates example GUIs to test the creation of example instances.	**IR41**: KAT should provide GUI screens to test the rule valida-tion with real data.	Will be implemented in the future.
R	R42	User friendly interface for clinicians including drag and drop capabil-ities to be able to manage multiple clinical information models easily.	**IR42**: User-friendly interface for clinicians including drag and drop/IntelliSense functiona-lities to manage knowledge rules in an easy way.	Our system provides a user-friendly interface for rule creation with IntelliSense functiona-lity and drag and drop mechanism of concept selection from the DCM tree.
R	R43	Editorial role can exa-mine changes, and accept or reject changes.	**IR43**: Editorial role should examine the created/updated knowledge rules.	Implementation Methodology for R32.
R	ER57		**ER57**: Provide DCM in hierarchical form for easy selection of required concepts during knowledge creation.	Our system provides aDCM tree that contains all understandable domain concepts used in local HMIS systems.
R	ER58		**ER58**: The knowledge editor should provide contextual selection of a required value of a concept from the value set using the IntelliSense window.	Our system facilitates physician selection of the desired concepts from the IntelliSense window during rule creation, which populate from DCM concepts or SNOMED CT concepts that depends on the experts choice.

References

[1] E. Sanchez, C. Toro, E. Carrasco, G. Bueno, C. Parra, P. Bonachela, M. Graña, F. Guijarro, An architecture for the semantic enhancement of clinical decision support systems, in: Knowledge-Based and Intelligent Information and Engineering Systems, Springer, 2011, pp. 611–620.

[2] M. Hussain, M. Afzal, W.A. Khan, S. Lee, Clinical decision support service for elderly people in smart home environment, in: Control Automation Robotics & Vision (ICARCV), 2012 12th International Conference on, IEEE, 2012, pp. 678–683.

[3] C.Y. Jung, K.A. Sward, P.J. Haug, Executing medical logic modules expressed in ardenml using drools, J. Am. Med. Inform. Assoc. 19 (4) (2012) 533–536.

[4] Y. Zhang, S. Wang, P. Sun, P. Phillips, Pathological brain detection based on wavelet entropy and Hu moment invariants, Biomed. Mater. Eng. 26 (s1) (2015) S1283–S1290.

[5] Y.-D. Zhang, S. Chen, S.-H. Wang, J.-F. Yang, P. Phillips, Magnetic resonance brain image classification based on weighted-type fractional fourier transform and nonparallel support vector machine, Int. J. Imaging Syst. Technol. 25 (4) (2015) 317–327.

[6] J. Moss, E.S. Berner, Evaluating clinical decision support tools for medication administration safety in a simulated environment, Int. J. Med. Inform. 84 (5) (2015) 308–318.

[7] J.A. Osheroff, J.M. Teich, B. Middleton, E.B. Steen, A. Wright, D.E. Detmer, A roadmap for national action on clinical decision support, J. Am. Med. Inform. Assoc. 14 (2) (2007) 141–145.

[8] K. Fehre, K.-P. Adlassnig, Service-oriented arden-syntax-based clinical decision support, in: Proceedings of eHealth2011. Vienna: Austrian Computer Society, 2011, pp. 123–128.

[9] A. Wright, D.F. Sittig, A four-phase model of the evolution of clinical decision support architectures, Int. J. Med. Inform. 77 (10) (2008) 641–649.

[10] T. Ali, M. Hussain, W.A. Khan, M. Afzal, B.H. Kang, S. Lee, Arden syntax studio: creating medical logic module as shareable knowledge, in: Innovations in Intelligent Systems and Applications (INISTA) Proceedings, 2014 IEEE International Symposium on, IEEE, 2014, pp. 266–272.

[11] D.F. Sittig, A. Wright, J.A. Osheroff, B. Middleton, J.M. Teich, J.S. Ash, E. Campbell, D.W. Bates, Grand challenges in clinical decision support, J. Biomed. Inform. 41 (2) (2008) 387–392.

[12] J. Choi, Y.A. Lussier, E.A. Mendonça, Adapting current arden syntax knowledge for an object oriented event monitor, in: AMIA Annual Symposium Proceedings, 2003, American Medical Informatics Association, 2003, p. 814.

[13] H.K. Park, J. pil Kim, J.W. Choi, J.J. Hwang, S.M. Ha, Design of real-time arden syntax based decision support system with minimum workload and building cost, in: Enterprise Networking and Computing in Healthcare Industry, 2005. HEALTHCOM 2005. Proceedings of 7th International Workshop on, IEEE, 2005, pp. 430–433.

[14] G. Hripcsak, P. Ludemann, T.A. Pryor, O.B. Wigertz, P.D. Clayton, Rationale for the arden syntax, Comput. Biomed. Res. 27 (4) (1994) 291–324.

[15] R. Jenders, H. Huang, G. Hripcsak, P. Clayton, Evolution of a knowledge base for a clinical decision support system encoded in the arden syntax, in: Proceedings of the AMIA Symposium, American Medical Informatics Association, 1998, p. 558.

[16] A. Moreno-Conde, F. Jódar-Sánchez, D. Kalra, Requirements for clinical information modelling tools, Int. J. Med. Inform. 84 (7) (2015) 524–536.

[17] W.A. Khan, A.M. Khattak, M. Hussain, M.B. Amin, M. Afzal, C. Nugent, S. Lee, An adaptive semantic based mediation system for data interoperability among health information systems, J. Med. Syst. 38 (8) (2014) 1–18.

[18] K. Kawamoto, G. Del Fiol, H.R. Strasberg, N. Hulse, C. Curtis, J.J. Cimino, B.H. Rocha, S. Maviglia, E. Fry, H.J. Scherpbier, et al., Multi-national, multi-institutional analysis of clinical decision support data needs to inform development of the hl7 virtual medical record standard, in: AMIA Annual Symposium Proceedings, 2010, American Medical Informatics Association, 2010, p. 377.

[19] S. Kim, P.J. Haug, R.A. Rocha, I. Choi, Modeling the arden syntax for medical decisions in xml, Int. J. Med. Inform. 77 (10) (2008) 650–656.

[20] P.D. Johnson, S.W. Tu, M. Musen, I. Purves, A virtual medical record for guideline-based decision support, in: Proceedings of the AMIA Symposium, American Medical Informatics Association, 2001, p. 294.

[21] S.M. Huff, R.A. Rocha, C.J. McDonald, G.J. De Moor, T. Fiers, W.D. Bidgood, A.W. Forrey, W.G. Francis, W.R. Tracy, D. Leavelle, et al., Development of the logical observation identifier names and codes (loinc) vocabulary, J. Am. Med. Inform. Assoc. 5 (3) (1998) 276–292.

[22] Z. Aleksovski, M. Sevenster, Identifying breast cancer concepts in SNOMED-CT using large text corpus, in: Electronic Healthcare, Springer, 2012, pp. 27–34.

[23] A. Wright, D.F. Sittig, A framework and model for evaluating clinical decision support architectures, J. Biomed. Inform. 41 (6) (2008) 982–990.

[24] R.A. Jenders, B. Dasgupta, Challenges in implementing a knowledge editor for the arden syntax: knowledge base maintenance and standardization of database linkages., in: Proceedings of the AMIA Symposium, American Medical Informatics Association, 2002, p. 355.

[25] T. Ali, M. Hussain, W. Ali Khan, M. Afzal, S. Lee, Authoring tool: acquiring sharable knowledge for smart cdss, in: Engineering in Medicine and Biology Society (EMBS), 2013 35th Annual International Conference of the IEEE, IEEE, 2013, pp. 1278–1281.

[26] V.L. Patel, J.F. Arocha, A.W. Kushniruk, Patients' And physicians' understanding of health and biomedical concepts: relationship to the design of EMR systems, J. Biomed. Inform. 35 (1) (2002) 8–16.

[27] M. Samwald, K. Fehre, J. De Bruin, K.-P. Adlassnig, The arden syntax standard for clinical decision support: experiences and directions, J. Biomed. Inform. 45 (4) (2012) 711–718.

[28] ARDENSUITE, medical knowledge representation and rule-based inference software with arden syntax, 2017, (http://www.medexter.com/arden-syntax/arden-syntax). (Last visited in March 2017).

[29] R.A. Jenders, K.-P. Adlassnig, K. Fehre, P. Haug, Evolution of the arden syntax: key technical issues from the standards development organization perspective, Artif. Intell. Med. (2016).

[30] M. Gietzelt, U. Goltz, D. Grunwald, M. Lochau, M. Marschollek, B. Song, K.-H. Wolf, Arden2bytecode: a one-pass arden syntax compiler for service-oriented decision support systems based on the osgi platform, Comput. Methods Progr. Biomed. 106 (2) (2012) 114–125.

[31] V. Anand, A.E. Carroll, P.G. Biondich, T.M. Dugan, S.M. Downs, Pediatric decision support using adapted arden syntax, Artif. Intell. Med. (2015).

[32] D. Dunsmuir, J. Daniels, C. Brouse, S. Ford, J.M. Ansermino, A knowledge authoring tool for clinical decision support, J. Clin. Monit. Comput. 22 (3) (2008) 189–198.

[33] A. Seitinger, K. Fehre, K.-P. Adlassnig, A. Rappelsberger, E. Wurm, E. Aberer, M. Binder, An arden-syntax-based clinical decision support framework for medical guidelines Lyme borreliosis as an example, Stud. Health Technol. Inform. 198 (2013) 125–132.

[34] Object management group business process model and notation, 2017, (http://www.bpmn.org/). (Last visited in February 2017).

[35] S. Ferrante, S. Bonacina, G. Pozzi, F. Pinciroli, S. Marceglia, et al., A design methodology for medical processes, Appl. Clin. Inform. 7 (1) (2016) 191–210.

[36] Exceptional high-end modeling power, 2017, (http://www.sparxsystems.com/products/ea/). (Last visited in February 2017).

[37] L. Raymond, G. Paré, D. Guinea, A. Ortiz, P. Poba-Nzaou, M.-C. Trudel, J. Marsan, T. Micheneau, The contribution of office-based EMR systems to the performance of family physicians and primary care medical practices, in: System Sciences (HICSS), 2015 48th Hawaii International Conference on, IEEE, 2015, pp. 3033–3042.

[38] G. Paré, L. Raymond, A.O. de Guinea, P. Poba-Nzaou, M.-C. Trudel, J. Marsan, T. Micheneau, Barriers to organizational adoption of EMR systems in family physician practices: a mixed-methods study in Canada, Int. J. Med. Inform. 83 (8) (2014) 548–558.

[39] J.E. Frenzel, Using electronic medical records to teach patient-centered care, Am. J. Pharm. Educ. 74 (4) (2010).

[40] L.L. Weed, Medical records that guide and teach, N. Engl. J. Med. 278 (12) (1968) 652–657.

[41] J. Kibble, P.A. Hansen, L. Nelson, Use of modified soap notes and peer-led small-group discussion in a medical physiology course: addressing the hidden curriculum, Adv. Physiol. Educ. 30 (4) (2006) 230–236.

[42] S. Cameron, I. Turtle-Song, Learning to write case notes using the soap format, J. Counsel. Dev. 80 (3) (2002) 286.

[43] J. Rumbaugh, I. Jacobson, G. Booch, The Unified Modeling Language Reference Manual, Pearson Higher Education, 2004.

[44] W.A. Khan, M.B. Amin, A.M. Khattak, M. Hussain, M. Afzal, S. Lee, E.S. Kim, Object-oriented and ontology-alignment patterns-based expressive mediation bridge ontology (mbo), J. Inf. Sci. 41 (3) (2015) 296–314.

[45] P.J. Frederiks, T.P. Van der Weide, Information modeling: the process and the required competencies of its participants, Data Knowl. Eng. 58 (1) (2006) 4–20.

[46] HL7 version 3 standard, 2017, (http://www.hl7.org). (Last visited in March 2017).

[47] SNOMED CT starter guide, 2017, (http://ihtsdo.org). (Last visited in March 2017).

[48] M. Hussain, M. Afzal, T. Ali, R. Ali, W.A. Khan, A. Jamshed, S. Lee, B.H. Kang, K. Latif, Data-driven knowledge acquisition, validation, and transformation into hl7 arden syntax, Artif. Intell. Med. (2015).

[49] National comprehensive cancer network, 2017, (http://www.nccn.org). (Last visited in February 2017).

[50] P. Cremonesi, F. Garzotto, R. Turrin, User-centric vs. system-centric evaluation of recommender systems, in: Human-Computer Interaction–INTERACT 2013, Springer, 2013, pp. 334–351.

[51] R. Ali, M. Afzal, M. Hussain, M. Ali, M.H. Siddiqi, S. Lee, B.H. Kang, Multimodal hybrid reasoning methodology for personalized wellbeing services, Comput. Biol. Med. (2015).

[52] M. Afzal, M. Hussain, W.A. Khan, T. Ali, S. Lee, B.H. Kang, Knowledgebutton: an evidence adaptive tool for cdss and clinical research, in: Innovations in Intelligent Systems and Applications (INISTA) Proceedings, 2014 IEEE International Symposium on, IEEE, 2014, pp. 273–280.