Thesis for the Degree of Doctor of Philosophy

GUIDELINE ENABLED VALIDATION AND FORMAL VERIFICATION METHOD OF CLINICAL KNOWLEDGE BASE

Maqbool Hussain

Department of Computer Science and Engineering
Graduate School
Kyung Hee University
South Korea

August 2016

GUIDELINE ENABLED VALIDATION AND FORMAL VERIFICATION METHOD OF CLINICAL KNOWLEDGE BASE

Maqbool Hussain

Department of Computer Science and Engineering
Graduate School
Kyung Hee University
South Korea

August 2016

GUIDELINE ENABLED VALIDATION AND FORMAL VERIFICATION METHOD OF CLINICAL KNOWLEDGE BASE

by

Maqbool Hussain

Supervised by

Prof. Sungyoung Lee

Submitted to the Department of Computer Science and Engineering and the Faculty of Graduate School of Kyung Hee University in partial fulfillment of the requirements of the degree of Doctor of Philosophy

Dissertation Committee:

Prof. Oksam Chae

Prof. TaeChoong Chung

Prof. Tae-Seong Kim

Prof. Il-Kon Kim

Prof. Sungyoung Lee

Dedicated to my

Parents & Family

During my educational career, my father plays a pivotal role and kept me encouraged to pursue higher education. Although, he only saw my career accomplishment till master, however, it was his intentions that ultimately come true in 2016. At the time of proceeding to Korea for PhD, I was the only one to take care of my family. In those crucial times, my mother and sisters encouraged me with full support to pursue my PhD study while bearing hard time during my absence. Moreover, my wife provides full support and accompanied me in Korea while taking care of my kids.

This thesis is dedicated to my parents and my family. I will always remember the affection, enthusiasm, and encouragement that they entrusted to me that gave me strength to face all the difficulties and challenges during my course of research.

Abstract

Clinical decision support systems (CDSSs) plays pivotal role in improving patient care and enhancing practitioner performance. Nevertheless, adaption of CDSSs in an actual healthcare workflow setup is challenging. The most prominent barriers include physician fears regarding validity of the services related to the knowledge bases and quality of published guidelines, heterogeneous healthcare workflow integration, lack of standard knowledge representation, complexity of knowledge representation languages, and lack of frameworks for clinical knowledge transformation into executable knowledge bases. The trust on knowledge is the key consideration for physicians which has direct association to validation and formal verification of the knowledge acquisition process.

Using practice data as primary source for knowledge and applying data-driven knowledge acquisition (machine learning techniques), the final prediction model can be considered as potential clinical knowledge which is directly integrated into healthcare system. However, performance of these approaches are mainly dependent on the quality of data, and different techniques perform in different ways which make it difficult to select appropriate one. Moreover, the prediction model is only validated empirically (from data) and having no evidences of association with standard guideline base practices. Clinical practice guidelines (CPGs) are considered as evidence-based knowledge source and final knowledge for practices are derived from the candidate guidelines. The main issue with CPGs is generalization - where most of the clinical concepts are not present in real healthcare practices. At the same time, physicians are also interested in knowledge which reflects the practices and properly conformed from guidelines. In order to overcome these barriers,this research work introduces i) three phase knowledge acquisition and validation process model, ii) formal validation and verification for clinical knowledge and, iii) the development framework - called SmartCDSS-DF to define unified processes required for development of the CDSS.

The three phase model is an iterative process model, which consider clinical guidelines as primary source of knowledge in phase-I and create clinical knowledge model (CKM) intended to fulfill target clinical objectives. In phase-II, data-driven knowledge acquisition approach is used mainly based on analysis of real patient datasets to generate a predictive model (PM). Moreover, the PM is converted into a refined-clinical knowledge model (R-CKM), which follows a rigorous validation process. The validation process uses a clinical knowledge model (CKM), which provides the basis for defining underlying validation criteria. Finally, in phase-III, the R-CKM is converted into a set of medical logic modules (MLMs) and is evaluated using real patient data from a hospital information system.

Keeping validation and verification of knowledge as necessary requirements, mathematical model is proposed to formalize the validation process and prove its consistencies with appropriate theorems proving mechanism. Using Z notation, PM, CKM, R-CKM, and the validation process in three phase model is formally represented through Z axioms and schemas. The consistency of the validation processes is proved through initialization theorem and pre-conditions principles.

SmartCDSS-DF exploits the core concepts of Rational Unified Process (RUP), aligns it with ISO RM-ODP viewpoints and formal methods (Z notation). It helps to support system specialised requirements (knowledge validation and verification) and enables uniform view of the system in an iterative manner using two separate process pools: clinical knowledge pool and knowledge supporting tool pool.

The proposed approach in three phase model, produces the high quality R-CKM in comparison to traditional data-driven knowledge acquisition methods. We selected the oral cavity as the intended site for derivation of all related clinical rules for possible associated treatment plans. The oral cavity R-CKM was converted into four candidate MLMs, and was used to evaluate real data from 1229 patients, yielding efficient performance with 72.0% accuracy. The formal validation and verification enhanced the validation processes for refinements and introduced additional necessary validation criteria to the three phase process model. SmartCDSS-DF, resides the conceptual and technical aspects associated to knowledge acquisition into a unified process model that ultimately guarantees the consistence specification for development of CDSS.

Acknowledgement

First and foremost, I render my humble and sincere thanks to the ***Almighty Allah*** for showering HIS blessings upon me. The Almighty gave me the strength, courage, and patience during my doctoral study.

Special thanks to my advisor Prof. Sungyoung Lee who provided me guidance, strength, support, and courage in overcoming the difficult challenges throughout my time as his student. I have learned a lot from him in becoming a productive person in diverse situations. He inspired me with his dynamic personality and his unreserved help and guidance lead me to finish my thesis. He has great role in polishing my skills such as thinking, creativity, and technical soundness which are key ingredients for high quality research.

I am grateful to my dissertation evaluation committee for their insight comments and valuable suggestions during the dissertation defense. Their valuable comments improved the presentation and contents of this dissertation.

I am extremely grateful to some of my colleagues who have always provided me time, expertise, and encouragement in my course of research. They were always present to guide me in difficult situations of my PhD duration. I would like to thank Muhammad Afzal, Dr.Wajahat Ali Khan,Taqdir Ali, Rahman Ali, Shujaat Hussain, Jamil Hussain, Dr. Asad Masood Khattak, Dr. Zeeshan Pervaiz, Dr. Muhammad Fahim, and Dr. Bilal Amin. They have contributed enormously in successfully performing various academic and personal tasks that confronted me during my stay at South Korea.

I am very thankful to all of my current and former Ubiquitous computing lab fellows and colleagues for their kind support to my personal and academic life at Kyung Hee University. I am highly obliged to brilliant researchers Idris, Byunggill Go, Tae Ho Hur, Sung Hoo, Jae Hun Bang,

Bilal Ali, Kifayat Ullah, Dr.Waqas Nawaz, Aamir Saleem, Omar Farooq, Mahmood Ahmad, Saeed Ullah, Dr. Ejaz Ahmed, Dr. Hameed Sidiqqi, Abdul Sattar, Ahsan Raza Kazmi, Zain Abbas, Kamran, Maqbool Ali, Asif Razaq, Imran, Usman, Shehryar, Muhammad Telha, Ammad, Dr. Hanif Sidiqqi, Muhammad Jalil Piran and Waseem. This journey would not have been possible without their support. They contributed to my personal and academic life to polish myself. Also, I appreciate all my Korean and international friends who worked as a team with me and developed my team work skills and provided me wonderful memories during my stay in South Korea.

I am also very thankful to all collaborators from Shuakat Khanum Memorial Cancer Hospital (SKMCH) Lahore Pakistan for their enthusiasm and efforts in developing clinical decision support system for head and neck cancer and shared with us the clinical domain knowledge. I am specially thankful to Dr. Arif Jamshid, Dr. Hassan Iqbal, Dr. Irfan, Dr. Sadaf, Mr. Manzar Abbas, Mr. Idress Kokhar, and other HMIS team members for their kind support. Moreover, I am thankful to MS advisor and co-advisors in SEECS, NUST, Islamabad especially Dr. Arshad Ali, Dr. Hafiz Farooq Ahmad, Dr. Khalid Latif, and Mr. Majjid Maqbool for their guidance and continuous support during my study in Korea.

Last but not the least, I would like to express my sincere gratitude to my close friends in Pakistan (especially Mr. Muhammad Ejaz, Mr. Ajmal Hussain, Mr. Syed Murtaza, Mr. Shahzad Hussain, Mr. Mumtaz Ali, and Mr. Masnad Ali) , my father-in law, brothers-in law and other relatives who support my family during my absence.

Maqbool Hussain

August, 2016

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Overview

Clinical decision support systems (CDSS) has significant role in improving the patient care. Furthermore, substantial evidences exist, related to efficient use of CDSS improving practitioners performance [2]. In recent study at Columbia, nurses performance is evaluated using CDSS capabilities in diagnosis of obesity, smoking and depression [3]. Using CDSS capabilities, nurses performance in diagnosis increased from 4.8 to 33.9 percent for obesity , 2.3 to 11.9 percent for tobacco use, 0.2 to 8.8 percent for adult depression and 1.1 to 4.6 percent for pediatric depression. Similarly in systematic review [4], the authors have found that, "CDSS improved practitioner performance in 62 of the 97 (64%) studies assessing this outcome, including 4 of 10 (40%) diagnostic systems, 16 of 21 (76%) reminder systems, 23 of 37 (62%) disease management systems, and 19 of 29 (66%) drug-dosing or prescribing systems". Considering clinical decision support (CDS) as primary means of supporting implementation of best evidences and new knowledge at point of care, Office of the National Coordinator for Health Information Technology (ONC),Department of Health and Human Services (HHS)has included CDS as crucial feature in Meaningful Use 2 and Meaningful Use 3 criteria for certified electronic health records (EHR) [5]. With all these encouraging evidences of CDSS effective performance and adaption initiatives nation wide, overall CDSS capabilities have not been realized in real practices.

Various barriers have been investigated in literature such as; lack of knowledge validation and formal verification, resource deficiency to maintain knowledge, lack of standards in knowledge representation, and difficulties with integration in complex healthcare workflows [6]. Besides these barriers, one important aspect of difficulties in existing systems is the ignorance of the actual stakeholders - physicians to facilitate with easy to use tools for knowledge acquisition. Providing

physicians with additional responsibility of knowledge creation and maintenance using complex system interfaces overburden and deviate them from real practices. According to a group physicians' study [7], physicians complaint about complexity of the systems in terms of the obscured responsibilities which resists their reasoning capability and loses their autonomy. Furthermore, without involvement of physicians in the development of the system; firstly they will not rely on the built in rules without proper validation on the independent sample practices dataset. Secondly, even rules are validated, physicians tend to prefer their own judgments for taking care of specific patient. Thirdly, physicians some time intentionally non-adhere to the valid guidelines based rules which is predominantly valid; caused by several genuine reasons - such as patient preferences, contra-indications, and demographics [8]. Involvement of physicians in development of CDSS can tailor success of the technology and improve patient care. McGinn et. al has demonstrated significant decrease in the ordering of atntibiotics and 50% decrease in the ordering of the broad spectrum quinolones with joint consensus of group of physicians using two simple clinical rules [9]. Besides involvement of practitioners, the development process of CDSS tools should also take into consideration balance among selection of practitioners with different background and nature of tasks. As indicated by [10], information seeking strategies of resident doctors for caring patient is patient-based which is different from nurses and physician assistants where they use source-base information.

Non-practitioners staff also influence on success of CDSS implementations. Most of the existing studies overlook organization roles (non-clinicians) in implementation of CDSS which cause failure of the system [11]. The hospital administration - chief executive and medical directors play pivotal role in allocation of appropriate resources for development of the CDSS. IT staff, which manage healthcare system workflows on daily bases, facilitate development of CDSS in terms of smooth integration of CDSS with the complex healthcare system workflows.

Trust in knowledge base is considered as one of key factor that practitioners take into account while using CDSS [12]. This factor is important from the perspective of acceptance of the technology and can leverage to involve the practitioners in the development of CDSS. Furthermore, CDSS equipped with easy to use toolset which enable knowledge acquisition from real practice data and validate it against published guidelines enhance trustiness of the practitioners on the knowledge.

From practices patient dataset, machine learning methods can give robust result on deriving initial clinical knowledge [13]. Validating the initial knowledge from published clinical practice guidelines (CPGs) in the domain - such as NCCN(National Comprehensive Cancer Network) guidelines in cancer [14] - gives final clinical knowledge which practitioner can trust in using patient care.

Keeping these considerations very specific to CDSS, the existing development frameworks are not suitable for all specialized requirements of the system. Example frameworks, such as Rational Unified Process (RUP) [15, 16], HL7 HDF [17], ISO HISA [18], National eHealth Transition Authority - Interoperability Framework (NEHTA-IF) [19] and HIS-DF [20] are either generic enough or totally focused on some particular requirements. In this research work, we propose a development framework - called SmartCDSS-DF, which is influenced from existing frameworks and exploits and integrate different capabilities of these approaches into single unified and comprehensive views. For example, to target evolutionary nature of the CDSS, RUP is adopted to enable iterative development environment. To provide distinct views of the system which target all stakeholders of the system, guidelines are adopted from HIS-DF, NEHTA-IF, HL7 HDF, and HISA to use RM-ODP viewpoints. Moreover, SmartCDSS-DF also introduces validation and formal verification for specialized requirements of knowledge validation process and verification process.

1.1 Motivation

As described, despite a long history of CDSS development, most of the CDSS systems evaluated in academia have not been realized in a real clinical practice environment. Among the most common barrier, physician fears regarding validity of the services related to the knowledge bases and quality of published guidelines [6] is main consideration of this research work. We therefore initiated the Smart CDSS project in collaboration with Shaukat Khanum Memorial Cancer Hospital and Research Centre (SKMCH)[1] to provide decision support services for head and neck cancer. We observed the above-mentioned barriers while working with clinical domain experts and hospital IT staff in gathering Smart CDSS requirements.

From a broad spectrum of CDSS requirements, knowledge acquisition and validation is one

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

of the most important aspect that leads towards successful adoption of CDSS. Development of knowledge acquisition with support of rigorous validation process is required to enhance trust of the domain experts on the clinical knowledge. Figure 1.1 depicts high level representation of the CDSS knowledge acquisition process and knowledge resources.

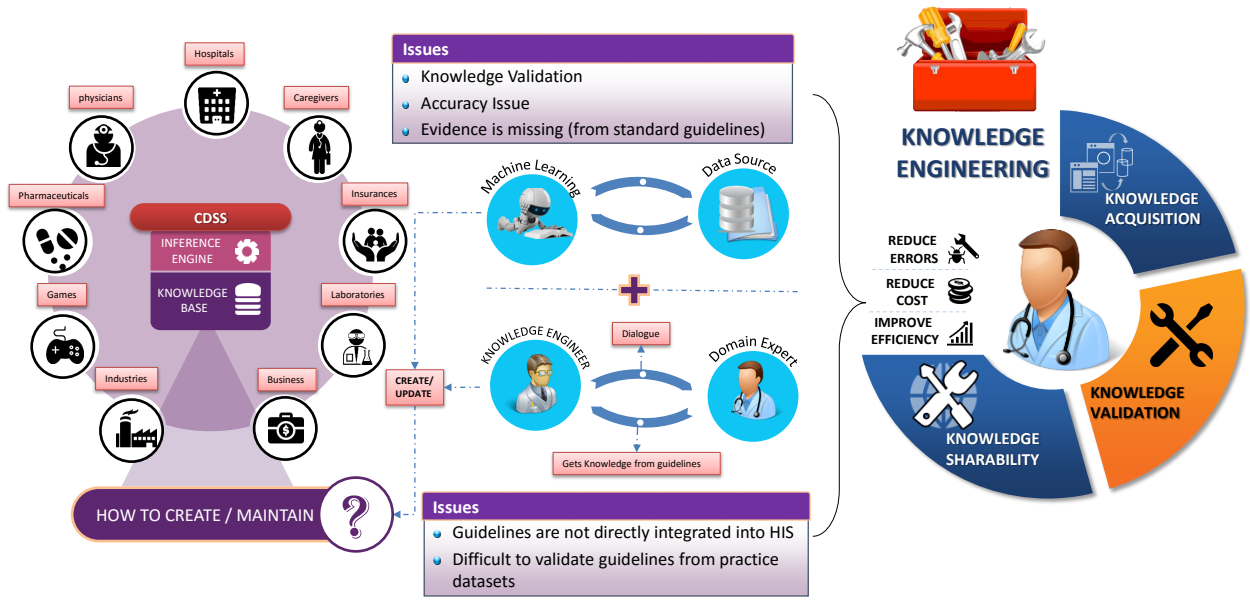

Figure 1.1: Motivations for guideline-enabled knowledge acquisition

In clinical domain, traditionally two approaches are most commonly used to acquire knowledge for CDSS. In first approach, patient data (from EMR) is used as primary resource of knowledge and machine learning algorithms are applied to create prediction models (data-driven knowledge acquisition). Data-driven knowledge acquisition is considered preferable approach to create knowledge for CDSS which reflects local practices. However, the final knowledge model has some intrinsic issues: i) it relies on statistical validations from existing data (e.g. 10-fold-cross validation), so the model performance may vary due to complexity and variety of medical dataset. ii) the model reflect only practice dataset and it have no conformance (evidence) from clinical guidelines and other resources (clinical trials, systematic reviews, and clinical meta reviews).

In second approach, the clinical guidelines are used as a primary resource of knowledge, domain experts derive the clinical knowledge and used as final executable clinical knowledge (guideline-based knowledge acquisition). The clinical knowledge which follow guidelines is con-

sidered more trustable to use across organization. However, it has limitations in adoption to CDSS: i) guidelines are too generic and it is hard to integrate it into HIS workflow. ii) it is hard to evaluate and validate the derived knowledge from patient data.

Keeping motivation to introduce knowledge acquisition method that exploits both methods and combines them into unified process using rigorous validation and verification. The method used practice dataset as primary source of knowledge - applying machine learning (decision tree) approach to create prediction model (PM). Clinical guidelines (such as for cancer: National Comprehensive Cancer Network (NCCN) [14] http://www.nccn.org/, (Accessed: 24 April, 2015)) are considered as secondary source of knowledge and using inspection method to derive clinical knowledge model (CKM). The refined clinical knowledge model (R-CKM) is created as a final knowledge after evolution of PM using rigorous validation and verification process which conforms the model from CKM. The R-CKM is finally transformed into shareable and executable format (such as HL7 Arden Syntax MLMs (Medical Logic Modules) [21]). Finally, the knowledge acquisition, validation and formal verification is incorporated into unified process framework - so called SmartCDSS-DF (Smart CDSS Development Framework), which provides unified viewpoint of the CDSS system for diverse capability stakeholders - domain experts, knowledge engineers, and developers.

CDSS requirements spectrum

Development of CDSS is naturally evolving, in terms the knowledge base should support the emerging research in the medical domain and evolve with up-to-date evidences. The evolutionary growing nature of the CDSS brings many challenges to adapt the existing system development approaches for successful deployment of CDSS in the real environment. Most importantly, the domain experts as real stakeholders and requirements providers have a dominant role in the development process which is beyond the role of the stakeholders considered mainly in the development of other software systems. The development process for CDSS needs to clearly define a set of processes in each development phase, which are feasible enough for domain experts and support efficient collaborative environment with a CDSS development team. Based on a set of desirable features of the CDSS the development framework is required to plug in appropriate processes.

Following is summary of CDSS requirements, for more granular level of requirements, readers can consult the work presented in [2, 21–25].

Identifying roles and assigning responsibilities

CDSS requires new kinds of essential people who influence the development process [22]. The people involved in the development of CDSS (often computer programmers/developers) and the people involved in providing the requirements in terms of generating the knowledge contents (usually clinicians and informaticists) are quite different [2]. Assigning the appropriate responsibilities to a diverse group of people with clearly defined roles will lead to more successful deployment of the CDSS.

Identifying candidate knowledge sources and providing formal acquisition methods

While developing CDSS, it is necessary to investigate all knowledge sources, provide an appropriate method for acquiring the knowledge, and develop the toolset required to convert this information into a computer-interpretable format. Data-driven approaches use patient data while applying machine-learning methods or semantic-based methods to acquire the knowledge; for an example see [26, 27]. Cognitive methods are used by domain experts to finalize the final knowledge from CPGs through well-defined steps [28]. By combining data-driven sources and CPGs, machine learning algorithms are applied to patient data in combination with CPGs to obtain the final clinical knowledge [1].

Specifying formal knowledge validation and verification methods

The main intention of CDSS is to enhance patient safety while decreasing errors and increasing quality. In this respect, knowledge validation and verification should be supported in the CDSS development life-cycle [24]. Knowledge acquisition from diverse sources, as well as the transformation of this knowledge into various intermediate representations (for shareability) and computer-interpretable formats, requires proper validation and verification methods to ensure that the final knowledgebase is comprised of valid knowledge that is thoroughly verified for internal consistencies.

Supporting shareable knowledge

The content of clinical studies, which represent knowledge resources, should be represented in a shareable format. Shareable content representation must adopt a formal representation of knowledge using existing standards such as HL7 Arden Syntax, GLIF, and SAGE [2]. Selection of the appropriate knowledge representation scheme is based on a consensus of the clinicians, their learning capabilities, the ability to easily understand the scheme, and the proper support of the toolset. The CDSS framework must support proper transformation processes that enable clinicians to easily represent clinical contents with a shareable knowledge representation scheme.

Integrating knowledge with healthcare system workflow

Seamless flow of information into the CDSS is challenging but is one of the key requirements. Clinical knowledge is not always integrated with the workflow of a healthcare system [24]; however, it is necessary to at least provide seamless integration with information that is already known to a healthcare system [23]. Achieving integration mainly depends on the target healthcare system, the type of CDSS knowledge representation, and the supported information flow interfaces. Unfortunately, universal plug and play for each clinical rule with the healthcare workflow is not possible; the design of CDSS should follow standard reference models (such as HL7 RIM or openEHR) [29, 30] to define standard communication interfaces.

Separating knowledge authoring from technical aspects of the system

Domain experts are the ultimate users of the CDSS. Providing a high level of abstraction over the technical aspects of the system is considered to be a common goal for knowledge management. This includes the separation of clinical contents (i.e., clinical rules) from the actual code, which operates the clinical rules (e.g., the execution environment and data integration) [2,25]. Moreover, knowledge should be represented in both human- and machine-interpretable formats with a distinct separation between technical and domain concerns.

Problem Statement

The knowledge acquisition is one of the important aspect for clinical knowledge to be acquired from divers resources - such as patient data, clinical guidelines and other online resources. The knowledge acquisition methods used from patient data are machine learning approaches - known as data-driven approaches. The knowledge acquired from guidelines and other resources are based on rigorous inspection process - known as expert driven guideline-based approaches. While combining the data-driven and guideline-based approaches for final knowledge acquisition - we call it guideline-enabled data driven approach.

Data driven approach are used mostly to reflect the actual practices of organization. However, the final knowledge acquired is not standardized. The guideline-based approach provides standardized clinical knowledge, but it is not integrable directly to healthcare workflow due to its generalized concepts. Combining both approaches can resolve the problem of both approaches provided that the final knowledge is acquired from patient data and validated against the guidelines. Current approaches in combined fashion are lacking proper validation process and mostly focusing on completion of guidelines from knowledge model created from patient data. The main goal is to introduce, methodology that exploits real practice dataset for recommendation model and leverage guideline for validating it for refined standard recommendation model. To achieve the goal, the candidate challenges for this research work are as follows:

- How to develop validation process and establish validation criteria that align diverse knowledge resources into standardized knowledge acquisition model?
- How to verify that validation process is consistent and its applicability will result in validated and consistent knowledge acquisition model?
- How to incorporate the designed validation and verification method into unified framework that support development of CDSS?

In order to provide solution to these research challenges, this work organised as follows:

- Three phase process models is proposed to consider patient data and guidelines as main knowledge resources and create refined model using rigorous validation process (Chapter 3

and Chapter 6).

- Formal methods are used to formally verify the three phase process models to ensure the refine model created after rigorous validation process is consistent and valid (Chapter 4 and Chapter 6).

- Proposed SmartCDSS-DF - a development framework to incorporate the three phase processes and formal verification processes into a unified viewpoints to support development of Smart CDSS project for head and neck cancer(Chapter 5).

1.2 Contributions

The goal of this research work is to provide a knowledge acquisition and validation method that ensure the clinical knowledge model is valid and consistent for using in final recommendation system. For validation, the clinical knowledge is acquired from patient data and conformed to the guidelines. The final knowledge is refined through rigours validation process before converting into shareable representation. To make sure the validation process is consistent and ensure valid knowledge, the validation process is formally verified. In this regard overall contributions are covered in two solutions as shown in Figure 1.2. Additional supporting solution is provided as a development framework to support adoption of the proposed methods for CDSS development.

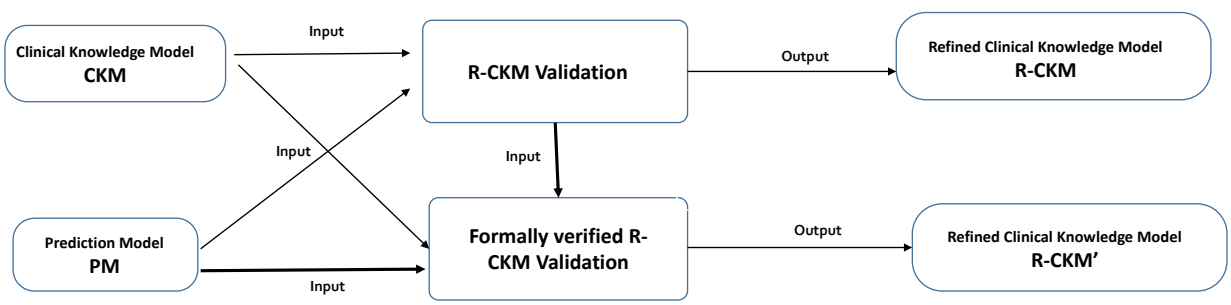

Figure 1.2: Conceptual Modeling of contribution

Validation process for R-CKM

A R-CKM validation process is rigorous process which evaluates the constituent decision paths in PM against conformance criteria. The conformance criteria reflects the recognition of CKM desired properties and some external evidences. After passing the conformance criteria the PM decision paths are refined and made parts of the R-CKM. The R-CKM, generated through this validation process inherits pros from data-driven and guideline approaches and overcome the individual limitations. For example, likewise PM, R-CKM is integrable and at the same time it is guideline conformed. So it removes the issue associated to PM for not supporting the standard practices. Similarly, likewise guidelines, R-CKM is CPGs (as it conforms to guidelines) but at the same time integrable to HIS which was not possible in case of only guideline base knowledge acquisition. Following are fundamental concepts which constitute the whole process as an unique compared to other approaches.

Clinical Knowledge Model:CKM

A CKM is the baseline model for validation of ultimate clinical rules in a knowledge base. It represents the clinical knowledge for a specific clinical domain, while referencing standard knowledge resources, such as CPGs. CPGs are widely used consulting knowledge resources that are applied in clinical practices for diagnoses and treatments.

Prediction Model: PM

A PM is the decision tree model derived from real patient data using decision tree machine learning algorithms. The criteria for PM is actually based on its understandability and comprehensibility of the knowledge. The PM is main input which is ultimately evolved to refined knowledge.

Refined Clinical Knowledge Model: R-CKM

A R-CKM is refined knowledge and ultimate outcome which is converted into shareable and executable format. The evolution of R-CKM is based on rigorous validation process which consumes PM and CKM and ensures final valid clinical knowledge model for CDSS.

Formally verified R-CKM validation

The formal verification is introduced to prove the consistency of the R-CKM validation process in order to make sure the final R-CKM is consistent after passing through verified validation process. Formal modeling of the PM, CKM, R-CKM and R-CKM validation process is performed and finally verified through well-established mathematical theorems.

Development framework for validation and formal verification processes

Unfortunately, most of the state of the art knowledge acquisition, validation and verification methods are introduced in silos and they do not consider its applicability in integrated development framework. In order to overcome this limitation, the knowledge validation and verification process is aligned and unified with development framework - so called SmartCDSS-DF. SmartCDSS-DF is development framework which incorporates the guideline-enabled acquisition, validation, and verification process into a unified process model. The framework support isolation of concerns by providing unified viewpoints of the development process - so that different stakeholders with diverse capability can easily understand the requirements for development of CDSS.

1.3 Terminologies clarifications

We are applying decision tree machine learning approach to create the model from patient data. There are different views in literature on defining the model as ”classification model” or ”prediction model”. For example, one definition, classification models predict categorical class labels; and prediction models predicts the continuous valued function. Similarly, these terms are used interchangeably for same model; the model which is used for existing data is referred as a classification model.While using new case and predict a class label, then the model is referred as a prediction model. In this research work, we are not clearly differentiating the two models and using ”prediction model” for the decision tree model created from patient data.

Moreover, we are using decision tree formalism to represent the CKM (from guidelines) and the R-CKM. It is important to know at this point, knowledge representation is capturing the set of clinical conditions in sequence and corresponding outcome. So CKM and R-CKM decision trees

are considered as action graphs which are different from decision trees of machine learning in the context that they are based on wide range of concepts related to statistical theories.

1.4 Thesis Organization

This dissertation is organized into chapters as following.

- **Chapter 1: Introduction**. Chapter 1 provides brief introduction of the research work on knowledge acquisition and CDSS requirements in general. It highlights the motivation for guideline enabled knowledge acquisition and introduces the methodology for guideline based knowledge acquisition.

- **Chapter 2: Related Work**. A background detail is provided in this chapter about the knowledge acquisition, knowledge validation and verifications. Furthermore, the architectural approaches of CDSS are introduced and evaluated for the proposed guideline enabled methodology.

- **Chapter 3: Three Phase Knowledge Acquisition and Validation Method**. This chapter provides detailed description of the Three-phase knowledge acquisition and validation process. Detailed methodology related to each phase is explored with corresponding outcomes - PM, CKM, R-CKM, and validation process. Finally, the chapter briefly highlighted the transformation of R-CKM into executable rules which are shareable MLMs and provide baseline for evaluation of the R-CKM model.

- **Chapter 4: Formal Verification of Three Phase Models**. This chapter describes formal verification of the knowledge acquisition and validation process proposed in three phase process model. The formal process model using Z notation is presented, to represent knowledge models (PM, CKM, and R-CKM) and validation process. The validation process is formally verified in context of mathematical theorem proving mechanism.

- **Chapter 5: Implementation of Three-phase model and Formal Verification using CDSS development framework**. In this chapter, we will demonstrate how to incorporate and align methods of validation and formal verification in real development environment of

CDSS. This chapter will focus on the SmartCDSS-DF - a development framework aligned with most of the existing development frameworks and specialized for CDSS development. We will introduce the method plugin mechanism to provide support to our proposed knowledge acquisition, validation, and verification methods in the development of CDSS.

- **Chapter 6: Results and Evaluation**. In this chapter results of the three phase model and formal verification is presented. Finally, the proposed method is compared with existing knowledge acquisition methods.

- **Chapter 7: Conclusion and Future Directions**. This chapter concludes the thesis and also provides future directions in this research area. The main contribution of the thesis is also highlighted in this chapter.

Chapter 2

Related Work

Among the key requirements of CDSS, knowledge validation and verification is main focus of this research work. Furthermore, incorporating these methods into unified development process of CDSS is additional contribution of this research work. Based on these contributions, the related work is categorized into two parts. First exploring the existing acquisition and validation methods and describing the key differences of the proposed method. Second, analysis of existing architectural approaches to support our propose knowledge acquisition and validation method and deriving motivation for the development of SmartCDSS-DF.

2.1 Knowledge acquisition and validation

Knowledge acquisition and validation are prerequisites for effective decision support services. Various approaches are used to target objectives in target system design. In a methodological review, Peleg [24] categorized the approaches of translating clinical practice guidelines (CPGs) into computer-interpretable guidelines (CIGs): cognitive methods, a collaborative modeling methodology and tools, an information extraction methodology, and specialized CIG authoring tools.

Cognitive methods examine how people mentally represent information to solve subsequent problems. CPG translation to CIGs from narrative guidelines discussed in [28] is a cognitive method in which domain experts are provided with predefined algorithmic steps to develop the guidelines. Information extraction methods use semi-automatic translation by extracting knowledge using templates from narrative guidelines. Serban et al. [31] extracted templates from narrative guidelines and used them as building blocks for guidelines based on background thesaurus knowledge. CIG authoring tools are used to directly transform CPGs into executable CIGs. CIGs follow standard formats, such as XML, RDF, and any other standard format. Examples of such

knowledge acquisition include use of an Arden Syntax editor to explicitly transform the clinical knowledge into an executable module, an approach mentioned in [32].

The systematic review [24] covers the methods that are based on translating narrative CPGs into CIGs; however, alternative approaches that use clinical datasets for knowledge extraction also exist. For example, Perera et al. [27] used a semantic-driven knowledge acquisition approach to establish missing relationships in data. Their proposed approach is limited to determining missing relationships in data whereby the initial knowledge base is constructed from UMLS vocabulary. Similarly, Gomoi et al. [26] used data mining techniques to generate rules from data and transform them into MLMs. This approach lacks a criterion definition for selection of candidate MLMs and final validation from reference guidelines. Our present study is closely related to [26, 27] in terms of our use of clinical data as a common source of knowledge acquisition. In addition, we employ a cognitive method to align the refined model in accordance with published NCCN guidelines.

The study in [1] resembles our study with respect to the objectives of using a data-driven approach to create rules. The authors derive rules from patient data and incorporate them into guidelines for completeness of missing recommendations. The methodology proposed in the study comprises a rigorous inspection of guidelines to find missing recommendations for all possible patient symptoms. A decision tree learning algorithm is used to generate the rules, aligned with the guideline tree for missing recommendations. The proposed methodology is robust for the clinical domain with less complexity. For complex clinical domains with comprehensive guidelines, it may not be feasible to identify the missing decisions. Furthermore, as complexity of the clinical domain increases, the number of patient conditions is also increased to verify the final recommendation. With an increased number of patient conditions, it becomes difficult to derive related profiles (as used in this study) for which a set of rules needs to be generated. Finally, the study defines no formal validation criteria for the generated rules, and depends on the quality and quantity of data. In contrast, we use a data driven approach to select rules from the PM that conform to the guidelines. The PM is evolved to R-CKM using a rigorous validation process.

Most existing validation approaches aim to enable developed CIGs to capture the requirements of corresponding CPGs. These approaches can be viewed in two broad categories of techniques: inspection and testing [24]. In the inspection technique, domain experts investigate the CIGs for

any possible errors in logic. Collaborative development of clinical guidelines is discussed in [33]. Teams of expert physicians and knowledge engineers create markups in final guidelines. These markups are evaluated together against the gold-standard markups with a thorough inspection to determine whether the final objective is achieved.

Testing techniques are used in addition to inspection techniques to minimize chances of errors in logic that may not be traced during inspection. These techniques use real patient cases to evaluate all possible decision paths in CIGs for obtaining correct decisions. Miller [34] used a domain-constraint-based approach with a clinical condition set while generating the minimal set of test cases required for evaluating the particular guidelines. We herein employ both approaches: inspection and testing with different perspectives. Domain experts keep the PM decision path as a candidate path in R-CKM after validation (inspection) from the CKM. Knowledge engineers provide the evidence of patient cases from the PM that correctly classify it into a correct decision (testing). This validation process allows the decision path to be included in R-CKM if it conforms to the validation criteria (which is based on the CKM), which is provided with optimum accuracy from the patient data. Finally, R-CKM is validated against real patient data while transforming it into executable MLMs and integrated into hospital information system (HIS) workflows.

PMs are considered primary sources that physicians can use in clinical setups for recommendations. Widespread computational methods and tools are available for data analysis and creating of PMs. These methods and tools require a systematic method of selecting an appropriate PM that best fits the clinical prediction problem. The authors in [35] provided a systematic review of the most commonly used methods and simple guidelines for using these methods in clinical medicine. In our work, we use decision trees (DTs) to create the PM. We demonstrate CHAID and a classification and regression tree (CRT/CART) in our PM creation. Details on using basic PM techniques are found in [35–38].

Various standards for sharing clinical knowledge are available. HL7 Arden Syntax is one of the open standards of procedural representation of medical knowledge. There are substantial studies are available which uses the HL7 Arden Syntax as ultimate knowledge representation scheme [39, 40]. Study in [39] mainly focusing on HL7 Arden Syntax to represent the clinical knowledge related to selection and eligibility of patients for clinical trials in breast cancer. Similarly, the

study in [40] exploited the modular and expressive power of HL7 Arden Syntax to design and develop multi-patient surveillance dashboard. The knowledge is represented in a modular logic unit–the so-called MLM–which is sharable across an organization [41]. HL7 Arden Syntax is used to specify a knowledge representation that is sharable, with the contents being readable by both humans and machines [2]. In this work, R-CKM, which was created from a PM, is represented in MLMs and evaluated in term of MLM performance on real patient medical data. We use HL7 Arden Syntax V2.7. However, readers can access detailed specifications with the current version (2.10) of HL7 Arden Syntax from the HL7 Arden Syntax working group repository [42].

Nevertheless, HL7 Arden Syntax has limitations in representing the 'curly braces' problem, the standard model used in logic. HL7 vMR is a standard data model proposed to resolve the curly braces problem that is associated with integration of MLMs with healthcare workflows [21, 43]. The main intention of vMR is to create a simplified representation of a sufficient amount of clinical records for capturing information relevant to clinical knowledge and enabling understanding in knowledge engineers [44]. Most importantly, vMR is influenced by the HL7 V3 reference information model (RIM), which makes it easier to integrate it with HL7-compliant healthcare systems. The HL7 vMR model was developed based on analysis of requirements from 22 US institutions [43]. We create a cancer clinical domain model derived from HL7 vMR and used as a data model for creating MLMs. To this end, we employ HL7 vMR version 1; however, readers may consult the recent version 2 of HL7 vMR for detailed specifications [44].

2.2 Architectural approaches

2.2.1 Descriptions of architectural approaches

Architectural approaches are interpreted in different ways depending on their context of usage. Accordingly, architectures can take meanings of [45] i) the formal description of the system components that leads towards their implementation; ii) the structural view of the components with the associated interrelationships, principles, and guidelines that govern their design and evolution over time; and iii) the organizational structure of the system or component. Consequently, the architecture has a diverse meaning, which ultimately leads towards different evaluations for differ-

ent architectural approaches. To the best of our knowledge, current CDSS architectural evaluation approach [2, 46] target architectures fall under contextual meanings i) or iii).

Most of the existing initiatives in CDSS target the standardization of clinical knowledge content (e.g., HL7 Arden Syntax [42]), provide methods for knowledge acquisition and representation (e.g., GEM: Guideline Element Model [47]), or concentrate on integrated decision support (e.g., SEBASTIAN [48] and HSSP CDS [49]). These initiatives have established a strong base for CDSS in corresponding areas. Unfortunately, these initiatives do not consider the overall governing processes, which provide baseline guidelines between the stages of knowledge acquisition and design in order to develop a sustainable system. In this work, we present an architectural approach for CDSS in context of ii); we attempt to provide a structural and uniform view of CDSS components (aligned with the capabilities of stakeholders) and define the guidelines and processes that govern the systemic evolution of the system design, which will ultimately lead to implementation.

The context of the architectural approach for CDSS in the current work is aligned with IEEE standard 1471: ”Recommended Practice for Architectural Description of Software-Intensive Systems-Description” [50]. This standard is based on recommended industrial practices and attempts to provide a conceptual framework for the architectural description of software-intensive systems. Moreover, Lopez and Blobel [20] presented architectural approaches in five different views: architectural frameworks, architectural models, description languages, middleware architectures, and architecture development processes. Figure 2.1 extends the example in [20] and includes reference architectures for the CDSS domain. The intended objectives of the current architectural approach for CDSS are well-aligned with the ”architecture development process”; as such, we will briefly explain relevant approaches in the same categories from the ”software engineering” and ”health informatics” domains. We will also explain the motivation for introducing the CDSS development framework.

2.2.2 Rational Unified Process (RUP)

The rational Unified Process (RUP) is a Unified Process (UP) that was developed by Jacobson et al. [51]. The UP is a generic process framework that can be specialized to large-scale software systems as well as to different application areas, different types of organizations, different competent

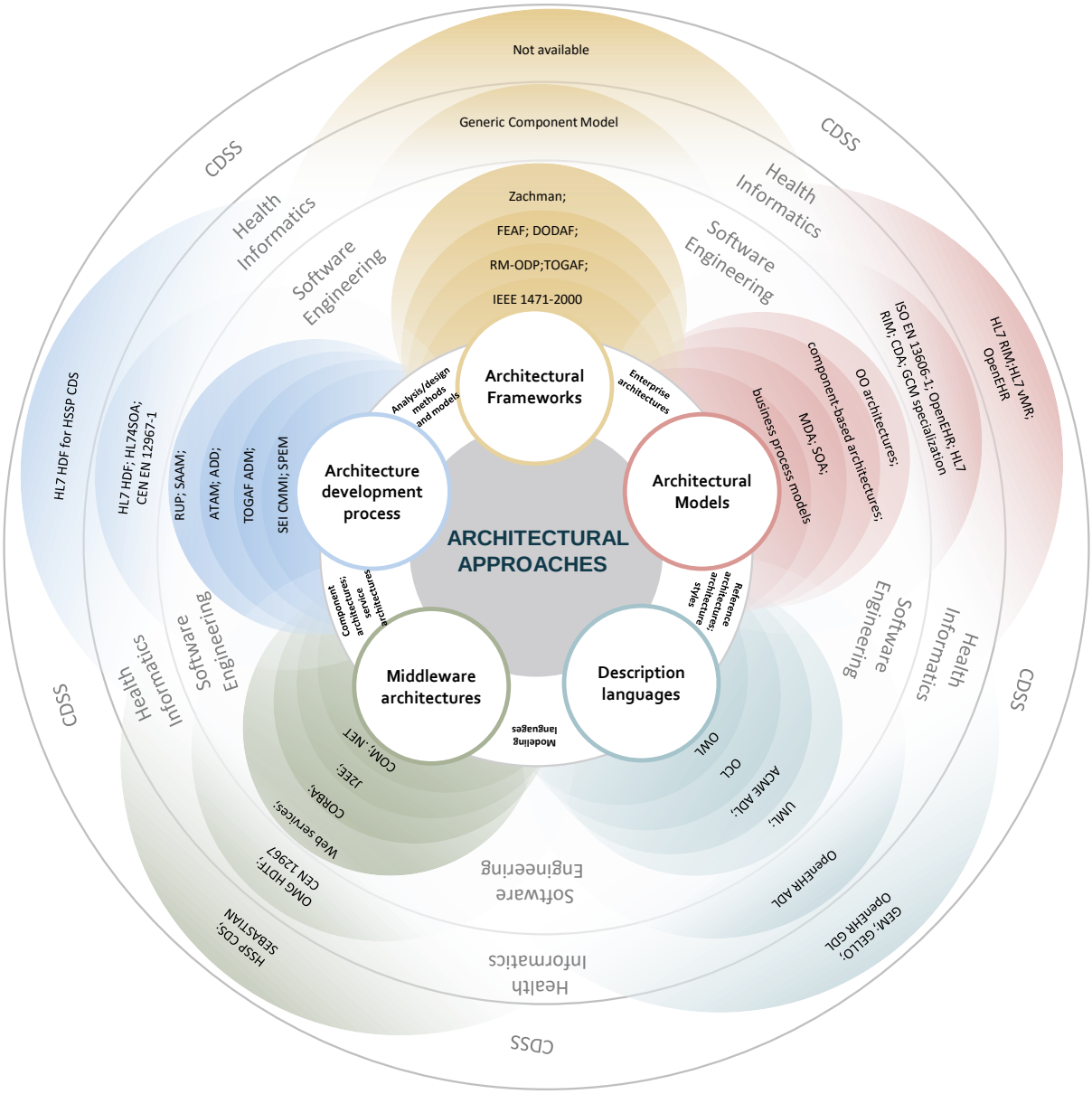

Figure 2.1: Architectural Approaches

levels, and different project sizes [16].

RUP exploits the key aspects of UP; it is case-driven, architecture-centric, employs iterative and incremental development, and introduces best practices that ensure the production of high-quality software to meet the needs of the end-users. RUP also has a predictable schedule and budget [52].

Initially, the RUP was developed and maintained as a process product by Rational Software. Rational Software comes with a set of tools, which provides all of the necessary artifacts required for RUP. For example, it includes all of the relevant models associated with RUP disciplines and the configuration process for the development of software that is suitable for the organizational requirements. In 2005, IBM acquired the RUP and the software toolset has evolved into a comprehensive software platform known as Rational Method Composer (RMC) [53]. Moreover, IBM has also released two components (i.e., the meta-model and core extensible process tooling frameworks) of the RUP as an open source Eclipse platform (Eclipse Process Framework (EPF)) [54], which intends to provide fundamental capabilities such as method authoring, process authoring, library management, and configuration/publishing of the RUP concepts and delivery processes. RUP processes are aligned in two dimensions, which cover distinct aspects of the system under development [52]:

- Dynamic aspects of the process enable timeline-based progress of the system development. Timeline-based progress is expressed in terms of cycles, phases, iterations, and milestones. The software development is broken into sets of cycles, which internally use four consecutive phases: *inception, elaboration, construction, and transition*. Each phase in the development cycle has a specific purpose and is concluded with a well-defined milestone. The RUP phase can be further broken down into iterations. Each iteration represents a complete development loop that concludes with a concrete outcome. This iteration strategy has many advantages compared to traditional software processes including mitigation of risks at the early stages, control changes, increased reusability, and enhanced overall quality
- Static aspects of the process enable detailed activities that are involved at different levels of system development. They express processes in terms of the set of activities, associated artifacts, responsible workers (roles), and target workflow. The static aspects describe *who* is doing *what*, *how*, and *when*. These aspects are expressed in RUP using four primary modeling elements: workers, activities, artifacts, and workflows. A worker defines the behavior and responsibility of an individual or group of individuals working as a team. An activity represents the unit of work that is intended to be performed by the worker. An artifact is a piece of information provided by an activity or set of activities; this could be

newly created or represent a modification to existing information. A workflow is a sequence of activities intended to produce an observable outcome.

In order to present sets of workflows in a more manageable way, to unify the core requirements for traceability, RUP divides the workflows into nine core workflows: six as core ”engineering” workflows and three as core ”supporting” workflows. Core engineering workflows are associated with the system development, while core supporting workflows deal with the management of the overall system development. Core engineering workflows include i) business modelling, ii) requirements, iii) analysis & design, iv) implementation, v) testing, and vi) deployment. Alternatively, core supporting workflows include i) project management, ii) configuration & change management, and iii) the environment.

In addition to the core aspects, RUP also has a provision for creating new processes and content that best suits the organizational needs to fulfil the specific system requirements. This mechanism allows a configurable framework where custom processes can be defined and reused at different levels of the software delivery process. The RUP mechanism is formally known as RUP tailoring, where RUP processes can be tailored with different levels of customization [15].

2.2.3 HL7 Development Framework(HDF)

The HIS development framework (HIS-DF) focuses on the development processes and components of the healthcare information system, which supports semantic interoperability. HIS-DF is an extension of the Generic Component Model (GCM) and uses RUP as a baseline methodology to support configurable content integration with the framework [20].

The HIS-DF framework provides extensions to the GCM component by constraining the core steps used in GCM. The following is a brief description of the core steps that reflect the philosophy of the HIS-DF:

- HIS-DF restricts the first step of GCM, which is the analysis of ”health information systems”.
- In the second step, dimensions of the domain related to health information are separated from the other domains and considered independently; these are subsequently aggregated.

This separation is intended to reduce the complexity of the domains involved in the health systems.

- In the third step, HIS-DF addresses the complexity of HIS by analyzing the granular level of GCM concepts (i.e., basic concepts, functionality and services, relation networks, and business concepts). Various reference models, such as HL7 RIM, HL7 CDA, openEHR archetypes, and other HIS architecture standards (CEN EN 12967, OMG HDTF), are used to define the domain knowledge for semantic integration.
- In the fourth step of HIS-DF, the system architecture is defined by restricting the analysis to the platform-independent aspects of the system. The logical aspects of the system, influenced by the ISO/IEC 10746-2 RM-ODP Enterprise viewpoint, Information viewpoint, and Computational viewpoint, are also considered.
- In the last step (step 5), HIS-DF takes advantages of RUP by extending its standard concepts while incorporating all of the previous steps into the process view to manage the overall development of interoperable HIS.

HIS-DF provides small healthcare IT projects (with restricted resources) with the opportunity to develop an interoperable HIS by adopting widely-used HL7 standards and advanced architectural approaches.

2.2.4 HIS Development Framework (HIS-DF)

HIS development framework (HIS-DF) is focusing on the development processes and components of healthcare information system which support semantic interoperability. HIS-DF is an extension of Generic Component Model (GCM) and using RUP as a baseline methodology to support configurable content integration to the framework [20].

HIS-DF framework provides extensions to GCM component in terms of constraining the core steps used in GCM. The following is a brief description of core steps that reflect the philosophy of the HIS-DF:

- HIS-DF restrict the first step of GCM of analysis to ”health information systems”.

- In the second step, dimensions of the domain related to health information are separated from other domains and consider independent analysis which is ultimately aggregated at the end. The separation is intended to reduce the complexity of overall domains involved in health systems.
- In the third step, HIS-DF address the complexity of HIS by analyzing the granular level of GCM concepts; basic concepts, functionality and services, relations networks and business concepts. Using various reference models such as HL7 RIM, HL7 CDA, openEHR archetypes and other HIS architecture standards (CEN EN 12967, OMG HDTF) to define domain knowledge for semantic integration.
- In the fourth step of HIS-DF, system architecture is defined by restricting the analysis to platform-independent aspects of the system and consider the logical aspects of the system influenced from ISO/IEC 10746-2 RM-ODP Enterprise viewpoint, Information viewpoint and Computational viewpoint.
- In the last step (step 5), HIS-DF take advantages of RUP by extending its standard concepts while incorporating all previous steps in process view to manage the overall development of interoperable HIS.

HIS-DF provides opportunity to small-size healthcare IT projects with restricted resources to develop an interoperable HIS by adopting widely used HL7 standards and advanced architectural approaches.

2.2.5 NEHTA Interoperability Framework (NEHTA-IF)

NEHTA-IF is based on ISO/IEC RM-ODP [55–58] and provides a set of interoperability concepts, patterns, and structuring rules to support the co-existence and instantiation of different solution frameworks. The solution frameworks are the candidate architectural choice for the organization used to represent the system in an eHealth ecosystem [19]. NEHTA-IF specifications are comprised of three core components:

- interoperability language specification, which is comprised of a set of different modelling languages that are used to form architectural description languages that are appropriate for

different stakeholders.

- Solution framework, which provides the guidelines for adding further architectural expressiveness to the interoperability framework in order to align the design and the modelling of the system architecture with the organizational choices.
- Set of solution specifications, which are developed as a result of any candidate solution framework.

NEHTA-IF extensively uses RM-ODP viewpoints to align the interoperability requirements of the system at different levels of concerns. Current specifications of NEHTA-IF provide specification languages at three levels (organizational, informational, and technical) to reflect the needs of different group of stakeholders.

Organizational interoperability exploits the concepts of ODP Enterprise viewpoints to model the organizational context of the interoperability that is required to deliver services in an eHealth system. Information interoperability deals with most of the existing healthcare standards from HL7 and CEN EHR to provide a canonical model that represents subsets of concepts from the ODP Information viewpoint. Technical interoperability uses several concepts from the ODP Computational viewpoint to represent behavioral aspects related to the interoperability of the system under consideration.

2.2.6 Motivation for CDSS development framework

The development of CDSS is different from conventional software systems in terms of the stakeholders, who have diverse capabilities and specialized requirements. In the same way, it also differs from the development of HIS; the requirements of CDSS are related to knowledge whereas HIS deals with healthcare workflows and associated information. Some of the HIS requirements are common to CDSS, such as integration of CDSS with HIS, which can be aligned with solutions that are commonly used for HIS integration. Therefore, the architectural approaches discussed in this section have the following limitations, which make it difficult to completely fulfil the requirements of the CDSS:

- CDSS requires a clearly differentiating the concerns of different stakeholders (who have

diverse capabilities) to properly align the requirements. For example, domain experts should be provided with processes that enable mechanisms for acquiring knowledge from diverse sources and representing this knowledge in a shareable format. Engineers and developers should be provided with processes that allow them to aid domain experts and develop the toolsets necessary for knowledge acquisition.

Using RUP with core processes, it is possible to develop knowledge acquisition toolsets in an iterative manner. However, the core requirements of domain experts for knowledge acquisition are not completely fulfilled.

HDF, NEHTA-IF, and HIS-DF support knowledge integration requirements in terms of distinctly separating concerns with respect to interoperability. The core processes in both frameworks focus on the interoperability aspects of HIS and the partial requirements of CDSS are fulfilled.

- The core competency of CDSS is to provide accurate knowledge that produces recommendations that domain experts trust. This requirement directly dictates proper validation of clinical knowledge and verification of the adopted knowledge acquisition and validation methods.

  The RUP processes provide specifications to generally align the requirements of the system. However, the existing processes provide no guidance for the specialized requirements of the system such as formal verifications of the knowledge validation process.

  NEHTA-IF and HIS-DF also fail to provide guidance for formal verifications; instead, they rely on the viewpoints of RM-ODP to achieve consistency in the requirements from an interoperability perspective. HDF provides consistency of the requirements, from analysis to design, while also ensuring that the requirements conform to the standard reference model.

Nevertheless, the CDSS development framework can leverage the fundamental concepts of existing frameworks and exploit some of their components for the specialized requirements of the CDSS. Some examples of this include:

- Exploiting the iterative process model of RUP to support evolutionary CDSS development.

- Leveraging RM-ODP viewpoints by adopting strategies from NEHTA-IF and HIS-DF to separate concerns for different stakeholders of the CDSS.
- Using the tailoring mechanism of the RUP to define processes for the formal verification of the knowledge acquisition and validation processes.

Chapter 3

Three Phase Knowledge Acquisition and Validation Method

3.1 Motivations for Three phase knowledge acquisition and validation process

Clinical knowledge acquisition is the main activity in achieving successful deployment of CDSSs. Unlike conventional requirements gathering and modeling of a system, knowledge creation needs a detailed set of activities that cover the actual scenarios and facts occurring in a real environment. Most importantly, the domain experts are not required to know the executable knowledge paradigm used as an integral part in a real healthcare workflow. Moreover, the ultimate goal of knowledge acquisition is to represent the knowledge that functions with an existing healthcare workflow and to enable a proper validation process for the final knowledge model. In this regard, we adapt a data-driven knowledge acquisition and validation approach that reflects a real clinical setup deriving the clinical knowledge from existing clinical practices. The proposed approach is an iterative model that includes three phases with ten activities. It enables a domain expert to create an executable knowledge base with coordination of a knowledge engineer. An abstract view of the phases with activity descriptions is depicted in Figure 3.1.

3.2 Phase-I: Clinical knowledge modelling

A CKM is the baseline model for validation of ultimate clinical rules in a knowledge base. It represents the clinical knowledge for a specific clinical domain, while referencing standard knowledge resources, such as CPGs. CPGs are widely used consulting knowledge resources that are applied

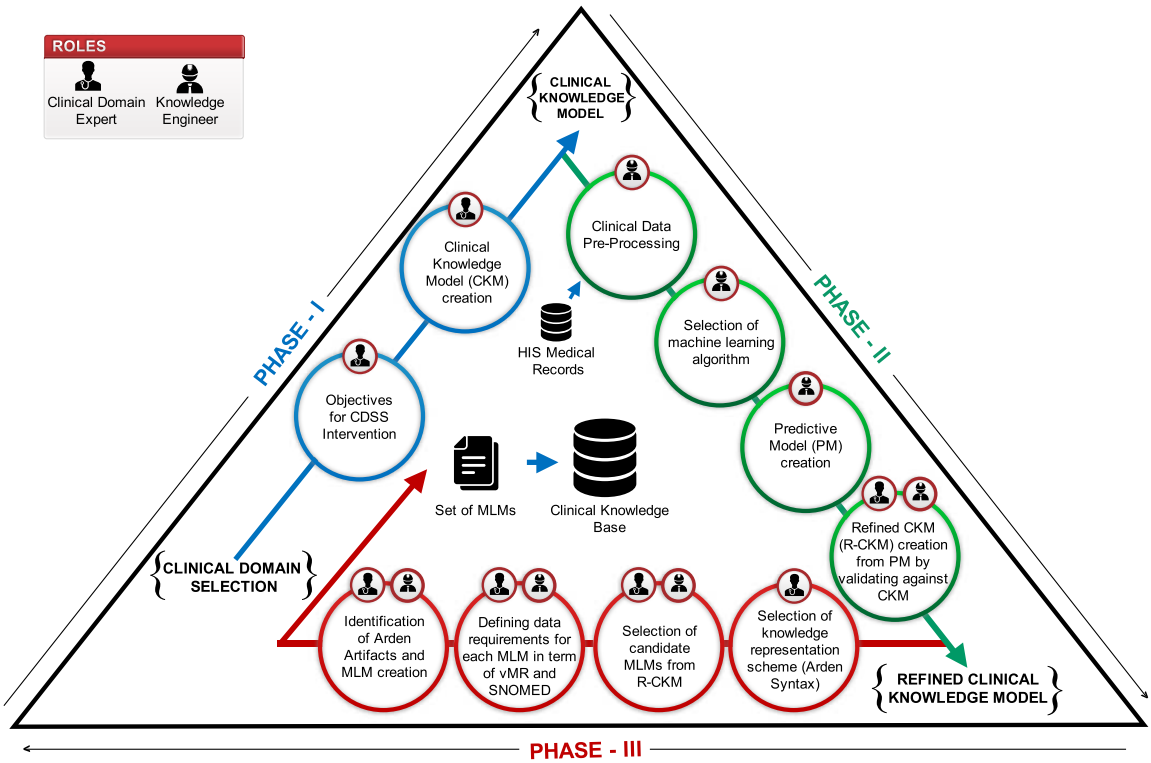

Figure 3.1: Data-driven clinical knowledge acquisition and validation process model

in clinical practices for diagnoses and treatments. CPGs are available in different formats, such as textual narratives and/or decision trees. A CKM is a formal decision tree representation of CPGs aligned with clinical objectives in a particular domain. In Phase I, a team of physicians is involved to finalize the CKM and perform the following activities.

Defining a set of clinical objectives: Clinical objectives specify the scope and intended outcomes of knowledge acquisition. In the context of a CDSS, clinical objectives are defined to specify possible CDSS interventions.

- We define the clinical objective for creating clinical knowledge covering a treatment plan recommendation for head and neck cancer. The treatment plan includes a single procedure or combination of procedures from radiotherapy, chemotherapy, induction chemotherapy, and surgery. The clinical knowledge helps physicians during treatment of a patient or gives recommendations of a treatment plan during a multidisciplinary

conference [1] of physicians.

- The recommendations are classified based on a particular tumor site. In this research work, the oral cavity is considered a primary site for an initial CDSS recommendation.

Selection of CPGs: Physicians consult CPGs for diagnoses and treatment plans during medical practices. In this step, physicians intend to select appropriate CPGs that align with the clinical objectives and use an appropriate knowledge representation scheme. In current work, SKMCH physicians use the following guidelines:

- NCCN guidelines are used as candidate CPGs to model clinical knowledge for tumors of the oral cavity and other sites.
- Tumor, Node, Metastases (TNM) staging guidelines are used to represent the clinical staging of tumors.
- A decision tree is selected as a formal representation of knowledge. Decision nodes are represented as rectangles and diamonds, while the conclusion nodes are represented with oval corner rectangles. Oval corner rectangles also play the role of condition nodes in case it comes in the middle of the tree.

Creation of CKM: A team of physicians converts the selected guidelines into a formal decision tree representation. For oral cavity CKM development, the following steps are used:

- Two resident doctors are assigned to initially create the draft decision tree from NCCN guideline trees and narratives.
- The initial draft of the decision tree is thoroughly inspected by a senior oncologist and approved as a final CKM with a possible amendment if needed.
- In the final CKM decision tree, a senior oncologist may incorporate some proven practices that may not be included in CPGs but bear evidence of its validity from other knowledge.

[1] A multidisciplinary conference comprises a panel of doctors including oncologists, radiologists, surgeons and other resident physicians. They conduct a conference on regular basis to select the final treatment plan for a patient.

3.3 Phase-II:Knowledge acquisition and validation

Knowledge acquisition and validation are a core aspect of this work. They are achieved through application of machine learning algorithms on existing medical records in HIS and applying a rigorous validation process supported by the CKM. Phase II comprises activities that are categorized into two broad perspectives: data-driven knowledge acquisition and knowledge validation.

3.3.1 Data-driven knowledge acquisition

Patient medical records in HIS are the primary resource for acquisition of clinical knowledge. Patient medical records reflect patient encounters and detail histories of diagnoses and ongoing or completed treatment. Various computational methods and tools are widely used for analyzing patient medical records and creating PMs for future recommendations. The authors in [35] discussed overall issues and provided state-of-the-art guidelines to use appropriate methods for PM creation in clinical medicine. In this study, patient data of head and neck cancers with tumors in the primary oral cavity site were imported into SPSS [38] http://www.sussex.ac.uk/its/pdfs/SPSS_Decision_Trees_21.pdf (Accessed: 24 December, 2014) and Weka [59] from HIS to create a PM using a decision tree (DT) classification method. The main advantage of DTs is the comprehensibility of the classification structure, whereby they can easily determine attributes for classifying and verifying new data [37]. In addition, owing to powerful heuristics, the computational complexity of the DT induction algorithm is low [35]. Finally, it provides the opportunity to generate readily comprehensible knowledge rules.

In the context of tasks and guidelines provided by [35] for the PM, the data-driven knowledge acquisition takes into account the following related activities. Figure 3.2 highlights the detailed tasks in each activity and possible sequences for performing these activities.

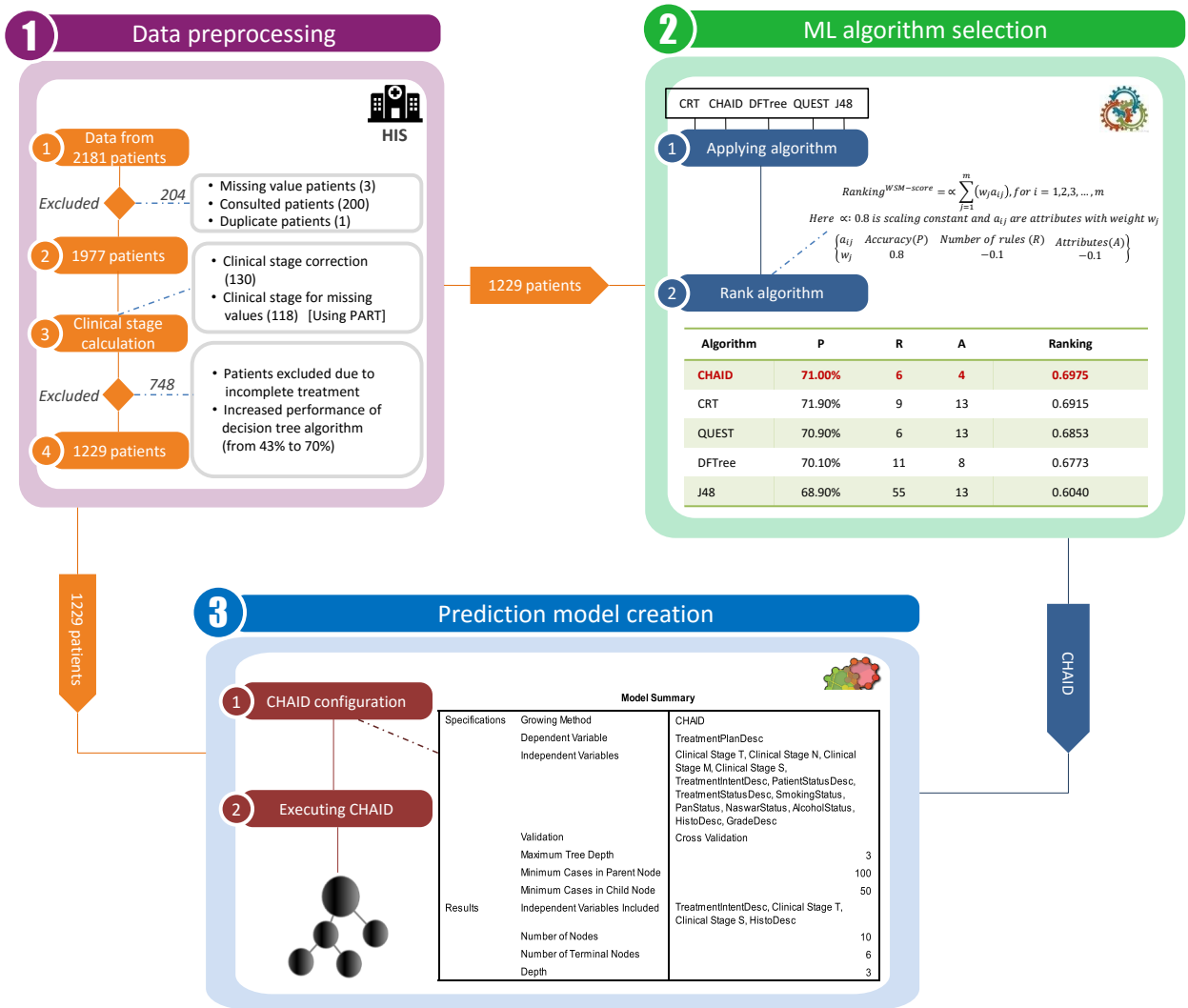

Figure 3.2: PM creation process

Clinical data description and pre-processing: The clinical data of oral cavity cancer patients for this study was imported from HIS with thirteen condition attributes and one decision attribute (treatment plan). Details of the condition attributes and decision attributes are shown in Table 1. For the final prediction model, 1,229 patient records were used after applying pre-processing (removing and calculating missing attributes) on 2,181 original patient records. Detailed steps of the pre-processing are shown in the ”Data preprocessing” activity in Figure 3.2.

Table 3.1: **Data description of oral cavity**

Data Summary: **Total Attributes: 14** **Decision Attribute: Treatment Plan Description**		
Attribute Type	**Attribute Title**	**Attribute Description**
Decision Attribute	Treatment Plan Description	Treatment plan for patient C: Induction Chemotherapy is done. CRT: Chemoradiotherapy is done. RT: Radiotherapy is done. S: Surgery is done. C CRT: C is done followed by CRT. C RT: C is done followed by RT. S CRT: S is done followed by CRT. S RT: S is done followed by RT. C S CRT: C is done followed by S and CRT. C S RT: C is done followed by S and RT. UK: Treatment unknown. NA: Treatment not applicable.
Condition Attributes	Grade Description	Indicate patient treatment status such as poor, moderate etc.
	Treatment Intent Description	Patient status for treatment, such as palliative or radical.
	Treatment Status Description	Indicate patient treatment status such as completed.
	Clinical Stage T	TNM Staging T value.
	Clinical Stage N	TNM Staging N value.
	Clinical Stage M	TNM Staging M value.
	Clinical Stage S	TNM Staging S value.
	Smoking	Smoking status.
	Alcohol	Alcohol status.
	Naswar	Naswar status. Naswar is a moist, powdered tobacco snuff.
	Pan	Pan status. Pan is type of, tobacco chewed and finally spat out or swallowed.
	Patient Status	Patient current status such alive, dead etc.
	Histology Description	Indicate patient disease such as Carcinoma.

Selection of machine learning algorithm: The main goal of this activity is to determine the appropriate best-performing decision tree algorithm on a given dataset for generating the final PM. PMs can be evaluated based on their predictive performance and comprehensibility. Predictive

performance can be quantified using classification accuracy, while comprehensibility is a subjective measure that is assessed by a domain expert. In our context, we combine both measures into quantitative measures to achieve our desired criteria. The criteria defined by the domain expert is the generating of a PM with high accuracy and providing a minimal set of decision paths by involving fewer dominant condition attributes. This criteria is translated into a quantitative measure using the weighted sum model (WSM). WSM ranking is expressed in Equation 3.1, which uses classification accuracy P, the number of rules generated R, and the number of attributes A involved in the conditions. Weights w_j are assigned based on the importance of the attributes. Classification accuracy P is the most important in the selection of algorithm, which is assigned w_p: +0.8. The number of rules R and number of attributes A are assigned $w_r : -0.1$ and $w_a : -0.1$, respectively. According to our criteria, an algorithm with a minimum rule set and involving fewer attributes is preferred. In this regard, we choose a negative scale to discourage algorithms that generate maximum rules and/or those involving more attributes.

According to our criteria, CHAID is the most suitable algorithm among CRT, QUEST, DFTree, and J48 for use with the PM and rules generation, as shown in the "ML algorithm selection" activity in Figure 3.2.

$$Ranking^{WSM-score} = \alpha \sum_{j=1}^{m} w_j a_{ij}, for\ i = 1, 2, 3, ..., m \tag{3.1}$$

Here $\alpha : (0.8)$ *is scaling constant and* a_{ij} *are attributes with weight* w_j

$$\begin{Bmatrix} a_{ij} & Accuracy(P) & NumberofRules(R) & Attributes(A) \\ w_j & 0.8 & -0.1 & -0.1 \end{Bmatrix}$$

Creation of a PM: CHAID is an appropriate candidate algorithm that provides a PM with reasonable classification accuracy on given data while generating easily understandable rules.

CHAID uses multiway splits to generate more than two nodes from a current node. It chooses the independent (predictor or condition attribute) variable with the strongest interaction with the dependent variable (decision attribute). It has the capability of merging the

Table 3.2: **Classification using the CHAID model for oral cavity treatment planning**

Observed	Predicted				
	C CRT	RT	S RT	Total Cases	Accuracy
C CRT	343	12	46	401	85.5%
RT	87	184	127	398	46.2%
S RT	81	3	346	430	80.5%
Total	511	199	519	1229	
Overall Percentage	**41.6%**	**16.2%**	**42.2%**		**71.0%**

category of each predictor if they are not significantly different with respect to the dependent variable. A detailed description of CHAID and other tree algorithms are given in [37, 38].

CHAID is applied on a 1,229-patient dataset using the 13 condition attributes mentioned in Table 3.1. The algorithm has default parameters of the SPSS tool, as shown in the prediction model creation activity in Figure 3.2.

A decision tree representation of the PM is shown in Figure 3.3. The confusion matrix shown in Table 3.2 presents the overall accuracy of the model. In summary, our final model achieved 71.0% accuracy for classification of final cancer treatment.

The PM of CHAID classified the designated treatment plan with the accuracy of C CRT (85.5%), S RT (80.5%) and RT (46.2%). The accuracy of RT was comparatively low, but the major proportion of cases (127;59.4%) of remaining cases (214) were classified as S RT. The S RT treatment plan covered radiotherapy (RT) following surgery as the main procedure; therefore, it compensated the lesser precision of the classification model to only the RT decision class. There are many reasons for the direct RT treatment rather than following standard treatment of S followed by RT. These include patient not willing for surgery or having some comorbidities associated with tumor site. The CHAID classification algorithm has the intrinsic property of selecting dominant attributes from a set of condition attributes, which provided a higher segmentation of data. It selected 4/13 attributes ”Treatment Intent Description”, ”Clinical Stage T”, ”Clinical Stage S” and ”Histology Description”. The model generated ten nodes overall, of which six nodes were terminal (leaf) nodes in the

tree.

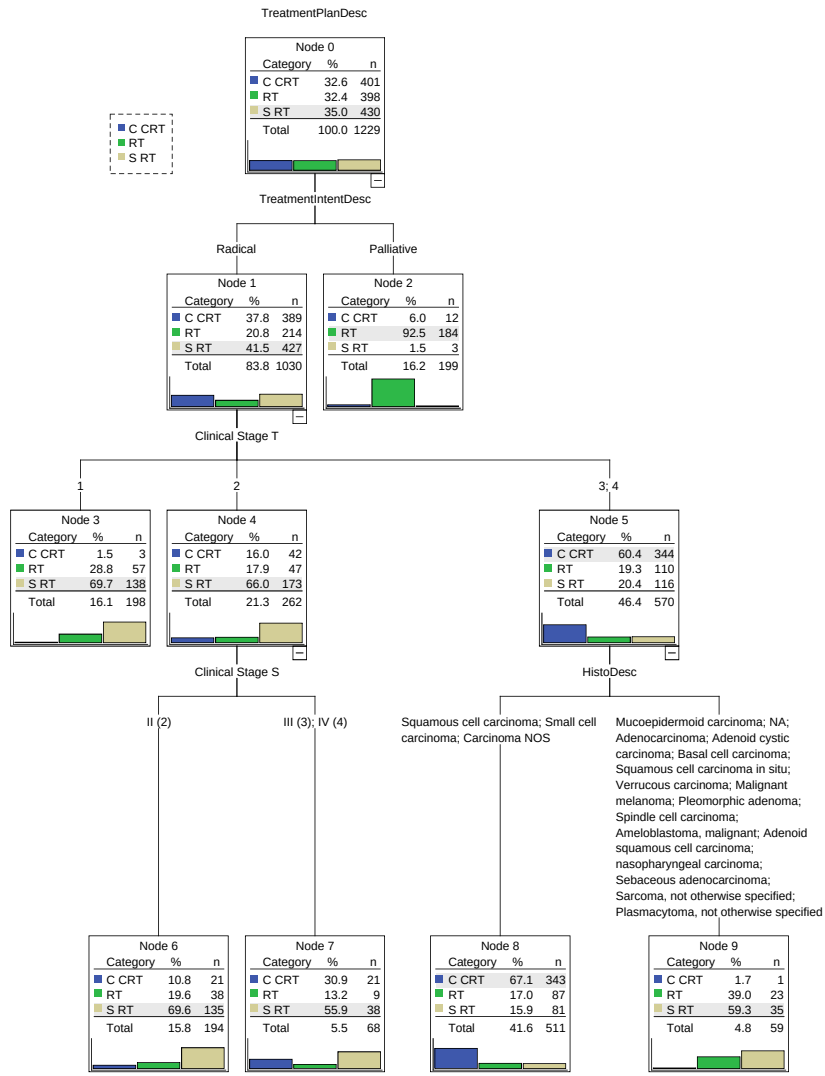

Figure 3.3: CHAID prediction model for oral cavity treatment plans

3.3.2 Knowledge validation

Knowledge validation is performed on the PM with conformance criteria that ensure that the final R-CKM model conforms to the CKM (standard CPGs). Domain experts and knowledge engineers worked in a collaborative environment to validate the PM against CKM using well-defined

inspection and testing mechanisms and developed the R-CKM conformed and validated model. Figure 3.4 depicts the process of validation with detailed flows of steps and the same process of validation is shown as three step process in Algorithm 3.3.1.

Algorithm 3.3.1: VALIDATIONPROCESSALGO(PM, CKM)

$Let\ rckm : R - CKM$

$Let\ conformedPaths : List < PMDecisionPath >$

$Let\ disgnatedAccuracy,\ refinements : List < Refinement >$

$Let\ CriteriaBank : Map < Criteria, priority >$

$/ * - - - [Step - 1] - - - * /$

$CriteriaBank = SetValidationCriteria(disgnatedAccuracy, CKM)$

$/ * - - - [Step - 2] - - - * /$

$conformedPaths = ConformancePMtoCKM(PM, CKM, CriteriaBank)$

for *dt in conformedPaths*

do $\begin{cases} / * - - - [Step - 3] - - - * / \\ rckm = EvolveRCKM(dt, refinements) \end{cases}$

return $(rckm)$

Activities performed in the validation process can be classified into three main categories: setting the validation criteria, validating the PM against the validation criteria, and finally evolving the R-CKM by inspecting and refining the PM.

Setting validation criteria: The validation criteria are a set of assertions that may be required to pass the decision paths in PM to be eligible for inclusion in R-CKM. Domain experts specify the validation criteria, which is influenced by their practices and conforms to the evidence from standard guidelines (CKM). While specifying the criteria, each criterion is assigned a priority and its primary status. The priority dictates the order of execution in the validation process; the primary status specifies that the given criteria must be satisfied by the decision path. Table 3.3 provides the set of criteria defined by domain experts to validate the oral

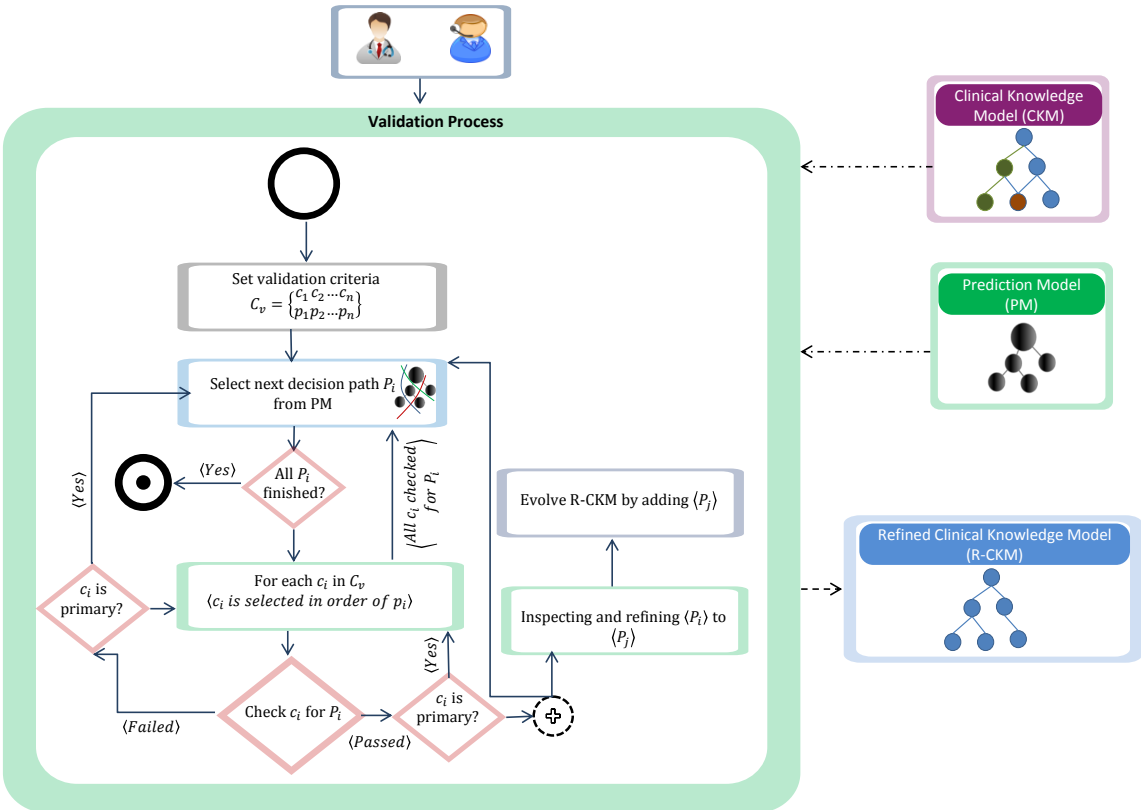

Figure 3.4: Knowledge validation process

cavity PM in this work. Algorithm 3.3.2 describes the internal detailed steps of the criteria setting process.

Table 3.3: **Validation criteria for oral cavity**

C.No	Criteria	Priority	Primary	Remarks
1.	$\{\forall P_i \in PM : Accuracy(P_i) > N\%\}$	1	Yes	• The domain expert assigns N, which represents the accuracy of PM based on the training data. • Trade off: Higher accuracy setting produces an efficient model, but coverage of involving more patient features is limited and vice versa.
2.	$\{\forall P_i \in PM \wedge \forall P_j \in CKM : !Conflict(P_i, P_j)\}$	1	Yes	• Conflicts with guidelines; conflicting treatments must not be exist. • Example: after surgery, chemo-induction has no meaning
3.	$\{\forall P_i \in PM \wedge \exists P_j \in CKM : Conform(P_i, P_j) \xrightarrow{yields} P_i \in \Delta RCKM\}$	2	No	• Decision path in PM conforming to any CKM path shall be part of R-CKM
4.	$\{\exists P_i \in PM \wedge \forall P_j \in CKM : !Conform(P_i, P_j) \xrightarrow{provides} Evidence(P_i) \xrightarrow{yields} P_i \in \Delta RCKM\}$	3	No	• Decision path in PM not conforming to any path in CKM can be part of R-CKM only if: • Sufficient evidence exists for effectiveness of the treatment • Evidence can be other standard clinical knowledge resources or local practices with a reasonable success ratio for the predicted treatment

Algorithm 3.3.2: SETVALIDATIONCRITERIA(*qualifiedAccuracy*, *CKM*)

Let CriteriaBank : *Map* $<$ *Criteria*, *priority* $>$

$/* --- [Criteria(1) : \ Accuracy] --- */$

Let accuracyCriteria : *Criteria*(*Primary*)

accruacyCriteria.addCItem(*qualifiedAccuracy*)

CriteriaBank.add(*accuracyCriteria*, 1)

$/* --- [Criteria(2,3) : \ Conflicts \ \& \ Conformance] --- */$

Let conflictsCriteria : *Criteria*(*Primary*)

Let conformanceCriteria : *Criteria*(*Non-Primary*)

for *dPath in CKM*

do
- *Let conclusionSequence* : *List* = *CreatConcSequence*(*dPath*)
- **while** *conc* : *conclusionSequence.hasNext*()
  - **do**
    - *conflictsCriteria.addConf*(*DefineConflict*(*conc*))
    - *conformanceCriteria.addConfr*(*DefineConformance*(*conc*))

CriteriaBank.add(*conflictsCriteria*, 1)

CriteriaBank.add(*conformanceCriteria*, 2)

$/* --- [Criteria(4) : \ Evidence] --- */$

Let evidenceCriteria : *Criteria*(*Non-Primary*)

evidenceCriteria.addEvid(*Local*, *EvidDesc*, *EvidFacts*)

evidenceCriteria.addEvid(*Online*, *EvidType*, *EvidDesc*, *EvidURL*)

CriteriaBank.add(*evidenceCriteria*)

return (*CriteriaBank*)

PM validation against criteria: Validation is an iterative process that selects one decision path P_i at a time from the PM for validation. The decision path is selected as part of R-CKM if it satisfies all primary validation criteria and passes at least one of the non-primary criteria.

For example, any decision path in the oral cavity PM becomes part of R-CKM if it satisfies criteria 1 and 2 and either 3 or 4, as mentioned in Table 6.3. The iteration is finished once all the decision paths in the PM are evaluated against the validation criteria. Algorithm 3.3.3 represents the process of each decision path and checks iteratively the conformance using Algorithm 3.3.4.

Algorithm 3.3.3: CONFORMANCEPMTOCKM(PM, CKM, $CriteriaBank$)

```
Let conformedDecPathPM : List < PMDecisionPath >
Let decPathPM : List < PMDecisionPath >= PM.getDecPath()
for dt in decPathPM
do {
    Let Conformed : Boolean = True
    while CriteriaBank.isNotEmpty()
    do {
        Let criteriaList = CriteriaBank.getNext()
        Let dPathCond = dt.getCondition()
        Let dPathConc = dt.getConclusion()
        Conformed = getConformance(criteriaList, CKM, dPathCond,
        dPathConc)
        if !Conformed & isPrimary(criteriaList) { break
          else if isNotPrimary(criteriaList) { break
    }
    if Conformed { conformedDecPathPM.add(dt)
}
return (conformedDecPathPM)
```

Inspection and refinement of selected PM decision path: The decision path P_i in the PM that passes the validation criteria is inspected and refined to P_j prior to becoming part of R-CKM. Inspection and refinement involves activities to identify conditional and decision values in a decision path for the same interpretation with existing CKM conditional and decision values. Therefore, in the refined decision path, the concepts are aligned according to the CKM. For example, the conditional value clinical stage S: 1 is interpreted as clinical stage T: 1

and clinical stage N: 0 in the CKM. Similarly, during refinement, the physician may add some other treatment to already given choices by providing evidence for the inclusion. The abstract representation of the inspection and refinement is depicted in Algorithm 3.3.5.

Algorithm 3.3.4: GETCONFORMANCE($criteriaList : List < Criteria >, CKM, dPathCond, dPathConc$)

$Let\ Conformed : Boolean(True/False)$

for $c\ in\ criteriaList$

do {
- $Let\ criteriaType = c.getCriteriaType()$
- **for** $cItem\ in\ c.criteriaItems$
  - **do** { $Conformed = checkConformance(decPathCond, decPathConc, CKM, criteriaType)$

return $(Conformed)$

Algorithm 3.3.5: EVOLVERCKM($P_i : PMDecisionPath, listRefinements : List < Refinement >$)

$Let\ rckm : R-CKM = \emptyset, P_j : RCKMDecPath$

$P_j = P_i$

for $ref\ in\ listRefinements$

do {
- $P_j.addrefinements(ref)$
- **if** $rckm = \emptyset$ { $rckm.addFirstPath(P_j)$
- **else** { $rckm.updatePath(P_j)$

return $(rckm)$

3.4 Phase-III:R-CKM transformation into executable rules

Most of the projects involving data mining and machine learning techniques in the clinical setup are stopped after the PM is created. This is unfortunate because the PM should be deployed in a

real clinical setup for clinical decision support as a daily base service. This would help in supporting the medical services, and stakeholders could easily monitor the efficiency of the technology in terms of improving quality of care and decreasing healthcare costs [35]. Maintaining the motivation to provide end-to-end implementation of the PM, R-CKM is represented in sharable format using HL7 Arden Syntax. R-CKM is converted into a set of candidate MLMs and implemented in a real setup for recommendation of a treatment plan for oral cavity cancer patients. Conversion of R-CKM into executable knowledge representation (MLM) is performed using the following activities.

Selection of candidate MLMs from the R-CKM: R-CKM can be transformed into different sets of MLMs depending on the domain expert intuitions and logical connections existing in the decision path of R-CKM. For R-CKM, three candidate approaches are analyzed for final executable knowledge. These are explained in Chapter 6 (Section 6.1.3).

Data requirements for MLMs using the HL7 vMR model: Data specifications for each MLM are important for the formal creation of logic. Data specifications include enlisting clinical data that is required for the MLM, representing clinical data in the standard data model (HL7 vMR), and mapping coded concepts into a standard vocabulary. Chapter 6 (Section 6.1.3) details the specifications of data requirements for candidate MLMs.

Identification of HL7 Arden Syntax artifacts and MLMs creation: Arden Syntax is a comprehensive specification supporting a large number of operators, various control structures, including decision and looping structures, and comprehensive models for various data types. Knowledge engineers summarize the basic artifacts required to transform the R-CKM into corresponding MLMs and provide physicians with training on using these artifacts.

Integration of MLMs with HIS workflow: Knowledge engineers implement the MLMs and integrate them with HIS workflows. Details of the implementation and integration of MLMs with HIS are available in our previous research [60, 61].

3.5 Summary

This chapter provides detailed description of the Three-phase knowledge acquisition and validation process. Detailed methodology related to each phase was described in terms of associated milestones - such as CKM, PM, and R-CKM. Finally, the chapter briefly highlighted the transformation of R-CKM into executable rules which are shareable MLMs and provide baseline for evaluation of the R-CKM model. After all, we have three phase knowledge acquisition and validation process, but natural question is raised: how can we prove that the acquisition and validation process is internally consistent and it will always guarantee valid R-CKM. Chapter 4 will provide answer in terms of: providing formal verification to support the proposed acquisition and validation method. Moreover, in Chapter 5, the processes and corresponding artifacts of the three-phase knowledge acquisition and validation process will be aligned and presented in implementation framework for CDSS.

Chapter 4

Formal Verification of Three Phase Models

4.1 Motivation of Formal Verification of Three phase models

The ability of domain experts to trust the knowledge content is a key factor that influences the success of CDSS implementation. This depends on how well the knowledge contents are passed through a sophisticated validation process to ensure consistencies in the refined knowledge model. According to a systematic review by Mor Peleg [24], formal verification techniques are used to validate the clinical knowledge for internal consistencies and to check for the fulfillment of the desired properties and specifications. These techniques are classified into two broad categories: model checking and theorem proving [24]. In model checking, the knowledge is transformed into an appropriate model-checker format and the model checker verifies the consistency of the knowledge model for the fulfilment of the desired properties. The model-checking approach was applied by Alessio Bottrighi et al. [62] to integrate the computerized guideline management system with a model checker. The guideline representation language GLARE is used and integrated with the SPIN model checker to verify the clinical guidelines. Theorem proving is based on the logical derivation of theorems in order to prove the consistency of the knowledge contents that are represented with the formal specifications. Annette T. Teije et al. [63] used KIV-based formalism to represent medical protocols and defined semantics of the desired properties. The desired properties for the protocol are verified using a formal proof of the KIV theorems.

Based on the substantial advantages and the need for formalism in knowledge validation and verification, the Formal methods Method Plugin covers one of the most important aspects of SmartCDSS-DF. Selection of an appropriate formal method requires formal guidelines to find the best fit for a knowledge representation scheme. In this work, we used Z notation as the formal representation language for knowledge representation and for modeling the validation method fea-

tures. We used the formal theorem proving mechanism to remove inconsistencies in the method, which ultimately ensures consistent knowledge. Selection of Z notation for SmartCDSS-DF is based on following guidelines.

1. Easy knowledge modeling: Using Z notation, it is simple to decompose the knowledge specifications into small pieces and formally define the static and dynamic aspects of the knowledge acquisition (i.e., the knowledge representation and validation method [64]). This aspect of Z notation is represented as the ”schema”, where the knowledge contents are defined as strongly typed-data and the constraints are represented as first order predicate logic. Moreover, the validation method is represented as a dynamic schema that operates within the boundaries of the knowledge representation schema. The subsequent section will introduce detailed contents of the ”Formal methods Method Plugin” in terms of Z specifications. Its application in SmartCDSS-DF validation is explained in detail in Section **??**.

2. Data-rich formalism: Another aspect of Z notation is the notion of ”types” [65]. Z types are mathematical data types that can be used to uniquely represent any object in a system. They specifically obey a rich collection of mathematical laws, which make it possible to determine the behavior of the system [64, 65]. This aspect of Z leverage, towards data-rich formalism of knowledge contents and the resulting artifacts, can be easily mapped to standard viewpoints of RM-ODP [66] (e.g., the information viewpoint). H. Bowman et al. used Z notation for consistency checking of two views in the information viewpoint [67]. In a similar way, Z notation can also be mapped to the ”analysis” and ”design” disciplines of the RUP framework.

3. Support of tools: The Z specification language not only enables formal specifications for a system and a language, but also allows for the systematic reduction of such specifications into implementations [68]. Moreover, there is a wide range of tools available to check for syntactic and type consistency in the specifications.

4.2 Using Z notations for formal verification of knowledge validation process

4.2.1 Overview of Z notation

Z specification contains a significant number of advantages that relate mathematical objects to features of the design (e.g., the system states, data structures, properties, and operations). There are several ways to represent objects in Z notation. Declaration, abbreviation, and axiomatic definitions are simple ways to represent objects in Z notation. ”Schema” and ”free” types are special ways to represent complex objects in Z notation. All of these types obey mathematical laws and have rules for reasoning with the information that they contain. At this point, introduction and use of these concepts is important; however, in this paper we skip the detailed description of the concepts used in Z notation. Readers may consult reference materials [64, 65] and other research works that have used Z extensively [68–70].

- *Declaration:* This is the simplest way to define an object. When an object is a set or some basic type, the name of an object is written in brackets. If there is more than one object, they can be separated with a comma. For example, type definition (1) in Figure 4.1 represents multiple object declarations. *ConditionAttribute* and *ConditionValue* are the set of concepts and the corresponding values, respectively, in the clinical knowledge model that construct the basic *Condition*.

- *Abbreviation:* Abbreviation introduces another name to an existing object. For example, type definition (2) in Figure 4.1 is the abbreviation for cancer treatments.

- *Free type:* Free type allows a variety of data structures to be represented using sets with explicit structuring information. For example, type definition (3) in Figures 4.1 highlights three different object definitions. *ConditionOperator* is a free type that distinctly represents the set of operators used in the *Condition*. The *Condition* further expresses the complex definition of the conditions used in the clinical rules. *treatmentSet* is a free type that covers high level semantics for cancer treatments that are provided to a patient in a proper sequence by using the guidelines.

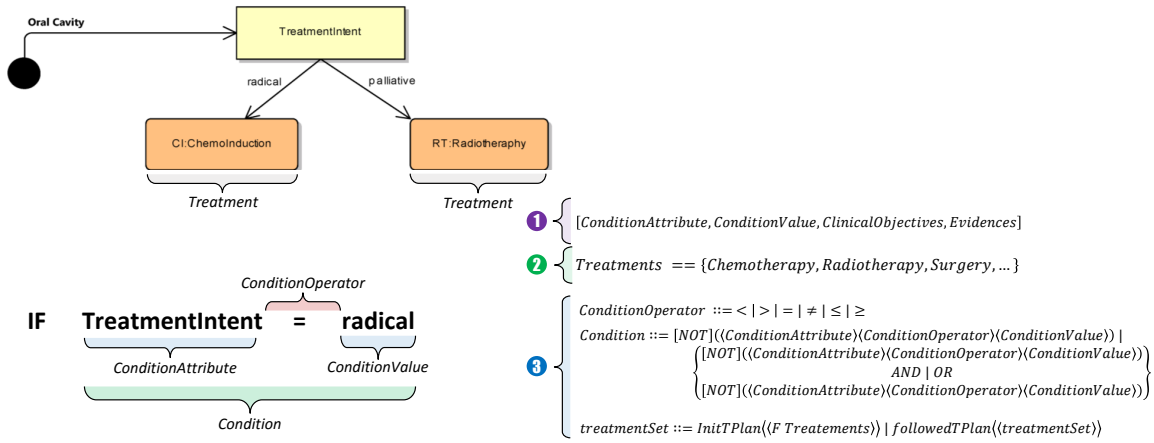

Figure 4.1: Declaration, Abbreviation and Free type example

- *Axiom:* Axiom provides the ability to define objects and includes constraints upon it. In an axiomatic definition, the object definition is represented in two compartments: declarations and predicates. Declarations represent the content structure of an object and predicates introduce constraints on the contents. An example axiomatic definition for CKM specification is shown in Figure 4.2.

- *Schema:* Schema is the most powerful artifact in Z notation and describes the system behavior. Similar to an axiom, it defines objects using declarations and predicates. However, the schema can take different forms such as a modeling static structure, modeling operations, and modeling different states of the object after operations. An example for modeling CKM is shown as the schema *"ClinicalKnowledgeModel"* in Figure 4.2.

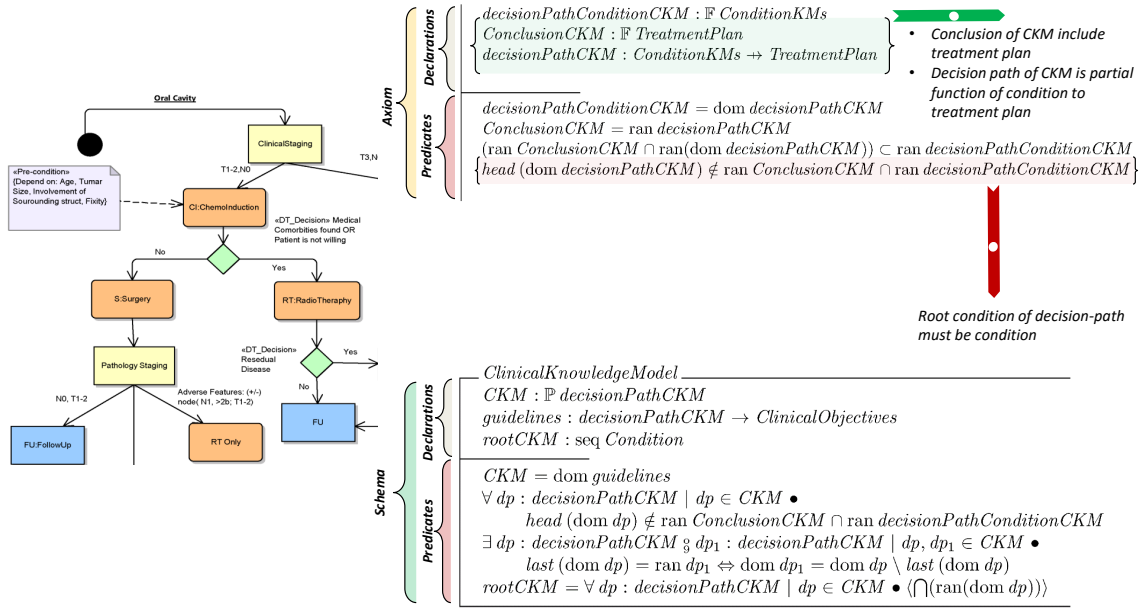

Figure 4.2: Axiomatic definition and Schema example

4.2.2 Formal modeling process

To the best of our knowledge, there is no substantial evidence that discusses formal modeling in discrete processes with proper guidance. Based on the capabilities of Z notation and the guidance available for applying different concepts of Z notation to formal modeling [64, 65], we formulate a formal modeling process for validation. This is comprised of four distinct processes: *"modeling problem"*, *"defining function and model states"*, *"proving consistency"*, and *"refine specification for concrete design"*. Below is a brief discussion of each of these processes. An abstract view is shown in Figure 4.3.

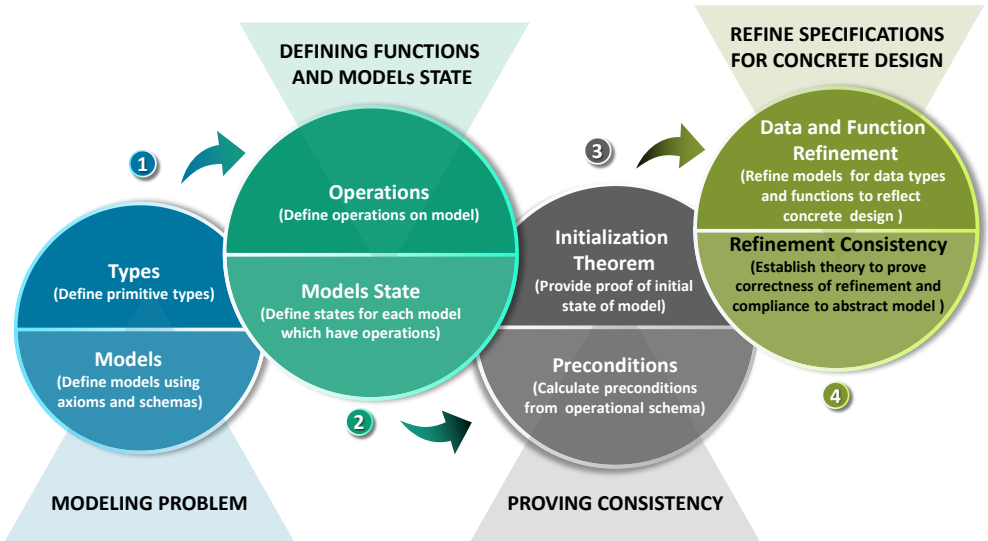

Figure 4.3: Formal modeling process

1. *Modeling problem:* This includes tasks used to analyze the problem context and identify all of the relevant concepts that contribute towards the final objectives. Different constructs of the selected formalism technique are used to model concepts at different granularity levels. Using Z specifications, primitive types, axioms, free types, and schema are candidate constructs that can be used to model the problem under consideration.

   For knowledge validation, different models involved in acquisition, such as PM, CKM, and R-CKM, will be modeled using different constructs of Z. The final outcomes of this process for knowledge validation are primitive types, free types, sets of axioms, and sets of static schema, which can be used to represent knowledge models.

2. *Defining functions and models state:* This includes tasks to define the behavioral aspects of the system under consideration. Defining operations related to the candidate models and associating the appropriate state model (as a consequence of the operation on the model) are the main activities of this process. Schemas are the main construct in Z and can represent the operations and states of the models. For knowledge validation, operations related to the retrieval of content from PM and CKM models are defined. These operations will have no effect on changing the state of the corresponding models. Different operations are defined

on the R-CKM model in order to validate the candidate decision path from PM against the CKM model and to evolve the final R-CKM model. As a result of the evolution of the R-CKM model, the corresponding state model is defined to formally represent possible changes in the contents of the R-CKM model.

3. *Proving consistency:* Identifying inconsistencies in the specifications of the modeled problem is the ultimate goal of formal methods. The main task is to make sure that the defined models are consistent and have no contradictions with their desired requirements. Moreover, it should be verified that the operations defined in various models are consistent and that their final outcomes are within the intended boundaries of the domain. Z specification provides a well-established way to achieve both of these goals. The first part is achieved, to prove the constraint part of the state schema of the model is satisfiable using *"initialization theorem"* to indicate that an initial state, at least, exists. The second part requires to investigate *"preconditions"* for the candidate operations - that may be calculated from operational schema using the one-point rule.

   For knowledge validation, the *"initialization theorem"* is required to prove the satisfiability of the R-CKM state schema. Moreover, *"preconditions"* are investigated for the operations that evolve the R-CKM model.

4. *Refining specification for concrete design:* The refinement process tends to construct and describe another model that complies with the original model of the design but is closer to implementation. The refinement process is comprised of extensive tasks that are applied in continuous iterations at the data and function levels to ensure that the specifications are free of any uncertainty. This is close to an executable program code. In order to prove that refinements are consistent within themselves and appropriately represent the original design model, it is necessary to establish a theory for refinement that includes a set of rules for proving the correctness.

In this research work, we exploit the first three processes to model the clinical knowledge (presented in Chapter 3) and formalize the validation process in order to prove that the validation framework is sufficiently consistent to produce valid knowledge during the knowledge acquisition

process. The refinement process is helpful for systems where the final outcomes of the design are required to be sufficiently close for direct conversion into executable code. This process is included because our knowledge specification can be easily converted into executable code if we properly exploit the Z refinement mechanism. Subsequent sections provide the detailed explanation and usage of the first three processes for SmartCDSS-DF and highlights the importance of formal validation by indicating inconsistencies that are left in the knowledge acquisition specifications before the formal method was applied.

4.3 Formal Modeling of Three Phase Model

In this section, we demonstrate the formal modeling process (Section 4.2.2) to formally represent the clinical knowledge models and validation process used in three phase knowledge acquisition process. Using theorem proving mechanism the validation process and knowledge models are verified for internal consistencies.

4.3.1 Modeling Problem

The modeling problem investigates the basic concepts used in knowledge acquisition for Smart CDSS, which target the clinical objectives. The fundamental concepts in Smart CDSS are PM, CKM, and R-CKM, which represent the clinical treatment plan for head and neck cancer. These concepts will be represented as primitive types, free types, axioms, and schema in Z notation.

4.3.1.1 Primitive types

Primitive types constitute the basic building blocks of the problem that is under consideration. In Smart CDSS, the concepts relevant to the clinical knowledge, which play a pivotal role in knowledge acquisition and validation, are cancer treatments (e.g., chemotherapy, radiotherapy, and surgery), clinical objectives (e.g., intervention for a treatment plan), and evidence (e.g., combined chemo-radiotherapy has a significant effect on patient survival; a success rate of 92%). These concepts are represented as a set using primitive types (Type Definition 1 (line 1). Furthermore, cancer treatment is abbreviated (line 3) as a general treatment to provide clarity in further

specifications.

Type Definition 1 Primitive types for clinical knowledge modelling

$$[CancerTreatment, ClinicalObjectives, Evidences] \quad (1)$$

$$[Condition, ConditionAttribute, ConditionOperator, ConditionValue] \quad (2)$$

$$Treatments == \{CancerTreatment\} \quad (3)$$

In order to define the formal representation of the knowledge model, primitive types are needed to capture the basic concepts used in the knowledge representation scheme. In Smart CDSS, the knowledge models follow decision tree representations where the combination of conditions with logical relationships constitutes the decision path. The *Condition* includes clinical concepts as an attribute with an exact value or a range of value sets. For example, patient categorization that is based on the severity of cancer can be represented as a condition in the decision tree test node $TreatmentIntent = radical$. These concepts are represented as Z primitive types (shown in Type Definition 1 (line 2), and the corresponding language syntax for the condition is provided by Type Definition 2.

Type Definition 2 BNF for some primitive types

$$\begin{aligned} Condition ::= {} & [NOT](\langle ConditionAttribute\rangle\langle ConditionOperator\rangle\langle ConditionValue\rangle) \mid \\ & \{[NOT](\langle ConditionAttribute\rangle\langle ConditionOperator\rangle\langle ConditionValue\rangle) AND \mid OR \\ & [NOT](\langle ConditionAttribute\rangle\langle ConditionOperator\rangle\langle ConditionValue\rangle)\} \end{aligned} \quad (1)$$

$$ConditionOperator ::= < \mid > \mid = \mid \neq \mid \leq \mid \geq \quad (2)$$

Moreover, free types in Smart CDSS are used to reflect the semantics of the clinical concepts and provide conformance to decision tree representation formalism. For example, treatments provided to patients follow a sequence according to standard guidelines and protocols; radiotherapy treatments and surgery are followed by ChemoInduction for radical patients (from CKM). In order to capture these semantics, Type Definition 3 defines two free types: TreatmentSet and TreatmentPlan (line 1 and line 2, respectively).

In Smart CDSS, the knowledge model typically uses decision tree representation; however, PM is different from CKM and R-CKM in terms the decision path. PM does not include treatments as a condition. To distinctly represent this formalism, ConditionCKs (line 3) defines a special

condition as a free type for CKM and R-CKM. Similarly, refinement in final R-CKM is represented as RefinedTreatmentPlan (line 4), which dictates the addition of a treatment to R-CKM as a type of refinement (indicating the placement at a particular position in the decision path).

Type Definition 3 Free types to capture semantics of knowledge artifacts

$$treatmentSet ::= InitTPlan\langle\langle \mathbb{F}\, Treatments\rangle\rangle \mid followedTPlan\langle\langle treatmentSet\rangle\rangle \quad (1)$$

$$TreatmentPlan ::= treatmentSet\langle\langle \mathbb{N} \times \mathrm{seq}\, TreatmentPlan\rangle\rangle \quad (2)$$

$$ConditionKMs ::= \mathrm{seq}\, Condition \frown TreatmentPlan \quad (3)$$

$$RefinedTreatmentPlan ::= \mathbb{N} \times TreatmentPlan \quad (4)$$

4.3.1.2 Knowledge Models

Clinical knowledge models, such as PM, CKM, and R-CKM, are represented as axioms and schemas. Subsequent sections explain the specifications for these models.

Prediction model specifications Prediction model specifications cover the properties associated with PM in accordance with decision tree formalism. The PM specifications are created using an axiom (Axiom 1) and the PredictionModel schema (Schema 1). The axiomatic definition for PM models the basic constructs using decision tree formalism. Accordingly, the decision paths are the main constituents of the decision tree skeleton where a combination of logically-related conditions makes a single decision path that has one conclusion. The conditions and conclusion are also known as nodes of the decision tree where the conclusion is always a leaf node. The decision tree obtained from the data (using machine-learning approaches) also has accuracy in terms of possessing correctly classified data cases (i.e., using 10-fold cross validation).

In Smart CDSS, PM follows decision tree formalism, which is obtained from patient medical records where conditions are used to represent patient information (e.g., symptoms, problems (diseases), clinical observations, and other demographic information (patient history)) and the conclusion represents the treatment plan. Axiom 1 includes declarations for the decision path as a partial function from the condition to the treatment plan (line 3). Its accuracy is represented as a total function from the decision path to the accuracy (line 5). The decision path conditions are

represented as a finite set of the *Condition* (line 1), and the conclusion is represented as a finite set of the *TreatmentPlan* (line 2). In order to reinforce the basic properties of the PM decision path, predicates are used to constrain the defined properties. For example, the PM decision path accuracy must lie between 0 and 100 (line 8). For all decision paths, there must exist one conclusion and the conclusion must be a *TreatmentPlan* (line 11).

Moreover, for validation purposes, we also associate the evidence (if it exists) with the treatment plan recommendation that is provided by the decision path in PM. Evidence is a finite set, which can represent the effectiveness of the treatment plan in terms of the success rate (as a percentage) in patient cases. It may also include external evidence from other research works. Therefore, the decision path may have evidence that is represented as a partial function from the decision path to the set of evidence (line 7 and line 12).

Axiom 1 Prediction model specifications

$$
\begin{array}{l}
decisionPathConditionPM : \mathbb{F}\,\mathrm{seq}\,Condition \quad (1)\\
Conclusion : \mathbb{F}\,TreatmentPlan \quad (2)\\
decisionPath : Condition \pfun TreatmentPlan \quad (3)\\
accuracy : \mathbb{Z} \quad (4)\\
decisionPathAccuracy : decisionPath \to accuracy \quad (5)\\
evidences : \mathbb{F}\,Evidences \quad (6)\\
decPathEvidences : decisionPath \pfun Evidences \quad (7)\\
\hline
0 \leq accuracy \leq 100 \quad (8)\\
decisionPathConditionPM = \mathrm{dom}\,decisionPath \quad (9)\\
Conclusion = \mathrm{ran}\,decisionPath \quad (10)\\
\forall\, con : Condition \mid con \in decisionPathConditionPM \bullet \quad (11)\\
\qquad \exists_1\, conclusion : TreatmentPlan \mid conclusion \in Conclusion \bullet decisionPath(con) = conclusion\\
evidences = \mathrm{ran}\,decPathEvidences \quad (12)
\end{array}
$$

Prediction model specification is further extended through the *PredictionModel* schema (Schema 1). PM is formally represented as a decision tree that is associated with the clinical objectives using the injective function from the decision path to the clinical objectives (lines 1, 2, and 7). The PM is associated with accuracy, which is the weighted mean accuracy of all of the decision paths in PM (lines 2, 4, and 8). For simplicity, we consider an equal number of patient cases for each decision path; this simplifies the accuracy of PM (line 8). In addition, PM is a decision tree, which means it must include one root node that must be a condition (lines 5 and 9).

Schema 1 Prediction model specifications

PredictionModel

$$PM : \mathbb{P}\, decisionPath \quad (1)$$
$$accuracyPM : \mathbb{F}\, \mathbb{Z} \quad (2)$$
$$predictionModels : decisionPath \rightarrowtail\!\!\!\!\!\!\rightarrow ClinicalObjectives \quad (3)$$
$$predictionModelsAccuracy : PM \rightarrow accuracy \quad (4)$$
$$rootPM : \mathrm{seq}\, Condition \quad (5)$$

$$0 \leq accuracyPM \leq 100 \quad (6)$$
$$PM = \mathrm{dom}\, predictionModels \quad (7)$$
$$accuracyPM = (\textbf{let}\ pathsAcc == \{pathsAcc : \mathbb{Z} \mid (\forall\, dp : decisionPath \mid dp \in PM \bullet pathsAcc = decisionPathAccuracy(dp) + pathsAcc)\})/\#PM \quad (8)$$
$$rootPM = \forall\, dp : decisionPath \mid dp \in PM \bullet \langle \bigcap(\mathrm{ran}(\mathrm{dom}\, dp))\rangle \quad (9)$$

Clinical knowledge models specifications Clinical knowledge model specification represents the formalism of CKM as an axiom (Axiom 2) and the schema *ClinicalKnowledgeModel*(Schema 2). CKM is a knowledge model that represents clinical guidelines with decision tree formalism. As described in a previous section, unlike PM, the CKM decision path also considers the treatment plan as a condition and the conclusion is always a treatment plan. Therefore, decision path modeling is represented as a partial function from free type *ConditionKMs* to the treatment plan with axiomatic definition Axion 2 (line 3). To reinforce the idea that the CKM decision path may contain treatment plans as a condition, a constraint is defined in the predicate at Axiom 2 (line 6). Moreover, every decision path must have a starting condition other than a treatment plan, which is defined in the predicate at Axiom 2 (line 7).

Axiom 2 Clinical knowledge model specifications

$$decisionPathConditionCKM : \mathbb{F}\, ConditionKMs \quad (1)$$
$$ConclusionCKM : \mathbb{F}\, TreatmentPlan \quad (2)$$
$$decisionPathCKM : ConditionKMs \nrightarrow TreatmentPlan \quad (3)$$

$$decisionPathConditionCKM = \mathrm{dom}\, decisionPathCKM \quad (4)$$
$$ConclusionCKM = \mathrm{ran}\, decisionPathCKM \quad (5)$$
$$(\mathrm{ran}\, ConclusionCKM \cap \mathrm{ran}(\mathrm{dom}\, decisionPathCKM)) \subset \mathrm{ran}\, decisionPathConditionCKM \quad (6)$$
$$head\, (\mathrm{dom}\, decisionPathCKM) \notin \mathrm{ran}\, ConclusionCKM \cap \mathrm{ran}\, decisionPathConditionCKM \quad (7)$$

The *ClinicalKnowledgeModel* schema Schema 2) further extends the CKM semantics. Ac-

cording to the definition of CKM, it possesses guidelines that follow decision tree formalism and are associated with the clinical objectives. For example, CKM (Figure 6.1) is an NCCN guideline that provides standard-based treatment plans for tumors in oral cavities. By using the schema definition (Schema 2), the guideline is a total function from the standard decision paths to the clinical objectives (line 2). CKM is a set of logically-related decision paths in the guidelines that fulfills target clinical objectives (lines 1 and 4).

Every decision path in CKM must start with a condition (other than a treatment plan) and CKM must have only one root condition, which should be common to all decision paths. These constraints are defined as predicates (lines 5 and 7) in the schema (Schema 2).

In CKM, the treatment plan comes as a condition in one decision path and may act as a conclusion for another decision path. In other words, the CKM conclusion may occur in an intermediate node. To capture this semantic predicate (line 6) in the schema, (Schema 2) is defined.

Schema 2 Clinical knowledge model specifications

ClinicalKnowledgeModel

$$CKM : \mathbb{P}\, decisionPathCKM \tag{1}$$
$$guidelines : decisionPathCKM \rightarrow ClinicalObjectives \tag{2}$$
$$rootCKM : \operatorname{seq} Condition \tag{3}$$

$$CKM = \operatorname{dom} guidelines \tag{4}$$
$$\forall\, dp : decisionPathCKM \mid dp \in CKM \bullet$$
$$\quad head\,(\operatorname{dom} dp) \notin \operatorname{ran} ConclusionCKM \cap \operatorname{ran} decisionPathConditionCKM \tag{5}$$
$$\exists\, dp : decisionPathCKM \fatsemi dp_1 : decisionPathCKM \mid dp, dp_1 \in CKM \bullet$$
$$\quad last\,(\operatorname{dom} dp) = \operatorname{ran} dp_1 \Leftrightarrow \operatorname{dom} dp_1 = \operatorname{dom} dp \setminus last\,(\operatorname{dom} dp) \tag{6}$$
$$rootCKM = \forall\, dp : decisionPathCKM \mid dp \in CKM \bullet \langle \bigcap(\operatorname{ran}(\operatorname{dom} dp)) \rangle \tag{7}$$

Refined clinical knowledge models specifications Refined clinical knowledge model specifications represent R-CKM formalism as an axiom (Axiom 3) and a schema (*RefinedClinicalKnowledgeModel*, Schema 3). R-CKM follows the formalism of CKM in that it also uses decision tree representation, which includes decision paths that have been formally validated from standard guidelines or possess sufficient evidence to prove their effectiveness. In this respect, the R-CKM decision path can be modeled (similarly to CKM) as a partial function from free type *ConditionKms* to the treatment plan; this is shown in the axiomatic definition (line 3). Similarly, decision paths that are not present in the guidelines are provided with evidence that is represented

as a partial function from the decision path to the evidence.

The predicates defined in Axiom 3 (lines 13, 14) capture the semantics of the decision path in R-CKM; a treatment plan can be used as a condition in the decision path and the decision path must start with a condition (this should not be a treatment plan).

In addition to CKM formalism, decision paths in R-CKM actually become a part of the model after passing through formal validation process and refinements (Figure **??**). In this respect, the decision path in R-CKM is associated with accuracy as a total function from the decision path to the accuracy (line 5). Similarly, refinement in R-CKM is represented as an injective function from the refined treatment plan (a free type, line 4, Type Definition 3) to the PM decision path (line 8).

Axiom 3 Refined clinical knowledge model specifications

$$
\begin{array}{|l}
\mathit{decisionPathConditionRCKM} : \mathbb{F}\ \mathit{ConditionKMs} \quad (1)\\
\mathit{ConclusionRCKM} : \mathbb{F}\ \mathit{TreatmentPlan} \quad (2)\\
\mathit{decisionPathRCKM} : \mathit{ConditionKMs} \nrightarrow \mathit{TreatmentPlan} \quad (3)\\
\mathit{accuracy} : \mathbb{Z} \quad (4)\\
\mathit{decPathRCKMAccuracy} : \mathit{decisionPathRCKM} \rightarrow \mathit{accuracy} \quad (5)\\
\mathit{evidences} : \mathbb{F}\ \mathit{Evidences} \quad (6)\\
\mathit{decPathRCKMEvidences} : \mathit{decisionPathRCKM} \nrightarrow \mathit{Evidences} \quad (7)\\
\mathit{refinedTPlan} : \mathbb{F}\ \mathit{RefinedTreatmentPlan} \quad (8)\\
\mathit{refinementsDecPath} : \mathit{RefinedTreatmentPlan} \rightarrowtail \mathit{decisionPath} \quad (9)\\
\hline
0 \leq \mathit{accuracy} \leq 100 \quad (10)\\
\mathit{decisionPathConditionRCKM} = \mathrm{dom}\ \mathit{decisionPathRCKM} \quad (11)\\
\mathit{ConclusionRCKM} = \mathrm{ran}\ \mathit{decisionPathRCKM} \quad (12)\\
(\mathrm{ran}\ \mathit{ConclusionRCKM} \cap \mathrm{ran}\ \mathit{decisionPathConditionRCKM}) \subset \mathrm{ran}\ \mathit{decisionPathConditionRCKM} \quad (13)\\
\mathit{head}\ (\mathit{decisionPathConditionRCKM}) \notin \mathrm{ran}\ \mathit{ConclusionRCKM} \cap \mathrm{ran}\ \mathit{decisionPathConditionRCKM} \quad (14)\\
\mathit{evidences} = \mathrm{ran}\ \mathit{decPathRCKMEvidences} \quad (15)\\
\mathit{refinedTPlan} = \mathrm{dom}\ \mathit{refinementsDecPath} \quad (16)
\end{array}
$$

The declarations and predicates of schema *RefinedClinicalKnowledgModel* (Schema 3) are mostly similar to those of CKM (Schema 2); both share the same formalism. New contents include support for the overall accuracy of R-CKM, which is declared as a total function form of RCKM into the accuracy (line 7). The intended accuracy is calculated as the weighted mean accuracy for all of the decision paths in R-CKM (line 12).

Moreover, R-CKM is derived from PM and validated against CKM (guidelines); thus, the total function is defined from the R-CKM decision paths to the intended CKM (line 4), and R-CKM is modeled as a finite set of related decision paths (line 3) associated with CKM (line 9).

Furthermore, a predicate is added to the schema (line 13), which constrains all of the decision paths; these must be derived from PM and aligned to CKM. Similarly, using schema inclusion, *PredictionModel* (Schema 1) and *ClinicalKnowledgeModel* (Schema 2) are also included (lines 1 and 2) into the *RefinedClinicalKnowledgeModel* (Schema 3) in order to make the contents of PM and CKM available to the R-CKM model.

Schema 3 Refined clinical knowledge model specifications

RefinedClinicalKnowledgeModel

$$
\begin{array}{ll}
\mathit{PredictionModel} & (1)\\
\mathit{ClinicalKnowledgeModel} & (2)\\
\mathit{RCKM} : \mathbb{F}\, \mathit{decisionPathRCKM} & (3)\\
\mathit{refinedCKM} : \mathit{decisionPathRCKM} \rightarrow \mathit{CKM} & (4)\\
\mathit{rootRCKM} : \operatorname{seq} \mathit{Condition} & (5)\\
\mathit{accuracyRCKM} : \mathbb{F}\, \mathbb{Z} & (6)\\
\mathit{refinedCKMsAccuracy} : \mathit{RCKM} \rightarrow \mathit{accuracy} & (7)\\
\hline
0 \leq \mathit{accuracyRCKM} \leq 100 & (8)\\
\mathit{RCKM} = \operatorname{dom} \mathit{refinedCKM} & (9)\\
\forall\, \mathit{dp} : \mathit{decisionPathRCKM} \mid \mathit{dp} \in \mathit{RCKM} \bullet & \\
\quad \mathit{head}\,(\operatorname{dom} \mathit{dp}) \notin \operatorname{ran} \mathit{ConclusionRCKM} \cap \operatorname{ran} \mathit{decisionPathConditionRCKM} & (10)\\
\exists\, \mathit{dp} : \mathit{decisionPathRCKM} \fatsemi \mathit{dp}_1 : \mathit{decisionPathRCKM} \mid \mathit{dp}, \mathit{dp}_1 \in \mathit{RCKM} \bullet & \\
\quad \mathit{last}\,(\operatorname{dom} \mathit{dp}) = \operatorname{ran} \mathit{dp}_1 \Leftrightarrow \operatorname{dom} \mathit{dp}_1 = \operatorname{dom} \mathit{dp} \setminus \mathit{last}\,(\operatorname{dom} \mathit{dp}) & (11)\\
\mathit{accuracyRCKM} = (\textbf{let}\ \mathit{pathsAcc} == \{\mathit{pathsAcc} : \mathbb{Z} \mid \mathit{RCKM} \neq \emptyset \wedge & \\
\quad (\forall\, \mathit{dp} : \mathit{decisionPathRCKM} \mid \mathit{dp} \in \mathit{RCKM} \bullet \mathit{pathsAcc} = & \\
\quad \mathit{decPathRCKMAccuracy}(\mathit{dp}) + \mathit{pathsAcc})\}) / \#\mathit{RCKM} & (12)\\
\forall\, p_{\mathit{rckm}} : \mathit{decisionPathRCKM} \mid p_{\mathit{rckm}} \in \mathit{RCKM} \bullet & \\
\quad \exists\, p_{\mathit{pm}} : \mathit{decisionPath}, p_{\mathit{ckm}} : \mathit{decisionPathCKM} \mid & \\
\quad p_{\mathit{pm}} \in \mathit{PM} \wedge p_{\mathit{ckm}} \in \mathit{CKM} \bullet \operatorname{dom} p_{\mathit{rckm}} = \operatorname{dom} p_{\mathit{pm}} \cup \operatorname{dom} p_{\mathit{ckm}} & (13)\\
\mathit{rootRCKM} = \forall\, \mathit{dp} : \mathit{decisionPathRCKM} \mid \mathit{dp} \in \mathit{RCKM} \bullet \langle \bigcap(\operatorname{ran}(\operatorname{dom} \mathit{dp})) \rangle & (14)
\end{array}
$$

Validation process specifications Validation process specifications encompass the validation process (Figure 3.4) and properly represent the validation criteria defined in the clinical knowledge pool (Section 3.3.2: Table 3.3). The schema *PMPathValidation* (Schema 4) models the basic semantics of the validation process. It includes schema *RefinedClinicalKnowledgeModel* (line 1), which is used to associate the validation process with R-CKM. It also provides declaration for the two inputs that the validation process is supposed to consume: the PM decision path (line 2) and the minimal accuracy (assigned by a domain expert) that is required for the PM decision path (line 3) to be accepted into R-CKM.

The validation criteria defined in the clinical knowledge pool are reflected as predicates in the schema *PMPathValidation* (lines 4-7). The first two primary (compulsory) criteria are defined in the schema as conjunction predicates (lines 4 and 5), and two alternate criteria are represented through disjunction predicates (lines 6 and 7).

Schema 4 Validation process specifications

PMPathValidation

$$
\begin{array}{ll}
\mathit{RefinedClinicalKnowledgeModel} & (1)\\
\mathit{dp_{pm}}? : \mathit{decisionPath} & (2)\\
\mathit{qualifiedAcc}? : \mathbb{Z} & (3)\\
\hline
\mathit{dp_{pm}}? \in \mathit{PM} \land \mathit{decisionPathAccuracy}(\mathit{dp_{pm}}?) \geq \mathit{qualifiedAcc}? & (4)\\
\forall t_1, t_2 : \mathit{treatmentSet} \mid t_1, t_2 \in \mathrm{ran}(\mathrm{ran}(\mathit{dp_{pm}}?)) \land \mathit{TreatmentPlan}^{\sim}(t_1) > \mathit{TreatmentPlan}^{\sim}(t_2) \bullet & \\
\quad \exists\, \mathit{dp_{ckm}} : \mathit{decisionPathCKM};\ t_3, t_4 : \mathit{treatmentSet} \mid \mathit{dp_{ckm}} \in \mathit{CKM}, & \\
\quad t_3, t_4 \in (\mathrm{ran}(\mathrm{dom}(\mathit{dp_{ckm}})) \cap \mathrm{ran}(\mathit{ConclusionCKM})) \cup \mathrm{ran}(\mathrm{ran}(\mathit{dp_{ckm}})) \bullet & \\
\quad (t_3 = t_1 \land t_4 = t_2) \Rightarrow \mathit{TreatmentPlan}^{\sim}(t_3) > \mathit{TreatmentPlan}^{\sim}(t_4) & (5)\\
\mathit{decPathEvidences}(\mathit{dp_{pm}}?) \neq \emptyset \lor & (6)\\
\quad \exists\, \mathit{dp_{ckm}} : \mathit{decisionPathCKM} \mid \mathit{dp_{ckm}} \in \mathit{CKM} \bullet & \\
\quad\quad (\mathrm{ran}(\mathrm{dom}(\mathit{dp_{pm}}?)) \subseteq \mathrm{ran}(\mathrm{dom}(\mathit{dp_{ckm}})) \Rightarrow & \\
\quad\quad \mathrm{ran}(\mathrm{ran}(\mathit{dp_{pm}}?)) \subseteq & \\
\quad\quad (\mathrm{ran}(\mathrm{dom}(\mathit{dp_{ckm}})) \cap \mathrm{ran}(\mathit{ConclusionCKM})) \cup \mathrm{ran}(\mathrm{ran}(\mathit{dp_{ckm}}))) & (7)
\end{array}
$$

4.3.2 Defining functions and models state

The main functions of knowledge models are to evolve R-CKM based on the validation of the decision path. Furthermore, operations related to the retrieval of concepts from knowledge models are discussed. The only evolving model is R-CKM, so the state model for R-CKM is presented.

4.3.2.1 Operations on knowledge models

Two types of operations are defined for the knowledge model. For PM and CKM, only retrieval operations are required to represent access to different components of the model. Alternatively, for R-CKM, retrieval and state change operations are required to represent the model.

Operations for PM Prediction model specification provides a set of operational schemas that can be used to access and retrieve various components of the PM. In order to retrieve the detailed components of the PM, the schema *RetreiveDetailedPredictionModel* (Schema 5) is defined

as the composite schema. This is a combination (using conjunction) of two individual schemas (Schema 6 and Schema 7) and provides details about PM components.

Schema 5 Retrieving prediction model with detail components

$$RetreiveDetailedPredictionModel \mathrel{\widehat=} RetreivePredictionModel \land RetreiveDecisionPathsPM$$

RetreivePredictionModel (Schema 6) is an operational schema that retrieves primitive information about PM for the given clinical objectives. It considers the clinical objective as an input (line 2) and produces PM and its accuracy as the output (lines 3 and 4). The retrieval process is represented as a predicate in the schema (line 5). Because *RetreivePredictionModel* only retrieves PM (but does not change PM), the state change indicated in the schema (line 1) is provided; there is no change in the original model *PredictionModel* (Schema 1).

Schema 6 Retrieving prediction model

RetreivePredictionModel

$\Xi PredictionModel$ (1)
$clinicalObj? : ClinicalObjectives$ (2)
$PreModel! : PM$ (3)
$PreModelAccuracy! : accuracyPM$ (4)

$PreModelAccuracy! = predictionModelsAccuracy(PreModel! = \{dp_{pm} : decisionPath \mid predictionModels(dp_{pm}) = clinicalObj?\})$ (5)

RetreiveDecisionPathsPM (Schema 7) is an operational schema that provides further details to the content of *RetreivePredictionModel* (line 1). The main input is a PM (line 3), and it yields detailed decision paths (line 4) and the corresponding accuracy (line 5). The retrieval process is represented as a predicate in the schema (line 6).

Operations for CKM The clinical knowledge model specifications used for the retrieval of CKM components followed the same pattern of operational schemas developed for PM. A composite schema *RetreiveDetailedCKM* (Schema 8) provides detailed information about CKM by using a combination of the two operational schemas (Schema 9 and Schema 10).

Schema 7 Retrieving prediction model decision paths

RetreiveDecisionPathsPM

$\Xi PredictionModel$ (1)
$RetreivePredictionModel$ (2)
$pm? : PM$ (3)
$decPath! : decisionPath$ (4)
$decPathAccuracy! : accuracy$ (5)

$\forall dp : decisionPath \mid dp \in (pm? \lhd predictionModels) \bullet decPath! = dp$
$\land\ decPathAccuracy! = decisionPathAccuracy(dp)$ (6)

Schema 8 Retrieving clinical knowledge model with detail components

$RetreiveDetailedCKM \mathrel{\widehat=} RetreiveCKM \land RetreiveDecisionPathsCKM$

RetreiveCKM (Schema 9) is an operational schema that consumes the clinical objective as an input (line 2) and produces the clinical knowledge model as an output (line 3) with primitive information. The output process is represented as a predicate in the schema (line 4).

Schema 9 Retrieving clinical knowledge model

RetreiveCKM

$\Xi ClinicalKnowledgeModel$ (1)
$clinicalObj? : ClinicalObjectives$ (2)
$ClinicalKnowledgeModel! : CKM$ (3)

$ClinicalKnowledgeModel! = \{dp_{ckm} : decisionPathCKM \mid guidelines(dp_{ckm}) = clinicalObj?\}$ (4)

RetreiveDecisionPathsCKM (Schema 10) extends the *RetreiveCKM* (Schema 9) to produce further details related to each decision path in CKM (line 2). In conjunction with *RetreiveCKM*, it consumes CKM as an input (line 3) and provides detailed information about all of the decision paths as an output (line 4) using a predicate in the schema (line 5).

Schema 10 Retrieving clinical knowledge model decision paths

RetreiveDecisionPathsCKM

$\Xi ClinicalKnowledgeModel$ (1)

$RetreiveCKM$ (2)

$ckm? : CKM$ (3)

$decPathCKM! : decisionPathCKM$ (4)

$\forall dp : decisionPathCKM \mid dp \in (ckm? \lhd guidelines) \bullet decPathCKM! = dp$ (5)

Operations for R-CKM R-CKM is the only knowledge model that evolves through proper validation processes using PM and CKM. Therefore, in addition to retrieval operations, R-CKM also requires definitions for operations that represent the addition of new decision paths into the final model (in the presence of the validation criteria). For brevity purposes, we only concentrate on operations that are related to the evolution of R-CKM. We skip the representation of the operational schema used for the retrieval of R-CKM components, which follow patterns similar to PM and CKM.

EvolveRCKM (Schema 11) is an operational schema that mainly represents the evolution of the R-CKM model. The evolution of R-CKM is mainly described as a two-step process: (1) a decision path from PM is evaluated against the validation criteria and (2) the selected decision path is refined further (if needed) and added to the R-CKM. Accordingly, *EvolveRCKM* (Schema 11) is defined as a composite operational schema to reflect these steps. This composition is modeled as the combination of two schemas: *PMPathValidation* (Schema 4) and *AddPathRCKM* (Schema 12).

Schema 11 Evolution of R-CKM

$EvolveRCKM \mathrel{\widehat=} PMPathValidation \land AddPathRCKM$

- *Declaration(Input):* The *AddPathRCKM* schema expects two inputs: a candidate decision path from PM (line 2) and the desired treatment plan refinements in the decision path (line 3).
- *Declaration(Output):* The final decision path of R-CKM, after refinements, is considered to be an output for the schema *AddPathRCKM* (line 4).

- *Predicates(Pre-conditions):* These include a set of predicates (lines 5-12) that must be met before any changes are made to the R-CKM model (Schema 3:*RefinedClinicalKnowledgeModel*). Most of these pre-conditions are not known in advance but are calculated using the one-point rule and simplification proofs (Section 4.3.3.2). We will describe some important pre-conditions, as evaluation results, for the formal method plugin in Section 6.2.1.

- *Predicates(Refinements):* The refinement process is performed on the candidate decision path of PM (line 14), and the path is modified (line 15) according to the required treatment plan that is mentioned by the suggested refinements, which are provided as an input (line 3).
- *Predicates(Evolution):* The R-CKM is evolved with the newly refined decision path. All of the relevant components of the *RefinedClinicalKnowledgeModel* schema are indicated through primed statements in the operational schema (lines 16-28). These primed statements basically represent the new change state of the R-CKM model; this will be explained in the following section.

4.3.2.2 Model states for knowledge models

Modifications are only made to R-CKM upon evolution through the *EvolveRCKM* (Schema 11) operational schema using the combination of schema *AddPathRCKM* and schema *PMPathValidation*. *PMPathValidation* (Schema 4) validates a decision path of PM against the validation criteria and makes no change to the R-CKM model. Thus, *AddPathRCKM* (Schema 12) makes refinements to the decision path of PM and adds the refined path to R-CKM, which ultimately makes changes to the relevant components of the R-CKM. In this respect, the state model of *RefinedClinicalKnowledgeModel* (Schema 3) reflects changes in accordance with the AddPathRCKM operational schema. The schema *RefinedClinicalKnowledgeModel'* (Schema 13) represents the R-CKM model state, which encapsulates all of the relevant statements from R-CKM specifications (Axiom 3 and Schema 3).

The *AddPathRCKM* operational schema is invoked in conjunction with *PMPathValidation* through the *EvolveRCKM* operational schema, and *PMPathValidation* validates the decision path of PM. Then, the changes made to the R-CKM model (*RefinedClinicalKnowledge-*

Schema 12 Adding PM decision path to R-CKM

AddPathRCKM

$$\Delta RefinedClinicalKnowledgeModel \tag{1}$$
$$dp_{pm}? : decisionPath \tag{2}$$
$$refinements? : \mathbb{F}\, RefinedTreatmentPlan \tag{3}$$
$$rckmPath! : decisionPathRCKM \tag{4}$$

$$RCKM \neq \emptyset \Rightarrow head\,(\mathrm{dom}\, dp_{pm}?) = rootRCKM \tag{5}$$
$$\forall pos : \mathbb{N} \mid pos \in \mathrm{dom}\, refinements? \bullet pos > 1 \land pos \leq (\#(\mathrm{dom}\, dp_{pm}?) + \#(\mathrm{ran}\, dp_{pm}?)) \tag{6}$$
$$\mathrm{ran}(\mathrm{dom}\, rckmPath!) \subset \mathrm{ran}\, decisionPathConditionRCKM \tag{7}$$
$$\mathrm{ran}(\mathrm{ran}\, rckmPath!) \subset \mathrm{ran}\, ConclusionRCKM \tag{8}$$
$$(\mathrm{ran}(\mathrm{ran}\, rckmPath!) \cap \mathrm{ran}\, decisionPathConditionRCKM) \subset \mathrm{ran}\, decisionPathConditionRCKM \tag{9}$$
$$0 \leq decPathRCKMAccuracy(rckmPath!) \leq 100 \tag{10}$$
$$head\,(\mathrm{dom}\, rckmPath!) \notin \mathrm{ran}\, ConclusionRCKM \cap \mathrm{ran}\, decisionPathConditionRCKM \tag{11}$$
$$\exists dp : decisionPathRCKM \mid dp \in RCKM \bullet \mathrm{dom}\, rckmPath! = \mathrm{dom}\, dp \setminus last\,(\mathrm{dom}\, dp) \Rightarrow last\,(\mathrm{dom}\, dp) = \mathrm{ran}\, rckmPath! \tag{12}$$
$$\mathrm{dom}\, rckmPath! = \exists p_{ckm} : decisionPathCKM \mid p_{ckm} \in CKM \bullet \mathrm{dom}(p_{pm}?) \cup \mathrm{dom}\, p_{ckm} \tag{13}$$
$$\mathrm{ran}\, rckmPaht! = \mathrm{ran}\, dp_{pm}? \tag{14}$$
$$\forall r : RefinedTreatmentPlan \mid r \in refinements? \bullet rckmPath! = \frown/\langle \{t_p : TreatmentPlan \bullet (1 \,..\, \mathrm{dom}\, r, t_p)\} \upharpoonright \mathrm{dom}\, rckmPath!,\ \mathrm{ran}\, r, \{t_p : TreatmentPlan \bullet (\mathrm{dom}\, r + 1 \,..\, \#(\mathrm{dom}\, rckmPath!), t_p)\} \upharpoonright \mathrm{dom}\, rckmPath! \rangle \tag{15}$$
$$decisionPathRCKM' = decisionPathRCKM \cup \{\mathrm{dom}\, rckmPath! \mapsto \mathrm{ran}\, rckmPath!\} \tag{16}$$
$$decisionPathConditionRCKM' = decisionPathConditionRCKM \cup \mathrm{dom}\, rckmPath! \tag{17}$$
$$refinedTPlan' = refinedTPlan \cup refinements? \tag{18}$$
$$refinementsDecPath' = refinementsDecPath \cup \{refinements? \mapsto dp_{pm}?\} \tag{19}$$
$$ConclusionRCKM' = ConclusionRCKM \cup \mathrm{ran}\, rckmPath! \tag{20}$$
$$decPathRCKMAccuracy' = decPathRCKMAccuracy \cup \{rckmPath! \mapsto decPathRCKMAccuracy(rckmPath!)\} \tag{21}$$
$$accuracyRCKM' = \frac{accuracyRCKM \times \#RCKM + decPathRCKMAccuracy'(rckmPath!)}{\#RCKM + 1} \tag{22}$$
$$\#RCKM' = \#RCKM + 1 \tag{23}$$
$$evidences' = evidences \cup decPathEvidences(dp_{pm}?) \tag{24}$$
$$decPathRCKMEvidences' = decPathRCKMEvidences \cup \{rckmPath! \mapsto decPathEvidences(dp_{pm}?)\} \tag{25}$$
$$RCKM' = RCKM \oplus \{\mathrm{dom}\, rckmPath! \mapsto \mathrm{ran}\, rckmPath!\} \tag{26}$$
$$refinedCKM' = refinedCKM \oplus \{rckmPath! \mapsto CKM\} \tag{27}$$
$$rootRCKM' = rootRCKM = head\,(\mathrm{dom}\, dp_{pm}?) \tag{28}$$

Schema 13 R-CKM state after modification

RefinedClinicalKnowledgeModel′

$$
\begin{array}{ll}
\mathit{PredictionModel} & (1)\\
\mathit{ClinicalKnowledgeModel} & (2)\\
\mathit{decisionPathConditionRCKM}' : \mathbb{F}\ \mathit{ConditionKMs} & (3)\\
\mathit{ConclusionRCKM}' : \mathbb{F}\ \mathit{TreatmentPlan} & (4)\\
\mathit{decisionPathRCKM}' : \mathit{ConditionKMs} \pfun \mathit{TreatmentPlan} & (5)\\
\mathit{decPathRCKMAccuracy}' : \mathit{decisionPathRCKM}' \fun \mathit{accuracy} & (6)\\
\mathit{evidences}' : \mathbb{F}\ \mathit{Evidences} & (7)\\
\mathit{decPathRCKMEvidences}' : \mathit{decisionPathRCKM}' \pfun \mathit{Evidences} & (8)\\
\mathit{refinedTPlan}' : \mathbb{F}\ \mathit{RefinedTreatmentPlan} & (9)\\
\mathit{refinementsDecPath}' : \mathit{RefinedTreatmentPlan} \rightarrowtail \mathit{decisionPath} & (10)\\
\mathit{RCKM}' : \mathbb{F}\ \mathit{decisionPathRCKM} & (11)\\
\mathit{refinedCKM}' : \mathit{decisionPathRCKM}' \fun \mathit{CKM} & (12)\\
\mathit{rootRCKM}' : \mathrm{seq}\ \mathit{Condition} & (13)\\
\mathit{accuracyRCKM}' : \mathbb{F}\ \mathbb{Z} & (14)\\
\mathit{refinedCKMsAccuracy}' : \mathit{RCKM}' \fun \mathit{accuracy} & (15)
\end{array}
$$

$$
\begin{array}{ll}
\mathit{decisionPathConditionRCKM}' = \mathrm{dom}\ \mathit{decisionPathRCKM}' & (16)\\
\mathit{ConclusionRCKM}' = \mathrm{ran}\ \mathit{decisionPathRCKM}' & (17)\\
(\mathrm{ran}\ \mathit{ConclusionRCKM}' \cap \mathrm{ran}\ \mathit{decisionPathConditionRCKM}') \subset \mathrm{ran}\ \mathit{decisionPathConditionRCKM}' & (18)\\
\mathit{head}\ (\mathit{decisionPathConditionRCKM}') \notin \mathrm{ran}\ \mathit{ConclusionRCKM}' \cap \mathrm{ran}\ \mathit{decisionPathConditionRCKM}' & (19)\\
\mathit{evidences}' = \mathrm{ran}\ \mathit{decPathRCKMEvidences}' & (20)\\
\mathit{refinedTPlan}' = \mathrm{dom}\ \mathit{refinementsDecPath}' & (21)\\
0 \leq \mathit{accuracyRCKM}' \leq 100 & (22)\\
\mathit{RCKM}' = \mathrm{dom}\ \mathit{refinedCKM}' & (23)\\
\forall\, \mathit{dp} : \mathit{decisionPathRCKM}' \mid \mathit{dp} \in \mathit{RCKM}' \bullet & \\
\quad \mathit{head}\ (\mathrm{dom}\ \mathit{dp}) \notin \mathrm{ran}\ \mathit{ConclusionRCKM}' \cap \mathrm{ran}\ \mathit{decisionPathConditionRCKM}' & (24)\\
\exists\, \mathit{dp} : \mathit{decisionPathRCKM}' \fatsemi \mathit{dp}_1 : \mathit{decisionPathRCKM}' \mid \mathit{dp}, \mathit{dp}_1 \in \mathit{RCKM}' \bullet & \\
\quad \mathit{last}\ (\mathrm{dom}\ \mathit{dp}) = \mathrm{ran}\ \mathit{dp}_1 \Leftrightarrow \mathrm{dom}\ \mathit{dp}_1 = \mathrm{dom}\ \mathit{dp} \setminus \mathit{last}\ (\mathrm{dom}\ \mathit{dp}) & (25)\\
\mathit{accuracyRCKM}' = (\textbf{let}\ \mathit{pathsAcc} == \{\mathit{pathsAcc} : \mathbb{Z} \mid \mathit{RCKM}' \neq \emptyset \wedge & \\
\quad (\forall\, \mathit{dp} : \mathit{decisionPathRCKM}' \mid \mathit{dp} \in \mathit{RCKM}' \bullet \mathit{pathsAcc} = & \\
\quad \mathit{decPathRCKMAccuracy}'(\mathit{dp}) + \mathit{pathsAcc})\}) / \#\mathit{RCKM}' & (26)\\
\forall\, p_{rckm} : \mathit{decisionPathRCKM}' \mid p_{rckm} \in \mathit{RCKM}' \bullet & \\
\quad \exists\, p_{pm} : \mathit{decisionPath}, p_{ckm} : \mathit{decisionPathCKM} \mid & \\
\quad p_{pm} \in \mathit{PM} \wedge p_{ckm} \in \mathit{CKM} \bullet \mathrm{dom}\ p_{rckm} = \mathrm{dom}\ p_{pm} \cup \mathrm{dom}\ p_{ckm} & (27)\\
\mathit{RCKM} \neq \emptyset \Rightarrow \mathit{rootRCKM}' = \mathit{rootRCKM} & (28)
\end{array}
$$

*Model':*Schema 13) by *AddPathRCKM* operational schema can be summarized as follows:

- A new decision path is added to R-CKM; this adds new conditions to the set of R-CKM conditions ((*Schema 12:* Lines 16 and 17). These changes are represented in the state model (Schema 13) at lines 3, 5, and 16.
- New refinements are introduced to a set of the R-CKM model, which results in the addition of a PM path with the associated refinements (*Schema 12:* Lines 18 and 19). These states are reflected in lines 9, 10, and 21 in Schema 13.
- With the new decision path, the R-CKM model is evolved for a new conclusion (*Schema 12:* Line 20), which yields new states in the model properties of *RCKMConclusion*, as indicated in the state model schema at lines 4 and 17.
- For the new R-CKM path, the accuracy of the path will be associated and the overall R-CKM accuracy is recalculated (*Schema 12:* Lines 21, 22, and 23). The resulting state changes are reflected at lines 6, 14, 15, 22, and 26 in the state model schema.
- Evidences of the PM's decision path is associated with the refined decision path in R-CKM (*Schema 12:* Lines 24 and 25). These changes are reflected at lines 7, 8, and 20 in the state model schema.
- Finally, R-CKM is evolved with the addition of a new decision path and the root condition is re-evaluated (*Schema 12:* Lines 26, 27, and 28). These evolutions change the states at multiple statements in the state model schema, as indicated in lines 11, 12, 13, 18, 19, 23, 24, 25, 27, and 28.

4.3.3 Proving validation process and R-CKM evolution consistency

4.3.3.1 Validation process consistency proof using Initialization Theorem

The Initialization theorem provides a mechanism to prove that the model (R-CKM) is consistent and fulfills the requirements. It is determined that the model at least has an initial state. Definition 1 defines the initialization theorem.

For the R-CKM model represented in the schema *RefinedClinicalKnowledgeModel* (Schema 3), the initial state is defined using the state schema *InitRCKM* (Schema 14).

Definition 1: *For the system state "State" and its initial state "StateInit", the initialization theorem takes the following form:*

$$\exists\, State' \bullet StateInit$$

Definition 1: Initialization Theorem

Schema 14 R-CKM Initial state

InitRCKM

$$RefinedClinicalKnowledgeModel' \tag{1}$$

$$accuracyRCKM' = 0 \tag{2}$$
$$RCKM' = \emptyset \tag{3}$$
$$refinedCKM' = \emptyset \tag{4}$$
$$rootRCKM' = \emptyset \tag{5}$$
$$refinedCKMsAccuracy' = \emptyset \tag{6}$$
$$decisionPathRCKM' = \emptyset \tag{7}$$
$$decisionPathConditionRCKM' = \emptyset \tag{8}$$
$$ConclusionRCKM' = \emptyset \tag{9}$$
$$decPathRCKMAccuracy' = \emptyset \tag{10}$$
$$evidences' = \emptyset \tag{11}$$
$$decPathRCKMEvidences' = \emptyset \tag{12}$$

For the given initial state *InitRCKM* of the R-CKM model's schema *RefinedClinicalKnowledgeModel*, the initialization theorem is represented by Theorem 1; this is inspired by the basic definition provided in Definition 1.

Theorem 1 Initialization theorem for initial state of R-CKM

$$\exists\, RefinedClinicalKnowledgeModel' \bullet InitRCKM$$

The proof of this initialization theorem leads to consistent specifications for the R-CKM model. For the modeling specifications, which include contradictions, it is almost impossible to prove the initial state; indirectly, this means that the model does not fulfill the desired requirements.

In order to prove the initialization theorem, we can take advantage of the one-point rule as well as some other set theory laws and fundamental definitions. The one-point rule is helpful in replacing the existential quantifier when the bound variable has an identity within the boundaries of the quantification expression. For the one-point rule, Definition 2 provides the basic background

related to replacing the existential quantifier.

Definition 2: *For the given predicate:*

$$\exists x : a \bullet p \wedge x = t$$

The one-point rule gives the following equivalence for the given existential quantifier.

$$(\exists x : a \bullet p \wedge x = t) \Leftrightarrow t \in a \wedge p[t/x]$$

Definition 2: The one-point rule

Following the definition of the one-point rule, and other fundamental laws and definitions, the initialization theorem is proven in Proof 1; this is straightforward and each step is explained with instructive definitions.

Proof 1 Proving initial state of R-CKM using initialization theorem *(Theorem 1)*

$$
\begin{aligned}
&\exists\, RefinedClinicalKnowledgeModel' \bullet InitRCKM \\
&\quad \Leftrightarrow RefinedClinicalKnowledgeModel' \bullet && [definition : InitRCKM] \\
&\qquad [RefinedClinicalKnowledgeModel' \mid \\
&\qquad\quad accuracyRCKM' = 0 \land \\
&\qquad\quad RCKM' = \emptyset \land \\
&\qquad\quad refinedCKM' = \emptyset \land \\
&\qquad\quad rootRCKM' = \emptyset \land \\
&\qquad\quad refinedCKMsAccuracy' = \emptyset] \\
&\quad \Leftrightarrow \exists\, RefinedClinicalKnowledgeModel' \bullet && [schema\ quantification] \\
&\qquad\quad accuracyRCKM' = 0 \land \\
&\qquad\quad RCKM' = \emptyset \land \\
&\qquad\quad refinedCKM' = \emptyset \land \\
&\qquad\quad rootRCKM' = \emptyset \land \\
&\qquad\quad refinedCKMsAccuracy' = \emptyset \\
&\quad \Leftrightarrow \exists\, RCKM' : \mathbb{P}\, decisionPathRCKM, && [definition : RefinedClinicalKnowledgeModel] \\
&\qquad rootRCKM' : decisionPathConditionRCKM, accuracyRCKM' : \mathbb{Z} \bullet \\
&\qquad\quad \exists\, refinedCKM' : RCKM \rightarrow CKM, \\
&\qquad\quad refinedCKMsAccuracy' : RCKM \rightarrow accuracyRCKM \bullet \\
&\qquad\qquad 0 \leq accuracyRCKM' \leq 100 \land \\
&\qquad\qquad RCKM' = \operatorname{dom} refinedCKM' \land \\
&\qquad\qquad accuracyRCKM' = (\textbf{let}\ pathsAcc == \{pathsAcc : \mathbb{Z} \mid RCKM' \neq \emptyset \land \\
&\qquad\qquad\qquad (\forall\, dp : decisionPathRCKM' \mid dp \in RCKM' \bullet pathsAcc = \\
&\qquad\qquad\qquad refinedCKMsAccuracy'(dp) + pathsAcc)\})/\#RCKM' \land \\
&\qquad\qquad rootRCKM' = rootRCKM \land \\
&\qquad\qquad accuracyRCKM' = 0 \land \\
&\qquad\qquad RCKM' = \emptyset \land \\
&\qquad\qquad refinedCKM' = \emptyset \land \\
&\qquad\qquad rootRCKM' = \emptyset \land \\
&\qquad\qquad refinedCKMsAccuracy' = \emptyset \\
&\quad \Leftrightarrow \quad \emptyset \in \mathbb{P}\, decisionPathRCKM \land && [one - point\ rule : 5 - times] \\
&\qquad \emptyset \in decisionPathConditionRCKM \land \\
&\qquad 0 \in \mathbb{Z} \land \\
&\qquad \emptyset \in RCKM \rightarrow CKM \land \\
&\qquad \emptyset \in RCKM \rightarrow accuracyRCKM
\end{aligned}
$$

4.3.3.2 R-CKM evolution consistency proof using simplification of preconditions and proving property composition

Calculating pre-conditions for R-CKM evolution: The pre-conditions of an operational schema represent a set of states, for which the outcome of the operations is properly defined. The pre-condition of an operation is another schema, obtained from a given operation, that hides components related to the state after operation and provides any output that results from an operation.

Definition 3: *For the operational schema "operation", the state of the system is modelled as "state" and the "output" is the list of outputs associated with the operation. Then, the following is equation represents the pre-condition of the schema.*

$$\text{pre } operation = \exists state' \bullet operation$$

Definition 3: Precondition of an operation

We establish a theorem (Theorem 2), which is based on the basic definition of the pre-condition schema (Definition 3), to calculate the pre-conditions for the operational schema *AddPathRCKM* (Schema 12).

Theorem 2 Pre-conditions calculation for R-CKM evolution operation

$$\text{pre } AddPathRCKM = \exists RefinedClinicalKnowledgeModel' \bullet AddPathRCKM$$

pre *AddPathRCKM*

$RefinedClinicalKnowledgeModel$
$dp_{pm}? : decisionPath$
$qualifiedAcc? : \mathbb{Z}$
$refinements? : \mathbb{F}\ RefinedTreatmentPlan$

$\exists RefinedClinicalKnowledgeModel'; \ rckmPath! : decisionPathRCKM \bullet AddPathRCKM$

In order to calculate the pre-condition, the predicate part of the theorem (Theorem 2) must be simplified by expanding all of the schemas. Moreover, after expansion of all possible schemas, the one-point rule plays a pivotal role in simplifying and proving the primed statements in the schema. Proof 2 is provided with instructive definitions at each evolving step of the schema. For brevity purposes, the proof does not discuss the pre-condition calculation in detail; however, we believe that the given explanation is sufficient to determine the pre-conditions for the *AddPathRCKM* operational schema.

Although the simplification process seems quite complex in terms of resolving all of the primed statements, by using set theory fundamental laws and the one-point rule it becomes straightforward. Additionally, it is interesting because it reveals new pre-condition predicates that were not known in advance. The primed predicates in Proof 2 are underlined (numbered 1-13); these required simplification to conclude the proof. These proofs are provided in a subsequent section.

Proof 2 Pre-condition calculation proof using one-point rule

$$\text{pre } \mathit{AddPathRCKM} \Leftrightarrow \tag{2.01}$$

$$\exists\, \mathit{RefinedClinicalKnowledgeModel}';\ \mathit{rckmPath}! : \mathit{decisionPathRCKM} \bullet [\mathit{def}.\text{pre } \mathit{AddPathRCKM}]\ \mathit{AddPathRCKM} \tag{2.02}$$

$$\Leftrightarrow$$

$$\exists\, \mathit{RefinedClinicalKnowledgeModel}';\ \mathit{rckmPath}! : \mathit{decisionPathRCKM} \bullet [\mathit{def}.\mathit{AddPathRCKM}] \tag{2.03}$$

$$\mathit{RCKM} \neq \emptyset \Rightarrow \mathit{head}\,(\text{dom } \mathit{dp}_{pm}?) = \mathit{rootRCKM} \wedge \tag{2.04}$$

$$\forall\, \mathit{pos} : \mathbb{N} \mid \mathit{pos} \in \text{dom } \mathit{refinements}? \bullet \mathit{pos} > 1 \wedge$$
$$\mathit{pos} \leq (\#(\text{dom } \mathit{dp}_{pm}?) + \#(\text{ran } \mathit{dp}_{pm}?)) \wedge \tag{2.05}$$

$$\text{ran}(\text{dom } \mathit{rckmPath}!) \subset \text{ran } \mathit{decisionPathConditionRCKM} \wedge \tag{2.06}$$

$$\text{ran}(\text{ran } \mathit{rckmPath}!) \subset \text{ran } \mathit{ConclusionRCKM} \wedge \tag{2.07}$$

$$(\text{ran}(\text{ran } \mathit{rckmPath}!) \cap \text{ran } \mathit{decisionPathConditionRCKM}) \subset$$
$$\text{ran } \mathit{decisionPathConditionRCKM} \wedge \tag{2.08}$$

$$0 \leq \mathit{decPathRCKMAccuracy}(\mathit{rckmPath}!) \leq 100 \wedge \tag{2.09}$$

$$\mathit{head}\,(\text{dom } \mathit{rckmPath}!) \notin \text{ran } \mathit{ConclusionRCKM} \cap \text{ran } \mathit{decisionPathConditionRCKM} \wedge \tag{2.10}$$

$$\exists\, \mathit{dp} : \mathit{decisionPathRCKM} \mid \mathit{dp} \in \mathit{RCKM} \bullet$$
$$\text{dom } \mathit{rckmPath}! = \text{dom } \mathit{dp} \setminus \mathit{last}\,(\text{dom } \mathit{dp}) \Rightarrow \mathit{last}\,(\text{dom } \mathit{dp}) = \text{ran } \mathit{rckmPath}! \wedge \tag{2.11}$$

$$\text{dom } \mathit{rckmPath}! = \exists\, p_{ckm} : \mathit{decisionPathCKM} \mid p_{ckm} \in \mathit{CKM} \bullet$$
$$\text{dom}(\mathit{p}_{pm}?) \cup \text{dom } p_{ckm} \wedge \tag{2.12}$$

$$\text{ran } \mathit{rckmPaht}! = \text{ran } \mathit{dp}_{pm}? \wedge \tag{2.13}$$

$$\forall\, r : \mathit{RefinedTreatmentPlan} \mid r \in \mathit{refinements}? \bullet$$
$$\mathit{rckmPath}! = \frown/\langle \{t_p : \mathit{TreatmentPlan} \bullet (1 \,..\, \text{dom } r, t_p)\} \upharpoonright \text{dom } \mathit{rckmPath}!,\ \text{ran } r,$$
$$\{t_p : \mathit{TreatmentPlan} \bullet (\text{dom } r + 1 \,..\, \#(\text{dom } \mathit{rckmPath}!), t_p)\} \upharpoonright \text{dom } \mathit{rckmPath}!\rangle \wedge \tag{2.14}$$

$$\mathit{decisionPathRCKM}' = \mathit{decisionPathRCKM} \cup \{\text{dom } \mathit{rckmPath}! \mapsto \text{ran } \mathit{rckmPath}!\} \wedge \tag{2.15}$$

$$\mathit{decisionPathConditionRCKM}' = \mathit{decisionPathConditionRCKM} \cup \text{dom } \mathit{rckmPath}! \wedge \tag{2.16}$$

$$\mathit{refinedTPlan}' = \mathit{refinedTPlan} \cup \mathit{refinements}? \wedge \tag{2.17}$$

$$\mathit{refinementsDecPath}' = \mathit{refinementsDecPath} \cup \{\mathit{refinements}? \mapsto \mathit{dp}_{pm}?\} \wedge \tag{2.18}$$

$$\mathit{ConclusionRCKM}' = \mathit{ConclusionRCKM} \cup \text{ran } \mathit{rckmPath}! \wedge \tag{2.19}$$

$$\mathit{decPathRCKMAccuracy}' = \mathit{decPathRCKMAccuracy} \cup$$
$$\{\mathit{rckmPath}! \mapsto \mathit{decPathRCKMAccuracy}(\mathit{rckmPath}!)\} \wedge \tag{2.20}$$

$$\mathit{accuracyRCKM}' = \frac{\mathit{accuracyRCKM} \times \#\mathit{RCKM} + \mathit{decPathRCKMAccuracy}'\,(\mathit{rckmPath}!)}{\#\mathit{RCKM}+1} \wedge \tag{2.21}$$

$$\#\mathit{RCKM}' = \#\mathit{RCKM} + 1 \wedge \tag{2.22}$$

$$\mathit{evidences}' = \mathit{evidences} \cup \mathit{decPathEvidences}(\mathit{dp}_{pm}?) \wedge \tag{2.23}$$

$$\mathit{decPathRCKMEvidences}' = \mathit{decPathRCKMEvidences} \cup$$
$$\{\mathit{rckmPath}! \mapsto \mathit{decPathEvidences}(\mathit{dp}_{pm}?)\} \wedge \tag{2.24}$$

$$\mathit{RCKM}' = \mathit{RCKM} \oplus \{\text{dom } \mathit{rckmPath}! \mapsto \text{ran } \mathit{rckmPath}!\} \wedge \tag{2.25}$$

$$\mathit{refinedCKM}' = \mathit{refinedCKM} \oplus \{\mathit{rckmPath}! \mapsto \mathit{CKM}\} \wedge \tag{2.26}$$

$$\mathit{rootRCKM}' = \mathit{rootRCKM} = \mathit{head}\,(\text{dom } \mathit{dp}_{pm}? \tag{2.27}$$

Continued.. 1 from Proof 2

$\Leftrightarrow$

$$\exists\, rckmPath! : decisionPathRCKM; \quad [def.RefinedClinicalKnowledgeModel'] \tag{2.28}$$
$$decisionPathConditionRCKM' : \mathbb{F}\, ConditionKMs; \tag{2.29}$$
$$ConclusionRCKM' : \mathbb{F}\, TreatmentPlan; \tag{2.30}$$
$$decisionPathRCKM' : ConditionKMs \nrightarrow TreatmentPlan; \tag{2.31}$$
$$decPathRCKMAccuracy' : decisionPathRCKM' \rightarrow accuracy; \tag{2.32}$$
$$evidences' : \mathbb{F}\, Evidences; \tag{2.33}$$
$$decPathRCKMEvidences' : decisionPathRCKM' \nrightarrow Evidences; \tag{2.34}$$
$$refinedTPlan' : \mathbb{F}\, RefinedTreatmentPlan; \tag{2.35}$$
$$RCKM' : \mathbb{F}\, decisionPathRCKM; \tag{2.36}$$
$$refinedCKM' : decisionPathRCKM' \rightarrow CKM; \tag{2.37}$$
$$refinementsDecPath' : RefinedTreatmentPlan \rightarrowtail decisionPath; \tag{2.38}$$
$$rootRCKM' : \mathrm{seq}\, Condition; \tag{2.39}$$
$$accuracyRCKM' : \mathbb{F}\, \mathbb{Z}; \tag{2.40}$$
$$refinedCKMsAccuracy' : RCKM' \rightarrow accuracy \bullet \tag{2.41}$$

$$^{(1)\cdot\cdot} decisionPathConditionRCKM' = \mathrm{dom}\, decisionPathRCKM' \wedge \tag{2.42}$$
$$^{(2)\cdot\cdot} ConclusionRCKM' = \mathrm{ran}\, decisionPathRCKM' \wedge \tag{2.43}$$
$$^{(3)\cdot\cdot} (\mathrm{ran}\, ConclusionRCKM' \cap \mathrm{ran}\, decisionPathConditionRCKM') \subset \mathrm{ran}\, decisionPathConditionRCKM' \wedge \tag{2.44}$$
$$^{(4)\cdot\cdot} head\,(decisionPathConditionRCKM') \notin \mathrm{ran}\, ConclusionRCKM' \cap \mathrm{ran}\, decisionPathConditionRCKM' \wedge \tag{2.45}$$
$$^{(5)\cdot\cdot} evidences' = \mathrm{ran}\, decPathRCKMEvidences' \wedge \tag{2.46}$$
$$^{(6)\cdot\cdot} refinedTPlan' = \mathrm{dom}\, refinementsDecPath' \wedge \tag{2.47}$$
$$^{(7)\cdot\cdot} 0 \leq accuracyRCKM' \leq 100 \wedge \tag{2.48}$$
$$^{(8)\cdot\cdot} RCKM' = \mathrm{dom}\, refinedCKM' \wedge \tag{2.49}$$
$$^{(9)\cdot\cdot} \forall\, dp : decisionPathRCKM' \mid dp \in RCKM' \bullet head\,(\mathrm{dom}\, dp) \notin \mathrm{ran}\, ConclusionRCKM' \cap \mathrm{ran}\, decisionPathConditionRCKM' \wedge \tag{2.50}$$
$$^{(10)\cdot\cdot} \exists\, dp : decisionPathRCKM' \fatsemi dp_1 : decisionPathRCKM' \mid dp, dp_1 \in RCKM' \bullet last\,(\mathrm{dom}\, dp) = \mathrm{ran}\, dp_1 \Leftrightarrow \mathrm{dom}\, dp_1 = \mathrm{dom}\, dp \setminus last\,(\mathrm{dom}\, dp) \wedge \tag{2.51}$$
$$^{(11)\cdot\cdot} accuracyRCKM' = (\textbf{let}\ pathsAcc == \{pathsAcc : \mathbb{Z} \mid RCKM' \neq \emptyset \wedge (\forall\, dp : decisionPathRCKM' \mid dp \in RCKM' \bullet pathsAcc = decPathRCKMAccuracy'(dp) + pathsAcc)\})/\#RCKM' \wedge \tag{2.52}$$
$$^{(12)\cdot\cdot} \forall\, p_{rckm} : decisionPathRCKM' \mid p_{rckm} \in RCKM' \bullet \exists\, p_{pm} : decisionPath, p_{ckm} : decisionPathCKM \mid p_{pm} \in PM \wedge p_{ckm} \in CKM \bullet \mathrm{dom}\, p_{rckm} = \mathrm{dom}\, p_{pm} \cup \mathrm{dom}\, p_{ckm} \wedge \tag{2.53}$$
$$^{(13)\cdot\cdot} RCKM \neq \emptyset \Rightarrow rootRCKM' = rootRCKM \wedge \tag{2.54}$$

Continued.. 2 from Proof 2

$$RCKM \neq \emptyset \Rightarrow head\,(\mathrm{dom}\, dp_{pm}?) = rootRCKM \wedge \quad (2.55)$$
$$\forall pos : \mathbb{N} \mid pos \in \mathrm{dom}\, refinements? \bullet pos > 1 \wedge$$
$$pos \leq (\#(\mathrm{dom}\, dp_{pm}?) + \#(\mathrm{ran}\, dp_{pm}?)) \wedge \quad (2.56)$$
$$\mathrm{ran}(\mathrm{dom}\, rckmPath!) \subset \mathrm{ran}\, decisionPathConditionRCKM \wedge \quad (2.57)$$
$$\mathrm{ran}(\mathrm{ran}\, rckmPath!) \subset \mathrm{ran}\, ConclusionRCKM \wedge \quad (2.58)$$
$$(\mathrm{ran}(\mathrm{ran}\, rckmPath!) \cap \mathrm{ran}\, decisionPathConditionRCKM) \subset$$
$$\mathrm{ran}\, decisionPathConditionRCKM \wedge \quad (2.59)$$
$$0 \leq decPathRCKMAccuracy(rckmPath!) \leq 100 \wedge \quad (2.60)$$
$$head\,(\mathrm{dom}\, rckmPath!) \notin \mathrm{ran}\, ConclusionRCKM \cap \mathrm{ran}\, decisionPathConditionRCKM \wedge \quad (2.61)$$
$$\exists dp : decisionPathRCKM \mid dp \in RCKM \bullet$$
$$\mathrm{dom}\, rckmPath! = \mathrm{dom}\, dp \setminus last\,(\mathrm{dom}\, dp) \Rightarrow last\,(\mathrm{dom}\, dp) = \mathrm{ran}\, rckmPath! \wedge \quad (2.62)$$
$$\mathrm{dom}\, rckmPath! = \exists p_{ckm} : decisionPathCKM \mid p_{ckm} \in CKM \bullet$$
$$\mathrm{dom}(p_{pm}?) \cup \mathrm{dom}\, p_{ckm} \wedge \quad (2.63)$$
$$\mathrm{ran}\, rckmPaht! = \mathrm{ran}\, dp_{pm}? \wedge \quad (2.64)$$
$$\forall r : RefinedTreatmentPlan \mid r \in refinements? \bullet$$
$$rckmPath! = \frown / \langle \{t_p : TreatmentPlan \bullet (1 \,..\, \mathrm{dom}\, r, t_p)\} \upharpoonright \mathrm{dom}\, rckmPath!,\ \mathrm{ran}\, r,$$
$$\{t_p : TreatmentPlan \bullet (\mathrm{dom}\, r + 1 \,..\, \#(\mathrm{dom}\, rckmPath!), t_p)\} \upharpoonright \mathrm{dom}\, rckmPath! \rangle \wedge \quad (2.65)$$

$$decisionPathRCKM' = decisionPathRCKM \cup \{\mathrm{dom}\, rckmPath! \mapsto \mathrm{ran}\, rckmPath!\} \wedge \quad (2.66)$$
$$decisionPathConditionRCKM' = decisionPathConditionRCKM \cup \mathrm{dom}\, rckmPath! \wedge \quad (2.67)$$
$$refinedTPlan' = refinedTPlan \cup refinements? \wedge \quad (2.68)$$
$$refinementsDecPath' = refinementsDecPath \cup \{refinements? \mapsto dp_{pm}?\} \wedge \quad (2.69)$$
$$ConclusionRCKM' = ConclusionRCKM \cup \mathrm{ran}\, rckmPath! \wedge \quad (2.70)$$
$$decPathRCKMAccuracy' = decPathRCKMAccuracy \cup$$
$$\{rckmPath! \mapsto decPathRCKMAccuracy(rckmPath!)\} \wedge \quad (2.71)$$
$$accuracyRCKM' = \frac{accuracyRCKM \times \#RCKM + decPathRCKMAccuracy'\,(rckmPath!)}{\#RCKM + 1} \wedge \quad (2.72)$$
$$\#RCKM' = \#RCKM + 1 \wedge \quad (2.73)$$
$$evidences' = evidences \cup decPathEvidences(dp_{pm}?) \wedge \quad (2.74)$$
$$decPathRCKMEvidences' = decPathRCKMEvidences \cup$$
$$\{rckmPath! \mapsto decPathEvidences(dp_{pm}?)\} \wedge \quad (2.75)$$
$$RCKM' = RCKM \oplus \{\mathrm{dom}\, rckmPath! \mapsto \mathrm{ran}\, rckmPath!\} \wedge \quad (2.76)$$
$$refinedCKM' = refinedCKM \oplus \{rckmPath! \mapsto CKM\} \wedge \quad (2.77)$$
$$rootRCKM' = rootRCKM = head\,(\mathrm{dom}\, dp_{pm}?) \quad (2.78)$$

Simplification of primed statements using logical proofs: This section describes the detailed steps used to prove the primed statements in Proof 2 (line 2.42 to 2.54). The primed statements are evolved using fundamental laws of set theory and deduction rules to obtain the simplified form. All proofs (Proof 5 - 15) are straightforward and instructions are provided for each logical statement.

We introduce the necessary definitions (if required) before each proof in order to clarify the logical steps in the corresponding and subsequent proofs. Proof 3 provides the simplification of the first prime statement in PProof 2 (line 2.42), which is concluded to the simplified statement

of the R-CKM model ((Axiom 3: line 11). In addition to the one-point rule (Definition 2), the following basic definitions (Definitions 4, 5) are used to deduce the final conclusion.

Definition 4: *For any two functions f and g, the dom property for the union is defined as follows;*

$$\mathrm{dom}(f \cup g) \Leftrightarrow \mathrm{dom}\, f \cup \mathrm{dom}\, g$$

Definition 4: *dom* over *union*

Definition 5: *For any two sets a and b, the set subtraction is formally defined as follows;*

$$a \setminus b = \{x \in a \mid x \notin b\}$$

Definition 5: Set subtraction

Proof 3 Simplification of primed statement-(1)

$$\mathit{decisionPathConditionRCKM}' = \mathrm{dom}\, \mathit{decisionPathRCKM}' \tag{3.01}$$

$$\mathit{decisionPathConditionRCKM} \cup \mathrm{dom}\, \mathit{rckmPath}! = \mathrm{dom}(\mathit{decisionPathRCKM} \cup \{\mathrm{dom}\, \mathit{rckmPath}! \mapsto \mathrm{ran}\, \mathit{rckmPath}!\}) \quad \mathit{Def}.2 : [\mathit{one-point\ rule}] \tag{3.02}$$

$$\mathit{decisionPathConditionRCKM} \cup \mathrm{dom}\, \mathit{rckmPath}! = \mathrm{dom}\, \mathit{decisionPathRCKM} \cup \mathrm{dom}\{\mathrm{dom}\, \mathit{rckmPath}! \mapsto \mathrm{ran}\, \mathit{rckmPath}!\} \mathit{Def}.4 : [\mathrm{dom}\, \mathit{property\ over} \cup] \tag{3.03}$$

$$\mathit{decisionPathConditionRCKM} \cup \mathrm{dom}\, \mathit{rckmPath}! = \mathrm{dom}\, \mathit{decisionPathRCKM} \cup \mathrm{dom}\, \mathit{rckmPath}! \quad [\mathrm{dom}\, \mathit{def}.] \tag{3.04}$$

$$\mathit{decisionPathConditionRCKM} = \mathrm{dom}\, \mathit{decisionPathRCKM} \quad \mathit{Def}.5 : [\mathit{Set\ subtraction}] \tag{3.05}$$

Proof 4 simplifies the primed statement in Proof 2 (line 2.43) to the refined statement of the R-CKM model (Axiom 3: line 12). Using the one-point rule (line 4.02), set subtraction, and *ran* properties (line 4.03- 4.05), the proof is easily concluded. The *ran* property for the union is defined as follows.

Definition 6: *For any two functions f and g, the ran property for the union is defined as follows;*

$$\mathrm{ran}(f \cup g) \Leftrightarrow \mathrm{ran}\, f \cup \mathrm{ran}\, g$$

Definition 6: *ran* over *union*

Proof 4 Simplification of primed statement-(2)

$$\mathit{ConclusionRCKM}' = \operatorname{ran} \mathit{decisionPathRCKM}' \tag{4.01}$$

$$\mathit{ConclusionRCKM} \cup \operatorname{ran} \mathit{rckmPath}! = \operatorname{ran}(\mathit{decisionPathRCKM} \cup \{\operatorname{dom} \mathit{rckmPath}! \mapsto \operatorname{ran} \mathit{rckmPath}!\}) \quad \mathit{Def}.2 : [\mathit{one-point\ rule}] \tag{4.02}$$

$$\mathit{ConclusionRCKM} \cup \operatorname{ran} \mathit{rckmPath}! = \operatorname{ran} \mathit{decisionPathRCKM} \cup \operatorname{ran}\{\operatorname{dom} \mathit{rckmPath}! \mapsto \operatorname{ran} \mathit{rckmPath}!\} \mathit{Def}.6 : [\operatorname{ran} \mathit{property\ over} \cup] \tag{4.03}$$

$$\mathit{ConclusionRCKM} \cup \operatorname{ran} \mathit{rckmPath}! = \operatorname{ran} \mathit{decisionPathRCKM} \cup \operatorname{ran} \mathit{rckmPath}! \quad [\operatorname{ran} \mathit{def}.] \tag{4.04}$$

$$\mathit{ConclusionRCKM} = \operatorname{ran} \mathit{decisionPathRCKM} \quad \mathit{Def}.5 : [\mathit{Set\ subtraction}] \tag{4.05}$$

Proof 5 simplifies the primed statement in Proof 2 (line 2.44) using the one-point rule (line 5.02), the definition of range (line 5.03 using Definition 6), and other laws and principles of set theory, which are described in the following definitions.

Definition 7: *For any two sets a and b, the following property holds;*

$$a \cup b = a \Leftrightarrow b \subset a$$

Definition 7: Union Properties

Definition 8: *Set intersection is distributive over A set union. For sets r, s, and t, the set intersection distribution over a union set can be defined as follows;*

$$r \cap (s \cup t) = (r \cap s) \cup (r \cap t)$$

Definition 8: Set intersection distribution law over union

Definition 9: *For sets a, b, and c, the following definition holds;*

$$a \cup b \subset c \Rightarrow (a \subset c \land b \subset c)$$

Definition 9: Set union and proper subset

Using the one-point rule (line 6.01) and definitions of basic set theory (lines 6.02 - 6.04), Proof 6 concludes the primed statement in Proof 2 (line 2.45) into the R-CKM model (Axiom 3: line 14).

Proof 5 Simplification of primed statement-(3)

$$(\text{ran}\ \mathit{ConclusionRCKM}' \cap \text{ran}\ \mathit{decisionPathConditionRCKM}') \subset \text{ran}\ \mathit{decisionPathConditionRCKM}' \tag{5.01}$$

$$(\text{ran}(\mathit{ConclusionRCKM} \cup \text{ran}\ \mathit{rckmPath}!) \cap \text{ran}(\mathit{decisionPathConditionRCKM} \cup \text{dom}\ \mathit{rckmPath}!)) \subset \text{ran}(\mathit{decisionPathConditionRCKM} \cup \text{dom}\ \mathit{rckmPath}!) \quad \mathit{Def}.2 : [\mathit{one-point\ rule}] \tag{5.02}$$

$$(\text{ran}\ \mathit{ConclusionRCKM} \cup \text{ran}(\text{ran}\ \mathit{rckmPath}!)) \cap (\text{ran}\ \mathit{decisionPathConditionRCKM} \cup \text{ran}(\text{dom}\ \mathit{rckmPath}!)) \subset \text{ran}\ \mathit{decisionPathConditionRCKM} \cup \text{ran}(\text{dom}\ \mathit{rckmPath}!) \quad \mathit{Def}.6 : [\text{ran}\ \mathit{property\ over}\ \cup] \tag{5.03}$$

$$((\text{ran}\ \mathit{ConclusionRCKM} \cup \text{ran}(\text{ran}\ \mathit{rckmPath}!)) \cap \text{ran}\ \mathit{decisionPathConditionRCKM}) \subset \text{ran}\ \mathit{decisionPathConditionRCKM} \quad \mathit{Def}.7 : [a \cup b = a \Leftrightarrow b \subset a] \tag{5.04}$$

$$(\text{ran}\ \mathit{ConclusionRCKM} \cap \text{ran}\ \mathit{decisionPathConditionRCKM}) \cup (\text{ran}(\text{ran}\ \mathit{rckmPath}!) \cap \text{ran}\ \mathit{decisionPathConditionRCKM}) \subset \text{ran}\ \mathit{decisionPathConditionRCKM} \quad \mathit{Def}.8 : [\mathit{Distribution\ law\ for}\ \cap] \tag{5.05}$$

$$(\text{ran}\ \mathit{ConclusionRCKM} \cap \text{ran}\ \mathit{decisionPathConditionRCKM}) \subset \text{ran}\ \mathit{decisionPathConditionRCKM} \wedge \text{ran}(\text{ran}\ \mathit{rckmPath}!) \cap \text{ran}\ \mathit{decisionPathConditionRCKM} \subset \text{ran}\ \mathit{decisionPathConditionRCKM} \quad \mathit{Def}.9 : [a \cup b \subset c \Rightarrow (a \subset c \wedge b \subset c)] \tag{5.06}$$

$$(\text{ran}\ \mathit{ConclusionRCKM} \cap \text{ran}\ \mathit{decisionPathConditionRCKM}) \subset \text{ran}\ \mathit{decisionPathConditionRCKM} \quad [a \wedge \mathit{true} \equiv a] \tag{5.07}$$

Proof 6 Simplification of primed statement-(4)

$$head\,(decisionPathConditionRCKM') \notin \operatorname{ran} ConclusionRCKM' \cap \operatorname{ran} decisionPathConditionRCKM' \tag{6.01}$$

$$head\,(decisionPathConditionRCKM \cup \operatorname{dom} rckmPath!) \notin \operatorname{ran}(ConclusionRCKM \cup \operatorname{ran} rckmPath!) \cap \operatorname{ran}(decisionPathConditionRCKM \cup \operatorname{dom} rckmPath!) \quad Def.2 : [one-point\ rule] \tag{6.02}$$

$$head\,(decisionPathConditionRCKM \cup \operatorname{dom} rckmPath!) \notin (\operatorname{ran} ConclusionRCKM \cup \operatorname{ran}(\operatorname{ran} rckmPath!)) \cap (\operatorname{ran} decisionPathConditionRCKM \cup \operatorname{ran}(\operatorname{dom} rckmPath!)) \quad Def.6 : [\operatorname{ran} property\ over \cup] \tag{6.03}$$

$$head\,(decisionPathConditionRCKM) \notin \operatorname{ran} ConclusionRCKM \cap \operatorname{ran} decisionPathConditionRCKM \quad Def.7 : [a \cup b = a \Leftrightarrow b \subset a] \tag{6.04}$$

Proof 7 concludes the primed statement in Proof 2 (line 2.46) into the R-CKM model (Axiom 3: line 15) using the one-point rule (line 7.02) and definitions of basic set theory (lines 7.03 - 7.05).

Proof 7 Simplification of primed statement-(5)

$$evidences' = \operatorname{ran} decPathRCKMEvidences' \tag{7.01}$$

$$(evidences \cup decPathEvidences(dp_{pm}?)) = \operatorname{ran}(decPathRCKMEvidences \cup \{rckmPath! \mapsto decPathEvidences(dp_{pm}?)\}) \quad Def.2 : [one-point\ rule] \tag{7.02}$$

$$(evidences \cup decPathEvidences(dp_{pm}?)) = \operatorname{ran} decPathRCKMEvidences \cup \operatorname{ran}\{rckmPath! \mapsto decPathEvidences(dp_{pm}?)\} \quad Def.6 : [\operatorname{ran} property\ over \cup] \tag{7.03}$$

$$(evidences \cup decPathEvidences(dp_{pm}?)) = \operatorname{ran} decPathRCKMEvidences \cup decPathEvidences(dp_{pm}?) \quad [\operatorname{ran} def.] \tag{7.04}$$

$$evidences = \operatorname{ran} decPathRCKMEvidences \quad Def.5 : [Set\ subtraction] \tag{7.05}$$

Using the one-point rule (line 8.02) and definitions of basic set theory (lines 8.03 - 8.05), Proof 8 concludes the primed statement in Proof 2 (line 2.47) into the R-CKM model (Axiom 3: line 16).

Proof 8 Simplification of primed statement-(6)

$$refinedTPlan' = \mathrm{dom}\ refinementsDecPath' \tag{8.01}$$

$$refinedTPlan \cup refinements? = \mathrm{dom}(refinementsDecPath \cup \{refinements? \mapsto dp_{pm}?\}) \quad Def.2 : [one-point\ rule] \tag{8.02}$$

$$refinedTPlan \cup refinements? = \mathrm{dom}\ refinementsDecPath \cup \mathrm{dom}\{refinements? \mapsto dp_{pm}?\} \quad Def.4 : [\mathrm{dom}\ property\ over\ \cup] \tag{8.03}$$

$$refinedTPlan \cup refinements? = \mathrm{dom}\ refinementsDecPath \cup refinements? \quad [\mathrm{dom}\ def.] \tag{8.04}$$

$$refinedTPlan = \mathrm{dom}\ refinementsDecPath \quad Def.5 : [Set\ subtraction] \tag{8.05}$$

Proof 9 concludes the primed statement in Proof 2 (line 2.48) into the R-CKM model (Schema 3: line 8). This proof is straightforward and its conclusion is reached by using the one-point rule (line 9.02, 9.09) and solving the inequalities with fundamental mathematics. The proof is logically decomposed into two parts (lines 9.03-9.07 and lines 9.08-9.11). Each part is proven separately and the final statement is concluded (line 9.12).

The remaining proofs (Proof 10-Proof 15) use the same pattern of logical proofs to simplify the remaining primed statements of Proof 2 (line 2.49-line 2.54). Each step in the proofs is provided with instructive definitions, and necessary definitions are included where explanation is required.

Proof 9 Simplification of primed statement-(7)

$$0 \leq accuracyRCKM' \leq 100 \tag{9.01}$$
$$\Leftrightarrow \tfrac{accuracyRCKM' \geq 0}{P_1} \wedge \tfrac{accuracyRCKM' \leq 100}{P_2} \tag{9.02}$$
$$accuracyRCKM' \geq 0 \qquad [Let\ consider\ P_1] \tag{9.03}$$
$$\tfrac{accuracyRCKM \times \#RCKM + decPathRCKMAccuracy(rckmPath!)}{\#RCKM+1} \geq 0 \qquad Def.2 : [one-point\ rule] \tag{9.04}$$
$$accuracyRCKM \times \#RCKM + decPathRCKMAccuracy(rckmPath!) \geq 0 \qquad [multiplication] \tag{9.05}$$
$$accuracyRCKM \times \#RCKM \geq 0 \qquad [a+b \geq 0 \wedge b \geq 0 \Rightarrow a \geq 0] \tag{9.06}$$
$$accuracyRCKM \geq 0 \qquad [Division] \tag{9.07}$$

$$accuracyRCKM' \leq 100 \qquad [Let\ consider\ P_2] \tag{9.08}$$
$$\tfrac{accuracyRCKM \times \#RCKM + decPathRCKMAccuracy(rckmPath!)}{\#RCKM+1} \leq 100 \qquad Def.2 : [one-point\ rule] \tag{9.09}$$
$$accuracyRCKM \times \#RCKM + decPathRCKMAccuracy(rckmPath!) \leq 100 \times (\#RCKM + 1) \qquad [multiplication] \tag{9.09}$$
$$accuracyRCKM \times \#RCKM \leq 100 \times (\#RCKM + 1) \qquad [a.x + y \leq c.(a+1) \wedge y \leq c \Rightarrow a.x \leq c.(a+1)] \tag{9.10}$$
$$accuracyRCKM \leq 100 \qquad [a.x \leq c.(a+1) \Rightarrow x \leq c] \tag{9.11}$$
$$0 \leq accuracyRCKM \leq 100 \qquad [P_1\ and\ P_2\ proofs] \tag{9.12}$$

Definition 10: *The union ($\cup$) of two functions is not always a function. However, $\oplus$ is the same as a union but ensures that combinations of the two functions are also a function. For two functions f and g, $\oplus$ is defined as follows:*

$$f \oplus g = (\mathrm{dom}\, g \lhd f) \cup g$$

For functions f and g, the dom property for $\oplus$ is defined as follows;

$$\mathrm{dom}(f \oplus g) \Leftrightarrow \mathrm{dom} f \oplus \mathrm{dom}\, g$$

Definition 10: *dom* property over $\oplus$

Proof 10 Simplification of primed statement-(8)

$$RCKM' = \mathrm{dom}\ refinedCKM' \tag{10.01}$$

$$RCKM \oplus \{\mathrm{dom}\ rckmPath! \mapsto \mathrm{ran}\ rckmPath!\} = \mathrm{dom}(refinedCKM \oplus \{rckmPath! \mapsto CKM\}) \quad Def.2 : [one - point\ rule] \tag{10.02}$$

$$RCKM \oplus \{\mathrm{dom}\ rckmPath! \mapsto \mathrm{ran}\ rckmPath!\} = \mathrm{dom}\ refinedCKM \oplus \mathrm{dom}\{rckmPath! \mapsto CKM\} \quad Def.10 : [\mathrm{dom}\ over \oplus] \tag{10.03}$$

$$RCKM \oplus \{\mathrm{dom}\ rckmPath! \mapsto \mathrm{ran}\ rckmPath!\} = \mathrm{dom}\ refinedCKM \oplus rckmPath! \quad [\mathrm{dom}\ def.] \tag{10.04}$$

$$RCKM \oplus rckmPath! = \mathrm{dom}\ refinedCKM \oplus rckmPath! \quad [Simplification] \tag{10.05}$$

$$RCKM = \mathrm{dom}\ refinedCKM \quad Def.5 : [Set\ subtraction] \tag{10.06}$$

Proof 11 Simplification of primed statement-(9)

$$\forall\, dp : decisionPathRCKM' \mid dp \in RCKM' \bullet \mathrm{head}\,(\mathrm{dom}\ dp) \notin \mathrm{ran}\ ConclusionRCKM' \cap \mathrm{ran}\ decisionPathConditionRCKM' \tag{11.01}$$

$$\forall\, dp : (decisionPathRCKM \cup \{\mathrm{dom}\ rckmPath! \mapsto \mathrm{ran}\ rckmPath!\}) \mid dp \in (RCKM \oplus \{\mathrm{dom}\ rckmPath! \mapsto \mathrm{ran}\ rckmPath!\}) \bullet \mathrm{head}\,(\mathrm{dom}\ dp) \notin \mathrm{ran}(ConclusionRCKM \cup \mathrm{ran}\ rckmPath!) \cap \mathrm{ran}(decisionPathConditionRCKM \cup \mathrm{dom}\ rckmPath!) \quad Def.2 : [one - point\ rule] \tag{11.02}$$

$$\forall\, dp : decisionPathRCKM \mid dp \in RCKM \bullet \mathrm{head}\,(\mathrm{dom}\ dp) \notin \mathrm{ran}\ ConclusionRCKM \cap \mathrm{ran}\ decisionPathConditionRCKM \wedge \mathrm{head}\,(\mathrm{dom}\{\mathrm{dom}\ rckmPath! \mapsto \mathrm{ran}\ rckmPath!\}) \notin \mathrm{ran}(ConclusionRCKM \cup \mathrm{ran}\ rckmPath!) \cap \mathrm{ran}(decisionPathConditionRCKM \cup \mathrm{dom}\ rckmPath!) \quad [\forall\ simplification] \tag{11.03}$$

$$\forall\, dp : decisionPathRCKM \mid dp \in RCKM \bullet \mathrm{head}\,(\mathrm{dom}\ dp) \notin \mathrm{ran}\ ConclusionRCKM \cap \mathrm{ran}\ decisionPathConditionRCKM \wedge \mathrm{head}\,(\mathrm{dom}\ rckmPath!) \notin \mathrm{ran}\ ConclusionRCKM \cup \mathrm{ran}(\mathrm{ran}\ rckmPath!) \cap \mathrm{ran}\ decisionPathConditionRCKM \cup \mathrm{ran}(\mathrm{dom}\ rckmPath!) \quad Def.4, 6 : [\mathrm{dom}\ def.\ and\ \mathrm{ran}\ over \cup] \tag{11.04}$$

$$\forall\, dp : decisionPathRCKM \mid dp \in RCKM \bullet \mathrm{head}\,(\mathrm{dom}\ dp) \notin \mathrm{ran}\ ConclusionRCKM \cap \mathrm{ran}\ decisionPathConditionRCKM \wedge \mathrm{head}\,(\mathrm{dom}\ rckmPath!) \notin \mathrm{ran}\ ConclusionRCKM \cap \mathrm{ran}\ decisionPathConditionRCKM \quad Def.7 : [a \cup b = a \Rightarrow b \subset a] \tag{11.05}$$

$$\forall\, dp : decisionPathRCKM \mid dp \in RCKM \bullet \mathrm{head}\,(\mathrm{dom}\ dp) \notin \mathrm{ran}\ ConclusionRCKM \cap \mathrm{ran}\ decisionPathConditionRCKM \quad [a \wedge true \equiv a] \tag{11.06}$$

Definition 11: *Modus ponens, or implication elimination, is a simple argument form and rule inference in logic. For predicates p and q, the modus ponens can be formally represented as follows;*

$$p \Rightarrow q, q \vdash p$$

Definition 11: Modus ponens

Proof 12 Simplification of primed statement-(10)

$$\exists\, dp : decisionPathRCKM' \fatsemi dp_1 : decisionPathRCKM' \mid dp, dp_1 \in RCKM' \bullet \\ \operatorname{dom} dp_1 = \operatorname{dom} dp \setminus last\,(\operatorname{dom} dp) \Rightarrow last\,(\operatorname{dom} dp) = \operatorname{ran} dp_1 \tag{12.01}$$

$$\exists\, dp : (decisionPathRCKM \cup \{\operatorname{dom} rckmPath! \mapsto \operatorname{ran} rckmPath!\}) \fatsemi \\ dp_1 : (decisionPathRCKM \cup \{\operatorname{dom} rckmPath! \mapsto \operatorname{ran} rckmPath!\}) \mid \\ dp, dp_1 \in (RCKM \oplus \{\operatorname{dom} rckmPath! \mapsto \operatorname{ran} rckmPath!\}) \bullet \\ \operatorname{dom} dp_1 = \operatorname{dom} dp \setminus last\,(\operatorname{dom} dp) \Rightarrow last\,(\operatorname{dom} dp) = \operatorname{ran} dp_1 \; Def.2 : [one - point\ rule] \tag{12.02}$$

$$\exists\, dp : decisionPathRCKM \mid dp \in RCKM \bullet \\ \operatorname{dom} rckmPath! = \operatorname{dom} dp \setminus last\,(\operatorname{dom} dp) \Rightarrow last\,(\operatorname{dom} dp) = \operatorname{ran} rckmPath! \\ \Rightarrow \\ \exists\, dp : decisionPathRCKM \fatsemi dp_1 : decisionPathRCKM \mid dp, dp_1 \in RCKM \bullet \\ \operatorname{dom} dp_1 = \operatorname{dom} dp \setminus last\,(\operatorname{dom} dp) \Rightarrow last\,(\operatorname{dom} dp) = \operatorname{ran} dp_1 \quad [\exists\, expansion] \tag{12.03}$$

$$\exists\, dp : decisionPathRCKM \fatsemi dp_1 : decisionPathRCKM \mid dp, dp_1 \in RCKM \bullet \\ \operatorname{dom} dp_1 = \operatorname{dom} dp \setminus last\,(\operatorname{dom} dp) \Rightarrow last\,(\operatorname{dom} dp) = \operatorname{ran} dp_1 \;\; Def.11 : [Modus\ ponens] \tag{12.04}$$

Proof 13 Simplification of primed statement-(11)

$$accuracyRCKM' = (\textbf{let}\ pathsAcc == \{pathsAcc : \mathbb{Z} \mid RCKM' \neq \emptyset \wedge \\ (\forall\, dp : decisionPathRCKM' \mid dp \in RCKM' \bullet pathsAcc = \\ decPathRCKMAccuracy'(dp) + pathsAcc)\})/\#RCKM' \tag{13.01}$$

$$\tfrac{accuracyRCKM \times \#RCKM + decPathRCKMAccuracy(rckmPath!)}{\#RCKM+1} = \\ (\textbf{let}\ pathsAcc == \{pathsAcc : \mathbb{Z} \mid (RCKM \oplus \{\operatorname{dom} rckmPath! \mapsto \operatorname{ran} rckmPath!\}) \neq \emptyset \wedge \\ (\forall\, dp : (decisionPathRCKM \cup \{\operatorname{dom} rckmPath! \mapsto \operatorname{ran} rckmPath!\}) \mid dp \in \\ (RCKM \oplus \{\operatorname{dom} rckmPath! \mapsto \operatorname{ran} rckmPath!\}) \bullet \\ pathsAcc = (decPathRCKMAccuracy(dp) + pathsAcc) \\ + decPathRCKMAccuracy(rckmPath!))\})/(\#RCKM + 1) \quad Def.2 : [one - point\ rule] \tag{13.02}$$

$$\tfrac{accuracyRCKM \times \#RCKM + decPathRCKMAccuracy(rckmPath!)}{\#RCKM+1} = \\ ((\textbf{let}\ pathsAcc == \{pathsAcc : \mathbb{Z} \mid RCKM \neq \emptyset \wedge \\ (\forall\, dp : decisionPathRCKM \mid dp \in RCKM \bullet \\ pathsAcc = decPathRCKMAccuracy(dp) + pathsAcc)\}))+ \\ decPathRCKMAccuracy(rckmPath!))/(\#RCKM + 1) \quad [\forall\, simplification] \tag{13.03}$$

$$accuracyRCKM \times \#RCKM + decPathRCKMAccuracy(rckmPath!) = \\ (\textbf{let}\ pathsAcc == \{pathsAcc : \mathbb{Z} \mid RCKM \neq \emptyset \wedge \\ (\forall\, dp : decisionPathRCKM \mid dp \in RCKM \bullet pathsAcc = \\ decPathRCKMAccuracy(dp) + pathsAcc)\}))+ \\ decPathRCKMAccuracy(rckmPath!) \quad [multiplication] \tag{13.04}$$

$$accuracyRCKM \times \#RCKM = \\ (\textbf{let}\ pathsAcc == \{pathsAcc : \mathbb{Z} \mid RCKM \neq \emptyset \wedge \\ (\forall\, dp : decisionPathRCKM \mid dp \in RCKM \bullet pathsAcc = \\ decPathRCKMAccuracy(dp) + pathsAcc)\})) \quad [subtraction] \tag{13.05}$$

$$accuracyRCKM = \\ (\textbf{let}\ pathsAcc == \{pathsAcc : \mathbb{Z} \mid RCKM \neq \emptyset \wedge \\ (\forall\, dp : decisionPathRCKM \mid dp \in RCKM \bullet pathsAcc = \\ decPathRCKMAccuracy(dp) + pathsAcc)\}))/\#RCKM \quad [division] \tag{13.06}$$

Proof 14 Simplification of primed statement-(12)

$$\forall p_{rckm} : decisionPathRCKM' \mid p_{rckm} \in RCKM' \bullet \exists p_{pm} : decisionPath, p_{ckm} : decisionPathCKM \mid p_{pm} \in PM \land p_{ckm} \in CKM \bullet \operatorname{dom} p_{rckm} = \operatorname{dom} p_{pm} \cup \operatorname{dom} p_{ckm} \quad (14.01)$$

$$\forall p_{rckm} : (decisionPathRCKM \cup \{\operatorname{dom} rckmPath! \mapsto \operatorname{ran} rckmPath!\}) \mid p_{rckm} \in (RCKM \oplus \{\operatorname{dom} rckmPath! \mapsto \operatorname{ran} rckmPath!\}) \bullet \exists p_{pm} : decisionPath, p_{ckm} : decisionPathCKM \mid p_{pm} \in PM \land p_{ckm} \in CKM \bullet \operatorname{dom} p_{rckm} = \operatorname{dom} p_{pm} \cup \operatorname{dom} p_{ckm} \quad Def.2 : [one - point\ rule] \quad (14.02)$$

$$\forall p_{rckm} : decisionPathRCKM \mid p_{rckm} \in RCKM \bullet \exists p_{pm} : decisionPath, p_{ckm} : decisionPathCKM \mid p_{pm} \in PM \land p_{ckm} \in CKM \bullet \operatorname{dom} p_{rckm} = \operatorname{dom} p_{pm} \cup \operatorname{dom} p_{ckm} \land \exists p_{ckm} : decisionPathCKM \mid p_{ckm} \in CKM \bullet \operatorname{dom} rckmPath! = \operatorname{dom}(p_{pm}?) \cup \operatorname{dom} p_{ckm} \quad [\forall\ expansion] \quad (14.03)$$

$$\forall p_{rckm} : decisionPathRCKM \mid p_{rckm} \in RCKM \bullet \exists p_{pm} : decisionPath, p_{ckm} : decisionPathCKM \mid p_{pm} \in PM \land p_{ckm} \in CKM \bullet \operatorname{dom} p_{rckm} = \operatorname{dom} p_{pm} \cup \operatorname{dom} p_{ckm} \quad [a \land true \equiv a] \quad (14.04)$$

Proof 15 Simplification of primed statement-(13)

$$RCKM \neq \emptyset \Rightarrow rootRCKM' = rootRCKM \quad (15.01)$$

$$RCKM \neq \emptyset \Rightarrow head\,(\operatorname{dom} dp_{pm}?) = rootRCKM \quad Def.2 : [one - point\ rule] \quad (15.02)$$

$$RCKM \neq \emptyset \Rightarrow rootRCKM = rootRCKM \quad [Substitution] \quad (15.03)$$

4.4 Summary

This chapter explored formal verification of the knowledge acquisition and validation process proposed in three phase process model. Using Z notations, the main models (PM, CKM, and R-CKM) and processes (validation process) in three phase process model were formally represented as Z axioms and schemas. The validation process was described as an operational schema and corresponding state change in R-CKM model was represented as a state schema. By establishing the initialization theorem and precondition theorem for validation process (operational schemas), the chapter concluded the consistency of the three phase validation process with proofs for all primed-statements in the theorems. Details of the inconsistencies found using formal verification is elaborated in the Results and Evaluation (Chapter 6). Furthermore, to incorporate the verifi-

cation processes in real development of the CDSS projects, Chapter 5 exploits the alignment and usage of the formal verification in the context of unified processes.

Chapter 5

Implementation of Three-phase model and Formal Verification using CDSS development framework

In this chapter, we will introduce a development framework for CDSS - so called SmartCDSS-DF which facilitate adoption of the proposed knowledge acquisition and validation process and the formal verification in context of unified development process required for the design and development of CDSS. SmartCDSS-DF is influenced from HIS-DF [20] in the context of using RUP with additional method plug-in of RM-ODP. HIS-DF is extending GCM core principal for healthcare domain and applying RUP artifacts and processes with RM-ODP and HL7 as additional method plug-ins(necessary for semantic interoperability). SmartCDSS-DF following all processes adopted in HIS-DF while separating the activities and processes into two different pools and introducing new processes that are specifically targeted for evolutionary CDSS. Figure 5.1 highlights the main process and activities of SmartCDSS-DF.

5.1 Overview of SmartCDSS-DF

SmartCDSS-DF works on three basic principles. First, it separates the roles, activities, and deliverables with clear distinctions in the development process and aligns and integrates these into a unified process model. Second, it exploits existing software architectural approaches and extends and integrates their components into a unified and comprehensive framework to overcome the limitations of existing approaches; it also supports the requirements of evolutionary CDSS. Third, it introduces a formal mechanism for the specialized requirements of CDSS (such as validation of knowledge) using the tailoring mechanisms of the existing approaches (e.g., the Methods Plugin of RUP). Based on these principles, the proposed framework is comprised of the following

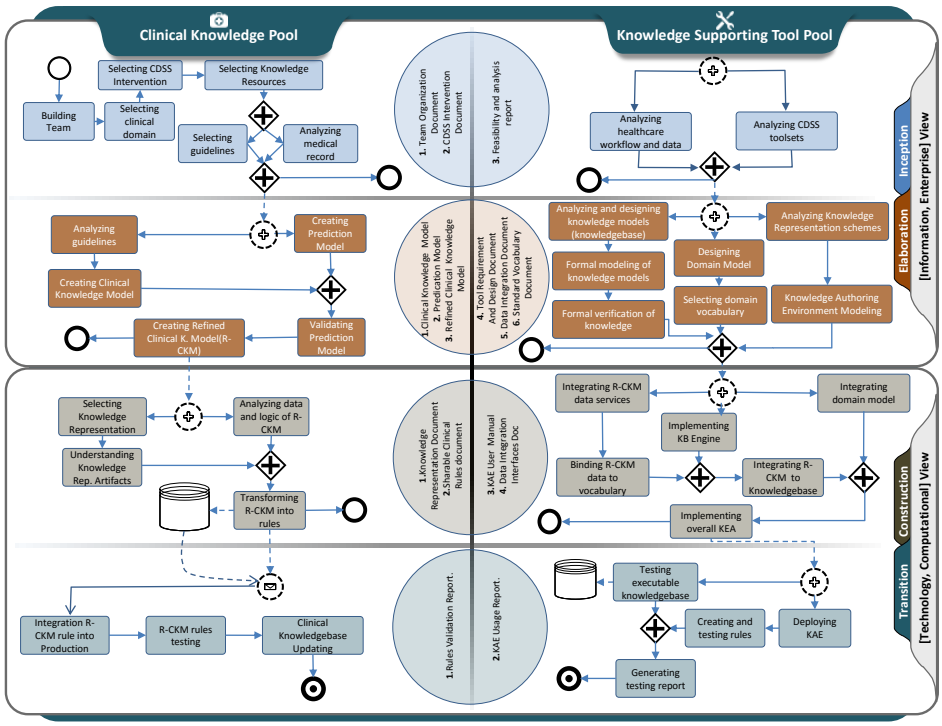

Figure 5.1: SmartCDSS-DF: Development framework for clinical decision support system components.

Introduce and separate conceptual and technical processes pools: Due to the unique requirements of CDSS, which has stakeholders with diverse capabilities, SmartCDSS-DF processes are divided into two distinct pools. The clinical knowledge pool supports activities related to clinical knowledge acquisition, validation, and representation. The knowledge supporting tool pool involves the fundamental activities required to design, develop, and implement all of the tools needed for CDSS to work in a real healthcare setup.

Tailoring the RUP to support specialized requirements: RUP processes are specialized and configured for the development of CDSS. Method Plugin is a tailoring mechanism in RUP that introduces processes that target the specialized requirements of CDSS. HIS-DF [20] is an example framework that exploits the Method Plugin mechanism in RUP and plugs in the specialized processes that are necessary for the development of semantically interoperable HIS. In SmartCDSS-DF, RM-ODP views and artifacts are used to model various compo-

nents of the system to support stakeholders with diverse capabilities. Most of the artifacts in the knowledge supporting tool pool are aligned and mapped to default artifacts in RUP; however, for the clinical knowledge pool, some knowledge artifacts may not have direct mapping to the RUP artifacts. In this case, RM-ODP provides flexibility to represent specialized artifacts using different views. Moreover, we introduce formal methods to provide formal representation and proofs for specialized requirements.

Formal validation and verification: Because validation and verification of knowledge are necessary requirements, a mathematical model is proposed to formalize the validation process and prove its consistencies with appropriate theorems.

5.2 SmartCDSS-DF: Separation of conceptual and technical processes

To deal with the complexity of CDSS in terms of its requirements, its evolutionary nature, and the fact that it must organize different stakeholders with different capabilities, SmartCDSS-DF uses two distinct and separate process pools. This separation of concerns mainly focuses on the conceptual and technical aspects of the CDSS. The conceptual aspects are covered in the ”clinical knowledge” pool, which unfolds relevant processes to represent the requirements of the main ingredient (i.e., knowledge) of the CDSS. The technical aspects are represented as the ”knowledge supporting tool” pool, which includes the processes that are used to model and implement the environment that facilitates the authoring and maintenance of the knowledge and provides integration with the healthcare workflow.

5.2.1 Clinical knowledge pool

The clinical knowledge pool comprises the core processes of the CDSS development environment. These processes are pivotal for domain modeling and provide baseline techniques for acquiring knowledge from diverse sources. They also incorporate the proper validation approaches for the refined knowledge use in CDSS. The clinical knowledge pool provides a set of processes that acquire knowledge from data using machine-learning approaches (data-driven), validate knowl-

edge from clinical guidelines (CPGs: evidence-based), and produce refined (guideline-enabled: hybrid) knowledge. It is important to note that the processes in the clinical knowledge model pool reflect the guideline-enabled hybrid knowledge acquisition; here, one can easily modify existing processes or introduce plugins for new processes to reflect other approaches. For example, for approaches that are only data-driven (as indicated in [26]), the "creating prediction model" process can be used directly and integrates the transformed rules into production. Based on our experiences and the many advantages of guideline-enabled hybrid knowledge acquisition, we will focus on the processes of the guideline-enabled hybrid approach [71]. The processes for this hybrid approach are classified into three broad categories: knowledge acquisition, knowledge validation, and knowledge transformation [71]. An additional category (i.e., test-based knowledge validation) includes processes to evaluate the knowledge after deployment. Below are brief descriptions of the process categories and their possible outcomes. For more detailed information about these processes and their corresponding outcomes, readers may consult our work presented in [71] for the first three categories, which are represented as a three-phase development process model.

1. Knowledge acquisition: This includes processes used to identify knowledge sources, analyze these sources for targeting objectives, and create a common representation model for further processing. The clinical knowledge model (CKM), which is a formal tree representation, is derived from clinical guidelines to determine the intended knowledge for targeted CDSS intervention. For example, CKM (presented in [71] and shown in Figure 6.1) is intended to represent the recommended treatment plan that is derived from NCCN guidelines [14] for oral cavity cancer (head and neck cancer). Similarly, a patients medical records are considered to be a substantial source of knowledge, and appropriate machine learning algorithms are used to acquire the prediction model (PM) as a potential knowledge model. For example, based on the intended targeted objective of CDSS intervention, PM is created from patient data using the decision tree approach [71] (shown in Figure 3.3).

2. Knowledge validation: Validation of the PM is performed during model creation using 10-fold cross validation. In order to derive refined knowledge (i.e., a refined clinical knowledge model (R-CKM)), PM is required to align and conform to CKM. The derivation of R-CKM requires a proper validation mechanism that ensures that the final knowledge model is valid

in terms of conformance to the clinical guidelines. Moreover, for any decision path that is not fully conformed to the CKM, it is necessary for the validation framework to have an appropriate reason for keeping the decision path in the R-CKM. Knowledge validation comprises processes related to validating PM from CKM and creating the final knowledge model (the R-CKM). Maintaining the intended objectives of validation, criteria-based validation is used (shown in Figure 3.4) to evaluate PM in the context of CKM and derive R-CKM [71]. The set of criteria is defined by domain experts in order to fulfill guideline conformance requirements and reflect some of the local evidence available in the patient data. In our case study for oral cavity R-CKM creation, the set of criteria defined by domain experts to keep the decision path of PM in final R-CKM was based on two primary (compulsory) criteria: i) to pass the minimum acceptable accuracy for the decision path (this was set to 50%) and ii) the treatment plan provided by the decision path must not conflict with the guidelines. There are also two alternate criteria: iii) the decision path should conform to any decision path in CKM and iv) in case criteria iii) is not fulfilled, the decision path must have sufficient evidence to prove its effectiveness and its necessity for inclusion in R-CKM. Based on the created PM, and after evaluation from CKM with these four criteria, the R-CKM was created for an oral cavity treatment plan [71] (shown in Figure 6.2).

3. Knowledge transformation: The ultimate outcome of the clinical knowledge pool is CIGs; this should be integrated into the healthcare workflow and follow a standard shareable representation. Knowledge transformation includes processes that guide the domain experts to transform the final R-CKM into shareable CIGs. In the current work, the final CIG is represented in HL7 Arden Syntax as medical logic modules (MLMs) using ”multiple dependent MLMs with root and sub-MLMs” approach. A summary of the three different approaches for transforming R-CKM into an MLM is depicted in Figure 6.3. Further details are provided in [71]. According to this approach, four MLMs are created; one root MLM works as a controller for three sub-MLMs.

4. Test-base knowledge validation: It includes the processes to properly test the transformed knowledge CIGs - as integral part in the form of MLMs using patient data. The test-base knowledge validation for the guideline-enable knowledge acquisition is preformed in two

phases;

(a) Inspecting and testing CIGs (i.e., the MLMs): In this phase, the MLMs are passed through two steps. In the first step, domain experts inspect the internal logic to try and locate any possible errors made with respect to the logic consistency and the target knowledge representation scheme. In order to reduce the chance of errors, which may result from a lack of exhaustive checking [24], a second step is performed. This tests the CIGs using patient data. In the current work, in order to reduce errors with respect to the knowledge representation scheme (MLM in HL7 Arden Syntax), a knowledge engineer has prepared an implementation guide that shows how to use the various artifacts of the HL7 Arden Syntax. The implementation guide provides relevant concepts and their semantics from a set of comprehensive specifications. It also assists domain experts in creating and inspecting the newly created or updated MLMs. Furthermore, in order to reduce the amount of test cases used for each testing step, a constrained-based clinical condition combinations approach [34] was adopted to test the MLMs.

(b) Evaluation of the R-CKM for quality: R-CKM-based knowledge is acquired from patient data using PM (after a proper validation process) to conform to the guidelines (represented as CKM). In this phase, the R-CKM performance (accuracy) is evaluated and compared to the PM on a common patient dataset. The difference in the performance between R-CKM and PM dictates the gap between real practices and guideline-based practices. As a result, the R-CKM with the smallest difference in performance relative to PM is considered to be more valuable for integration with real practice workflows. For the oral cavity, R-CKM was evaluated by using four MLMs on a common test dataset of 739 patients; this had a performance of 53.0%, yielding only a small difference compared to the performance of PM (59.0%) [71].

5.2.2 Knowledge supporting tool pool

The knowledge supporting tool pool includes processes that are used to model the clinical domain and design and develop toolsets to support the transformation of R-CKM into a shareable and executable format (CIGs). Moreover, these processes support standard design principles and avail-

able healthcare standards, which make it easier to integrate the CIGs into the healthcare workflow. Knowledge supporting tool pool processes must be aligned with the outcomes of the corresponding processes in the clinical knowledge pool. The alignment of both pools helps in establishing collaboration between different stakeholders (domain experts, knowledge engineers, and developers) and reduces the chance that the final toolsets will deviate from the real requirements. The knowledge supporting tool pool also contains processes used to represent the special requirements of CDSS by using formal methods to ensure consistency. After investigation of the CDSS requirements, the processes can be organized into the following broad categories.

1. System modeling processes: System modeling processes cover the modeling and design features of toolsets that are needed by domain experts to process the clinical knowledge. Moreover, they encapsulate the conceptual view of the system usage and the design view for implementation. Most of the processes in this category are common to the development processes of traditional software systems in other domains.

2. Standardization processes: Standardization processes are intended to enable the knowledge supporting tools to support shareable knowledge, be easily integrable, and possess reasonable clinical semantics with universal interpretation. These intentions lead towards the adoption of readily available healthcare standards such as HL7 Arden Syntax [42] (for shareable knowledge), HL7 vMR [44] (for easy integration), and vocabulary standards (SNOMED CT) [72] (for standard content). In addition, to adopt healthcare standards, mapping processes are also required in order to map different standard models and support localization requirements. For example, our work with the semantic reconciliation model (SRM) [73] demonstrates multi-model mappings for the development of an easy to use and flexible authoring environment that allows physicians to use localized concepts and produce shareable knowledge.

3. Formalism processes: Formalism processes recognize the specialized requirements of the CDSS, such as the validation and verification of knowledge. Formal validation and verification processes ensure validity and internal consistencies of the knowledge, which has a direct association with the clinical goals and great implications on the final execution of

knowledge for the targeted goal [74]. Based on the importance of knowledge verification for CDSS, separate sections are dedicated to the formalism approach.

4. Architectural processes: Architectural processes represent activities used to identify the appropriate architectures for CDSS components. Different architectural choices can be applied including the stand-alone system, integrated system, standard-based system, and service model [46]. The architecture selection is mainly based on the knowledge model [74] scheme of the CDSS. Therefore, the knowledge representation scheme should be carefully selected so that it complies with most of the desired requirements of the system. A. Wright and D. F. Sittig provided a four-phase model to evaluate architectural approaches for CDSS; this model is based on a set of desirable features [2]. In this work, we combined two architectural approaches by leveraging a service model (defining HL7 vMR-based service interfaces) to deliver the knowledge services; the knowledge is represented in a standard format (using HL7 Ardent Syntax).

5. Implementation processes: Implementation processes include processes used to develop knowledge supporting tools as well as related modules of the CDSS using the appropriate technology, which covers the necessary components of the selected architecture. These processes intend to develop a knowledge execution engine, an easy to use knowledge-authoring environment, and the integration components necessary to expose the knowledge services.

6. Deployment processes: Deployment processes consider the deployment of various components of the CDSS in a healthcare setup and incorporate the testing environment used for testing the corresponding component. The knowledge-authoring environment is tested for rule creation, and the created rules are validated using real data from the healthcare system. The final report of the on-site testing is shared with stakeholders and any possible errors or inconsistencies in the system are fixed.

5.3 SmartCDSS-DF: Tailoring the RUP to support specialized requirements

5.3.1 RUP tailoring mechanism

The RUP framework provides guidance with a rich set of software engineering principles. Its processes are generic and applicable to a wide range of projects, with respect to the size and complexity, in various domains with diverse development environments. Due to its generic nature and ability to cover a wide range of aspects, all of the RUP processes are not needed at the same time for a single project. Similarly, projects with special requirements may need additional features supported by the process model. In both cases, the process model requires a proper mechanism to fulfill the intended objectives of the project. Benefits of RUP include its flexibility towards the selection of a subset of focal processes and its ability to provide further guidance for customization. Moreover, it also provides guidance for the specialized requirements of a project, which are not available within the RUP framework. This mechanism of the RUP that allows customization of the components is referred to as ”tailoring the RUP”.

Tailoring the RUP is a non-trivial mechanism and requires thorough investigations to align the scope of the tailoring effort; select the suitable level of tailoring; and develop, configure, and publish the contents of the tailored components of the RUP. These considerations depend on a number of factors including the business context, the size of the software development efforts, the degree of novelty, and the type of development. Details of some candidate tailoring factors are provided in [15], and a summary of those factors and their effects on the tailoring process is shown in Figure 5.2. According to these factors, CDSS development has a high degree of novelty in terms of the involvement of stakeholders with diverse capabilities. The most important thing is that the end-user (i.e., domain experts) has high involvement throughout the development of the system. Even with the continuous evolution of the system, in terms of the knowledge, domain experts (i.e., the end-users) will always act as potential developers of the system. With this degree of novelty, the CDSS development framework requires customization in phases. Appropriate iterations are required to support the evolutionary knowledge requirements. Moreover, CDSS is a special type of healthcare application that needs special requirements for validation and formal verification of

the knowledge. In this respect, customization of the processes and associated tasks is required at different phases of the development framework.

Figure 5.2: RUP candidate tailoring factors

5.3.2 RUP tailoring adoption process for SmartCDSS-DF components

Different levels of tailoring can be applied to RUP; these range from very simple to incredibly complex [15]. The easiest tailoring level only affects the presentation of the published processes, which use external documents to describe the tailored processes and any customized methods. The medium level of tailoring includes customization of the configuration for the content pub-

lished from existing method contents, such as the ”thin” Method Plugin, which customizes the ”guidance” of existing method contents and defines a new ”delivery” process. The most complex tailoring level includes the extension of the existing method framework with new method contents (e.g., tasks, work products, and roles). Extension of the existing framework for new method contents is known as the ”structural” Method Plugin.

In order to define processes for SmartCDSS-DF components, the RUP Method Plugin tailoring mechanism is adopted. The following steps are used, according to the RUP tailoring guidelines, to formally develop the content of SmartCDSS-DF components:

1. RUP tailoring level for SmartCDSS-DF: Compared to classic software development, CDSS is a special type of a healthcare application and has a high degree of novelty that requires customization of method contents at different levels; these include disciplines, work products, processes, roles, and delivery processes. Based on these requirements, the structural Method Plugin is suitable for use with the CDSS.

2. Tailor the RUP: In this step, the method contents that are necessary to represent the CDSS requirements are developed. SmartCDSS-DF defines three Method Plugins to target the specialized requirements. i) The RM-ODP Method Plugin defines the contents to clearly distinguish the separation of concerns in the development of the CDSS. ii) The Formal methods Method Plugin describes the contents to formally verify the knowledge of the CDSS. iii) The HL7 vMR Method Plugin describes the contents to map the domain data into a standard HL7 vMR format in order to align CDSS integration with healthcare workflows.

3. Configure method contents: The configuration step is comprised of defining and organizing appropriate delivery processes into unified views as SmartCDSS-DF specifications.

4. Make the SmarCDSS-DF available: The configured contents and views are published as a website for the stakeholders. The Eclipse Process Framework (EPF) [54] is an open source platform that mainly focuses on the RUP process composition. The method contents are developed using EPF and published as a website and XML Metadata Interchange (XMI). The XMI format is imported to the Enterprise Architect (EA) [75] for the development of working products.

5.3.3 Motivation of RM-ODP Method Plugin for CDSS development

RM-ODP is a widely used and accepted reference architecture in the field of system and software model engineering [76]. There are substantial studies available in the field of healthcare that use RM-ODP as a baseline framework for the development of healthcare enterprise systems. The international standard Health Information Services Architecture (HISA) [18] uses three viewpoints of RM-ODP to present a flexible architecture in terms of selecting common use cases, actors, information, and services. This makes it easier to use with specific services, systems, and information [76]. NEHTA is an Australian eHealth initiative that seeks to develop an Interoperability Framework (NEHTA-IF) that is directly based on the RM-ODP specifications [19]. NEHTA-IF provides a common set of interoperability concepts and patterns and structures a set of rules to support the instantiation of the different frameworks that are used in the development of eHealth systems. Similarly, Lopez and Blobel [20] exploited different views of RM-ODP by aligning them with RUP processes. They proposed a development framework for interoperable HIS, which is referred to as HIS-DF. The flexibility of different reference models has yielded wide-range adoption of RM-ODP in enterprise systems (in general and healthcare system, in particular). It fulfills most of the demanding requirements that are common to most applications such as portability and interoperability of the systems, ease of integration with different architectural approaches, accommodating system evolution, and the ability to federate across technical and domain-specific aspects [77].

Adoption of RM-ODP for CDSS is mainly motivated by the successful implementation of healthcare systems using RM-ODP as a reference framework; this dictates that most of the common requirements of the CDSS are used in healthcare systems, including requirements related to interoperability and security and the fact that integration can be resolved using RM-ODP viewpoints. Moreover, below is a list of key aspects of RM-ODP that support CDSS-specific requirements.

1. Using RM-ODP viewpoints: Different viewpoints of RM-ODP help in the separation of the conceptual and technical processes of CDSS. Using clear distinctions in the views while developing CDSS, different stakeholders (domain experts, knowledge engineers, and developers) with different concerns and levels of maturity are required to understand one another.

For example, domain experts will represent clinical knowledge via formal tree representation, while developers will simultaneously analyze this knowledge for its technical aspects in terms of its integration with healthcare workflows using different mapping models.

2. Using RM-ODP modelling concepts: RM-ODP provides a rich set of modelling concepts that help represent the special artifacts (work products) created during the development process. For example, various artifacts created by the ”clinical knowledge pool” (e.g., CKM; a tree-based knowledge model) can be represented with RM-ODP (i.e., the ”template object” in the Enterprise viewpoint).

3. Using UML for RM-ODP concepts: RM-ODP provides a wide range of concepts that can be used to represent the special requirements of the target domain. At the same time, the artifacts and concepts of the RM-ODP can also be represented in UML standard notations. These concepts and artifacts are formally standardized and released as an ISO standard, which is known as UML4ODP [78]. UML4ODP is easily integrable with UML-based tools, which provide benefits for the traceability of the requirements. For example, in the current work, UML4ODP has been integrated into the Enterprise Architect [75] by importing the corresponding XMI format.

5.3.4 RM-ODP Method Plugin: supported viewpoints for CDSS

RM-ODP viewpoints bring together various elements of the specifications in a self-contained unit. This is done by using high level abstractions to represent the system components at different levels of concerns [79]. SmartCDSS-DF exploits the RM-ODP's principle of separating concerns by making two separate and distinct process pools (as mentioned in subsection 5.2) to separate the technical and conceptual aspects of the CDSS. Furthermore, processes in both pools are accommodated and mapped to the appropriate viewpoints of RM-ODP to produce consistent and coherent specifications. Finally, the RM-ODP Method Plugin combines these perspectives into comprehensive method contents that are aligned with the RUP specifications. Later, we will explain the method contents of the RM-ODP Method Plugin for viewpoints applicable to CDSS requirements. For detailed information about RM-ODP reference models and viewpoints, we suggest that readers

consult reference specifications [55–58] and other readings [19, 70, 79, 80].

5.3.4.1 Method contents for clinical knowledge pool

Domain experts are the main stakeholders that participate in the development of the conceptual aspects of the CDSS in terms of knowledge acquisition, validation of the knowledge, and transformation of the knowledge. The set of processes in this pool are iterative in nature; they are repeated for each set of new CDSS interventions, which are targeted towards the fulfilment of the common clinical objective. This iterative nature tends toward evolution of the clinical knowledge, and the processes involved range from conceptual to technical and include different transformations of the knowledge into a final executable format. In this regard, the different RM-ODP viewpoints help represent the different perspectives of knowledge acquisition, and the RUP framework allows the CDSS to develop in an evolutionary fashion.

- *Enterprise viewpoint*: Most of the concepts and artifacts of the clinical knowledge pool are mapped to Enterprise viewpoints. These include ODP concepts (used to represent the clinical objectives of the CDSS system for a particular domain) and support artifacts (used to represent the corresponding knowledge model). For example, the ODP concepts ”community” and ”objective” are applied to introduce the CDSS intervention used in the head and neck cancer domain in order to provide a second opinion during multidisciplinary conferences of physicians to develop treatment plans for oral cavity-based cancer patients. The detailed concepts of the Enterprise viewpoint that are used for the clinical knowledge pool are mapped with RUP concepts, as shown in Table 5.1.

- *Information viewpoint*: The Information viewpoint in the clinical knowledge pool represents information that is used in the clinical knowledge as content for recommendations. The Invariant schema concept is best-suited for representing the domain model for the target clinical knowledge model. Table 5.1 provides mapping of the Invariant schema to RUP-related concepts.

- *Computational viewpoint*: The Computational viewpoint expresses the functional design of CDSS in the clinical knowledge pool as a set of CIGs, which represent the refined clini-

Table 5.1: RM-ODP Method Plugin contents for clinical knowledge pool

ODP-Concepts	Related RUP Concepts	RUP Guidance	RUP Tasks	Example: Clinical knowledge pool
Enterprise viewpoint:				
Community	Target Organization	Business Vision	Setting business objectives, Identify business goal	Head and Neck Cancer department
Objective	Business Goal	Artifact: Business Goal	Identify business goal	CDSS Intervention for treatment plan of oral cavity cancer
ODPSystem	Business System	Artifact:Business System, Guideline: Business System	Business Architectural Analysis	Oral Cavity Cancer Management
Process	Business Process	Business Use Case Model	Find business actors and use cases	Creating clinical knowledge model
Role	Sub System	Business System	Business Architectural Analysis	Oral cavity cancer treatment plan clinical knowledge model
TemplateObject	Analysis Class	Artifact:Analysis Class, Guideline: Analysis Class	Analysis of guidelines	Refined Clinical Knowledge Model
Information viewpoint:				
Invariant Schema	Analysis Class (Entity Classes)	Artifact:Analysis Class, Guideline: Analysis Class	Analysis of R-CKM	Domain Model for oral cavity treatment plan
Computational viewpoint:				
Computational Object	Design Subsystem	Artifact: Design Subsystem	Analysis of R-CKM	Set of Medical Logic Modules as CIGs
Behavior	Design Component	Artifact: Design Component	Analysis of R-CKM	Root MLM and corresponding set of child MLMs
Interface	Interface	Artifact: Interface	Analysis of R-CKM	Set of MLMs as CIG with required data specification

cal knowledge model. For example, using HL7 Arden Syntax as the CIG language, a set of MLMs can be represented as the ”computational object” of a computational viewpoint, which represents a sub-part of clinical knowledge that is intended to trigger some intervention in the healthcare workflow. Candidate concepts of the computational viewpoint for the clinical knowledge pool, with the corresponding mapping to RUP concepts, are shown in Table 5.1.

5.3.4.2 Method contents for knowledge supporting tool pool

Knowledge engineers and developers are the main stakeholders of the knowledge supporting tool pool; these stakeholders know technical details and have experience in the development of software systems. They analyze the domain of CDSS in terms of designing easy to use toolsets that support knowledge acquisition and integrate refined knowledge into the healthcare workflow. Most of the processes in the knowledge supporting tool pool are similar with those in existing software development; therefore, it is easy to adopt the existing processes of RUP and its mapping to RM-ODP viewpoints is straightforward. Because the knowledge supporting tool pool also includes processes to target the specialized requirements of CDSS, such as formal verification of knowledge and its integration with the healthcare workflow, the contents for those requirements are exposed in separate Method Plugins as the ”Formal methods Method Plugin” and ”HL7 vMR

Method Plugin". Below is a brief overview of the RM-ODP viewpoints for the knowledge supporting tool pool. Detailed mappings of the ODP concepts with RUP concepts are shown in Table 5.2.

- *Enterprise viewpoint*: Most of the concepts of Enterprise viewpoints for the knowledge supporting tool pool are common with the clinical knowledge pool. For example, the development of CDSS will share same "community" and lead towards common "objectives". However, the approach used to achieve the objective in terms of the "process" and the perspective of the "ODPSystem" are different. For example, the "ODPSystem" in the knowledge supporting tool pool will cover the "authoring environment" perspective for the oral cavity cancer management system and the "process" will cover the case modeling that is used to represent the operation of the authoring environment to support the creation of the clinical knowledge model.

- *Information viewpoint*: Using the "Invariant schema" as the domain model is common between the clinical knowledge pool and the knowledge supporting tool pool. However, additional entity classes are required to model the authoring environment and the rule execution environment concepts with the Invariant schema. Moreover, the object model for the authoring environment will be represented as a static schema.

- *Computational viewpoint*: The Computational viewpoint for the knowledge supporting tool pool is completely different from the clinical knowledge pool; it reflects the separation of conceptual and technical concerns. For example, the "computational object" in the clinical knowledge pool represents a refined clinical knowledge model (subsystem) as a set of MLMs (CIGs). Alternatively, in the knowledge supporting tool pool, multiple subsystems will be involved to represent CIGs in action (e.g., the rule creation environment and execution environment).

5.3.5 Motivation of HL7 vMR Method Plugin for CDSS

Seamless integration of CDSS with HIS is one of the key requirements for successful implementation of the CDSS [81]. Integration processes are comprised of fundamental tasks to standardize the contents of the knowledge and make a standard clinical model baseline that clearly specifies

Table 5.2: RM-ODP Method Plugin contents for knowledge supporting tool pool

ODP-Concepts	Related RUP Concepts	RUP Guidance	RUP Tasks	Example: Knowledge supporting tool pool
Enterprise viewpoint:				
Community	Target Organization	Business Vision	Setting business objectives, Identify business goal	Head and Neck Cancer department
Objective	Business Goal	Artifact: Business Goal	Identify business goal	CDSS Intervention for treatment plan of oral cavity cancer
ODPSystem	Business System	Artifact:Business System, Guideline: Business System	Business Architectural Analysis	Oral Cavity Cancer Management (Authoring Environment)
Process	Business Process	Business Use Case Model	Find business actors and use cases	Use cases for knowledge authoring environment
Information viewpoint:				
Invariant Schema	Analysis Class (Entity Classes)	Artifact:Analysis Class, Guideline: Analysis Class	Use case analysis	Domain model for oral cavity treatment plan Class model for knowledge authoring envrionment
Static Schema	Object diagram	Artifact:Analysis Class, Guideline: Analysis Class	Use case analysis	Object diagram for oral cavity domain and knowledge authoring environment
Computational viewpoint:				
Computational Object	Design Subsystem	Artifact: Design Subsystem	Identify design elements. (Identifying classes, associations, and organizing in subsystems)	Rule creation environment Rule execution environment
Behavior	Design Component	Artifact: Design Component	Subsystem design: Distribution of responsibility of functions among components	Rule creation environment: ->rule creation ->transform rule to production
Interface	Interface	Artifact: Interface	Identify design elements. (Identify subsystem interfaces)	I_tranformRule: An interface between ”Rule Creation Environment” and ”Production Rule Environment”

the standard communication interfaces for CDSS. There are a wide range of interoperability standards available; the most commonly used standards are HL7 RIM-based standards (HL7 v3, HL7 CDA, HL7 FHIR, HL7 vMR) [44, 82, 83], and openEHR [84]. Example studies are available in the literature [29,30,43,81,85–89], where these standards are used for CDSS knowledge representation and integration with HIS. SmartCDSS-DF recommends HL7 vMR for knowledge content representation and standard interface definition. Selecting HL7 vMR for CDSS is mainly based on the following guidelines.

1. *Clinical Domain Model for CDSS:* HL7 vMR mainly focuses on the clinical domain model, which is specialized for CDSS requirements. The scope of the other HL7 standards and openEHR is broad; the reference models tend to cover the interoperability requirements for general healthcare systems. With generalized reference models, it becomes difficult to represent the CDSS requirements in an optimal way. Therefore, the efforts made by specialized groups in HL7 for the CDSS domain can be leveraged to narrow down the scope for CDSS knowledge content definitions in terms of using HL7 vMR.

2. *Easy integration with HIS:* The HL7 vMR clinical model is influenced by HL7 RIM. Most of the concepts' structures and semantics in HL7 vMR are derived from the specifications of HL7 RIM. CDSS, using HL7 vMR as a standard model, helps ease the integration with HIS, which is compliant with HL7 RIM-based standards.

3. *Comprehensive yet simple model:* The HL7 vMR model encapsulates and represents the clinical contents in a simple manner using the CDSS input and CDSS output specifications [44]. The input and output specifications are sufficiently comprehensive to cover all of the relevant patient information and other clinical information in terms of a strong clinical statement model. Moreover, adopting a rich collection of data types from HL7 RIM allows the model to be comprehensive and reflect the rich semantics of clinical contents. In addition to the standard input and output, it also provides specifications to configure and customize the interactions of CDSS systems.

4. *UML based specifications:* Lastly, the HL7 vMR model that uses standard UML for modeling, as well as all of its relevant specifications, is available in the XMI format. This enables

easy plug in of the specifications into existing development platforms and tools; we have integrated HL7 vMR specifications into the Enterprise Architect by importing standard XMI representation.

5.3.6 HL7 vMR Method Plugin: standard data model for CDSS knowledge contents

HL7 vMR specification only covers the structural aspects of the knowledge and provides models for standardization of the interfaces for CDSS. Unlike other HL7 standards (such as HL7 v3.0, which also provides guidelines and processes to model healthcare interpretability requirements [20]), HL7 vMR specifications provide only the data model and leave the processes required for mapping to the implementation side. Therefore, SmartCDSS-DF identifies candidate processes in order to align the CDSS requirements with HL7 vMR specifications.

5.3.6.1 HL7 vMR model specification process

SmartCDSS-DF introduces six processes to align the CDSS requirements with the HL7 vMR specifications and defines the standard interfaces with HIS integration. Figure 5.3 depicts the HL7 vMR Method processes, and detailed descriptions are provided in the following section.

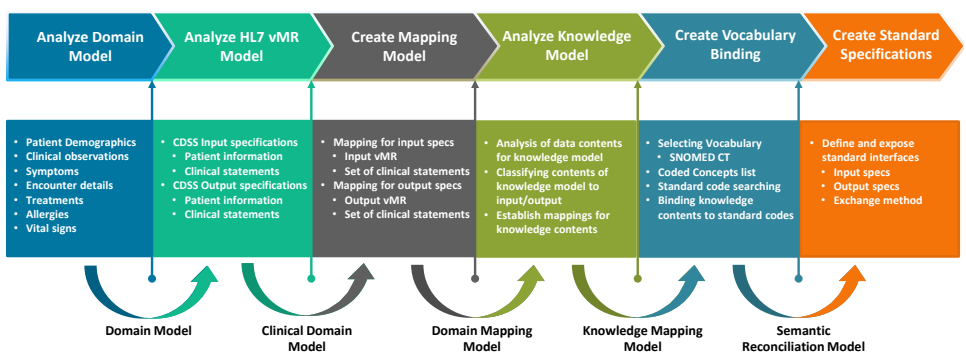

Figure 5.3: HL7 vMR model specification process

1. *Analyze Domain Model:* This includes tasks used to identify entities and relationships and classifies these based on their contextual relationships. The outcome of the process is the

"domain model", which represents a group of entities in the target domain. Example concepts include patient demographics, clinical observations, symptoms, treatments, allergies, and vital signs.

2. *Analyze HL7 vMR Model:* This is comprised of tasks used to identify the set of HL7 vMR concepts that are the best candidates for reflecting the domain concepts and contributes to the input and output contents of the desired knowledge model. The outcome of the process is the "clinical domain model", which has a subset of HL7 vMR concepts, such as patient-related concepts, and a set of clinical statements.

3. *Create Mapping Model:* This process encompasses mapping tasks and establishes mappings between the domain model and the clinical domain model at the concept level as well as the mappings of attributes for each mapped concept. The final outcome of this process is the "domain mapping model", which reflects the domain concepts as standard vMR concepts; these are used in the final knowledge content.

4. *Analyze Knowledge Model:* The clinical knowledge model (e.g., R-CKM) is analyzed to identify the scope of the knowledge contents. These are then classified to the appropriate input and output specifications and mappings are created in the "knowledge mapping model". The knowledge mapping model keeps the subset of mappings from the domain mapping model that represent the target scope of the knowledge under consideration.

5. *Create Vocabulary Bindings:* This includes tasks that select the appropriate clinical terminology standards and search standards codes for all of the relevant concepts of the knowledge mapping model in the target terminology repository. Finally, the contents of the knowledge are codified using formal bindings with standard terminology codes. For example, we used SNOMED CT [72] as the primary terminology for the cancer domain. The final outcome is the content model, referred to as SRM, which reconciles the domain model with the standard data model (HL7 vMR) and reinforces further semantics using the appropriate standard terminologies.

6. *Create Standard Interface Specifications:* The final process of the HL7 vMR Method Plugin uses content from SRM and determines the appropriate standard interfaces for communica-

tion with healthcare systems. The final interfaces expose only contents relevant to fragments of knowledge and obey further restrictions to control the interaction between HIS and CDSS.

5.3.6.2 Method contents for HL7 vMR Method Plugin

Knowledge engineers and developers are the main stakeholders who play active roles in most of the HL7 vMR Method Plugin processes. However, domain experts will also be involved as facilitators during the finalization of the appropriate clinical terminologies that are required for the final knowledge model. The plugin introduces concepts and guidelines related to the domain model, knowledge model, and standard specifications of the interfaces for integration with HIS.

The concepts of the plugins are well aligned with the ”analysis” and ”design” disciplines of the RUP. The processes of the plugin can be invoked and well distributed among the ”inception”, ”elaboration”, and ”construction” phases of RUP. From RUP, the ”analysis class” is the most frequently used and well-aligned concept with most of the HL7 vMR concepts. Concepts for the detailed mappings of candidate concepts from HL7 vMR to RUP are shown in Table 5.3.

5.3.7 Method Plugin for formal verification

As previously described, formal validation processes are included in the knowledge supporting tool pool; the main stakeholders are knowledge engineers and developers who understand mathematics and Z notation. Detailed processes and models are already explained in Chapter 4. In this section, we will introduce the method contents in context of SmartCDSS-DF. The method contents developed for the formal method in SmartCDSS-DF are generic enough to be applied for any requirements of the system. However, we intentionally narrowed down the scope of the contents to only reflect the formalization of the important aspects of the CDSS. Therefore, the scope of the formalism is limited to validation of the knowledge and other aspects of the CDSS such as formalizing the toolsets; the final CIGs are beyond the scope of this research.

Most of the constructs of Z notation are aligned with RUP concepts, and the processes covered in the formal specifications are well-matched with the ”analysis” and ”design” disciplines of the RUP. Analysis class and object diagrams are the main artifacts in RUP that are closely matched with Z constructs. Moreover, for SmartCDSS-DF, it is important that Z notation is used to formal-

Table 5.3: HL7 vMR Method Plugin contents

HL7 vMR concepts	**Related RUP concepts**	**RUP Guidance**	**RUP Tasks**	**Example: SmartCDSS-DF**
cdsInput	Analysis Class	*Artifact:* Analysis Class	Use case analysis	*Task:* Analysis of domain model Analysis of HL7 vMR input specification Mappings of domain model with vMR specifications *Artifact:* Domain Mapping Model
cdsOutput	Analysis Class	*Artifact:* Analysis Class	Use case analysis	*Task:* Analysis of domain model Analysis of HL7 vMR output specification Mapping of domain model with vMR specification *Artifact:* Domain Mapping Model
vmr	Analysis Class	*Artifact:* Analysis Class	Use case analysis	*Task:* Analysis of domain model (only patient information) Analysis of vmr specifications *Artifact:* Domain Mapping Model
dataTypes	Analysis Class	*Artifact:* Analysis Class	Use case analysis	*Task:* Analysis of domain model Analysis of HL7 vMR data types *Artifact:* Domain Mapping Model
cdsInputSpecification	Analysis Class, Interface	*Artifact:* Analysis Class, Interface; *Guideline:* Analysis Class	Use case analysis Identify subsystem interfaces	*Task:* Analysis of data contents for knowledge model Vocabulary bindings Define input specification *Artifact:* Knowledge Mapping Model, Semantic Reconciliation Model
cdsOutputSpecification	Analysis Class, Interface	*Artifact:* Analysis Class, Interface; *Guideline:* Analysis Class	Use case analysis Identify subsystem interfaces	*Task:* Analysis of data contents for knowledge model Vocabulary bindings Define output specification *Artifact:* Knowledge Mapping Model, Semantic Reconciliation Model

ize the models developed in the clinical knowledge pool, such as PM, CKM, and R-CKM, which use decision tree formalism. These models are represented as the ”TemplateObject” in RM-ODP, which has ”analysis class” as a counterpart concept in RUP (as shown in the RM-ODP plugin). Therefore, the representations of these models are well-aligned and mapped to RUP concepts; this causes the specifications to have a consistent view for all stakeholders. For example, R-CKM will provide a decision tree view to domain experts, a Z axioms and schemas view for knowledge engineers and developers, and the analysis class as a common view for the diverse range of stakeholders. A detailed mapping of Z notation constructs with RUP concepts is shown in Table 5.4.

Table 5.4: Formal Method Plugin contents using Z notations

Z concepts	**Related RUP concepts**	**RUP Guidance**	**RUP Tasks**	**Example: SmartCDSS-DF**
type	Analysis Class	*Artifact:* Analysis Class	Use case analysis	*Task:*PM, CKM, and R-CKM model analysis *Artifact:* Set of treatment
Free Type	Analysis Class	*Artifact:* Analysis Class	Use case analysis	*Task:*PM, CKM, and R-CKM model analysis *Artifact:* TreatmentPlan
Axiom	Analysis Class	*Artifact:* Analysis Class	Use case analysis	*Task:*PM, CKM, and R-CKM model analysis *Artifact:* Axiomatic definition of R-CKM concepts
Schema as type	Analysis Class	*Artifact:* Analysis Class	Use case analysis	*Task:*PM, CKM, and R-CKM model analysis *Artifact:*PredictionModel, ClinicalKnowledgModel, RefinedClinicalKnowledgeModel
Schema as operation	Object diagram (Analysis Model, Object diagram)	*Artifact:* Analysis Class; *Guideline:* Analysis Class	Use case analysis	*Task:*Analysis of validation criteria *Artifact:* EvolveRCKM

5.4 Summary

In this chapter, we demonstrated the SmartCDSS-DF - a development framework aligned with most of the existing development frameworks and specialized for CDSS development. We introduced the method plugin mechanism to provide support to our proposed knowledge acquisition, validation, and verification methods in the development of CDSS. These method plugins allows the separation of concerns for stakeholders with diverse capabilities. SmartCDSS-DF exploits this feature and introduced two separate process pools - for domain experts and technical experts (developers and knowledge engineers).

Chapter 6

Results and Evaluation

The results and evaluation for knowledge acquisition, validation, and verification is divided into three parts. First, the results of the three phase model is explained in the context of knowledge models - CKM, validated R-CKM, and R-CKM transformation into MLMs. Second, the inconsistencies explored in the three phase process using formal verification is described in detail and the refined validation processed is presented. Third, R-CKM is compared with: i) data-driven knowledge acquisition for quality (in terms of keeping accuracy preserve) and adherence to the guideline conformance, and ii) combined approach to ensure knowledge acquisition without proper validation produces inconsistent and non-integrable knowledge model.

6.1 Results of Three Phase Model

6.1.1 Phase-I:CKM

The team of physicians establish the clinical objective of incorporating the CDSS intervention for the *treatment plan recommendations for patients with an oral cavity tumor*. NCCN guidelines are selected to develop the CKM for the decision of treatment plans for these patients. Figure 6.1, shows the final outcome of the CKM of the clinical knowledge modeling phase from the NCCN guidelines. These guidelines cover the domain as a whole and provide a general view of the decision model. In order to convert NCCN into the final CKM decision tree model, physicians also integrate local practices into decision paths with sufficient evidence for improving patient care. For example, *ChemoInduction* was added to the final CKM with local practices, suggesting that induction chemotherapy followed by chemoradiotherapy before surgery improves overall survival of patients [90].

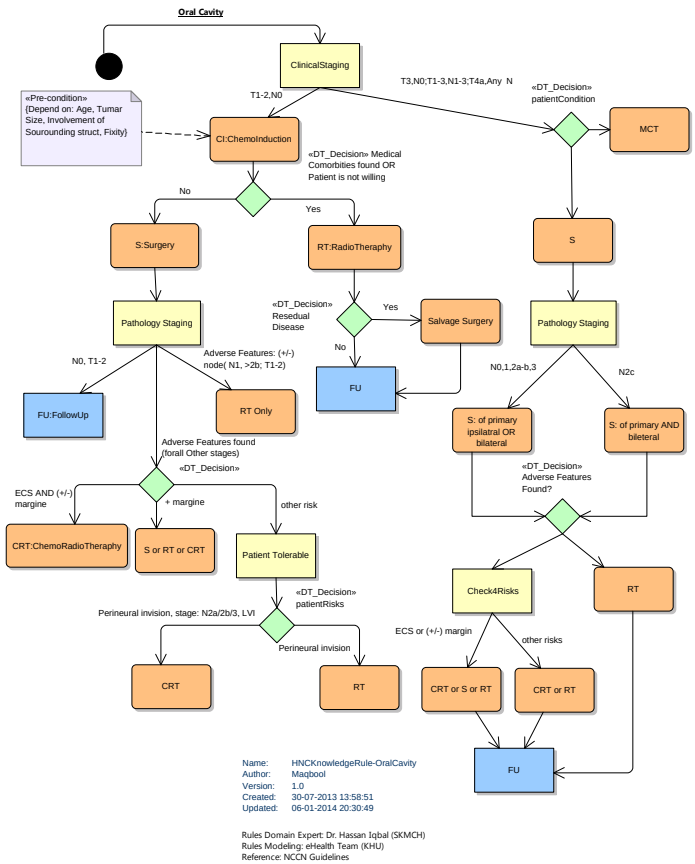

Figure 6.1: Clinical knowledge model for the oral cavity using the NCCN treatment plan

6.1.2 Phase-II:R-CKM

R-CKM is created from the PM while using the validation criteria defined by domain experts. Using a set of four validation criteria for the oral cavity, the validation process is applied on the oral cavity PM, which resulted in R-CKM, as shown in Figure 6.2.

Table 6.1 shows the details of the applicable validation criteria (AVC) for each decision path of the PM. For validation criteria 1, physicians establish the accuracy level of N = 50% (accuracy of paths based on training data). Five decision paths in the PM are conformed to the CKM (i.e., passing validation criteria of 1, 2, 3), while one decision path represent the local practices (passing criteria 1, 2, 4). Moreover, the PM decision always existed in leaf nodes, whereas in R-CKM the decision node could have occurred in the middle and would not work as a conditional node for the following sub-tree. For example, in the PM, decision path N0→N1→N4→N6, N6 is the decision

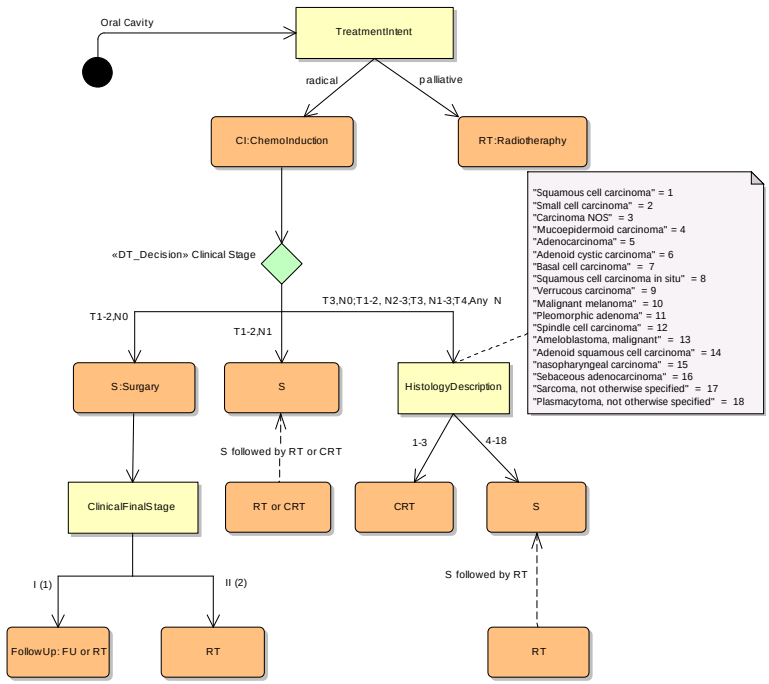

Figure 6.2: Refined clinical knowledge model for the oral cavity treatment plan

node representing S RT (surgery followed by radiotherapy). The same decision path is divided into two decision paths in R-CKM; i.e., TreatmentIntent→CI→S and TreatmentIntent→CI→S→RT, where S (as the decision node) in the first decision path is the condition node for the second decision path.

Table 6.1: **R-CKM evolution details**

Decision Path (PM)	**AVC**	**Refined Decision Path (R-CKM)**	**Remarks (Using CKM)**
N0→N2	$\{1, 2, 3\}$	TreatmentIntent→RT	• Conforms to CKM as: ○ RT is a secondary-level treatment for clinical stage III and IV patients in CKM ○ Palliative patients have stage III or stage IV (HIS statistics: 90%) ○ Thus, RT for palliative care conforms

Continued on next page

Table 6.1 – *Continued from previous page*

Decision Path (PM)	**AVC**	**Refined Decision Path (R-CKM)**	**Remarks (Using CKM)**
N0→N1→N3	$\{1, 2, 3\}$	i. TreatmentIntent→CI→S ii. TreatmentIntent→CI→S→FU or RT	• Conforms to CKM as: ◦ Surgery is done followed by RT for all clinical stages in CKM ◦ Radical patients may have any of the clinical stages (HIS statistics: I: 17.48%, II: 18.83%, III: 17.67% and IV: 46.02%) ◦ Thus, the given decision path conforms to CKM ◦ FU: FollowUp is added to refine the path because after pathology stage I some patients may ask for FU.
N0→N1→N4→N6	$\{1, 2, 3\}$	i. TreatmentIntent→CI→S ii. TreatmentIntent→CI→S→RT	• Same conformance as for N0→N1→N3
N0→N1→N4→N7	$\{1, 2, 3\}$	TreatmentIntent→CI→S→RT or CRT	• Same conformance as for N0→N1→N3 • CRT is added to refined path because N7 in PM suggests CRT with 30% accuracy. • Moreover, CRT is a tertiary level treatment in CKM for advanced clinical staging
N0→N1→N5→N8	$\{1, 2, 4\}$	TreatmentIntent→CI→CRT	• Not conforming to CKM, but physicians provide the following evidence from existing practices • C CRT makes 32.63% of the dataset and 84.5% patients among C CRT are stage III and stage IV • Because PM shows significant accuracy, 67.1%, for C CRT and is very effective
N0→N1→N5→N9	$\{1, 2, 3\}$	TreatmentIntent→CI→S→RT	• Same conformance as for N0→N1→N3

6.1.3 Phase-III:R-CKM transformation into executable rules

Evaluation of R-CKM to MLM strategies: Clinical models can be transformed into different sets of MLMs depending on the domain expert intuitions and logical connections in the decision path. For R-CKM, three candidate approaches were analyzed for the final executable knowledge.

The creation of a single MLM with a single event, which covers all decision paths of the R-CKM, is the most common approach. This approach has the advantage of publishing minimal MLMs to executable clinical knowledge, which makes it easy to maintain the growing knowledge base. However, this approach has limited MLM re-usability. Furthermore, it becomes difficult to identify and fix logical errors in a single MLM depending on a large clinical model.

The second approach is to create a set of MLMs for a given R-CKM shown in Figure 6.2; that is, create an individual MLM with a common event for each single decision node or a set of decision nodes sharing the immediate decision node. For R-CKM, six candidate MLMs can be created: one MLM for tree level 1 (covering ChemoInduction and Radiotherapy), one for tree level 2 (covering Surgery), two for tree level 3 (MLM covering RT or CRT and MLM covering CRT and S), and two for tree level 4 (MLM covering FollowUp or RT and RT and MLM covering RT). In this case, MLM for children decision node(s) include logic for a complete selected path originating from the root. For example, the MLM at level 2 concludes with Surgery after checking that the patient has done ChemoInduction and (s)he was on a radical treatment care. This approach has the potential to generate separate MLMs for each decision node, which can easily be traced to the original clinical model without digging into the detailed logic. Moreover, this approach is close to the rule generation of most machine learning tree classification, such as CHAID. The main limitations of this approach include duplications due to logic tracing parent nodes, multiple MLMs evoked for a single event, which will need the same amount of data from HIS, and maintenance problems occurring in the case a change is made to the clinical model.

The third approach is to create a set of MLMs controlled by the root MLM through the

MLM-calling mechanism. It allows modular logic to create reusable and understandable MLMs. The root MLM is exposed to a trigger for a particular event and imports all data required for the decision logic used in the ”called” MLMs - so called *subMLMs*. SubMLMs can also play the role of ”caller” for other subMLMs. Using this approach, a different set of MLMs for the same clinical model can be created depending on the modularity of the logic. For R-CKM, four candidate MLMs are selected to cover all decision paths. Figure 6.3 shows a detailed decision part of each MLM highlighted in R-CKM.

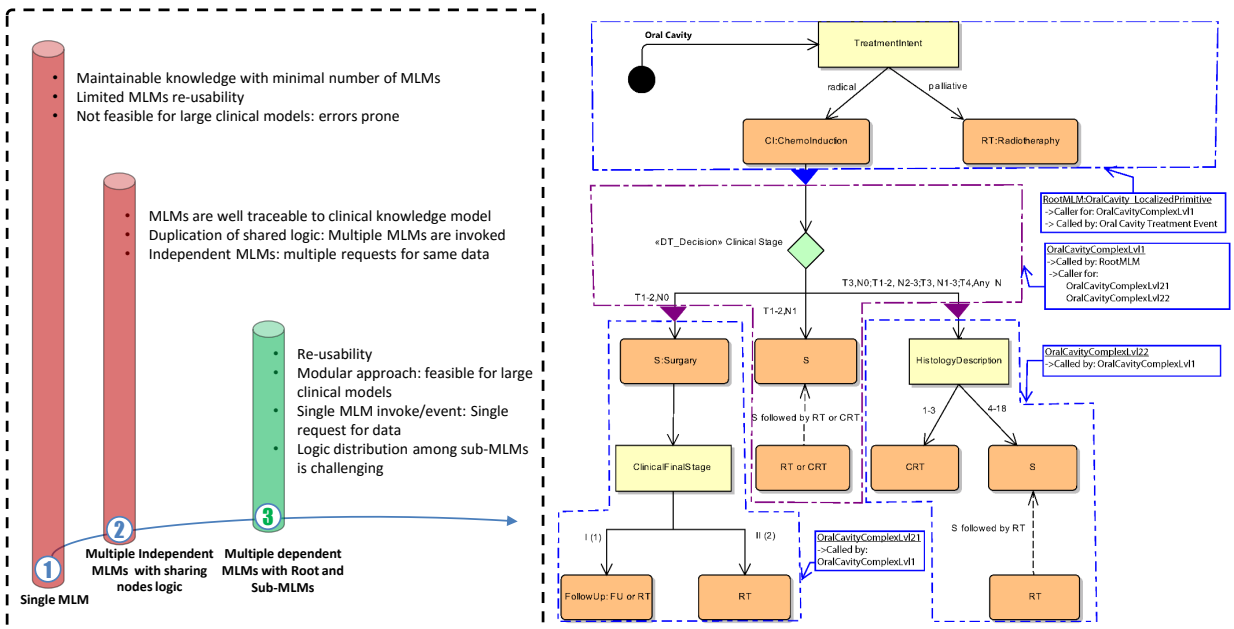

Figure 6.3: Candidate MLMs for oral cavity R-CKM

At first level of R-CKM, RootMLM:OralCavityLocalizedPrimitive is selected as the primitive root MLM, which is exposed for the oral cavity treatment plan event. It assigns a decision part at level 1 of the model for the primitive treatment plan. For further decisions it calls subMLM_ :OralCavityComplexLvl1 MLM, which represents the logic at level 2 based on clinical staging. Some treatment plans at level 2 and all other treatment plans in subsequent levels (i.e., levels 3 and 4) are delegated to two subMLMs OralCavityComplexLvl21 and OralCavityComplexLvl22, which are called from subMLM:OralCavityComplexLvl1.

RootMLM_ :OralCavityLocalizedPrimitive is the controller MLM triggered on the oral cavity treatment plan, and recommendations are made in coordination with subMLMs. This approach avoids duplication in logic. Furthermore, distribution of logic in subMLMs gives control to managing the overall logic of the clinical model.

Data specifications for MLMs and mapping to a standard model (HL7 vMR): Data specifications for each MLM are important for formal development of internal logic. Data specifications include activities, enlisting clinical data required for the MLM, representation of clinical data in a standard data model, and mapping of coded concepts into a standard vocabulary.

For the oral cavity treatment event, four candidate MLMs need different sets of clinical data to recommend an appropriate treatment plan. OralCavity_LocalizedPrimitive is the root MLM triggered against the oral cavity treatment event; therefore, the data required for subMLMs are imported by the root MLM and passed as an argument to caller subMLMs. The root MLM OralCavity_LocalizedPrimitive requires data regarding treatment intent and primary level treatment for the treatment plan recommendation at the primary level. It needs further detailed data if primary level treatment is done and it delegates control to subMLM with corresponding data for secondary or tertiary level recommendations.

OralCavityComplexLvl1 subMLM is called by the root MLM OralCavityLocalizedPrimitive and expects clinical data regarding clinical staging T and N values and treatment completed at secondary and tertiary levels. It also expects clinical data regarding clinical staging S and disease (histology findings), which are passed to subMLMs OralCavityComplexLvl21 and OralCavityComplexLvl22, respectively, if necessary to call further recommendations.

Clinical data specified for each MLM are required to have standard representation of vMR that can be understood by CDSS. All individual clinical data concepts are needed for mapping into vMR concepts. Table 6.2 lists HIS data mapping to vMR concepts in detail.

Table 6.2: **HIS clinical data mapping to HL7 vMR**

HIS Concepts	**vMR Concepts**	**Attributes Mappings (HIS-vMR)**
Clinical Stage T	ObservationResult	Clinical Stage T = observationFocus Clinical Stage T value = observationValue
Clinical Stage N	ObservationResult	Clinical Stage N = observationFocus Clinical Stage N value = observationValue
Clinical Stage S	ObservationResult	Clinical Stage S = observationFocus Clinical Stage S value = observationValue
Treatment Intent	ProcedureEvent	Treatment Intent = procedureCode Treatment Intent value = procedureMethod
Histology Description	Problem	Histology Description = problemCode
Treatment Plan	ProcedureEvent	Treatment Plan = procedureCode Treatment Plan value = procedureMethod

As indicated in Table 6.2, all of the values used in clinical data are coded concepts and mapped to vMR coded attributes. SNOMED codes for all HIS concepts and the corresponding value set is searched in the SNOMED repository. Table A.1 (Appendix A) lists all related SNOMED codes associated with HIS concepts and corresponding values.

Creation of candidate MLMs: Arden Syntax is a comprehensive specification supporting large numbers of operators, various control structures, including decision and looping structures, and comprehensive models for various data types. Knowledge engineers summarize the basic artifacts that are needed to transform the R-CKM into corresponding MLMs and provide physicians with training on using these artifacts. The basic logic for MLMs:OralCavity_LocalizedPrimitive is provided in Appendix B.

6.2 Results of Formal Verification

6.2.1 Evaluation of formally verified validation model

In this section, we will evaluate the validation process for the knowledge acquisition in the context of formal verification. By applying formal verification to the proposed model, it is revealed that the current validation process has many hidden inconsistencies. In the presence of these inconsistencies, the R-CKM evolution is not always guaranteed to be valid.

After formal verification, the validation process is updated to cover up the inconsistencies (revealed as pre-conditions) of the evolution operation for the R-CKM model. The final validation process, with the suggested improvements, is depicted in Figure 6.4.

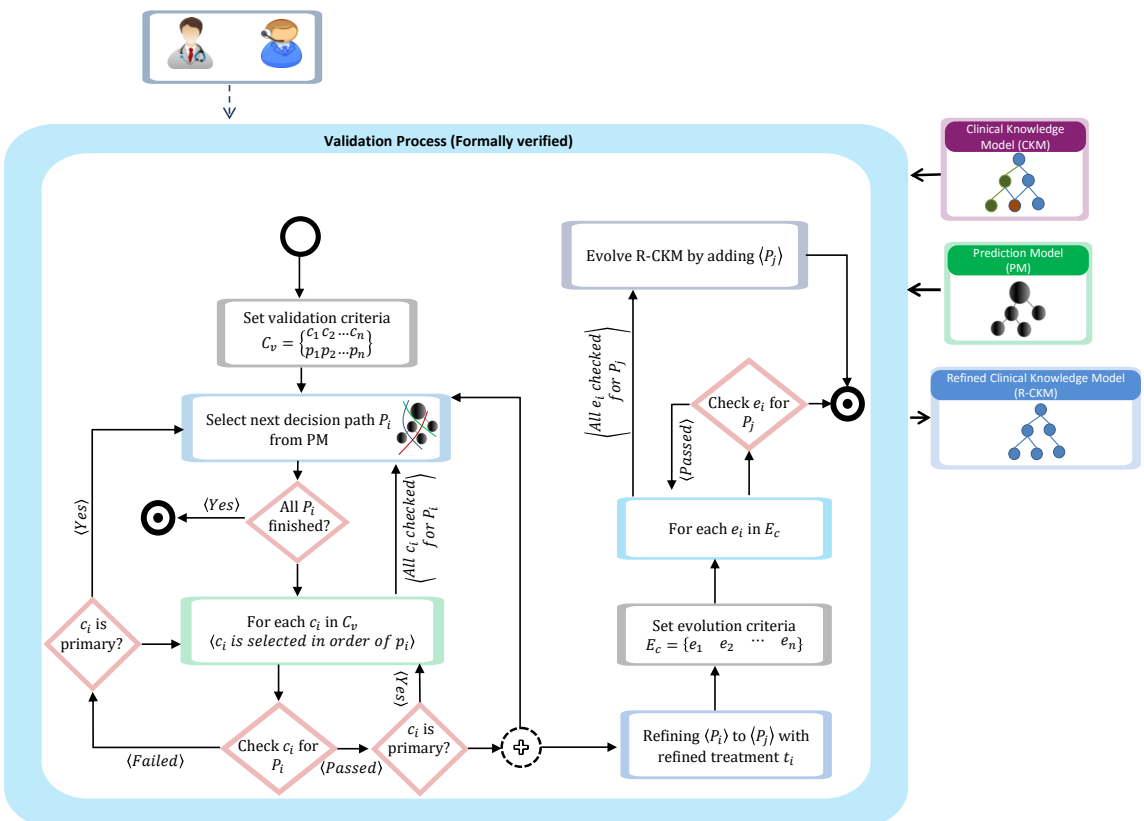

Figure 6.4: Formally verified knowledge validation process for R-CKM

In validation process(Figure 3.4), the repercussions of the refinement in decision path was unknown in advance. Formal verifications introduce further criteria to make sure the R-CKM model

created is valid using consistent validation process. The validation processes is further extended (Figure 6.4) to support refinements by considering the additional criteria explored during formal verification.

Table 6.3 summarise the nine criteria in addition to the four basic validation criteria. The newly discovered criteria covers up the consistency of the R-CKM model after refinements are made to the PM decision path.

Table 6.3: **Validation criteria derived from formal verification**

C.No	Criteria	Remarks
1.	$RCKM \neq \emptyset \Rightarrow head\,(\mathrm{dom}\, dp_{pm}?) = rootRCKM$	• Root of the R-CKM remains same for any decision path when R-CKM have already some decision paths. • Root of the R-CKM will be first condition for decision path when R-CKM have no decision path.
2.	$\{\forall pos : \mathbb{N} \mid pos \in \mathrm{dom}\, refinements? \bullet pos > 1 \wedge pos \leq (\#(\mathrm{dom}\, dp_{pm}?) + \#(\mathrm{ran}\, dp_{pm}?))$	• Refinements in PM decision path for treatment must be conformed. • Example: Treatment refinements in root of the decision path is not conformed.
3.	$\mathrm{ran}(\mathrm{dom}\, rckmPath!) \subset \mathrm{ran}\, decisionPathConditionRCKM$	• Conditions in the refined decision path must comes from defined condition set of the R-CKM. • Example: Conditions outside condition set make R-CKM non-integrable to HIS workflows.
4.	$\mathrm{ran}(\mathrm{ran}\, rckmPath!) \subset \mathrm{ran}\, ConclusionRCKM$	• Conclusion in the refined decision path must must be within scope of the defined treatments. • Example: Conclusions for treatment plan must be valid cancer treatment.
5.	$(\mathrm{ran}(\mathrm{ran}\, rckmPath!) \cap \mathrm{ran}\, decisionPathConditionRCKM) \subset \mathrm{ran}\, decisionPathConditionRCKM$	• Conclusion of the refined path may be condition of another decision path in R-CKM. • Example: The refined path may be an extension to existing decision path.
6.	$0 \leq decPathRCKMAccuracy(rckmPath!) \leq 100$	• Refined decision path accuracy must be within range of 0 and 100. • Example: The refined decision path should be tested for set of patient data.

Continued on next page

Table 6.3 – *Continued from previous page*

C.No	Criteria	Remarks
7.	$head\ (\mathrm{dom}\ rckmPath!) \notin \mathrm{ran}\ ConclusionRCKM \cap \mathrm{ran}\ decisionPathConditionRCKM$	• The first condition in refined decision path must not be treatment plan. • Example: Treatment plan is given based on some available symptoms (conditions).
8.	$\exists\, dp : decisionPathRCKM \mid dp \in RCKM \bullet \mathrm{dom}\ rckmPath! = \mathrm{dom}\ dp \setminus last\ (\mathrm{dom}\ dp) \Rightarrow last\ (\mathrm{dom}\ dp) = \mathrm{ran}\ rckmPath!$	• Detailed explanation of the criteria 5.
9.	$\mathrm{dom}\ rckmPath! = \exists\, p_{ckm} : decisionPathCKM \mid p_{ckm} \in CKM \bullet \mathrm{dom}(p_{pm}?) \cup \mathrm{dom}\ p_{ckm}$	• Refined decision path must be conformed to CKM. • Example: The refined path is obtained from PM and refined after confirmation from CKM.

6.3 Evaluation against data-driven approach

The R-CKM was evaluated with data-driven approach to prove that R-CKM is better in two aspects from data-driven approach(Decision Tree). First the R-CKM will preserve performance (accuracy) on patient data which follows the standard CPGs. As a result the performance (accuracy) of the R-CKM will remain the same or higher from prediction model attained on the training data. Second, adhering to CPGs, the R-CKM will discourage the practices data which is not following standard CPGs. So in this context, in terms of performance (accuracy) R-CKM will show low accuracy by indicating non-adherence of the patient data to standard CPGs.

6.3.1 Evaluation on CPGs based practices dataset

Using data-driven approach: The predication model was created on dataset of 1229 patients with completed treatments from SKMCH. SKMCH follows the NCCN guidelines and also provides with treatment plans which are effective for patients. In this respect the patients with completed treatment plans are considered as CPGs adhered practices. The final prediction model achieved 71.0% accuracy for classification of final cancer treatment plan (Table 3.2).

Using R-CKM: We implemented the R-CKM by integrating four candidate MLMs into HIS. MLMs

are tested (validated) on the same 1229 patient medical records with complete treatments that were used for PM evaluation. The distribution of the patient test cases into MLM decision paths is based on their qualification for the condition part of the paths.

R-CKM accuracy is equivalent to the accuracy of RootMLM: MLM_1 where accuracy of MLM_1 is a weighted mean accuracy of the disjoint paths as shown in Equation 6.1.

$$R - CKM_{acc} = A_{MLM_1} = \frac{\sum_{i=1}^{n}(pat_{c_i} \times A_{p_i})}{\sum_{i=1}^{n} pat_{c_i}} \tag{6.1}$$

Where pat_{c_i} and A_{p_i} represent the number of patient cases assigned to path p_i and accuracy of path p_i respectively

MLM_1 includes the decision path that calls other subMLMs. The accuracy of calling disjoint path A_p is calculated as the weighted mean accuracy of individual MLMs, as shown in Equation 6.2.

$$A_p = \frac{\sum_{i=1}^{n}(pat_{c_i} \times A_{MLM_i})}{pat_{c_p}} \tag{6.2}$$

Where pat_{c_i} and A_{MLM_i} represent number of patient cases assigned to MLM_i and its accuracy respectively.

pat_{c_p} represents patient cases assigned to path p

The overall classification accuracy of the R-CKM for 1229 patient cases is 72.57%, which covers testing of the eight decision paths of all the four MLMs.

Table 6.4 presents the detail of MLMs and their paths. The distribution of patient cases over MLMs and its detailed steps of calculating accuracies are shown in Figure 6.5.

Table 6.4: **R-CKM validation using MLMs**

MLM	MLM Path
MLM1	P1: TreatmentIntent→RT
	P2: TreatmentIntent→CI
MLM2	P1: TreatmentIntent→CI→S→RT or CRT (CS: T1-2,N1)

Continued on next page

Table 6.4 – *Continued from previous page*

MLM	MLM Path
MLM3	P1: TreatmentIntent→CI→S (CS: T1-2,N0) P11: TreatmentIntent→CI→S→ FU or RT (FCS: I (1)) P12: TreatmentIntent→CI→S→ RT (FCS: II (2))
MLM4	P1: TreatmentIntent→CI→CRT (HistoDesc: 1,2,3) P2: TreatmentIntent→CI→S→RT (Other than HistoDesc: 1,2,3)

- MLM1:RootMLM:OralCavityLocalizedPrimitive
- MLM2:OralCavityComplexLvl1
- MLM3:OralCavityComplexLvl21
- MLM4: OralCavityComplexLvl22
- CS: Clinical Stage

- FCS:Final Clinical Stage
- HistoDescription 1:Squamous cell carcinoma
- HistoDescription 2:Small cell carcinoma
- HistoDescription 3:Carcinoma NOS

In Figure 6.5, p_i represents path number in MLM_i, pat_c represents patient cases for path p_i, C_c represents correctly classified patient cases, and W_c represent incorrectly classified patient cases of path p_i.

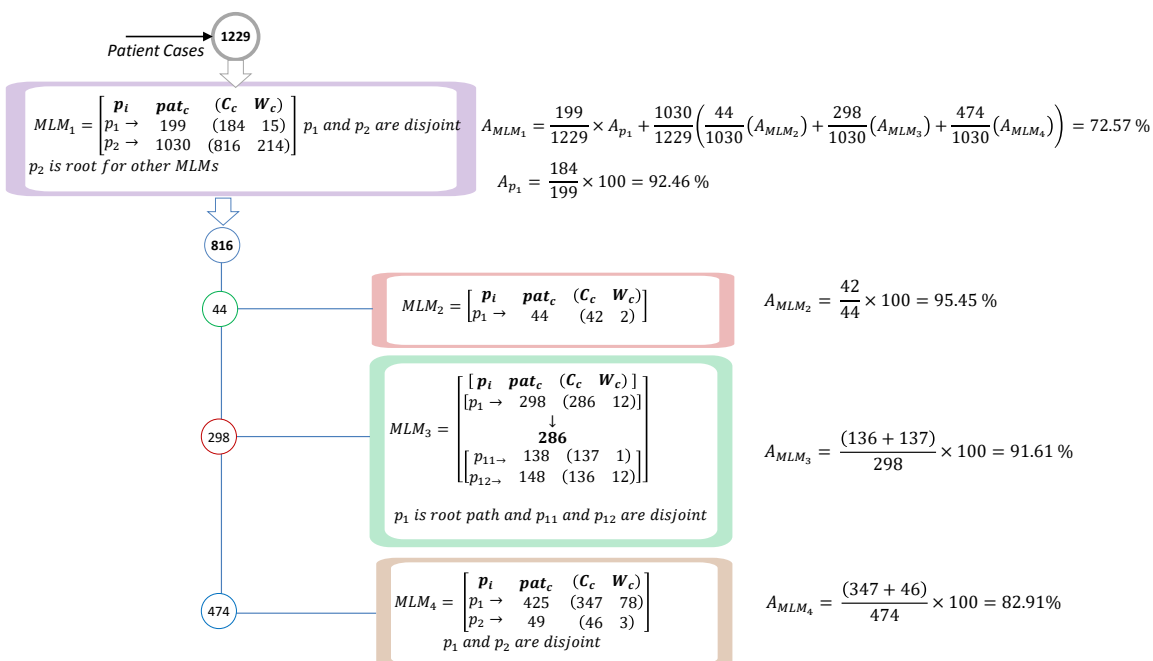

Figure 6.5: R-CKM accuracy using MLMs on CPGs based patient data

Discussion: Data-driven knowledge acquisition is considered as a suitable solution to represent the localized practices. It becomes worthwhile, if the predication model created on the given data achieve high performance. In current research on the given cancer data, we have achieved 71.0% for the final predication model. Using conventional data-driven approaches, we had a chance to directly plug-in the prediction model for final recommendations. However, as earlier mentioned, it only reflects the local practices and the model is not used as standard based practice model. We used the existing PM and evolved into R-CKM after rigorous validation process that conforms it from CKM - the CPGs. Although it is not important that the R-CKM performance should be increased compared to PM, however, it is intended that the final model evolved from PM should at least preserve the performance on standard CPGs practice dataset. From this experiment results, it is obvious that the R-CKM preserve the accuracy of PM to 72.57% (slightly improved). As a concluding remarks, R-CKM will give same performance on standard based dataset yet integrable to the HIS for final recommendation.

6.3.2 Evaluation on non-CPGs based practices dataset

Using data-driven approach: The experiment is performed on a (disjoint) dataset of 739 patients with incomplete treatments. The incomplete treatments needs further investigations to finalize the treatment plan according to CPGs. For this experiment, we considered the treatment plans as completed - which in reality, it is not conformed to the standard CPGs. Each decision path of the PM is evaluated for a set of candidate patient cases. The patient cases are distributed into six decision paths, where distribution is based on qualification for the condition part in the path.

The overall accuracy of the PM is calculated as the weighted mean of the accuracies of all of the decision paths, as shown in Equation 6.3.

The prediction model accuracy was recorded as 59.0% on the given incomplete treatment plans. Table 6.5 shows the detailed distribution of patient test cases over six decision paths and their corresponding accuracies. The PM Decision Path represents the decision paths covered by PM, Candidate Patient Cases are the record number of cases that qualify the

decision path and PM Path Accuracy is the percentage of patient cases correctly classified by the PM decision path.

$$PM_{acc} = \frac{\sum_{i=1}^{n}(pat_{c_i} \times A_{dp_i})}{\sum_{i=1}^{n} pat_{c_i}} \tag{6.3}$$

Where pat_{c_i} and A_{dp_i} represent the number of patient cases assigned to path dp_i and accuracy of path dp_i respectively

Table 6.5: **PM evaluation on disjoint patient test data**

Path#	**PM Decision Path**	**Candidate Patient Cases**	**PM Path Accuracy**
Path-1	Node0→RT	Palliative patients: 69	40.58% (C:28 W:41)
Path-2	Node0→Node1→Node3→S RT	Patient with radical and CS: T1: 139	95.68%(C:133, W:6)
Path-3	Node0→Node1→Node4→Node6→S RT	Patients with radical and FCS II: 123	73.98%(C:91, W:32)
Path-4	Node0→Node1→Node4→Node7→S RT	Patients with radical and FCS III;IV: 56	67.86%(C:38, W:18)
Path-5	Node0→Node1→Node5→Node8→C CRT	Patients with radical, CS:T3-4 and HistoDescription 1,2,3: 324	38.58% (C:125, W:199)
Path-6	Node0→Node1→Node5→Node9→S RT	Patients with radical, CS:T3-4 and HistoDescription other than 1,2,3: 28	85.71%(C:24, W:4)
	Overall PM Accuracy: PM_{acc}		**59.0%**

- CS: Clinical Stage
- FCS: Final Clinical Stage
- HistoDescription 1: Squamous cell carcinoma
- HistoDescription 2: Small cell carcinoma
- HistoDescription 3: Carcinoma NOS
- C: Correctly classified patient cases
- W: Wrongly classified patient cases

Using R-CKM: We implemented the R-CKM by integrating four candidate MLMs into HIS. MLMs are tested (validated) on the same 739 patient medical records with incomplete treatments that were used for PM evaluation. The distribution of the patient test cases into MLM decision paths is based on their qualification for the condition part of the paths.

R-CKM accuracy is equivalent to the accuracy of RootMLM: MLM_1 where accuracy of MLM_1 is a weighted mean accuracy of the disjoint paths as shown in Equation 6.1. The overall classification accuracy of the R-CKM for 739 patient cases is 53.0%, which covers

testing of the eight decision paths of all the four MLMs.

Table 6.4 presents the detail of MLMs and their paths. The distribution of patient cases over MLMs and its detailed steps of calculating accuracies are shown in Figure 6.6.

In Figure 6.6, p_i represents path number in MLM_i, pat_c represents patient cases for path p_i, C_c represents correctly classified patient cases, and W_c represent incorrectly classified patient cases of path p_i.

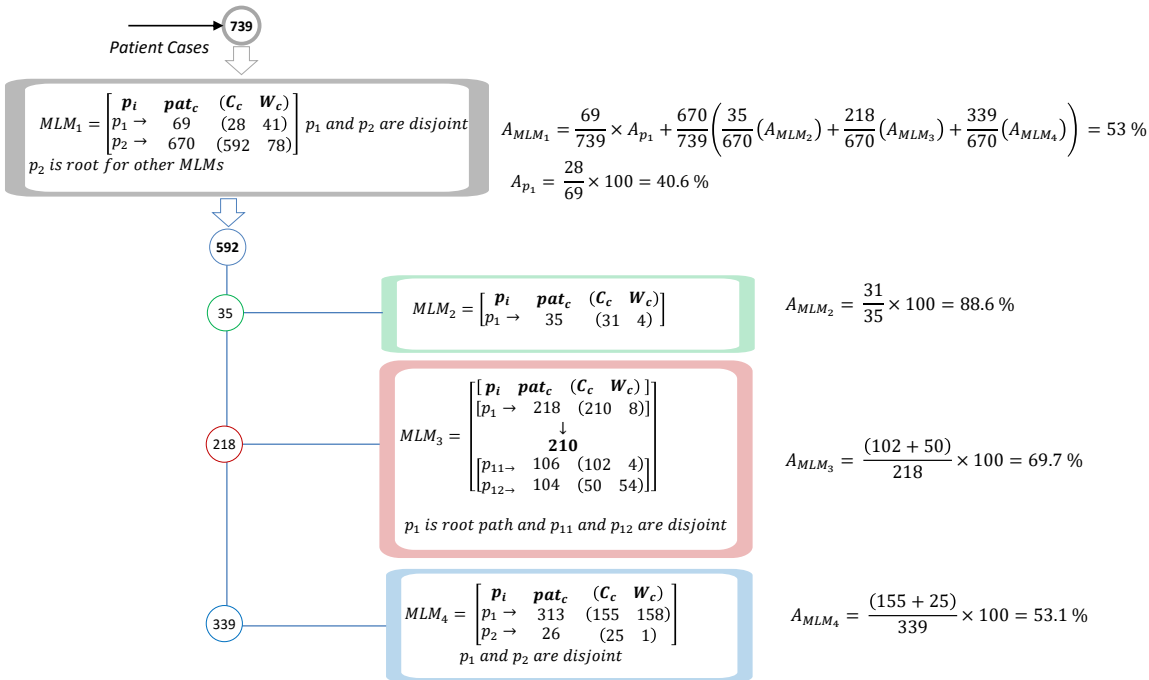

Figure 6.6: R-CKM accuracy using MLMs on non-CPGs based patient data

Discussion: R-CKM accuracy tends to increase if the provided patient data is following standard CPGs. The accuracy of R-CKM will deviate towards zero with the ratio the data is non-adhering to the standard CPGs. In this context, the R-CKM accuracy is comparatively less than PM accuracy, likely due to the facts that incomplete treatment plans are not yet finalized according to CPGs. So the validation process and finally the refinements process in the three phase model will yield always the refined knowledge - R-CKM which make sure the quality model for assessing the guideline based practices. The R-CKM will be integrable likewise PM, but yet ensures standard CPGs based recommendations.

6.4 Evaluation against combined approach

Comparison with combined approach: Using PM as a source and transforming it into the final knowledge model R-CKM after rigorous validation process which is conformed from CKM - the guidelines, is one way of combining the traditional data-driven approach and guideline based approach. However, the combination of these approaches can be done in another fashion - considering CKM as a source and evolving it from PM with all newly missing decision path in the CKM. In this section, we will discuss one of the existing approach which lies in the second category [1]. We cannot evaluate and compare our approach with existing one using quantitative method, as the ultimate outcome of the existing work is not integrable to HIS. However, we indicate limitations intrinsically exists in the given approach. Figure 6.7 depicts the high level comparison of the existing and our approach. Detailed comparison with our proposed approach is shown in Table 6.6.

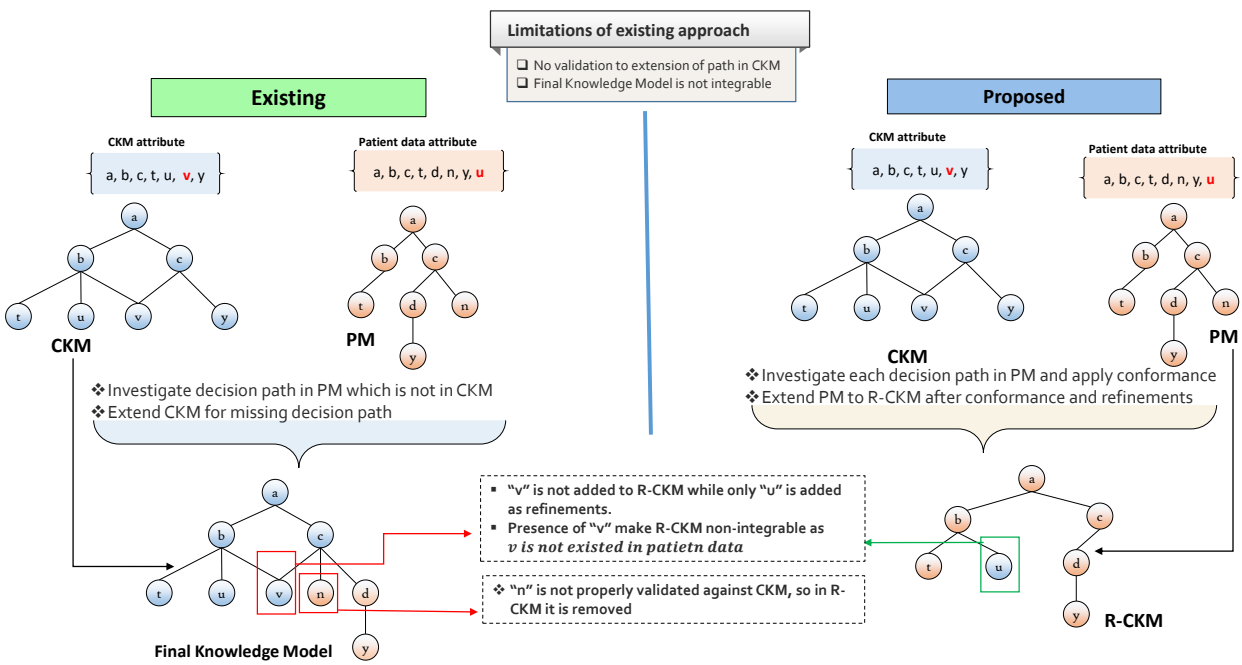

Figure 6.7: Comparison of R-CKM with Existing approach [1]

Discussion: The existing approach has different objective compared to our research work. They intend to make the guidelines more explicit for any missing contents and complete the decision

Table 6.6: Comparison of R-CKM with existing approach

SNO	Proposed Approach	Existing Approach	Remarks
1.	Evolve the PM into R-CKM using CKM	Evolve the CKM using PM	Existing approach is not integrable Example: Concept "v" is not existed in patient data.
2.	Evolving decision path based on conformance criteria from CKM	Evolving decision path is considered, based on performance (accuracy)	Every decision path is eligible: Only based on performance. (No validation) Example: i. "n" is not validated against CKM, so, removed from R-CKM. ii. "d" is added in final R-CKM, based, on evidence support.
3.	Evolving decision path if it has no conflict with guidelines		
4.	Evolving decision path if not conformed from CKM,only if sufficient evidences exists.		

tree to more specialized clinical knowledge model. The limitation of this work comes as it relies only on performance of the predication model created from local practices dataset. So evolution of the guidelines from PM is not following any rigorous validation process. In absence of such validation process, there is chance of adding decision path from PM to the CKM which may conflicts with standard guideline practices. In addition, only relying on the performance of the model without provision of any external evidences, it makes CKM localized guidelines rather than universally accepted practices. After all, the approach is beneficial to make guidelines explicit from local practices, however, the final CKM at the end is not necessary to be integrable with HIS workflow.

6.5 Summary

In this chapter, we demonstrated the results of the three phase model processes. Detailed description of the inconsistencies in three phase model was explored using formal verifications. The final refined verified validation process is presented with additional compulsory criteria. Finally, the proposed approach is compared with data-driven approach and combine knowledge acquisition approach.

Chapter 7

Conclusion and Future Directions

7.1 Conclusion and Future Direction

7.1.1 Conclusion

Integrating CDSS into the real practices is one of the demands to improve the patient care. The effectiveness of the CDSS is realised only if its decisions are based on a trusted clinical knowledge. Trust on clinical knowledge comes from the selection of appropriate knowledge resources and the knowledge acquisition process which is based on a rigorous validation and verification. In addition, the knowledge acquisition process shall possess the characteristic of explicit presentation of knowledge in a unified manner in order to provide the opportunities to different stakeholders with diverse capability to understand the requirements of CDSS. In this research work, we exploited diverse knowledge resources - patient data (EMR) and guidelines (CPGs) and introduced a novel guideline enabled knowledge acquisition method which combined the data-driven and guideline base knowledge acquisition.

The proposed work is based on rigorous validation process using a three-phase process model that produce three knowledge models: clinical knowledge model (CKM), prediction model (PM), and refined clinical knowledge model (R-CKM), whereby a R-CKM was created using a PM, which conforms to a CKM. This approach of knowledge acquisition ensures the CDSS interventions are well-aligned with patient data schemas, which otherwise would not be feasible with the direct implementation of general guidelines. The proposed approach encourages physicians to transform their professional experience into a sharable clinical knowledge using HL7 Arden Syntax. Moreover, HL7 vMR was used in the creation of MLMs to avoid the intrinsic integration issue (curly brace problem) of HL7 Arden Syntax.

In order to ensure that the proposed knowledge acquisition and validation method yields always a valid and a consistent knowledge, formal verification is required. The formal verification represents the semantic of the knowledge acquisition and validation and the associated concepts using mathematical tool-sets and allows reasoning on it to prove the consistency. For guideline enabled knowledge acquisition and validation process, we demonstrated the formal representation of models using Z notations. The inconsistencies left at initial design was refined after a formal theorem proving mechanism on the process models. Provision of the formal verification of our proposed method, enhances the trust of the domain experts on the knowledge acquisition and they integrate the final knowledge model (R-CKM) confidently to the healthcare workflow.

Finally, this research work contributes to the CDSS development framework SmartCDSS-DF, that provides a unified representation of the proposed method and presented it to different stakeholders involved in a real implementation project - Smart CDSS for head and neck cancer. The benefits of incorporating the knowledge acquisition and validation method into a development framework is two-fold: first, these methods are available with unified views in the development of CDSS. Second, the method is configurable and can be adopted for other CDSS projects in domains.

7.1.2 Future Work

Transforming knowledge from one representation form to another form is a necessary step of a decision support system for the execution. For example, in proposed work, PM and CKM were combined into R-CKM. R-CKM is a knowledge representation - so called CPGs, which is not directly executable in computer. The CPGs are required to be transformed into a sharable and an executable form in order to be interpreted by the computer such as, MLMs which are shareable and executable knowledge representation - so called CIGs. The knowledge transformation is consider trustable, if it follows a rigorous validation and verification process. In current research work, we evaluated the MLMs only on test based validation process. This validation is not sufficient which needs another level of validation process making sure the consistent executable knowledge for CDSS.

Our future work will expand on the research presented here and will consider the knowledge

validation and verification techniques to validate and verify the executable MLMs against R-CKM. Furthermore, completeness of R-CKM will be evaluated against the CKM with respect to available datasets and domain expert opinions.

Bibliography

[1] M. Toussi, J.-B. Lamy, P. Le Toumelin, and A. Venot, "Using data mining techniques to explore physicians' therapeutic decisions when clinical guidelines do not provide recommendations: methods and example for type 2 diabetes," *BMC medical informatics and decision making*, vol. 9, no. 1, p. 28, 2009.

[2] A. Wright and D. F. Sittig, "A framework and model for evaluating clinical decision support architectures," *Journal of biomedical informatics*, vol. 41, no. 6, pp. 982–990, 2008.

[3] S. Bakken, H. Jia, E. S. Chen, J. Choi, R. M. John, N.-J. Lee, E. Mendonca, W. D. Roberts, O. Velez, and L. M. Currie, "The effect of a mobile health decision support system on diagnosis and management of obesity, tobacco use, and depression in adults and children," *The Journal for Nurse Practitioners*, vol. 10, no. 10, pp. 774–780, 2014.

[4] A. X. Garg, N. K. Adhikari, H. McDonald, M. P. Rosas-Arellano, P. Devereaux, J. Beyene, J. Sam, and R. B. Haynes, "Effects of computerized clinical decision support systems on practitioner performance and patient outcomes: a systematic review," *Jama*, vol. 293, no. 10, pp. 1223–1238, 2005.

[5] (2015) Health information technology certification criteria, base electronic health record definition, and onc health it certification program modifications. National Coordinator for Health Information Technology (ONC),Department of Health and Human Services (HHS). (Accessed: March 26, 2015). [Online]. Available: https://federalregister.gov/a/2015-06612

[6] A. Wright, D. F. Sittig, J. S. Ash, S. Sharma, J. E. Pang, and B. Middleton, "Clinical decision support capabilities of commercially-available clinical information systems," *Journal of the American Medical Informatics Association*, vol. 16, no. 5, pp. 637–644, 2009.

[7] H. Varonen, T. Kortteisto, M. Kaila *et al.*, "What may help or hinder the implementation of computerized decision support systems (cdsss): a focus group study with physicians," *Family practice*, vol. 25, no. 3, pp. 162–167, 2008.

[8] D. L. Arts, A. G. Voncken, S. Medlock, A. Abu-Hanna, and H. C. van Weert, "Reasons for intentional guideline non-adherence: A systematic review," *International Journal of Medical Informatics*, vol. 89, pp. 55–62, 2016.

[9] T. G. McGinn, L. McCullagh, J. Kannry, M. Knaus, A. Sofianou, J. P. Wisnivesky, and D. M. Mann, "Efficacy of an evidence-based clinical decision support in primary care practices: a randomized clinical trial," *JAMA internal medicine*, vol. 173, no. 17, pp. 1584–1591, 2013.

[10] T. G. Kannampallil, L. K. Jones, V. L. Patel, T. G. Buchman, and A. Franklin, "Comparing the information seeking strategies of residents, nurse practitioners, and physician assistants in critical care settings," *Journal of the American Medical Informatics Association*, vol. 21, no. e2, pp. e249–e256, 2014.

[11] L. Moja, E. G. Liberati, L. Galuppo, M. Gorli, M. Maraldi, O. Nanni, G. Rigon, P. Ruggieri, F. Ruggiero, G. Scaratti *et al.*, "Barriers and facilitators to the uptake of computerized clinical decision support systems in specialty hospitals: protocol for a qualitative cross-sectional study," *Implement Sci*, vol. 9, p. 105, 2014.

[12] R. Shibl, M. Lawley, and J. Debuse, "Factors influencing decision support system acceptance," *Decision Support Systems*, vol. 54, no. 2, pp. 953–961, 2013.

[13] P. B. Jensen, L. J. Jensen, and S. Brunak, "Mining electronic health records: towards better research applications and clinical care," *Nature Reviews Genetics*, vol. 13, no. 6, pp. 395–405, 2012.

[14] "National comprehensive cancer network," http://www.nccn.org/, 2014, (Accessed: 24 Arpil, 2015).

[15] (2016) Rational unified process. IBM Corporation. (Accessed: January 22, 2016). [Online]. Available: http://home.iscte-iul.pt/~hro/RUPSmallProjects/

[16] I. Jacobson, G. Booch, and J. Rumbaugh, "The unified process," *IEEE Software*, vol. 16, no. 3, p. 96, 1999.

[17] B. Blobel, K. Engel, and P. Pharow, "Hl7 version 3 compared to advanced architecture standards," *Methods Inf Med*, vol. 45, pp. 343–53, 2006.

[18] ISO/TC215, "Iso 12967-1, 2, 3:2009, health informatics - service architecture," ISO, Standard, 2009.

[19] A. Bond, A. Hacking, Z. Milosevic, and A. Zander, "Specifying and building interoperable ehealth systems: Odp benefits and lessons learned," *Computer Standards & Interfaces*, vol. 35, no. 3, pp. 313–328, 2013.

[20] D. M. Lopez and B. G. Blobel, "A development framework for semantically interoperable health information systems," *International journal of medical informatics*, vol. 78, no. 2, pp. 83–103, 2009.

[21] M. Samwald, K. Fehre, J. De Bruin, and K.-P. Adlassnig, "The arden syntax standard for clinical decision support: Experiences and directions," *Journal of biomedical informatics*, vol. 45, no. 4, pp. 711–718, 2012.

[22] J. S. Ash, D. F. Sittig, C. K. McMullen, A. Wright, A. Bunce, V. Mohan, D. J. Cohen, and B. Middleton, "Multiple perspectives on clinical decision support: a qualitative study of fifteen clinical and vendor organizations," *BMC medical informatics and decision making*, vol. 15, no. 1, p. 35, 2015.

[23] A. Wright, J. S. Ash, J. L. Erickson, J. Wasserman, A. Bunce, A. Stanescu, D. Stqualitative study of the Hilaire, M. Panzenhagen, E. Gebhardt, C. McMullen *et al.*, "A qualitative study of the activities performed by people involved in clinical decision support: recommended practices for success," *Journal of the American Medical Informatics Association*, vol. 21, no. 3, pp. 464–472, 2014.

[24] M. Peleg, "Computer-interpretable clinical guidelines: A methodological review," *Journal of biomedical informatics*, vol. 46, no. 4, pp. 744–763, 2013.

[25] I. Cho, J. Kim, J. H. Kim, H. Y. Kim, and Y. Kim, "Design and implementation of a standards-based interoperable clinical decision support architecture in the context of the korean ehr," *International journal of medical informatics*, vol. 79, no. 9, pp. 611–622, 2010.

[26] V. Gomoi, M. Vida, R. Robu, V. Stoicu-Tivadar, E. Bernad, and O. Lupse, "From data mining rules to medical logical modules and medical advices." *Studies in health technology and informatics*, vol. 192, pp. 1094–1094, 2012.

[27] S. Perera, C. Henson, K. Thirunarayan, A. Sheth, and S. Nair, "Semantics driven approach for knowledge acquisition from emrs," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 2, pp. 515–524, 2014.

[28] M. Peleg, L. A. Gutnik, V. Snow, and V. L. Patel, "Interpreting procedures from descriptive guidelines," *Journal of biomedical informatics*, vol. 39, no. 2, pp. 184–195, 2006.

[29] M. Peleg, S. Keren, and Y. Denekamp, "Mapping computerized clinical guidelines to electronic medical records: Knowledge-data ontological mapper (kdom)," *Journal of biomedical informatics*, vol. 41, no. 1, pp. 180–201, 2008.

[30] M. Marcos, J. A. Maldonado, B. Martínez-Salvador, D. Boscá, and M. Robles, "Interoperability of clinical decision-support systems and electronic health records using archetypes: a case study in clinical trial eligibility," *Journal of biomedical informatics*, vol. 46, no. 4, pp. 676–689, 2013.

[31] R. Serban and A. Ten Teije, "Exploiting thesauri knowledge in medical guideline formalization," *Methods of information in medicine*, vol. 48, no. 5, pp. 468–474, 2009.

[32] G. Hripcsak, J. J. Cimino, S. B. Johnson, and P. D. Clayton, "The columbia-presbyterian medical center decision-support system as a model for implementing the arden syntax." in *Proceedings of the Annual Symposium on Computer Application in Medical Care*. American Medical Informatics Association, 1991, pp. 248–252.

[33] E. Shalom, Y. Shahar, M. Taieb-Maimon, G. Bar, A. Yarkoni, O. Young, S. B. Martins, L. Vaszar, M. K. Goldstein, Y. Liel *et al.*, "A quantitative assessment of a methodology

for collaborative specification and evaluation of clinical guidelines," *Journal of biomedical informatics*, vol. 41, no. 6, pp. 889–903, 2008.

[34] P. L. Miller, "Domain-constrained generation of clinical condition sets to help test computer-based clinical guidelines," *Journal of the American Medical Informatics Association*, vol. 8, no. 2, pp. 131–145, 2001.

[35] R. Bellazzi and B. Zupan, "Predictive data mining in clinical medicine: current issues and guidelines," *International journal of medical informatics*, vol. 77, no. 2, pp. 81–97, 2008.

[36] N. Pombo, P. Araújo, and J. Viana, "Knowledge discovery in clinical decision support systems for pain management: A systematic review," *Artificial intelligence in medicine*, vol. 60, no. 1, pp. 1–11, 2014.

[37] A. C. Lorena, L. F. Jacintho, M. F. Siqueira, R. D. Giovanni, L. G. Lohmann, A. C. De Carvalho, and M. Yamamoto, "Comparing machine learning classifiers in potential distribution modelling," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5268–5275, 2011.

[38] IBMSPSS_User_Guide, "Ibm spss decision trees 21," http://www.sussex.ac.uk/its/pdfs/SPSS_Decision_Trees_21.pdf, 2012, (Accessed: 24 December, 2014).

[39] L. Ohno-Machado, S. J. Wang, P. Mar, and A. A. Boxwala, "Decision support for clinical trial eligibility determination in breast cancer." in *Proceedings of the AMIA symposium*. American Medical Informatics Association, 1999, p. 340.

[40] S. Kraus, C. Drescher, M. Sedlmayr, I. Castellanos, H.-U. Prokosch, and D. Toddenroth, "Using arden syntax for the creation of a multi-patient surveillance dashboard," *Artificial intelligence in medicine*, 2015.

[41] P. A. de Clercq, J. A. Blom, H. H. Korsten, and A. Hasman, "Approaches for creating computer-interpretable guidelines that facilitate decision support," *Artificial intelligence in medicine*, vol. 31, no. 1, pp. 1–27, 2004.

[42] ArdenSyntax_WG, "Health level seven arden syntax for medical logic systems," Ann Arbor, MI:Health Level Seven, Inc, 2014, (Version 2.10).

[43] 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*, vol. 2010. American Medical Informatics Association, 2010, pp. 377–381.

[44] HL7CDS_WG, "Hl7 version 3 standard: Virtual medical record for clinical decision support (vmr-cds) logical model," Ann Arbor, MI:Health Level Seven, Inc, 2014, (Release 2).

[45] D. Chen, G. Doumeingts, and F. Vernadat, "Architectures for enterprise integration and interoperability: Past, present and future," *Computers in industry*, vol. 59, no. 7, pp. 647–659, 2008.

[46] A. Wright and D. F. Sittig, "A four-phase model of the evolution of clinical decision support architectures," *International journal of medical informatics*, vol. 77, no. 10, pp. 641–649, 2008.

[47] R. N. Shiffman, B. T. Karras, A. Agrawal, R. Chen, L. Marenco, and S. Nath, "Gem: a proposal for a more comprehensive guideline document model using xml," *Journal of the American Medical Informatics Association*, vol. 7, no. 5, pp. 488–498, 2000.

[48] K. Kawamoto and D. F. Lobach, "Design, implementation, use, and preliminary evaluation of sebastian, a standards-based web service for clinical decision support." in *AMIA*, 2005.

[49] K. Kawamoto and L. D. F, "Proposal for fulfilling strategic objectives of the us roadmap for national action on decision support through a service-oriented architecture leveraging hl7 services," *Journal of the American Medical Informatics Association*, vol. 14, no. 2, pp. 146–155, 2007.

[50] "IEEE 1471, recommended practice for architectural description of software- intensive systems," IEEE, 2000.

[51] I. Jacobson, G. Booch, J. Rumbaugh, J. Rumbaugh, and G. Booch, *The unified software development process*. Addison-Wesley Reading, 1999, vol. 1.

[52] IBM-Team, "Rational unified process: Best practices for software development teams," White Paper. [Online]. Available: https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf

[53] (2016) Ibm rational method composer. IBM. (Accessed: February 17, 2016). [Online]. Available: http://www.ibm.com/developerworks/rational/library/dec05/haumer/

[54] (2016) Eclipse process framework project (epf). Eclipse Foundation. (Accessed: January 25, 2016). [Online]. Available: http://www.eclipse.org/epf/

[55] ISO/IEC, "Is 107461, information technology open distributed processing reference model: Overview, also published as itu-t recommendation x.901," ISO, Standard, 1998.

[56] ISO-IEC, "Is 107462, information technology open distributed processing reference model: Foundations, also published as itu-t recommendation x.902," ISO, Standard, 2010.

[57] ISO/IEC, "Is 107463, information technology open distributed processing reference model: Architecture, also published as itu-t recommendation x.903," ISO, Standard, 2010.

[58] ISO-IEC, "Is 107464, information technology open distributed processing reference model: Architectural semantics, also published as itu-t recommen- dation x.904," ISO, Standard, 1998.

[59] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[60] M. Hussain, M. Afzal, W. A. Khan, and 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.

[61] M. Hussain, T. Ali, W. A. Khan, M. Afzal, S. Lee, and K. Latif, "Recommendations service for chronic disease patient in multimodel sensors home environment," *Telemedicine and e-Health*, vol. 21, no. 3, pp. 185–199, 2015.

[62] A. Bottrighi, L. Giordano, G. Molino, S. Montani, P. Terenziani, and M. Torchio, "Adopting model checking techniques for clinical guidelines verification," *Artificial intelligence in medicine*, vol. 48, no. 1, pp. 1–19, 2010.

[63] A. Ten Teije, M. Marcos, M. Balser, J. van Croonenborg, C. Duelli, F. van Harmelen *et al.*, "Improving medical protocols by formal methods," *Artificial intelligence in medicine*, vol. 36, no. 3, pp. 193–209, 2006.

[64] J. M. Spivey and J. Abrial, *The Z notation*. Prentice Hall Hemel Hempstead, 1992.

[65] J. Woodcock and J. Davies, *Using Z: specification, refinement, and proof*. Prentice-Hall, Inc., 1996.

[66] E. A. Boiten and J. Derrick, "From odp viewpoint consistency to integrated formal methods," *Computer Standards & Interfaces*, vol. 35, no. 3, pp. 269–276, 2013.

[67] H. Bowman, J. Derrick, P. Linington, and M. Steen, "Fdts for odp," *Computer Standards & Interfaces*, vol. 17, no. 5, pp. 457–479, 1995.

[68] M. Luck, P. Noriega, J. A. Rodriguez-Aguilar, C. Sierra *et al.*, "Communicating open systems," *Artificial Intelligence*, vol. 186, pp. 38–94, 2012.

[69] M. d'Inverno, M. Luck, M. Georgeff, D. Kinny, and M. Wooldridge, "The dmars architecture: A specification of the distributed multi-agent reasoning system," *Autonomous Agents and Multi-Agent Systems*, vol. 9, no. 1-2, pp. 5–53, 2004.

[70] R. O. Sinnott and K. J. Turner, "Specifying odp computational objects in z," in *Proc. of FMOODS*, vol. 96, 1996, pp. 375–390.

[71] M. Hussain, M. Afzal, T. Ali, R. Ali, W. A. Khan, A. Jamshed, S. Lee, B. H. Kang, and K. Latif, "Data-driven knowledge acquisition, validation, and transformation into hl7 arden syntax," *Artificial intelligence in medicine*, 2015.

[72] IHTSDO, "Snomed ct," http://www.ihtsdo.org/snomed-ct/, 2014, (Accessed: 26 December, 2014).

[73] T. Ali, M. Hussain, W. A. Khan, M. Afzal, J. Hussain, W. Hassan, A. Jamshed, B. H. Kang, and S. Lee, "Multi-model-based interactive authoring environment for creating shareable medical knowledge," 2015, under-review.

[74] P. Gooch and A. Roudsari, "Computerization of workflows, guidelines, and care pathways: a review of implementation challenges for process-oriented health information systems," *JAMIA-Journal of the American Medical Informatics Association*, vol. 18, no. 6, p. 738, 2011.

[75] (2016) Enterprise architect. SPARX SYSTEM. (Accessed: January 25, 2016). [Online]. Available: http://www.sparxsystems.com/

[76] J. Calvillo, I. Román, S. Rivas, and L. M. Roa, "Easing the development of healthcare architectures following rm-odp principles and healthcare standards," *Computer Standards & Interfaces*, vol. 35, no. 3, pp. 329–337, 2013.

[77] L. Kutvonen, "Using the odp reference model for enterprise architecture," in *EDOC Conference Workshop, 2007. EDOC'07. Eleventh International IEEE*. IEEE, 2007, pp. 231–238.

[78] "Iso/iec is 19793,information technology open distributed processing use of uml for odp system specifications, also published as itu-t recommendation x.906, 2009." ISO, Standard, 2009.

[79] H. Kilov, P. F. Linington, J. R. Romero, A. Tanaka, and A. Vallecillo, "The reference model of open distributed processing: foundations, experience and applications," *Computer Standards & Interfaces*, vol. 35, no. 3, pp. 247–256, 2013.

[80] J. R. Romero and A. Vallecillo, "Modeling the odp computational viewpoint with uml 2.0: The templeman library example," in *Workshop on ODP for Enterprise Computingg (WODPEC 2005))*. Citeseer, 2005, p. 24.

[81] C. Sáez, A. Bresó, J. Vicente, M. Robles, and J. M. García-Gómez, "An hl7-cda wrapper for facilitating semantic interoperability to rule-based clinical decision support systems," *Computer methods and programs in biomedicine*, vol. 109, no. 3, pp. 239–249, 2013.

[82] H. S. D. W. Group, "Hl7 clinical document architecture, release 2.0," Ann Arbor, MI:Health Level Seven, Inc, 2010, (Version 2.0).

[83] (2016) Hl7 fhir dstu2. HL7 Oranization. (Accessed: February 06, 2016). [Online]. Available: http://www.hl7.org/fhir/index.html

[84] (2016) openehr homepage. openEHR Foundation. (Accessed: February 06, 2016). [Online]. Available: http://www.openehr.org/

[85] Y.-F. Zhang, Y. Tian, T.-S. Zhou, K. Araki, and J.-S. Li, "Integrating hl7 rim and ontology for unified knowledge and data representation in clinical decision support systems," *Computer methods and programs in biomedicine*, vol. 123, pp. 94–108, 2016.

[86] H. S. Goldberg, M. D. Paterno, R. W. Grundmeier, B. H. Rocha, J. M. Hoffman, E. Tham, M. Swietlik, M. H. Schaeffer, D. Pabbathi, S. J. Deakyne *et al.*, "Use of a remote clinical decision support service for a multicenter trial to implement prediction rules for children with minor blunt head trauma," *International Journal of Medical Informatics*, vol. 87, pp. 101–110, 2016.

[87] P. V. Kukhareva, K. Kawamoto, D. E. Shields, D. T. Barfuss, A. M. Halley, T. J. Tippetts, P. B. Warner, B. E. Bray, and C. J. Staes, "Clinical decision support-based quality measurement (cds-qm) framework: Prototype implementation, evaluation, and future directions," in *AMIA Annual Symposium Proceedings*, vol. 2014. American Medical Informatics Association, 2014, p. 825.

[88] C. Marcos, A. González-Ferrer, M. Peleg, and C. Cavero, "Solving the interoperability challenge of a distributed complex patient guidance system: a data integrator based on hl7s virtual medical record standard," *Journal of the American Medical Informatics Association*, vol. 22, no. 3, pp. 587–599, 2015.

[89] N. Anani, R. Chen, T. P. Moreira, and S. Koch, "Retrospective checking of compliance with practice guidelines for acute stroke care: a novel experiment using openehrs guideline definition language," *BMC medical informatics and decision making*, vol. 14, no. 1, p. 1, 2014.

[90] A. Akbar, A. Bhatti, A. Syed, S. Khattak, A. Jamshed, and S. Kazmi, "P10 induction chemotherapy followed by chemoradiotherapy before surgery for rectal adenocarcinoma," *European Journal of Cancer*, vol. 50, no. 1, p. S12, 2014.

Appendix A

Oral cavity:SNOMED codes for HIS concepts

Table A.1: **HIS concept mapping with SNOMED concepts**

HIS Concept	SNOMED Concept	HIS Designated Values	SNOMED Code (for Values)	SNOMED Description (for Value)
Clinical Stage T	385356007 Tumor stage finding (finding)	T0	58790005	T0 category (finding), T0 category, T0 stage, Tumor stage T0,Tumor stage T0
		T1	23351008	T1 category (finding), T1 category, T1 stage, Tumor stage T1,Tumor stage T1
		T2	67673008	T2 category (finding), T2 category, T2 stage, Tumor stage T2,Tumor stage T2
		T3	14410001	T3 category (finding), T3 category, T3 stage, Tumor stage T3,Tumor stage T3
		T4	65565005	T4 category (finding), T4 category, T4 stage, Tumor stage T4,Tumor stage T4
Clinical Stage N	385382003 Node (category finding (finding), N stage finding, Node category finding, Node stage finding)	N0	62455006	N0 category (finding),N0 category,N0 stage, Node stage N0
		N1	53623008	N1 category (finding),N1 category,N1 stage, Node stage N1
		N2	46059003	N2 category (finding),N2 category,N2 stage, Node stage N2
		N3	5856006	N3 category (finding), N3category, N3 stage, Node stage N3
Clinical Stage S	80631005 Clinical stage finding (finding), Clinical stage finding	I	13104003	Clinical stage finding, Tumor stage finding, Clinical stage I (finding)
		II	60333009	Clinical stage finding, Tumor stage finding, Clinical stage II (finding)
		III	50283003	Clinical stage finding, Tumor stage finding, Clinical stage III (finding)
		IV	2640006	Clinical stage finding, Tumor stage finding, Clinical stage IV (finding)
Treatment Intent	395077000 Treatment intent (situation)	Radical	27762005	Radical procedure
		Palliative	363676003	Palliative - procedure intent

Continued on next page

Table A.1 – *Continued from previous page*

HIS Concept	SNOMED Concept	HIS Designated Values	SNOMED Code (for Values)	SNOMED Description (for Value)
Histology Description	250537006 Histopathology finding	Squamous cell carcinoma	402815007	Squamous cell carcinoma (disorder)
		Small cell carcinoma	74364000	Small cell carcinoma (morphologic abnormality)
		Carcinoma NOS	68453008	Carcinoma, no subtype (morphologic abnormality)
		Adenocarcinoma	35917007	Adenocarcinoma, no subtype (morphologic abnormality)
		Adenoid cystic carcinoma	11671000	Adenoid cystic carcinoma (morphologic abnormality)
		Adenoid cystic carcinoma	1338007	Basal cell carcinoma (morphologic abnormality)
		Squamous cell carcinoma in situ	59529006	Squamous cell carcinoma in situ, no International Classification of Diseases for Oncology subtype (morphologic abnormality)
		Verrucous carcinoma	89906000	Verrucous carcinoma (morphologic abnormality)
		Malignant melanoma	2092003	Malignant melanoma, no International Classification of Diseases for Oncology subtype (morphologic abnormality)
		Pleomorphic adenoma	8360001	Pleomorphic adenoma (morphologic abnormality)
		Spindle cell carcinoma	65692009	Spindle cell carcinoma (morphologic abnormality)
		Ameloblastoma, malignant	88253001	Ameloblastoma, malignant (morphologic abnormality)
		Adenoid squamous cell carcinoma	85956000	Adenoid squamous cell carcinoma (morphologic abnormality)
		nasopharyngeal carcinoma	449248000	Nasopharyngeal carcinoma (disorder)
		Sebaceous adenocarcinoma	54734006	Sebaceous adenocarcinoma (morphologic abnormality)
		Sarcoma, not otherwise specified	397355008	Dendritic cell sarcoma, not otherwise specified (morphologic abnormality)
		Plasmacytoma, not otherwise specified	415112005	Plasmacytoma (disorder)
		Mucoepidermoid carcinoma	4079000	Mucoepidermoid carcinoma (morphologic abnormality)

Continued on next page

Table A.1 – *Continued from previous page*

HIS Concept	SNOMED Concept	HIS Designated Values	SNOMED Code (for Values)	SNOMED Description (for Value)
Treatment Plan	1. 413737006 Cancer hospital treatment completed (situation) 2. 225292002 Developing a treatment plan (procedure)	Chemotherapy	367336001	Chemotherapy (procedure)
		CRT (Chemoradiotherapy)	703423002	Combined chemotherapy and radiation therapy (procedure)
		RT (Radiotherapy)	108290001	Radiation oncology AND/OR radiotherapy (procedure)
		Surgery	387713003	Surgical procedure (procedure)
		Induction Chemotherapy	450827009	Induction chemotherapy (procedure)

Appendix B

Oral cavity primitive root MLM

```
maintenance:
 title: Localized Treatment Plan for Oral Cavity Cancer: Primitive MLM;;
 mlmname: OralCavity_LocalizedPrimitive;;
 arden: ASTM-E1460-1995;;
 version: 2.7;;
 institution: UC Lab and SKMCH;;
 author: Mr. Maqbool Hussain;;
 specialist: Dr. Muhammad Irfan and Dr. Hassan Iqbal ;;
 date: 2014-05-30;;
 validation: Prototype Testing;;
library:
 purpose: OralCavity_LocalizedPrimitive will trigger for treatment plan of oral cavity
     cancer as prototype in local clinical set-up for head and neck cancer;;
 explanation: This MLM is evoked when patient is provided in encounter for proposition
     of treatment plan having oral cavity cancer. The result of this MLM is
     recommendation of treatment plan based on patient ;;
 keywords: oral cavity cancer; treatment plan; NCCN Guidelines;;
 citations: NCCN Guidelines and SKMCH treatment knowledge model from data;;

knowledge:
 type: data-driven;;

data:
  oralCavityTreatment_plan := event {treatment_plan where class =
      oralCavityTreatmentPlanEvent};
  PatientClinicalStatementTreatments := object [ProcedureEvent, ProcedureEvent,
      ProcedureEvent, ProcedureEvent];
  PatientClinicalStatementObservations :=  object [ObservationResult,ObservationResult,
      ObservationResult];
  PatientClinicalStatementProblem := object [Problem];
  RecommendationRemarks := object [CS,ST];

```

```
27  patientCompletedTreatments := read as PatientClinicalStatementTreatments
28  { select ProcedureEvent from client where ProcedureEvent.procedureCode = "413737006"
        OR ProcedureEvent.procedureCode = "395077000" };
29
30  patientClinicalStaging := read as PatientClinicalStatementObservations
31  { select ObservationResult from client where ObservationResult.observationFocus IN ( "
        385356007", "385382003", "80631005" ) };
32
33  patientDisease := read as PatientClinicalStatementProblem
34  { select Problem from client where Problem.problemCode = "250537006" };
35
36 /* Treatment Recommendations as result of patient status */
37 Recommendation := object[ ProcedureEvent, RecommendationRemarks ];
38 recommendationList := (); /* List of Recommendations */
39
40 /*.........Child MLM declarations ..........*/
41 mlmOralCavityComplexLvl1:= MLM 'mlmOralCavityComplexLvl1.mlm'
42 ;;
43
44 evoke:
45 oralCavityTreatment_plan;;
46
47 logic:
48
49 /* Using "Extract Attribute Names" operator. It returns all names of attributes as
       string list for given object */
50 /* patClinicalStagingList := EXTRACT ATTRIBUTE NAMES patientClinicalStaging; */
51 patCompletedTreatmentsList := EXTRACT ATTRIBUTE NAMES patientCompletedTreatments;
52
53
54 /* Retrieving Treatment Intent */
55 treatmentIntent := ATTRIBUTE patCompletedTreatmentsList[1] FROM
       patientCompletedTreatments;
56 firstLvlTreatment := ATTRIBUTE patCompletedTreatmentsList[2] FROM
       patientCompletedTreatments;
57
58 IF (treatmentIntent.procedureCode = "395077000" AND treatmentIntent.procedureMethod = "
       363676003") THEN
59  IF ( (firstLvlTreatment IS NULL) OR (firstLvlTreatment.procedureCode = "413737006" AND
        treatmentIntent.procedureMethod != "108290001") )  THEN
60   plannedTreatment := new ProcedureEvent with "225292002" , "108290001";
61   plannedTreatmentRemarks := new RecommendationRemarks with "225292002", "Radiotherapy
```

```
        is recommended treatment plan !!!";
62    treatmentRecommedation := new Recommendation with plannedTreatment,
        plannedTreatmentRemarks;
63    recommendationList := recommendationList, treatmentRecommedation;
64
65    ELSE
66
67    plannedTreatmentRemarks := new RecommendationRemarks with "413737006", "Radiotherapy
        is already done, evaluate patient case for further treatment !!!";
68    treatmentRecommedation := new Recommendation with firstLvlTreatment,
        plannedTreatmentRemarks;
69    recommendationList := recommendationList, treatmentRecommedation;
70
71    END IF;
72
73    CONCLUDE TRUE;
74
75  ELSEIF (treatmentIntent.procedureCode = "395077000" AND treatmentIntent.procedureMethod
        = "27762005") THEN
76    IF ( (firstLvlTreatment IS NULL) OR (firstLvlTreatment.procedureCode = "413737006" AND
        treatmentIntent.procedureMethod != "450827009") )  THEN
77
78    plannedTreatment := new ProcedureEvent with "225292002" , "450827009";
79    plannedTreatmentRemarks := new RecommendationRemarks with "225292002", "Induction
        Chemotherapy is recommended treatment plan !!!";
80    treatmentRecommedation := new Recommendation with plannedTreatment,
        plannedTreatmentRemarks;
81    recommendationList := recommendationList, treatmentRecommedation;
82
83    ELSE
84
85    plannedTreatmentRemarks := new RecommendationRemarks with "413737006", "Induction
        Chemotherapy is already done !!!";
86    treatmentRecommedation := new Recommendation with firstLvlTreatment,
        plannedTreatmentRemarks;
87    recommendationList := recommendationList, treatmentRecommedation;
88
89    /*CALLING SUB MLM: */
90    recommendationLvl1List := call mlmOralCavityComplexLvl1 with
        patientCompletedTreatments, patientClinicalStaging, patientDisease;
91    recommendationList := recommendationList MERGE recommendationLvl1List;
92
```

```
93   END IF;
94
95   CONCLUDE TRUE;
96
97  ELSE
98
99   CONCLUDE FALSE;
100
101 END IF;;
102
103 action:
104
105 FOR recommendations IN recommendationList DO
106
107  WRITE recommendations.ProcedureEvent.procedureMethod || recommendations.
         RecommendationRemarks;
108
109 END DO;
110   ;;
111
112 End;;
```

Listing B.1: Oral cavity cancer: primitive MLM

Appendix C

List of Publications

International Journal Papers:

[1] **Maqbool Hussain**, Muhammad Afzal, Taqdir Ali, Rahman Ali, Wajahat Ali Khan, Arif Jamshed, Sungyoung Lee, Byeong Ho Kang, Khalid Latif; "Data-driven knowledge acquisition, validation, and transformation into HL7 Arden Syntax", Artificial Intelligence in Medicine (Elsevier, SCI, IF:2.03), published-online (September 15, 2015)

[2] **Maqbool Hussain**, Taqdir Ali, Wajahat Ali Khan, Muhammad Afzal, Sungyoung Lee, Khalid Latif,;"Recommendations service for chronic disease patient in multi-model sensors home environment" Telemedicine and e-Health (SCI, IF:1.6), 2014

[3] **Maqbool Hussain**, AsadMasood Khattak,Wajahat Ali Khan, Iram Fatima, Muhammad Bilal Amin, Zeeshan Pervez, Rabia Batool, Muhammad Amir Saleem, Muhammad Afzal, Muhammad Fahim, Muhammad Hameed Saddiqi, Sungyoung Lee, and Khalid Latif, "Cloud-based Smart CDSS for Chronic Diseases", In Journal of Health and Technology - 3, no. 2 (2013): 153-175.

[4] Wajahat Ali Khan, M Bilal, AM Khattak, **Maqbool Hussain**, Afzal, SY Lee and Eun Soo Kim, "Object Oriented and Ontology Alignment Patterns based Expressive Mediation Bridge Ontology (MBO)", Journal of Information Science, (SCIE, IF:1.08)(Accepted), 2014.

[5] Wajahat Ali Khan, AM Khattak, **Maqbool Hussain**, Bilal Amin, M Afzal, Christopher Nugent and Sungyoung Lee, "An Adaptive Semantic based Mediation System for Data Interoperability among Health Information Systems", Journal of Medical Systems (SCIE, IF 1.372), 38(8):28, 2014.

[6] Wajahat Ali Khan, **Maqbool Hussain**, Muhammad Afzal, Muhammad Bilal Amin, Muhammad Aamir Saleem and Sungyoung Lee, "Personalized-Detailed Clinical Model for Data Interoperability among Clinical Standards", Telemedicine and EHealth (SCI, IF:1.544), Vol. 19 Issue 8, pp.632-642, 2013.

[7] Wajahat Ali Khan, **Maqbool Hussain**, Khalid Latif, Muhammad Afzal, Farooq Ahmad, Sungyoung Lee, "Process Interoperability in Healthcare Systems with Dynamic Semantic Web Services", Computing Journal (SCI, IF:1.055), Vol. 95, Issue 9, pp 837-862, 2013.

International Conference Papers:

[8] **Maqbool Hussain** and Sungyoung Lee, "Recommendation Framework for Detecting Sedentary Behavior in Smart Home Environment", 12th International Conference on Ubiquitous Healthcare (u-Healthcare 2015), Osaka, Japan, Nov 30- Dec 02, 2015

[9] **Maqbool Hussain**, Taqdir Ali, Muhammad Afzal, Wajahat Ali Khan, Sungyoung Lee, Lets Fire the CDSS integration issue with FHIR, 14th International HL7 Interoperability Conference (IHIC 2013) and FHIR Connectathon, Sydney, Australia, 27-29 October, 2013

[10] **Maqbool Hussain**, Muhammad Afzal, Wajahat Ali Khan, and Sungyoung Lee. "Clinical decision support service for elderly people in smart home environment." In Control Automation Robotics & Vision (ICARCV), 2012 12th International Conference on, pp. 678-683. IEEE, 2012.

[11] **Maqbool Hussain**, Wajahat Ali Khan, Muhammad Afzal, and Sungyoung Lee. "Smart CDSS for smart homes." In Impact Analysis of Solutions for Chronic Disease Prevention and Management, pp. 266-269. Springer Berlin Heidelberg, 2012.

[12] Wajahat Ali Khan, Muhammad Bilal Amin, Asad Masood Khattak, **Maqbool Hussain**, and Sungyoung Lee. "System for Parallel Heterogeneity Resolution (SPHeRe) results for OAEI 2013." In OM, pp. 184-189. 2013.

[13] Wajahat Ali Khan, **Maqbool Hussain**, Bilal Amin, Asad Masood Khattak, Muhammad Afzal, and Sungyoung Lee. "AdapteR Interoperability ENgine (ARIEN): An approach of

Interoperable CDSS for Ubiquitous Healthcare." In Ubiquitous Computing and Ambient Intelligence. Context-Awareness and Context-Driven Interaction, pp. 247-253. Springer International Publishing, 2013.

[14] Wajahat Ali Khan, **Maqbool Hussain**, Muhammad Afzal, Muhammad Bilal Amin, and Sungyoung Lee. "Healthcare standards based sensory data exchange for Home Healthcare Monitoring System." In Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE, pp. 1274-1277. IEEE, 2012.

[15] Wajahat Ali Khan, **Maqbool Hussain**, Asad Masood Khattak, Muhammad Afzal, Bilal Amin, and Sungyoung Lee. "Integration of HL7 compliant smart home healthcare system and HMIS." In Impact Analysis of Solutions for Chronic Disease Prevention and Management, pp. 230-233. Springer Berlin Heidelberg, 2012.

[16] Wajahat Ali Khan, **Maqbool Hussain**, Asad Masood Khattak, Muhammad Bilal Amin, and Sungyoung Lee. "Saas based interoperability service for semantic mappings among healthcare standards." In 8th International Conference on Innovations in Information Technology. 2012.

[17] Wajahat Ali Khan, Asad Masood Khattak, Sungyoung Lee, **Maqbool Hussain**, Bilal Amin, and Khalid Latif. "Achieving interoperability among healthcare standards: building semantic mappings at models level." In Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication, p. 101. ACM, 2012.

[18] Wajahat Ali Khan, **Maqbool Hussain**, Muhammad Afzal, Sungyoung Lee, Interoperability Engine for Smart CDSS, AAL Summit 2012, Spain

[19] Jamil Hussain, Wajahat Ali Khan, Muhammad Afzal, **Maqbool Hussain**, Byeong Kang and Sungyoung Lee, "Adaptive User Interface and User Experience based Authoring Tool for Recommendation Systems", Ubiquitous Computing and Ambient Intelligence (UCAmI), Belfast, Northern Ireland, December 2-4, 2014.

[20] Muhammad Afzal, **Maqbool Hussain**, Taqdir Ali, Wajahat Ali Khan, Sungyoung Lee, Byeong Ho Kang, "MLM-based Automated Query Generation for CDSS Evidence Support", Ubiquitous Computing and Ambient Intelligence (UCAmI), Belfast, Northern Ireland, December 2-4, 2014.

[21] Taqdir Ali, **Maqbool Hussain**, Wajahat Ali Khan, Muhammad Afzal, and Sungyoung Lee. "Customized clinical domain ontology extraction for knowledge authoring tool." In Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication, p. 23. ACM, 2014.

[22] Taqdir Ali, **Maqbool Hussain**, Wajahat Ali Khan, Muhammad Afzal, Byeong Ho Kang, and Sungyoung 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, pp. 266-272. IEEE, 2014.

[23] Muhammad Afzal, **Maqbool Hussain**, Wajahat Ali Khan, Taqdir Ali, Sungyoung Lee, and Byeong Ho 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, pp. 273-280. IEEE, 2014.

[24] Muhammad Afzal, **Maqbool Hussain**, Wajahat Ali Khan, Taqdir Ali, Sungyoung Lee, and Hafiz Farooq Ahmad. "Meaningful Integration of Online Knowledge Resources with Clinical Decision Support System." In Inclusive Society: Health and Wellbeing in the Community, and Care at Home, pp. 280-285. Springer Berlin Heidelberg, 2013.

[25] Taqdir Ali, **Maqbool Hussain**, Wajahat Ali Khan, Muhammad Afzal, and Sungyoung Lee. "Authoring tool: Acquiring sharable knowledge for Smart CDSS." In Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE, pp. 1278-1281. IEEE, 2013.

[26] Rabia Batool, Wajahat Ali Khan, **Maqbool Hussain**, Jahanzeb Maqbool, Muhammad Afzal, and Sungyoung Lee. "Towards personalized health profiling in social network." In Informa-

tion Science and Service Science and Data Mining (ISSDM), 2012 6th International Conference on New Trends in, pp. 760-765. IEEE, 2012.

[27] Muhammad Afzal, **Maqbool Hussain**, Wajahat Ali Khan, Sungyoung Lee, and Hafiz Farooq Ahmad. "Social media canonicalization in healthcare: Smart cdss as an exemplary application." In e-Health Networking, Applications and Services (Healthcom), 2012 IEEE 14th International Conference on, pp. 419-422. IEEE, 2012.