Acquiring Guideline-enabled data driven clinical knowledge model using formally verified refined knowledge acquisition method

Maqbool Hussain[1,2,*],Muhammad Afzal[1,2], Khalid M. Malik[2,*],Taqdir Ali[3],Wajahat A. Khan[4] , Muhammad Irfan[5,6], Arif Jamshed[5], Sungyoung Lee[3,*]

[1] Department of Software, Sejong University, South Korea; {maqbool.hussain, mafzal}@sejong.ac.kr
[2] Department of Computer Science and Engineering, Oakland University, Rochester, MI, USA; {maqboolhussain, mahmood}@oakland.edu
[3] Department of Computer Engineering, Kyung Hee University, Seocheon-dong, Giheung-gu, Yongin-si, Gyeonggi-do, Republic of Korea, 446-701; {taqdir.ali, sylee}@oslab.khu.ac.kr
[4] College of Engineering and Technology, Universty of Derby, Derby, UK; w.khan@derby.ac.uk
[5] Department of Radiation Oncology, Shaukat Khanum Memorial Cancer Hospital and Research Centre, 7A Block R-3, M.A.Johar Town, Lahore, Pakistan, 54782; drmhammadirfan@gmail.com, arifj@skm.org.pk
[6] Department of Radiation Oncology, National Guard-Health Affairs, King Abdulaziz Medical City Riyadh, Kingdom of Saudi Arabia
* Correspondence: maqbool.hussain@sejong.ac.kr; sylee@oslab.khu.ac.kr; mahmood@oakland.edu;

Abstract:

Background and Objective: Validation and verification are the critical requirements for the knowledge acquisition method of the clinical decision support system (CDSS). After acquiring the medical knowledge from diverse sources, the rigorous validation and formal verification process are required before creating the final knowledge model. Previously, we have proposed a hybrid knowledge acquisition method with the support of a rigorous validation process for acquiring medical knowledge from clinical practice guidelines (CPGs) and patient data for the treatment of oral cavity cancer. However, due to lack of formal verification process, it involves various inconsistencies in knowledge relevant to the formalism of knowledge, conformance to CPGs, quality of knowledge, and complexities of knowledge acquisition artifacts.

Methods: This paper presents the refined knowledge acquisition (ReKA) method, which uses the Z formal verification process. The ReKA method adopts the verification method and explores the mechanism of theorem proving using the Z notation. It enhances a hybrid knowledge acquisition method to thwart the inconsistencies using formal verification.

Results: ReKA adds a set of nine additional criteria to be used to have a final valid refined clinical knowledge model. These criteria ensure the validity of the final knowledge model concerning formalism of knowledge, conformance to GPGs, quality of the knowledge, usage of stringent conditions and treatment plans, and inconsistencies possibly resulting from the complexities. Evaluation, using four medical knowledge acquisition scenarios, shows that newly added knowledge in CDSS due to the additional criteria by the ReKA method always produces a valid knowledge model. The final knowledge model was also evaluated with 1229 oral cavity patient cases, which outperformed with an accuracy of 72.57% compared to a similar approach with an accuracy of 69.7%. Furthermore, the ReKA method identified a set of decision paths (about 47.8%) in the existing approach, which results in a final knowledge model with low quality, non-conformed from standard CPGs.

Conclusion: ReKA refined the hybrid knowledge acquisition method by discovering the missing steps in the current validation process at the acquisition stage. As a formally proven method, it always yields a valid knowledge model having high quality, supporting local practices, and influenced by standard CPGs. Furthermore, the final knowledge model obtained from ReKA also preserves the performance – such as the accuracy of the individual source knowledge models.

Keywords: Knowledge acquisition; Clinical practice guidelines; Data driven knowledge acquisition; Cancer treatment plan; Clinical decision support system; Formal verification;

1. Introduction

Trust in the knowledge base is a crucial factor in the adoption of clinical decision support systems (CDSS) used for medical diagnosis and treatment plan [1]. It mainly depends on the reliability of the knowledge source and the consistency of the knowledge acquisition method [2]. There are diverse sources of clinical knowledge, such as patient data, clinical practice guidelines (CPGs), clinical trials, systematic reviews, and even social media. Various knowledge acquisition approaches have been proposed to acquire clinical knowledge from these sources. For example, using machine learning and ontological approaches, knowledge models from patient data are created [3–5], and different cognitive approaches are used to develop knowledge models from CPGs and other medical resources [6–9]. Depending on the requirements, these knowledge models may need to be transformed into different model formats. For example, the knowledge model from CPGs can be converted into computer-interpretable guidelines (CIGs) so that it could be directly plugged into CDSS for inferencing. Furthermore, sometimes it is required that the knowledge acquisition methods transform two different knowledge models (sharing the same domain problem possibly with different sources) into a unified knowledge model. It is critical in knowledge engineering disciplines that each transformation, provided by the designed knowledge acquisition method, shall ensure the two basic requirements:

1. The *transformed knowledge model* is the *valid* representation of the source *knowledge model(s)*.
2. The *transformation process* is *consistent* enough to produce always a *valid knowledge model*.

Figure 1 shows the knowledge transformation with a set of knowledge acquisition methods in general. The two basic requirements, for each transformation, are depicted as necessary questions to be answered at each knowledge acquisition method of transformation. Question 1 reflects the first requirement mentioned above, and the answer is to provide a *validation mechanism* in the knowledge acquisition method. Question 2 represents the second requirement of the knowledge acquisition method, which necessitates the *verification mechanism* in the knowledge acquisition method. In a nutshell, *validation, and verification* are the critical requirements in the CDSS development process to ensure that the knowledge model is valid, and the entire knowledge acquisition method is consistent.

In terms of verification, most of the existing approaches [8,10,11] emphasize the principles of knowledge engineering. However, none of them have focused on the alignment of the verification process to the development processes of CDSS. On the other hand, formal methods are widely used in software engineering disciplines such as verification of program [12], formal modeling for scenario-based requirement specification [13], formal verification of secured online registration protocols [14], and formal verification of web services on cloud infrastructure [15]. Additionally, some attempts were made to use the formal method (Z notation) to express the knowledge base structure and reasoning mechanism in the form of a software architectural style. For example, Gamble et al. [16] applied Z notation to formally model the knowledge base to get the clear distinction of reusability of knowledge, enhanced understandability, and flexibility of specification in comparison to traditional knowledge specification approaches.

This paper introduces the formal verification process, using Z notation, for our earlier proposed hybrid knowledge acquisition method of Smart CDSS [17] – which is intended to produce guideline-enabled data-driven knowledge model. In hybrid knowledge acquisition, we integrated/supported the method with the sophisticated validation process. The knowledge model created for oral cavity cancer was validated based on the well-established validation criteria and test-based validation process. However, the knowledge acquisition method was not formally verified for internal consistency. The adaption of the formal verification process gives an enhanced knowledge acquisition method – which is known as a refined knowledge acquisition (ReKA) method. In ReKA, we use the Z notation. The selection of Z notation was mainly based on its key features such as *data rich formalism, ease in knowledge modeling, and support of tools*. It is important to mention here that the artifacts of the proposed verification process (using z notation) align to the content of a development framework that we have indigenously used for the development of Smart CDSS in the cancer domain. The development framework for Smart CDSS is based on RUP [18,19] and ISO RM-ODP processes [20,21]. To the best of our knowledge, the existing approaches had neither explored the use of Z notations for the verification of knowledge acquisition nor used the formal methods as a method content in a CDSS development framework.

Before the introduction of the ReKA method, the validity of the knowledge model in the proposed Smart CDSS relied solely on the domain experts. They were free to refine the decision paths in the final knowledge model. This freedom in refinement leads to a set of inconsistencies. Examples of some possible inconsistencies

in terms of clinical context of oral cavity cancer that could be introduced into knowledge model; i) domain expert may add inappropriate follow-up treatments to knowledge models – such as treatment surgery followed by radiotherapy for palliative patients that is a deviation from CPGs which suggest follow-up without further treatment and ii) domain expert may add or refine the rule of evaluating next treatment plan for a variable or patient condition that is not readily available or not in use in existing clinical practices – such as evaluating the palliative patient for radiotherapy based on histopathological risk factor perineural invasion (PNI). In the scope of the current study, this refinement produces inconsistency of introducing non-recordable risk factors (outbound refinement as PNI does not exist in the healthcare system).

This paper addresses following research questions: a) Does the introduction of formal verification using Z notation can identify the inconsistencies in the developed knowledge acquisition method concerning standard knowledge resources such as CPG?; b) Does formal verification ensures that knowledge acquisition methods will always maintain the quality of the knowledge?; c) Does the proposed formal verification can prevent inconsistencies occurred due to complexity and freestyle usage of refinement in the knowledge?; d) Is the knowledge model created using ReKA superior to existing hybrid knowledge models in terms of validity, quality, and integration with workflows?

The detailed evaluation shows that the introduction of formal verification has significantly contributed to revealing hidden inconsistencies in earlier proposed hybrid knowledge acquisition method. In the presence of these inconsistencies, the knowledge model evolution is not always guaranteed to be valid. The ReKA method, as a result of the verification, can identify the leading cause of the inconsistencies and guarantees always producing the valid final knowledge model.

The main contributions of this work are as follow:

- Various aspects of Z notation exploited for the knowledge modeling and associated processes are expressed as the inferenceable mathematical models.
- The proofs of the theorem using Z notations provide a comprehensive explanation for checking the consistency of the knowledge acquisition method. These proofs enable detection of hidden inconsistency in the acquisition method (hybrid knowledge acquisition) and provide an additional set of nine criteria to ensure that the enhanced method (ReKA) always produces a valid knowledge model.
- The formal verification activities are streamlined into a particular set of processes that align with various artifacts of Z notation. it is worthwhile to

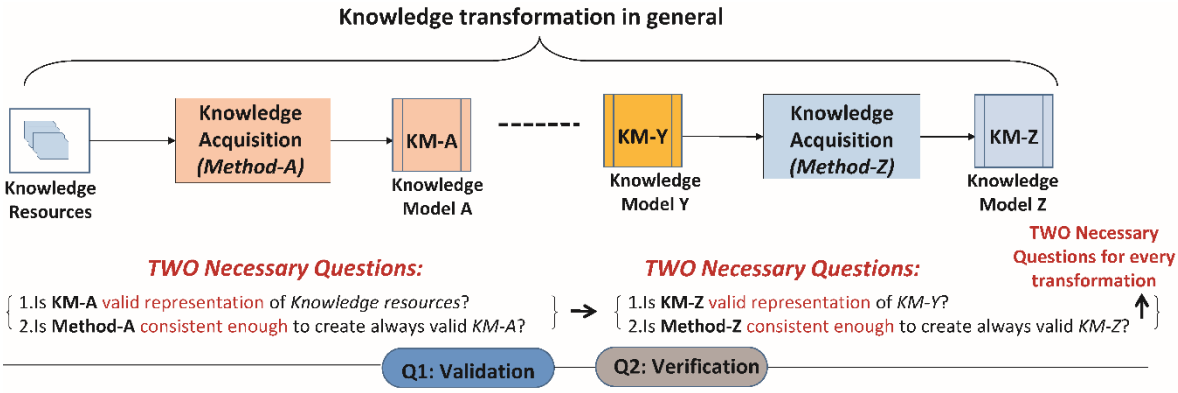


Figure 1: Knowledge acquisition process

2. Overview of knowledge acquisition for Smart CDSS

In our earlier work, we proposed a novel *hybrid knowledge acquisition* method for Smart CDSS [17]. The acquisition method was accompanied by the proper validation process to ensure the validity of the final knowledge model. Before going into details of formal verification, it is worthwhile to briefly introduce the knowledge models and validation processes of the hybrid knowledge acquisition method. We encourage the readers to read [17] for detailed descriptions of the models and validation processes used in the hybrid knowledge acquisition method.

2.1. Hybrid knowledge acquisition approach for Guideline enabled data-driven knowledge model

In the clinical domain, patient data and CPGs are the most common sources of knowledge for CDSS. Most of the existing knowledge acquisition methods use both sources of knowledge independently. From patient data, the knowledge models are created using machine learning, while from CPGs, various cognitive methods of knowledge acquisition apply to the knowledge models. Both methods have potential pros; however, there exist some limitations for each of them. The knowledge acquisition method which combines both approaches can overcome somehow those limitations. The key limitations of data-driven knowledge acquisition methods using machine learning are as follows:

- The quality of the knowledge model depends on the quality of the patient dataset. Therefore, the performance of the model (such as accuracy) may vary for the same domain with different datasets.
- The model validation relies on the statistical validation process (e.g., 10-fold cross-validation). In this case, the validation purely depends on data; and the domain experts are unable to assert any additional criteria to apply constraints on the final knowledge model.
- The final knowledge model supports only local evidence as it is derived from patient data. The recommendation becomes trustworthy for another organization if standard evidence from CPGs and other published studies also associate with the data-driven knowledge model.

The use of CPGs as a knowledge source somehow resolves the inherent problems with the data-driven approach. CPGs covers population-based knowledge supported by standard clinical evidence gathered from different clinical studies. Although it overcomes some cons of the data-driven approach, however, the knowledge models derived from CPGs also come with at least the following limitations:

- CPGs are generic, and the model representing CPGs may not be able to integrate into health-care work-flows directly.
- The knowledge model strictly conforming to CPGs discourages local practices. In most cases, it is possible that local practices may not fully conform and contradict to CPGs, but may have a considerable impact on patient care at that particular region.

Very few studies include CPGs and patient data as a combined source for hybrid knowledge modeling. For example, Toussi et al. [22] used a model derived from patient data to complete the missing decisions in the CPGs. However, the primary motivation of *hybrid knowledge acquisition* method is to combine the data-driven knowledge acquisition method and CPGs based knowledge acquisition method to dilute their cons and take advantages of their pros in terms of the refined knowledge model. This knowledge acquisition method is adopted under the umbrella of the three-phase iteration process model of creating an executable knowledge model for Smart CDSS [17] in the cancer domain. The first two phases of the process model dedicated to knowledge acquisition, which covers knowledge model creations from CPGs and patient data and the validation process. The third phase focuses on the executable knowledge model and the development of associated toolset [23]. Figure 2 depicts the abstract representation of hybrid knowledge acquisition method, and the next section provides a brief description of the core knowledge models and validation process of this approach.

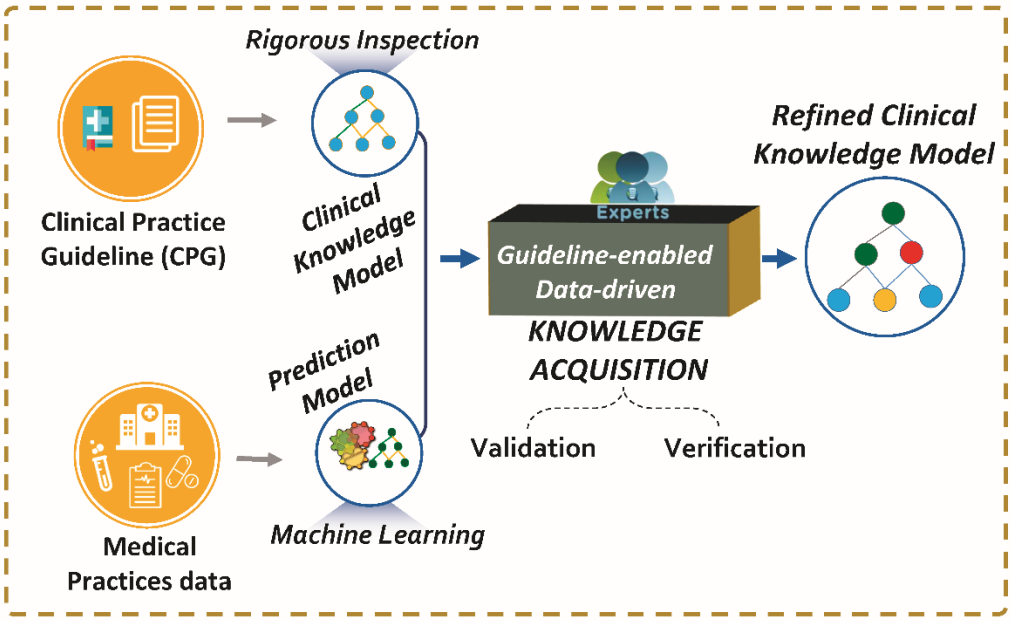


Figure 2: Hybrid knowledge acquisition method

2.2. Knowledge models and validation mechanism

Hybrid knowledge acquisition method includes a set of tasks encompassing two phases of the iterative three-phase model [17]. In this section, we briefly describe the knowledge models and the process associated with the validation of the models (see Figure 2). The outcome of this method is the final knowledge model - known as a refined clinical knowledge model (R-CKM), which is obtained after the rigorous validation process. It consumes the knowledge models created from CPGs - known as a clinical knowledge model (CKM) and prediction model (PM) created from patient data.

Prediction Model: A PM is a decision tree obtained from patient data using decision tree algorithms. The decision tree algorithm used for this study was CHAID [24], which was selected based on rigorous selection criteria. The selection criteria were based on the consensus of the domain experts, which was intended to create a PM with high accuracy and providing a minimal set of decision paths by involving fewer dominant condition attributes. These criteria were translated into a quantitative measure using the weighted sum model. As a decision tree formalism of the machine learning paradigm - it includes the root node and grows in a top-down fashion. The nodes represent conditions and leaf nodes as conclusions. The conclusion always lies at the leaf node where the branch selection at each condition uses proper statistical evaluation processes to proceed for the appropriate decision path. Finally, performance (such as accuracy) for each decision path evaluates from patient data, and its overall performance represents as mean accuracy of all the decision paths in PM.

Clinical Knowledge Model: A CKM is a formal decision tree created from CPGs after a rigorous inspection process by a team of physicians. It follows decision tree formalism started with a root node. The tree grows in a top-down fashion from the root node by adding subsequent nodes to make a decision path. The nodes represent a decision node and a conclusion node. The decision node represents condition(s) (such as patient symptoms) to select the next branch of the tree among decision paths. The conclusion node reflects the recommendations (such as treatment plan). In CKM, the conclusion node can also play the role of condition node for the next follow-up conclusion. For example, an initial treatment plan for cancer patients may be surgery, and after follow-up, the secondary treatment plan can be radiotherapy only if surgery is already done. In this context, unlike the decision tree formalism of PM, the conclusion node may appear as an intermediate node in the CKM decision tree. Moreover, the branch selection of the CKM decision tree does not follow any probabilistic evaluation of the condition because CKM is a reference model of CPGs, so its performance evaluation against local patient data is not required.

Refined Clinical Knowledge Model: A R-CKM obtained after a rigorous validation process by combining PM and CKM. It follows the same formalism as of CKM. However, it also reflects some of the properties of PM to encourage decision making from local practices. Unlike CKM, all decision paths in R-CKM evaluated from local patient data, and it also requires evidence for decision paths that are refined but have no direct conformance to the CKM (i.e., guidelines).

Validation Process: A validation process is the core of the hybrid knowledge acquisition method, which unifies two different models to a single refined knowledge model. Figure 3 depicts detailed steps of the process. It consumes PM and CKM as an input model and produces R-CKM as an output model. Each decision path in PM is selected and added to the decision path of R-CKM after passing conformance criteria based on CKM. The PM decision path may be refined by domain expert if required. The activities for the validation process briefly summarized in three steps:

1. *Setting validation criteria:* Domain experts define criteria based on CKM (guidelines) and other evidence to be fulfilled by the decision path in PM. At the same time, each criterion is classified as primary (compulsory) or non-primary (optional with an alternate), and the order of checking specifies by priority. In the case of an oral cavity cancer treatment plan, domain experts decided two primary and two non-primary criteria. i) The minimum performance limit ***must*** be satisfied by each selected decision path in PM (e.g., an accuracy greater than 50% in this study); ii) the selected decision path in PM ***must not*** conflict with the CKM (guidelines); iii) the decision path in PM ***should*** conform to any decision path in CKM, and iv) ***if*** criterion iii) is not fulfilled, then the decision path in PM ***must*** be associated with an evidence which proves its necessity and effectiveness of inclusion into R-CKM.

2. *PM validation against criteria:* During this step, each decision path is selected and evaluated against the well-established criteria. The decision path of PM becomes part of R-CKM if it fulfills the criteria.

3. *Inspection and refinement of selected PM decision path:* The selected decision path can become directly part of R-CKM. However, the domain expert may want to refine it further to reflect the most concrete concepts used in the healthcare workflows. Moreover, the refinement process also allows adding further choices of the treatment plan in the decision path if required.

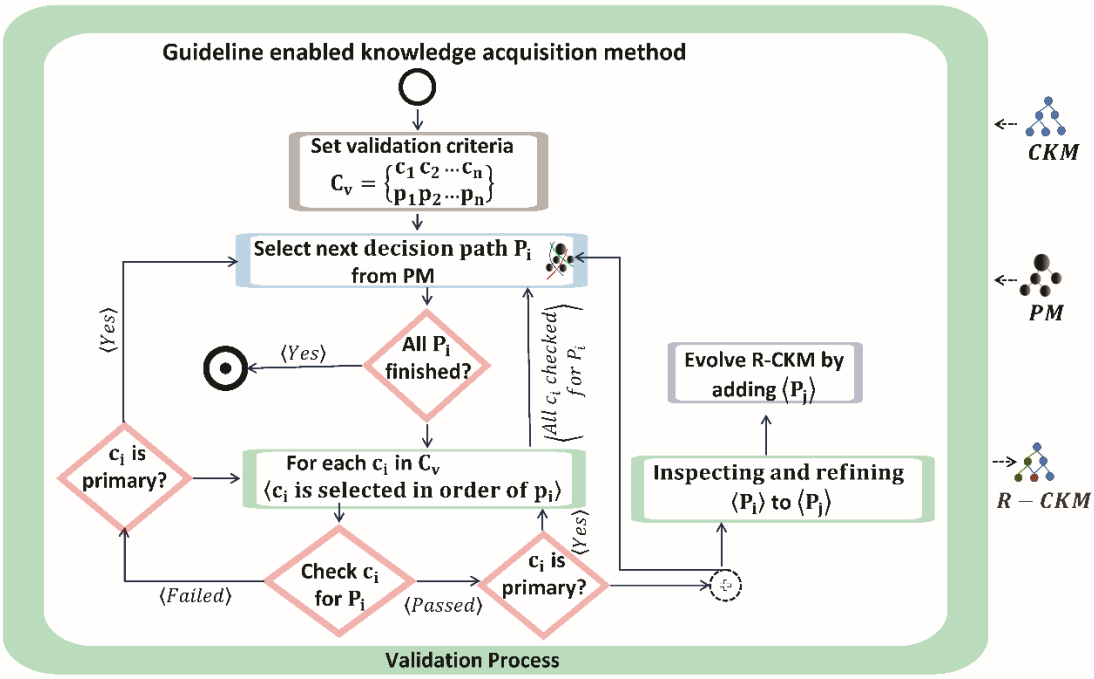


Figure 3: Validation Process [17]

The hybrid knowledge acquisition method was used in the creation of the knowledge model for Smart CDSS in oral cavity cancer [17,23] with proper validation mechanism. However, it was not formally verified even after using it as a core method of knowledge acquisition for Smart CDSS. In the validation process at the refinement step, the process provides the freedom to domain experts for adding further treatment plans as a condition to the selected decision path. It leads toward inconsistency and cannot guarantee the validity of R-CKM at all times. In order to cope with this issue, the content of this work introduces the verification process using formal Z notations. The formal processes are aligned with the existing knowledge acquisition method and presented as an enhanced ReKA method.

3. Preliminaries and key motivation of using the formal method for knowledge acquisition

3.1. Preliminaries

There are several ways to represent objects in the 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 these types obey mathematical laws and have rules for reasoning with the information that they contain. The important concepts are briefly introduced below. Figure 4 outlines all other concepts used in this paper. For a detailed review of these concepts, see [25,26] and other research works that have used Z notation extensively [27–29].

Declaration: This is the simplest way to define an object. When an object is a set of some basic type, brackets are used to enclose the name of an object. If there are more than one objects, comma is used for separation between them. For example, type definition (1) in Figure 5 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*.

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 Figure 5 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 provided to a patient in a proper sequence by using the guidelines.

Axiom: Axiom provides the ability to define objects and includes constraints upon it. In an axiomatic definition, the object definition represents in two compartments: declarations and predicates. Declarations represent the content structure of an object and predicates introduce constraints on the contents. Figure 6 shows an example of the axiomatic definition for CKM specification.

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. Figure 6 shows an example of modeling CKM as a "*ClinicalKnowledgeModel*" schema.

Definitions and declarations		Relations	
a, b	Identifiers	$A \leftrightarrow B$	Binary relation
p, q	Predicates	$\mathrm{dom}\ R$	Relation domain
s, t	Sequences	$\mathrm{ran}\ R$	Relation range
x, y	Expressions	$R^{\sim}$	Relational inverse (relational transpose)
A, B	Sets	$A \lhd R$	Domain restriction
R, S	Relations	$A \rhd R$	Range restriction
d; e	Declarations	$A \ntriangleleft R$	Domain subtraction (Anti-domain restriction)
a == x	Abbreviated definition	$A \ntriangleright R$	Range subtraction (Anti-range restriction)
[A]	Given set	$R \oplus S$	Relation overriding
$A ::= b\langle\langle B\rangle\rangle \mid c\langle\langle C\rangle\rangle$	Free type declaration	$R \times S$	Cartesian product
let *a == x*	Local variable declaration	$a \mapsto b$	Maplet (order pair: same as (a,b))

Logic		Sequences	
$\neg p$	Logical negation	$\mathrm{Seq}\ A$	Sequence
$p \wedge q$	Logical conjunction	$\mathrm{seq}_1\ A$	Non-Empty sequence
$p \vee q$	Logical disjunction	$\langle\ \rangle$	Empty sequence
$p \Rightarrow q$	Logical implication	$\langle x, y, \ldots \rangle$	Sequence
$p \Leftrightarrow q$	Logical equivalence	$s \frown t$	Concatenation
$\forall x: p$	Universal quantification	$A \upharpoonleft s$	Extract ($\{1,3,6\} \upharpoonleft \langle a, b, c, d\rangle$ will give $\langle a, c\rangle$)
$\exists x: q$	Existential quantification	$s \upharpoonright A$	Filter ($\langle a, b, c, d\rangle \upharpoonright \{b, d, e\}$ will give $\langle b, d\rangle$)
$\exists_1 x: q$	Existential quantification (exactly one element)	s in t	Is in ($\langle a,b\rangle$ in $\langle w,y,a,b,c\rangle$ is true. $\langle b,a\rangle$ in $\langle w,y,a,b,c\rangle$ is false)
if *p* **then** *q* **else** *r*	structural conditional logic		

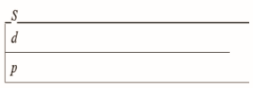
Sets		Schema Notation	
$x \in y$	Set membership		Schema
$\{\ \}$	Empty set		Axiomatic definition
$\mathbb{N}$	Set of natural numbers		
$A \subseteq B$	Set inclusion		
$\{x,y,\ldots\}$	Set of elements		Schema inclusion
$(x,y,\ldots)$	Ordered tuple		
$\mathbb{P}A$	Power set		
$\mathbb{P}_1 A$	Non-empty power set		
$A \cap B$	Set intersection		
$A \cup B$	Set union		
$A \setminus B$	Set difference	ΔS	Change in schema
$\bigcup A$	Generalization union	ΞS	No schema change
$\bigcap A$	Generalization intersection	$S \mathrel{\widehat{=}} T \wedge V$	Schema definition as value of schema expression
$\#A$	Size of finite set	$a?$	Input to an operation
$\{d;e\ldots \mid p \bullet x\}$	Set comprehension	$a!$	Output of an operation
Functions		a'	State component after operation
$A \rightarrow\!\!\!\!\!+ B$	Partial function	S'	State schema after operation
$A \longrightarrow B$	Total function		

Figure 4: Z notation concepts overview

Abbreviation: Abbreviation introduces another name to an existing object. For example, type definition (2) in Figure 5 is the abbreviation for cancer treatments.

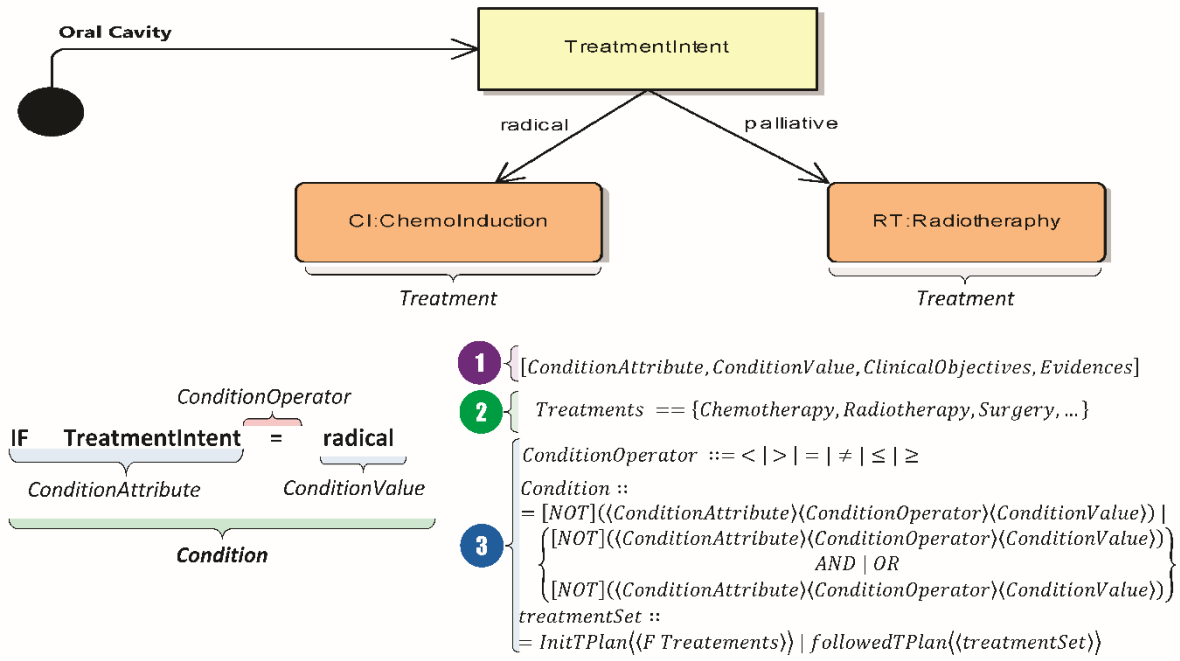


Figure 5: Declaration, abbreviation and free type examples

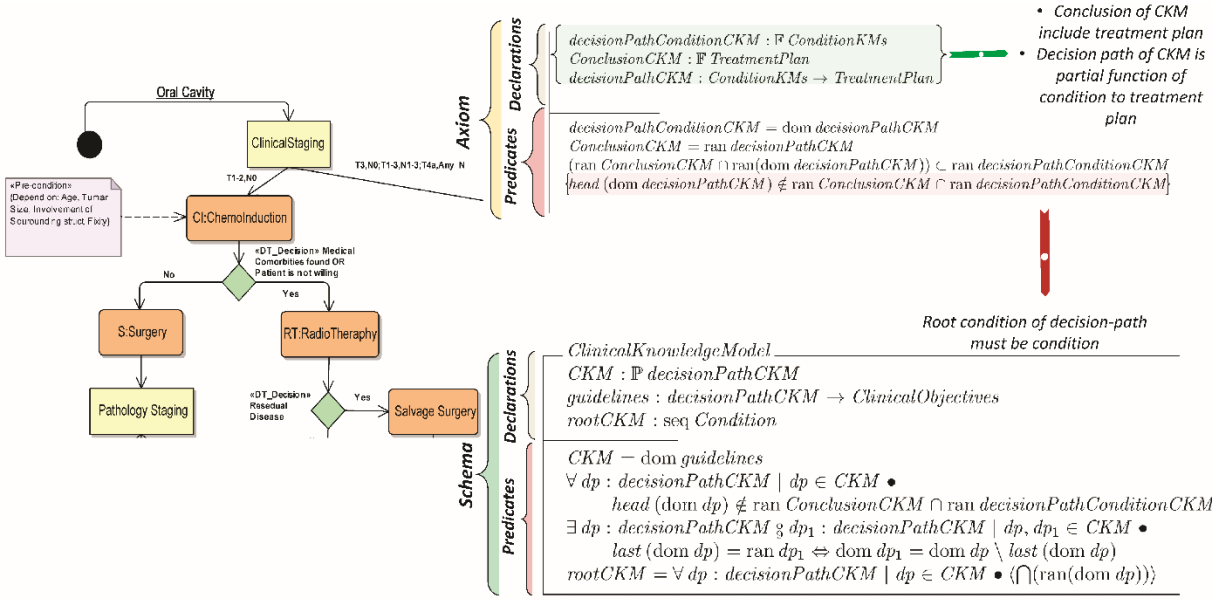


Figure 6: Axiomatic definition and schema example

3.2. Motivation: Formal methods for Knowledge Verification

The ability of domain experts to trust knowledge content is a key factor that influences the success of CDSS implementation. The trust in knowledge primarily depends on how well the knowledge contents have passed through a sophisticated validation process to ensure consistency in the refined knowledge model. According to a systematic review by Mor Peleg [2], formal verification techniques are used to validate the clinical knowledge for internal consistency and to check for the fulfillment of the desired properties and specifications. There are two broad categories of these techniques: model checking and theorem proving [2]. 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 fulfillment of the desired properties. Alessio Bottrighi et al. applied the model checking approach to integrating the computerized guideline management system [30]. The guideline representation language GLARE is used and integrated with the SPIN model checker

to verify the clinical guidelines. Theorem proving uses the logical derivation of theorems in order to prove the consistency of the knowledge contents available in the formal specification. Annette T. Teije et al. [31] used KIV-based formalism to represent medical protocols and defined semantics of the desired properties. The desired properties of the protocol are verified using formal proof of the KIV theorems.

Based on the substantial advantages and the need for formalism in knowledge validation and verification, we introduce the formal verification process as a formal method content into the development framework of Smart CDSS. Selection of an appropriate formal method requires formal guidelines to find the best fit for a knowledge representation scheme. In this work, we used the Z notation as the formal representation language for knowledge representation and for modeling the validation method features. Following are fundamental features of Z notation, which compels its suitability for clinical knowledge modeling and verification of the acquisition process.

1. *Easy knowledge modeling:* While using the 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 process [25]). The "Schema" represents this aspect of Z notation, where the first-order predicate logic uses the constraints on the typed knowledge contents. Moreover, dynamic schema represents the validation process that operates within the boundaries of the knowledge representation schema. The subsequent sections will elaborate, detailed contents of the formal verification process for the knowledge acquisition method in terms of Z specifications.
2. *Data-rich formalism:* Another aspect of Z notation is the notion of "types" [26]. Z types are mathematical data types that can be used to represent any object in a system uniquely. They specifically obey a rich collection of mathematical laws, which make it possible to determine the behavior of the system [25,26]. 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 [32] (e.g., the information viewpoint). H. Bowman et al. used Z notation for consistency checking of the two views in the information viewpoint [33]. Similarly, artifacts of Z notation can also map 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 [27]. Moreover, there is a wide range of tools available to check for syntax and type consistency in the specifications.

4. Methods

4.1. Refined Knowledge Acquisition (ReKA) method

ReKA uses all the steps of hybrid knowledge acquisition described in section 2.1. Besides, it introduces new processes that involve the formal verification artifacts at different phases of the three-phase model. Figure 7 shows the extended three-phase model used by the ReKA method. The extended processes are reflected as an additional layer of the underlying processes.

This study focuses on the newly adopted processes of formal verification, so we skip details of the common process used with hybrid knowledge acquisition. The model created for oral cavity cancer in the earlier study is reused for this study with new patient cases of 1229 from Shaukat Khanum Memorial Cancer Hospital (SKMCH), Lahore, Pakistan. Example scenarios have been created by physicians to modify our earlier oral cavity treatment model. Based on the earlier hybrid knowledge acquisition method, the modifications are valid; however, as demonstrated in the results section, ReKA identifies that the modifications are not valid because it creates inconsistencies in the final knowledge model.

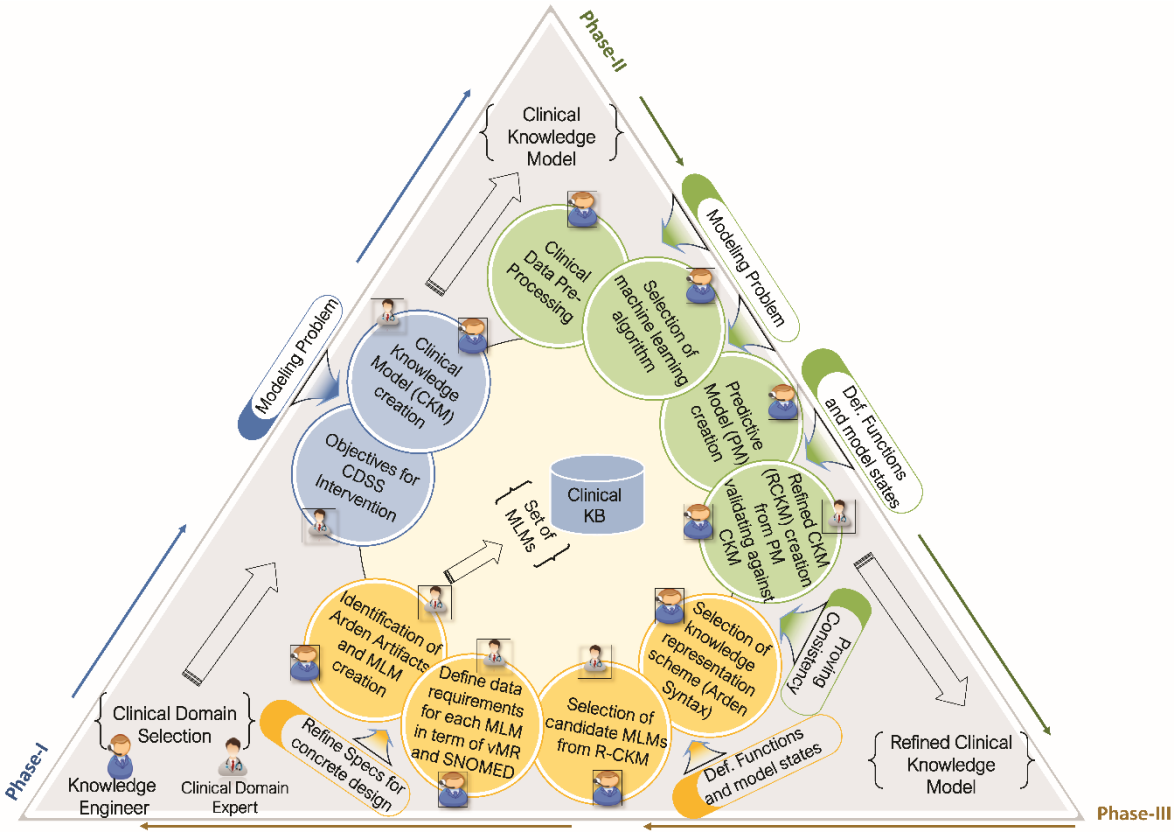


Figure 7: Extended three-phase model for ReKA

4.2. Establishing a formal modeling process

To the best of our knowledge, no substantial evidence exists in a knowledge engineering discipline that discusses Z notation with discrete processes having proper guidance. Based on the capabilities of Z notation and the guidance available for applying different concepts of Z notation to formal modeling [25,26], we formulate a formal modeling process for knowledge acquisition method. It comprises 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. Figure 8 shows an abstract view of these processes.

1. *Modeling problem*: This includes tasks used to analyze the problem context and identifies all the relevant concepts that contribute to the final objectives. Different constructs of the selected formalism technique are used to model concepts at different granularity levels. Primitive types, axioms, free types, and schema are the candidate constructs in Z notations that assist in modeling the problem under consideration. During the knowledge acquisition method, various models were created such as PM, CKM, and R-CKM. Different constructs of Z notation were used in representing these models. The outcomes of this process produce primitive types, free types, sets of axioms, and sets of the static schema, which represents the knowledge models.

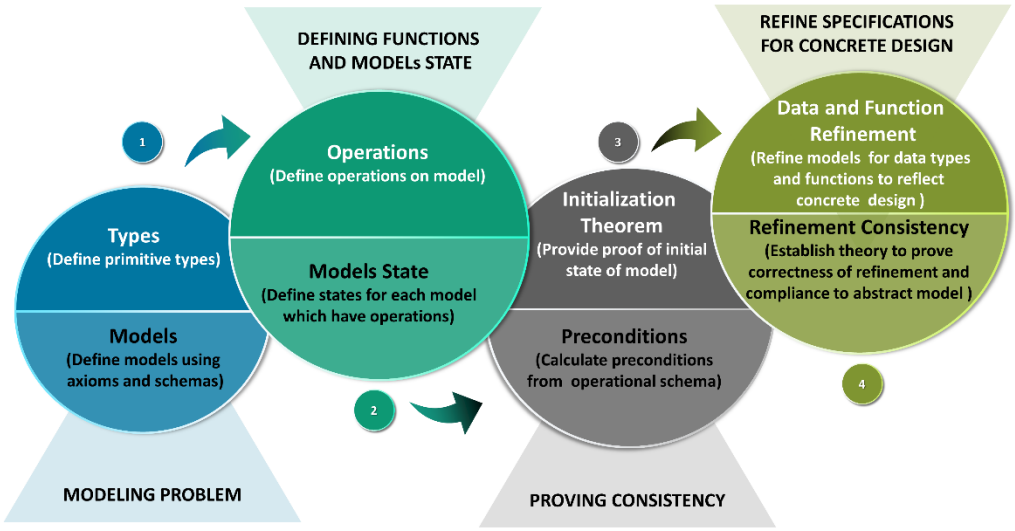


Figure 8: Formal modeling process

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 central construct in Z and can represent the operations and states of the models. For the knowledge acquisition method, operations are defined for the retrieval of contents from PM, and CKM models. These operations will not affect changing the state of the corresponding models. Different operations define for 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 is desirable to verify that the operations defined in various models are consistent and that their outcomes are within the intended boundaries of the domain. Z specification provides a well-established way to achieve both goals. The first part achieves, 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 the operational schema using the one-point rule. For the knowledge acquisition method, the "*initialization theorem*" proves the satisfiability of the R-CKM state schema. Moreover, "*preconditions*" investigate for all 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 comprises large tasks that are applied in consecutive iterations at the data and function levels to ensure that the specifications are free of any uncertainty. These specifications are closer than previously modeled specification to 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 and the validation process in order to prove that the knowledge acquisition is sufficiently consistent with always producing valid final knowledge models. The refinement process is helpful for systems where the outcomes of the design are required to be sufficiently close for direct conversion into executable code. This process is

included purposefully because our knowledge specification can be easily converted into the executable code if we properly exploit the Z refinement mechanism. Furthermore, we are presenting the "*Proving consistency*" step in the results section to emphasize the outcome of the formal verification process.

4.3. Modeling problem

The modeling problem investigates the basic concepts used in knowledge acquisition for Smart CDSS. The fundamental concepts used in Smart CDSS are PM, CKM, and R-CKM, which represent the clinical treatment plan for head and neck cancer. Primitive types, free types, axioms, and schema in Z notation are candidate constructs to represent these concepts.

4.3.1. Primitive types

Primitive types constitute the basic building blocks of the problem 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, a condition in the decision tree test node *TreatmentIntent* = *radical* represents a patient categorization primarily based on the severity of cancer. Z primitive types (shown in Type Definition 1 (line 2) represents these concepts, and Type Definition 2 provides the corresponding language syntax for the condition.

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 reflects the semantics of the clinical concepts and provides conformance to decision tree representation formalism. For example, treatments provided to patients follow a sequence according to standard guidelines and protocols; ChemoInduction follows radiotherapy treatments and surgery 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).

Type Definition 3 Free types to capture semantics of knowledge artifacts

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

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

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

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

In Smart CDSS, the knowledge model typically uses decision tree representation; however, PM is different from CKM and R-CKM in terms of the decision path. PM does not include treatments as a condition. To distinctly represent this formalism, *ConditionCKs* (line 3) defines a particular condition as a free type for CKM and R-CKM. Similarly, *RefinedTreatmentPlan* (line 4) represents a refinement in final R-CKM, which dictates the addition of a treatment to R-CKM as a type of refinement (indicating the placement of treatment plan at a particular position in the decision path).

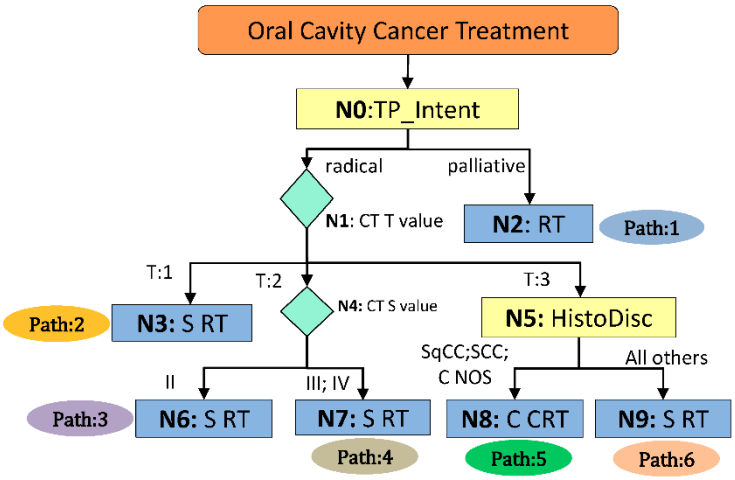
4.3.2. Knowledge models

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 by decision tree formalism. Figure 9 shows the PM created (using CHAID decision tree) for oral cavity cancer treatment intervention [17] with details of corresponding attributes and their formalism semantics. The PM specifications are created using an axiom (Axiom 1) and the *PredictionModel* schema (Schema 1). The axiomatic definition for PM represents the basic constructs of PM 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).

Axiom 1 Prediction model specifications

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



Attribute	Formalism	Remarks
RootNode	Only one; starting node; condition node	TP_Intent
LeafNode(s)	One or more; End of decision path; indicate conclusion	RT, S RT, C CRT
Condition Node(s)	One or more; indicate condition; must not be leaf node	CT S value; CT T value; HistoDisc
Decisionpath	Set of nodes (conjunction) with conclusion: start with root node, set of condition nodes, ends with leaf node	1.TP_Intent: palliative → RT; 2. TP_Intent: radical → CT T value: T:1 → S RT ; (4 other paths)
Decisionpath Accuracy	Each path must have accuracy in range of 0 and 100	Decisionpath 1: 40.6 %; Decisionpath 2: 95.7 %
PM Accuracy	Mean accuracy of all decision paths; PM accuracy must be in range 0 and 100	PM accuracy for 6 paths: 59 %

Figure 9: PM for treatment intervention in oral cavity cancer and its formalism (C CRT: Chemo Induction followed by chemotherapy; C NOS: Carcinoma NOS; CT S: clinical stage S value; CT T: clinical stage T value; HistoDisc: Histology description; RT: radiotherapy; SCC: Small cell carcinoma; SqCC: Squamous cell carcinoma; S RT: Surgery followed by RT; TP_Intent: Treatment Plan Intent)

Schema 1 Prediction model specifications

PredictionModel

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

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

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 by 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 represented by 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. The evidence is a finite set (line 6), which can represent the effectiveness of the treatment plan in given patient cases in terms of the success rate (as a percentage). It may also include external evidence from other research works. Therefore, the decision path may have evidence represented by a partial function from the decision path to the set of evidence (line 7 and line 12).

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, 3, 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). Also, PM is a decision tree, which means it must include one root node that must be a condition (lines 5 and 9).

Clinical knowledge model 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 using a decision tree formalism. Figure 10 is reference CKM created from clinical guidelines [17]. For the brevity purpose, we are not displaying the pictorial representation of the formalism as it shares most of the structure artifacts with the R-CKM and hence Figure 11 shows a formalism used as a reference for CKM.

Axiom 2 Clinical knowledge model specifications

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

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

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 represented by a partial function from free type *ConditionKMs* to the treatment plan with axiomatic definition Axion 2 (line 3). The constraint defined by a predicate at Axiom 2 (line 6) reinforces the idea of the CKM decision path that may contain treatment plans in condition. Moreover, every decision path must have a starting condition other than a treatment plan, which defined by a predicate at Axiom 2 (line 7). Axiom 2 (line 1,4 and 2,5) are representing the conditions (*decisionPathConditionCKM*) and conclusion (*ConclusionCKM*) of decision path in CKM as finite set of *ConditionKMs* and *TreatmentPlan* respectively.

Schema 2 Clinical knowledge model specifications

ClinicalKnowledgeModel

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

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

The *ClinicalKnowledgeModel* schema (Schema 2) further extends the CKM semantics. According to the definition of CKM, it covers-up the guidelines and follows decision tree formalism. Furthermore, it is associated with clinical objectives. For example, CKM (in Smart CDSS) consults NCCN guidelines, and its main objective is the provision of 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 fulfill 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 (line 3) shared by all decision paths. Schema (Schema 2) defines these constraints by predicates at (lines 5 and 7).

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. Schema 2 defines a predicate (at line 6) to reflect this semantic.

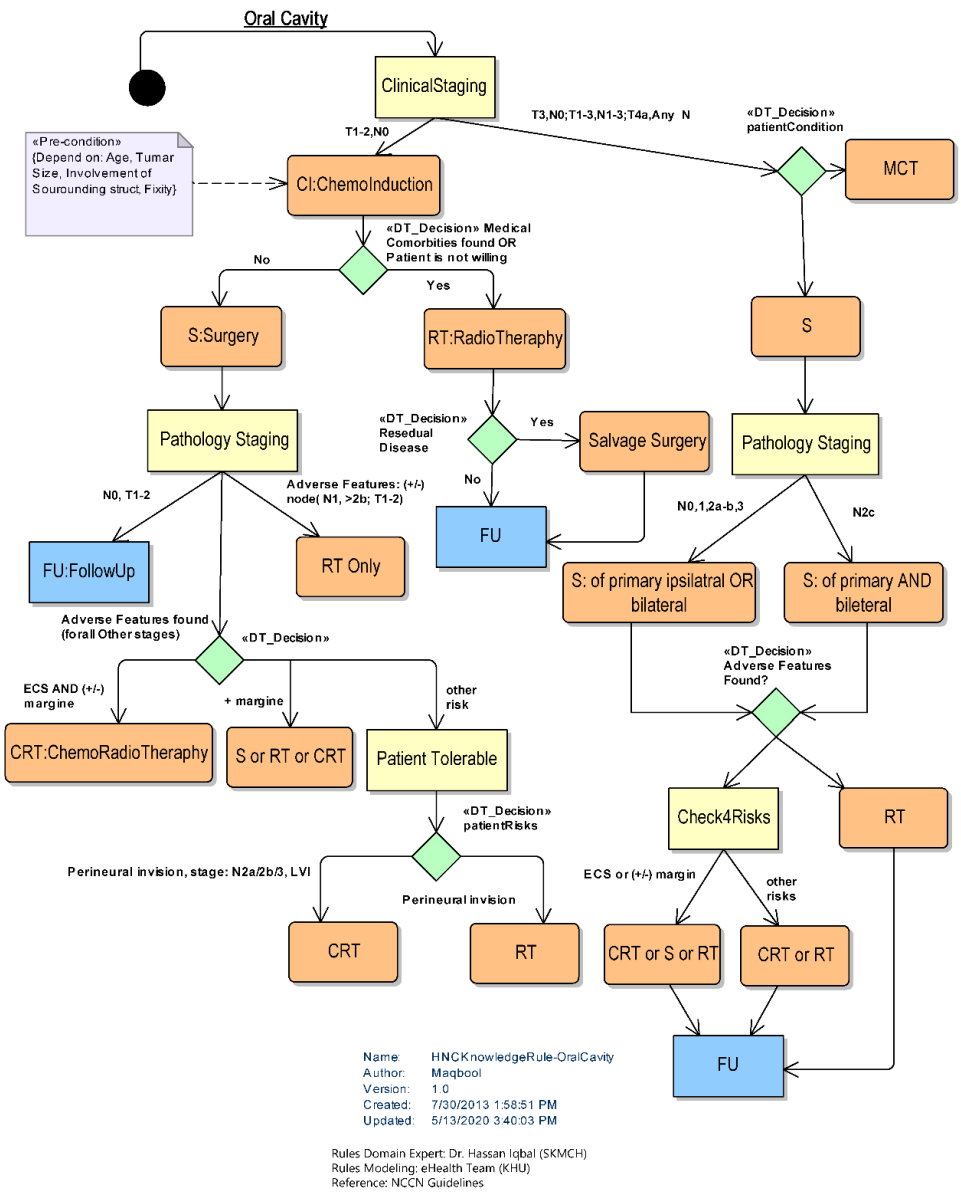


Figure 10: CKM for treatment intervention in oral cavity cancer(CI: Chemoinduction; CRT: Chemotherapy; CT N: clinical stage N value; CT S: clinical stage S value; CT T: clinical stage T value;

ECS: Extracapsular spread; FU: Follow-up; MCT: Multidisciplinary consultation; RT: radiotherapy; S: Surgery) [17]

Refined clinical knowledge model 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. Figure 11 shows the R-CKM of a treatment plan for oral cavity cancer [17] with precise semantics and formalism. In this respect, the R-CKM decision path modeled (similar to CKM) by a *partial function* from free type *ConditionKMs* to the treatment plan; this is shown in the axiomatic definition (line 3).

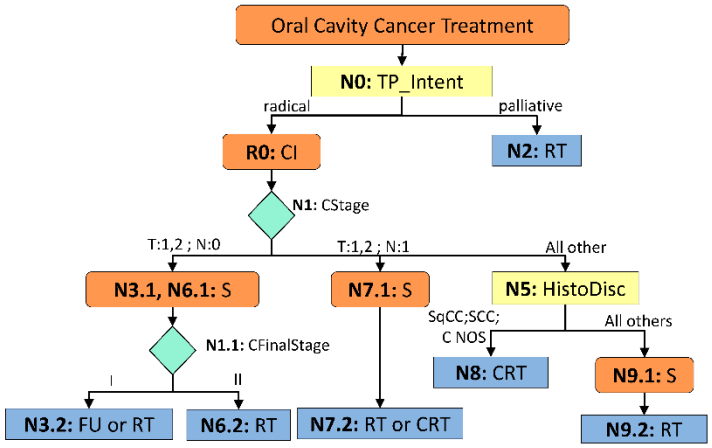
Axiom 3 Refined clinical knowledge model specifications

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

As a result of refinements, the decision path of R-CKM may fully not conform to guidelines (CKM). In such cases, the evidence is required to justify the effectiveness of the refinements made to the decision path of R-CKM. To capture this context, a finite set of *Evidences* (line 6) is associated with the decision path of R-CKM as a *partial function* (line 7,15).

Schema 3 Refined clinical knowledge model specifications

$$
\begin{array}{ll}
\multicolumn{2}{l}{RefinedClinicalKnowledgeModel}\\
PredictionModel & (1)\\
ClinicalKnowledgeModel & (2)\\
RCKM : \mathbb{F}\, decisionPathRCKM & (3)\\
refinedCKM : decisionPathRCKM \rightarrow CKM & (4)\\
rootRCKM : \mathrm{seq}\, Condition & (5)\\
accuracyRCKM : \mathbb{F}\, \mathbb{Z} & (6)\\
refinedCKMsAccuracy : RCKM \rightarrow accuracy & (7)\\
\hline
0 \leq accuracyRCKM \leq 100 & (8)\\
RCKM = \mathrm{dom}\, refinedCKM & (9)\\
\forall\, dp : decisionPathRCKM \mid dp \in RCKM \bullet & \\
\quad head\,(\mathrm{dom}\, dp) \notin \mathrm{ran}\, ConclusionRCKM \cap \mathrm{ran}\, decisionPathConditionRCKM & (10)\\
\exists\, dp : decisionPathRCKM \fatsemi dp_1 : decisionPathRCKM \mid dp, dp_1 \in RCKM \bullet & \\
\quad last\,(\mathrm{dom}\, dp) = \mathrm{ran}\, dp_1 \Leftrightarrow \mathrm{dom}\, dp_1 = \mathrm{dom}\, dp \setminus last\,(\mathrm{dom}\, dp) & (11)\\
accuracyRCKM = (\mathbf{let}\ pathsAcc == \{pathsAcc : \mathbb{Z} \mid RCKM \neq \emptyset \wedge & \\
\quad (\forall\, dp : decisionPathRCKM \mid dp \in RCKM \bullet pathsAcc = & \\
\quad decPathRCKMAccuracy(dp) + pathsAcc)\})/\#RCKM & (12)\\
\forall\, p_{rckm} : decisionPathRCKM \mid p_{rckm} \in RCKM \bullet & \\
\quad \exists\, p_{pm} : decisionPath, p_{ckm} : decisionPathCKM \mid & \\
\quad p_{pm} \in PM \wedge p_{ckm} \in CKM \bullet \mathrm{dom}\, p_{rckm} = \mathrm{dom}\, p_{pm} \cup \mathrm{dom}\, p_{ckm} & (13)\\
rootRCKM = \forall\, dp : decisionPathRCKM \mid dp \in RCKM \bullet \langle \bigcap(\mathrm{ran}(\mathrm{dom}\, dp))\rangle & (14)
\end{array}
$$



Attribute	Formalism	Remarks
RootNode	Only one; starting node; condition node	TP_Intent
LeafNode(s)	One or more; End of decision path; indicate conclusion	RT or CRT, FU or RT, RT, CRT
Condition Node(s)	One or more; indicate condition; must not be leaf node; condition node may indicate conclusion which will be condition for other node	Only condition: CStage; CFinalStage; HistoDisc Condition & Conclusion: CI, S
Decisionpath	Set of nodes (conjunction) with conclusion: start with root node, set of condition nodes, ends with leaf node | condition node (act as conclusion)	1.TP_Intent: radical → CI; 2. TP_Intent: radical → CI → Cstage: T:1,2;N:1 → S 3: TP_Intent: radical → CI → Cstage: T:1,2;N:1 → S → RT or CRT (5 other paths)
Decisionpath Accuracy	Each path must have accuracy in range of 0 and 100	Decisionpath 1: 54.2 %; Decisionpath 2: 88.6 % Decisionpath 3: 88.6 %
RCKM Accuracy	Mean accuracy of all decision paths and it must be in range 0 and 100	RCKM accuracy for 8 paths: 53 %

Figure 11: R-CKM for treatment intervention in oral cavity cancer and its formalism (CI: Chemoinduction; C NOS: Carcinoma NOS; CRT: Chemotherapy; CT N: clinical stage N value; CT S: clinical stage S value; CT T: clinical stage T value; ECS: Extracapsular spread; FU: Follow-up; HistoDisc: Histology description; RT: radiotherapy; S: Surgery; SCC: Small cell carcinoma; SqCC: Squamous cell carcinoma; TP_Intent: Treatment Plan Intent)

The predicates defined in Axiom 3 (lines 13, 14) capture the semantics of the decision path in R-CKM; a treatment plan can be 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 become a part of the model after passing through a formal validation process and refinements (Figure 3). In this respect, the decision path in R-CKM has an accuracy represented by a *total function* from the decision path to the accuracy (line 5). Also, the accuracy of the decision path must be a finite value bounded interval [0,100] indicated in line (4,10). An injective function represents the refinement in R-CKM, as shown in (line 9, 16), which maps the refined treatment plan (a free type, line 4, Type Definition 3) to the PM decision path (line 8).

The declarations and predicates of schema *RefinedClinicalKnowledgModel* (Schema 3) are mostly similar to those of CKM (Schema 2); both share the same formalism. A *total function* (line 7) defines the new content to support the overall accuracy of R-CKM. The intended accuracy calculated by 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* defines from the R-CKM decision paths to the intended CKM (line 4), and R-CKM modeled by a finite set of related decision paths (line 3) associated with CKM (line 9). Furthermore, a predicate adds 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.

Validation process specifications: Validation process specifications encompass the validation process (Figure 3) and adequately represent the validation criteria defined for the final knowledge model - R-CKM (See step 1: *Setting validation criteria* in Section 2.2). The schema *PMPathValidation* (Schema 4) models the underlying 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 a 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 and acceptable for R-CKM) that requires the PM decision path (line 3).

Schema 4 Validation process specifications

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

The validation criteria defined in the validation process of the knowledge acquisition method reflected by predicates in the schema *PMPathValidation* (lines 4-7). The first two primaries (compulsory) criteria defined in the schema by conjunction predicates (lines 4 and 5) and two other criteria represented by disjunction predicates (lines 6 and 7).

4.4. Defining functions and state models

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

4.4.1. Operations on knowledge models

Two types of operations defined for the knowledge model. For PM and CKM, only retrieval operations are required to represent access to different components of the model. So for as R-CKM is concerned, it requires specifications for both retrieval and state change operations.

Operations for PM and CKM: PM and CKM specification provide a set of operational schema related retrieval of various components of the PM and CKM, respectively. For the brevity purpose, we concentrate on operational schema related to the evolution of the knowledge model. Retrieval schema for the PM and CKM are straight forward, and we shall not discuss it further.

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.

EvolveRCKM (Schema 5) is an operational schema that mainly represents the evolution of the R-CKM model. The evolution of R-CKM mainly describes as a two-step process after setting the validation criteria: (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.

Schema 5 Evolution of R-CKM

$$\textit{EvolveRCKM} \mathrel{\widehat{=}} \textit{PMPathValidation} \wedge \textit{AddPathRCKM}$$

Accordingly, *EvolveRCKM* (Schema 5) is defined as a composite operational schema to reflect these steps. This composition is modeled as a combination of two schemas: *PMPathValidation* (Schema 4) and *AddPathRCKM* (Schema 6).

To get a clear picture of this process, Figure 12 demonstrates three paths of the PM(Figure 9) in the context of the validation process(Figure 3) and produce the R-CKM(Figure 11). The two paths (path 1 and path 2) are fulfilling the first two compulsory criteria (having a minimum threshold of accuracy without any conflicts with CKM) and passing the criteria regarding conformance to CKM (Figure 10). Path 3 fulfilling the compulsory criteria; however, it goes for alternate criterion "Evidence" because of the suggested treatment plan does not conform to CKM. In the refinement step, path 2 and path 5 are refined to path 2.1, 2.2, and path 5, respectively. So far as path 1 is used without any refinements.

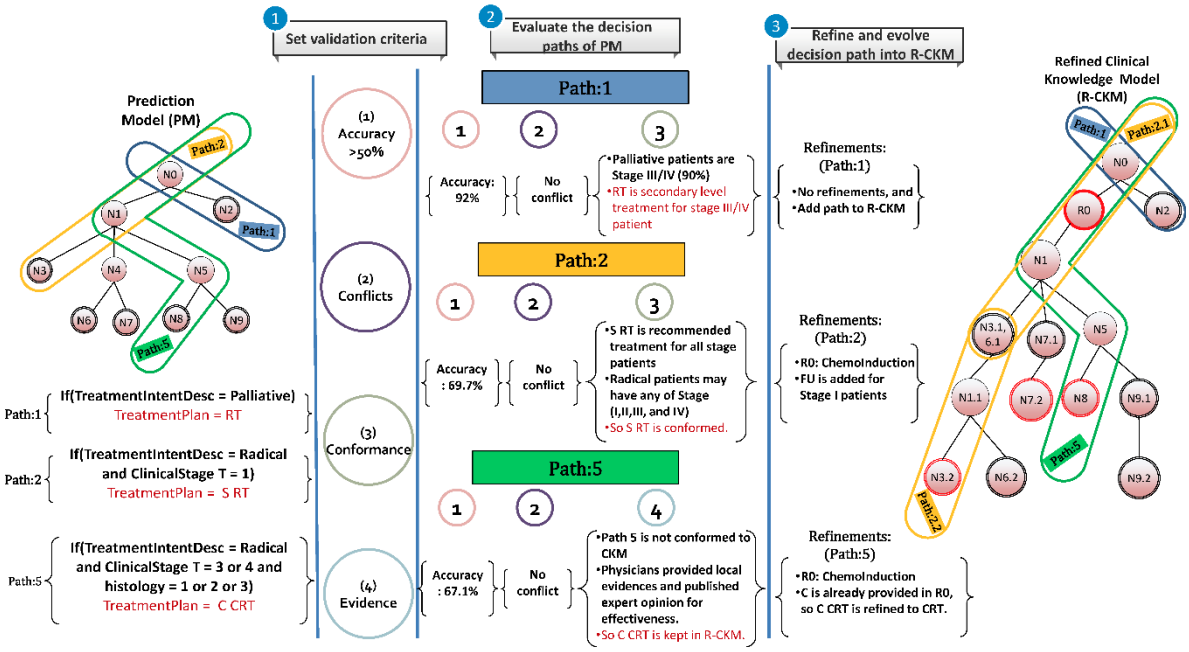


Figure 12: A running example of validation and refinement of three decision paths of PM

AddPathRCKM is the main operational schema (Schema 6) that evolves the R-CKM and changes the original state of the model (Schema 3: *RefinedClinicalKnowledgeModel*), which represented by a change state in the schema (line 1). In order to understand the complexity of the *AddPathRCKM* operational schema, we divide the declarations and predicates into the following explanatory sections:

AddPathRCKM is the main operational schema (Schema 6) that evolves the R-CKM and changes the original state of the model (Schema 3: *RefinedClinicalKnowledgeModel*), which represented by a change state in the schema (line 1). In order to understand the complexity of the *AddPathRCKM* operational schema, we divide the declarations and predicates into the following explanatory sections:

- *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. We shall describe some important pre-conditions, as evaluation results, for the formal verification process in Section 5.

Schema 6 Adding PM decision path to R-CKM

AddPathRCKM

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

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

- *Predicates (Refinements):* The refinement process performed on the candidate decision path of PM (line 14), and the modified path (line 15) according to the necessary treatment plan that is mentioned by the suggested refinements, provided by 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 primarily represent the new change state of the R-CKM model; subsequent sections explain further details.

4.4.2. Model states for knowledge models

Modifications are only made to R-CKM upon evolution through the *EvolveRCKM* (Schema 5) 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 6) 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 following the *AddPathRCKM* operational schema. The schema *RefinedClinicalKnowledgeModel'* (Schema 7) 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. The changes made to the R-CKM model (*RefinedClinicalKnowledgeModel'*: Schema 7) by *AddPathRCKM* operational schema are summarized as follows:

Schema 7 R-CKM state after modification

$$
\begin{array}{ll}
\multicolumn{2}{l}{\textit{RefinedClinicalKnowledgeModel}'} \\
\hline
PredictionModel & (1) \\
ClinicalKnowledgeModel & (2) \\
decisionPathConditionRCKM' : \mathbb{F}\, ConditionKMs & (3) \\
ConclusionRCKM' : \mathbb{F}\, TreatmentPlan & (4) \\
decisionPathRCKM' : ConditionKMs \nrightarrow TreatmentPlan & (5) \\
decPathRCKMAccuracy' : decisionPathRCKM' \rightarrow accuracy & (6) \\
evidences' : \mathbb{F}\, Evidences & (7) \\
decPathRCKMEvidences' : decisionPathRCKM' \nrightarrow Evidences & (8) \\
refinedTPlan' : \mathbb{F}\, RefinedTreatmentPlan & (9) \\
refinementsDecPath' : RefinedTreatmentPlan \rightarrowtail decisionPath & (10) \\
RCKM' : \mathbb{F}\, decisionPathRCKM & (11) \\
refinedCKM' : decisionPathRCKM' \rightarrow CKM & (12) \\
rootRCKM' : \mathrm{seq}\, Condition & (13) \\
accuracyRCKM' : \mathbb{F}\, \mathbb{Z} & (14) \\
refinedCKMsAccuracy' : RCKM' \rightarrow accuracy & (15) \\
\hline
decisionPathConditionRCKM' = \mathrm{dom}\, decisionPathRCKM' & (16) \\
ConclusionRCKM' = \mathrm{ran}\, decisionPathRCKM' & (17) \\
(\mathrm{ran}\, ConclusionRCKM' \cap \mathrm{ran}\, decisionPathConditionRCKM') \subset \mathrm{ran}\, decisionPathConditionRCKM' & (18) \\
head\,(decisionPathConditionRCKM') \notin \mathrm{ran}\, ConclusionRCKM' \cap \mathrm{ran}\, decisionPathConditionRCKM' & (19) \\
evidences' = \mathrm{ran}\, decPathRCKMEvidences' & (20) \\
refinedTPlan' = \mathrm{dom}\, refinementsDecPath' & (21) \\
0 \leq accuracyRCKM' \leq 100 & (22) \\
RCKM' = \mathrm{dom}\, refinedCKM' & (23) \\
\forall\, dp : decisionPathRCKM' \mid dp \in RCKM' \bullet & \\
\quad head\,(\mathrm{dom}\, dp) \notin \mathrm{ran}\, ConclusionRCKM' \cap \mathrm{ran}\, decisionPathConditionRCKM' & (24) \\
\exists\, dp : decisionPathRCKM' \fatsemi dp_1 : decisionPathRCKM' \mid dp, dp_1 \in RCKM' \bullet & \\
\quad last\,(\mathrm{dom}\, dp) = \mathrm{ran}\, dp_1 \Leftrightarrow \mathrm{dom}\, dp_1 = \mathrm{dom}\, dp \setminus last\,(\mathrm{dom}\, dp) & (25) \\
accuracyRCKM' = (\mathbf{let}\, pathsAcc == \{pathsAcc : \mathbb{Z} \mid RCKM' \neq \emptyset \wedge & \\
\quad (\forall\, dp : decisionPathRCKM' \mid dp \in RCKM' \bullet pathsAcc = & \\
\quad decPathRCKMAccuracy'(dp) + pathsAcc)\})/\#RCKM' & (26) \\
\forall\, p_{rckm} : decisionPathRCKM' \mid p_{rckm} \in RCKM' \bullet & \\
\quad \exists\, p_{pm} : decisionPath, p_{ckm} : decisionPathCKM \mid & \\
\quad p_{pm} \in PM \wedge p_{ckm} \in CKM \bullet \mathrm{dom}\, p_{rckm} = \mathrm{dom}\, p_{pm} \cup \mathrm{dom}\, p_{ckm} & (27) \\
RCKM \neq \emptyset \Rightarrow rootRCKM' = rootRCKM & (28)
\end{array}
$$

- A new decision path added to R-CKM; this adds new conditions to the set of R-CKM conditions ((Schema 6: Lines 16 and 17)). These changes represented in the state model (Schema 7) at lines 3, 5, and 16.
- New refinements introduced to a set of the R-CKM model, which results in the addition of a PM path with the associated refinements (Schema 6: Lines 18 and 19). These states reflected in lines 9, 10, and 21 in Schema 7.
- With the new decision path, the R-CKM model evolved for a new conclusion (Schema 6: 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 6: Lines 21, 22, and 23). The resulting state changes reflected at lines 6, 14, 15, 22, and 26 in the state model schema.
- Evidence of the PM's decision path associated with the refined decision path in R-CKM (Schema 6: Lines 24 and 25). These changes reflected in lines 7, 8, and 20 in the state model schema.
- Finally, R-CKM evolved with the addition of a new decision path, and the root condition re-evaluated (Schema 6: 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.

5. Results

This section explains the evaluation of the proposed work using two perspectives. First, it demonstrates the theorem proving mechanism to show inconsistencies in the hybrid knowledge acquisition method before formal verification. The outcome of the formal verification is presented as an enhanced knowledge acquisition method – as the ReKA method. We evaluate the ReKA method (in the context of formal verification) against our initial approach and describes its discrepancies using real clinical scenarios. Second, we compare our enhanced approach with one of the existing relevant approaches developed by Tossie et al. [22].

5.1. Proving consistency of the knowledge acquisition method

The creation of the formal specification for clinical knowledge models using abstract data type (i.e., Z schema), provides the opportunity of two significant proofs that enable consistency of the knowledge acquisition method. The first proof ensures consistency of various clinical domain requirements with the clinical knowledge models. The second proof demonstrates that each evolution operation on clinical knowledge models is never applied outside its specified domain. Both proofs involve mathematical tasks that lead to the overall consistency of the knowledge acquisition method. The first proof is achieved by proving the 'Initialization Theorem.' The second proof endeavor to investigate the 'Preconditions' of each evolution operation defined over the clinical knowledge model.

5.1.1. Consistency proof using the Initialization Theorem

The Z schema encapsulates different variables and operations to represent the behavior and state of the model. The predicate component in the schema represents a state invariant of the model, i.e., enlisting the domain requirements that must be true in any valid state. In case of contradiction, the model description becomes vacuous, i.e., the desired domain requirements are not fulfilled; therefore, no state exists. To verify that this is not the case, and the developed models are of some use, the proof of the 'Initialization Theorem' ensures that at least one state, i.e., the initial state exists for the model. Supplementary Appendix A provides detailed background and proof of the initialization theorem for the R-CKM model.

5.1.2. R-CKM evolution consistency proof using simplified preconditions and proving the property composition

Investigation of preconditions for an operational schema that evolves the model ensures consistencies of the process (i.e., operation) in terms of its applicability within the boundary of the domain model. More specifically, it represents a set of states for which the outcome of the operations is adequately defined. Supplementary appendices B and C provide detailed steps of the proofs that investigate the precondition of operations, i.e., AddPathRCKM (Schema 6), which evolves the R-CKM model. The proofs show a new set of pre-condition predicates that were not known in advance. The next section provides a detailed evaluation of the newly discovered precondition, which gives birth to an enhanced ReKA method.

5.2. Evaluation: Comparative analysis of ReKA and hybrid knowledge acquisition method

As a consequence of "Proving consistency" mechanism, the main problem is inconsistencies identified in step-3 (selection and refinement of the selected PM decision path) of the validation process in the hybrid knowledge acquisition method. The inconsistencies are eliminated by introducing nine additional criteria (see Table 1) that are placed after refinements. As an outcome of the formal verification, the enhanced ReKA method accommodates the newly discovered criteria. The ReKA criteria cover the broad categories of inconsistencies defined below. Each criterion contributes to one or more categories of inconsistencies.

1. *Category-1 (Violating formalism of the R-CKM):* These inconsistencies occur because of bypassing the construction norms of the R-CKM. This category of inconsistencies makes the final model invalid in terms of affecting outcomes of other decision paths. Criteria 1, 2, and 7 ensures avoiding inconsistencies related to R-CKM formalism.
2. *Category-2 (Violating conformance to guidelines (CKM)):* This category represents refinements, which produce inconsistencies in a decision path that does not conform to clinical guidelines without associating any additional significant evidences. These criteria were in place during the initial steps of the acquisition process (in the hybrid knowledge acquisition method); however, it was not available to ensure the conformance after refinements. Criteria 5 and 9 explicitly discuss that each refined path must conform to CKM.
3. *Category-3 (Compromising quality of R-CKM):* These inconsistencies are related to the quality of R-CKM, which are mainly instigating from the refinements to the existing PM decision path without re-evaluation on patient data. Criteria 6 defines performance (such as accuracy) associated with each refined decision path after evaluating against existing patient data.

4. *Category-4 (Introducing out-bounded refinements):* This category discusses the inconsistencies in decision path that comes intentionally or unintentionally by introducing conditions or treatment plans which do not exist in the hospital information system or out of the scope of the healthcare provider. Criteria 3 and 4 dictates that a domain expert must include only appropriate conditions and treatments that exist within the boundary of the capacity of the healthcare provider.

Table 1. Evolution 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 the same for any decision path when R-CKM already has some decision paths. • Root of the R-CKM will be the first condition for the decision path when R-CKM has 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 the PM decision path for treatment must be conformed. • Example: Treatment refinements in the root of the decision path are not conformed.
3.	$\mathrm{ran}(\mathrm{dom}\, rckmPath!) \subset$ $\mathrm{ran}\, decisionPathConditionRCKM$	• Conditions in the refined decision path must come from the defined condition set of the R-CKM. • Example: Conditions outside the 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 be within the scope of the defined treatments. • Example: Conclusions for the 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 the condition of another decision path in R-CKM. • Example: The refined path may be an extension of an existing decision path.
6.	$0 \leq decPathRCKMAccuracy(rckmPath!) \leq 100$	• Refined decision path accuracy must be within the range of 0 to 100. • Example: The refined decision path should be tested for the set of patient data.
7.	$head\,(\mathrm{dom}\, rckmPath!) \notin \mathrm{ran}\, ConclusionRCKM \cap$ $\mathrm{ran}\, decisionPathConditionRCKM$	• The first condition in the refined decision path must not be a treatment plan. • Example: A 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.

5. *Category-5 (Introducing inconsistencies due to complexities):* This category is related to Category-1. However, it further covers the inconsistencies that occur due to lack of availability of descriptions for the

construction of R-CKM. Criteria 8 is the detailed formal description of how to refine the path in order to avoid any inconsistencies. Criteria 1, 2, and 7 of Category-1 also comes under this category.

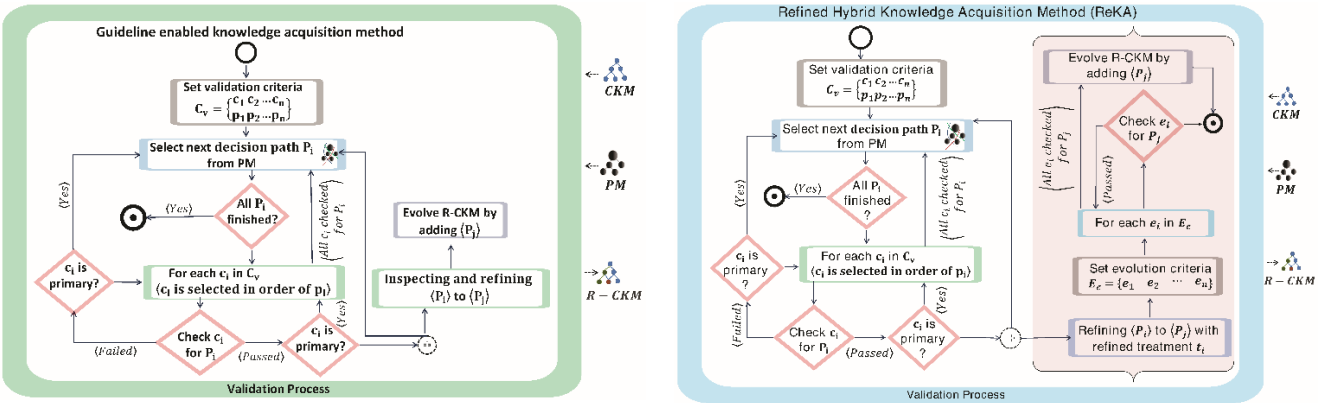
Table 2: ReKA method Vs. earlier hybrid knowledge acquisition method (Knowledge validation perspective: All refinements are valid in earlier method)

Invalid Refinements to decision path #2	*ReKA Validation*
Given decision path: *TreatmentIntent: palliative → Treatment Plan: RT (Radiotherapy)*	
Scenario 1: Modified decision path after refinements: *TreatmentIntent: palliative → Treatment Plan(RT): done → Surgery*	**Criteria violation:** *Category: 2* *Specific Criteria: 9*
A rationale for violation: The guidelines (CKM) recommends follow-up after RT for palliative patients.	
Scenario 2: Modified decision path after refinements: *Treatment Plan(RT): done → TreatmentIntent: palliative → Follow-up*	**Criteria violation:** *Category: 1, 5* *Specific Criteria: 1, 2, 7*
A rationale for violation: The treatment path after modification represents valid guideline-based treatment. However, it violates the formalism of the decision tree because modification in root node affects other decision paths.	
Scenario 3: Modified decision path after refinements: 1. *TreatmentIntent: palliative → Check for risk. ECS: Yes → Treatment Plan: RT or Surgery* 2. *TreatmentIntent: palliative → Check for risk. PNI: Yes → Treatment Plan: RT*	**Criteria violation:** *Category: 4* *Specific Criteria: 3* *Category: 3* *Specific Criteria: 6*
A rationale for violation: Although the refinement 1 and refinement 2 conform to the guidelines (CKM), they are invalid due to the following reasons: *Category-4:* The hospital healthcare information system (HIS) at current stage maintain data for essential adverse histopathologic risk factors such as extracapsular spread (ECS). It keeps all other related factors such as perineural invasion (PNI) and lymphovascular invasion (LVI) in a broad category of others-histopathologic risks. Hence introducing PNI as a discrete data item raises an issue of the direct integration of the decision path to HIS workflows. Furthermore, the HIS records the risk of ECS to the patient with the radical status of treatment intent. So based on the above discussion; refinement#1 is not valid knowledge within the scope of local practices, and refinement#2 is also invalid as it introduces unknown data items. *Category-3:* Violation of criteria 3 in category-4 leads to the undefined quality of final RCKM. The decision path acquired from refinement#1 and refinement#2 cannot evaluate against patient data because of the unavailability of the missing data for ECS and PNI risk factor respectively.	
Scenario 4: Modified decision path after refinements: *TreatmentIntent: palliative and Treatment Plan(RT): done → Salvage Surgery: done → Follow-up*	**Criteria violation:** *Category 4:* *Specific Criteria: 4* *Category: 3* *Specific Criteria: 6* *Category: 2* *Specific Criteria: 9 and 5*
A rationale for violation: The Salvage Surgery refers to surgical treatment which uses after the failure of initial treatment. It is recommended explicitly having residual disease exist after RT for neck metastasis. The refinement for decision path with extended Salvage Surgery is invalid due to the following reasons: *Category-4 and Category-3:* It creates an issue of direct integration to HIS workflows because the existing treatment plan cannot differentiate between standard surgery and salvage surgery. This specific violation in turns raises category-3 violation which has the same interpretation mentioned in Scenario 3. *Category-2:* The refinements do not conform to the given guideline (CKM) even after interpretation of salvage surgery to standard surgery. The main reason is the salvage surgery is applicable for patients who are categorized as radical by TreatmentIntent.	

The ReKA method can elaborate on the ambiguous steps in the validation process related to refinements. To better understand the impact of the ReKA method, Table 2 discusses four refinement scenarios to the decision path by the domain expert (see Path-2 in Figure 12). Each of these scenarios introduces inconsistencies

that relate to one or more categories. It is important to note that these refinements are valid according to the hybrid knowledge acquisition method.

Figure 13 depicts the enhancements made in the knowledge acquisition method in the validation process. Figure. 13a shows the initially proposed hybrid knowledge acquisition method with detailed steps of the validation process. The ReKA method, with the suggested improvements due to formal verification, is depicted in Figure. 13b. The ReKA method enhances the hybrid acquisition method by introducing the following specific sub-steps:



(a) The hybrid acquisition method (*Before Formal Verification*)

(b) The ReKA method (*After Formal Verification:* Highlighted part shows enhancements)

Figure 13: Comparison of ReKA with earlier hybrid knowledge acquisition method (Process enhancement perspective)

- *Evolution criteria setting:* Any refinement in the decision path suggested by domain expert must be evaluated against a set of evolution criteria (specified in Table 1) to avoid inconsistencies as mentioned earlier in the R-CKM.
- *Criteria checking:* All evolution criteria are compulsory, and a refined decision path in R-CKM must fulfill each criterion. Any refinement to decision path which is not fulfilling any of the nine criteria lists must not be considered, and the domain expert is prompted for the violation and indicated with a non-valid evolution of the R-CKM model.
- *Evolution of R-CKM:* After passing the evolution criteria, the refined decision path becomes part of the R-CKM, and the process terminates faithfully.

5.3. Comparison with the existing approach

One way of combining the traditional data-driven approach and guideline-based approach is to use PM as a source and transforming it into the final knowledge model R-CKM after rigorous validation process which conforms the transformation from CKM - the guidelines. However, the combination of these approaches can be done in another manner - considering CKM as a source and adding the decision paths from PM, which are missing in the CKM. In this section, we will discuss one of the existing most relevant approaches [22], which lies in the second category and draw a comparative analysis with our approach. In order to know the detailed description, Figure 14 shows the high-level steps in both knowledge acquisition approaches. These are given the same CKM and PM as an input. The resulting outcome - we called the R-CKM model is different with both approaches.

The main limitations of the existing approach are highlighted in Figure 14, and a detailed discussion is provided in Table 3.

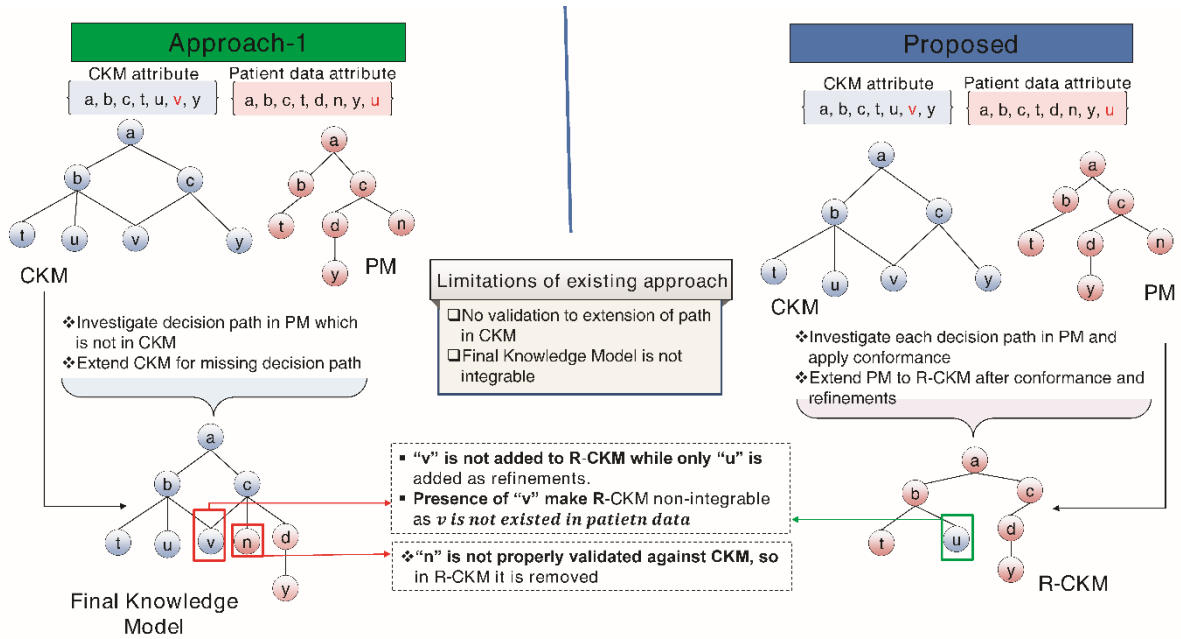


Figure 14: Demonstration of existing approach vs. proposed approach [22]

Table 3: Detailed description of guideline-enabled data-driven formally verified approach vs. 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.		

In a nutshell, the existing approach tends to use the PM as a key source to refine the decision paths in the CKM while compromising the quality of the model (missing rigorous validation) and integration to existing healthcare workflows. We also applied the existing approach to the SKMCH dataset of 1229 oral cavity cancer patients and used the CKM as a reference guideline model (derived for oral cavity NCCN guidelines). In the final knowledge model, we identified that 26% of the decision paths were violating the quality criteria of lower accuracy (in our case, it should be greater than 50%), 30% of the decision paths were not conformed to guidelines, and 9% decision paths were violating multiple criteria, i.e., lower accuracy and non-conformance. Overall, 47.8% of decision paths lacked to pass the validation criteria. Figure 15 shows the details of the decision tree C5.0 algorithm (which is referred by Toussi et al.) with highlighted decision paths lacking one or more validation criteria.

As described in Table 3, the final model obtained from Toussi et al. approach is not necessarily integrable to evaluate its performance against patient data available at a local organization. However, as shown in Figure 15, the source model for Toussi et al. approach is C5.0, which has an overall accuracy of 69.7% on the SKMCH dataset. Even considering the Toussi et al. approach produces the final knowledge integrable to existing healthcare workflow, still there exists a chance that overall model accuracy will fall from 69.7% because of its generalization. In the case of proposed work, we have a rigorous selection process for choosing the appropriate machine learning algorithm and as indicated the CHAID decision tree is a candidate algorithm with an accuracy of 71% on a data set of 1229 oral cavity patients (see [17] for details of part of knowledge acquisition related to this part). Moreover, the final knowledge model – R-CKM performance is improved to 72.57%, as shown in

Figure 16. To conclude, the proposed approach also gives higher performance over Toussi et al. on the local SKMCH dataset.

It is important to note that in Toussi et al. work, the selection of algorithm C5.0 is based on the criteria to produce more decision paths to assist in the completion of CPGs. Furthermore, C5.0 is suitable for the dataset with a large number of features and having a higher chance of missing data. So, for fair comparison and evaluation of ReKA against Toussi et al. work, the C5.0 is applied upon the SKMCH dataset.

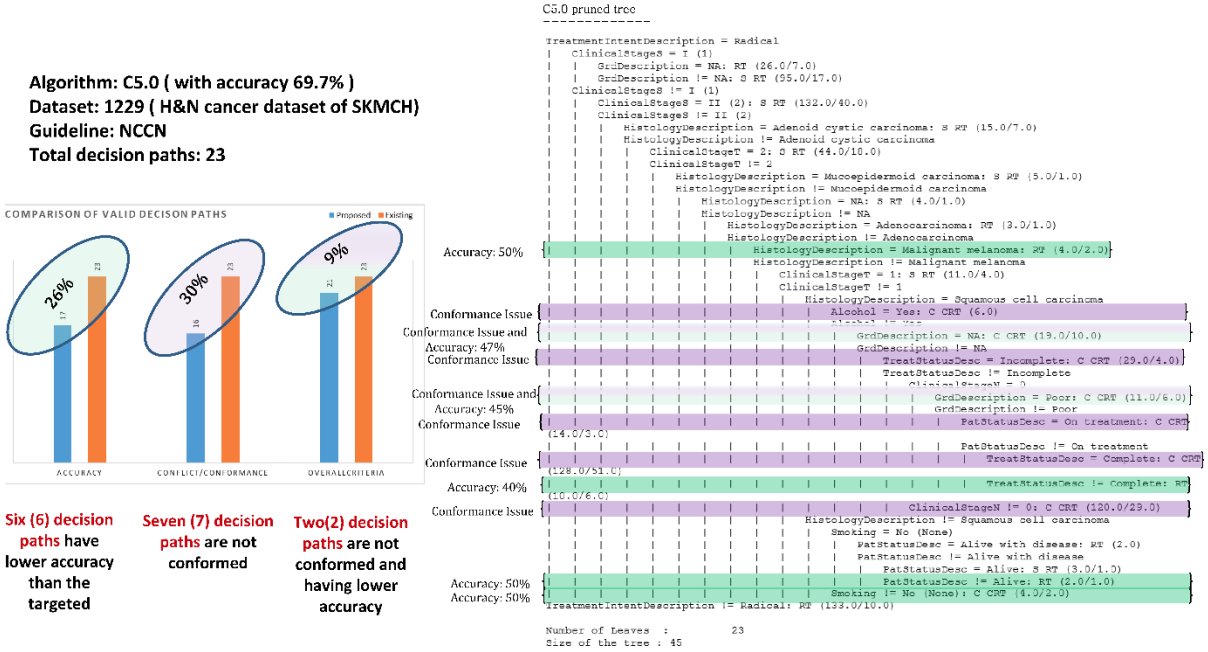


Figure 15: Comparison of the proposed approach with the existing approach using SKMCH oral cavity cancer data [22]

6. Discussion

6.1. Is verification always required for clinical knowledge modeling?

Validation and verifications are the key pillars of trust on the domain knowledge. However, the use of applying both processes mainly contributes to the criticality and complexity of the clinical knowledge modeling method and the final knowledge models. In other words, for less critical and less complex clinical knowledge acquisition methods and models, validation alone mainly plays a role in establishing the required level of trust in the knowledge. For example, in TNM staging for cancer [34], predefined rules or algorithms and deterministic mappings are used for final knowledge modeling. The TNM staging knowledge model does not need exhaustive validation. It requires a limited set of validation cases to test the validity of complete knowledge.

In contrast, validation and verification should be in place for the knowledge acquisition method, which involves different sources of knowledge with diverse structures and semantics. In this research work, the ReKA method involves diverse knowledge sources, i.e., CPGs, decision trees (from patient data), and domain expert heuristics (in refinements), where each of them has different nature of structure and semantics. Therefore, the only validation could not guarantee that the knowledge model is always valid. The formal verification in the ReKA method has introduced the necessarily missing steps in our previously hybrid knowledge acquisition method, which originally only relied on the validation process. The outcomes of the ReKA method have been evaluated through a set of the real clinical scenarios provided in Table 2.

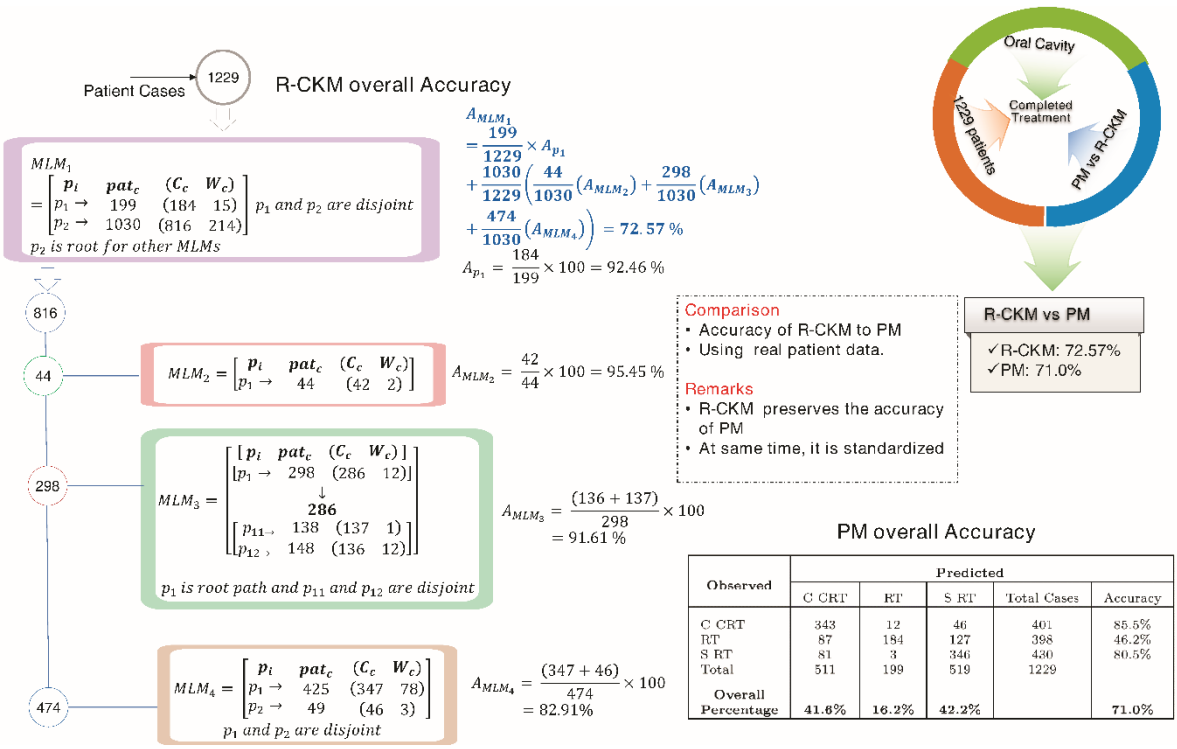


Figure 16: R-CKM results using oral cavity cancer data [R-CKM is implemented as set a of HL7 MLMs and the evaluation results drawn are based on the structure of MLMs]

6.2. What is the overhead of the formal verification process and to what extent the formalism is required?

Formal verification introduces significant overhead in terms of time and selection of expert resource who has sufficient skills to model the domain knowledge mathematically. It is important to note that the applicability of the formally verified model is not reduced in terms of efficiency rather delay is expected in the delivery of the final knowledge model. Therefore, to tune the trade-off among timelines, cost-effective delivery, and producing high-quality knowledge model, the maximum extent of formalism is identified during clinical knowledge modeling.

In ReKA method, the formal verification is involved in the first two phases of modeling of CPGs (CKM), decision trees (PM) and final model (R-CKM). In the third phase, R-CKM is converted into standard knowledge representation of HL7 MLM (medical logic modules) for executation and sharability. There was a choice of applying the formal verification (Z refinements) in the third phase of the transformation of R-CKM to MLMs. However, we already demonstrated in [17] that the conversion is straight forward, and the MLMs are easily validated against the real patient cases, as shown in Figure 16.

6.3. Whether validation or semi or less formalism sufficient for consistency of complex knowledge acquisition models or methods?

As discussed earlier, only validation is not enough for complex knowledge acquisition methods or models. Similarly, semi or less formal analysis of the knowledge acquisition method does not always guarantee consistent and valid knowledge. As an example, Grando et al. have demonstrated that using the formal analysis for the expressiveness of CIG languages yields satisfiability of some patterns, which was ignored or not detected by the less formal analysis method [35].

In the hybrid knowledge acquisition method [17], initially, we relied on a rigorous validation process, the partial formalism of the decision tree, and freedom of domain expert to modify the existing knowledge model with additional constraints. While using the ReKA method, we have realized that relying upon only the validation process and partial formalism support of the decision tree, the final knowledge model is not always valid. The ReKA method has highlighted a set of inconsistencies, which includes violation of knowledge representation formalism (R-CKM), conformance to CPGs, compromising the quality of the model, outbound

refinements in the model, and inconsistencies due to the complexity of the knowledge. Each category of inconsistency is demonstrated with real clinical knowledge modeling scenarios, as shown in Table 2.

6.4. Whether formally verified clinical knowledge models or methods always guarantee consistency concerning the essential context of the domain?

According to Boehm definition, validation is all about, “are we building the right product?” and the verification is all about, “are we building the product right?” [36]. In terms of clinical knowledge modeling, we can interpret the definition in simple words as: “do we create right knowledge model for the clinical domain under consideration?” and “do the method of acquisition is right to reflect all essential context of the clinical domain in the creation of knowledge model?”. At this stage, knowing all the essential context of the clinical domain is important. The verification process will make sure to find the inconsistencies in the acquisition process according to the context provided during the validation process. In conclusion, the formal method always guarantees the consistency of the model based on the given domain context. However, if the essential context of the domain is missing or wrongly perceived in the design, then formal verification is also not the ultimate solution to discover the missing or detect the wrongly perceived knowledge.

6.5. Do intrinsic properties of the formal method influence overall consistencies of the model?

The underlying expressive power of the formal method has a relationship with the consistency of models. The more expressive a language is, the better ease it provides in representing the models; however, it becomes hard to prove the overall consistency of the representative model. This challenge is commonly known as the non-decidable expression logic, used as a part of a formal specification for the given model. Z formal language is one such expressive language that is based on predicate logic. Predicate logic is intrinsically non-decidable, and hence it turns out that Z notation inherits this property as well.

On the contrary, decidability is considered as one of the key features of Z notation in its standard ISO specification [37]. Although the reasons are not explicitly in the specifications; however, one reason is that the Z is built on the underlying Zermilu Frankline Set theory[38]. This theory tends to restrict the use of the universal set in the core set theory. More precisely, Zermilu Frankline introduced a set of axioms known as "ZFC" – C stands for axioms of choice, that replace the naïve set theory with more natural axiomatic set theory. It reduces ambiguities coming from intuitions, such as reported in Russell's paradox. In this case, even Z is using quantifiers' expression from predicate logic; however, the variables in the quantifiers are bounded to a finite set, which can be proved easily. Also, the baseline ZFC bounds Z specification to be provable. Although Z is decidable, however, it is highly recommended that one needs to restrict the creation of specification with minimal use of quantifiers to keep the proofs simple.

6.6. What key characteristics make Z superior from the ontological approach for clinical knowledge modeling?

Z and ontologies both tend to represent knowledge formally. Both approaches have their pros and cons that have to be assessed before selecting the more appropriate one. This section highlights the aspects of Z that resulted in our adoption of this approach compared to ontologies.

- An intricate knowledge acquisition process is used to acquire the clinical knowledge model. This process ensures to produce a non-ambiguous (valid) knowledge model (or artifacts) by involving consistent (formally verified) acquisition tasks. Z formalism is rich in the toolset to fulfill the requirements of an intricate clinical knowledge acquisition process. In contrast, ontological approaches focus on the knowledge model (or artifact) while keep relying on external processes to define the additional acquisition tasks.
- To achieve the goal of clinical objective (of the models), passing through the inspection of raw clinical knowledge resources, modeling the intermediate clinical knowledge models, and producing the final executable knowledge model, the Z has the potential of identifying inconsistencies at every knowledge transformation phase. In contrast, ontological approaches are only concentrating on the final executable knowledge model.
- Z is much more expressive than ontological languages. It is due to the support of extensive mathematical toolsets and relationships. Ontological languages come with a spectrum of lower to higher expressive power, but the later becomes non-decidable.

- Z is human-readable but not fully machine-interpretable; however, Z refinements lead the specification very close to the real implementation. The ZML is an XML markup for Z artifacts [39], which is used to exchange the Z specification between different toolsets. For example, ZML is consumed by the type checker to ensure the syntax, passes it to the tools to convert predicates in the schema into a disjunctive normal form, and finally, the test generator uses the outcome (in XML) to generate the test cases for each disjunction.

6.7. How easy is it to use Z formalism, and what skills are required by the team for formalism in clinical knowledge modeling?

Formal methods have apparent successes in substantially improving the modeling precision, requirements clarity, and verification confidence that enables cost-effective and error-free models. Despite these facts, formal methods are still not widely adopted. The crucial obstacles are; learning curve, efforts for integration with existing processes, and difficulties in maintenance [40]. Although the learning curve for formal methods is higher in comparison to procedural languages, in the real development process, very few resources are required in the team to handle artifacts related to formal methods. The other team members are only required to observe the outcomes of the formal process and to align the milestones accordingly. One example is the WSDL 2.0 specification.WSDL 2.0 specification [41] is accompanied by Z specification as informative material to support the normative specification. A single team member was responsible for handling the formal specifications and sharing the outcome (inconsistencies) in the WSDL 2.0 core specifications with other team members.

In the ReKA, three team members have worked on the formal specification. One team member was an expert in Z notation, the second was having the know-how of Z notation, and the third was an expert in ontologies. For internal correctness, the developed specifications were inspected by the formal verification team, and the outcomes were shared with other team members, including developers and domain experts. So in this work, the Z specification was developed by a single team member, verified by two team members, and the outcomes were shared with domain experts and developers without indulging them into the technical details.

6.8. What is the possibility of the evolution of the final knowledge model to comply with locally emerging evidence from local practices?

The ReKA combines the knowledge from CPGs and patient data (retrospective), and it assumes the change in model is made if CPGs are revised or sufficient new data is available. In both cases, the clinical knowledge model is not supposed to change frequently. However, the ReKA does not impose any restrictions on the evolution of growing knowledge. The final executable knowledge model in ReKA provides the opportunity to cope with the evolution of knowledge and provides support for emerging local evidence in the clinical model. The final executable knowledge model is a set of MLMs (HL7 based standard procedural rules), which has been thoroughly examined in our patent [42] with the philosophy of case-based reasoning (CBR) methodology. CBR based approach provides validation for existing MLMs, and it also revises the knowledge if the domain expert modifies the final decision based on newly added patient cases. This extension brings the CBR mechanism to evolve the final knowledge model to comply with emerging local evidence. The extension is not fully conformed to ‘belief variations’ based on knowledge acquisition. However, the case base and MLM are revised based on the discovery of local evidence; therefore, it has a very close analogy with ‘belief variations’ of the agents in terms of conformance with new knowledge discovery.

6.9. How is ReKA extendable to other domains?

ReKA is easily extendable to other clinical domains, which rely on established CPGs and medical records. The core components of ReKA are underlying knowledge resources CPGs and patient data. At the same time, it involves humans in the loop to produce a refined knowledge model. So, the ReKA components are used without any change if the underlying semantics of knowledge models remained the same.

In one of our work, the ReKA is implemented in the cardiovascular domain. The final knowledge model outperformed for the diagnosis of heart failure with an overall concordance rate of 98.3% [43]. Moreover, by involving the CPGs and domain expert heuristics, a subset processes of ReKA is also demonstrated and implemented successfully in thyroid nodule treatment[44]. In both domains, well established CPGs and patient

data (or only domain expert heuristics) are used without changing underlying semantics of the models. Therefore, formal verification is not required.

Recently we are extending the ReKA to evaluate subarachnoid hemorrhage stroke CDSS, which is mainly caused by the rupture of intracranial aneurysm (also known as brain aneurysm). We already developed CDSS, which uses a knowledge graph as an underlying representation model; however, it relies only on patient data as knowledge resource [45]. The challenge in developing a brain aneurysm prediction model is the unavailability of well established (standard) CPGs and non-crisp rupture risk prediction. Instead of using CPGs, the study relies on online published resources (non-standard), and for non-crispy risk prediction, we are combining various rule-based and probabilistic models. In such knowledge artifacts formulations, the core processes of ReKA are reused; however, the underlying semantics of the final model is changed. Therefore, we are expecting to apply formal verification to ensure that the acquisition process is consistent with the newly emerged semantic of the rule-based probabilistic model.

6.10. "Why-Not" always can trust the clinical knowledge model created through a non-verified knowledge acquisition process?

Trust in knowledge is one of the main enablers of CDSS use [1]. It is considered a multidimensional area for evaluation, particularly in an ecosystem such as CDSS at its early development phase. So trust in CDSS gives the potential for patients to avoid harm from healthcare that is not evidence-based. According to the Trust Framework Working Group (TFWG) – a chartered workgroup of the Patient-Centered Clinical Decision Support Learning Network, nine trust attributes are associated with the overall ecosystem of CDSS development. It includes competency, compliance, consistency, discoverability & accessibility, evidence based, feedback & updating, organizational capacity, patient-centeredness, and transparency [46]. These attributes target various aspects of trustworthiness associated with knowledge artifacts, actors who develop or use the knowledge artifacts, CDSS repositories, and its implementations. The following discussion highlights the critical trust attributes (compliance, competency, consistency, and transparency), which may not be adequately addressed if a non-verified knowledge acquisition process is adapted for CDSS development.

- Compliance to standard processes

  CPGs are considered one of the trustworthy sources of clinical evidence. The trust in CPGs comes with a standard development process. The most common component of standard CPGs comprises; rigorous and transparent processes, properly managed conflict of interests, adequate group composition, strict intersectional systematic reviews, highly manageable scope, and reliable external reviewing and maintenance [47].

  Unlike the CPGs, no standard process exists to develop a clinical knowledge model from diverse resources with multiple acquisition methods. In the case of ReKA, the source of clinical evidence is not only the CPGs but also patient data to accommodate the local evidence. Currently, it is a norm that machine learning (AI-based) acquisition methods are followed for knowledge acquisition from patient data. From the vast majority of stakeholders from healthcare – more specifically, the domain experts have no confidence in adapting the AI-based clinical knowledge models. Their critical demands from the AI community are in providing and establishing a firm foundation of trust in their acquisition methods' validity, consistency, objectivity, and reliability [48]. The ReKA is developed indigenously with support of making all knowledge artifacts (AI-models – PM, CPG-based models – CKM, and combined-models – R-CKM) explicit and traceable through formal specifications. The method is equipped with a firm theorem proving mechanism and driven by clear clinical objectives to establish a trust in the consistency, transparency, and reliability of the overall acquisition process. In the absence of formal verification, most of the upper mentioned concerns yield a lack of trust in using the knowledge models. The fact about these concerns has been revealed in the refinement process of knowledge acquisition through formal verification. The knowledge produced with hybrid knowledge acquisition (before ReKA) should not be trusted because of its non-alignment and conformance to a standard process. This led us to find an answer to "Why-Not" so that stakeholders shall trust the knowledge produced with ReKA. The ReKA formal verification ensures consistency and identifies all hotspots of concerns leading to untrustworthiness. Table 2 provides a couple of scenarios that cover-up the

potential of formal verification and answering the “why not” that ReKA is capable of producing a trustable knowledge model even it does not fully conform to standard processes.

- Competence and experiences of domain experts

Expertise in the domain and capability of the domain expert to create a clinical knowledge model is considered one of the important factors of trust in knowledge. According to the study [1], the reason for the slower adoption of CDSS by general practitioners was the lack of confidence in the acquisition methods and contents of the knowledge, i.e., the expertise of people who created the knowledge base in terms of contents and the acquisition process toolsets. The workgroup TFWG also recognizes the influence of expertise of domain knowledge experts in the creation of knowledge, and trust attribute 'Competency' is associated with knowledge artifacts ensuring trust in the knowledge [46]. The competency dictates the role of the actor (a domain expert) to be competent in the CDSS ecosystem. The competency factors can be recognized by investigating past performance, professional qualifications, or certifications of the domain expert. More precisely, the domain expert should be competent in the knowledge management life cycle, competently interpret, encode, and execute the knowledge [46]. Unfortunately, the era is still far away from this idealistic situation where every domain expert will be competent enough to set up the same level of trust for the creation of knowledge. The most common reasons include; insufficient informatics training programs for the domain experts to translate the evidence into CDSS knowledge artifacts, low literacy of domain experts in the domain of underlying CDSS technology, and hard to manage the timing for the creation of knowledge – they are involved most of the time with patients at clinics [49]. Besides these facts, there are ways to create astute knowledge artifacts that may establish a sufficient level of trust. Firstly, the provision of a knowledge development framework which provides room to accommodate and facilitate the collaboration of stakeholders with a diverse background and different competencies. Secondly, the provision of the knowledge acquisition process that facilitates all stakeholders and evades any loophole in the acquisition that leads to the creation of inconsistent knowledge artifact – possibly caused by the stakeholders with low-level competency or by other well-known mistakes.

The ReKA is used under the umbrella of a Smart CDSS development framework that provides a collaborative environment for the creation of clinical knowledge models. Furthermore, the formal verification processes in ReKA always ensure the creation of consistent knowledge artifacts. “Why-not”, always a valid knowledge model is expected in the presence of a non-verified hybrid knowledge acquisition process (before ReKA) – because it comes with free refinement process that leads towards easily exploitable loopholes in the creation of knowledge models. The low competent domain expert may cause inconsistent knowledge due to less knowledge in a domain or by mistake. Considering the scenario 1 in Table 2; it reflects the violation of knowledge artifacts by non-conformance to CPGs. Moreover, scenario 2 in Table 2 reflects the violation of building proper knowledge artifacts. The first violation possibly caused by domain expert having less experience with evidence based knowledge, and the second violation most probably caused by a domain expert with less expertise of the toolset and knowledge artifacts. The ReKA facilitates in these cases to restrict the stakeholders from creating any knowledge artifact, which reduces trust in overall knowledge models.

- Transparent and consistent knowledge acquisition process and knowledge models

The promises of AI is undeniable in many domains. There is a race in the research community for setting down a record high accuracy benchmark through the new acquisition model. However, the domain of healthcare demands from AI-based knowledge models the more equitable humanistic care, not just more accurate, scientific care [50]. The black-box approaches are unacceptable as CDSS requires transparency so that the user can trace the basis of recommendations that are offered [51]. The knowledge model (acquired through black-box) that does not explain the source or nature of recommendations erodes the trust [52]. The knowledge acquisition processes are not traceable to be consistent over a set of different patient cases.

Furthermore, non-transparent knowledge recommendations may weaken the trust relationship between the domain expert and patients. The most transparent and consistent knowledge acquisition process and

knowledge model may lead to a high level of trust in the recommendations. The ReKA produces knowledge models that are conformed to standard CPGs (evidence supportable), and at the same time, only white-box approaches (such as decision trees and production rules) are used for acquiring knowledge from patient data. Moreover, to keep the models and acquisition process more explicit (traceable, transparent, and consistent), the formal verification provided detailed specifications. In conclusion, ReKA uses formal verification to justify the "Why-Not" answer, that non-verified properties of knowledge are not transparent, possibly not consistent, and not traceable, which ultimately ends up with untrustworthiness. Scenario 4 in Table 2 explains how a single refinement in knowledge artifacts triggers a set of inconsistencies that yields an invalid knowledge model due to non-transparent and non-traceable hybrid acquisition process (before ReKA).

7. Conclusion

This paper has introduced an enhanced ReKA method which performs formal verification using Z notation. Z notation proves the consistency of the acquisition process and hence, improves the hybrid knowledge acquisition method. The ReKA method is established based on the three steps formal verification process to represent the knowledge models formally. Also, it involves the associated validation process of hybrid knowledge acquisition using various artifacts of Z notation. Subsequently, the mechanism of theorem proving in formal verification has identified inconsistencies in the previously established knowledge acquisition by introducing nine additional criteria. These criteria address the broad categories of inconsistencies related to the formalism of knowledge, conformance to CPGs, quality of knowledge, and complexities of knowledge acquisition artifacts. The ReKA method produces a guideline-enabled data-driven knowledge model that supports the high-quality recommendation, global evidence, local practices, and always consistent model compared to existing hybrid knowledge models. It is important to mention that the key advantages of the ReKA method include its generality, which can be easily adapted in other cancer domains. Moreover, to the best of our knowledge, this is the first attempt to use Z notation in the modeling of medical knowledge and to align its core step as contents of method plugin, in the Smart CDSS development framework.

Acknowledgments: The present study was supported by the Korean Ministry of Science and ICT, under the auspices of the Information Technology Research Center (ITRC) support program (IITP-2017-01629). The ITRC is supervised by the Korean Institute for Information and Communications Technology Promotion.

This work was also supported by the National Research Foundation of Korea under the grant (NRF2017R1A2B2012337), in part by Sejong University Faculty Research Fund. Moreover, the research was also partially supported by the National Research Foundation of Korea (NRF-2019R1G1A1011296).

The authors would like to acknowledge the support of the medical and IT staff of Shaukat Khanum Memorial Cancer Hospital and Research Center, Lahore, especially; Dr. Raza Hussain, Dr. Sadaf, Dr. Hassan Iqbal, Mr. Manzar Abass, and Mr. Idrees Khokhar.

Conflicts of Interest: None.

References

[1] R. Shibl, M. Lawley, J. Debuse, Factors influencing decision support system acceptance, Decision Support Systems 54 (2) (2013) 953-961.

[2] M. Peleg, Computer-interpretable clinical guidelines: A methodological review, Journal of biomedical informatics 46 (4) (2013) 744-763.

[3] B. Canovas-Segura, A. Morales, J. M. Juarez, M. Campos, F. Palacios, A lightweight acquisition of expert rules for interoperable clinical decision support systems, Knowledge-Based Systems 167 (2019) 98-113.

[4] V. Gomoi, M. Vida, R. Robu, V. Stoicu-Tivadar, E. Bernad, O. Lupse, From data mining rules to medical logical modules and medical advices., Studies in health technology and informatics 192 (2013) 1094-1094.

[5] S. Perera, C. Henson, K. Thirunarayan, A. Sheth, S. Nair, Semantics driven approach for knowledge acquisition from emrs, IEEE Journal of Biomedical and Health Informatics 18 (2) (2014) 515-524.

[6] 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 41 (6) (2008) 889-903.

[7] R. Serban, A. Ten Teije, Exploiting thesauri knowledge in medical guideline formalization, Methods of information in medicine 48 (5) (2009) 468-474.

[8] P. L. Miller, Domain-constrained generation of clinical condition sets to help test computer-based clinical guidelines, Journal of the American Medical Informatics Association 8 (2) (2001) 131-145.

[9] M. Peleg, L. A. Gutnik, V. Snow, V. L. Patel, Interpreting procedures from descriptive guidelines, Journal of biomedical informatics 39 (2) (2006) 184-195.

[10] G. Duftschmid, S. Miksch, Knowledge-based verification of clinical guidelines by detection of anomalies, Artificial intelligence in medicine 22 (1) (2001) 23-41.

[11] B. Perez, I. Porres, Authoring and verification of clinical guidelines: A model driven approach, Journal of biomedical informatics 43 (4) (2010) 520-536.

[12] S. Liu, Y. Chen, F. Nagoya, J. A. McDermid, Formal specification-based inspection for verification of programs, IEEE Transactions on software engineering 38 (5) (2012) 1100-1122.

[13] S. Uchitel, G. Brunet, M. Chechik, Synthesis of partial behavior models from properties and scenarios, IEEE Transactions on Software Engineering 35 (3) (2009) 384-406.

[14] J. Diaz, D. Arroyo, F. B. Rodriguez, On securing online registration protocols: Formal verification of a new proposal, Knowledge-Based Systems 59 (2014) 149-158.

[15] C. Newcombe, T. Rath, F. Zhang, B. Munteanu, M. Brooker, M. Deardeuff, How amazon web services uses formal methods, Communications of the ACM 58 (4) (2015) 66-73.

[16] R. F. Gamble, P. Stiger, R. Plant, Rule-based systems formalized within a software architectural style, Knowledge-Based Systems 12 (1-2) (1999) 13-26.

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

[18] Rational unified process, (Accessed: December 30, 2018) (2018).URL http://home.iscte-iul.pt/~hro/RUPSmallProjects/

[19] I. Jacobson, G. Booch, J. Rumbaugh, J. Rumbaugh, G. Booch, The unified software development process, Vol. 1, Addison-Wesley Reading, 1999.

[20] ISO/IEC, Is 10746?, information technology ?open distributed processing?reference model: Overview, also published as itu-t recommendation x.901, ISO, Standard (1998).

[21] H. Kilov, P. F. Linington, J. R. Romero, A. Tanaka, A. Vallecillo, The reference model of open distributed processing: foundations, experience and applications, Computer Standards & Interfaces 35 (3) (2013) 247-256.

[22] M. Toussi, J.-B. Lamy, P. Le Toumelin, 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 9 (1) (2009) 28.

[23] T. Ali, M. Hussain, W. A. Khan, M. Afzal, J. Hussain, R. Ali, W. Hassan, A. Jamshed, B. H. Kang, S. Lee, Multi-model-based interactive authoring environment for creating shareable medical knowledge, Computer Methods
and Programs in Biomedicine 150 (2017) 41-72.

[24] IBMSPSS User Guide. IBM SPSS Decision Trees 21, (Accessed: 15 May, 2019). URL http://www.sussex.ac.uk/its/pdfs/SPSS_Decision_Trees_21.pdf

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

[26] J. Woodcock, J. Davies, Using Z: specificationcation, refinement, and proof, Prentice-Hall, Inc., 1996.

[27] M. Luck, P. Noriega, J. A. Rodriguez-Aguilar, C. Sierra, et al., Communicating open systems, Artificial Intelligence 186 (2012) 38-94.

[28] M. d'Inverno, M. Luck, M. Georgeff, D. Kinny, M. Wooldridge, The dmars architecture: A specification of the distributed multi-agent reasoning system, Autonomous Agents and Multi-Agent Systems 9 (1-2) (2004) 5-53.

[29] R. O. Sinnott, K. J. Turner, Specifying odp computational objects in z, in: Proc. of FMOODS, Vol. 96, 1996, pp. 375-390.

[30] A. Bottrighi, L. Giordano, G. Molino, S. Montani, P. Terenziani, M. Torchio, Adopting model checking techniques for clinical guidelines verifcationcation, Artificial intelligence in medicine 48 (1) (2010) 1-19.

[31] 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 36 (3) (2006) 193-209.

[32] E. A. Boiten, J. Derrick, From odp viewpoint consistency to integrated formal methods, Computer Standards & Interfaces 35 (3) (2013) 269-276.

[33] H. Bowman, J. Derrick, P. Linington, M. Steen, Fdts for odp, Computer Standards & Interfaces 17 (5) (1995) 457-479.

[34] J. D. Brierley, K. G. Mary, C. Wittekind, TNM classification of malignant tumours. John Wiley & Sons, 2016.

[35] M. A. Grando, D. Glasspool, J. Fox, A formal approach to the analysis of clinical computer-interpretable guideline modeling languages. Artificial intelligence in medicine. 54(1) (2012) 1-3.

[36] B. Boehm, Software Risk Management. IEEE Computer Society Press. Washington DC 1989.

[37] Information Technology - Z Formal Specification Notation-Syntax, Type System and Semantics, 1st ed., RefNo: ISO/IEC 13568:2002(E), 2002.

[38] G.H. Moore, Zermelo's Axiom of Choice: Its Origins, Development, and Influence, Courier Corporation, 2012.

[39] M. Utting, I. Toyn, J. Sun, A. Martin, J.S. Dong, N. Daley, D. Currie, ZML: XML support for standard Z, Lect. Notes Comput. Sci. 2651 (2003) 437–456. https://doi.org/10.1007/3-540-44880-2_26.

[40] M. Gleirscher, S. Foster, J. Woodcock, New opportunities for integrated formal methods, ACM Comput. Surv. 52 (2019) 1–36. https://doi.org/10.1145/3357231.

[41] R. Chinnici, A. Ryman, J.-J. Moreau, Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language Specification, W3C Recomm. (2007). http://www.w3.org/TR/2006/CR-wsdl20-20060327/wsdl20-z.pdf.

[42] S. Lee, M. Hussain, CLINICAL KNOWLEDGE VALIDATION SYSTEM AND METHOD BASED ON CASE BASE REASONING, WO/2016/093406, 2016. https://patentscope.wipo.int/search/en/detail.jsf?docId=WO2016093406.

[43] D.-J. Choi, J.J. Park, T. Ali, S. Lee, Artificial intelligence for the diagnosis of heart failure, Npj Digit. Med. 3 (2020). https://doi.org/10.1038/s41746-020-0261-3.

[44] H.W. Yu, M. Hussain, M. Afzal, T. Ali, J.Y. Choi, H.S. Han, S. Lee, Use of mind maps and iterative decision trees to develop a guideline-based clinical decision support system for routine surgical practice: Case study in thyroid nodules, J. Am. Med. Informatics Assoc. 26 (2019) 524–536. https://doi.org/10.1093/jamia/ocz001.

[45] K.M. Malik, M. Krishnamurthy, M. Alobaidi, M. Hussain, F. Alam, G. Malik, Automated domain-specific healthcare knowledge graph curation framework: Subarachnoid hemorrhage as phenotype, Expert Syst. Appl. 145 (2020) 113120. https://doi.org/10.1016/j.eswa.2019.113120.

[46] B. Middleton, J. Platt, J.E. Richardson, B.H. Blumenfeld, Recommendations for Building and Maintaining Trust in Clinical Decision Support Knowledge Artifacts, Res. Triangle Park. NC Patient-Centered Clin. Decis. Support Learn. Netw. (2018) 1–21. https://pccds-ln.org/sites/default/files/2018-09/TFWG White Paper_final.pdf.

[47] R. Graham, M. Mancher, D.M. Wolman, S. Greenfield, E. Steinberg, Clinical Practice Guidelines We Can Trust, THE NATIONAL ACADEMIES PRESS Washington, D.C., 2011. https://doi.org/10.17226/13058.

[48] B. Jennifer, Success of AI in Healthcare Relies on User Trust in Data, Algorithms, Heal. IT Anal. Xtelligent Healthc. Media. (2018). https://healthitanalytics.com/news/success-of-ai-in-healthcare-relies-on-user-trust-in-data-algorithms (accessed July 6, 2020).

[49] B. Middleton, D.F. Sittig, A. Wright, Clinical Decision Support: a 25 Year Retrospective and a 25 Year Vision, Yearb. Med. Inform. 25 (2016) S103–S116. https://doi.org/10.15265/IYS-2016-s034.

[50] S.T. Israni, A. Verghese, Humanizing Artificial Intelligence, JAMA. 321 (2019) 29–30. https://doi.org/10.1001/jama.2018.19398.

[51] E.H. Shortliffe, M.J. Sepúlveda, Clinical Decision Support in the Era of Artificial Intelligence, JAMA. 320 (2018) 2199–2200. https://doi.org/10.1001/jama.2018.17163.

[52] S. Nundy, T. Montgomery, R.M. Wachter, Promoting trust between patients and physicians in the era of artificial intelligence, JAMA. 322 (2019) 497–498. https://doi.org/10.1001/jama.2018.20563.

Supplementary Appendices

Supplementary Appendix A. Initialization Theorem and proof for R-CKM domain consistency

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

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

For the R-CKM model represented in the schema *RefinedClinicalKnowledgeModel* (Schema 3), the initial state is defined using the state schema *InitRCKM* (Schema 8).

Schema 8 R-CKM Initial state

InitRCKM

$$RefinedClinicalKnowledgeModel' \quad (1)$$

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

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. It is almost impossible to prove the initial state of the modeling specifications, which include contradictions. Hence, it can conclude 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* helps to replace 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 essential background related to replacing the existential quantifier.

Following the definition of the *one-point rule*, and other fundamental laws and definitions, the proof of *initialization theorem* is given in Proof 1. The proof is straightforward, and each step is explained with instructive definitions.

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

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

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

Supplementary Appendix B. Calculating pre-conditions for R-CKM evolution

The precondition of operation is another schema, obtained from a given operation, that hides components related to the state after the operation and provides an output that results from 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 6).

Definition 3: *For the operational schema "operation," the state of the system modeled by "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.*

$$\mathrm{pre}\ operation = \exists\, state' \bullet operation$$

Definition 3: Precondition of an operation

Calculation of pre-condition requires simplification of the predicate part of the theorem (Theorem 2), which involves the expansion of all schemas. After the expansion of all possible schemas, the one-point rule plays a pivotal role in simplifying and proving the primed statements in the schema. Due to space limits, the proof is provided as supplementary appendices. The Supplementary Appendix A explains the proof 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.

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

$$\mathrm{pre}\ AddPathRCKM = \exists\, RefinedClinicalKnowledgeModel' \bullet AddPathRCKM$$

$$\begin{array}{|l}
\mathrm{pre}\ AddPathRCKM \\
\hline
RefinedClinicalKnowledgeModel \\
dp_{pm}? : decisionPath \\
qualifiedAcc? : \mathbb{Z} \\
refinements? : \mathbb{F}\, RefinedTreatmentPlan \\
\hline
\exists\, RefinedClinicalKnowledgeModel';\ rckmPath! : decisionPathRCKM \\
\quad \bullet AddPathRCKM \\
\hline
\end{array}$$

Although the simplification process seems quite complicated in terms of resolving all of the primed statements, however using set theory fundamental laws and the *one-point rule*, it becomes straightforward. The primed predicates in Proof 2 are underlined (numbered 1-13). The prime predicates require simplifications to conclude the proof. To save space, the Supplementary Appendix C presents the simplification proofs.

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

$$\mathrm{pre}\ AddPathRCKM \Leftrightarrow \tag{2.01}$$

$$\exists\, RefinedClinicalKnowledgeModel';\ rckmPath! : decisionPathRCKM \bullet \quad [def\_\mathrm{pre}\ AddPathRCKM] \tag{2.02}$$
$$AddPathRCKM$$

$$\Leftrightarrow$$

$$\exists\, RefinedClinicalKnowledgeModel';\ rckmPath! : decisionPathRCKM \bullet \quad [def\_AddPathRCKM] \tag{2.03}$$
$$RCKM \neq \emptyset \Rightarrow head\,(\mathrm{dom}\, dp_{pm}?) = rootRCKM \wedge \tag{2.04}$$
$$\forall\, pos : \mathbb{N} \mid pos \in \mathrm{dom}\, refinements? \bullet pos > 1 \wedge$$
$$pos < (\#(\mathrm{dom}\, dp_{pm}?) + \#(\mathrm{ran}\, dp_{pm}?)) \wedge \tag{2.05}$$
$$\mathrm{ran}(\mathrm{dom}\, rckmPath!) \subsetneq \mathrm{ran}\, decisionPathConditionRCKM \wedge \tag{2.06}$$
$$\mathrm{ran}(\mathrm{ran}\, rckmPath!) \subseteq \mathrm{ran}\, ConclusionRCKM \wedge \tag{2.07}$$
$$(\mathrm{ran}(\mathrm{ran}\, rckmPath!) \cap \mathrm{ran}\, decisionPathConditionRCKM) \subseteq$$
$$\mathrm{ran}\, decisionPathConditionRCKM \wedge \tag{2.08}$$
$$0 \leq decPathRCKMAccuracy(rckmPath!) \leq 100 \wedge \tag{2.09}$$
$$head\,(\mathrm{dom}\, rckmPath!) \notin \mathrm{ran}\, ConclusionRCKM \cap \mathrm{ran}\, decisionPathConditionRCKM \wedge \tag{2.10}$$
$$\exists\, dp : decisionPathRCKM \mid dp \in RCKM \bullet$$
$$\mathrm{dom}\, rckmPath! = \mathrm{dom}\, dp \setminus last\,(\mathrm{dom}\, dp) \rightarrowtail last\,(\mathrm{dom}\, dp) = \mathrm{ran}\, rckmPath! \wedge \tag{2.11}$$
$$\mathrm{dom}\, rckmPath! = \exists\, p_{ckm} : decisionPathCKM \mid p_{ckm} \in CKM \bullet$$
$$\mathrm{dom}(p_{pm}?) \cup \mathrm{dom}\, p_{ckm} \wedge \tag{2.12}$$
$$\mathrm{ran}\, rckmPaht! = \mathrm{ran}\, dp_{pm}? \wedge \tag{2.13}$$
$$\forall\, r : RefinedTreatmentPlan \mid r \in refinements? \bullet$$
$$rckmPath! = \bigcup/\{\{t_p : TreatmentPlan \bullet (1 \mathinner{..} \mathrm{dom}\, r, t_p)\} \mathbin{\ominus} \mathrm{dom}\, rckmPath!,\ \mathrm{ran}\, r,$$
$$\{t_p : TreatmentPlan \bullet (\mathrm{dom}\, r + 1 \mathinner{..} \#(\mathrm{dom}\, rckmPath!), t_p)\} \mathbin{\ominus} \mathrm{dom}\, rckmPath!\} \wedge \tag{2.14}$$
$$decisionPathRCKM' = decisionPathRCKM \cup \{\mathrm{dom}\, rckmPath! \mapsto \mathrm{ran}\, rckmPath!\} \wedge \tag{2.15}$$
$$decisionPathConditionRCKM' = decisionPathConditionRCKM \cup \mathrm{dom}\, rckmPath! \wedge \tag{2.16}$$
$$refinedTPlan' = refinedTPlan \cup refinements? \wedge \tag{2.17}$$
$$refinementsDecPath' = refinementsDecPath \cup \{refinements? \rightarrow dp_{pm}?\} \wedge \tag{2.18}$$
$$ConclusionRCKM' = ConclusionRCKM \cup \mathrm{ran}\, rckmPath! \wedge \tag{2.19}$$
$$decPathRCKMAccuracy' = decPathRCKMAccuracy \cup$$
$$\{rckmPath! \mapsto decPathRCKMAccuracy(rckmPath!)\} \wedge \tag{2.20}$$
$$accuracyRCKM' = \frac{accuracyRCKM \times \#RCKM - decPathRCKMAccuracy'(rckmPath!)}{\#RCKM - 1} \wedge \tag{2.21}$$
$$\#RCKM' = \#RCKM + 1 \wedge \tag{2.22}$$
$$evidences' = evidences \cup decPathEvidences(dp_{pm}?) \wedge \tag{2.23}$$
$$decPathRCKMEvidences' = decPathRCKMEvidences \cup$$
$$\{rckmPath! \mapsto decPathEvidences(dp_{pm}?)\} \wedge \tag{2.24}$$
$$RCKM' = RCKM \oplus \{\mathrm{dom}\, rckmPath! \rightarrow \mathrm{ran}\, rckmPath!\} \wedge \tag{2.25}$$
$$refinedCKM' = refinedCKM \oplus \{rckmPath! \mapsto CKM\} \wedge \tag{2.26}$$
$$rootRCKM' = rootRCKM = head\,(\mathrm{dom}\, dp_{pm}? \tag{2.27}$$

$$\Leftrightarrow$$

$$\exists\, rckmPath! : decisionPathRCKM; \quad [def\_RefinedClinicalKnowledgeModel'] \tag{2.28}$$
$$decisionPathConditionRCKM' : \mathbb{F}\, ConditionKMs; \tag{2.29}$$
$$ConclusionRCKM' : \mathbb{F}\, TreatmentPlan; \tag{2.30}$$
$$decisionPathRCKM' : ConditionKMs \rightarrow TreatmentPlan; \tag{2.31}$$
$$decPathRCKMAccuracy' : decisionPathRCKM' \rightarrow accuracy; \tag{2.32}$$
$$evidences' : \mathbb{F}\, Evidences; \tag{2.33}$$
$$decPathRCKMEvidences' : decisionPathRCKM' \rightarrowtail Evidences; \tag{2.34}$$
$$refinedTPlan' : \mathbb{F}\, RefinedTreatmentPlan; \tag{2.35}$$
$$RCKM' : \mathbb{F}\, decisionPathRCKM; \tag{2.36}$$
$$refinedCKM' : decisionPathRCKM' \rightarrow CKM; \tag{2.37}$$
$$refinementsDecPath' : RefinedTreatmentPlan \rightarrowtail decisionPath; \tag{2.38}$$
$$rootRCKM' : \mathrm{seq}\, Condition; \tag{2.39}$$
$$accuracyRCKM' : \mathbb{F}\, \mathbb{Z}; \tag{2.40}$$
$$refinedCKMsAccuracy' : RCKM' \rightarrow accuracy \bullet \tag{2.41}$$
$$(1) \cdot\cdot\ \underline{decisionPathConditionRCKM' = \mathrm{dom}\, decisionPathRCKM'} \wedge \tag{2.42}$$
$$(2) \cdot\cdot\ \underline{ConclusionRCKM' = \mathrm{ran}\, decisionPathRCKM'} \wedge \tag{2.43}$$
$$(3) \cdot\cdot\ \underline{(\mathrm{ran}\, ConclusionRCKM' \cap \mathrm{ran}\, decisionPathConditionRCKM') \subset \mathrm{ran}\, decisionPathConditionRCKM'} \wedge \tag{2.44}$$
$$(4) \cdot\cdot\ \underline{head\,(decisionPathConditionRCKM') \notin \mathrm{ran}\, ConclusionRCKM' \cap \mathrm{ran}\, decisionPathConditionRCKM'} \wedge \tag{2.45}$$
$$(5) \cdot\cdot\ \underline{evidences' = \mathrm{ran}\, decPathRCKMEvidences'} \wedge \tag{2.46}$$
$$(6) \cdot\cdot\ \underline{refinedTPlan' = \mathrm{dom}\, refinementsDecPath'} \wedge \tag{2.47}$$

Continued.. 1 from Proof 2

$$^{(7)..}\underline{0 \leq accuracyRCKM' \leq 100} \wedge \tag{2.48}$$
$$^{(8)..}\underline{RCKM' = \mathrm{dom}\, refinedCKM'} \wedge \tag{2.49}$$
$$^{(9)..}\underline{\forall\, dp : decisionPathRCKM' \mid dp \in RCKM' \bullet \; head\,(\mathrm{dom}\, dp) \notin \mathrm{ran}\, ConclusionRCKM' \cap \mathrm{ran}\, decisionPathConditionRCKM'} \wedge \tag{2.50}$$
$$^{(10)..}\underline{\exists\, dp : decisionPathRCKM' \fatsemi dp_1 : decisionPathRCKM' \mid dp, dp_1 \in RCKM' \bullet \; last\,(\mathrm{dom}\, dp) = \mathrm{ran}\, dp_1 \Leftrightarrow \mathrm{dom}\, dp_1 = \mathrm{dom}\, dp \setminus last\,(\mathrm{dom}\, dp)} \wedge \tag{2.51}$$
$$^{(11)..}\underline{accuracyRCKM' = (\mathbf{let}\, pathsAcc == \{pathsAcc : \mathbb{Z} \mid RCKM' \neq \emptyset \wedge (\forall\, dp : decisionPathRCKM' \mid dp \in RCKM' \bullet pathsAcc = decPathRCKMAccuracy'(dp) + pathsAcc)\})/\#RCKM'} \wedge \tag{2.52}$$
$$^{(12)..}\underline{\forall\, p_{rckm} : decisionPathRCKM' \mid p_{rckm} \in RCKM' \bullet \exists\, p_{pm} : decisionPath, p_{ckm} : decisionPathCKM \mid p_{pm} \in PM \wedge p_{ckm} \in CKM \bullet \mathrm{dom}\, p_{rckm} = \mathrm{dom}\, p_{pm} \cup \mathrm{dom}\, p_{ckm}} \wedge \tag{2.53}$$
$$^{(13)..}\underline{RCKM \neq \emptyset \Rightarrow rootRCKM' = rootRCKM} \wedge \tag{2.54}$$

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

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

Supplementary Appendix C. Simplification of primed statements using logical proof

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 Proof 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.

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 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

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 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;*

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

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 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).

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 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).

Definition 10: *The union ($\cup$) of two functions is not always a function. However, $\uplus$ 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 $\uplus$ is defined as follows;

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

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

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

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 the explanation is required.

Proof 3 Simplification of primed statement-(1)

$$decisionPathConditionRCKM' - \mathrm{dom}\, decisionPathRCKM' \qquad (3.01)$$

$$decisionPathConditionRCKM \cup \mathrm{dom}\, rckmPath! - \mathrm{dom}(decisionPathRCKM \cup \{\mathrm{dom}\, rckmPath! \rightarrow \mathrm{ran}\, rckmPath!\}) \qquad Def.2 : [one - point\ rule] \qquad (3.02)$$

$$decisionPathConditionRCKM \cup \mathrm{dom}\, rckmPath! - \mathrm{dom}\, decisionPathRCKM \cup \mathrm{dom}\{\mathrm{dom}\, rckmPath! \mapsto \mathrm{ran}\, rckmPath!\} \qquad Def.4 : [\mathrm{dom}\ property\ over\ \cup] \qquad (3.03)$$

$$decisionPathConditionRCKM \cup \mathrm{dom}\, rckmPath! = \mathrm{dom}\, decisionPathRCKM \cup \mathrm{dom}\, rckmPath! \qquad [\mathrm{dom}\ def.] \qquad (3.04)$$

$$decisionPathConditionRCKM = \mathrm{dom}\, decisionPathRCKM \qquad Def.5 : [Set\ subtraction] \qquad (3.05)$$

Proof 4 Simplification of primed statement-(2)

$$ConclusionRCKM' = \mathrm{ran}\, decisionPathRCKM' \qquad (4.01)$$

$$ConclusionRCKM \cup \mathrm{ran}\, rckmPath! = \mathrm{ran}(decisionPathRCKM \cup \{\mathrm{dom}\, rckmPath! \mapsto \mathrm{ran}\, rckmPath!\}) \qquad Def.2 : [one - point\ rule] \qquad (4.02)$$

$$ConclusionRCKM \cup \mathrm{ran}\, rckmPath! \quad \mathrm{ran}\, decisionPathRCKM \cup \mathrm{ran}\{\mathrm{dom}\, rckmPath! \rightarrow \mathrm{ran}\, rckmPath!\} \qquad Def.6 : [\mathrm{ran}\ property\ over\ \cup] \qquad (4.03)$$

$$ConclusionRCKM \cup \mathrm{ran}\, rckmPath! - \mathrm{ran}\, decisionPathRCKM \cup \mathrm{ran}\, rckmPath! \qquad [\mathrm{ran}\ def.] \qquad (4.04)$$

$$ConclusionRCKM - \mathrm{ran}\, decisionPathRCKM \qquad Def.5 : [Set\ subtraction] \qquad (4.05)$$

Proof 5 Simplification of primed statement-(3)

$$(\mathrm{ran}\ ConclusionRCKM' \cap \mathrm{ran}\ decisionPathConditionRCKM') \subseteq \mathrm{ran}\ decisionPathConditionRCKM' \tag{5.01}$$

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

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

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

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

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

$$(\mathrm{ran}\ ConclusionRCKM \cap \mathrm{ran}\ decisionPathConditionRCKM) \subseteq \mathrm{ran}\ decisionPathConditionRCKM \quad [a \wedge true \quad a] \tag{5.07}$$

Proof 6 Simplification of primed statement-(4)

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

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

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

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

Proof 7 Simplification of primed statement-(5)

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

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

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

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

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

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 Simplification of primed statement-(7)

$$0 \leq accuracyRCKM' \leq 100 \qquad (9.01)$$

$$\Leftrightarrow \underbrace{accuracyRCKM' \geq 0}_{P_1} \wedge \underbrace{accuracyRCKM' \leq 100}_{P_2} \qquad (9.02)$$

$$accuracyRCKM' \geq 0 \qquad \text{Let consider } P_1 \qquad (9.03)$$

$$\frac{accuracyRCKM \times \#RCKM + decPathRCKMAccuracy(rckmPath!)}{\#RCKM+1} \geq 0 \qquad Def.2: one-point\ rule \qquad (9.04)$$

$$accuracyRCKM \times \#RCKM + decPathRCKMAccuracy(rckmPath!) > 0 \qquad [multiplication] \qquad (9.05)$$

$$accuracyRCKM \times \#RCKM \geq 0 \qquad a - b \geq 0 \wedge b \geq 0 \Rightarrow a \geq 0 \qquad (9.06)$$

$$accuracyRCKM > 0 \qquad Division \qquad (9.07)$$

$$accuracyRCKM' < 100 \qquad \text{Let consider } P_2 \qquad (9.08)$$

$$\frac{accuracyRCKM \times \#RCKM + decPathRCKMAccuracy(rckmPath!)}{\#RCKM+1} \leq 100 \qquad Def.2: one-point\ rule \qquad (9.09)$$

$$accuracyRCKM \times \#RCKM + decPathRCKMAccuracy(rckmPath!) < 100 \times (\#RCKM + 1) \qquad [multiplication] \qquad (9.09)$$

$$accuracyRCKM \times \#RCKM < 100 \times (\#RCKM + 1) \qquad [a.x + y < c.(a + 1) \wedge y < c \Rightarrow a.x < c.(a + 1)] \qquad (9.10)$$

$$accuracyRCKM < 100 \qquad [a.x < c.(a + 1) \Rightarrow x < c] \qquad (9.11)$$

$$0 \leq accuracyRCKM \leq 100 \qquad [P_1 \text{ and } P_2 \text{ proofs} \qquad (9.12)$$

Proof 10 Simplification of primed statement-(8)

$$RCKM' = \mathrm{dom}\, refinedCKM' \qquad (10.01)$$

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

$$RCKM \oplus \{\mathrm{dom}\, rckmPath! \rightarrow \mathrm{ran}\, rckmPath!\} = \mathrm{dom}\, refinedCKM \lhd \mathrm{dom}\{rckmPath! \rightarrow CKM\} \qquad Def.10: [\mathrm{dom}\ over\ \oplus] \qquad (10.03)$$

$$RCKM \oplus \{\mathrm{dom}\, rckmPath! \rightarrow \mathrm{ran}\, rckmPath!\} = \mathrm{dom}\, refinedCKM \lhd rckmPath! \qquad [\mathrm{dom}\, def.] \qquad (10.04)$$

$$RCKM \lhd rckmPath! = \mathrm{dom}\, refinedCKM \lhd rckmPath! \qquad Simplification \qquad (10.05)$$

$$RCKM = \mathrm{dom}\, refinedCKM \qquad Def.5: [Set\ subtraction] \qquad (10.06)$$

Proof 11 Simplification of primed statement-(9)

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

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

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

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

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

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

Proof 12 Simplification of primed statement-(10)

$$\sqcup dp : decisionPathRCKM' \mathbin{⨾} dp_1 : decisionPathRCKM' \mid dp, dp_1 \in RCKM' \bullet$$
$$\mathrm{dom}\, dp_1 = \mathrm{dom}\, dp \setminus last(\mathrm{dom}\, dp) \Rightarrow last(\mathrm{dom}\, dp) = \mathrm{ran}\, dp_1 \qquad (12.01)$$

$$\sqcup dp : (decisionPathRCKM \cup \{\mathrm{dom}\, rckmPath! \mapsto \mathrm{ran}\, rckmPath!\}) \mathbin{⨾}$$
$$dp_1 : (decisionPathRCKM \cup \{\mathrm{dom}\, rckmPath! \mapsto \mathrm{ran}\, rckmPath!\}) \mid$$
$$dp, dp_1 \in (RCKM \oplus \{\mathrm{dom}\, rckmPath! \rightarrow \mathrm{ran}\, rckmPath!\}) \bullet$$
$$\mathrm{dom}\, dp_1 = \mathrm{dom}\, dp \setminus last(\mathrm{dom}\, dp) \Rightarrow last(\mathrm{dom}\, dp) = \mathrm{ran}\, dp_1 \qquad Def.2 : [one - point\ rule] \qquad (12.02)$$

$$\sqcup dp : decisionPathRCKM \mid dp \in RCKM \bullet$$
$$\mathrm{dom}\, rckmPath! - \mathrm{dom}\, dp \setminus last(\mathrm{dom}\, dp) \rightarrow last(\mathrm{dom}\, dp) - \mathrm{ran}\, rckmPath!$$
$$\Rightarrow$$
$$\exists dp : decisionPathRCKM \mathbin{⨾} dp_1 : decisionPathRCKM \mid dp, dp_1 \in RCKM \bullet$$
$$\mathrm{dom}\, dp_1 = \mathrm{dom}\, dp \setminus last(\mathrm{dom}\, dp) \rightarrow last(\mathrm{dom}\, dp) = \mathrm{ran}\, dp_1 \qquad [\sqcup expansion] \qquad (12.03)$$

$$\exists dp : decisionPathRCKM \mathbin{⨾} dp_1 : decisionPathRCKM \mid dp, dp_1 \in RCKM \bullet$$
$$\mathrm{dom}\, dp_1 = \mathrm{dom}\, dp \setminus last(\mathrm{dom}\, dp) \rightarrow last(\mathrm{dom}\, dp) = \mathrm{ran}\, dp_1 \qquad Def.11 : [Modus\ ponens] \qquad (12.04)$$

Proof 13 Simplification of primed statement-(11)

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

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

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

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

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

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

Proof 14 Simplification of primed statement-(12)

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

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

$$\forall p_{rckm} : decisionPathRCKM \mid p_{rckm} \in RCKM \bullet$$
$$\exists p_{pm} : decisionPath, p_{ckm} : decisionPathCKM \mid$$
$$p_{pm} \in PM \wedge p_{ckm} \in CKM \bullet \mathrm{dom}\, p_{rckm} = \mathrm{dom}\, p_{pm} \cup \mathrm{dom}\, p_{ckm} \wedge$$
$$\exists p_{ckm} : decisionPathCKM \mid p_{ckm} \in CKM \bullet$$
$$\mathrm{dom}\, rckmPath! - \mathrm{dom}(p_{pm}?) \cup \mathrm{dom}\, p_{ckm} \qquad \forall expansion \qquad (14.03)$$

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

Proof 15 Simplification of primed statement-(13)

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

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

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