Peer-reviewed version available at Computer Methods and Programs in Biomedicine 2020; doi:10.1016/j.cmpb.2020.105701

*Article*

# Guideline-enabled data driven clinical knowledge model for the treatment of oral cavity cancer acquired through a refined knowledge acquisition method

**Maqbool Hussain[1,2,*], Muhammad Afzal[1], Khalid M. Malik[2,*], Taqdir Ali[3], Wajahat A. Khan[3], Muhammad Irfan[4,5], Arif Jamshed[4], and 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, wajahat.alikhan, sylee}@oslab.khu.ac.kr
[4] 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
[5] 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; (MH, SL, and KMM)

**Abstract:** Validation and verification are the critical requirements in the knowledge acquisition method for 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 for acquiring medical knowledge from clinical practice guidelines (CPGs) and patient data in the Smart CDSS for treatment of oral cavity cancer. The final knowledge model was created by combining knowledge models obtained from CPGs and patient data after passing through a rigorous validation process. However, detailed analysis shows that 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. Therefore, it is required to enhance a hybrid knowledge acquisition method that thwarts the inconsistencies using formal verification. This paper presents the verification process using the Z formal method and its outcome as an enhanced acquisition method – known as the refined knowledge acquisition (ReKA) method. The ReKA method adopted verification method and explored the mechanism of theorem proving using the Z notation. It enables to identify inconsistencies in the validation process used for hybrid knowledge acquisition. Additionally, it refines the hybrid knowledge acquisition method by discovering the missing steps in the current validation process at the acquisition stage. Consequently, ReKA adds a set of nine additional criteria to be used to have a final valid refined clinical knowledge model. The criteria ensure the validity of 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 addition of criteria by 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, 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. In conclusion, ReKA is formally proved method which always yields valid knowledge model having high quality, supporting local practices, and influenced from standard guidelines.

**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 the *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, is 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 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 equipped the method with the sophisticated validation process. Although, at that time, 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

 doi:10.20944/preprints201906.0179.v2

Peer-reviewed version available at Computer Methods and Programs in Biomedicine 2020; doi:10.1016/j.cmpb.2020.105701



internal consistencies. 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 are using 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 was 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 ReKA method, the validity of the knowledge model relied on domain experts. They were free to refine the decision paths in the final knowledge model. This freedom in refinement leads toward a set of inconsistencies which were ignored by the previous method. 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 (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 factor (outbound refinement as PNI does not exist in the healthcare system).

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 main cause of the inconsistencies and guaranteed always producing the valid final knowledge model.

This paper addresses following research questions: a) Does introduction of formal verification using Z notation is able to identify the inconsistencies in the developed knowledge acquisition method with respect to 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 propose formal verification is able to prevent inconsistencies occurred due to complexity and freestyle usage of refinement in the knowledge?; d) Is the knowledge model created using ReKA comparable with existing hybrid knowledge models in terms of validity, quality, and integration with workflows?

The main contribution of this work is as follow:

- The proofs of the theorem using Z notations provides 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 with 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 concrete set of processes which align to various artifacts of Z notation.
- Various aspects of Z notation exploited for the knowledge modeling and associated processes are expressed as the inferenceable mathematical models.
- The ReKA method is formally proved approach which always produces valid knowledge, reflects the CPGs as global evidence and encourage the recommendations well supported by local evidence. At the same time, it is revealed that the model created using ReKA is outperformed compared to the similar approaches available.

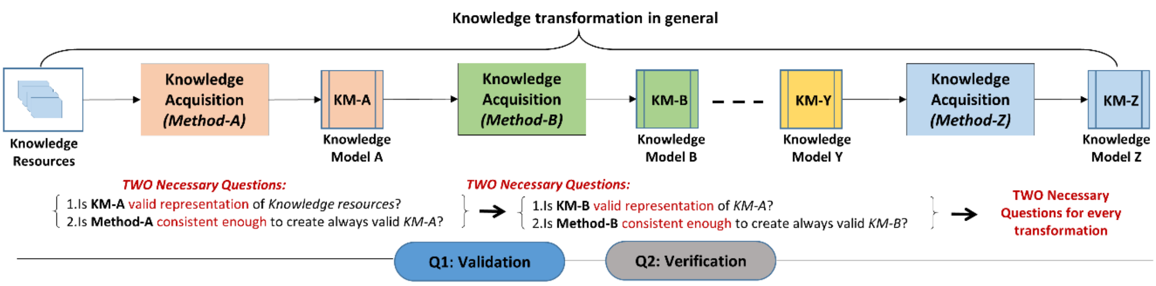



**Figure 1: Knowledge acquisition process**

## 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. In this paper, we are introducing a formal verification for the hybrid knowledge acquisition method, which results in an enhanced method - ReKA. Before going into details of formal verification, it is worthwhile to introduce the knowledge models and validation processes briefly 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 acquisitions 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. So 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 derives from patient data. The recommendation becomes trustworthy for another organization if standard evidence from CPGs and other published studies also associate with the 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 covers-up some cons of the data-driven approach, however, the knowledge models derived from CPGs also come with 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 following 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 huge impact on patient care at that particular jurisdiction.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2019 doi:10.20944/preprints201906.0179.v2

Peer-reviewed version available at Computer Methods and Programs in Biomedicine 2020; doi:10.1016/j.cmpb.2020.105701



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 concentrates 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.

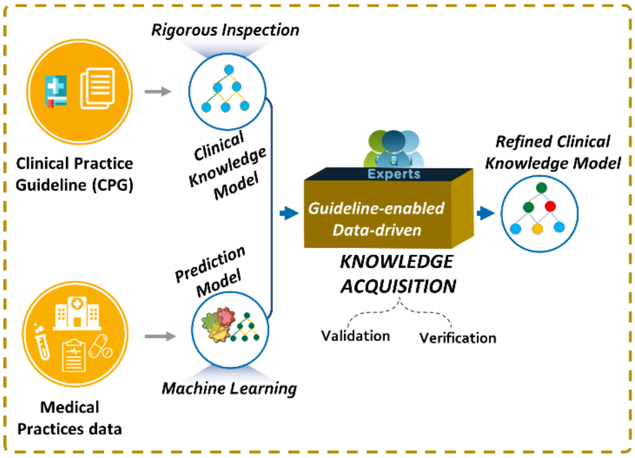



**Figure 2: Hybrid knowledge acquisition method**

*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 this method of explaining 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. PM creation involves the formal machine learning method - CHAID and reflects the local practices from patient data. 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. Decision node represents condition(s) (such as patient symptoms) to select the next branch of the tree among decision paths. Conclusion node reflects the recommendations (such as treatment plan). In CKM, the conclusion node can also play a role of condition node for the next follow-up conclusion. For example, an initial treatment plan for cancer patient may be surgery, and after follow-up, the secondary treatment plan can be radiotherapy only if surgery already is 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 which 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 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., 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.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2019 doi:10.20944/preprints201906.0179.v2

Peer-reviewed version available at Computer Methods and Programs in Biomedicine 2020; doi:10.1016/j.cmpb.2020.105701



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.

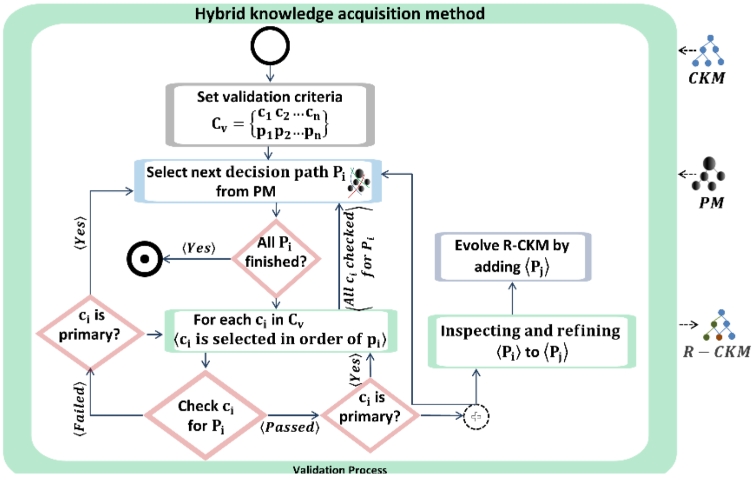



**Figure 3: Validation Process [17]**

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 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 expert 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 time. In order to cope with this issue, the content of this work introduces the verification process using formal Z notations. After applying the Z formalism, the outcome concludes with the detection of inconsistencies in the step of refinement of the validation process. It explicitly enlisted *nine additional criteria* that must be in place after refinements are made to the decision path in R-CKM. The existing knowledge acquisition method is enhanced as ReKA method to accommodate the newly discovered criteria.

## 3. Preliminaries and key motivation of using 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 of these types obey mathematical laws and have rules for reasoning with the information that they contain. At this point, the introduction and use of these concepts are important; however, in this paper, we skip the detailed description of the concepts used in Z notation. So, the important concepts introduced with brief details and all other concepts used in this paper provided in Figure 4. Readers may consult reference materials [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 use to enclose the name of an object. If there are more than one objects, comma uses 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*.

**Definitions and declarations**

| | |
|---|---|
| $a, b$ | Identifiers |
| $p, q$ | Predicates |
| $s, t$ | Sequences |
| $x, y$ | Expressions |
| $A, B$ | Sets |
| $R, S$ | Relations |
| $d; e$ | Declarations |
| $a == x$ | Abbreviated definition |
| $[A]$ | Given set |
| $A ::= b\langle\langle B\rangle\rangle \mid c\langle\langle C\rangle\rangle$ | Free type declaration |
| **let** $a == x$ | Local variable declaration |

**Relations**

| | |
|---|---|
| $A \leftrightarrow B$ | Binary relation |
| dom $R$ | Relation domain |
| ran $R$ | Relation range |
| $R^{\sim}$ | Relational inverse (relational transpose) |
| $A \lhd R$ | Domain restriction |
| $A \rhd R$ | Range restriction |
| $A ⩤ R$ | Domain subtraction (Anti-domain restriction) |
| $A ⩥ R$ | Range subtraction (Anti-range restriction) |
| $R \oplus S$ | Relation overriding |
| $R \times S$ | Cartesian product |
| $a \mapsto b$ | Maplet ( order pair: same as (a,b) ) |

**Logic**

| | |
|---|---|
| $\neg p$ | Logical negation |
| $p \wedge q$ | Logical conjunction |
| $p \vee q$ | Logical disjunction |
| $p \Rightarrow q$ | Logical implication |
| $p \Leftrightarrow q$ | Logical equivalence |
| $\forall x: p$ | Universal quantification |
| $\exists x: q$ | Existential quantification |
| $\exists_1 x: q$ | Existential quantification (exactly one element) |
| **if** $p$ **then** $q$ **else** $r$ | structural conditional logic |

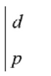
**Sequences**

| | |
|---|---|
| Seq $A$ | Sequence |
| $\mathrm{seq}_1\ A$ | Non-Empty sequence |
| $\langle\rangle$ | Empty sequence |
| $\langle x, y, \ldots\rangle$ | Sequence |
| $s ^\frown t$ | Concatenation |
| $A \upharpoonleft s$ | Extract ( {1,3,6} ↿ ⟨a, b, c, d⟩ will give ⟨a, c⟩ ) |
| $s \upharpoonright A$ | Filter ( ⟨a, b, c, d⟩ ↾ {b, d, e} will give ⟨b, d⟩ ) |
| $s$ in $t$ | Is in (⟨a,b⟩ in ⟨w,y,a,b,c⟩ is true. ⟨b,a⟩ in ⟨w,y,a,b,c⟩ is false) |

**Sets**

| | |
|---|---|
| $x \in y$ | Set membership |
| $\{\}$ | Empty set |
| $\mathbb{N}$ | Set of natural numbers |
| $A \subseteq B$ | Set inclusion |
| $\{x, y, \ldots\}$ | Set of elements |
| $(x, y, \ldots)$ | 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 |
| $\bigcup A$ | Generalization union |
| $\bigcap A$ | Generalization intersection |
| $\#A$ | Size of finite set |
| $\{d; e \ldots \mid p \bullet x\}$ | Set comprehension |

**Schema Notation**

| | |
|---|---|
|  | Schema |
|  | Axiomatic definition |
|  | Schema inclusion |
| $\Delta S$ | Change in schema |
| $\Xi S$ | No schema change |
| $S \widehat{=} T \wedge V$ | Schema definition as value of schema expression |
| $a?$ | Input to an operation |
| $a!$ | Output of an operation |
| $a'$ | State component after operation |
| $S'$ | State schema after operation |

**Functions**

| | |
|---|---|
| $A \rightarrow\!\!\!\!\!\!+\, B$ | Partial function |
| $A \longrightarrow B$ | 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.

*Free type*: Free type allows a variety of data structures to be represented using sets with explicit structuring information. For example, type definition (3) in Figures 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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2019 doi:10.20944/preprints201906.0179.v2
Peer-reviewed version available at Computer Methods and Programs in Biomedicine 2020; doi:10.1016/j.cmpb.2020.105701



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.

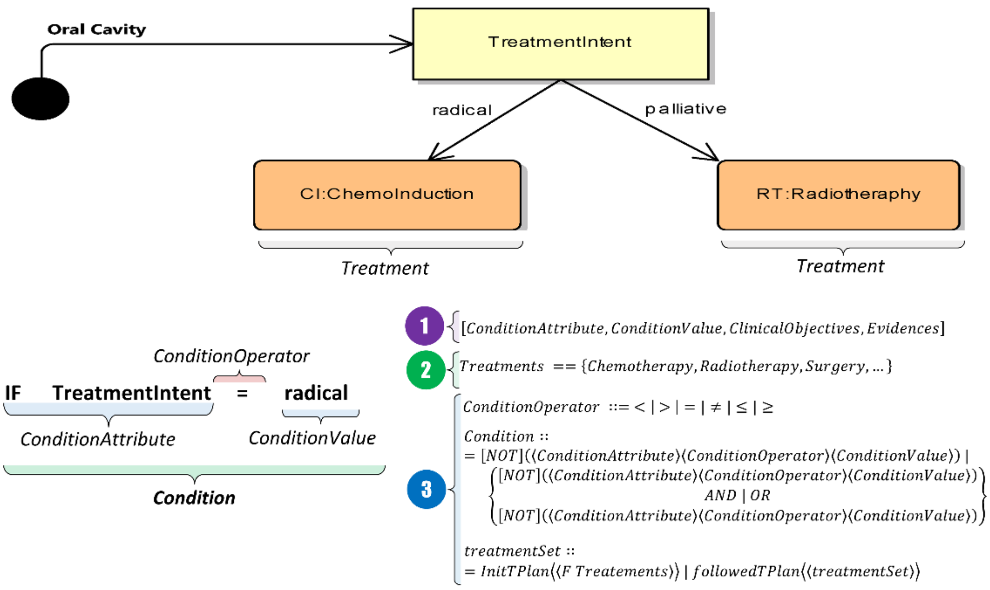



**Figure 5: Declaration, abbreviation and free type examples**

*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.

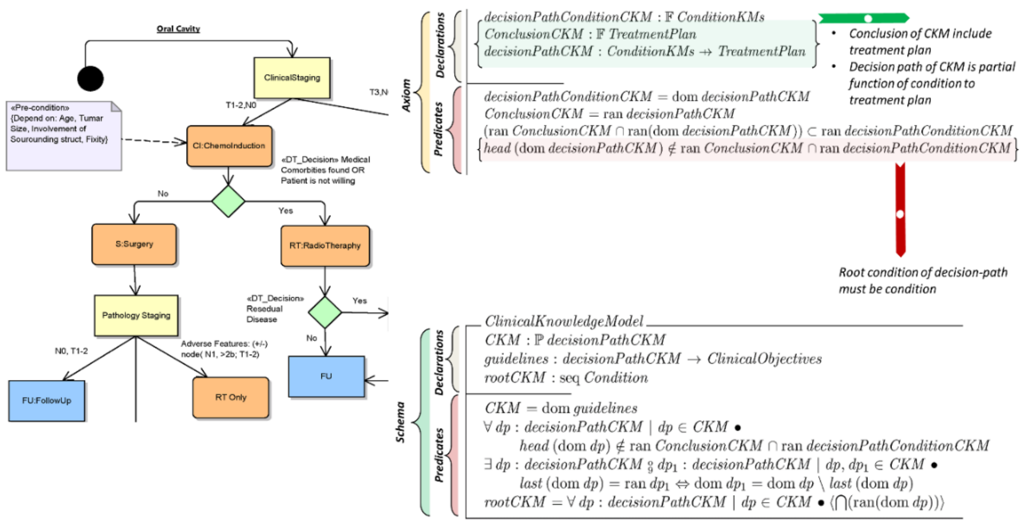



**Figure 6: Axiomatic definition and schema example**

### *3.2. The motivation for formal methods*

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 consistencies 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 consistencies 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 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 with a model checker [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 introduced 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. We used the formal theorem proving mechanism to remove inconsistencies in the method, which ultimately ensures a consistent and valid knowledge model. 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 for 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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2019 doi:10.20944/preprints201906.0179.v2

Peer-reviewed version available at Computer Methods and Programs in Biomedicine 2020; doi:10.1016/j.cmpb.2020.105701



## 4. Methods and Materials

### *4.1. Refined Knowledge Acquisition (ReKA) method*

ReKA method is an enhancement of our hybrid knowledge acquisition method. It follows the same three-phase model used for hybrid knowledge acquisition method. It 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 ReKA method. The extended processes are reflected as an additional layer on the basic processes.

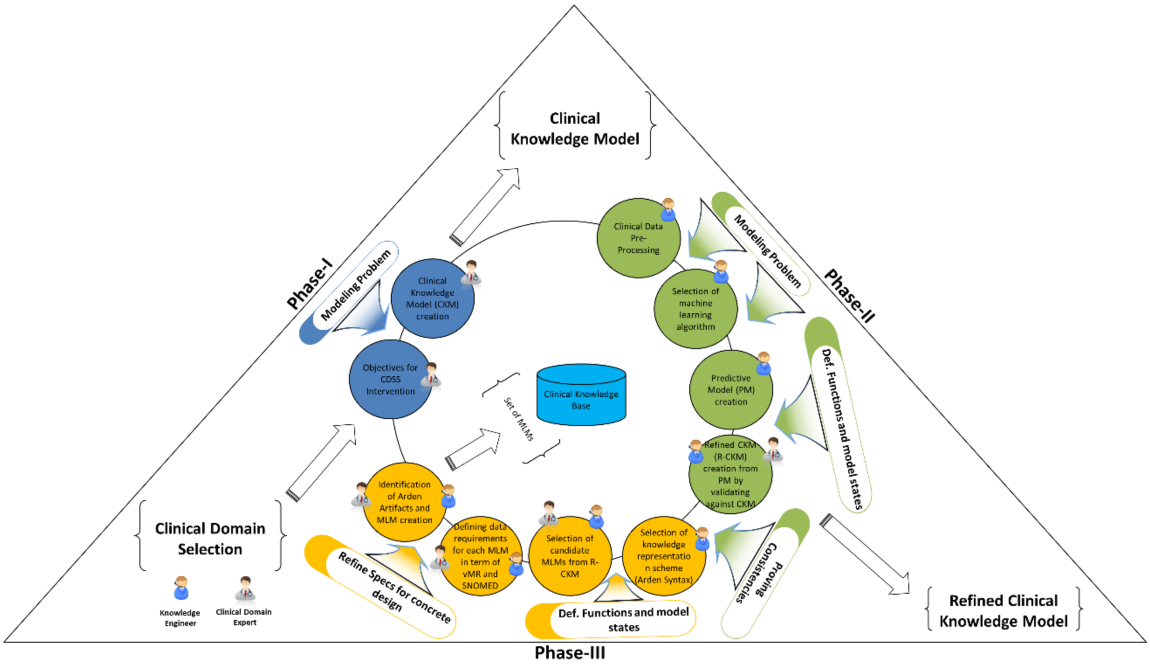



**Figure 7: Extended three-phase model for ReKA**

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 re-used 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 result section, ReKA identified that the modifications are not valid because it creates inconsistencies in the final knowledge model. Furthermore, we have also compared ReKA approach with the most relevant hybrid model acquisition approach. The subsequent sections further discuss the technical details of the formal verification processes used in the ReKA method.

### *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.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2019 doi:10.20944/preprints201906.0179.v2

Peer-reviewed version available at Computer Methods and Programs in Biomedicine 2020; doi:10.1016/j.cmpb.2020.105701



1. *Modeling problem*: This includes tasks used to analyze the problem context and identify all of 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 created which include PM, CKM, and R-CKM. Different constructs of Z notation 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.

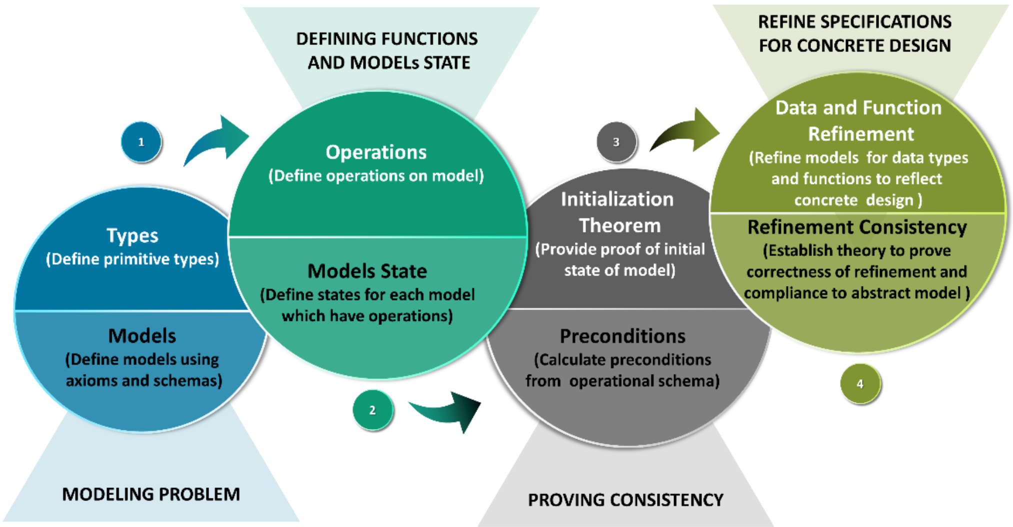



**Figure 8: Formal modeling process**

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 of these 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 model. 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, which target the clinical objectives. 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.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2019 doi:10.20944/preprints201906.0179.v2

Peer-reviewed version available at Computer Methods and Programs in Biomedicine 2020; doi:10.1016/j.cmpb.2020.105701



**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 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).

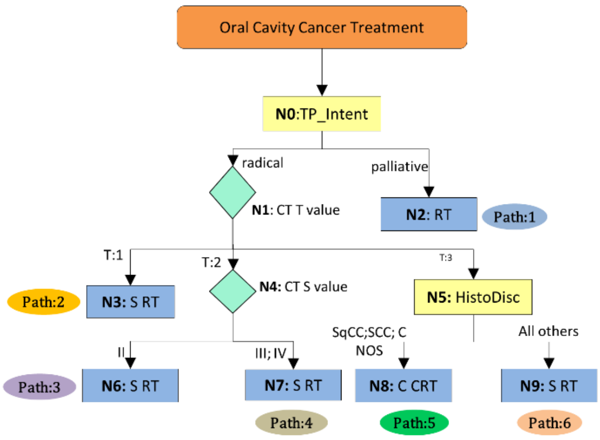
Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2019 doi:10.20944/preprints201906.0179.v2

Peer-reviewed version available at Computer Methods and Programs in Biomedicine 2020; doi:10.1016/j.cmpb.2020.105701



**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)**

 doi:10.20944/preprints201906.0179.v2

Peer-reviewed version available at Computer Methods and Programs in Biomedicine 2020; doi:10.1016/j.cmpb.2020.105701



**Schema 1** Prediction model specifications

*PredictionModel*

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

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

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 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

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

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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2019 doi:10.20944/preprints201906.0179.v2

Peer-reviewed version available at *Computer Methods and Programs in Biomedicine* **2020**; doi:10.1016/j.cmpb.2020.105701



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) reinforce 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*

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

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

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.

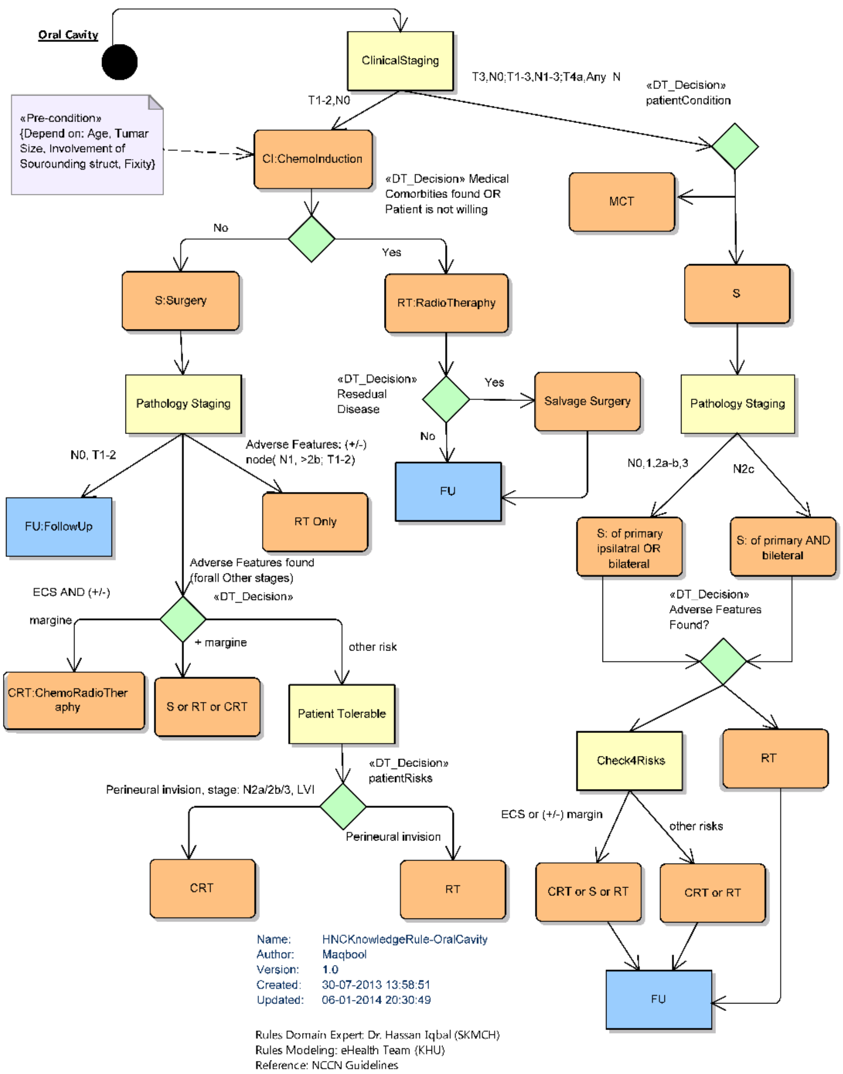



**Figure 10: CKM for treatment intervention in oral cavity cancer(CI: Chemoinduction; CRT: Chemotherapy; CT N: clinical stage N value; CT S: clinical stage S value; CT T: clinical stage T value; ECS: Extracapsular spread; FU: Follow-up; MCT: Multidisciplinary consultation; RT: radiotherapy; S: Surgery) [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

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

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).

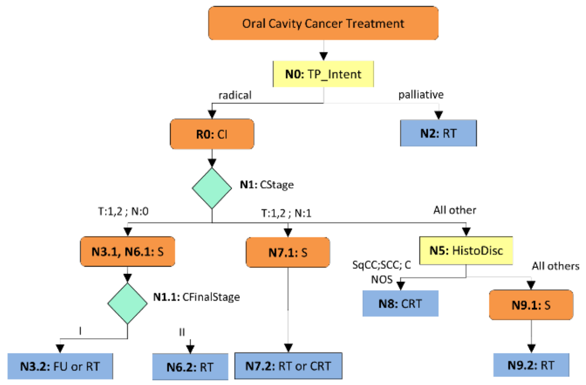
**Schema 3** Refined clinical knowledge model specifications

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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2019 doi:10.20944/preprints201906.0179.v2

Peer-reviewed version available at Computer Methods and Programs in Biomedicine 2020; doi:10.1016/j.cmpb.2020.105701






| 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<br>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;<br>2. TP_Intent: radical → CI → Cstage: T:1,2;N:1 → S<br>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 %;<br>Decisionpath 2: 88.6 %<br>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). The refinement in R-CKM is represented by an *injective function* 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 contents 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 properly represent the validation criteria defined for final knowledge model - R-CKM (See step 1: *Setting validation criteria* at Section 2.2). The schema *PMPathValidation* (Schema 4)

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2019 doi:10.20944/preprints201906.0179.v2

Peer-reviewed version available at *Computer Methods and Programs in Biomedicine* **2020**; doi:10.1016/j.cmpb.2020.105701



models the basic semantics of the validation process. It includes schema *RefinedClinicalKnowledgeModel* (line 1), which is used to associate the validation process with R-CKM. It also provides 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 for the PM decision path (line 3).

**Schema 4** Validation process specifications

*PMPathValidation*

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

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

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

doi:10.20944/preprints201906.0179.v2

Peer-reviewed version available at Computer Methods and Programs in Biomedicine 2020; doi:10.1016/j.cmpb.2020.105701



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.

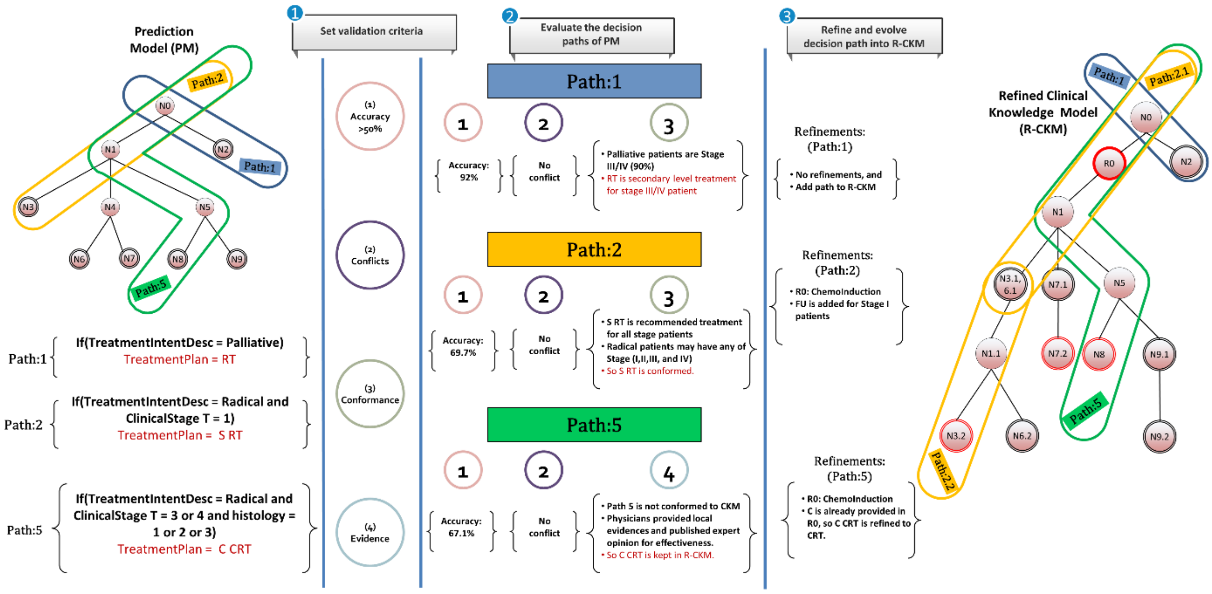



**Figure 12: A running example of validation and refinement of three decision paths of PM**

*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

$$
\begin{array}{ll}
\multicolumn{2}{l}{\underline{\mathit{AddPathRCKM}}} \\
\Delta \mathit{RefinedClinicalKnowledgeModel} & (1) \\
dp_{pm}? : \mathit{decisionPath} & (2) \\
\mathit{refinements}? : \mathbb{F}\ \mathit{RefinedTreatmentPlan} & (3) \\
\mathit{rckmPath}! : \mathit{decisionPathRCKM} & (4) \\
\hline
\mathit{RCKM} \neq \emptyset \Rightarrow \mathit{head}\,(\mathrm{dom}\,dp_{pm}?) = \mathit{rootRCKM} & (5) \\
\forall\, pos : \mathbb{N} \mid pos \in \mathrm{dom}\,\mathit{refinements}? \bullet pos > 1 \land & \\
\qquad pos \leq (\#(\mathrm{dom}\,dp_{pm}?) + \#(\mathrm{ran}\,dp_{pm}?)) & (6) \\
\mathrm{ran}(\mathrm{dom}\,\mathit{rckmPath}!) \subset \mathrm{ran}\,\mathit{decisionPathConditionRCKM} & (7) \\
\mathrm{ran}(\mathrm{ran}\,\mathit{rckmPath}!) \subset \mathrm{ran}\,\mathit{ConclusionRCKM} & (8) \\
(\mathrm{ran}(\mathrm{ran}\,\mathit{rckmPath}!) \cap \mathrm{ran}\,\mathit{decisionPathConditionRCKM}) \subset & \\
\qquad \mathrm{ran}\,\mathit{decisionPathConditionRCKM} & (9) \\
0 \leq \mathit{decPathRCKMAccuracy}(\mathit{rckmPath}!) \leq 100 & (10) \\
\mathit{head}\,(\mathrm{dom}\,\mathit{rckmPath}!) \notin \mathrm{ran}\,\mathit{ConclusionRCKM} \cap \mathrm{ran}\,\mathit{decisionPathConditionRCKM} & (11) \\
\exists\, dp : \mathit{decisionPathRCKM} \mid dp \in \mathit{RCKM} \bullet & \\
\qquad \mathrm{dom}\,\mathit{rckmPath}! = \mathrm{dom}\,dp \setminus \mathit{last}\,(\mathrm{dom}\,dp) \Rightarrow \mathit{last}\,(\mathrm{dom}\,dp) = \mathrm{ran}\,\mathit{rckmPath}! & (12) \\
\mathrm{dom}\,\mathit{rckmPath}! = \exists\, p_{ckm} : \mathit{decisionPathCKM} \mid p_{ckm} \in \mathit{CKM} \bullet & \\
\qquad\qquad \mathrm{dom}(p_{pm}?) \cup \mathrm{dom}\,p_{ckm} & (13) \\
\mathrm{ran}\,\mathit{rckmPaht}! = \mathrm{ran}\,dp_{pm}? & (14) \\
\forall\, r : \mathit{RefinedTreatmentPlan} \mid r \in \mathit{refinements}? \bullet & \\
\qquad \mathit{rckmPath}! = {\frown}/\langle\{t_p : \mathit{TreatmentPlan} \bullet (1\,..\,\mathrm{dom}\,r, t_p)\} \upharpoonright \mathrm{dom}\,\mathit{rckmPath}!,\ \mathrm{ran}\,r, & \\
\qquad \{t_p : \mathit{TreatmentPlan} \bullet (\mathrm{dom}\,r + 1\,..\,\#(\mathrm{dom}\,\mathit{rckmPath}!), t_p)\} \upharpoonright \mathrm{dom}\,\mathit{rckmPath}!\rangle & (15) \\
\mathit{decisionPathRCKM}' = \mathit{decisionPathRCKM} \cup \{\mathrm{dom}\,\mathit{rckmPath}! \mapsto \mathrm{ran}\,\mathit{rckmPath}!\} & (16) \\
\mathit{decisionPathConditionRCKM}' = \mathit{decisionPathConditionRCKM} \cup \mathrm{dom}\,\mathit{rckmPath}! & (17) \\
\mathit{refinedTPlan}' = \mathit{refinedTPlan} \cup \mathit{refinements}? & (18) \\
\mathit{refinementsDecPath}' = \mathit{refinementsDecPath} \cup \{\mathit{refinements}? \mapsto dp_{pm}?\} & (19) \\
\mathit{ConclusionRCKM}' = \mathit{ConclusionRCKM} \cup \mathrm{ran}\,\mathit{rckmPath}! & (20) \\
\mathit{decPathRCKMAccuracy}' = \mathit{decPathRCKMAccuracy} \cup & \\
\qquad \{\mathit{rckmPath}! \mapsto \mathit{decPathRCKMAccuracy}(\mathit{rckmPath}!)\} & (21) \\
\mathit{accuracyRCKM}' = \frac{\mathit{accuracyRCKM} \times \#\mathit{RCKM} + \mathit{decPathRCKMAccuracy}'(\mathit{rckmPath}!)}{\#\mathit{RCKM} + 1} & (22) \\
\#\mathit{RCKM}' = \#\mathit{RCKM} + 1 & (23) \\
\mathit{evidences}' = \mathit{evidences} \cup \mathit{decPathEvidences}(dp_{pm}?) & (24) \\
\mathit{decPathRCKMEvidences}' = \mathit{decPathRCKMEvidences} \cup & \\
\qquad \{\mathit{rckmPath}! \mapsto \mathit{decPathEvidences}(dp_{pm}?)\} & (25) \\
\mathit{RCKM}' = \mathit{RCKM} \oplus \{\mathrm{dom}\,\mathit{rckmPath}! \mapsto \mathrm{ran}\,\mathit{rckmPath}!\} & (26) \\
\mathit{refinedCKM}' = \mathit{refinedCKM} \ominus \{\mathit{rckmPath}! \mapsto \mathit{CKM}\} & (27) \\
\mathit{rootRCKM}' = \mathit{rootRCKM} = \mathit{head}\,(\mathrm{dom}\,dp_{pm}?) & (28)
\end{array}
$$

- *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:

Peer-reviewed version available at *Computer Methods and Programs in Biomedicine* **2020**; doi:10.1016/j.cmpb.2020.105701



**Schema 7** R-CKM state after modification

$$
\begin{array}{lr}
\multicolumn{2}{l}{\mathit{RefinedClinicalKnowledgeModel}'} \\
\mathit{PredictionModel} & (1) \\
\mathit{ClinicalKnowledgeModel} & (2) \\
\mathit{decisionPathConditionRCKM}' : \mathbb{F}\, \mathit{ConditionKMs} & (3) \\
\mathit{ConclusionRCKM}' : \mathbb{F}\, \mathit{TreatmentPlan} & (4) \\
\mathit{decisionPathRCKM}' : \mathit{ConditionKMs} \nrightarrow \mathit{TreatmentPlan} & (5) \\
\mathit{decPathRCKMAccuracy}' : \mathit{decisionPathRCKM}' \rightarrow \mathit{accuracy} & (6) \\
\mathit{evidences}' : \mathbb{F}\, \mathit{Evidences} & (7) \\
\mathit{decPathRCKMEvidences}' : \mathit{decisionPathRCKM}' \nrightarrow \mathit{Evidences} & (8) \\
\mathit{refinedTPlan}' : \mathbb{F}\, \mathit{RefinedTreatmentPlan} & (9) \\
\mathit{refinementsDecPath}' : \mathit{RefinedTreatmentPlan} \rightarrowtail \mathit{decisionPath} & (10) \\
\mathit{RCKM}' : \mathbb{F}\, \mathit{decisionPathRCKM} & (11) \\
\mathit{refinedCKM}' : \mathit{decisionPathRCKM}' \rightarrow \mathit{CKM} & (12) \\
\mathit{rootRCKM}' : \mathrm{seq}\, \mathit{Condition} & (13) \\
\mathit{accuracyRCKM}' : \mathbb{F}\, \mathbb{Z} & (14) \\
\mathit{refinedCKMsAccuracy}' : \mathit{RCKM}' \rightarrow \mathit{accuracy} & (15) \\
\hline
\mathit{decisionPathConditionRCKM}' = \mathrm{dom}\, \mathit{decisionPathRCKM}' & (16) \\
\mathit{ConclusionRCKM}' = \mathrm{ran}\, \mathit{decisionPathRCKM}' & (17) \\
(\mathrm{ran}\, \mathit{ConclusionRCKM}' \cap \mathrm{ran}\, \mathit{decisionPathConditionRCKM}') \subset \mathrm{ran}\, \mathit{decisionPathConditionRCKM}' & (18) \\
\mathit{head}\,(\mathit{decisionPathConditionRCKM}') \notin \mathrm{ran}\, \mathit{ConclusionRCKM}' \cap \mathrm{ran}\, \mathit{decisionPathConditionRCKM}' & (19) \\
\mathit{evidences}' = \mathrm{ran}\, \mathit{decPathRCKMEvidences}' & (20) \\
\mathit{refinedTPlan}' = \mathrm{dom}\, \mathit{refinementsDecPath}' & (21) \\
0 \leq \mathit{accuracyRCKM}' \leq 100 & (22) \\
\mathit{RCKM}' = \mathrm{dom}\, \mathit{refinedCKM}' & (23) \\
\forall\, \mathit{dp} : \mathit{decisionPathRCKM}' \mid \mathit{dp} \in \mathit{RCKM}' \bullet & \\
\quad \mathit{head}\,(\mathrm{dom}\, \mathit{dp}) \notin \mathrm{ran}\, \mathit{ConclusionRCKM}' \cap \mathrm{ran}\, \mathit{decisionPathConditionRCKM}' & (24) \\
\exists\, \mathit{dp} : \mathit{decisionPathRCKM}' \fatsemi \mathit{dp}_1 : \mathit{decisionPathRCKM}' \mid \mathit{dp}, \mathit{dp}_1 \in \mathit{RCKM}' \bullet & \\
\quad \mathit{last}\,(\mathrm{dom}\, \mathit{dp}) = \mathrm{ran}\, \mathit{dp}_1 \Leftrightarrow \mathrm{dom}\, \mathit{dp}_1 = \mathrm{dom}\, \mathit{dp} \setminus \mathit{last}\,(\mathrm{dom}\, \mathit{dp}) & (25) \\
\mathit{accuracyRCKM}' = (\mathbf{let}\ \mathit{pathsAcc} == \{\mathit{pathsAcc} : \mathbb{Z} \mid \mathit{RCKM}' \neq \emptyset \wedge & \\
\quad (\forall\, \mathit{dp} : \mathit{decisionPathRCKM}' \mid \mathit{dp} \in \mathit{RCKM}' \bullet \mathit{pathsAcc} = & \\
\quad \mathit{decPathRCKMAccuracy}'(\mathit{dp}) + \mathit{pathsAcc})\})/\#\mathit{RCKM}' & (26) \\
\forall\, p_{rckm} : \mathit{decisionPathRCKM}' \mid p_{rckm} \in \mathit{RCKM}' \bullet & \\
\quad \exists\, p_{pm} : \mathit{decisionPath}, p_{ckm} : \mathit{decisionPathCKM} \mid & \\
\quad p_{pm} \in \mathit{PM} \wedge p_{ckm} \in \mathit{CKM} \bullet \mathrm{dom}\, p_{rckm} = \mathrm{dom}\, p_{pm} \cup \mathrm{dom}\, p_{ckm} & (27) \\
\mathit{RCKM} \neq \emptyset \Rightarrow \mathit{rootRCKM}' = \mathit{rootRCKM} & (28)
\end{array}
$$

- 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 and evaluation

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 ReKA method. We evaluate the enhanced method (in the context of formal verification) against our initial approach and describes its discrepancies

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2019 doi:10.20944/preprints201906.0179.v2

Peer-reviewed version available at Computer Methods and Programs in Biomedicine 2020; doi:10.1016/j.cmpb.2020.105701



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*

#### 5.1.1. Consistency proof using the Initialization Theorem

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.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2019 doi:10.20944/preprints201906.0179.v2

Peer-reviewed version available at Computer Methods and Programs in Biomedicine 2020; doi:10.1016/j.cmpb.2020.105701



***Definition 2**: For the given predicate:*

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

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

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

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

#### 5.1.2. R-CKM evolution consistency proof using simplified pre-conditions and proving the property composition

The pre-conditions of an operational schema represent a set of states, for which the outcome of the operations is properly defined. The pre-condition of 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 predicate part of the theorem (Theorem 2) which involves 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

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

pre *AddPathRCKM*

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

---

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

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. Additionally, another aspect is that it reveals new pre-condition predicates that were not known in advance. The next section provides a detailed evaluation of the newly discovered pre-condition, which gives birth to an enhanced ReKA method. The primed predicates in Proof 2 (at Supplementary Appendix A) are underlined (numbered 1-13). The prime predicates require simplifications to conclude the proof. To save space, the Supplementary Appendix B presents the simplification proofs.

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

As a consequence of "Proving consistency" mechanism, the main problem is regarding 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 covered-up by introducing nine additional criteria (see Table 1), that are placed after refinement. As an outcome of the formal verification, the enhanced ReKA method introduces to accommodate 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 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.<br>• 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.<br>• 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.<br>• 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.<br>• 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.<br>• 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.<br>• 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.<br>• 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.<br>• 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.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2019 doi:10.20944/preprints201906.0179.v2

Peer-reviewed version available at *Computer Methods and Programs in Biomedicine* **2020**; doi:10.1016/j.cmpb.2020.105701



**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:**<br>*TreatmentIntent: palliative → Treatment Plan(RT): done → Surgery* | **Criteria violation:**<br>*Category: 2*<br>*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:**<br>*Treatment Plan(RT): done → TreatmentIntent: palliative → Follow-up* | **Criteria violation:**<br>*Category: 1, 5*<br>*Specific Criteria: 1, 2, 7* |
| **A rationale for violation:**<br>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:**<br>1. *TreatmentIntent: palliative → Check for risk. ECS: Yes → Treatment Plan: RT or Surgery*<br>2. *TreatmentIntent: palliative → Check for risk. PNI: Yes → Treatment Plan: RT* | **Criteria violation:**<br>*Category: 4*<br>*Specific Criteria: 3*<br>*Category: 3*<br>*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:<br>*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.<br>*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:**<br>*TreatmentIntent: palliative and Treatment Plan(RT): done → Salvage Surgery: done → Follow-up* | **Criteria violation:**<br>*Category 4:*<br>*Specific Criteria: 4*<br>*Category: 3*<br>*Specific Criteria: 6*<br>*Category: 2*<br>*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:<br>*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.<br>*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

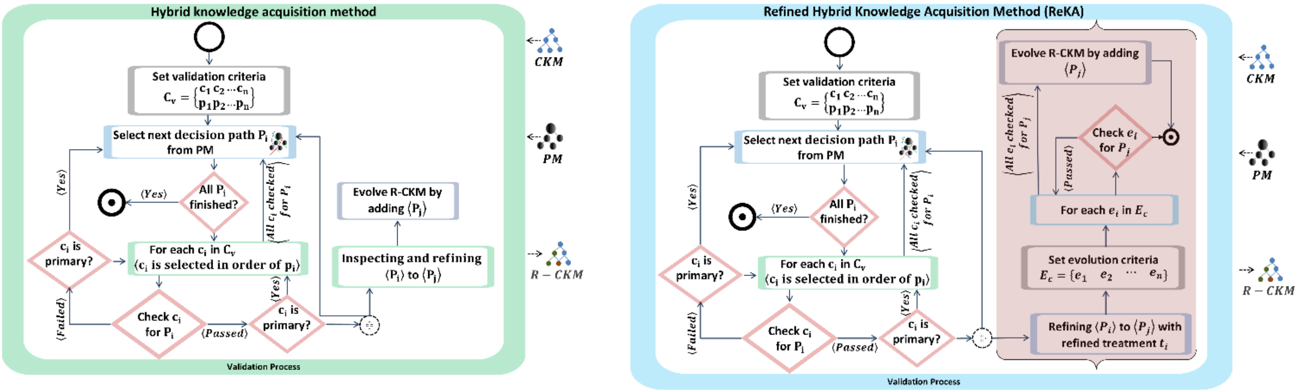
Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2019 doi:10.20944/preprints201906.0179.v2

Peer-reviewed version available at Computer Methods and Programs in Biomedicine 2020; doi:10.1016/j.cmpb.2020.105701



scenarios to decision path by the domain expert( see Path-2 in Figure 12 ). Each of these scenarios introduces inconsistencies which 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 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.

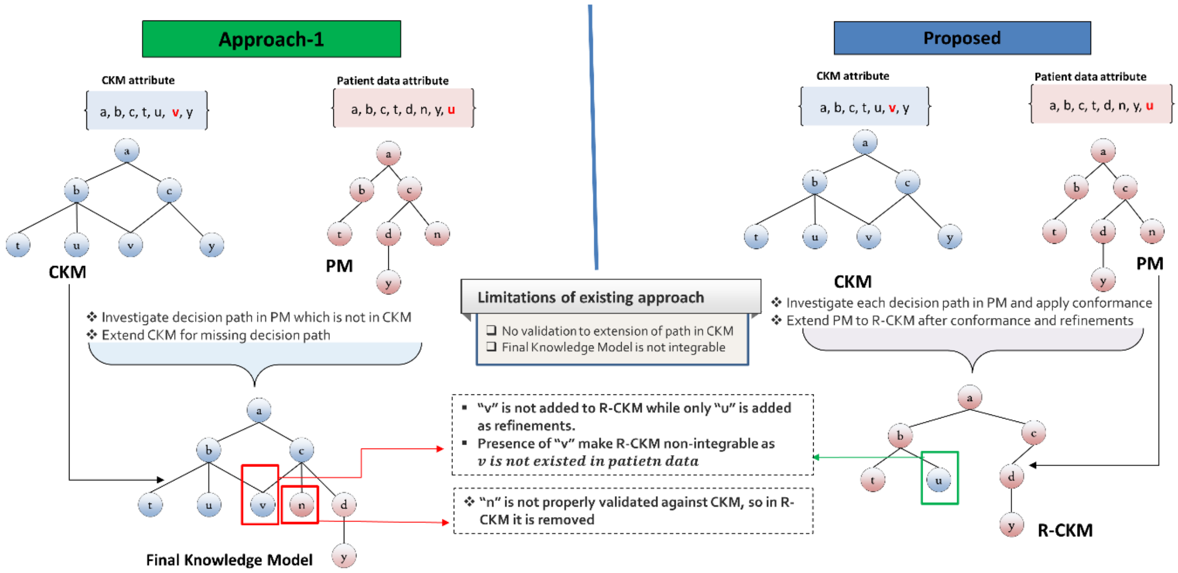



**Figure 14: Demonstration of existing approach vs. proposed approach [22]**

**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 on the SKMCH data set 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 C4.5 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 C4.5, which has overall accuracy of 69.7% on 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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2019 doi:10.20944/preprints201906.0179.v2

Peer-reviewed version available at Computer Methods and Programs in Biomedicine 2020; doi:10.1016/j.cmpb.2020.105701



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 greater performance over Toussi et al. on the local SKMCH dataset.

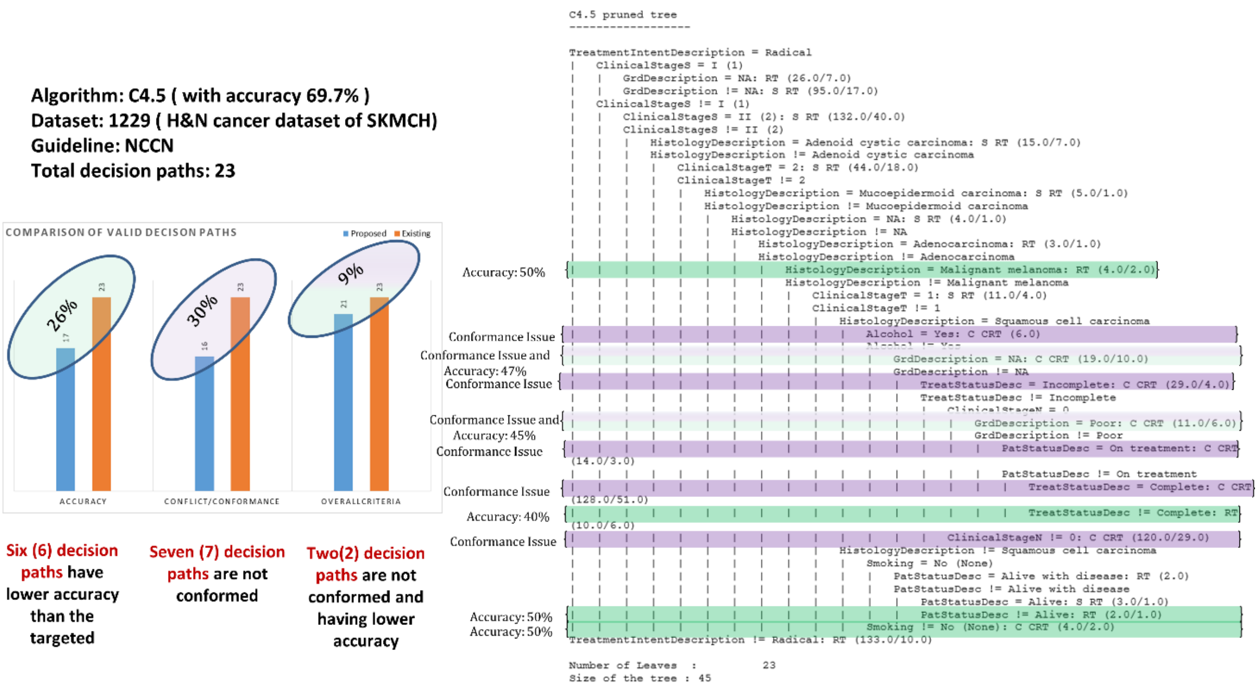



**Figure 15: Comparison of the proposed approach with the existing approach using SKMCH oral cavity cancer data [22]**

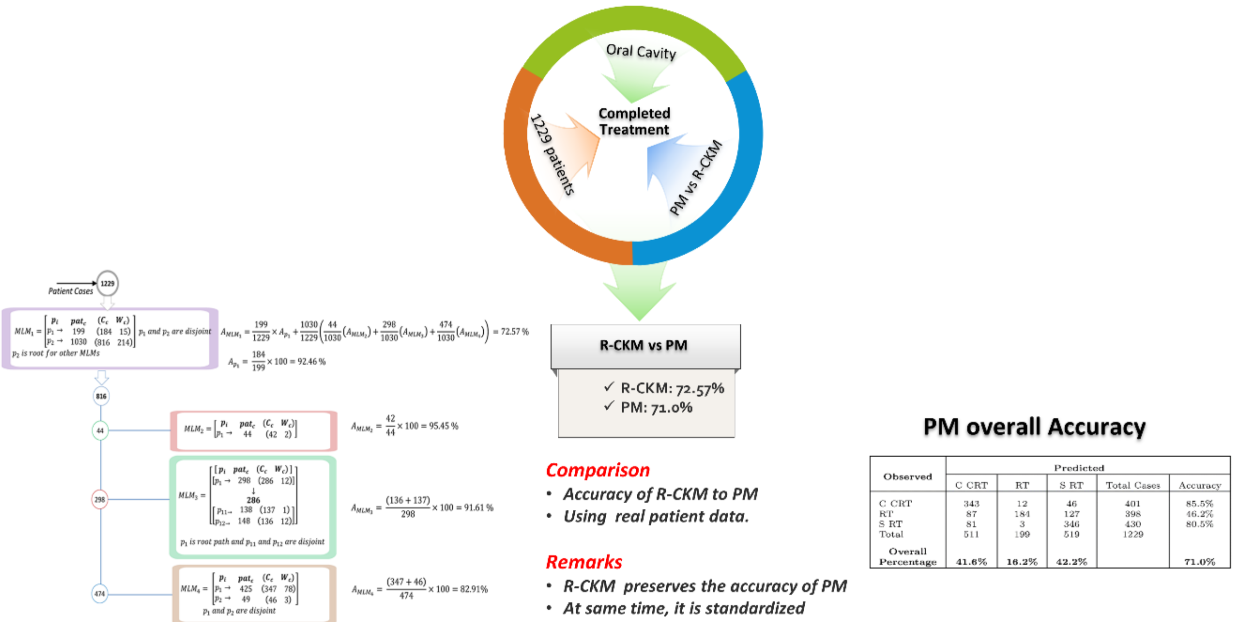



**Figure 16: R-CKM results using oral cavity cancer data [R-CKM is implemented as set a of HL7 MLMs and the evaluation results drawn are based on the structure of MLMs]**

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2019 doi:10.20944/preprints201906.0179.v2

Peer-reviewed version available at Computer Methods and Programs in Biomedicine 2020; doi:10.1016/j.cmpb.2020.105701



## 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 contributed towards 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 only may play a role to establish the required level of trust on the knowledge. Consider an example of calculating TNM staging for cancer [34]. Predefined rules or algorithms and deterministic mappings use for final knowledge modeling of TNM staging. 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 structure and semantics. In this research work, the ReKA method involves diverse knowledge sources; 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 introduced the necessarily missing steps in our previously hybrid knowledge acquisition method, which 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.

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

Formal verification has significant overhead in terms of time and selection of expert resource who have 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 of timely, cost-effective delivery with 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 executable knowledge representation of HL7 MLM (medical logic modules). 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 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 found 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 existing knowledge model with additional constraints. While using ReKA method, it has realized that relying upon only the validation process and partial formalism support of decision tree, the final knowledge model is not always valid. The ReKA method 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. As a 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.

## 7. Conclusion

This paper introduced enhanced ReKA method as a result of the formal verification using Z notation. Z notation proves the consistency of the acquisition process and hence, improved the hybrid knowledge acquisition method. 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 guideline-enabled data-driven knowledge model which support 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 ReKA method include its generality, that 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.

**Author Contributions:** MH designed the study and wrote the contents of the manuscript. MA and KMM assisted in the evaluation of results. TA helped in the implementation of the models into the real healthcare system. WAK and SL provided the review and suggestions for the contents improvements. MI and AJ played roles in the validation of the clinical interpretation of the results and provided the dataset of patient of oral cavity cancer.

**Funding:** 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).

**Acknowledgments:** The authors would like to acknowledge the support of 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. Shibl, R.; Lawley, M.; Debuse, J. Factors influencing decision support system acceptance. *Decision Support Systems* 2013, 54, 953–961.

2. Peleg, M. Computer-interpretable clinical guidelines: A methodological review. *Journal of biomedical informatics* 749 2013, 46, 744–763.
3. Cánovas-Segura, B.; Morales, A.; Juarez, J.M.; Campos, M.; Palacios, F. A lightweight acquisition of expert rules for interoperable clinical decision support systems. *Knowledge-Based Systems* 2019, 167, 98–113.
4. Gomoi, V.; Vida, M.; Robu, R.; Stoicu-Tivadar, V.; Bernad, E.; Lups¸e, O. From data mining rules to medical logical modules and medical advices. *Studies in health technology and informatics* 2013, 192, 1094–1094.
5. Perera, S.; Henson, C.; Thirunarayan, K.; Sheth, A.; Nair, S. Semantics Driven Approach for Knowledge Acquisition from EMRs. *IEEE Journal of Biomedical and Health Informatics* 2014, 18, 515–524.
6. Shalom, E.; Shahar, Y.; Taieb-Maimon, M.; Bar, G.; Yarkoni, A.; Young, O.; Martins, S.B.; Vaszar, L.; Goldstein, M.K.; Liel, Y.; others. A quantitative assessment of a methodology for collaborative specification and evaluation of clinical guidelines. *Journal of biomedical informatics* 2008, 41, 889–903.
7. Serban, R.; Ten Teije, A. Exploiting thesauri knowledge in medical guideline formalization. *Methods of information in medicine* 2009, 48, 468–474.
8. Miller, P.L. Domain-constrained generation of clinical condition sets to help test computer-based clinical guidelines. *Journal of the American Medical Informatics Association* 2001, 8, 131–145.
9. Peleg, M.; Gutnik, L.A.; Snow, V.; Patel, V.L. Interpreting procedures from descriptive guidelines. *Journal of biomedical informatics* 2006, 39, 184–195.
10. Duftschmid, G.; Miksch, S. Knowledge-based verification of clinical guidelines by detection of anomalies. *Artificial intelligence in medicine* 2001, 22, 23–41.
11. Pérez, B.; Porres, I. Authoring and verification of clinical guidelines: A model driven approach. *Journal of biomedical informatics* 2010, 43, 520–536.
12. Liu, S.; Chen, Y.; Nagoya, F.; McDermid, J.A. Formal specification-based inspection for verification of programs. *IEEE Transactions on software engineering* 2012, 38, 1100–1122.
13. Uchitel, S.; Brunet, G.; Chechik, M. Synthesis of partial behavior models from properties and scenarios. *IEEE Transactions on Software Engineering* 2009, 35, 384–406.
14. Diaz, J.; Arroyo, D.; Rodriguez, F.B. On securing online registration protocols: Formal verification of a new proposal. *Knowledge-Based Systems* 2014, 59, 149–158.
15. Newcombe, C.; Rath, T.; Zhang, F.; Munteanu, B.; Brooker, M.; Deardeuff, M. How Amazon web services uses formal methods. *Communications of the ACM* 2015, 58, 66–73.
16. Gamble, R.F.; Stiger, P.; Plant, R. Rule-based systems formalized within a software architectural style. *Knowledge-Based Systems* 1999, 12, 13–26.
17. Hussain, M.; Afzal, M.; Ali, T.; Ali, R.; Khan, W.A.; Jamshed, A.; Lee, S.; Kang, B.H.; Latif, K. Data-driven knowledge acquisition, validation, and transformation into HL7 Arden Syntax. *Artificial intelligence in medicine* 2018, 92, 51–70.
18. Rational Unified Process. http://home.iscte-iul.pt/~{}hro/RUPSmallProjects/ , accessed on December 30, 2017.
19. Jacobson, I.; Booch, G.; Rumbaugh, J.; Rumbaugh, J.; Booch, G. *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. Kilov, H.; Linington, P.F.; Romero, J.R.; Tanaka, A.; Vallecillo, A. The reference model of open distributed processing: foundations, experience and applications. *Computer Standards & Interfaces* 2013, 35, 247–256.
22. Toussi,M.; Lamy, J.B.; Le Toumelin, P.; Venot, A. 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* 2009, 9, 28.
23. Ali, T.; Hussain, M.; Khan, W.A.; Afzal, M.; Hussain, J.; Ali, R.; Hassan, W.; Jamshed, A.; Kang, B.H.; Lee, S. Multi-model-based interactive authoring environment for creating shareable medical knowledge. *Computer Methods and Programs in Biomedicine* 2017, 150, 41–72.
24. IBMSPSS User Guide. IBM SPSS Decision Trees 21. http://www.sussex.ac.uk/its/pdfs/SPSS_Decision_Trees_21.pdf , 2016. (Accessed: 15 May, 2019).
25. Spivey, J.M.; Abrial, J. *The Z notation*; Prentice Hall Hemel Hempstead, 1992.
26. Woodcock, J.; Davies, J. *Using Z: specification, refinement, and proof*; Prentice-Hall, Inc., 1996.
27. Luck, M.; Noriega, P.; Rodriguez-Aguilar, J.A.; Sierra, C.; others. Communicating open systems. *Artificial Intelligence* 2012, 186, 38–94.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2019 doi:10.20944/preprints201906.0179.v2

Peer-reviewed version available at Computer Methods and Programs in Biomedicine 2020; doi:10.1016/j.cmpb.2020.105701




28. d'Inverno, M.; Luck, M.; Georgeff, M.; Kinny, D.;Wooldridge, M. The dMARS architecture: A specification of the distributed multi-agent reasoning system. *Autonomous Agents and Multi-Agent Systems* 2004, 9, 5–53.
29. Sinnott, R.O.; Turner, K.J. Specifying ODP computational objects in Z. *Proc. of FMOODS*, 1996, Vol. 96, pp. 375–390.
30. Bottrighi, A.; Giordano, L.; Molino, G.; Montani, S.; Terenziani, P.; Torchio, M. Adopting model checking techniques for clinical guidelines verification. *Artificial intelligence in medicine* 2010, 48, 1–19.
31. Ten Teije, A.; Marcos, M.; Balser, M.; van Croonenborg, J.; Duelli, C.; van Harmelen, F.; others. Improving medical protocols by formal methods. *Artificial intelligence in medicine* 2006, 36, 193–209.
32. Boiten, E.A.; Derrick, J. From ODP Viewpoint Consistency to Integrated Formal Methods. *Computer Standards & Interfaces* 2013, 35, 269–276.
33. Bowman, H.; Derrick, J.; Linington, P.; Steen, M. FDTs for ODP. *Computer Standards & Interfaces* 1995, 17, 457–479.
34. Brierley, J.D.; Gospodarowicz, M.K.;Wittekind, C. TNM classification of malignant tumours; JohnWiley & Sons, 2016.
35. Grando, M.A.; Glasspool, D.; Fox, J. A formal approach to the analysis of clinical computer-interpretable guideline modeling languages. *Artificial intelligence in medicine* 2012, 54, 1–13.
36. Boehm, B. Software Risk Management. *IEEE Computer Society Press.* Washington DC 1989.

# Supplementary Appendices

## Supplementary Appendix A. Calculating pre-conditions for R-CKM evolution

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

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

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

$$\Leftrightarrow$$

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

$$\Leftrightarrow$$

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

*Preprints* (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 July 2019 doi:10.20944/preprints201906.0179.v2

Peer-reviewed version available at *Computer Methods and Programs in Biomedicine* **2020**; doi:10.1016/j.cmpb.2020.105701



**Continued.. 1** from Proof 2

$$^{(7)}\cdot\cdot\,\underline{0 \leq \mathit{accuracyRCKM}' \leq 100} \wedge \tag{2.48}$$

$$^{(8)}\cdot\cdot\,\underline{\mathit{RCKM}' = \mathrm{dom}\, \mathit{refinedCKM}'} \wedge \tag{2.49}$$

$$^{(9)}\cdot\cdot\,\underline{\forall\, dp : \mathit{decisionPathRCKM}' \mid dp \in \mathit{RCKM}' \bullet} \quad \underline{\mathit{head}\,(\mathrm{dom}\, dp) \notin \mathrm{ran}\, \mathit{ConclusionRCKM}' \cap \mathrm{ran}\, \mathit{decisionPathConditionRCKM}'} \wedge \tag{2.50}$$

$$^{(10)}\cdot\cdot\,\underline{\exists\, dp : \mathit{decisionPathRCKM}' \,⨾\, dp_1 : \mathit{decisionPathRCKM}' \mid dp, dp_1 \in \mathit{RCKM}' \bullet} \quad \underline{\mathit{last}\,(\mathrm{dom}\, dp) = \mathrm{ran}\, dp_1 \Leftrightarrow \mathrm{dom}\, dp_1 = \mathrm{dom}\, dp \setminus \mathit{last}\,(\mathrm{dom}\, dp)} \wedge \tag{2.51}$$

$$^{(11)}\cdot\cdot\,\underline{\mathit{accuracyRCKM}' = (\mathbf{let}\ \mathit{pathsAcc} == \{\mathit{pathsAcc} : \mathbb{Z} \mid \mathit{RCKM}' \neq \emptyset \wedge} \quad \underline{(\forall\, dp : \mathit{decisionPathRCKM}' \mid dp \in \mathit{RCKM}' \bullet \mathit{pathsAcc} =} \quad \underline{\mathit{decPathRCKMAccuracy}'(dp) + \mathit{pathsAcc})\})/\#\mathit{RCKM}'} \wedge \tag{2.52}$$

$$^{(12)}\cdot\cdot\,\underline{\forall\, p_{rckm} : \mathit{decisionPathRCKM}' \mid p_{rckm} \in \mathit{RCKM}' \bullet} \quad \underline{\exists\, p_{pm} : \mathit{decisionPath}, p_{ckm} : \mathit{decisionPathCKM} \mid} \quad \underline{p_{pm} \in \mathit{PM} \wedge p_{ckm} \in \mathit{CKM} \bullet \mathrm{dom}\, p_{rckm} = \mathrm{dom}\, p_{pm} \cup \mathrm{dom}\, p_{ckm}} \wedge \tag{2.53}$$

$$^{(13)}\cdot\cdot\,\underline{\mathit{RCKM} \neq \emptyset \Rightarrow \mathit{rootRCKM}' = \mathit{rootRCKM}} \wedge \tag{2.54}$$

$$\mathit{RCKM} \neq \emptyset \Rightarrow \mathit{head}\,(\mathrm{dom}\, dp_{pm}?) = \mathit{rootRCKM} \wedge \tag{2.55}$$

$$\forall\, \mathit{pos} : \mathbb{N} \mid \mathit{pos} \in \mathrm{dom}\, \mathit{refinements}? \bullet \mathit{pos} > 1 \wedge \quad \mathit{pos} \leq (\#(\mathrm{dom}\, dp_{pm}?) + \#(\mathrm{ran}\, dp_{pm}?)) \wedge \tag{2.56}$$

$$\mathrm{ran}(\mathrm{dom}\, \mathit{rckmPath}!) \subset \mathrm{ran}\, \mathit{decisionPathConditionRCKM} \wedge \tag{2.57}$$

$$\mathrm{ran}(\mathrm{ran}\, \mathit{rckmPath}!) \subset \mathrm{ran}\, \mathit{ConclusionRCKM} \wedge \tag{2.58}$$

$$(\mathrm{ran}(\mathrm{ran}\, \mathit{rckmPath}!) \cap \mathrm{ran}\, \mathit{decisionPathConditionRCKM}) \subset \quad \mathrm{ran}\, \mathit{decisionPathConditionRCKM} \wedge \tag{2.59}$$

$$0 \leq \mathit{decPathRCKMAccuracy}(\mathit{rckmPath}!) \leq 100 \wedge \tag{2.60}$$

$$\mathit{head}\,(\mathrm{dom}\, \mathit{rckmPath}!) \notin \mathrm{ran}\, \mathit{ConclusionRCKM} \cap \mathrm{ran}\, \mathit{decisionPathConditionRCKM} \wedge \tag{2.61}$$

$$\exists\, dp : \mathit{decisionPathRCKM} \mid dp \in \mathit{RCKM} \bullet \quad \mathrm{dom}\, \mathit{rckmPath}! = \mathrm{dom}\, dp \setminus \mathit{last}\,(\mathrm{dom}\, dp) \Rightarrow \mathit{last}\,(\mathrm{dom}\, dp) = \mathrm{ran}\, \mathit{rckmPath}! \wedge \tag{2.62}$$

$$\mathrm{dom}\, \mathit{rckmPath}! = \exists\, p_{ckm} : \mathit{decisionPathCKM} \mid p_{ckm} \in \mathit{CKM} \bullet \quad \mathrm{dom}(p_{pm}?) \cup \mathrm{dom}\, p_{ckm} \wedge \tag{2.63}$$

$$\mathrm{ran}\, \mathit{rckmPaht}! = \mathrm{ran}\, dp_{pm}? \wedge \tag{2.64}$$

$$\forall\, r : \mathit{RefinedTreatmentPlan} \mid r \in \mathit{refinements}? \bullet \quad \mathit{rckmPath}! = {}^\frown/\langle\{t_p : \mathit{TreatmentPlan} \bullet (1 \mathinner{..} \mathrm{dom}\, r, t_p)\} \restriction \mathrm{dom}\, \mathit{rckmPath}!,\ \mathrm{ran}\, r, \quad \{t_p : \mathit{TreatmentPlan} \bullet (\mathrm{dom}\, r + 1 \mathinner{..} \#(\mathrm{dom}\, \mathit{rckmPath}!), t_p)\} \restriction \mathrm{dom}\, \mathit{rckmPath}!\rangle \wedge \tag{2.65}$$

$$\mathit{decisionPathRCKM}' = \mathit{decisionPathRCKM} \cup \{\mathrm{dom}\, \mathit{rckmPath}! \mapsto \mathrm{ran}\, \mathit{rckmPath}!\} \wedge \tag{2.66}$$

$$\mathit{decisionPathConditionRCKM}' = \mathit{decisionPathConditionRCKM} \cup \mathrm{dom}\, \mathit{rckmPath}! \wedge \tag{2.67}$$

$$\mathit{refinedTPlan}' = \mathit{refinedTPlan} \cup \mathit{refinements}? \wedge \tag{2.68}$$

$$\mathit{refinementsDecPath}' = \mathit{refinementsDecPath} \cup \{\mathit{refinements}? \mapsto dp_{pm}?\} \wedge \tag{2.69}$$

$$\mathit{ConclusionRCKM}' = \mathit{ConclusionRCKM} \cup \mathrm{ran}\, \mathit{rckmPath}! \wedge \tag{2.70}$$

$$\mathit{decPathRCKMAccuracy}' = \mathit{decPathRCKMAccuracy} \cup \quad \{\mathit{rckmPath}! \mapsto \mathit{decPathRCKMAccuracy}(\mathit{rckmPath}!)\} \wedge \tag{2.71}$$

$$\mathit{accuracyRCKM}' = \frac{\mathit{accuracyRCKM} \times \#\mathit{RCKM} - \mathit{decPathRCKMAccuracy}'(\mathit{rckmPath}!)}{\#\mathit{RCKM} - 1} \wedge \tag{2.72}$$

$$\#\mathit{RCKM}' = \#\mathit{RCKM} + 1 \wedge \tag{2.73}$$

$$\mathit{evidences}' = \mathit{evidences} \cup \mathit{decPathEvidences}(dp_{pm}?) \wedge \tag{2.74}$$

$$\mathit{decPathRCKMEvidences}' = \mathit{decPathRCKMEvidences} \cup \quad \{\mathit{rckmPath}! \mapsto \mathit{decPathEvidences}(dp_{pm}?)\} \wedge \tag{2.75}$$

$$\mathit{RCKM}' = \mathit{RCKM} \oplus \{\mathrm{dom}\, \mathit{rckmPath}! \mapsto \mathrm{ran}\, \mathit{rckmPath}!\} \wedge \tag{2.76}$$

$$\mathit{refinedCKM}' = \mathit{refinedCKM} \oplus \{\mathit{rckmPath}! \mapsto \mathit{CKM}\} \wedge \tag{2.77}$$

$$\mathit{rootRCKM}' = \mathit{rootRCKM} = \mathit{head}\,(\mathrm{dom}\, dp_{pm}?) \tag{2.78}$$

## Supplementary Appendix B. 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;*

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

Definition 9: Set union and proper subset

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

Proof 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 (∪) of two functions is not always a function. However, ⊕ is the same as a union but ensures that combinations of the two functions are also a function. For two functions f and g, ⊕ is defined as follows:*

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

*For functions f and g, the dom property for ⊕ is defined as follows;*

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

Definition 10: *dom* property over ⊖

***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)

$$decisionPathConditionRCKM' = \operatorname{dom} decisionPathRCKM' \tag{3.01}$$

$$\begin{aligned} &decisionPathConditionRCKM \cup \operatorname{dom} rckmPath! = \\ &\quad \operatorname{dom}(decisionPathRCKM \cup \{\operatorname{dom} rckmPath! \mapsto \operatorname{ran} rckmPath!\}) \qquad Def.2 : [one - point\ rule] \end{aligned} \tag{3.02}$$

$$\begin{aligned} &decisionPathConditionRCKM \cup \operatorname{dom} rckmPath! = \\ &\quad \operatorname{dom} decisionPathRCKM \cup \operatorname{dom}\{\operatorname{dom} rckmPath! \mapsto \operatorname{ran} rckmPath!\} \qquad Def.4 : [\operatorname{dom} property\ over \cup] \end{aligned} \tag{3.03}$$

$$decisionPathConditionRCKM \cup \operatorname{dom} rckmPath! = \operatorname{dom} decisionPathRCKM \cup \operatorname{dom} rckmPath! \qquad [\operatorname{dom} def.] \tag{3.04}$$

$$decisionPathConditionRCKM = \operatorname{dom} decisionPathRCKM \qquad Def.5 : [Set\ subtraction] \tag{3.05}$$

**Proof 4** Simplification of primed statement-(2)

$$ConclusionRCKM' = \operatorname{ran} decisimPathRCKM' \tag{4.01}$$

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

$$\begin{aligned} &ConclusionRCKM \cup \operatorname{ran} rckmPath! = \\ &\quad \operatorname{ran} decisionPathRCKM \cup \operatorname{ran}\{\operatorname{dom} rckmPath! \mapsto \operatorname{ran} rckmPath!\} \qquad Def.6 : [\operatorname{ran} property\ over \cup] \end{aligned} \tag{4.03}$$

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

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

**Proof 5** Simplification of primed statement-(3)

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

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

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

$$\begin{aligned}&((\text{ran}\ ConclusionRCKM \cup \text{ran}(\text{ran}\ rckmPath!)) \cap \\ &\quad \text{ran}\ decisionPathConditionRCKM) \subset \text{ran}\ decisionPathConditionRCKM \qquad Def.7 : [a \cup b \quad a <> b \subset a]\end{aligned} \tag{5.04}$$

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

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

$$\begin{aligned}&(\text{ran}\ ConclusionRCKM \cap \text{ran}\ decisionPathConditionRCKM) \subset \\ &\quad \text{ran}\ decisionPathConditionRCKM \qquad [a \wedge true \quad a]\end{aligned} \tag{5.07}$$

**Proof 6** Simplification of primed statement-(4)

$$head\ (decisionPathConditionRCKM') \not\subset \text{ran}\ ConclusionRCKM' \cap \text{ran}\ decisionPathConditionRCKM' \tag{6.01}$$

$$\begin{aligned}&head\ (decisionPathConditionRCKM \cup \text{dom}\ rckmPath!) \not\subset \\ &\quad \text{ran}(ConclusionRCKM \cup \text{ran}\ rckmPath!) \cap \\ &\quad \text{ran}(decisionPathConditionRCKM \cup \text{dom}\ rckmPath!) \qquad Def.2 : [one - point\ rule]\end{aligned} \tag{6.02}$$

$$\begin{aligned}&head\ (decisionPathConditionRCKM \cup \text{dom}\ rckmPath!) \not\subset \\ &\quad (\text{ran}\ ConclusionRCKM \cup \text{ran}(\text{ran}\ rckmPath!)) \cap \\ &\quad (\text{ran}\ decisionPathConditionRCKM \cup \text{ran}(\text{dom}\ rckmPath!)) \qquad Def.6 : [\text{ran}\ property\ over \cup]\end{aligned} \tag{6.03}$$

$$\begin{aligned}&head\ (decisionPathConditionRCKM) \not\subset \\ &\quad \text{ran}\ ConclusionRCKM \cap \text{ran}\ decisionPathConditionRCKM \qquad Def.7 : [a \cup b = a \Leftrightarrow b \subset a]\end{aligned} \tag{6.04}$$

**Proof 7** Simplification of primed statement-(5)

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

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

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

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

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

**Proof 8** Simplification of primed statement-(6)

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

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

$$\begin{aligned}&refinedTPlan \cup refinements? = \\ &\quad \text{dom}\ refinementsDecPath \cup \text{dom}\{refinements? \mapsto dp_{pm}?\} \qquad Def.4 : [\text{dom}\ property\ over \cup]\end{aligned} \tag{8.03}$$

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

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

**Proof 9** Simplification of primed statement-(7)

$$0 \leq accuracyRCKM' \leq 100 \tag{9.01}$$

$$\Leftrightarrow \frac{accuracyRCKM' \geq 0}{P_1} \wedge \frac{accuracyRCKM' \leq 100}{P_2} \tag{9.02}$$

$$accuracyRCKM' \geq 0 \qquad [Let\ consider\ P_1] \tag{9.03}$$

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

$$accuracyRCKM \times \#RCKM \quad decPathRCKMAccuracy(rckmPath!) > 0 \qquad [multiplication] \tag{9.05}$$

$$accuracyRCKM \times \#RCKM \geq 0 \qquad [a - b \geq 0 \wedge b \geq 0 \rightarrow a \geq 0] \tag{9.06}$$

$$accuracyRCKM \geq 0 \qquad [Division] \tag{9.07}$$

$$accuracyRCKM' < 100 \qquad [Let\ consider\ P_2] \tag{9.08}$$

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

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

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

$$accuracyRCKM \leq 100 \qquad [a.x \leq c.(a + 1) \Rightarrow x \leq c] \tag{9.11}$$

$$0 \leq accuracyRCKM \leq 100 \qquad [P_1\ and\ P_2\ proofs] \tag{9.12}$$

**Proof 10** Simplification of primed statement-(8)

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

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

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

$$RCKM \cup \{\mathrm{dom}\ rckmPath! \rightarrow \mathrm{ran}\ rckmPath!\} = \mathrm{dom}\ refinedCKM \cup rckmPath! \qquad [\mathrm{dom}\ def.] \tag{10.04}$$

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

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

**Proof 11** Simplification of primed statement-(9)

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

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

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

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

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

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

Peer-reviewed version available at Computer Methods and Programs in Biomedicine 2020; doi:10.1016/j.cmpb.2020.105701



**Proof 12** Simplification of primed statement-(10)

$$\begin{array}{l} \exists\, dp : \mathit{decisionPathRCKM}' \fatsemi dp_1 : \mathit{decisionPathRCKM}' \mid dp, dp_1 \in \mathit{RCKM}' \bullet \\ \quad \operatorname{dom} dp_1 = \operatorname{dom} dp \setminus \mathit{last}\,(\operatorname{dom} dp) \Rightarrow \mathit{last}\,(\operatorname{dom} dp) = \operatorname{ran} dp_1 \end{array} \qquad (12.01)$$

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

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

$$\begin{array}{l} \exists\, dp : \mathit{decisionPathRCKM} \fatsemi dp_1 : \mathit{decisionPathRCKM} \mid dp, dp_1 \in \mathit{RCKM} \bullet \\ \quad \operatorname{dom} dp_1 = \operatorname{dom} dp \setminus \mathit{last}\,(\operatorname{dom} dp) \Rightarrow \mathit{last}\,(\operatorname{dom} dp) = \operatorname{ran} dp_1 \end{array} \qquad \mathit{Def.11} : [\mathit{Modus\ ponens}] \qquad (12.04)$$

**Proof 13** Simplification of primed statement-(11)

$$\begin{array}{l} \mathit{accuracyRCKM}' = (\textbf{let } \mathit{pathsAcc} == \{\mathit{pathsAcc} : \mathbb{Z} \mid \mathit{RCKM}' \neq \emptyset \wedge \\ \quad (\forall\, dp : \mathit{decisionPathRCKM}' \mid dp \in \mathit{RCKM}' \bullet \mathit{pathsAcc} = \\ \quad \mathit{decPathRCKMAccuracy}'(dp) + \mathit{pathsAcc})\})/\#\mathit{RCKM}' \end{array} \qquad (13.01)$$

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

$$\begin{array}{l} \dfrac{\mathit{accuracyRCKM} \times \#\mathit{RCKM} + \mathit{decPathRCKMAccuracy}(\mathit{rckmPath!})}{\#\mathit{RCKM}+1} = \\ \quad ((\textbf{let } \mathit{pathsAcc} == \{\mathit{pathsAcc} : \mathbb{Z} \mid \mathit{RCKM} \neq \emptyset \wedge \\ \quad (\forall\, dp : \mathit{decisionPathRCKM} \mid dp \in \mathit{RCKM} \bullet \\ \quad \mathit{pathsAcc} = \mathit{decPathRCKMAccuracy}(dp) - \mathit{pathsAcc})\})) + \\ \quad \mathit{decPathRCKMAccuracy}(\mathit{rckmPath!}))/(\#\mathit{RCKM} + 1) \end{array} \qquad [\forall\, \mathit{simplification}] \qquad (13.03)$$

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

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

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

**Proof 14** Simplification of primed statement-(12)

$$\begin{array}{l} \forall\, p_{rckm} : \mathit{decisionPathRCKM}' \mid p_{rckm} \in \mathit{RCKM}' \bullet \\ \quad \exists\, p_{pm} : \mathit{decisionPath}, p_{ckm} : \mathit{decisionPathCKM} \mid \\ \quad p_{pm} \in \mathit{PM} \wedge p_{ckm} \in \mathit{CKM} \bullet \operatorname{dom} p_{rckm} = \operatorname{dom} p_{pm} \cup \operatorname{dom} p_{ckm} \end{array} \qquad (14.01)$$

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

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

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

**Proof 15** Simplification of primed statement-(13)

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

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

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