Modeling the Form and Function of Clinical Practice Guidelines: An Ontological Model to Computerize Clinical Practice Guidelines

Syed Sibte Raza Abidi [1] and **Shapoor Shayegani**[1]

Abstract. Computerization of Clinical Practice Guidelines (CPG) render them to be executable at the point-of-care. In this paper, we present a knowledge modeling methodology to model the form and function of CPG in terms of a new CPG ontology that supports CPG computerization and execution. We developed a CPG ontology, in OWL using Protégé, to represent both the structural elements and the knowledge objects encapsulated in a CPG. We instantiated over 10 different CPG using our CPG ontology, whereby the instantiated CPG can be executed, with patient data, using a logic-based execution engine to provide patient-specific recommendations. We also investigated the dynamic merging of multiple CPG, at the encoding and execution levels, to handle patient co-morbidities. We evaluated the CPG ontology by examining its representational efficacy to adequately model the salient constructs of a CPG based on an existing CPG modeling formalism.

1 INTRODUCTION

Clinical Practice Guidelines (CPG) comprise a set of evidence-based recommendations to both standardize and optimize the care process, whilst ensuring patient safety and quality of care. Studies show that if CPG are integrated into the clinical workflow they reduce practice variations and costs, whilst improving the quality of care [7]. CPG are written in a free-text format, whereby they describe a set of care plans, described at different levels of abstraction, to manage a specific clinical condition. Basically, a CPG entails a set of systematically orchestrated *processes* that are applied in an episodic manner in line with the patients evolving conditions. A CPG process comprises a set of functional and temporal constraints, desired outcomes, set of actions and decision criterion. It is interesting to observe that the arrangement of processes within a CPG entails a rather intuitive and systematic structure which can be extrapolated to most CPG in general. Modeling, capturing and systematizing this 'generic' CPG structure is an interesting challenge, but it offers the potential to standardize the way we perceive the form and function of CPG, thus facilitating the computerization of CPG for execution purposes. Lately, there has been a renewed interest in computerizing CPG and incorporating them within clinical workflow to provide evidence-mediated decision support. A number of CPG representation formalisms, such as GLIF, EON [5], SAGE and Proforma, have emerged with distinct approaches to computerize a CPG. However, execution of a computerized CPG is still a challenge and only a few CPG modeling frameworks offer execution of a CPG with real-life patient data.

Typically, most CPG conform to a generic form (i.e. structure) that is not necessarily a standard but is inherently omnipresent in most CPG and is a likely consequence of the similarities in the mental models of the CPG authors in general. For instance, most CPG include *care plans* that comprise a number of distinct *tasks* that are systematically *related* and are executed based on certain *decision criteria* and their execution follows a *temporal sequence*. Most tasks have observable *outcomes* that can be measured to determine a particular *recommendation* [6]. The presence of such implicit knowledge constructs and their systematic arrangement implies the presence of a high-level model for CPG; such a model potentially describes a systematic skeletal plan that may serve as the building blocks for a CPG. These plans are both generic and common, hence they are reusable across multiple CPG. We argue that it is both important and useful to first abstract a high-level structual model of a CPG–i.e. identify and model the key knowledge constructs, concepts, relationships, constraints and paragmatcis that are encapsulated within a CPG. In the next step, we can use the CPG model (as a template) to computerize CPG along common concepts and well-recognized relationships. It is our contention that a high-level CPG model representing the form and function of CPG can serve as the vehicle to computerize the CPG knolwedge in a standardized, re-usable and consistent manner. Potentially, there are two ways of developing a CPG model: (a) acquiring it from domain experts through interviews; or (b) inducing it by studying the knowledge artifact. We take an inductive learning approach to develop a high-level CPG model whereby we analyze a corpus of CPG to identify their constituent elements (i.e. form of CPG) and understand how these elements are used to address clinical issues (i.e. function of CPG).

In this paper, we present our methodology to abstract the underlying structural model of a CPG and represent it in terms of an ontological model–i.e. as a rich CPG ontology developed in OWL using Protégé. We present our CPG ontology that semantically models (a) the structrual, conceptual and pragmatic constructs of a CPG; (b) the domain knowledge present in the CPG; and (c) the points to merge/align multiple CPG along common steps to handle patient co-morbidities. Our CPG ontology is used to computerize CPG in a manner that they can be executed through a logic-based CPG execution engine to provide patient-specific recommendations. We establish the efficacy of our CPG ontology by (a) instantiating (i.e. computerizing) 10 different CPG; and (b) comparing its constructs with Peleg's CPG modeling formalism [3].

[1] NICHE Research Group, Faculty of Computer Science, Dalhousie University, Halifax, Canada. Email: sraza@cs.dal.ca

2 METHODOLOGY FOR DEVELOPING CPG ONTOLOGY

To develop our CPG ontology we take an inductive learning approach whereby we analyzed a large number of CPG (i.e. knowledge artifacts) to abstract a high-level 'semantic' model that is representative of the knowledge artifact. This abstraction is represented in terms of a CPG ontology. Our methodology comprises four steps:

2.1 CPG Classification

This step was incorporated to categorize the large body of available CPG along medical and operational dimensions. We classified the range of CPG along six dimensions as follows: (i) Acute vs. Chronic, (ii) Primary vs. Secondary, (iii) Specialty Group (Medical vs. Surgical), (iv) Setting (Inpatient vs. Outpatient), (v) Age Group, and (vi) Orientation (Problem Oriented vs. Task Oriented)–problem oriented CPG provide decision support for specific healthcare problems, such as CPG to manage acute breathing difficulties in children, whereas task oriented CPG focus on how to perform a specific medical task, such as childhood immunization.

2.2 CPG Selection

This step involved the informed selection of multiple CPG to be used as 'exemplar' knowledge artifacts to develop the CPG ontology. The CPG classification scheme was used to objectively select a representative set of CPG to develop the CPG ontology. In total, we selected 20 different CPG that covered all the defined CPG categories. The number of CPG necessary to be analyzed was based on the premise that after analyzing a sufficient number of CPG we will reach a saturation point after which further analysis of additional CPG will not yield any new concepts. In this case, we set the initial saturation point to be 20 CPG, with the provision to select more CPG if new concepts were still being discovered after analyzing 20 CPG.

2.3 CPG Knowledge Modeling and Ontology Engineering

This step involved the analysis of the selected CPG to develop the CPG ontology. The knowledge modeling and ontology engineering was pursued based on the Model-based Incremental Knowledge Engineering (MIKE) approach that involves cyclical iterations of knowledge acquisition, model design, implementation, and evaluation [1]. In the first iteration, we conceptualized the salient CPG elements and modeled them as a preliminary CPG ontology constituting classes, attributes, and constraints. The preliminary CPG ontology was developed using the following concepts: (a) CPG metadata, such as name and description, inclusion and exclusion criteria; (b) clinical activities concerning diagnosis and treatment; (c) clinical decisions; (d) sequential organization of care activities–i.e. the modeling the order, frequency and duration of care activities; (e) clinical interventions, such as surgery and biopsy; (f) examinations; (g) medications; and (h) temporal concepts.

Next, we iteratively applied a middle-out approach to extend the CPG ontology. In each iteration, using the current version of the CPG ontological model, we instantiated the set of CPG selected for that iteration. In this process we extended and refined the existing version of the CPG ontology to account for new concepts, specializations and generalizations of existing concepts and alternate interpretations of existing concepts. The following tasks were performed in each iteration: (i) the set of candidate CPG were studied to extract and explicate the clinical knowledge; (ii) CPG elements were identified and analyzed, which led to either the specification of new or the refinement of existing ontology classes, attributes and constraints to model the CPG elements; (iii) Changes to the ontological model were re-evaluated to ensure semantic consistency.

By the time we reached the final iteration–i.e. working with CPG number 16 to 20, the CPG ontology had consolidated to the extent that no significant alternations/additions were necessary. At this point, we concluded that the CPG ontology had 'saturated'–i.e. it was sufficiently expressive in its representational constructs (i.e. classes and relationships) and was deemed capable of instantiating any given CPG. At this point the ontology engineering exercise was stopped and the resultant CPG ontology was next subjected to evaluation.

2.4 CPG Ontology Evaluation

In this step, we evaluated the representational adequacy and efficacy of our CPG ontology by instantiating five new test CPG. The test CPG were selected guided by our initial CPG categorization scheme and included a diverse set of CPG. During evaluation we examined for possible ontology deletions (missing concepts), substitutions (ambiguous concepts) and insertions (superfluous concepts) that were necessary to instantiate the test CPG. In addition, we measured our CPG model using the eight dimensions of the guideline comparison proposed by [3]. Finally, we measured the ability of our CPG ontology to merge two different CPG so that they can be executed concurrently.

3 DESCRIPTION OF CPG ONTOLOGY

Our CPG ontology represents the structural constructs and practice-oriented knowledge inherent in CPG in terms of 50 classes, 161 attributes and 589 instances. The class hierarchy is linked by the class subsumption relation, i.e. is-a relationship. Classes are denoted using UPPERCASE and attributes with *italics*. Description of all the characteristics of the CPG is not possible due to space constraints, but below we briefly describe our CPG Ontology.

The metadata (or maintenance information) for a CPG is captured by the class CLINICAL-GUIDELINE which identifies each CPG using the following attributes: *approved-by*, *author*, *authoring-date*, *comments*, *description*, *desired-outcome*, *exclusion-criteria*, *goals*, *inclusion-criteria*, *references*, etc.

In order to model the sequential execution of clinical activities suggested by a CPG, we decomposed a CPG into a set of **steps** that are followed sequentially-i.e. in order for a step to execute its preceding step must be completed. We defined a class GUIDELINE-STEP to represent the steps of a CPG (shown in figure 2). In order to model the sequence of steps in a CPG, we defined the *first-step* for CLINICAL-GUIDELINE to denote the first step in the CPG, and *next-step* attribute for each GUIDELINE-STEP instance to signify the next step that needs to be pursued. We identified three sub-classes of the class GUIDELINE-STEP–i.e. ACTION-STEPS, DECISION-STEPS and ROUTE-STEPS, with further sub-classifications.

ACTION-STEPS represent activities performed in the CPG's workflow. We have modeled various clinical activities as sub-classes of ACTION-STEPS—i.e. ASSESSMENT-STEP, DIAGNOSTIC-STEP, VISIT-STEP DIAGNOSTIC-CHOICE-STEP, TREATMENT-STEP, TREATMENT-CHOICE-STEP, SCHEDULE-STEP, PLAN-EXPLICATION-STEP, NOTIFICATION-STEP, EDUCATION-STEP and ADMISSION-STEP.

DECISION-STEPS represent points in the CPG where a decision needs to be made to determine the next set of activities. DECISION-STEPS are different from ACTION-STEPS because their next steps are based on the result of a decision. Their next step is modeled using the *decision-option* attribute that can hold multiple instances of *decision-option*, each instance specifying the next step that need to be taken should it be selected. We have defined two sub-classes for DECISION-STEP i.e. (a) PROVIDER-DECISION-STEP that models decisions made by the care provider, whereby its *responsible* attribute specifies the care provider who is responsible to make the decision; and (b) SYSTEM-DECISION-STEP is used when the decision-making logic is clearly specified in the CPG and in the presence of the necessary data the system can make a decision.

ROUTE-STEPS specify the flow of activities in the CPG, and have the following three sub-classes: (a) BRANCH-STEP to specify a branching point that coordinates two or more subsequent steps to be executed in parallel; (b) SYNC-STEP to synchronize (or merge) steps that have been previously branched. In order to ensure that all the steps are synchronized we have introduced an attribute *preceding-steps-to-be-completed* that ensures that all the preceding steps are completed before the control is passed to the next step; and (c) LOOP-STEP to repeat one or more guideline steps.

INTERVENTION models the set of diagnostic and treatment interventions performed during the delivery of care. There are two sub-classes: INTERVENTION-FOR-TREATMENT and INTERVENTION-FOR-DIAGNOSIS. INTERVENTION-FOR-TREATMENT represents the different types of treatment interventions, and has attributes *indication*, *contraindication*, *criteria-to-check-effects*, *action-if-adverse-effects*. Its sub-classes are: PRESCRIPTION, PROCEDURE-FOR-TREATMENT and RADIOTHERAPY. INTERVENTION-FOR-DIAGNOSIS represents different diagnostic interventions that are further distinguished by the following sub-classes: PROCEDURE-TO-DIAGNOSIS, DIAGNOSTIC-IMAGING GROUP-OF-DIAGNOSTIC-PROCESSES, PHYSICAL-EXAM, and LABORATORY-EXAM.

DRUG-ORDER models the type of medication(s) and their ordering information through attributes such as *drug*, *drug-route* and *dose-schedule*. A separate class DRUG is created to facilitate the merging of CPG (explained later) with the following attributes *allowed-roles-to-request*, *concept-URI* refers to the right concept in a standard medical terminology, *drug-contraindication*, *drug-indication*, *generic-name*, *notes-for-patient*, *other-names*, *recommended-dose* and *toxic-dose*. The DOSE-SCHEDULE captures information about the dosage of the ordered medication and the schedule for its consumption using attributes such as *dose*, *dose-unit* and *dose-measured*. The schedule for consuming the drug is defined by the *schedule* which holds an instance of the SCHEDULE class.

To model temporal concepts, we have defined the following two classes: DURATION that defines a time measurement value–i.e. *time-value* one week, and a measurement unit-i.e. *time-unit* with values such as Minute, Hour, Day or Week. The SCHEDULE class models different types of temporal schedules to organize activities. To specify a schedule we defined attributes such as *schedule-type*, *repetitions* and *duration*.

4 USING OUR CPG ONTOLOGY FOR CPG MERGING

The objective of CPG merging is to align two or more CPG to potentially handle a patient's co-morbidities which may demand the concurrent application of more than one CPG. The net outcome of CPG merging is not a new 'merged' CPG, rather the alignment of common plans/steps that exist across multiple active CPG in order to (a) realize a comprehensive decision model, encompassing multiple CPG, that targets the overall care of the patient as opposed to the treatment of just a disease, (b) optimize resources by reducing repetitive tests/actions, and (c) efficient execution of overlapping processes and interventions. Merging two (or more) CPG whilst maintaining clinical pragmatics is quite challenging because (a) recommendations that are common across multiple CPG are not necessarily administered at the same time, and (b) certain parts of the merging CPG may later result in contradictions or adverse effects. Our CPG ontology allows CPG merging at the following two levels:

Encoding level: This level approaches CPG merging during the ontology-based CPG encoding stage. The CPG ontology decomposes a CPG at the level of generic skeletal plans that can be re-used across multiple CPG. For instance, concepts such as tests and medications are usually included in multiple CPG and can therefore be defined once and then can be re-used in multiple CPG. At the encoding level, two CPG can be merged if they entail a similar plan. Figure 1 shows that CPG A and B can be merged because they both have the common step of CT-Scan. This concept is defined through the IMAGING class, which is a sub-class of INTERVENTION-FOR-DIAGNOSIS, which will have 'CT Scan' as a common instance found in both CPG A and B. In our CPG ontology, we purposely separated the INTERVENTION class from GUIDELINE-STEPS to allow medical interventions to be defined once but re-used across multiple CPG, thus serving as CPG merging points.

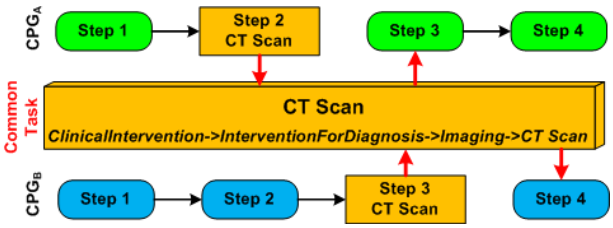

Figure 1. CPG merging at the concept CT scan

Execution level: This level proposes the merging of common steps between multiple CPG whilst they are in execution-i.e. the execution engine looks forward for common steps and tries to merge the two concurrently running CP in order to avoid duplication of common steps. Merging CPG during execution can help eliminate repetitive steps that have long wait times, are expensive, or have potential adverse effects (i.e. radiography). Figure 2 shows the potential CPG merging points in terms of GUIDELINE-STEPS. We do not use Decision Steps as merging points because decisions are CPG-dependent and their effect is local to a CPG. Likewise, Route Steps do not qualify as merging points because they do not represent any activity, rather they facilitate the flow of activities in a CPG.

Merging CPG at the execution level needs to take into account the fact that concurrently running CPG may not necessarily have a common step to be executed at the same time. This means that the merging CPG may need to synchronize their execution, for instance one CPG may have to wait for the other CPG so that the common step can be executed, or one CPG may have to use the results of a common test done by another earlier executed CPG. We can handle three CPG merging scenarios via our CPG ontology.

Scenario 1: Both guidelines recommend a common step at the

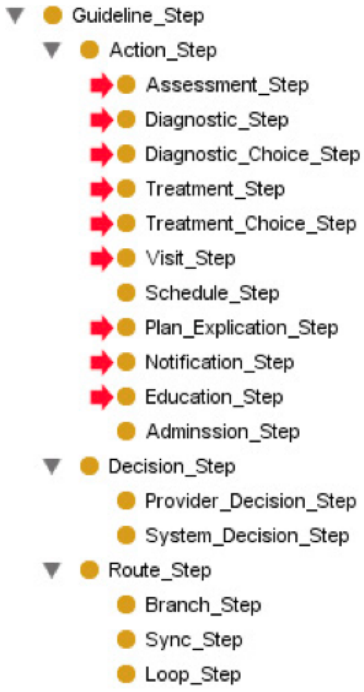

Figure 2. CPG steps as merging points are highlighted with an arrow

same time. Both CPG merge at the common step and then branch off to their respective paths when the common step is completed.

Scenario 2: In case the common step is not executed at the same time by two CPG, then CPG merging is still possible if the CPG in front (in terms of its execution order) can wait before executing the common step–i.e. the *ability-to-wait* constraint for the common step can be satisfied. To model this merging scenario, for each ACTION-STEP we have specified the following attributes: (a) *expected-duration* to represent the average execution time for a step; and (b) *logic-to-calculate-acceptable-wait* to specify the criteria to calculate the maximum acceptable wait time before starting the step. To estimate the length of time needed for the trailing CPG to reach the common step, the execution engine can add up *expected-duration* attributes of each step from its current state to the common step and if this time is less than the acceptable wait time for the common step, then the execution engine can withhold the execution of the leading CPG so that the execution of the common step is synchronized with the trailing CPG.

Scenario 3: Two CPG can be merged if they can re-use the results of a common step. To ensure that the result is not outdated, we have specified an attribute *acceptable-duration-of-results-if-available* that will ensure that the trailing CPG is using a valid result. If the result of a common test performed by the leading CPG is deemed outdated then the test will be repeated.

To understand CPG merging via our CPG ontology, lets assume that for a patient we need to apply two CPG: (1) Evaluation of Acute Chest Pain for Acute Coronary Syndromes and (2) Detection and Diagnosis of Hypertension. For CPG-1, a fragment of the CPG says "Consider treatment for other diagnoses if ACS is ruled out. If ACS is possible, admit patient to emergency department, perform ECG and measure cardiac markers, and decide if ACS is present". CPG-2 states "Do the following laboratory tests for patients with hypertension: Urinalysis, Blood chemistry (potassium, sodium and creatinine), Fasting glucose, Standard 12 lead ECG. Note that both CPG are recommending to perform ECG (Electrocardiogram)". An execution engine using our CPG ontology will be able to detect the common step (ECG) because both CPG will have an instance of DIAGNOSTIC-STEP in their sequence of recommendations which will indicate the need to perform an ECG. This class has an attribute *diagnostic-intervention* which holds an instance 'ECG' of the class INTERVENTION-FOR-DIAGNOSIS. Separating the intervention object from guideline steps not only allows definition of reusable objects which leads to a smoother encoding process, but also enables the execution engine to detect that multiple CPG may require the same intervention.

5 CPG ONTOLOGY EVALUATION

For CPG ontology evaluation we conducted three activities:

1. Instantiating 5 new test CPG to measure the representational efficacy of the ontology [4]. The evaluation concluded that our CPG ontology possessed the necessary representational expressiveness to instantiate the test CPG.
2. Evaluating the semantic correctness of the ontology [2]. We satisfied the three main principles relevant to ontologies–i.e. for our CPG ontology (i) each hierarchy had a single root; (ii) Non-leaf classes had at least two children; and (iii) each child was different from its parent and the siblings were different from one another.
3. Using Peleg et al. [3] framework for CPG modeling formalism as a comparator, we checked whether our CPG ontology supports the eight dimensions suggested by Peleg. This is a rather novel way of comparing a CPG against an existing modeling formalism. Below we briefly report how our CPG ontology complies with the eight structural dimensions of Peleg's formalism.

5.1 Organization of CPG Plan Components

This dimension demands the CPG modeling formalism to describe the structure of CPG plans, its components, and control flow of its processes. Our CPG ontology represents CPG as Task Network Models using distinct classes to model the core plans and components of a CPG, such as actions, decisions, and sub-plans. Our CPG ontology defines the control flow of processes using sequential, parallel or iterative activities. Earlier, we highlighted several classes to model different CPG components, most notably the class GUIDELINE-STEPS that serves as CPG plan components.

5.2 Specification of Goals and Intentions

This dimension entails the specification of the CPG goals and intentions. In our CPG ontology we are able to specify various CPG goals and intentions both as free text and formal expressions. More specifically, the CLINICAL-GUIDELINE class has two attributes for this purpose: *goals* is used to address the CPG goals as free text for user display or CPG indexing, and *desired-outcome* which expresses the desired intention of the CPG as a formal expression.

5.3 Model of Guideline Actions

This dimension concerns both the representation structure of CPG actions and how refining actions are handled if they fail to produce the intended outcomes? Our CPG ontology represents a wide range

of clinical actions, such as diagnostic and treatment actions, visits to healthcare providers, communications with providers through notifications, patient education, admissions into a medical setting, and the scheduling of clinical actions. Furthermore, the CPG ontology adequately handles action refinement at various levels. For instance the INTERVENTION-FOR-TREATMENT has an attribute *criteria-to-check-effects* which holds the criteria to check the effects of the treatment. If these criteria show that there is an adverse effect associated with the treatment, then the action specified in *action-if-adverse-effect* is to be carried out to refine the treatment. Furthermore, the class OUTCOME has an attribute *achievement-measurement-criteria* which holds the criteria to be checked if the desired outcomes of a CPG are achieved. If the criteria is met then the actions specified through *action-if-achieved* will be executed, else activities modeled by *action-if-not-achieved* will be performed.

5.4 Decision Models

This dimension concerns the presence of a definitive structure for decision constructs. In our CPG ontology, we have various instances of PROVIDER-DECISION-STEP and SYSTEM-DECISION-STEP that serve as switch constructs because they describe mutually exclusive CPG branches that are selected based on the result of the decision making process.

5.5 Languages to Specify Decision Criteria

This dimension addresses the expression languages used to represent decision criteria, including pre- and post-conditions of CPG plan components, and the criteria that control plan execution states. For this purpose, we have created the class CONDITION with attributes *logic-text* which holds the actual logic text as an expression language, and attribute *data-elements-involved* to explicitly define the data elements used in the logic.

5.6 Data Interpretation or Abstraction

This dimension deals with the interpretation or abstractions of data elements to conceptualize CPG logic and data. An abstraction example is to use drug groups, such as ACE Inhibitor, to represent individual drugs. We have defined two classes, TREATMENT-CHOICE-STEP and DIAGNOSTIC-CHOICE-STEP to abstract general treatment or diagnostic concepts, for instance Proton Pump Inhibitor drugs are modeled through a TREATMENT-CHOICE-STEP which allows several choices for individual drugs belonging to this group.

5.7 Representation of a Medical Concept Model

This dimension deals with the ability to refer to medical terminology concepts. We support the incorporation of medical terminology concepts through attributes, such as *concept-URI* that hold the URI of the concept in the target terminology, which in turn facilitates communication between our ontology and health information systems.

5.8 Patient Information Model

This dimension deals with mechanisms to reference patient data. In our CPG ontology, patient data can also be addressed by DATA-ELEMENTS which refers to the URI of the patient data element through its *concept-URI* attribute.

In conclusion, we were able to establish that our CPG ontology is (a) sufficiently generic and expressive and generic in nature to potentially computerize any previously unseen (new) CPG with execution capabilities; (b) compliant to ontological principles; and (c) representative of key CPG constructs.

6 CONCLUDING REMARKS

In this paper we have described a knowledge modeling approach to model both the form and function of CPG in terms of a detailed CPG ontology. The CPG ontology not just captures the structural elements of a CPG but also the domain-specific knowledge held within the CPG. We recognize that our CPG ontology identifies some CPG constructs that overlap with other CPG representation formalisms, such as SAGE and EON, nevertheless we argue that our CPG ontology provides a more fine-grained classification of CPG elements which allows for a more detailed representation and a specialized classification of concepts inherent within a CPG. Execution of a computerized CPG is made possible by the decomposition of a CPG into multiple skeletal components and interlinking them such that each action entails the next action link, thus forming a chain of actions. This approach renders the CPG execution to be modular and better tractable.

Our CPG ontology addresses the merging of concurrently active CPG. This has been made possible by defining independent and reusable CPG objects which render CPG merging at the encoding and execution levels. We believe that CPG merging need to be pursued at the CPG knowledge level, whereby we apply high-level axioms to the domain knowledge, encoded using a CPG ontology, to achieve CPG merging that is both medically valid and clinically pragmatic.

We believe that our CPG ontology can help standardize CPG development as it can serve as a standard 'template' for knowledge explication and crystallization between health professionals engaged in a CPG authoring exercise.

Finally, we believe that this exercise of modeling the knowledge structures inherent in CPG has led to a deeper understanding of the form and function of CPG. Our CPG ontology can serve as an intermediate representation or mediator between the original CPG text and existing CPG representation formalisms.

ACKNOWLEDGEMENTS

This project is funded by a grant from Agfa Healthcare (Canada).

REFERENCES

[1] J. Angele, D. Fensel, and R. Studer, *Domain and task modeling in MIKE*, 149–163, Proceedings of IFIP WG 8.1/13.2 Joint Working Conference, 1996.

[2] O. Bodenreider, B. Smith, A. Kumar, and A Burgun, 'Investigating subsumption in snomed ct: An exploration into large description logic-based biomedical terminologies', *Journal of Artificial Intelligence in Medicine*, **39**, 183–195, (2007).

[3] M. Peleg, S. Tu, J. Bury, P. Ciccarese, J. Fox, R. A. Greenes, R. Hall, P. D. Johnson, Jones N., and A. Kuma, 'Comparing computer-interpretable guideline models: A case-study approach', *Journal of American Medical Informatics Association*, **10**, (2003).

[4] R. Porzel and R. Malaka, *A task-based approach for ontology evaluation*, Proc. ECAI Workshop on Ontology Learning and Population, 2004.

[5] S. Tu and M. Musen, *Modeling data and knowledge in the EON guideline architecture*, Medinfo, 2001.

[6] S. W. Tu, J. Campbell, and M. A. Musen, *The structure of guideline recommendations: A synthesis*, 679–683, Proc. AMIA Symposium, 2003.

[7] R. D. Zielstorff, 'Online practice guidelines: Issues, obstacles and future practice', *Journal of American Medical Informatics Association*, **5**, 227–236, (1998).