



Downloaded from: Dalhousie's Institutional Repository

Type of print: Publisher's PDF
Originally published as:

Hurley, K. F., & Abidi, S. S. R. (June 20, 2007). Ontology Engineering to Model Clinical Pathways: Towards the Computerization and Execution of Clinical Pathways. 536-541.

Ontology Engineering to Model Clinical Pathways: Towards the Computerization and Execution of Clinical Pathways

Katrina F Hurley and Syed Sibte Raza Abidi
NICHE Research Group, Faculty of Computer Science, Dalhousie University,
Halifax, Canada
kfhurley@dal.ca & sraza@cs.dal.ca

Abstract

Clinical pathways translate evidence-based recommendations into locally practicable, process-specific algorithms that reduce practice variations and optimize quality of care. Our objective was to abstract practice-oriented knowledge from a cohort of real clinical pathways and represent this knowledge as a clinical pathway ontology. We employed a four step methodology: (1) knowledge source identification and classification of clinical pathways according to variations in setting, stage of care, patient type, outcome and specialty; (2) iterative knowledge abstraction using grounded theory; (3) ontology engineering as adapted from the Model-based Incremental Knowledge Engineering approach; and, (4) ontology evaluation through encoding a sample of real clinical pathways. We present our Clinical Pathway Ontology that offers a detailed ontological model describing the structure and function of clinical pathways. Our ontology can potentially integrate with a healthcare semantic web, and ontologies for clinical practice guidelines, patients and institutions to form the foundational knowledge for generating patient-specific CarePlans.

1. Introduction

Clinical Pathways (CPs) are evidence-based patient care algorithms that describe the process of care for specific medical conditions within a localized setting. CPs are derived from clinical practice guidelines (CPG), clinical evidence and best practices. CPs transform practice recommendations into locally practicable, process-specific algorithms that reduce variations in practice and optimize quality of care [1]. From a practical standpoint, CPs can be operationalized at the point-of-care because they are sensitive to the resource constraints of the healthcare institution such as the availability of medications, procedures, and diagnostic interventions. We posit that these localized CPs can be further ‘adapted’ to the evolving healthcare needs of individual patients to generate patient-specific *CarePlans* [2] that can guide patients’ process of care.

Existing paper-based CPs are static, stand-alone, disease-specific, non-adaptive and isolated from clinical applications [3]. Operationalization of CPs is premised on their ‘computerization’ so that they can be integrated within clinical workflow to provide point-of-care decision-support. The computerization of CPs brings to the forefront several challenges: (i) abstracting practice-oriented knowledge from paper-based CPs; (ii) representing CP knowledge and structure in a semantically-rich formalism [4]; (iii) managing clinical evidence to maintain CP integrity [5]; (iv) adapting institution-specific CPs to individual patient needs; (v) integrating CPs with patient-specific data and clinical applications; and, (vi) executing CPs in real-time to provide decision-support and care planning. In this paper we address the first two challenges.

The objective of our research program is to generate patient-specific *CarePlans* [2]. As a first step, we take a knowledge management approach to ontologically model CPs in terms of their content, structure and function. In this paper, we present our *CP*

Ontology that entails a semantic abstraction of CPs in terms of practice-related knowledge, structural elements and relationships between elements. We describe our CP Ontology development methodology through the following tasks: (a) knowledge source identification and classification; (b) knowledge abstraction from real CPs; (c) ontology engineering and (d) ontology evaluation. We demonstrate the integrity and medical significance of our CP Ontology, through the instantiation of real CPs.

2. Developing a Clinical Pathway Ontology: Our Approach

2.1. Knowledge identification and classification

In identifying sources of CP knowledge, we considered (i) tacit knowledge of domain experts; (ii) previously developed CP ontologies; (iii) published literature on CPs; and (iv) existing CPs in use at healthcare institutions. However, we faced the following challenges: (a) access to the tacit knowledge of CP developers, researchers and clinicians was not practical; and, (b) there was no available ontology that describes CPs. We found a substantial number of paper-based CPs, and these served as the primary source of practice-related knowledge to guide our CP ontology engineering. We classified CPs along five axes: (i) setting; (ii) stage of care; (iii) patient condition; (iv) intervention; and, (v) medical specialty. For ontology engineering purposes, the classification of the available CPs along these five axes ensures that the final ontology purports a reasonable representation of the spectrum of CPs that are currently in use in healthcare institutions.

2.2. Knowledge abstraction

The knowledge abstraction process was guided by the principles of grounded theory which involves building understanding from the ground up such that the resultant framework is grounded in the artifact itself [7]. Our *reverse engineering* approach involved an iterative analysis and constant comparison of emerging theory to that derived from the previous iteration. In the first iteration, we developed a preliminary CP model by abstracting knowledge from: (i) the Care Pathway Conceptual Structure (CPCS) [4]; (ii) published literature on CPs; and (iii) a pilot study involving five CPs that vary along the classification axes. This process yielded eleven descriptors of CPs: collaborative, targeted, process-specific, institution-specific, time-oriented, quality-focused, evidence-based, adaptable, variance-conscious, patient-centered, and documentation and communication medium. In the next iteration, we extended and refined our preliminary model of CPs by applying it to five additional CPs. In all we conducted five such iterations, using a total of 30 different CPs to develop a CP model that was subsequently used to engineer our CP Ontology.

2.3. Ontology engineering

Our ontology engineering process was adapted from the Model-based Incremental Knowledge Engineering (MIKE) Process, comprising cyclical iterations of knowledge acquisition, model design, implementation, and evaluation [6]. Using our CP model as a starting point, we used a middle-out approach to ontology development. This approach defines the most salient concepts first, and generalizes (higher level classes) and specializes (lower level classes) as needed [8]. In order to realize our CP Ontology, we identified a class hierarchy, classes, slots and their facets (cardinality, value type, and domain) [9] – i.e. a preliminary CP Ontology that models the structure and content of paper-based CPs. We refined the preliminary CP ontology through a process of instantiation – i.e. using our ontology to encode CPs from our sample and examining its representational adequacy and accuracy. Based on our cyclical engineering approach, we redefined, added and/or deleted classes, slots and relationships as required. The final outcome of the ontology engineering

step was an ‘expressive’ ontology that semantically represented CPs. We used the Protégé-Frames ontology editor and knowledge acquisition system to develop our CP Ontology [10].

2.4. Ontology evaluation

We adapted a task-based evaluation strategy [11]. The evaluation task was to encode real CPs with our CP Ontology. We chose a broad sample of five new CPs that varied along the five CP classification axes. Next, we mapped concepts expressed in the paper-based CP to our CP Ontology which required qualitative judgments about the meaning and intent of lexical items. Through concept mapping, we identified ontology deletions (missing concepts), substitutions (ambiguous concepts) and insertions (superfluous concepts).

3. Description of Our CP Ontology

Our CP Ontology represents the practice-oriented knowledge inherent in CPs. Using the knowledge elements abstracted from our sample of CPs, we defined 141 classes, 230 slots, 1600 instances and 10 constraints. The class hierarchy is linked by the class subsumption relation (the **is-a** relationship). Class names are denoted using **SMALL CAPS** and slots with *italics*. Below we describe classes from our CP Ontology.

CLEARINGINFORMATION specifies maintenance information for the CP via slots such as *ClinicalPathwayTitle*, *IntendedAudience*, *DateDeveloped* and *ContentSource*.

TARGETPOPULATION defines the patients for whom the CP is intended, using *Age*, *Sex*, *InclusionCriteria*, and *ExclusionCriteria*. Criteria may be linked using logical operators. For example, the Acute Coronary Syndrome (ACS) Pathway [12] specifies inclusion of patients with: chest pain at rest for more than 15 minutes OR suspected acute myocardial infarction.

GOAL describes the overall aim or intention of the CP. For example, the goal may be to reduce length of stay to three days. This class differs from **OUTCOME** which is patient-centered and indicates the intention or purpose of particular processes and tasks. Both goals and outcomes should be specific and measurable.

SETTING describes the location or environment in which the CP is to be carried out, with the understanding that some CPs integrate care across multiple care settings.

ROLE indicates the parties to whom accountability for particular tasks is assigned. The slots specify the *Name*, *InstitutionalAffiliation*, *ClinicalPathwayAffiliation*, *DescriptionOfDuties*, and tasks (*AccountableFor*). For a *CareTeamMember*, one can also specify the *CareMemberType* (e.g. physician, nurse) and *ClinicalSpeciality*.

KNOWLEDGESOURCE denotes the evidence from which the CP is derived including *KnowledgeSourceType* (e.g. CPG), *KnowledgeSourceCitation* and *CitationSource*. This class links to **CLEARINGINFORMATION** as *ContentSource* is an instance of **KNOWLEDGESOURCE**.

PROCESS denotes the larger processes that comprise a CP, where each process has a specified start and end point. Processes may have outcomes and are comprised of a series of tasks. **TASK** specifies the action(s) that are to be carried out by the healthcare team, where the patient is the subject of the task. For example, the ACS Pathway comprises admission, assessment, diagnosis, treatment and decision processes. The assessment process comprises four tasks that are carried out concurrently: attach the ECG monitor, estimate patient weight, measure the patient’s blood pressure, and apply

the pulse oximeter. Tasks can also be carried out in parallel (*TaskConcurrentWith*) or in series (*TaskFollowedBy* and its inverse *TaskPrecededBy*). In our conceptual framework, tasks are related based on finish-to-start relationships – i.e. the next task does not begin until the predecessor tasks are complete. Unless otherwise specified, successive tasks are assumed to be eligible to commence immediately upon completion of the preceding task. Lags or waits can be specified via **CLINICALPATHWAYCONTROL** or implied through the task time interval. **TASK** has 14 subclasses that capture task type such as **ASSESSMENT**, **PRESCRIPTION**, and **DECISION-MAKING**.

DECISION-MAKING has two types: **DECISIONBYUSER**, denoting those decisions that necessitate the use of human judgment; and, **AUTOMATEDDECISION**, denoting decisions that can be made by a computerized system using available information. **DECISIONBYUSER** includes the additional slot *UrgencyLevel* to communicate the urgency with which the user should undertake the decision.

Decision-making involves the evaluation of a set of decision options, represented using: **BINARYLOGIC**, **CRITERIA** and **DECISIONLOGIC**. **BINARYLOGIC** expresses logical conjunctions. **CRITERIA** includes *CriterionDescription*, *LogicalConjunction* (instance of **BINARYLOGIC**) and the option of recursively specifying the *NextCriterion* (instance of **CRITERIA**). **DECISIONLOGIC** represents *DescriptionOfDecisionOption* and *CriteriaToEvaluateOption* (instance of **CRITERIA**). **BRANCHTASK** encodes the pathway branches that correspond to the different decision options. Taken together, these classes and slots allow our CP Ontology to present a decision-making task, the set of decision alternatives, the criteria along which each alternative is to be evaluated and the CP branches contingent on the decision.

VARIANCE describes deviations from the program of care outlined in the CP, as noted during CP execution. It includes documentation of **VARIANCEDATA**, allowing clinicians to describe the nature of the variance. Variances are classified by **VARIANCETYPE** as relating to the patient or family, health care provider, institution, community, or task. Variance data are used as inputs for **VARIANCEANALYSIS**.

AUDIT describes a process of CP assessment or evaluation. **AUDIT** is a superclass of **VARIANCEANALYSIS** and also includes classes that describe baseline audit, implementation analysis and clinical performance analysis.

Drawing on the OWL Time Ontology, we defined five main classes to describe the concept of time in the CP Ontology: **TEMPORALPROPERTY**, **DATE TIME**, **MILESTONE**, **TEMPORALUNIT**, and **TEMPORALENTITY**. **TEMPORALPROPERTY** specifies duration and frequency. **DATE TIME** specifies actual dates and times and is used to timestamp items such as the CP development (*DateDeveloped*) and CP revision (*DateRevised*). **MILESTONE** represents a significant event or stage in patient care such as completion of a procedure or discharge from care. **MILESTONEUNIT** (subclass of **TEMPORALUNIT**) denotes the units by which milestones are measured – i.e. **CLINICALEVENT**, **ACHIEVEMENTOF OBJECTIVE**, and **CLINICALSTAGE**. It was necessary to represent the concept of milestones as many CPs are specified in terms of these units rather than proper units of time. **PROPERTIMEUNIT** (subclass of **TEMPORALUNIT**) represents traditionally recognized units of time such as second, minute, hour, and day.

With respect to time, we divided CPs into three types: (i) proper time driven; (ii) milestone driven; and, (iii) time integrated. Each CP is set against a time axis with a defined beginning and end. Proper time driven CPs specify the beginning and end in terms of reference points using proper time units and progress according to particular

time intervals. Milestone driven CPs specify the start and end points using milestone units and progress according to the achievement of particular milestones, depicted as a chain of tasks. Time integrated CPs either set chains of events against proper time intervals or set specific target end points for each milestone in proper time units.

Using the time relationships for the different types of CPs, we defined the classes **TIMEAXIS** and **INTERVAL**, subclasses of **TEMPORALENTITY**. **TIMEAXIS** has defined start and end points and is comprised of a series of intervals. An **INTERVAL** also has specified start and end points, where interval duration is the elapsed time. The subclasses of **INTERVAL** are **PROPERINTERVAL**, **MILESTONEINTERVAL**, and **INSTANT**. A **PROPERINTERVAL** is one where the start and end points are specified in proper time units. For **MILESTONEINTERVAL** its start and end points are specified in milestone units rather than proper time units. The start and end points of an **INSTANT** are the same – i.e. the duration of an instant is zero.

CLINICALPATHWAY captures the CP as a whole, describing the way in which patient care, for a specified cohort, should unfold for a specified episode demarcated by a time axis. Through instances of the classes described, **CLINICALPATHWAY** directly relates to **CLEARINGINFORMATION**, **TARGETPOPULATION**, **GOAL**, **TIMEAXIS**, and **PROCESS**.

To write constraints, we used Protégé's EZPal plugin to specify 10 constraints. For example, using EZPal's template statement "Every instance of class __ must have a unique value in slot __" we wrote the following constraint: "Every instance of **KNOWLEDGESOURCE** must have a unique *KnowledgeSourceCitation*." The specification of constraints was limited by EZPal's template, so we were not able to specify all constraints.

To improve the usability of our ontology, we provided class and slot definitions through Protégé's documentation feature, and annotated it with Unified Medical Language System (UMLS) concepts. Using Protégé's plugin for UMLS, we were able to annotate our CP Ontology with UMLS concept IDs.

4. CP Ontology Evaluation

Overall the CP Ontology performed well in the instantiation of five diverse CPs. We noted six ontology deletions, four substitutions and five potential insertion errors.

Three of the deletion errors related to concepts outside the scope of our ontology: (1) representation of reference information for educational and decision support uses; (2) documentation for the patient record; and, (3) representation of institution-specific policies and constraints. These would be best addressed by linking our CP Ontology to a reference knowledge base, a patient record, and an institutional resource database, respectively. Other deletions related to the task of obtaining patient consent, task discontinuation activities, and delivering test results to the patient. These may be addressed by refining the **TASK** class and its slots.

The most significant substitution was the specification of task duration and its impact on the finish to start relationships with succeeding activities. The ontology worked well for: (a) tasks that represented one-time occurrences; and, (b) proper time driven CPs where the interval is specified for each task. However, some CPs do not entirely fit this pattern, thus demanding minor modifications to our CP Ontology.

We identified five potential insertion errors: **AUDIT**, **VARIANCE**, **VARIANCETYPES**, **DISEASECLASSIFICATIONCODE**, and **INSTANT**. Each of these classes was superfluous

with respect to the instantiation of particular CP documents. These classes are meant to be populated when CPs are executed in practice.

5. Concluding Remarks

We described our knowledge management approach toward capturing the practice-oriented knowledge in CPs through: (i) knowledge source identification and classification; (ii) knowledge abstraction; (iii) ontology engineering; and, (iv) ontology evaluation. Our CP Ontology identifies salient concepts as classes, and attributes as slots. We also identified relationships between classes and specified some pertinent constraints. We demonstrated the functionality of our CP Ontology by instantiating a sample of diverse CPs that span medical and surgical diagnoses; adult and pediatric populations; and, hospital and ambulatory settings. We posit that our CP Ontology is novel, in that it is the only ontological model to date that describes both the structure and function of CPs in their entirety.

Abstracting and articulating the knowledge in CPs has led to a deeper and more formal understanding of the characteristics of CP. Translating our ontological framework into practice will help standardize CP development across healthcare institutions and facilitate knowledge sharing through linkage and exchange amongst CP developers. Providing a common reference framework for CPs will also facilitate multi-professional communication and collaborative development on an institutional level.

Acknowledgement: The authors wish to acknowledge the research grant by Agfa Healthcare to support this project.

6. References

- [1] L. de Bleser, R. Depreitere, K. de Waele, K. Vanhaecht, J. Vlayen and W. Sermeus, "Defining pathways," *J Nurs Manag*, vol. 14, pp. 553-63, 2006.
- [2] S. R. R. Abidi and H. Chen. Adaptable personalized care planning via a semantic web framework. 20th International Congress of the European Federation for Medical Informatics (MIE 2006), Maastricht, August 27-30, 2006. IOS Press.
- [3] S. Chu, "Reconceptualising CP system design," *Collegian*, vol. 8, pp. 33-36, Jan. 2001.
- [4] R. Page and I. Herbert, "Developing e-pathway standards," in *E-Pathways: Computers and the Patient's Journey through Care* K. de Luc and J. Todd, Eds. Oxon, United Kingdom: Radcliffe Medical Press, 2003, pp. 155-173.
- [5] K. de Luc and J. Todd, "A way forward?" in *E-Pathways: Computers and the Patient's Journey through Care* K. de Luc and J. Todd, Eds. Oxon, United Kingdom: Radcliffe Medical Press, 2003, pp. 183-198.
- [6] J. Angele, D. Fensel and R. Studer, "Domain and task modeling in MIKE," in *Domain Knowledge for Interactive System Design: Proceedings of IFIP WG 8.1/13.2 Joint Working Conference*, 1996, pp. 149-163.
- [7] A. L. Strauss, *Basics of Qualitative Research : Techniques and Procedures for Developing Grounded Theory*, 2nd ed. Thousand Oaks: Sage Publications, 1998, pp. 312.
- [8] M. Uschold and M. Gruninger, "Ontologies: Principles, Methods and Applications," *Knowl. Eng. Rev.*, vol. 11, pp. 93-155, 1996.
- [9] F. N. Noy and D. L. Guinness. (2001, March). *Ontology development 101: A guide to create your first ontology*. Stanford KSL Technical Report KSL-01-05, 2000
- [10] Stanford University Department of Medical Informatics. (2006, December). *Protégé. 3.2* Available: <http://protege.stanford.edu/index.html>
- [11] R. Porzel and R. Malaka. *A task-based approach for ontology evaluation*. ECAI Workshop on Ontology Learning and Population, Valencia, Spain, 2004.
- [12] NHS Orkney, "Acute Coronary Syndrome Pathway," Version 46, pp. 1, May. 2003.