MEDINFO 2010
C. Safran et al. (Eds.)
IOS Press, 2010

doi:10.3233/978-1-60750-588-4-1104

Ontology Based Modeling and Execution of Nursing Care Plans and Practice Guidelines

Muzammil Abdulrehman Din, Syed Sibte Raza Abidi, Borna Jafarpour

NICHE group, Faculty of Computer Science, Dalhousie University, Halifax, NS, Canada

Abstract

Nursing Care Plans (NCP) and Nursing Clinical Practice Guidelines (NCPG) promote evidence-based patient care, but in their paper form they are difficult to be applied at the point-of-care. We present our approach to generate patient-specific nursing care plans by modeling and computerizing these nursing knowledge resources. We present a Nursing CarePlan Ontology (NCO) that models the NCP and NCPG to realize an integrated knowledge base for designing and executing patient-specific nursing CarePlans. We adapted the METHONTOLOGY methodology for ontology engineering to develop our OWL-based NCO, and instantiated a set of NCP and NCPG. We have developed an execution engine that provides recommendations to nurses based on the patient's data. NCO was successfully evaluated for representational accuracy and completeness using a set of test NCP and NCPG.

Keywords:

Nursing care plan, Nursing clinical guidelines, Ontology, Care planning, Clinical decision support systems.

Introduction

Nurses represent the largest group of health care professionals that are directly involved in patient care in hospitals. The quality of care to hospital patients is strongly linked to the performance of nursing staff [1]. Hence, nurses need to be informed about evidence-based methods as well as being skillful in applying these methods, in a timely manner, to achieve the desired patient care outcomes [2].

Nurses organize the care process in terms of a Nursing Care Plan (NCP) that are concise, structured written 'plans of action' targeted at providing patient-specific care. A patient-specific NCP is designed guided by the disease diagnosis (performed by a physician) and a critical assessment of the patient care needs in terms of his/her condition—the resulting NCP comprises a summarized plan of action and tools to monitor the care activities undertaken by nurses and to record the patient's progress [3]. To execute the care activities stipulated in the NCP, nurses need to have knowledge of numerous medical procedures and how to interpret patient's physiological parameters. Nursing Clinical Practice Guidelines (NCPG) provide evidence-based instructions/recommendations about how to handle specific patient care issues related to nursing.

For quality patient care, we argue that a patient specific CarePlan [4] needs to be both customized to the patient's care needs and standardized in terms of best evidence. This means that to design a patient-specific CarePlan the following activities are needed: (a) the available NCP need to be customized as per the patient care requirements; and (b) the care activities within the customized CarePlan need to be supplemented with corresponding NCPG to ensure quality and standardization. Given the work pressures on nurses, the reality is that (i) the manual design, modification and maintenance of the patient's NCP is quite difficult, especially in response to the changing dynamics of the patient; (ii) the manual referencing of NCPG, which are not readily available, is not practical; and (iii) the execution of a paper-based care plan, in a timely and coordinated manner, is challenging. In this regard, we argue that to improve the quality and standardization of patient care there is a need to provide nurses with computerized nursing care planning and management systems to support their care roles.

In this paper, we present our research covering the computerization of NCP and NCPG in order to design and execute patient-specific nursing CarePlans. We take a semantic web approach, whereby we model the NCP and NCPG in terms of an OWL-based ontology. We modeled both the form and function of NCP and NCPG in terms of domain and workflow concepts, establishing semantic interrelationships between the concepts, and instantiating select NCP and NCPG using our Nursing CarePlan Ontology (NCO).

Nursing Care Planning: Concept & Solution

Individualized care has been described as ".. the management of care of the patient on the basis of his unique needs, the objective being maximum independence from the necessity of such care" [5]. This demands that the nursing care plan is customized to the specific needs of a patient and is able to dynamically adapt as the patient needs change. Although, nursing care is guided by the medical diagnoses, nurses work on the basis of a subsequent nursing diagnosis that deals with the nursing interventions required to treat the patient. At a conceptual level, the nursing care plans are categorized on the basis of the medical diagnosis and illustrate a set of care activities pertinent to the patient care for that specific disease (as shown in Figure 1). At the operational level, the patient-specific CarePlan is based on nursing diagnosis that determines a selection of care processes relevant to the patient—a single medical diagnosis can lead to multiple nursing diagnoses. Therefore, a patient' CarePlan transcends across multiple nursing care plans and is a composite of individual patient-specific tasks originating from multiple disease-specific nursing care plans.

Here, two patients with the same medical diagnosis can have different nursing care plans (shown in Figure 2).

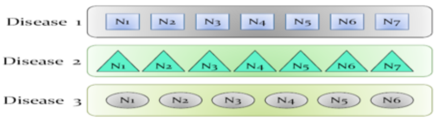

Figure 1–Nursing care plans for different medical diagnosis

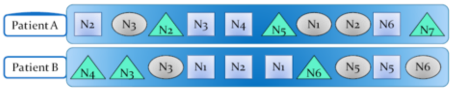

Figure 2–Patient-specific CarePlan comprising tasks, in a particular order, selected from the disease-specific NCP based on the nursing diagnosis to handle co-morbidities.

To model NCP for the generation of CarePlans, we argue that it is important to work with modeling formalisms that allow the decomposition of larger concepts into constituent components, and then inter-relate the relevant components to realize a customized solution. Semantic web ontologies provide such a knowledge modeling formalism, whereas Task Network Models (TNM) organize knowledge components to realize a composite executable process workflow [6]. A number of existing formalisms to model clinical practice guidelines, such Asbru, EON, GLIF, PRO*forma,* and SAGE use ontologies and some form of TNM. Therefore, since NCP and NCPG outline complex actions, we plan to model them as ontological TNM as it will allow us semantic descriptions of concepts and reusability of the components to generate a range of CarePlans.

Modeling Nursing Care Process

In line with our modeling decisions of using ontologies and TNM, as the first step we developed a hierarchal representation scheme that takes into account the hierarchical knowledge and workflow classifications observed in NCP and NCPG. We examined NCP and NCPG to determine the different levels of care process classifications. We observed that the terms 'Procedure' and 'Activity' consistently appeared in NCP and NCPG. We used UMLS to disambiguate their meaning and functional purpose in the care process, and determined a three-level care process classification as follows: *Procedures* → *Activities* → *Tasks*. Each process can have *n* number of procedures (P_n), while a procedure itself can have *n* number of activities (A_n) and finally an activity can have *n* number of tasks (T_n) as shown in Figure 3. It may be noted that processes in the NCP/NCPG are typically recommendation/intervention. Thus, in our nursing care model, NCPG based Recommendations and NCP based Interventions are represented as procedures, activities, and tasks. A CarePlan is a composite of recommendations/interventions employing individually modeled procedures, activities and tasks in a patient-specific workflow.

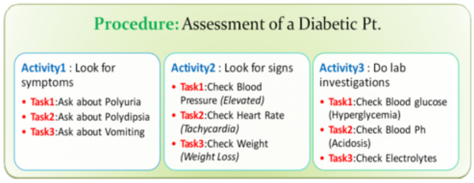

Figure 3 – Hierarchical decomposition of NCP and NCPG

Nursing CarePlan Ontology Engineering

To develop NCO we designed a three-stage methodology, an adaptation of METHONTOLOGY [7], as discussed below:

Stage 1 - Ontology Specification and Knowledge Identification: The first stage involved the specification of ontology parameters such as domain, purpose, scope, and identification of knowledge sources. A large set of NCP and NCPG were collected and used to design the NCO.

Stage 2 - Ontology Modeling: The second stage involved ontology conceptualization in which we abstracted concepts from a sample of NCP and NCPG, described concept hierarchies, defined relationships, and finally axioms. We then constructed our primitive Ontological model using Protégé-OWL. We used a cyclic inductive approach for concept abstraction, where in each cycle we refined our model based on concepts from the sample NCP and NCPG. Model refinement was concluded when we achieved a concept saturation point whereby no further modifications were required to the NCO to model additional concepts abstracted from new NCP/NCPG.

Stage 3 - Ontology Evaluation: The final stage involved evaluating our ontology in three steps. First, we evaluated our model for representational accuracy by encoding five randomly selected NCP and NCPG. Secondly, we evaluated our model with the guideline modeling dimensions described by Peleg [8]. Finally, we evaluated our ontology against the standard ontological design principles [9, 10]. The resultant NCO was found to be consistent and complete.

Ontological Representation of NCP and NCPG

To describe our NCO, class names are given small caps e.g. Disease, properties using italics e.g. *isFollowedBy*, while Individuals are written within quotes e.g. "Tuberculosis".

NCO depicts NCP (NursingCarePlan) and NCPGs (NursingClinicalPracticeGuideline) as a set of Interventions (NursingIntervention) and Recommendations (NursingGuidelineRecommendation) respectively, and in order to activate them, a predefined InclusionCriteria has to be satisfied. The inclusion criteria can be a particular AgeGroup, Gender, Symptom, Sign, or a particular Diagnosis. An ExclusionCriteria can also be defined which excludes a NCP or NCPG from activation. The top-level recommendation/intervention is modeled as NursingGuidelineRecommendation, and it has sub-classes Procedure, Activity and Task, where each has Preconditions and ExpectedOutcomes. This is modeled as NursingGuidelineRecommendation *hasPrecondition*

Precondition and *hasExpectedOutcome* ExpectedOutcome. To model sequences among procedures, activities and tasks we defined properties *isFollowedBy* and *isPreceededBy*. To describe the status of tasks we modeled four discrete Task States by defining a class State with individuals “Inactive”, “Active”, “Completed”, and “Failure” (also shown in Figure 4). Next, we describe how we use NCO to model the workflow of NCPG.

Modeling Workflow of a NCPG

Outcomes play an important role in our model, as they not only define the result of a task but also link different tasks together. We modeled the interconnection between tasks by relating the ExpectedOutcomes of one or more tasks as Precondtions for other tasks (see Figure 4). In this regard, our model operates as a workflow, where states have binary outcomes such as ‘Yes’ or ‘No’ depending on their a priori desired outcomes. Depending upon the outcome of a task the workflow takes a specific path. The advantage of this approach is that it is intuitive to model using an ontology as this does not demand complex rules. Furthermore, the interlinking of tasks in this manner allows us to control the workflow for tasks that entail multiple choices and need the satisfaction of multiple constraints before proceeding to the next task.

Modeling NCPG Execution Criterion

The first step in executing the CarePlan is the satisfaction of the InclusionCriteria of the NCP/NCPG—the said NCP/NCPG is considered active and our execution engine searches for the satisfied Preconditions of their component tasks. The component tasks are by default in an “Inactive” State even after the activation of the NCP or NCPG. Once the Preconditions for any task are satisfied the State of that task changes to “Active”. In Figure 4, the lower left task is “Active” since its Precondition has been satisfied (depicted by the yellow color). Activation of any task causes the execution engine to display that task to the user who then acts upon it. After executing the task, the user is presented with a list of ExpectedOutcomes for that particular task and can choose the one that reflects what actually happened. This user feedback is modeled as the ActualOutcome of that task. When the ActualOutcome entered by the user is one of the predefined ExpectedOutcomes, the state of the task changes to “Completed” (depicted by a blue color, First task in Figure 4). In Figure 4, for the first task there are only two ExpectedOutcomes defined, the number “1” or “2”, the ActualOutcome in this case turned out to be the number “1” and this outcome had already been defined as a Precondition of the next task and thus the following task is activated. The newly activated task then goes through the same steps as described for the preceding one to reach completion. In cases where the ActualOutcome is not any one of the predefined ExpectedOutcomes, the state of the task changes to “Failure” (depicted by a red color). Here we imply that the task did undergo completion but was deemed a “Failure” since the ActualOutcome was simply not what was intended and such an outcome is modeled as an UnexpectedOutcome. An unexpected outcome could be because of an adverse event or reaction. We intend to use unexpected outcomes to gain insight into that particular clinical scenario. They are saved for variance analysis and provide feedback to the authors of the original NCP and NCPG.

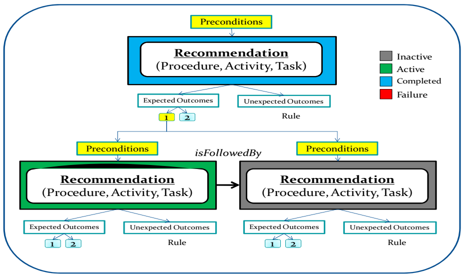

Figure 4 - The interrelationships between tasks and their states are shown using the color code

Modeling NCPG Execution Control Rules

To model the execution of a NCP we designed a set of ontology-based rules that work in tandem with the NCO. The rules are written using the Semantic Web Rule Language (SWRL). Below we give some rules in natural language.

Rules governing Preconditions

- A Recommendation/Intervention may or may not have Preconditions but their ‘Expected Outcomes’ should be defined whenever possible.
- IF two or more Preconditions are defined, THEN a ‘Satisfaction Criteria’ (All, Any One, Any Two, Major, Minor) has to be declared for activation.

Rules governing State Change

- Interventions are all initially in an ‘Inactive’ State.
- After activation, all Tasks will terminate with an outcome. This will be provided by the user by selecting from a list of predefined ExpectedOutcomes or manual input and will be declared as that Task's ActualOutcome.
- The “Inactive” state changes to “Active” only if Preconditions & satisfaction criteria are satisfied i.e. Precondition (Expected Outcome of a Task) = ActualOutcome of any 'Completed' Task and/or Precondition = ‘User Input’.
- IF the ActualOutcome of a Task is equal to any of its ExpectedOutcomes THEN set State to “Completed”.
- IF the ActualOutcome of a Task is not equal to any of the predefined ExpectedOutcomes THEN set State of that Task to “Failure”. “Failure” implies that the task has completed but failed to achieve the expected outcome.
- IF ExpectedOutcomes have not been defined for a task THEN set the State to be “Completed”.

Rules for competing Recommendations/Interventions

In situations where multiple interventions are available and only one intervention has to be selected but the selection criteria is not clearly defined, then interventions will be prioritized based on the number or type of satisfied preconditions—i.e.

by employing major and minor criteria. For example if three interventions are competing then the one with the most preconditions satisfied will be activated.

Rules governing 'Task Reuse'

- IF two different ACTIVITIES want to reuse the same task THEN the PRECONDITIONS and Sequence cannot be directly related to one TASK. The TASK should then have a PRECONDITION defined that belongs to the class SCENARIO.
- Individuals belonging to the subclasses of SCENARIO will serve to define uniquely the PRECONDITIONS and Sequence for that TASK for a particular scenario.

From NCO to Patient-Specific CarePlans

The modeling of NCP as a set of atomic procedures, activities and tasks, allows these constituent elements to be systematically selected and organized, as a task network, to design a patient-specific CarePlan. A CarePlan comprises a linked graph of procedures/activities/tasks from different NCP (as shown in Figure 5), where the individual properties of its components and its execution workflow (i.e. pre-conditions and outcomes) is determined based on the patient parameters.

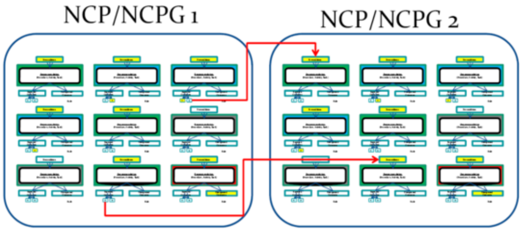

Figure 5 – A CarePlan comprising connected atomic components spanning multiple NCP/NCPG

NCO-Driven Execution Engine

The execution engine assists nurses in coordinating the care process as follows: (a) Recommending the sequence of care steps that need to be performed as per the NCPG; (b) Showing all the currently active steps; (c) allowing nurses to record the patient's parameters—note that a patient's condition determines the next steps; (d) allowing nurses to record the completion of a step by specifying its outcome or by specifying their decision; (e) the completion of previous steps and/or the availability of the observed outcomes triggers the recommendation of the next appropriate step. In this manner, the execution engine enacts the NCPG to guide nurses through the care process. At any stage, the execution engine can inform nurses the completed steps, past outcomes, the current active steps and their expected outcomes and the prospective next care steps.

Our execution engine executes NCP/NCPG, modeled using NCO, based on patient data and nurse input. The domain and execution concepts modeled in the NCO (described earlier) were extended as per the execution logic to achieve the desired outcomes, for instance the NCO describes four possible states of a task, however to control the execution as per the intended semantics of the NCP/NCPG we added three additional states. We use JENA reasoning API to access and manipulate RDF statements defined in the instantiated NCO.

Our execution strategy is to consider each NCPG step (Procedure, Activity or Task) as a Deterministic Finite State Machine (DFSM). The entire NCPG is therefore a directed graph of DFSMs. A DFSM is a quintuple (Σ,S,s_0,δ,F). Σ is the input (finite and non-empty), S is set of possible states, s_0 is the start state, δ is the state transition function: $\delta{:}S\times\Sigma\rightarrow S$ and F is the set of final states. Below is a brief description of the transitions needed for executing NCPG (shown in Figure 6a).

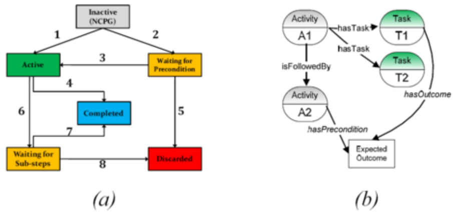

Figure 6 – (a) The DFSM assigned to each step, its states and possible state transitions (b) A small instantiated ontology

- **1** and **2:** An active step can activate its sub-steps and a completed step can activate the steps following it**.** For example when a *procedure* is executed, its first *activity* (or *task*) is added to the active list. If the step that is going to be added to active list is waiting for a precondition to be satisfied, it will go the "Waiting_for_Preconditions" state instead of "Active" state.
- **3:** If the precondition of a step is satisfied (either because of the outcome of other steps or user input) the task's state changes to "Active" state.
- **4:** If the step is a leaf in the hierarchy (it has no further sub-steps) and it is selected for execution, then subsequently it will be moved to "Completed" state.
- **5:** If all the steps that produce outcomes necessary for satisfaction of the preconditions of a task undergo completion and the precondition remains unsatisfied, the waiting step moves to "Discarded" state.
- **6:** If a non-task step (a procedure/activity) gets selected for execution, it will move to "Waiting_for_Sub-steps" state.
- **7** and **8:** If a step is waiting for completion of its sub-steps, then as they are completed it will be regarded a completed task. If any of its sub-steps were discarded instead of being completed, it will move to "Discarded" state as well.

To execute a NCPG we first build its RDF graph, where nodes are instances of classes and edges are the properties of the NCPG. This RDF graph captures the information flow between DFSMs as per the NCPG workflow. During execution, the Σ for each DFSM (step) is the state and the expected outcome of other incoming steps. For instance, in Fig 6b the Σ for A2 is the state of A1 and expected outcome of T1 and Σ for

A1 is the states of T1 and T2. In figure 6b, we show a segment of a NCPG with multiple steps and explain an execution scenario through DFSM. Suppose that the states of the steps are A1 (Waiting_for_sub-steps), A2 (Inactive), T1 (Completed) and T2 (Active). Now, when the engine executes T2, its DFSM will move to "Completed" state. This state transition will cause A1 to change its state from "Waiting_for_sub-steps" to "Complete". State change of A1 will trigger DFSM of A2 to change its state from "Inactive" to "Active". In this manner, execution proceeds till all DFSMs reach their stable state and no further state transition is possible. Our execution engine is able to handle multiple active steps, whereby we have implemented a timesharing breadth first search strategy that allows all active tasks to progress in parallel.

Evaluation Results

We evaluated our ontological model in three phases. We first tested our model for representational accuracy by encoding a sample of test guidelines and care plans and documenting any instantiation problems. Secondly, we tested our model against the guideline modeling dimensions described by Peleg et al [8]. Finally, we tested our ontology against the standard ontological design principles [9, 10].In terms of representational accuracy, NCO was found to miss knowledge about various drug doses and titrations. We plan to link NCO with a drug ontology to address this issue. Otherwise, NCO successfully represented the concepts found within the test NCP and NCPG. We evaluated NCO using the 8 CPG representational dimensions, proposed by Peleg et al [8], that cover structural and linkage aspects of a CPG model. We instantiated the original test guidelines used by Peleg et al using our NCO, and observed that NCO was able to aptly represent these guidelines. Finally to assess compliance with the design principles mentioned by Gomez-Perez [9] we manually went through the class hierarchy, properties and individuals in NCO. Additionally, we checked NCO to be compliant with the ontological principles described by Bodenreider for the medical domain [10]. Hence, NCO was deemed to be fully compliant with these principles.

Conclusion

We have encoded 6 NCPG using the NCO and they are now available for execution through our execution engine.

The computerization of NCP and NCPG is the first step towards their incorporation in the care process and their availability at the point of care. We presented an ontology-based modeling and execution framework for the computerization of NCP and NCPG, leading to the generation of patient-specific CarePlans. The key feature of our approach is that we were able to decompose large monolithic NCPG and NCP into small-scale, independent atomic components that can subsequently be used (and re-used) to design patient-specific CarePlans that cover clinical workflow, nursing knowledge and care coordination. The linkages between the care processes given within NCP with corresponding evidence within NCPG provides an integrated care environment where knowledge is impacting the care process—such linkages are desired by nurses but are neither practical in a paper-based setting nor implemented in current nursing systems. At the execution level, we have demonstrated the execution of complex NCP/NCPG through a rather simple graph traversal approach that is supported by lightweight SWRL rule based reasoning. At the next stage, we will be building data transfer interfaces to both collect and store data from the patient's EMR. Here we are planning a semantic web services based architecture built using HL-7 standards—the use of a OWL-S service ontology will allow for improved data interoperability between different care services. Although, our execution engine is capable of handling concurrently active care processes (i.e. execute processes in parallel), we lack the ability to reason with time and duration. Our future plan is to incorporate the OWL-Time ontology, developed by W3C, to handle the temporal aspects of NCP/NCPG execution.

References

[1] Hassmiller SB, Cozine M . Addressing The Nurse Shortage To Improve The Quality Of Patient Care. Health Aff 2006; 25:268.

[2] Fitzpatrick JM, While AE, Roberts JD. The role of the nurse in high-quality patient care: a review of the literature. J Adv Nurs 1992 10;17(10):1210-9.

[3] Mason C. Guide to practice or 'load of rubbish'? The influence of care plans on nursing practice in five clinical areas in Northern Ireland. J Adv Nurs 1999;29(2):380.

[4] Abidi SSR, Chen H. Adaptable Personalized Care Planning via a Semantic Web Framework. 20th Intl. Congress of the European Federation for Medical Informatics (MIE'2006); 27-30 August; : IOS Press; 2006

[5] Jacox AK, Carr DB. Nursing & technology. Clinical practice guidelines. Nurs Econ 1991 Mar-Apr;9(2):118-20.

[6] Nejati N, Langley P, Konik T. Learning hierarchical task networks by observation. Proceedings of the 23rd international conference on Machine learning: ACM Press, 2006.

[7] Fernández M, Gómez-Pérez A, Juristo N. METHONTOLOGY: From ontological art towards ontological engineering. AAAI97 Spring Symposium, 1997.

[8] Peleg M, Tu S, Bury J, Ciccarese P, Fox et al. Comparing Computer-interpretable Guideline Models: A Case-study Approach. JAMIA 2003;10(1):52.

[9] López MF, Gómez-Pérez A, Sierra JP, Sierra AP. Building a chemical ontology using Methontology and ontology design environment. IEEE Intelligent Systems 1999:37-46.

[10] Bodenreider O, Smith B, Kumar A, Burgun A. Investigating subsumption in SNOMED CT: An exploration into large description logic-based biomedical terminologies. Artif Intell Med 2007;39(3):183.

Address for correspondence

Syed SR Abidi. Email: sraza@cs.dal.ca
Muzammil A Din. Email: muzammil@dal.ca