Semantic Web-based modeling of Clinical Pathways using the UML Activity Diagrams and OWL-S

Ali Daniyal, Syed Sibte Raza Abidi

NICHE Research Group, Faculty of Computer Science, Dalhousie University, Canada
{daniyal, sraza}@cs.dal.ca

Abstract: Clinical Pathways can be viewed as workflows, comprising an ordering of activities with associated execution constraints. Workflow models allow formal representation, analysis and execution of workflows in the Clinical Pathways. We present a semantic web-based approach where the domain knowledge and the workflow model are modeled separately as ontologies, while the Clinical Pathway and the associated workflows are modeled as the instantiations of these ontologies. Our workflow model is based on the UML Activity Diagrams and OWL-S service ontology, and the execution semantics are based on Place/Transition Petri Nets. We demonstrate our approach by capturing the workflow of the Prostate Cancer Care Pathway.

1. Introduction

Clinical Pathways (CP) aim to coordinate the care process for a specific condition at the institution level. In essence, CP describe the functional knowledge pertaining to an institution's clinical practices in terms of time-sensitive and outcome-driven processes—represented as a combination of plans, tasks, decisions, resources and care providers—that essentially resemble a *workflow*. Execution of CP for clinical decision support is a complex activity and demands (a) knowledge modeling—i.e. modeling the domain knowledge and the CP's functional knowledge that describes workflows involving multiple resources and actors; and (b) definition of execution semantics. Researchers have argued that CP knowledge modeling requires a formal Knowledge Representation and Reasoning (KRR) framework, and an execution model is required for describing the execution semantics [13].

The Semantic Web (SW) framework offers interesting methods to execute CP as it provides (a) semantically-rich knowledge modeling and representation formalism in terms of ontologies; (b) reusability of the knowledge models, (c) neat separation between domain and functional concepts, yet their easy integration to describe the CP knowledge; and (d) reasoning mechanisms to execute the CP knowledge represented in ontologies. To execute CP, we propose a synthesis of SW and workflow modeling techniques—SW based ontologies capture the domain specific aspects of a CP, whereas workflow modeling techniques such as UML activity diagrams allow the translation of the procedural aspects of CP into formal workflow models that characterize an ordering of clinical tasks and their associated executional constraints. Furthermore, we argue that our approach can potentially lead to the integration of CP with Clinical Information Systems (CIS), and as such we explore web-services technologies since they offer standards for interrelating heterogeneous applications.

In this paper we present our SW-based approach for the representation, analysis and execution of CP workflows. In our framework, medical domain knowledge is modeled as *domain ontologies* and workflow knowledge as a *workflow model ontology*—CP workflows are modeled as instantiations of the domain and workflow model ontologies. We have developed an OWL-based CP workflow model based on UML Activity Diagrams and OWL-S process model (represented as a service ontology) as both these approaches have well defined Place/Transition Petri Nets (PTN) based execution semantics [2,3]. We argue that the translation of CP workflow descriptions to PTN allows us to define execution semantics for our workflow model thus enabling us to (a) execute the CP workflows; and (b) analyze the modeled CP workflows for various correctness issues such as deadlock, reachability, liveness, safeness and boundedness. In summary, we are developing a single framework for (i) ontology-based modeling of CP workflows employing ontology-encoded domain knowledge, (ii) analyzing and executing the workflows, and (iii) potential integration of CP workflows with CIS operations. We demonstrate our CP workflow execution framework by modeling and executing workflows for Prostate Cancer Care Pathways where the domain knowledge is encoded in an ontology developed by Abidi et al. [5].

2. Related Work

A number of approaches exist to address the knowledge modeling and execution modeling needs for computerizing CP. These approaches can be classified as: (i) approaches focusing on the structural and functional modeling of the CP knowledge, (ii) approaches focusing on the execution models and analyses of the workflows, and (iii) approaches focusing on the integration of workflows with existing CIS.

Peleg et al. [8], Ye et al. [10] and Tu et al. [6] employed ontology models for capturing the domain knowledge. Peleg et al. [8] use the BioWf model for representing the structural domain knowledge by employing Protege-2000 KRR framework. Protege-2000 is also employed by Tu et al. [6] for domain knowledge modeling, while Ye et al. [10] employs OWL for describing the domain ontology. Dominguez et al. [9] developed a Life Assistance Protocol (LAP) model for capturing medical and workflow knowledge.

The workflow model employed by Peleg et al. [8] is based on the workflow model of the Workflow Management Coalition (WfMC) while Petri Nets (PN) are used as the execution model, through which they were able to perform different types of analyses and answer questions related to workflows. Ye et al. [10] employed OWL-S for modeling workflows, while the workflow management is achieved through OWL-S and SWRL-Rule-based modeling approach to temporal relationships. Dominguez et al. [9] employ Timed Parallel finite Automata for modeling workflows while the execution model is based a Multi-Agents Systems approach.

Tu et al [6] focus on the modeling of workflows in CP that allows integration of multiple data sources and the CIS operations. Anyanwu et al. [7] propose the METEOR system that employs WPDL of WfMC to describe the workflows and develops standards for interoperation of disparate sites. Tu et al. [6] discuss the SAGE system that is based on the previous work on guideline modeling including Proforma, GLIF, Asbr, EON, GEM, GLIF3, GUIDE and PRODIGY. A comparison of these systems in terms of their expressivity and features can be found in [13].

PN have been studied extensively to capture the execution logic of CP and biological processes. PN enable different types of analyses on the workflows and different PN tools facilitate study of the workflows. Peleg et al. [11] study properties and dynamics of biological systems and care pathways using different PN tools. [8,11,12] have based their execution models on PN. Du et al. [12] propose a framework of CP Adaptive Workflow Modeling based on Extended Workflow Nets, which is an extension of PN.

3. Our Solution Approach

In our work, we build on the above research with the approach of separating the domain knowledge with the functional knowledge, but allowing integration of the knowledge sources to execute specific CP knowledge within specific institutional settings. Furthermore, we extend the research to the exploration of the integration of the CP workflow within existing CIS through a services oriented approach that employs SW services standard—i.e. OWL-S. To execute and analyze CP, our solution approach is to (a) use SW-based methods for modeling CP workflow; (b) use the UML Activity Diagrams and the OWL-S process model to describe CP workflows; (c) use the OWL-S grounding to integrate CP workflows with CIS; and (d) use Petri Nets execution model to both execute and analyze CP workflows. At this stage, we have not attempted to connect to a working CIS to execute a modelled CP.

The first step to formally represent a CP is to develop a workflow model that captures both the domain and functional aspects of a CP. We have developed a CP workflow model by combining two workflow and process modeling approaches—i.e. the UML Activity Diagrams and OWL-S service ontology—through the import and extension mechanism in SW (as shown in figure 1).

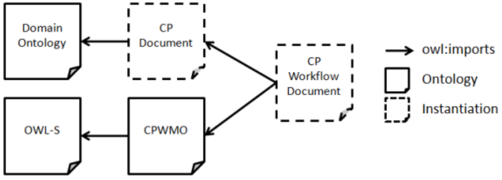

Figure 1: Import hierarchy of documents for modeling workflows in CP.

The UML Activity Diagrams provide an intuitive and expressive method to capture different workflow patterns for a variety of domains [1] with the provision of formal execution semantics [2]. We use UML Activity Diagrams (UML 1.0) to model CP because they are able to capture the complex, and at times nested, control-flow amongst multiple clinical tasks through standard constructs that represent complex workflow patterns, and the ordering constraints among the states. Since individual clinical tasks, modeled in the domain ontology, are now instantiated as independent

UML Activity Diagrams constructs, we are able to re-use defined clinical tasks by combining or nesting them with other tasks to realize a functional CP workflow. The feature of using UML Activity Diagrams for modeling CP tasks is that it allows a re-usability of tasks through the separation of the institution-specific functional details from the domain knowledge—a generic clinical task can now be customized to meet local criterion by modulating the functional constraints.

OWL-S is a semantic web services standard that allows the linking of semantically compatible services based on WSDL descriptions. We leverage OWL-S to (i) enable integration of CIS operations as services within CP workflows, and (ii) capture preconditions, effects, inputs and outputs of the tasks in a CP. Our idea is to expose CIS operations as WSDL-based services which can then be integrated within workflow descriptions using OWL-S. It may be noted that, although OWL-S offers a standard process model to capture control-flow among sub-processes of a process, we use UML Activity Diagrams to model the CP process flow for two reasons: (i) The constructs of the OWL-S process model are less expressive for describing a complex control-flow as compared to the UML Activity Diagrams [4], and (ii) the UML Activity Diagrams offer a more intuitive approach to capture the ordering of tasks as direct relationships between tasks, as compared to the imposition of constraints practiced by OWL-S. The use of UML and OWL-S model offer PTN based execution semantics [2,3] allowing us to execute and analyze CP workflows.

To describe a CP workflow we import (a) RDFS/OWL based medical domain ontologies, and (b) CP Workflow Model Ontology (CPWMO) that entails the workflow knowledge. CPWMO is an OWL ontology of the UML Activity Diagrams which imports and extends the OWL-S service ontology. The execution model for CPWMO is based on the combined PTN semantics of the UML Activity Diagrams and OWL-S. We represent the workflow knowledge of a CP as an instantiation of the CP workflow model.

4. Ontology-based modeling of CP workflows using CPWMO

CPWMO has been developed as an OWL ontology for the UML Activity Diagrams importing and extending OWL-S service ontology. CPWMO ontology has classes for each of the UML Activity Diagram construct namely: ACTIVITYDIAGRAM, INITIAL, FINAL, ACTIONSTATE, SUBACTIVITYSTATE, FORK, JOIN, DECISION, MERGE, SENDSIGNAL and RECEIVESIGNAL. The ACTIONSTATE and SUBACTIVITYSTATE classes are described as subclasses of the PROC:PROCESS class, where proc refers to OWL-S Process Model ontology namespace. STATE is defined as the subclass of PROC:PERFORM, the class of the instances of tasks and activities modeled as individuals of ACTIONSTATE and SUBACTIVITYSTATe respectively. A number of properties are used to capture ordering relationships among the UML Activity Diagrams constructs e.g. *hasInitialState*, *hasFinalState*, *hasEdgeTo*, *hasCondition* etc.

4.1 Modeling Actions and Activities in CPWMO

Atomic actions and complex activities are modelled in CPWMO as instances of the class STATE. Consider a domain ontology defining a task *t* in a CP *P* as an individual of a certain concept of domain. To define an atomic action that corresponds to a particular execution of the task *t* in a workflow associated with *P*, *t* is also declared as an instance of the class ACTIONSTATE. A particular execution of the task *t* can then be modelled as an individual *t'* of the class STATE along with the assertion (*t'*,proc:process,*t*). Modeling of tasks corresponding to complex activities that are themselves an ordering of a number of actions is achieved by employing the class SUBACTIVITYSTATE and a particular execution of such an activity is modelled in a similar fashion. Note that the ordering constraints on the execution of actions in a workflow are modelled as statements about the individuals of the class STATE instead of the individuals of ACTIONSTATE or SUBACTIVITYSTATE. This approach allows reusing one task description in defining multiple workflows involving that same task.

4.2 PTN-based execution model for CPWMO

Our workflow model consists of the UML Activity Diagram constructs and the OWL-S service ontology. We employ the PTN based execution semantics for the analyses and execution of the workflows. In the following we present the PTN based execution semantics for the UML Activity Diagrams and OWL-S process model that forms our execution model.

Place/Transition Petri Net (PTN). A *PTN* is an algebraic structure (P, T, I, O) where; (a) P and T are the sets of Places and Transition respectively, (b) $I:T\rightarrow MultiSet(P)$ is the input function mapping a transition to the multiset of its input places, and (c) $O:T\rightarrow MultiSet(P)$ is the output function mapping a transition to a multiset of its output places. A PTN can be viewed as a directed bipartite graph (V,E) where $V=P\cup T$, $(p,t)\in E$ iff $p\in I(t)$ and $(t,p)\in E$ iff $p\in O(t)$. A *marking* of a PTN N is a function $\mu:P(N)\rightarrow N$ which assigns a non-negative integer of *tokens* to every place in the net. A transition t is *enabled* w.r.t. a marking μ iff $p\in P$, $\#(p,I(t))\leq\mu(p)$, where $\#(p,Q)$ is the number of occurrences of p in multiset Q. Firing of an enabled transition t changes a marking μ to another marking μ' such that $\mu'(p)=\mu(p)-\#(p,I(t))+\#(p,O(t))$.

UML Activity Diagrams to PTN. We employ the PTN based execution semantics for the UML Activity Diagrams discussed by Ivana [2]. Figure 2 illustrates the translation of the UML Activity Diagrams constructs to PTN. For the sake of convenience we use a naming convention in which I, F, A, D, M, F, J, S and R stand for Initial node, Final node, Action State, Decision node, Merge node, Fork node, Join node, Send Signal node and Receive Signal node respectively. In figure 2 the circles in PTN represent places while the rectangles represent the transitions.

OWL-S service ontology to PTN. OWL-S captures three aspects of services, namely: Profile, Process Model and Grounding. The OWL-S process model provides with a number of constructs to define invocation order of a number of services to define a more complex process, called a composite process. We have employed OWL-S process model in our workflow model to capture preconditions, effects, inputs and outputs of tasks in CP (figure 3). We employ the PTN execution semantics of the OWL-S process model presented by Narayana [3] (see figure 4).

For modeling conditions in the UML Activity Diagrams we only consider SWRL conditions, that are conjunctions of RDF statements, in our workflow model. These are used to capture the conditions that decide alternative paths emerging from decision points in CP.

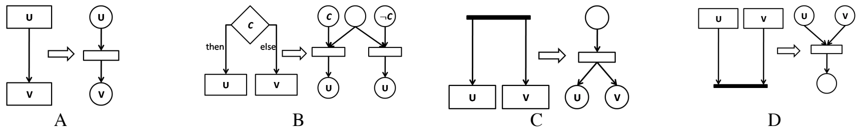

Figure 2: (A) Translation of UML Activity Diagram constructs to PTN. U is a node of type I, A, M, J, S or R while V is a node of type F, A, D, F, S or R. (B) Translation of the decision construct to PTN. C is a condition while ¬ stands for logical negation. U and V are of type F, A, D, F, S or R. (C) Translation of the fork construct to PTN. U and V are of type F, A, D, F, S or R. (D) Translation of the join construct to PTN. U and V are of type I, A, M, J, S or R.

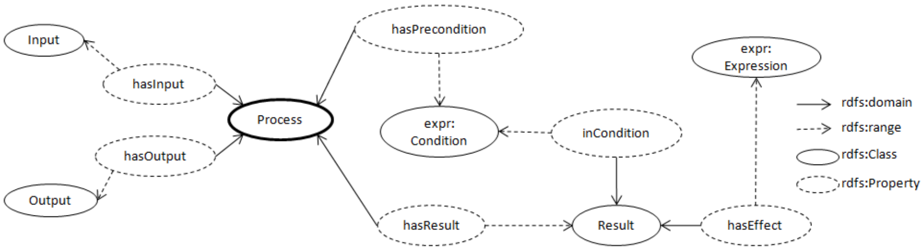

Figure 3: Inputs, Outputs, Preconditions and Effects of services in OWL-S process model.

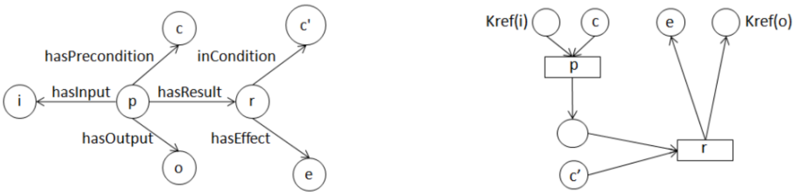

Figure 4: (Left) Description of process p along with its IOPE in OWL-S, (Right) PTN execution semantics of the OWL-S given description

Translation of CPMWO workflows into PTN. Using the PTN based execution semantics of the UML Activity Diagrams and OWL-S service model, we first translate the workflows described using CPWMO into PTN and then perform safety, reachability and deadlock analyses on the workflow. Execution of a workflow is also based on the transition firing mechanism of the corresponding PTN.

5. Modeling workflows in Prostate Cancer Clinical Pathways

Abidi et al. [5] presented the Prostate Cancer Clinical Pathways Ontology and described pathways for three regions—namely Halifax, Winnipeg and Calgary—as instantiations of the ontology. We refer to this ontology as PCONT in this paper. PCONT provides a subsumption hierarchy of 28 classes along with 34 properties to capture the clinical workflows and relevant medical knowledge. PCONT captures fine details about the classification of different types of tasks, decision criteria, treatments and actors. We use PCONT to demonstrate the working of our CP execution framework in terms of the modeling and execution of it.

As a first step, we translated the PCONT workflow information into the classes of CPWMO and captured the execution constraints in terms of the CPWMO ontology. Figure 5 shows a part of the prostate cancer clinical pathway encoded in PCONT, while figure 6 is the RDF graph of the CPWMO encoding of the corresponding workflow.

Figure 5: RDF graph of a part of the PCONT encoded prostate cancer clinical pathway.

The PCONT encoding of the pathway provides information related to the clinical tasks, information states, decisions, temporal information about the tasks and information about the pathway institutions. For example the encoding presented in figure 5 captures information related to tasks EvaluatePatientConditionSeverity and TakePatientConsent, and information states BiopsyIsBookedWithSecondConsultation and BiopsyIsNotBookedWithSecondConsultation.

To model the corresponding workflow information in CPWMO, instances corresponding to the above clinical tasks and information states are created as instances of the STATE class, and these instances are linked to the abstract tasks and states defined in PCONT using the proc:process. This approach allows us to separate descriptions of the tasks (defined in domain ontology) from the description of its invocation in a workflow, thus allowing re-use of the same task description in different workflows.

Decisions concerning the patient condition's severity, patient's location and patient's consent for biopsy are modeled by defining individuals of the class DECISION, while merging of multiple paths into common paths is achieved by defining individuals of type Merge (figure 6). The conditions for decision making are captured as SWRL conditions. (?p hasConditionSeverity Urgent) is the condition for determining the path leading from the EvaluatePatientConditionSeverity task, where ?p is a variable for patient that is bound to a patient in knowledge base at the time of invoking the pathway.

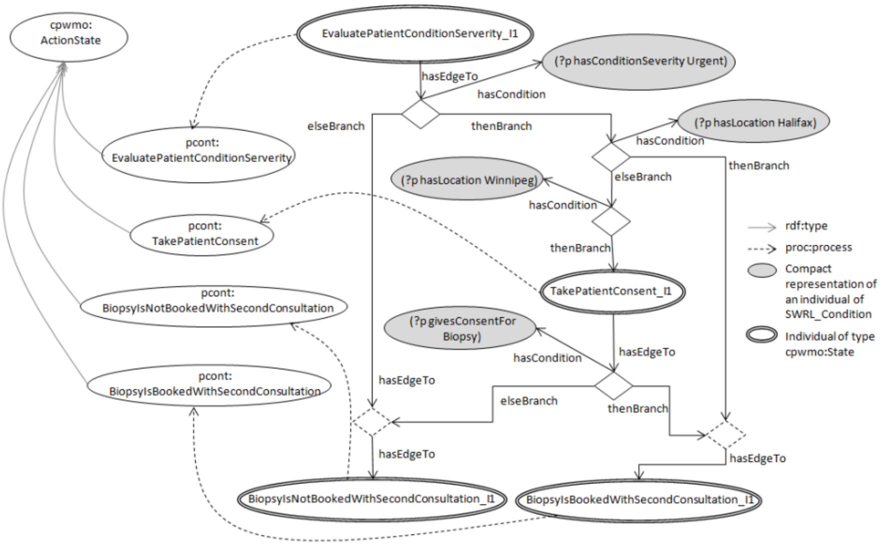

Figure 6: CPWMO encoding of the workflow in the PCONT encoded pathway in figure 5.

In step two, we generated PTN for the CPWMO workflow encodings by using the execution semantics for our workflow model (figure 7). We then performed correctness analyses on the resulting PTN to verify the workflow descriptions. The execution of the workflows is based on the transition firing mechanism of the PTN. For example after the execution of the *EvaluatePatientConditionSeverity_I1*, the transition leading to D_21 is enabled and therefore can be fired. Firing of this transition leads to the invocation of decision D_21. Now based on the patient's condition severity either one of the triples (?p hasConditionSeverity Urgent) or not(?p hasConditionSeverity Urgent) would hold, thus resulting in the invocation of D_22 or M_122 respectively.

We developed a Prostate Cancer Care Planning System that uses the execution model as backend engine. The system updates the knowledge base according to the input provided by the user and queries the execution model for next state. Figure 8 shows snap-shots of our system for the two different executions of the part of workflow defined in figure 7. In the first case (left) the institution is Halifax while the patient lives outside of Halifax, while in the second case the institution is Winnipeg and the execution is at a point where patient's consent for biopsy is required.

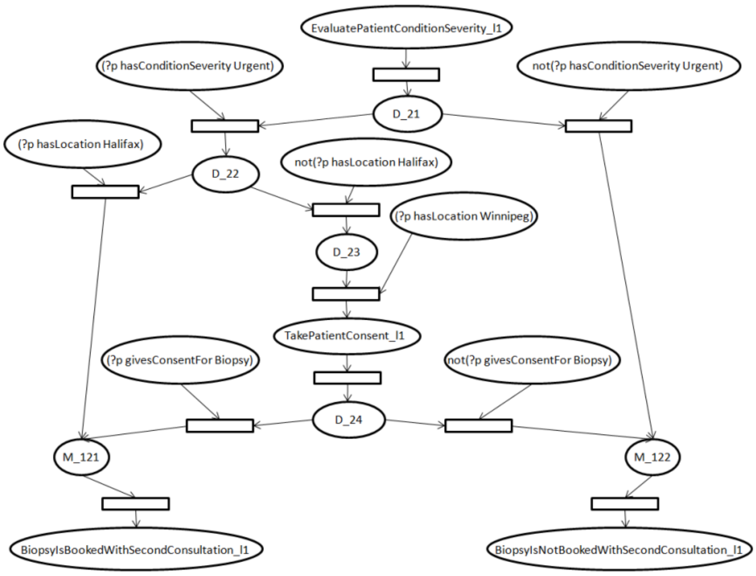

Figure 7: PTN generated by translating the CPWMO-encoded workflow in figure 6.

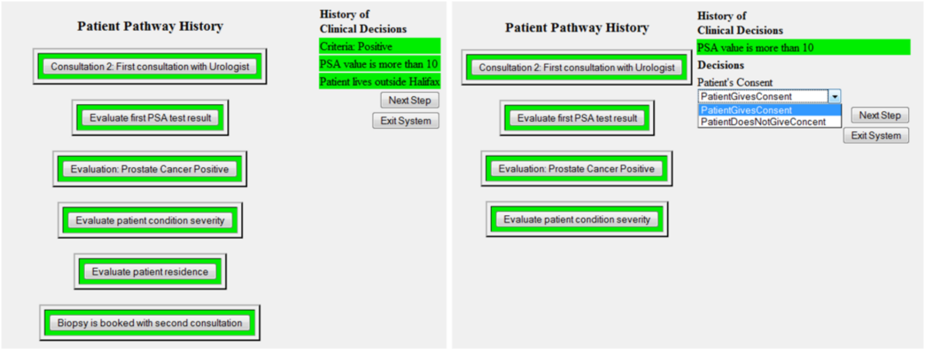

Figure 8: Snap shots of the prostate cancer execution system for different scenarios.

We were able to study correctness of the CP workflow by generating PTN from the workflow descriptions and then execute the CP by developing a system that leverages the execution model. In the next phase we are planning to integrate our system with CIS operations leveraging OWL-S grounding in WSDL.

6. Conclusion and Future Work

We presented a SW-based approach for modeling CP such that (i) the medical domain knowledge is captured as RDFS/OWL ontologies, (ii) the workflow model is described by an OWL ontology, CPWMO, (iii) the workflow knowledge in a CP is described as an instantiation of CPWMO and domain ontologies, and (iv) the execution model is based on the PTN. Our workflow modeling approach allowed us to (i) describe CP workflows by using concepts described in ontology-encoded domain ontologies, (ii) analyze and execute workflows, and (iii) at the same time integrate CP workflows with CIS operations exposed as services using semantic web services standard OWL-S. Using the UML Activity Diagrams for workflow modeling provides an intuitive way of capturing workflow knowledge, while using OWL-S in our workflow model allowed us to address the integration problem with disparate data services and CIS operations. PTN based execution model allowed us to perform different types of analyses on the CP workflows. Studying deadlock, reachability, liveness and safeness properties of the generated PTN allowed us to validate the workflow descriptions.

References

1. Dumas , M., ter Hofstede, A.H.M.: UML Activity Diagrams as Workflow Specification Language. In: 4th International Conference on The Unified Modeling Language, Modeling Languages, Concepts, and Tools, pp. 76--90. Springer, Heidelberg (2001)
2. Ivana, T.: Formalizing activity diagram of UML by Petri nets. Journal of Mathematics, Novi Sad, 30, 161--171 (2000)
3. Narayana, S., McIlraith, S.: Simulation, verification and automated composition of web services. In: 11th Intl. Conf on World Wide Web, pp. 77--88. ACM, Hawaii (2002)
4. Kim, I., Lee, K.: Describing Semantic Web Services: From UML to OWL-S. In: IEEE Intl. Conf. on Web Services, pp. 529--536. IEEE Press, Seoul (2007)
5. Abidi, S., Abidi, S.S.R., Butler, L., Hussain, S. (2009) Operationalizing prostate cancer clinical pathways: An ontological model to computerize, merge and execute institution-specific clinical pathways. In Riano, D. (Ed.) Knowledge Management for Health Care Procedures, ECAI 2008 Workshop K4HelP 2008, LNAI 5626, Springer, Berlin, pp. 1-12.
6. Tu , S., et al..: The SAGE Guideline Model: Achievements and Overview. J. Am. Med. Inform. Assoc., 14, 589--598 (2007)
7. Anyanwu, K., Sheth, A., Cardoso, J., Miller, J., Kochut, K.: Healthcare Enterprise Process Development and Integration. Journal of Research and Practice in Information Technology, Special Issue in Health Knowledge Management, 83--98 (2003)
8. Peleg, M., Tu, S., Manindroo, A., Altman, R.: Modeling and analyzing biomedical processes using workflow/Petri Net models and tools. Studies in health technology and informatics, 107, 74--78 (2004)
9. Dominguez, D., Fernandez, C., Meneu, T., Mocholi, J.B., Serafin, R.: Medical guidelines for the patient: introducing the life assistance protocols. Studies in health technology and informatics, 139, 193--202 (2008)
10. Ye, Y., Jiang, Z., Yang, D., Du, G.: A semantics-based clinical pathway workflow and variance management framework. IEEE Intl. Conf. on Service Operations and Logistics, and Informatics, pp. 758--763. IEEE Press (2008)
11. Peleg, M., Rubin, D., Altman, R.B.: Using Petri Net tools to study properties and dynamics of biological systems. J. Am. Med. Inform. Assoc. 12, 181--99 (2005)
12. Du, G., Jiang, Z., Diao, X.: The integrated modeling and framework of clinical pathway adaptive workflow management system based on Extended Workflow-nets (EWF-nets). IEEE International Conference on Service Operations and Logistics, and Informatics, pp. 914--919. IEEE Press (2008)
13. Peleg, M., et al.: Comparing Computer-interpretable Guideline Models: A Case-study Approach. J. Am. Med. Inform. Assoc. 10, 52--68 (2003)