Using Healthcare Planning Features to Drive Scientific Workflows on the Web

Bruno S. C. M. Vilar, André Santanchè and Claudia Bauzer Medeiros
IC - UNICAMP, 13083-852, Campinas, SP, Brazil

Keywords: Scientific Workflows, Context-Adaptation, Task-Network Model, Healthcare.

Abstract: Automated healthcare planning (care-flow) systems are usually designed to afford the dynamicity of health environments, in which changes occur constantly as a patient's treatment progresses. This dynamic adaptation mechanism is based on blocks of activities, triggered and combined according to contextual data, producing a plan, which emerges from the interaction between these blocks and the context. However, tools that implement care-flow systems are still incipient, missing support for features like extensibility, collaboration and traceability of procedures. On the other hand, these features can be found in workflow systems that are widely used in a variety of environments (in business and scientific domains), with consolidated standards and technologies. However, workflow systems are not well suited to address the dynamicity of healthcare environments. In this paper we argue that care-flow and workflow systems have complementary characteristics and we present a software architecture that incorporates the emergent and context-driven approach of care-flow systems into workflow systems. We present a prototypical implementation validating the key concepts of our proposal, which uses an ontology representation of workflows combined with an ontology and SWRL rules.

1 INTRODUCTION

The growth in the number of patients of a hospital brings the challenge of managing appointments, admissions and surgical interventions. There is the need to increase the efficiency and flexibility to manage associated data. The use of computational resources to address this challenge implies on more technical requirements. The possibility of using data available on the Web has added a new dimension to this problem, with additional heterogeneity factors.

Some researchers (Unertl et al., 2009), in the context of chronic disease care, identified requirements that should be fulfilled by systems to be applied to the health domain, among which we single out: (i) Support for the shared needs and behaviors in care; (ii) Allow customization for disease-specific needs and to support the needs of different types of users; (iii) Explore new approaches for information input into the EHR (Electronic Health Record) as well as transfer, efficiently, data from medical devices into them.

Care-flow is at the core of healthcare management, involving directly or indirectly the requirements presented by (Unertl et al., 2009). As a consequence, one natural approach to start to deal with the problem has been to use Computer-Interpretable Guidelines (CIGs). Informally, a CIG is a specification in some kind of computer interpretable language that defines the flow of steps to be taken in each situation met by a health professional. Those guidelines manage the care-flow, customize needs and details for patient treatment and deal with data gathered, clinical guidelines, while preserves the rationale of healthcare professionals. Though CIGs are suitable for guiding the care-flow, from the perspective of the planning of the paths chosen, the tools that implement a CIG approach are still incipient, missing support to extensibility, reuse and share of content, collaboration among professionals and traceability.

Even though it is possible to define guidelines for "blocks of actions", the whole process involved in the healthcare of a given patient is driven by the context, which is captured during the process itself. In a typical scenario, data from given care-flow step will define the next steps to be followed. The flow of actions *emerge* from the interaction between available blocks and the context. Therefore, one of the most successful approaches for CIG is the Task-Network Model (TNM) (Peleg et al., 2003), which mimics this context-driven evolution. The process starts by a seed block of actions, which is unfolded according to context data collected during its execution.

Vilar B., Santanchè A. and Medeiros C..
Using Healthcare Planning Features to Drive Scientific Workflows on the Web.
DOI: 10.5220/0004844500490056
In *Proceedings of the 10th International Conference on Web Information Systems and Technologies* (WEBIST-2014), pages 49-56
ISBN: 978-989-758-024-6

Scientific workflows, on the other hand, are concerned with the planning and flow of tasks, but associated to scientific experiments and general tasks. They have been developed for years and tested on areas such as Astronomy, Biology, Gravitational Physics and Earthquake Science (Bharathi et al., 2008). As a result, those tools have consolidated mechanisms to deal with large amounts of data, collaboration, extensibility of their resources, and association with external sources of data, tools and algorithms. In general, Scientific Workflow Management Systems (SWfMS) manage experiments' provenance, keeping detailed and meaningful records of data involved on experiments and processes that affect the data. Provenance is fundamental for scientific processes because it provides important documentation that is key to preserve the data, to determine data quality and authorship, and to reproduce as well as validate the results of such processes (Davidson and Freire, 2008).

Workflow execution has been adopted where resources are distributed on the Web, helping the coordination and monitoring of processes. They are being increasingly used to create complex Web applications by Web service composition or by providing a thin client, accessible through a browser, to conduct large scale processes and experiments (Wei Tan, 2013). In order to enhance their applicability, research efforts are concerned with providing workflow mechanisms with more flexibility, including in the healthcare scenario such as (Dang et al., 2008), or (Schick et al., 2012). Even though there are mechanisms that can be used to pre-define exceptions and alternative paths along workflows, they were not designed to dynamically evolve the workflow specification driven by the context, e.g., unfolding blocks of actions according to contextual data collected during a process execution.

Compared to CIG systems, workflow systems have been widely adopted and have consolidated standards and tools. In order to exploit the advantages of workflow systems in the health context, this work proposes to incorporate into workflows the dynamic and context-driven approach followed by care-flow systems. This is based on our experience (Vilar et al., 2013) that shows that the effort to do this is less complex than the process to adapt CIG systems so that they can acquire the features that are causing scientific workflows to become widely adopted. This paper analyses the features that confer high flexibility to CIGs – mainly those based on the TNM due to its emergent behavior – and presents our initial approach to bring them to a SWfMS.

2 CLINICAL PRACTICE GUIDELINES AND TASK-NETWORK MODELS

Clinical Practical Guidelines (CPGs) are written guidelines that describe the evidence-based procedures to be followed during diagnosis, treatment, and clinical decision making for a specific disease. Their textual format can be easily diffused, but not easily used in daily work (Panzarasa and Stefanelli, 2006), because of a multitude of forms and specialized vocabulary. Moreover, they depend heavily on the expertise of health professionals.

An approach to solve these problems is the dissemination of guidelines' content in machine-interpretable representations, which are more suitable for use in individual clinical decision support (Panzarasa and Stefanelli, 2006). This approach has led to the creation of Computer-Interpretable Guidelines (CIGs), which implement guidelines in active computer-based decision support systems. CIGs adopt models to represent the content to support decisions. Some examples of such models are Task-Network Models (TNMs), Medical Logic Modules (MLMs) and Augmented Decision Tables (ADTs).

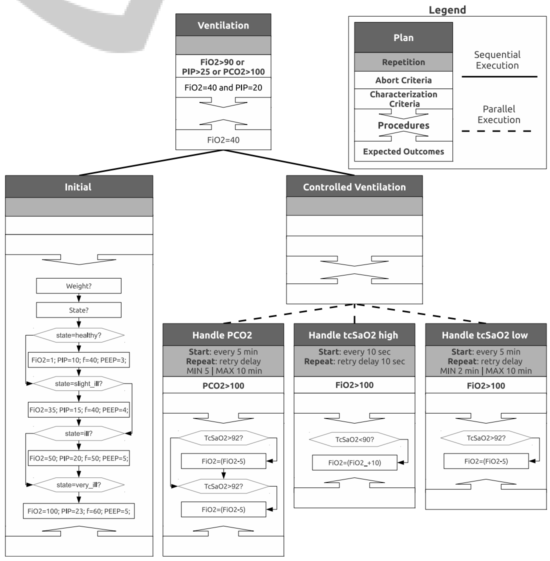

Figure 1: Example of CIG structure for a *Controlled Ventilation plan*.

(Gooch and Roudsari, 2011) identified 8 knowledge models implemented, related to clinical decisions, concluding that TNM was the most commonly adopted. The authors characterize TNM in two ways: general and formal. General TNMs are "flowcharts or process maps without formal semantics". Formal TNMs are "guideline-based clinical tasks – actions, decisions, queries – that unfold over time, with a formal syntax and semantics". According to (Peleg et al., 2003), TNMs succeed over alternative approaches, such as MLMs and ADTs, because they do not provide full support for conceptualizing a multi-step guideline that unfolds over time. Considering those aspects, in this work we focus on TNMs.

Figure 1 shows a usual structure of a TNM. As the 'Legend' (top right) shows, the basic component is a plan and it represents a procedure to be applied to some case. In each box, a rectangle with a label is the name of the plan. The top plan – labeled *Ventilation*– is the starting (seed) plan. The following plans (below) are sub-plans that can be triggered by the upper plan according to rules. To decide whether a plan should be applied or not, there are *Characterization Criteria* that define conditions to be analyzed in order to match the patient case to the diagnosis. Another aspect that can be used to determine if the plan is suitable to the case is the *Expected Outcomes*. The flow of actions and conditions appear whithin the *Procedures* box. Whenever a plan is recognized as a way to achieve the same result desired by the healthcare professional, the plan will be triggered.

A triggered plan will follow the *Procedures* that are recommended to be applied to the patient. The repetition of those internal procedures is guided according to the *Repetition* specification associated to the plan. Also, it is possible to associate *Abort Criteria* to interrupt the plan when a specific situation is achieved. A plan can trigger potential sub-plans, allowing to modularize and reuse plans that already encapsulate needed practices. The fact that the results can be unexpected, and consequently the sub-plans will be selected during the execution, produces an *emergent behavior* in the flow of activities. Several potential flows, constrained by rules, will dynamically shape one flow on-the-fly during the execution, interacting with contextual values. This work captures this rule-based and context-driven emergent behavior and adapts it to workflows.

To explain the characteristics of TNMs we use the same scenario that Aigner and Miksch (Aigner and Miksch, 2006) adopt to validate the CareVis platform: Infants Respiratory Distress Syndrome (I-RDS). As Miksch (Miksch et al., 1998) explains, "after I-RDS is diagnosed, a plan dealing with limited monitoring possibilities is activated, called *initial-phase*. Then follows, depending on the severity of the disease, three different kinds of plans, *controlled-ventilation*, *permissive-hypercapnia*, or *crisis-management*. Only one plan at a time can be activated, however the order of execution and the activation frequency of the three different plans are depending on the severity of the disease".

The scenario is shown on Figure 1. The plan execution order is read as top-down, left to right. Dashed lines from a plan represent alternative sub-plans that are used according to the conditions. To represent the internal `Procedures` , we use the same representation as CareVis: flow-charts. An internal procedure is described as a flow of CareVis 'single-steps'. Each single-step is either a variable assignment, a if-then-else construct (hexagon), an ask element (rectangle).

The *Ventilation* plan is activated when the characterization criteria *FiO2>40 and PIP=20* is satisfied. The plan and all sub-plans are aborted if the condition *FiO2>90 or PIP>25 or PCO2>100* is achieved. As *Ventilation* plan does not have internal procedures, the *Initial* plan is executed, following the sequential order specified. The *Initial* plan just has a set of internal procedures, such as ask *weight* and *state* of the patient and define variable values according to the state value (*healthy*, *slightly ill*, *ill* and *very ill*).

3 WORKFLOWS

A workflow is defined as the movement of tasks through a work process describing how tasks are structured, who performs them, the resources needed and their relative order (Dallien et al., 2008). On a computational context, a workflow specification can be seen as an abstraction that allows the structured composition of programs as a sequence of activities aiming a desired result (Ogasawara et al., 2009).

Business and science are the two main driving forces behind the development of Workflow Management Systems (WfMSs). According to Sonntag et al. (Sonntag et al., 2010), business workflows are focused on the control of the flow, adopt agreed-upon communication standards, in order to facilitate interoperation between different software systems and companies, and commonly are concerned about fault handling, transactions, or quality of service features. Scientific workflows, on the other hand, are focused on data transformation, commonly may involve computation-intensive tasks, and are concerned with the specification of explicit data flow, the exact reproducibility of workflows, or processing of data streams.

The Workflow Management Coalition (WfMC) is one important participant of those driving forces behind WfMSs development. Because of its involvement on the earlier stage of the creation of workflows, it defines the workflow components under the business perspective. In this work we use the WfMC definitions, relaxing them to a broader perspective that also covers scientific workflows (WfMC, 1999). Here we differentiate between a workflow specification and its instantiation (an actual execution). Moreover, we use the term 'task' as a synonym for 'activity' (where an activity is "a description of a piece of work that forms one logical step of a process" (WfMC, 1999). We likewise distinguish *Process definition* from *Process/Activity Instance*.

In order to help compare workflow and TNM approaches, we re-engineered the *Controlled Ventilation* plan of Section 2 using part of the graphical notation of flow-charts used on TNMs, but following a representation similar to that adopted by the Taverna (Hull et al., 2006) workflow system – see Figure 2.

The 'legend' (top right) shows the basic notion of structure of a process. The top rectangle with a text is the name of the process. The second one is the repetition criteria, which allows to define criteria to repeat the process and an interval for each repetition. Like TNMs, workflows may contain internal procedures, specified program code or instructions, that allows to perform activities. For instance, processes 'Handle PCO2' contains repetition criteria and embeds code that is repeated according to the criteria.

An important distinction between TNMs and workflows is the access to the variable values. While on TNM each value may act as a global variable, which can be accessed from anywhere in a plan, on workflows the values should be explicitly passed for each process through a port. Input ports receive the values of a process, while outport ports contains the result of a process. Of course, processes can read and write from a common storage, but this approach reduces the generality of processes.

To exemplify the use of a workflow, consider again the *Controlled Ventilation plan*, now executed as a workflow depicted in Figure 2. To determine whether the *Ventilation process* should be executed, the *Ventilation Activation Condition* process is introduced. If the condition *FiO2=40 and PIP=20* is satisfied, the result is passed to the *Pass Output port*, which indicates that the *Ventilation process* should be performed. If the condition is not satisfied, the result is passed to *Fail Output port*, finishing the execution of the workflow. Ports thus enable conditioning the path according to the obtained result.

Considering that the *Ventilation Activation Condition process* generated a *Pass Output* value, the *Ventilation process* receives an input value and passes it to its internal processes. As well as on Figure 1, *Initial* and *Controlled Ventilation* are sequentially executed. This execution is explicitly defined by the fact that the

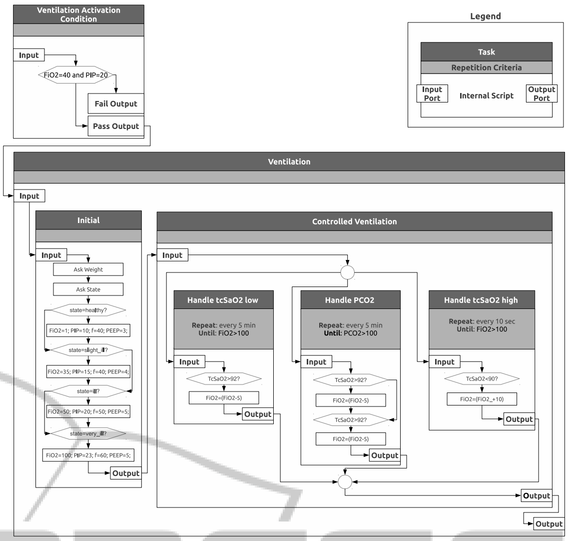

Figure 2: *Controlled Ventilation plan* re-engineered by us as a workflow structure.

Initial Output port is connected to the *Controlled Ventilation Input port*. While the *Initial process* uses its internal script to obtain the values of *weight* and *state*, *Controlled Ventilation* has sub-processes that are performed on any order. In fact, differently from TNMs, those sub-processes can be executed at the same time. The split-join, represented by a circle before and after the processes, indicates that any process (among *Handle tcSaO2 low*, *Handle PCO2* and *Handle tcSaO2 high*) can be executed. The split-join allows to execute different processes and then combines the results so that can be passed to another process or port.

If we proceed mapping a TNM specification to a workflow, the alternatives in each stage will grow exponentially. This effect shows the limits of trying to directly map a TNM specification in a classic workflow specification. Our proposal addresses this problem. It blends the TNM rule-based context-driven mechanism in a workflow system, providing an equivalent TNM ability of adapting itself according to the context.

4 DATA-DRIVEN VS PROCESS-DRIVEN APPROACHES

In order to summarize some of the main distinctions between the Workflow and TNM approaches is: scientific workflows are process driven and TNMs are data driven. Thus, bringing TNM characteristics into scientific workflows will make the latter both data and process driven.

TNMs are built to reduce the complexity to deal with large amounts of content that must be interpreted and to guide the use of procedures that are recommended to achieve some state. Because a patient may change his/her condition unexpectedly, TNMs allow to activate or deactivate content (guides and procedures) to treat the current state of the patient. All guides and procedures are pre-specified according to the recommendations of medical councils and well established procedures.

Scientific workflows are focused on processing high amounts of data. Usually there are two common situations: (i) a scientist wants to represent and automate a well known/consolidated experiment and execute it with different data (parameters) with none or few changes during the execution; (ii) the scientist wants to create an experiment to test a new hypothesis, so (s)he starts to build a workflow including and changing processes and parameters after different executions to analyze the results.

The difference between both approaches can be observable on the basic component of each one. TNMs have plans that associate content, procedures and other plans. Also, there are criteria that must be satisfied to activate the plan; such criteria allow selecting a plan over all other possible situations that may occur. Scientific workflows have processes that can be associated to others to create the flow of data and transform/process it. The data that is passed from one process to another can lead to different paths, but commonly the possible number of paths and changes are not as high as on TNMs. When a new situation occurs and there is a need to incorporate a new process, the scientist changes the workflow – e.g., choosing from a repository of processes, including an external tool or creating a new process.

Another factor is that, to deal with all resources involved, it is necessary to provide tools to visualize and filter information, as well as be aware of the origin and the involvement of resource with respect to the data. A key issue is to maintain provenance information on data and procedures – e.g., what kind of data was used where, how and by whom. Here, workflow systems already provide some sort of traceability mechanism.

In this paper we focus on the main features required to achieve a data-driven workflow: changing workflow tasks to adapt them to a situation. Our approach is presented next.

5 TOWARDS MAKING SCIENTIFIC WORKFLOWS DATA-DRIVEN

As presented in previous sections, a key factor to produce workflows able to adapt to the high dynamic health environments is the rule-based TNM approach, which triggers modules according to context values, producing an emergent behavior. Different from TNM specifications, which start from a seed and expands the flow, a workflow has a predefined starting flow. Therefore, we translated the mechanism of *unfolding* new activities on-the-fly according rules (TNM) to a mechanism of *adapting* the existing workflow on-the-fly according to rules (our approach).

In order to fully support dynamic workflow changes according to context, and give more flexibility to workflow execution, we decided to adopt reasoning capabilities and combine them with domain semantics. The solution found was to (a) use ontologies as the basis for workflow and concept representation, and (b) create adaptation rules to represent experts' knowlege. An *adaptation rule* characterizes a situation (cause) and defines the change (consequence) that should be performed to workflow so that it can be adapted to the context.

Given that there exist distinct workflow representations and formats (according to the WfMS adopted), the overall dynamic adaptation cycle can be described as follows. The workflow execution on the Web is monitored at each activity.Every new workflow state is mapped to an ontology instance, to which reasoning is applied, resulting in a new ontology instance (containing recommended modifications to that state). This new instance is transformed back to the workflow representation of the WfMS adopted, which is shown back to the user on the Web interface, to be validated or modified.

The architecture of our work is presented in Figure 3, where CRec (Context Record) is the linearized representation of an ontology, representing a workflow state. The main components are the following. A **Scientific Workflow Management System** (SWfMS) used to create and execute workflows, through a Web Interface, extended to incorporate the adaptation mechanism. A **Context-Aware Extension**, responsible for monitoring workflow execution and identifying the events that occur during that execution. A **Semantic Mapper** that is responsible for translating a workflow to a CRec and vice-versa, allowing the *Context-Aware Extension* to update the original workflow on SWfMS. An **Ontology** that specifies the main classes used to represent workflow activities and their parameters. Also, there are additional classes used to complement activity information (such as activation status) and associated content (for instance, patient data). A CRec (resp. CRec') is a **Context Record** that is an ontology instance that represents the current state of a workflow and all information regarding the patient under care. A set of **Adaptation Rules** (*AR*), written in SWRL, that will be used with an Ontology and a CRec to identify the changes recommended to the current situation. A **Context Adaptation Engine** (*CAEng*) that uses an Ontology Reasoner to process the *AR* combined to an Ontology and a *CRec*. The reasoning updates *CRec* to guide the adaptations that should be made on workflow.

The architecture enables dynamic adaptation of workflows to a context as follows. A SWfMS is accessed through a Web Interface (Figure 3, interaction 1). A workflow is assigned to be executed on the SWfMS (on interaction 2) and the SWfMS loads it (interaction 3) and updates the Web Interface (interaction 4). A user requests the SWfMS to start the executing workflow (interactions 5 and 6). When an activity is performed, the *Context-Aware Extension* captures the workflow state (interaction 7) and passes it to the *Semantic Mapper* (interaction 8). Next, the *Semantic Mapper* uses the workflow state to create an ontology instance that represents it (interaction 9) – *CRec*. The *CAEng* combines *AR*, *Ontology*, and *CRec* to identify the adaptations recommended to the workflow and apply them to *CRec*, creating a new version of it (*CRec'*. The *Semantic Mapper* receives *CRec'*(interaction 10) and maps its data to the workflow representation, passing it back to the *Context-Aware Extension* (interaction 11). The workflow on the SWfMS is updated by the *Context-Aware Extension* according to the new workflow (interaction 12). The SWfMS updates the Web Interface (interaction 13). Finally, the user sees the adapted version of the workflow (interaction 14).

Our architecture may be used on different scenarios and on different SWfMS, but its components created must be changed according to the scenario (e.g., rules and ontology customization) and platform adopted (e.g., Semantic Mapper).

6 INSTANTIATION ON THE WEB

We applied the principles behind our architecture to the context of nursing care to attest the flexibility of execution of the workflows. The tasks used to construct the ontology and the rules were based on the PROCEnf system (Peres et al., 2009), developed and adopted by the hospital of the University São Paulo. PROCEnf is also in process of adoption at the our

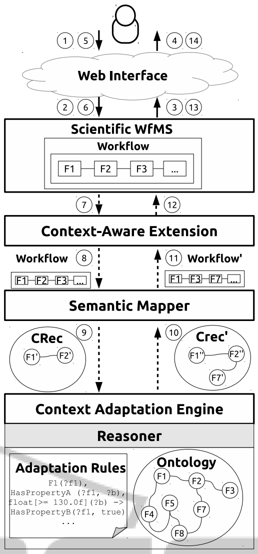

Figure 3: Architecture for context-based workflow adaptation.

University Hospital[1]. Like most such systems, PROCEnf is heavily centered on offering health professionals sets of forms to fill.

We chose PROCEnf among other reasons, because we can reuse its components. Moreover, given our need to validate our implementation with actual users, this choice offers to the nursing staff of our University Hospital a set of forms and vocabulary they are familiar with.

Based on PROCEnf and on the work described in (Doenges and Moorhouse, 2008), we described the flow of a patient's admission and monitoring process in a hospital. In a general way, the process includes an iterative step which is started by an anamnesis interrogation, which is an assessment phase to evaluate the patients' conditions. Next, there is the analysis of recorded data to diagnose the problem and to identify expected outcomes (prognosis). Based on those expected outcomes, health interventions are planned and applied. To identify whether the outcomes were achieved or not, the intervention results are analyzed and the amamnsesis records are updated. If the treatment achieves the expected outcomes, the patient can be released. Otherwise, a new iteration occurs.

To model PROCEnf forms as workflow activities, we created an ontology. In our current stage of research, we have about 30 activity classes involved with the diagnosis phase. Those classes are directly related to the PROCEnf system. As can be seen on

[1]The hospital complex of the University of Campinas alone receives about 500,000 appointments, with over 43,000 admissions and 34,000 surgical interventions per year.

classes 'CardiacFunction' and 'VitalSigns', there are different attributes that can be filled to draw a diagnosis. Depending on how an attribute is filled, distinct decisions have to be taken – i.e., the workflow has to be changed dynamically.

Another issue we faced was the integration of workflow structures, rules, and reasoning capabilities. As explained before, this was solved by treating all information within a single ontology-based reasoning framework. The *Semantic Mapper* transforms the workflow structures (together with a current workflow state) into an ontology instance, that is enhanced with the current patient state. This is then treated by the reasoner as any other ontology, using adaptation rules that encode domain and user knowledge. Such adaptations are performed based on the adaptation rules we created in SWRL. Those rules can be classified into four types: I) **Propagate field (parameter) value**: repeat the value of a field to related fields, avoiding the need to fill the value in other forms. II) **Infer field (parameter) value**: the value of a field according to the value of other fields. III) **Change form (activity) position**: the order in forms which are filled can be changed, increasing or decreasing filling priority, making related forms closer and unrelated or unlikely forms more distant in the filling order. IV) **Include/Remove form (activity)**: a form can be removed when a field makes it incompatible with the purpose of the form (e.g., pregnancy issues related to men). Inclusion occurs whenever a situation makes a previoulsy removed form viable.

Table 1 presents a subset of the rules, highlighting the types and some of their potential. The rules are specified in SWRL and are specified in cause/consequence form. The cause (antecedent) can be defined according to the value of fields and forms, by the composition of different predicates. A consequence (consequent) is the result of checking the antecedent, e.g., changing of a property value, including or removing values. For instance, rule *R2* says that when the *Systolic Blood Pressure* value of *Vital Signs* is large than 130, then the field *Has Systolic Blood Pressure Out of Expected Range* should be true.

Let us exemplify how adaptation rules are used in the reasoning process. Consider the following scenario: a nurse should fill a set of forms about different patient conditions, including vital signs, self-care, and comfort. At the beginning of the process, the professional fills a form about *Vital Signs*. If the value of *systolic blood pressure* is filled with value large than 130, rule *R2* (Table 1) has its antecedent condition satisfied, so the inference engine applies the consequent which characterizes the patient as having a *systolic pressure out of expected range*. As a result, two other rules are to be executed, leading to the new adaptations: *R1*, to propagate the value of *systolic blood pressure* to the *Cardiac Function* task and *R4* to increase the priority of the *Cardiac Function* task, which will put this form closer to the *Vital Signs* form. This order change highlights the need to provide additional information about Cardiac Function details and to avoid the loss of focus and information regarding this problem.

7 CONCLUSIONS

In this paper, we presented a proposal to adapt scientific workflows to the context of healthcare management, and its Web implementation. Our modifications to an SWfMS place such systems closer to the domain of healthcare, thanks to the inspiration from CIGs and to the fact that such systems, thanks to the data-driven flow, are suitable to dynamic scenarios.

We designed the model in which an SWfMS is extended by a layer which uses ontologies combined to rules as a means to make scientific worflows more data-driven than just process-driven. This in turn, brings more flexibility to the execution of the tasks as well as allows to choose whether a task should be executed according to more sophisticated conditions.

Future steps of this research will cope with TNM features, presented in Section 4, working on the better graphical integration and adaptation of the work with the SWfMS, allowing to: (i) Integrate the ontology content to the workflow tasks, allowing users to navigate through the concepts of ontologies and use those concepts also as a way to filter tasks and conveniently find resources on SWfMS; (ii) Provide support to the hierarchical organization of workflow tasks, giving more dynamic flow to task execution as well as making it more similar to the TNM approach.

ACKNOWLEDGMENTS

Work partially financed by FAPESP (grant 2011/17196-0), FAPESP/Cepid in Computational Engineering and Sciences (2013/08293-7), the Microsoft Research FAPESP Virtual Institute (NavScales project), CNPq (MuZOO Project), FAPESP-PRONEX(eScience project), INCT in Web Science, and individual grants from CNPq. We also thank healthcare providers from University of Campinas and University of São Paulo, who provided us support in this research.

Table 1: Adaptation Rules.

Rule	Antecedent	Consequent	Type
R1	CardiacFunction(?cf), VitalSigns(?vs), systolicBloodPressure(?vs, ?x)	systolicBloodPressure(?cf, ?x)	Propagate Value
R2	VitalSigns(?vs), systolicBloodPressure(?vs, ?x), float[>= 130.0f](?x)	hasSystolicPressureOut-OfExpectedRange(?vs, true)	Infer Value
R3	VitalSigns(?vs), isActive(?vs, true), hasSystolicPressureOutOfExpectedRange(?vs, false), CardiacFunction(?cf)	position(?cf, 99)	Decrease Priority
R4	VitalSigns(?vs), hasSystolicPressureOutOfExpectedRange(?vs, true), isActive(?vs, true), CardiacFunction(?cf), position(?vs, ?po)	position(?cf, ?po)	Increase Priority
R5	ValuesAndBeliefs(?va), Evaluation(?e), hasPatient(?e, ?p), isActive(?e, true), hasAge(?p, ?age), integer[<= 12](?age)	isRemoved(?va, true)	Remove
R6	ValuesAndBeliefs(?va), Evaluation(?e), hasPatient(?e, ?p), isActive(?e, true), hasAge(?p, ?age), integer[>12](?age), isRemoved(?va, true)	isRemoved(?va, false)	Include

REFERENCES

Aigner, W. and Miksch, S. (2006). Carevis: Integrated visualization of computerized protocols and temporal patient data. *Artificial Intelligence in Medicine*, 37(3):203–218.

Bharathi, S., Chervenak, A., Deelman, E., Mehta, G., Su, M.-H., and Vahi, K. (2008). Characterization of scientific workflows. *2008 Third Workshop on Workflows in Support of Large-Scale Science*, pages 1–10.

Dallien, J., MacCaull, W., and Tien, A. (2008). Initial work in the design and development of verifiable workflow management systems and some applications to health care. In *Proceedings of the 2008 5th International Workshop on Model-based Methodologies for Pervasive and Embedded Software*, MOMPES '08, pages 78–91, Washington, DC, USA. IEEE Computer Society.

Dang, J., Hedayati, A., Hampel, K., and Toklu, C. (2008). An ontological knowledge framework for adaptive medical workflow. *Journal of biomedical informatics*, 41(5):829–36.

Davidson, S. and Freire, J. (2008). Provenance and scientific workflows: challenges and opportunities. In *SIGMOD Conf.*, pages 1345–1350. Citeseer.

Doenges, M. and Moorhouse, M. (2008). *Application of nursing process and nursing diagnosis: an interactive text for diagnostic reasoning*. G - Reference, Information and Interdisciplinary Subjects Series. F.A. Davis.

Gooch, P. and Roudsari, A. (2011). Computerization of workflows, guidelines, and care pathways: a review of implementation challenges for process-oriented health information systems. *Journal of the American Medical Informatics Association : JAMIA*, 18(6):738–48.

Hull, D., Wolstencroft, K., Stevens, R., Goble, C., Pocock, M. R., Li, P., and Oinn, T. (2006). Taverna: a tool for building and running workflows of services. *Nucleic acids research*, 34(-):W729–32.

Miksch, S., Kosara, R., Shahar, Y., and Johnson, P. D. (1998). AsbruView: Visualization of Time-Oriented, Skeletal Plans. pages 11–18.

Ogasawara, E., Paulino, C., Murta, L., Werner, C., and Mattoso, M. (2009). Experiment line: Software reuse in scientific workflows. In Winslett, M., editor, *Scientific and Statistical Database Management*, volume 5566 of *Lecture Notes in Computer Science*, pages 264–272. Springer Berlin Heidelberg.

Panzarasa, S. and Stefanelli, M. (2006). Workflow management systems for guideline implementation. *Neurological sciences : official journal of the Italian Neurological Society and of the Italian Society of Clinical Neurophysiology*, 27 Suppl 3:S245–9.

Peleg, M., Tu, S., Bury, J., Ciccarese, P., Fox, J., Greenes, R. A., Miksch, S., Quaglini, S., Seyfang, A., Shortliffe, E. H., Stefanelli, M., and et al. (2003). Comparing computer-interpretable guideline models: A case-study approach. *JAMIA*, 10:2003.

Peres, H. H. C., Cruz, D. D. A. L. M. D., Lima, A. F. C., Gaidzinski, R. R., Ortiz, D. C. F., Trindade, M. M. E., Tsukamoto, R., and Conceição, N. B. (2009). Desenvolvimento de Sistema Eletrônico de Documentação Clínica de Enfermagem estruturado em diagnósticos, resultados e intervenções. *Revista da Escola de Enfermagem da USP*, 43(spe2):1149–1155.

Schick, S., Meyer, H., Bandt, M., and Heuer, A. (2012). Enabling yawl to handle dynamic operating room management. In Daniel, F., Barkaoui, K., and Dustdar, S., editors, *Business Process Management Workshops (2)*, volume 100 of *Lecture Notes in Business Information Processing*, pages 249–260. Springer.

Sonntag, M., Karastoyanova, D., and Deelman, E. (2010). Bridging the gap between business and scientific workflows: Humans in the loop of scientific workflows. In *e-Science (e-Science), 2010 IEEE Sixth International Conference on*, pages 206–213.

Unertl, K. M., Weinger, M. B., Johnson, K. B., and Lorenzi, N. M. (2009). Describing and modeling workflow and information flow in chronic disease care. *Journal of the American Medical Informatics Association : JAMIA*, 16(6):826–36.

Vilar, B. S. C. M., Medeiros, C. B., and Santanchè, A. (2013). Towards adapting scientific workflow systems to healthcare planning. In *HEALTHINF - International Conference on Health Informatics*.

Wei Tan, M. Z. (2013). *Business and Scientific Workflows: A Web Service-Oriented Approach*. Wiley.

WfMC (1999). Terminology and Glossary Document Number WFMC-TC-1011 - Issue 3.0. Technical report, Workflow Management Coalition.