1-1-2010

A framework for fetecting interactions between co-incident clinical processes

Aditya K. Ghose
University of Wollongong, aditya@uow.edu.au

Kerry Hinge
University of Wollongong, khinge@uow.edu.au

Andrew Alexis Miller
University of Wollongong, amiller@uow.edu.au

A framework for fetecting interactions between co-incident clinical processes

Abstract

The detection of treatment conflicts between multiple treatment protocols that are co-incident is a difficult and open problem that is particularly exacerbated regarding the treatment of multiple medical conditions co-occurring in aged patients. For example, a clinical protocol for prostate cancer treatment requires the administration of androgen-suppressing medication, which may negatively interact with another, co-incident protocol if the same patient were being treated for renal disease via haemodialysis, where androgen-enhancers are frequently administered. These treatment conflicts are subtle and difficult to detect using automated means. Traditional approaches to clinical decision support would require significant clinical knowledge. In this paper, the authors present an alternative approach that relies on encoding treatment protocols via process models (in BPMN) and annotating these models with semantic effect descriptions, which automatically detects conflicts. This paper describes an implemented tool (ProcessSEER) used for semantic effect annotation of a set of 12 cancer trial protocols and depicts the machinery required to detect treatment conflicts. The authors also argue whether the semantic effect annotations of treatment protocols can be leveraged for other tasks.

Disciplines

Medicine and Health Sciences

Publication Details

Hinge, K., Ghose, A. & Miller, A. (2010). A framework for fetecting interactions between co-incident clinical processes. International Journal of E-Health and Medical Communications, 1 (2), 24-35.

A Framework for Detecting Interactions Between Co-Incident Clinical Processes

Kerry Hinge, University of Wollongong, Australia

Aditya Ghose, University of Wollongong, Australia

Andrew Miller, University of Wollongong, Australia

ABSTRACT

The detection of treatment conflicts between multiple treatment protocols that are co-incident is a difficult and open problem that is particularly exacerbated regarding the treatment of multiple medical conditions co-occurring in aged patients. For example, a clinical protocol for prostate cancer treatment requires the administration of androgen-suppressing medication, which may negatively interact with another, co-incident protocol if the same patient were being treated for renal disease via haemodialysis, where androgen-enhancers are frequently administered. These treatment conflicts are subtle and difficult to detect using automated means. Traditional approaches to clinical decision support would require significant clinical knowledge. In this paper, the authors present an alternative approach that relies on encoding treatment protocols via process models (in BPMN) and annotating these models with semantic effect descriptions, which automatically detects conflicts. This paper describes an implemented tool (ProcessSEER) used for semantic effect annotation of a set of 12 cancer trial protocols and depicts the machinery required to detect treatment conflicts. The authors also argue whether the semantic effect annotations of treatment protocols can be leveraged for other tasks.

Keywords: *BPMN, Co-Incident, Co-Occurring, Encoding Treatment Protocols, Multiple Medical Conditions, Multiple Treatment Protocols, Semantic Effect Annotation*

INTRODUCTION

The notion of *care-flow management* (Panzarasa, Maddè, Quaglini, Pistarini, & Stefanelli, 2002) has become the focus of considerable research attention in the recent past. It builds on the premise that process management principles and techniques can deliver value in clinical settings as much as it delivers value in settings such as business process management. Clinical process management can help encode clinical guidelines which can provide a reference baseline for clinicians. These can leverage the coordination capabilities of process engines in ensuring that treatment steps are executed correctly relative to reference guidelines. More generally, care-flow management also addresses the administrative aspects of health care, both

DOI: 10.4018/jehmc.2010040103

from the perspective of health care providers and patients (Curry, McGregor, & Tracey, 2006).

Existing techniques/notations for modelling processes, such as the industry-standard Business Process Modelling Notation (BPMN) (OMG, 2006), only model the coordination semantics of processes, but offer no facility for describing the effects of processes (or steps/activities within processes). Thus, we are able to clearly specify the required sequencing of activities, for instance, but cannot specify the effects that these activities would have on the domain/context in which the process would execute (beyond the minimal information that can be conveyed via the nomenclature of tasks). However, these effect descriptions are critical in determining whether process designs have been correctly formulated. As well, much of the analysis required for process compliance management (Ghose & Koliadis, 2007), change management (Koliadis & Ghose, 2006), enterprise process architectures (Koliadis, Ghose, & Padmanabhuni, 2008) and management of a business process life cycle (Koliadis, Vranesevic, Bhuiyan, Krishna, & Ghose, 2006) relies on being able to refer to the *effect semantics* of business processes.

In this paper, we leverage a technique (and supporting tool - ProcessSEER) (Hinge, Ghose, & Koliadis, 2009) that provides a practitioner-accessible means for providing semantics effect annotations of process models to deliver value in clinical process management in a range of different ways. We focus primarily on the use of this machinery in detecting *treatment conflicts* between co-incident treatment protocols, i.e., situations where the application of a treatment protocol on a patient contra-indicates the application of another treatment protocol on the same patient. These are often subtle, and otherwise difficult to detect using automated means. We also argue that semantic effect annotations of treatment protocols can be leveraged for a variety of other tasks, including identifying instances where different specialists arrive at differing interpretations of the same protocol, as well as pedagogical applications.

The language in which these effects need to be specified should ideally be formal, permitting sophisticated tool support for several of the analysis and reasoning tasks mentioned above. Formal languages are typically not practitioner-accessible while informal annotations make substantive tool support difficult to devise. The use of controlled natural language (CNL) (Schwitter & Fuchs, 1996) is an effective compromise between these two extremes, by offering the analyst a repertoire of sentence schemas in which to describe the effects - populating a sentence schema generates a correspondingly instantiated formal annotation. To avoid placing an unduly heavy burden of annotation on analysts, our approach only requires that analysts provide a description of the *immediate effects* of each process task, i.e., a context-independent specification of the functionality (together with relevant associated ramifications) of each task. These are then accumulated into *cumulative effect annotations* in a context-sensitive manner, such that the cumulative effect annotations associated with any task in a BPMN process model would describe the effects achieved by the process were it to execute up to that point. We note that such a description will necessarily be non-deterministic, i.e., there might be alternative *effect scenarios* that might transpire if a process has executed up to a certain point in a process model. The non-determinism stems from two sources. First, a process might have taken different paths through a process model to arrive at a certain point. Second, the effects of certain process steps might "undo" the effects of prior process steps. This is often described as the *belief update* or *knowledge update* problem - multiple alternative means of resolving the inconsistencies generated by the "undoing" of effects is another source of non-determinism.

After reviewing relevant background in Section 2, we summarise key elements of the ProcessSEER framework (Hinge et al., 2009) for semantic effect annotation of process models in Section 3. In Section 4, we describe the machinery for detecting conflicts between co-incident clinical processes. In Section 5, we

provide a detailed example illustrating some of the capabilities of the framework, followed by some brief comments on evaluation, and conclusions.

BACKGROUND

Clinical Process/Care-flow Management: The notion of *care-flow management* (Panzarasa et al., 2002) has become the focus of considerable research attention in the recent past. It builds on the premise that process management principles and techniques can deliver value in encoding (and coordinating execution via process engines) of clinical guidelines. More generally, care-flow management also addresses the administrative aspects of health care, both from the perspective of health care providers and patients (Curry et al., 2006). Clinical procedures are performed by a variety of clinicians and treatments are often prescribed autonomously. Even with the abundance of text-based documentation on medical procedure it can be difficult to identify potential conflicts between treatments. Patient records may contain a summary of existing and past treatments but lack certain details that could impact on future or concurrent treatments if undetected. Having this text-based documentation does not guarantee that it will be used whereas computer-based patient-specific reminders that are integrated into the clinician's work flow have proven to be far more effective (Stefanelli, 2002).

Care-flow processes are often represented in a diagrammatical format that provides a visually intuitive representation of the activities required to treat a patient's condition. Although process modelling has been used extensively in the business community it is a relatively new innovation within the health care industry. Described in Curry et al. (2006) is a tool that utilises Workflow Reference Models (Hollingsworth, 1995) to visualise a patient's journey through a Health Care Organisation (HCO). The models prove to be particularly useful for encouraging group communication within a HCO and promoting ownership and responsibility among active participants involved in the process. The models are particularly good for training purposes because they visually translate much easier than a text document.

The integration of Care-flow Management Systems (CMS) can greatly improve a patient's journey through the health care system. CMSs can be used in a variety of ways, as control mechanisms to constrain the operations of health care workers to predefined treatment protocols (Curia, Gallucci, & Ruffolo, 2005), as decision support mechanisms that assist clinicians with prescribing treatment protocols (Maximini & Schaaf, 2003) and as central repositories for the comparison and analysis of different treatment protocols (Ruffolo, Manna, Cozza, & Ursino, 2007). Of these three only (Ruffolo et al., 2007) addresses the need for internet compatibility so that stored information may be accessed by any HCO. This is particularly important given that a patient's journey through the health care system will place them in the hands of many autonomous HCOs. Internet access to other autonomous HCOs' treatment protocols would greatly assist clinicians with the prescription of their own treatments (Anzböck & Dustdar, 2004).

Clinical Decision Support Systems (CDSS) fall into six categories, 'Alerts and Reminders', 'Diagnostic Assistance', 'Therapy Critiquing and Planning', 'Prescribing Decision Support Systems', 'Information Retrieval' and 'Image Recognition and Interpretation' (Coiera, 2003). Relative to this taxonomy, the research reported here corresponds most closely to 'Prescribing Decision Support Systems', and to a lesser degree to 'Alerts and Reminders' and 'Therapy Critiquing and Planning'.

Business Process Modelling Notation: The Business Process Modelling Notation (BPMN) (OMG, 2006), standardised by the Object Management Group (OMG) in February 2006 was motivated be a need to develop a modelling notation that could bridge the gap between process design and process implementation and translate easily into executable code, notably Business Process Execution Language for Web Services (BPEL4WS). While other

modelling notations have proven effective at modelling programming code artifacts, in the case of Unified Modelling Language (UML) (OMG, 2009), or providing an intuitive visual representation for describing a process, in the case of a Workflow Diagram produced in Microsoft Visio, BPMN has combined both these attributes into a single notation. A BPMN model can therefore be used as a visual instructional tool for human consumption and as a template for automated code generation. When process segments are identified for automation then a programmer can use the BPMN model to automatically generate code. This is an important economic consideration when planning for future automation of processes that are currently performed manually. BPMN is an evolving standard (currently at version 1.2) that is continually growing in expressiveness through consultation with domain experts.

Semantic Process Management: Annotating and analysing specifications of program functionality, in order to help establish program correctness, has a long tradition dating back to the introduction of the axiomatic techniques proposed by Hoare and Dijkstra (Huth & Ryan, 2004). With sufficient information, these forms of annotations provide (Shanahan, 1997) a basis for answering questions relating the identification of: the conditions enabling a process to be performed (i.e., postdiction); the conditions resulting from a process being performed in some context (i.e., prediction); and, the processes with the capability of realising a set of conditions when executed in some context (i.e., planning). Recently, similar proposals have emerged in the domain of web services (McIlraith, Son, & Zen, 2001) (Martin & Domingue, 2007). These forms of specification can be effective for performing analyst related tasks, however their utility and availability in some situations can be limited (e.g., cost restrictions) - warranting a need for "partiality" and "lightweight" approaches (Jackson & Wing, 1996). The contribution in this paper are techniques to leverage a partial specification of functional effects annotated to business process models.

A lot of effort is currently being directed into semantic annotation for web service or process discovery. Recently, a semantic annotation framework was developed to facilitate the interchange of process models and their discovery (Lin, 2008). Ontologies are used in this framework as a classification repository for the identification of processes or sub-processes that satisfy the selection criteria. Our tool will reduce the risk of modifying existing processes by alerting the analyst to the consequences of design time decisions. We use ontologies in conjunction with a CNL taxonomy to define the vocabulary used in the effect annotations for the purpose of translation into formal logic. Our process differs from that described in (Lin, 2008) in that effect annotations are not simply used for term comparison but also for reasoning about process outcomes.

In (Soffer & Wand, 2004), a Generic Process Model (GPM) is proposed to encode and extend the representation of processes with state and stability (i.e., goal) relevant information. These notions of state and stability lead to a general notion of validity of process models (primarily w.r.t. goal reachability). In (Soffer, 2005), the GPM is used as a basis for identifying the scope of changes that can be made to an existing process given changes to GPM-related phenomena (e.g., goal change). Some of the techniques outlined in this paper, such as the accumulation procedure, help leverage partial and symbolic state descriptions to perform goal and change relevant analysis. In the SBPV approach (Weber, Hoffman, & Mendling, 2008), a scheme for annotating and propagating a restricted form of axiomatic task descriptions is introduced for a restricted class of process models, but differs in several key ways to our work. Our approach provides a parsimonious extension to the modelling framework (the analyst's effort is only extended by requiring immediate effect specifications of tasks in the BPMN model) and is driven by the need to identify the minimal amount of semantic annotation required to meet the requirements of functions such as compliance management, process change and

life-cycle management, enterprise process architectures etc. The SBPV approach, on the other hand, requires complete specifications of both pre-conditions and post-conditions that are context-sensitive, thus placing a somewhat onerous burden on the analyst (besides additional annotations required for reachability analysis, which we do not consider). Our machinery for contextualising context-independent task effect specifications provided by analysts solves a harder problem, by permitting non-determinism in effect scenarios. We consequently cannot provide polynomial-time guarantees as the SBPV framework can. We believe this is not a significant impediment since design, annotation and propagation tasks do not normally involve real-time constraints, and afford the luxury of slower off-line computation. Our evaluation has shown that we still are able to meet reasonable processing-time bounds.

SEMANTIC EFFECT ANNOTATION OF PROCESS MODELS

ProcessSEER (Hinge et al., 2009) is a tool that allows practitioners to annotate semantic effects to process activities/tasks and performs on-demand, anytime computation of cumulative effects. There are two stages to effect accumulation. The first stage in effect accumulation involves deriving a *scenario label*(Ghose & Koliadis, 2008) which provides the organising locus for our procedure. For obtaining the effect scenario at a given point in a process we compute the set of scenario labels at that point. A scenario label is a precise list of tasks that define a path leading from the Start Event in a model to the selected task. The simplest form of scenario label is a sequence of tasks. For example, $\langle S, T_1, G_1, T_2, G_2, T_6 \rangle$ is a scenario label where S is the start event. A scenario label can either be a sequence, denoted by the $\langle\rangle$ delimiters, or a set denoted by the $\{\}$ delimiters or combinations of both. The set delimiters are used to deal with parallel splits, and distinct elements in a set can be performed in any order.

The second stage of effect accumulation involves the processing of immediate effect annotations for each of the tasks listed in the scenario label using a pair-wise operation where the immediate effect of S is combined with the immediate effect of T_1, the result being the cumulative effect at T_1. The cumulative effect at T_1 is then combined with the immediate effect of T_2 resulting in the cumulative effect at T_2 and so on up to T_n.

Contiguous Tasks: We define a process for *pair-wise effect accumulation*, which, given an ordered pair of tasks with effect annotations, determines the cumulative effect after both tasks have been executed in contiguous sequence. We assume throughout, the existence of a background knowledge-base (KB) that provides an additional basis for consistency. Consider the following simple example, where task T_2 follows task T_1, such that T_2 somehow "undoes" the effects of T_1 or changes the status of some entity referred to in T_1. For instance, the status of a cheque submitted in T_1 might be "not yet cleared", while the immediate effect of the "cheque clearance" task T_2 might be to set its status to "cleared". A background rule that specifies that a cheque cannot have a "cleared" and "not yet cleared" status simultaneously ensures that we do not counter-intuitively obtain both status descriptions in the same effect scenario.

The procedure serves as a methodology for analysts to follow if only informal annotations are available. We assume that the effect annotations have been represented in conjunctive normal form (CNF) where each clause is also a *prime implicate* (Raut & Singh, 2004) (this provides a non-redundant canonical form).

Simple techniques exist for translating arbitrary sentences into the conjunctive normal form, and for obtaining the prime implicates of a theory (references omitted for brevity). Let $\langle T_i, T_j \rangle$ be an ordered pair of tasks connected via a sequence flow such that T_i precedes T_j, let e_i be an effect scenario associated with T_i and e_j be the immediate effect annotation associated with T_j. Let $e_i = \{c_{i1}, c_{i2}, \ldots, c_{im}\}$ and $e_j = \{c_{j1}, c_{j2}, \ldots, c_{jm}\}$ (we can view CNF sentences as sets of clauses, without loss of generality). If $e_i \cup e_j$ is consistent, then the resulting cumulative effect, denoted by $acc(e_i, e_j)$, is $e_i \cup e_j$. Else, we define $e_{i'} \subseteq e_i$ such that $e_{i'} \cup e_j$ is consistent and there exists no $e_{i''}$ such that $e_{i'} \subset e_{i''} \subseteq e_i$ and $e_{i''} \cup e_j$ is consistent. We define $acc(e_i, e_j) = e_{i'} \cup e_j$. We note that $acc(e_i, e_j)$ is non-unique i.e., there are multiple alternative sets that satisfy the requirements for e_i. In other words, the cumulative effect of the two tasks consists of the effects of the second task plus as many of the effects of the first task as can be consistently included. We remove those clauses in the effect annotation of the first task that contradict the effects of the second task. The remaining clauses are undone, i.e., these effects are overridden by the second task.

In the preceding, we assume that all consistency checks implicitly include a background knowledge base (KB) containing rules and axioms. Thus, the statement that $e_{i'} \cup e_j$ is consistent, effectively entails $e_{i'} \cup e_j \cup KB$ is consistent. We omit references to KB for ease of exposition. The following example illustrates an application of this definition.

Example: Let e_1 and e_2 represent effect annotations at T_1 and T_2 in a process model where T_2 immediately follows T_1. Let e_1 represent a cumulative effect annotation, i.e., an effect scenario, while e_2 represents an immediate effect annotation. At T_1 the cumulative effect is $(p \wedge q)$ and the immediate effect of T_2 is r. A rule exists in the KB that states $KB = r \rightarrow \neg(p \wedge q)$.

$$e_1 = (p \wedge q)$$
$$e_2 = r$$
$$KB = r \rightarrow \neg(p \wedge q)$$
$$\neg(p \wedge q) \equiv (\neg p \vee \neg q)$$

Applying the definition above, the two alternative effect scenarios describing the cumulative effects at T_2 are $\{p, r\}$ and $\{q, r\}$.

In addition to pair-wise effect accumulation across scenario labels, we need to make special provision for the following: (1) accumulation across AND-joins, and (2) accumulation of effects over message flows (extending the framework presented in (Ghose & Koliadis, 2008)). Consider the scenario label $\langle S, T_h, \{\langle T_{i1}, T_{i2}, \ldots, T_{in}\rangle, \langle T_{j1}, T_{j2}, \ldots, T_{jm}\rangle\}, T_k \rangle$. Let the immediate effects of T_{i1}, T_{j1} and T_k be e_{i1}, e_{j1} and e_k respectively. Let E_h, E_{in} and E_{jm} be the set of cumulative effect scenarios associated with T_h, T_{in} and T_{jm} respectively. The set of cumulative effect scenarios associated with T_{i1} is given by $\{acc(e, e_{i1}) \mid e \in E_h\}$. Similarly, the set of cumulative effect scenarios associated with T_{j1} is given by $\{acc(e, e_{j1}) \# e \in E_h\}$. In other words, we accumulate over the pair of tasks $\langle T_h, T_{i1} \rangle$ as if they constitute a con-

tiguous pair (and similarly for the pair $\langle T_h, T_{j1} \rangle$). We accumulate across AND-joins in the following manner. The set of cumulative effect scenarios associated with T_h is given by $\{acc(es_i, e_k) \cup acc(es_j, e_k) \# es_i \in E_{in}, es_j \in E_{jm}$ and es_i, es_j are exclusion-compatible$\}$. In other words, we pair-wise accumulate the immediate effect of T_k with each effect scenario of each of tasks preceding the AND-join, but then combine them via set union since every possible combination of the prior scenarios could potentially transpire. *Exclusion-compatibility* provides a guarantee that the effect scenarios could potentially occur together, i.e., that they do not have a mutually exclusive (XOR) split in their antecedents relative to each other. Exclusion-compatibility is determined using the *exclude-set* mechanism described in the next section. Note that we do not consider the possibility of a pair of effect scenarios es_{1i} and es_{2j} being inconsistent, since this would only happen in the case of intrinsically and obviously erroneously constructed process models.

Much of the earlier and following discussion pertains to flows within individual pools. Message flow links across pools can be dealt with in a relatively straightforward fashion by requiring an immediate effect annotation for each incoming message. These effects are combined via conjunction with the immediate effects of the task associated with the incoming message. We assume again that no inconsistencies appear between the message and task effects - such inconsistencies would only appear in erroneous process designs.

The procedure described above does not satisfactorily deal with loops, but we can perform approximate checking by partial loop unravelling. Some effect scenarios generated using this approach might be infeasible, but we note that our objective is to devise decision-support functionality in the compliance management space, with human analysts vetting key changes before deployment.

DETECTING INTER-PROCESS INTERACTIONS

In this section, we will discuss how the Process-SEER framework described above was extended to address the problem of detecting interactions between co-incident clinical protocols. Two key extensions have been made. First, we introduce a distinction between *mandatory effects* and *potential effects*. Such a distinction is common in clinical settings, for instance, between the (mandatory) intended effects of medication, and (potential) side-effects/complications. Second, we develop the machinery required to be able to detect conflicts between treatment protocols based on semantic effect annotation of clinical process models.

The notion of *scenario labels* (Ghose & Koliadis, 2008) is central to the effect accumulation procedure - these effectively describe the paths taken through a process model to obtain the corresponding *effect scenarios* (as a sequence of gateway and task identifiers). This is important given that it is possible to arrive at a given task in a process model via multiple alternative paths. A scenario label also includes an *exclude set* - these are used to ensure that effects from paths that are mutually exclusive (originating in XOR-splits) are not combined. We extend the notion of scenario label in this work to include the *applicable condition* for a scenario - this is represented as the conjunction of conditions associated with the (labelled) outgoing flows from each OR- or XOR-split gateway preceding the task with which this scenario is associated. In the event that this conjunction is unsatisfiable (this is unlikely, but possible) this condition will be empty (i.e., the logical TRUE value). The notion of an applicable condition associated with an effect scenario helps us identify situations where certain treatment steps are unlikely to logically co-occur in the instance of the same patient, because of contradictory applicable

conditions (e.g., a treatment step that applies only to diabetics vs. one that applies only to non-diabetics).

We assume the existence of a *patient-specific knowledge-base* (P-KB) as well as a *background knowledge-base* (B-KB), but the techniques we present can be of use even when these are empty (as the example in the next section illustrates). The *interaction-checking mechanism* takes as input a set of semantically annotated clinical process models of the form discussed above, and a P-KB. It returns a set of *conflict flags* if a potential conflict is detected between the input processes. In the following, a *conflict* refers to situations where $\{es_1,\ldots,es_n\} \cup P-KB \cup B-KB$ is inconsistent, where each es_i represents an effect scenario obtained from a distinct process that is part of the input. Note that when conflict checking must be restricted to mandatory (resp. potential) effects, we can directly extract these components from an effect scenario (since each assertion in an effect scenario is thus labelled). Conflict flags can be of various kinds:

- *Strong conflicts:* These involve situations where all effect scenarios associated with a task in one process conflict with all the effect scenarios associated with a task in a distinct process.
- *Weak conflicts:* These involve situations where some (but not all) effect scenarios associated with a task in one process conflict with some (but not all) effect scenarios associated with a task in a distinct process.
- *Mandatory-Mandatory (MM) effect conflicts:* These involve situations where the mandatory effects within an effect scenario associated with a task in one process conflict with the mandatory effects in an effect scenario associated with a task in a distinct process.
- *Mandatory-Potential (MP) effect conflicts:* These involve situations where the mandatory effects within an effect scenario associated with a task in one process conflict with the potential effects in an effect scenario associated with a task in a distinct process.
- *Potential-Potential (PP) effect conflicts:* These involve situations where the potential effects within an effect scenario associated with a task in one process conflict with the potential effects in an effect scenario associated with a task in a distinct process.

These categories provide a rich vocabulary for describing conflicts and are not mutually exclusive (we may obtain a strong MP conflict or a weak PP conflict).

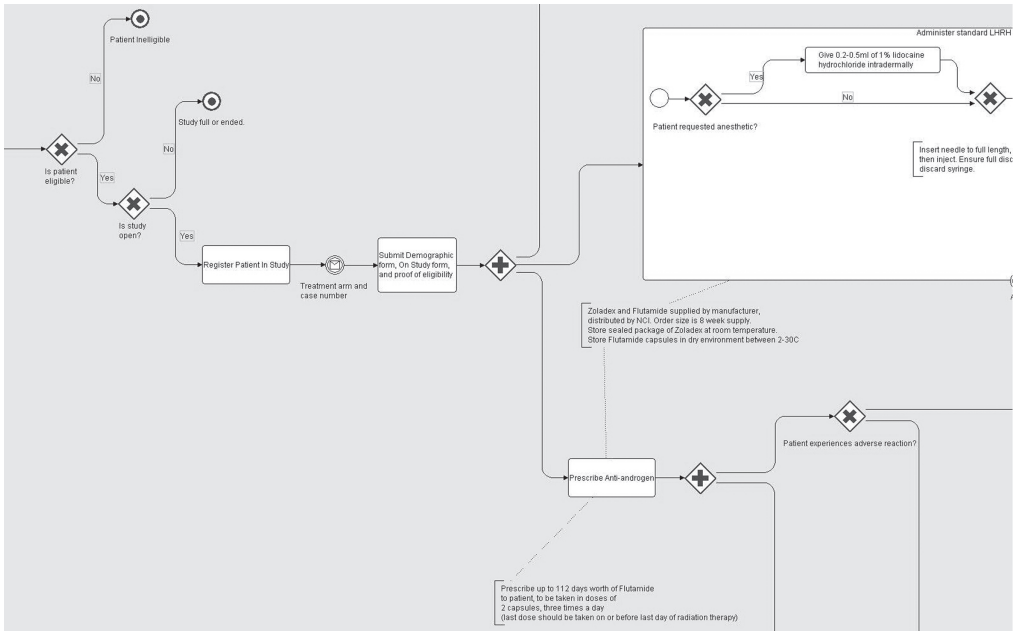
For each kind of conflict, we can further distinguish the severity of the problem by using the *applicable condition* associated with the effect scenario. Conflicts between effect scenarios whose applicable conditions are consistent are likely to be the most severe. Conflicts between effect scenarios whose applicable conditions are inconsistent are less likely, but still possible, given than the notion of an applicable condition is a coarse approximation and conditions associated with gateways might be transient in highly dynamic processes.

In the following section, we provide one substantive example of the detection of treatment conflicts via semantically annotated process models. Even with this setting, the size of the process models, and the effect annotations makes it impossible to display the models in their entirety. The conflict flag obtained in this instance is a *strong MM conflict*. We cannot illustrate the other categories of conflict due to space restrictions, but their should be nevertheless be self-evident.

EXAMPLE

As part of this research program, we have initiated a large-scale exercise in process modelling (in the BPMN notation) of clinical protocols. A total of twelve cancer trial protocols have been modelled in BPMN. The resulting models are large, and after semantic effect annotation, larger still. Figure 2 describes a small portion of a prostate cancer trial protocol modelled in

Figure 1. A section from a prostate treatment BPMN process model showing the prescription of anti-androgen medication



BPMN. The effect annotation of the task 'Adjust dosage/schedule' in Figure 2 proceeds as follows (we provide only the natural language version, and omit the formal version obtained via ontological markup): *If there was evidence of bowel toxicity then administer constipating pain relief. This may contra-indicate medication having diarrhea as a side effect. If there is increasing fatigue in the patient then a full blood count should be requested.*

As part of the treatment of prostate cancer it is common to prescribe anti-androgen medication to reduce androgen levels in the patient's blood. In (Figure 1) the action of prescribing an anti-androgen (Flutamide) is represented by a task/action icon in a BPMN model.

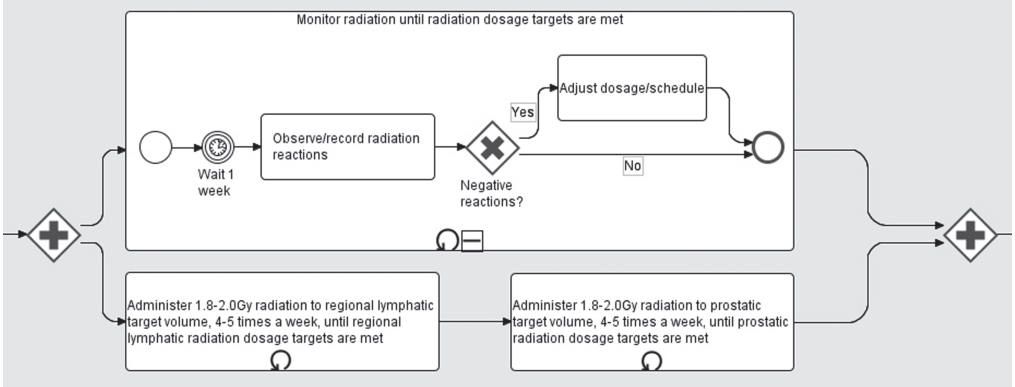
A common step in a renal haemodialysis protocol (full BPMN model omitted here due to space restrictions) is the prescription of androgen-enhancing medication. This would clearly generate a conflict flag using the machinery described in the previous section (note that this would require a formalisation of the effect descriptions, which have been omitted here for brevity).

EVALUATION AND CONCLUSIONS

The ProcessSEER tool, implemented using the Eclipse environment, the STP BPMN modelling tool (*Eclipse SOA Tools Platform, BPMN Modeller*, 2009), the Prover9 theorem prover (McCune, 2009) and the ACE-CNL controlled natural language toolkit (Schwitter & Fuchs, 1996)), provides an adequate basis for semantic effect annotation of clinical process models, based on the experience from modelling and annotating cancer trial protocols described above.

The most compelling motivation for semantic effect annotation of clinical process models is the detection of treatment conflicts, but there are other useful applications for this machinery as well. A BPMN model explains WHAT to do, but rarely HOW to do it (in any significant detail) or WHY an action is being

Figure 2. A section from a prostate treatment BPMN process model showing a sub process for monitoring radiation treatment



performed. Semantic effect annotations added to a process model provide a mechanism for describing the HOW in considerably greater detail. We believe, based on preliminary experience, that this can help identify situations where specialists agree on (the broad picture of) a clinical protocol, but disagree on the detail of its implementation. Semantic effect annotations can help in answering the WHY question, which has pedagogical applications.

REFERENCES

Anzböck, R., & Dustdar, S. (2004, June). Modeling medical e-services. In *Proceedings of Business process management* (Vol. 3080, pp. 49-65). New York: Springer. Retrieved from http://www.infosys.tuwien.ac.at/Staff/sd/papers/ModelingMedicale-servic%es_final.pdf

Coiera, E. (2003, October). Clinical decision support systems. In *The guide to health informatics* (2nd ed., p. 25). Arnold. Retrieved from http://www.coiera.com/aimd.htm

Curia, R., Gallucci, L., & Ruffolo, M. (2005, August). Knowledge management in health care: an architectural framework for clinical process management systems. In *Proceedings of the sixteenth international workshop on database and expert systems applications* (p. 393-397). Washington, DC: IEEE.

Curry, J., McGregor, C., & Tracey, S. (2006, August). A communication tool to improve the patient journey modeling process. In *Proceedings of the 28th ieee embs annual international conference Eclipse soa tools platform, bpmn modeller.* Retrieved from http://www.eclipse.org/bpmn/

Ghose, A., & Koliadis, G. (2007). Business process compliance: Techniques for design-time auditing and resolution. In *Proceedings of the first international workshop on juris-informatics (jurisin07).*

Ghose, A., & Koliadis, G. (2008). Pctk: A toolkit for managing business process compliance. In *Proceedings of the second international workshop on juris-informatics (jurisin'08).*

Hinge, K., Ghose, A., & Koliadis, G. (2009, September). Process SEER: A tool for semantic effect annotation of business process models. In *Proceedings of the thirteenth ieee international edoc conference.* Washington, DC: IEEE.

Hollingsworth, D. (1995, January). *The workflow reference model* (Tech. Rep.). Retrieved from http://www.wfmc.org/standards/docs/tc003v11.pdf

Huth, M., & Ryan, M. (2004). *Logic in computer science: Modelling and reasoning about systems*. Cambridge, UK: Cambridge University Press.

Jackson, D., & Wing, J. (1996). Lightweight formal methods. *IEEE Computer*, 21-22.

Koliadis, G., & Ghose, A. K. (2006, December). Correlating business process and organizational models to manage change. In *Proceedings of the Australasian conference on information systems.*

Koliadis, G., Ghose, A. K., & Padmanabhuni, S. (2008, July). Towards an enterprise business process architecture standard. In *Proceedings of the ieee services congress symposium on soa standards 2008.* Washington, DC: IEEE.

Koliadis, G., Vranesevic, A., Bhuiyan, M., Krishna, A., & Ghose, A. (2006). A combined approach for supporting the business process model lifecycle. In *Proceedings of the 10th pacific asia conference on information systems (pacis'06).*

Lin, Y. (2008). *Semantic annotation for process models: Facilitating process knowledge management via semantic interoperability*. Unpublished doctoral dissertation, Norwegian University of Science and Technology, Trondheim, Norway.

Martin, D., & Domingue, J. (2007). Semantic web services, part 2. *IEEE Intelligent Systems*, 8–15. doi:10.1109/MIS.2007.118

Maximini, K., & Schaaf, M. (2003, June). The progemm approach for managing clinical processes. In *Proceedings of the twelfth ieee international workshops on enabling technologies: Infrastructure for collaborative enterprises* (pp. 332-337). Washington, DC: IEEE.

McCune, W. (2009). *Prover9*. Retrieved from http://www.cs.unm.edu/~mccune/prover9/

McIlraith, S. A., Son, T. C., & Zen, H. (2001). Semantic web services. *IEEE Intelligent Systems*, 46–53. doi:10.1109/5254.920599

OMG. (2006, February). *Business process modeling notation (bpmn) specification*. Retrieved from http://www.bpmn.org/Documents/OMGFinalAdoptedBPMN1-0Spec06-02-01.pdf

OMG. (2009, February). *Omg unified modeling language (omg uml), infrastructure*. Retrieved from http://www.omg.org/spec/UML/2.2/

Panzarasa, S., Maddè, S., Quaglini, S., Pistarini, C., & Stefanelli, M. (2002, April). Evidence-based careflow management systems: the case of post-stroke rehabilitation. *Journal of Biomedical Informatics*, *35*(2), 123–139. doi:10.1016/S1532-0464(02)00505-1

Raut, M., & Singh, A. (2004). Prime implicates of first order formulas. *International Journal of Computer Science and Applications, 1*.

Ruffolo, M., Manna, M., Cozza, V., & Ursino, R. (2007). Semantic clinical process management. In *Proceedings of the twentieth ieee international symposium on computer-based medical systems* (pp. 518-523). Washington, DC: IEEE.

Schwitter, R., & Fuchs, N. (1996, May). Attempto - from specifications in controlled natural language towards executable specifications. In CoRR, cmp-lg/9603004.

Shanahan, M. (1997). *Solving the frame problem - a mathematical investigation of the common sense law of inertia*. Cambridge, MA: MIT Press.

Soffer, P. (2005). Scope analysis: identifying the impact of changes in business process models. *Software Process Improvement and Practice*, *10*(4), 393–402. doi:10.1002/spip.242

Soffer, P., & Wand, Y. (2004, June). *Goal-driven analysis of process model validity* (. *LNCS*, *3084*, 521–535.

Stefanelli, M. (2002). *Careflow management systems.* Retrieved from http://www.openclinical.org/briefingpaperStefanelli.html

Weber, I., Hoffman, J., & Mendling, J. (2008). Semantic business process validation. In *Proceedings of the 3rd international workshop on semantic business process management.*

Kerry Hinge is a graduate student from the School of Information Systems and Technology at the University of Wollongong (UOW) graduating with honours in the field of Internet Science and Technology in 2008 and majoring in internet applications. He is currently undertaking a PhD with the School of Computer Science and Software Engineering at UOW. Kerry has had an extensive career in graphic design for over 25 years and for more than 10 of those years has specialised in internet application and website development. His research is primarily focused around the development of a decision support tool (ProcessSEER) that allows business process

analysts to add semantic effect annotations to process models for the purpose of identifying consequences of process change at design time. The tool is also designed to translate between semantically annotated process models and semantic web services.

Aditya Ghose holds a PhD and MSc degrees in Computing Science from the University of Alberta, Canada and a Bachelor of Engineering degree in Computer Science and Engineering from Jadavpur University, Kolkata, India. While at the University of Alberta, he received the Jeffrey Sampson Memorial Award. His research has been published in the top venues in service-oriented computing (SCC and ICSOC), software modelling (ER), software evolution (IWSSD, IWPSE) and AI (AAAI, AAMAS and ECAI). He has been an invited speaker at the Schloss Dagstuhl Seminar Series in Germany and the Banff International Research Station in Canada. He has also been a keynote speaker at several conferences, and program/general chair of several others. He is a senior technical advisor to several companies in the areas of constraint programming and business process management, both in Australia and Canada.

Andrew Miller is an Australian trained radiation oncologist and a Clinical Associate Professor of Medicine at the University of Wollongong's Graduate School of Medicine, undertaking research in the School of Informatics, the Centre for Medical Radiation Physics, and now also at the Centre for Oncology Informatics. Andrew's domain expert areas are Medical Informatics and Technical & Conformal Radiotherapy, with sub-site clinical specialty in Prostate Cancer, Head & Neck Cancer.