SEMANTIC PROCESS MINING FOR THE VERIFICATION OF MEDICAL RECOMMENDATIONS

M. A. Grando
School of Informatics, Edinburgh University, 10 Crichton street, Edinburgh, U.K.

M. H. Schonenberg, W. van der Aalst
Department of Mathematics and Computer Science, Technische Universiteit Eindhoven, Eindhoven, Netherlands

Keywords: Medical guidelines, Computer interpretable guidelines, Workflow languages, Process mining, Semantic conformance checking, Declarative specification.

Abstract: The dissemination of best medical practices should contribute to a higher quality of care. Because natural language specifications can be ambiguous, their miss interpretation can lead to all kinds of errors. Here we propose a declarative approach for precisely defining medical recommendations. We also propose an approach based on semantic process mining to verify that an arbitrary Computer Interpretable Guideline (CIG) complies with the medical recommendations. Taking into account that some medical recommendations are critical, our work can be seen as a contribution to the design of safer CIGs.

Moreover, we introduce some novel strategies to take full advantage of the information provided by the semantic conformance checker in order to: 1) suggest scenarios than could lead to violations of the medical constraints in the CIG and, 2) estimate how flexible is the CIG with respect to the medical recommendations used as starting point.

1 INTRODUCTION

Medical guidelines are used to disseminate the consensus reached on best medical practices. They are paper-based and expressed in natural language, therefore they can be ambiguously interpreted if the medical evidence provided in the guideline is not carefully considered. Some medical guidelines are detailed enough to suggest possible care paths. For such guidelines a general practice is to provide workflow diagrams that model the care paths explained in the natural language description. The provided workflows are generally specified as networks of tasks and states connected by relational constraints of succession. In (Mulyar et al., 2007) the declarative language CIGDec has been proposed to precisely specify medical recommendations. The CIGDec specification language is supported by DECLARE, a constraint-based Workflow Management System. DECLARE (Pesic and van der Aalst, 2006; van der Aalst and Pesic, 2006) offers a graphical notation for its constraints that have precise semantics in Linear Temporal Logic (LTL). This provides a platform for the verification of constraint-based models. Furthermore, DECLARE supports the execution of constraint-based models. Constraint-based models do not explicitly define the possible execution paths, but rather specify the boundaries of execution, i.e. any execution that does not violate the constraints is allowed. The advantages of using the CIGDdec language for specifying medical recommendations (Mulyar et al., 2007) are (1) its *flexibility*, i.e. many care paths can be specified with relative few constraints, and (2) its *extendability*, i.e. it is possible to define additional constraints specified in LTL to the default language to express situation specific constraints.

Unfortunately, the translation of medical recommendations into Computer Interpretable Guidelines (CIGs) is not automatic and therefore not necessarily error free. However, some languages for the specification of CIGs, like Asbru (Shahar et al., 1998), have been provided with formal techniques based on theorem proving for verifying the satisfaction of properties (Teije et al., 2006) (Gendt et al., 2005). Moreover, the area of business process mining has contributed with techniques to analyze processes based on their execution history (event logs). Process mining can amongst others be applied to automatically

A. Grando M., H. Schonenberg M. and van der Aalst W..
SEMANTIC PROCESS MINING FOR THE VERIFICATION OF MEDICAL RECOMMENDATIONS.
DOI: 10.5220/0003112600050016
In *Proceedings of the International Conference on Health Informatics* (HEALTHINF-2011), pages 5-16
ISBN: 978-989-8425-34-8
Copyright © 2011 SCITEPRESS (Science and Technology Publications, Lda.)

derive process models from the event logs, to check the conformance of the actual execution with a presumed model of the process and to check properties on the execution traces in the log.

More recently more accurate and robust process mining techniques, known as semantic conformance checking, were introduced in (Casati and Shan, 2002) (Kharbili and Stein, 2008) to analyze processes by adding semantics to the event logs. Here our aim is to explore the use of semantic conformance checking for verifying if a CIG satisfies/violates the recommendations from the medical guideline on which it is based.

We propose a novel strategy for a further analysis and interpretation of the results generated during semantic conformance checking. The strategy is based on 1) generating the classes of unexplored scenarios that could expose untested cases of violation of medical recommendations with the DECLARE model of medical recommendations, and 2) reusing the scenarios from (1) exposing modeling decisions that reduce the flexibility of the CIG with respect to the medical recommendations on which it is based.

In Section 2 we take as starting point two natural language recommendations from the chronic cough guideline from (Irwin et al., 1998) and we disambiguate them by considering the medical evidence provided in the guideline. Then in Section 3 we explain how to specify the chosen recommendations as a set of declarative constraints in the DECLARE framework. While the analysis presented in sections 2 and 3 can not be automatized, the methodology presented in sections 4 and 5 has been implemented in the DECLARE tool and the ProM framework. In Section 4 we explain how to use the ProM framework (van Dongen et al., 2005) for checking semantic conformance of a CIG with respect to the DECLARE constraints from Section 3. The CIG used in Section 4 is specified using the PRO*forma* (Fox et al., 1997) language and it has been taken from the Open Clinical repository (http://www.openclinical.org) where other CIGs for the same medical recommendations are available. Every CIG from the repository is specified in a different language: Asbru (Shahar et al., 1998), EON, GLIF and GUIDE. The PRO*forma* CIG has been selected for pragmatic reasons and the explained technique is generic and independent of the language and decision-support system used for the specification of the CIG. In Section 5 we explain some techniques for allowing further analysis of the results obtained from the DECLARE tool and the ProM framework after performing the semantic conformance checking. After mentioning related approaches in Section 6 we finish the conclusions in Section 7.

2 CHRONIC COUGH GUIDELINE RECOMMENDATIONs

In the chronic cough guideline (Irwin et al., 1998) the eligibility criterion is that a patient has a cough that lasts at least 3 weeks. Here we only consider the case of immunocompetent adult patients. The chronic cough guideline combines a strategy of diagnosis with empirical treatment, it guides the physician to make an assumption regarding the most likely cause of cough and start treating it with the intention of confirming the diagnosis by resolving the cough. Therefore, the guideline's main goal is diagnosing chronic cough for inmunocompetent patients and stopping it by treating the most likely cause of cough.

Through out this paper we will consider the following medical recommendations from the chronic cough guideline for inmunocompetent adult patients:

R1) ”chest radiographs should be ordered before any therapy is prescribed in nearly all patients with chronic cough. Chest radiographs do not have to be routinely obtained before beginning treatment for presumed PNDS [post nasal drip syndrome] in young nonsmoker, or in pregnant women, or before observing the result of discontinuation of an ACEI [angiotensin-converting enzyme inhibitor].”

R2) ”When the chest X-ray is normal, PNDS, Asthma, and GERD [Gastroesophageal reflux disease] are the likely causes of chronic cough.”

Figure 1 shows the graphical representation of the recommendations R1) and R2) explained above. This simplified diagram has been extracted from a more extended diagram that provides a workflow-like explanation of the main medical recommendations in the chronic cough medical guideline (Irwin et al., 1998).

The analysis and interpretation of natural language medical recommendations constitutes a time-consuming, non-error free task that requires in most of the cases medical expertise in order to disambiguate statements and to make explicit implicit knowledge hidden in the guidelines. So far this process has to be manually performed. In the rest of this section we summarize our conclusions during the disambiguation of the natural language medical recommendations from the chronic cough guideline after analyzing in detail the provided medical evidence. In the case of the chronic cough guideline the following evidence grading scale has been provided with the guideline's specification, ordered from high to low quality:

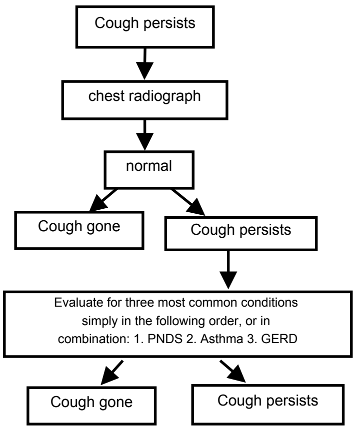


Figure 1: Diagram providing a workflow-based view of the recommendations R1) and R2) from the chronic cough guideline for immunocompetent adult patients (Irwin et al., 1998).

I: Evidence obtained from at least one properly randomized controlled trial.

II: Evidence obtained from well-designed controlled trials without randomization.

II-2: Evidence obtained from well-designed cohort or case-control analytic studies, preferably from more than one center or research group.

II-3: Evidence obtained from multiple time series with or without the intervention.

III: Opinions of respected authorities, based on clinical experience.

We start by explaining our interpretation of recommendations R1) and R2):

R1)(a) **Pregnant Patient or Young Non Smoker with Presumed PNDS:** behind the recommendation of not routinely obtaining chest radiographs for pregnant women there is the implicit knowledge that *the X-ray exposes the embryo to radiation*. It is known that this desirability criterion (value) is intrinsic to any X-ray based plan, although this is not explicitly explained in the chronic cough guideline. Therefore for pregnant women this is a critical recommendation supported by evidence of grade II-2. Behind the recommendation of not routinely obtaining chest radiographs before beginning treatment for presumed PNDS in young nonsmoker, or before observing the result of discontinuation of an ACEI there is medical evidence also of grade II-2 which promotes the value of maximizing the *likelihood of diagnosis*. Behind this recommendation there is also the medical evidence of grade II-2 that for young nonsmokers the probability of PNDS/Asthma/GERD is higher than the average population, therefore it is more *cost-effective* and less time consuming to skip *Chest X-ray*.

R1) (b) **Patients for whom Recommendation R1)(a) does Not Apply (Not Pregnant and Not Young Non Smokers with Presumed PNDS):** therefore for this class of patients obtaining a *Chest X-ray* is strongly recommended based on evidence of grade II-2, promoting the values of maximizing *likelihood of diagnosis* and maximizing *cost-effectiveness* because the X-ray may contain results that can aid in making a correct diagnosis.

R2) For the same reasons the opposite recommendations apply to the treatment of PNDS/Asthma/GERD with the same grade II-2. This plan is recommended without necessarily doing a *Chest X-ray* first, for pregnant women and for young non-smokers with presumed PNDS. Quoting (Irwin et al., 1998) ” PNDS either singly or in combination with other conditions, is the most common cause of chronic cough, followed by asthma and GERD” (grade II-2), therefore the multi-treatment is recommended to any patient to maximize the *likelihood of diagnosis*. The treatment of PNDS/Asthma/GERD should consider the following generic candidate plans:

1) Sequentially treating PNDS, followed by treating asthma and finally treating GERD;

2) Sequentially treating two conditions while concurrently treating the third condition (6 possible combinations);

3) Concurrently treating PNDS, asthma and GERD: quoting (Irwin et al., 1998) ”Properly chosen empiric therapy for PNDS should start to yield a favorable response within 1 week; for asthma within 1 week; for GERD within 1 week to 3 months” (grade III). From this and the previous quote from the chronic cough guideline that indicates that PNDS is the most common cause of chronic cough we can deduce that to minimize the *cost of the treatment* PNDS should not be preceded by the other treatments. The reason is that if it is discovered that the chronic cough is related to PNDS then all the other treatments can be dropped.

Some of the benefits that can be obtained by incorporating value-based decision making in healthcare have been considered in (McCartney, 2005) (Black et al., 2009). Here we suggest simple questions that can help to determine the level of compliance of a medical recommendation based on its attached values:

- Which properties are related to the patient's safeness and therefore are critical to verify?

  For the case of pregnant women the recommendation R1)(a) of not performing a chest X-ray has associated the value of minimizing the *risk of damaging the embryo*. This value is critical and therefore this property should be satisfied in every CIG that models the chronic cough guideline from (Irwin et al., 1998).

- Which properties are not critical but are provided with a high medical evidence and therefore should be mandatorily enforced?

  A good example of this type of criterion are the following recommendations supported by medical evidence of grade II-2:

  1) performing multi-treatment for pregnant women in recommendation R1) (a); 2) performing a chest X-ray and a multi-treatment for asthma/GERD/PNDS for the case of young non smoker with presumed PNDS in recommendation R1) (a) and 3) performing a chest X-ray and a multi-treatment to patients for whom the recommendation R1) (a) does not apply.

- Which properties are related to the efficient use of resources (time, money, medical staff, etcetera) and can mainly suggest optional improvements in the quality of medical treatment?

  This is the case of the recommendation R2) supported with medical evidence of grade III that suggests that the treatment of PNDS should not be preceded by the treatment of Asthma and GERD in order to minimize the *cost of the treatment*.

3 DECLARE MODEL

In this section we present the medical cough guideline (cf. Section 2) expressed in CIGDec (Mulyar et al., 2007) constraints. CIGDec is one of the languages offered by the constraint-based Workflow Management System DECLARE (van der Aalst and Pesic, 2006; Pesic and van der Aalst, 2006) that can be used to define constraint-based models. A language in DECLARE is defined by a set of constraint templates, each template having a name, a graphical representation and an LTL expression. This allows users to interpret DECLARE models without requiring LTL knowledge while having precise semantics at the same time.

Table 1 summarizes the CIGDec constraint templates used in the rest of this paper. From now on we write DECLARE model when we refer to a DECLARE CIGDec model. The state formulas of the LTL expressions are tuples (A,t) where A is the task parameter and t refers to the state of the task, i.e. started (t_s), executed (t_x) or completed (t_c). A constraint between tasks in the model is expressed by associating the parameters of the template for that constraint to the tasks in the model. For example, consider the fifth template from the table. The *response* constraint is used to express that every time activity A executes, activity B has to be executed after it. B does not have to execute straight after A, and another A can be executed between the first A and the subsequent B. To specify that task $multitreatment$ is a response to task *xray*, parameters A and B of the *response* constraint are associated to the tasks *xray* and $multitreatment$ respectively. To avoid task *xray* to be enacted more than once before task *multitreatment* is enacted additional conditional mandatory constraints *exactly1* and *absence2* are specified for task *xray*, each applying to different conditions. The graphical representation of the constraint contains the associated task instead of the parameters as can be seen in Figure 2.

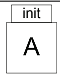
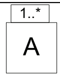
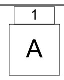
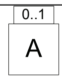
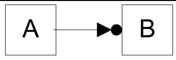
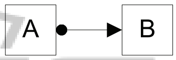
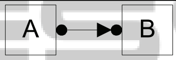
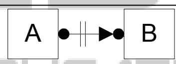
Verification of models is essential to detect modeling errors. DECLARE offers *verification* to check constraint-models for (1) *conflicts*, i.e. constraints that can never be satisfied and (2) *dead activities*, i.e. activities that can never be executed. An error message about the problem and the cause helps the modeler to understand and to resolve the cause of the error.

The *data perspective* of DECLARE specifies how data is handled throughout the execution of the model. Data attributes can be specified and associated to relevant tasks, e.g. the X-ray task reads the pregnancy status of a patient. While executing a task, its data attributes can be read or written, as specified for that task at design-time. DECLARE also offers a *resource-perspective* that specifies which tasks should be executed by whom. An organization of roles and people can be specified and associated with tasks.

Constraints that should only hold under some condition can be specified as *conditional constraints*. A condition is a boolean expression that can involve data attributes from the model. At any point during execution the condition evaluates to true of false, depending on current the value of data attributes. When the condition evaluates to true, the constraint is active, otherwise the constraint is inactive. For instance, the pregnancy status of a patient can be used as condition for constraints that should only be active for this group of patients. In the graphical representation of a conditional constraint the condition is displayed next to the constraint.

By default constraints are *mandatory* (hard) constraints and DECLARE does not allow the execution of tasks that violate one or more active constraint in the model. In DECLARE it is also possible to specify

Table 1: Relevant CIGDec constraints.

Name	LTL expression	Graphical
init(A)	$((A, t_s) \vee (A, t_x))W(A, t_c)$	
existence(A)	$\Diamond(A, t_c)$	
exactly1(A)	$existence(c, A) \wedge !(\Diamond((A, t_c) \wedge \bigcirc(existence(A))))$	
absence2(A)	$!(\Diamond((A, t_c) \wedge \bigcirc(existence(A))))$	
precedence(A,B)	$(!((B, t_s) \vee (B, t_c) \vee (B, t_x)))W(A, t_c)$	
response(A,B)	$\Box((A, t_c) \Rightarrow \Diamond(B, t_c))$	
succession(A,B)	$response(A, B) \wedge precedence(A, B)$	
not succession(A,B)	$not\ response(A, B) \wedge not\ precedence(A, B)$	

optional (soft) constraints that can be violated. Violations of mandatory constraints trigger warnings that are specified for the violation of that constraint, explaining the violation and the consequence of the violation. Graphically, mandatory constraints are depicted as solid lines and optional constraint as dashed lines.

Next, we explain how to model the recommendations from the chronic cough guideline (cf. Section 2) in DECLARE. The model contains the following tasks that correspond to the activities from the cough guideline depicted in Figure 1:

(1) *choice* has no equivalent in Figure 1 because the case of pregnant patient or young non smoker with presumed PNDS has not been modeled in (Irwin et al., 1998) in order to simplify the diagram; we have added it into the DECLARE model to differentiate between the care path recommended for patients with the mentioned medical conditions (R1) (a)) and other patients (R1)(b));

(2) *xray* that corresponds to acquiring an chest X-ray;

(3) *multitreatment* that corresponds to the evaluation of Asthma, GERD and PNDS;

(4) *asthma* that corresponds to the treatment of Asthma;

(5) *pnds* that corresponds to the treatment of PNDS;

(6) *gerd* that corresponds to the treatment of GERD.

For clarification, constraints in Figure 2 are annotated with the corresponding guideline fragments. The following general recommendations should always be satisfied:

RG1) The *choice* of ordering a chest X-ray is made once (constraint exactly1 over task *choice*);

RG2) Performing the treatment of asthma, GERD or PNDS more than once will not change the diagnosis therefore the constraint exactly1 is associated to the tasks *multitreatment*, *asthma*, *GERD* and *PNDS*. This restriction is related to the value of maximizing the *likelihood of diagnosis*.

The general constraints are independent of the patient's medical condition, so these constraints do not have conditions. The following boolean data attributes (in italics) have been introduced to capture the different conditions in the guideline. (1) *cough* is true if the cough is persistent, (2) *normalXray* is true if the result of the X-ray is normal, (3) *pregnant* is true if the patient is pregnant, (4) *young* is true if the patient is less than 18 years old, (5) *smoking* is true if the patient is a smoker, (6) *pnds* is true if PNDS is presumed. These six boolean data attributes are used in the conditions of the next constraints.

The recommendation concerning the X-ray (R1) is captured using mandatory conditional constraints. For the case of pregnant patient or young non smoker with presumed PNDS (R1 (a)) we consider condition *c* which returns true when the patient belongs to the mentioned class. When we consider recommendation R1) (b) we use the negation of *c* denoted as *!(c)*.

R1) (a) **Pregnant Patient or Young Smoker with**

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

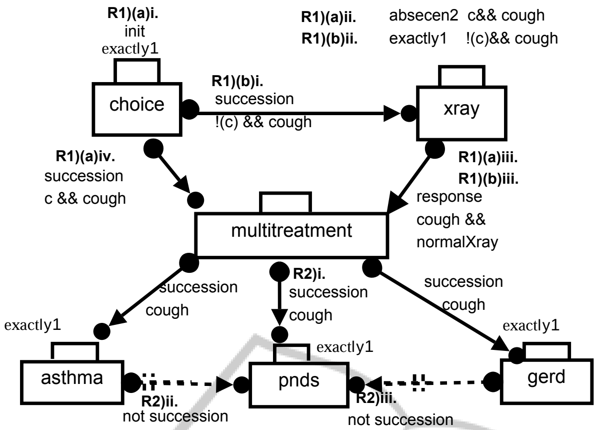


Figure 2: DECLARE model for the considered recommendations from the chronic cough guideline.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

Presumed PNDS: according to the recommendation R1 the care path for this class of patients corresponds to first optionally performing a chest X-ray and then if the cough persists treat for PNDS/Asthma/GERD. Therefore, for patients who are pregnant or are young smokers with presumed PNDS and have chronic cough (c && $cough$) the following care path is recommended:

i. first the decision of performing a *X-ray* has to be taken (init constraint over task *choice*);

ii. performing a *X-ray* is optional (conditional absence2 constraint in task *xray*);

iii. in the case that after the *X-ray* the patient has persistent cough and the result of the *X-ray* is normal ($cough$ && $normalXray$) then eventually the treatment for PNDS/Asthma/GERD has to be started (conditional response relation between tasks *xray* and *multitreatment*);

iv. it is possible to decide not to perform the *Chest X-ray* and to immediately start the treatment of PNDS/Asthma/GERD (conditional succession between tasks *choice* and *multitreatment*). In this case the enactment of *xray* task is not possible anymore because it would violate the precedence constraint between tasks *choice* and *xray*.

R1)(b) **Patients for whom Recommendation R1)(a) does Not Apply:** according to the recommendation R1 if cough persists the correct decision is to perform the *Chest X-ray* and then in case of normal X-ray avoid irritants. If after avoiding irritants the cough persist then treat for PNDS/Asthma/GERD. Therefore for patients with chronic cough who are not pregnant or are not young smokers with presumed PNDS ($!(c)$ && $cough$):

i. the result of the choice is to perform a chest X-ray (conditional succession constraint between tasks *choice* and *xray*).

ii. exactly one chest X-ray has to be done (conditional exactly1 constraint on task *xray*);

iii. similarly to R1) (a) iii. if the result of the X-ray is normal and the patient still has a cough ($cough$ && $normalXray$) then the multitreatment of PNDS/Asthma/GERD has to be started (conditional response constraint between tasks *xray* and *multitreatment*).

R2) Instead of specifying all possible combinations of treatments, for PNDS, asthma and GERD where the treatment of PNDS should not preceded by the other treatments, it is enough to define the preferred scenarios by conditional constraints, that hold in case condition *cough* is true:

i. a mandatory relation of conditional succession between the task *multitreatment* and each of the treatments, where the condition is that the patient still has a cough ($cough$). This constraint has associated the value of maximizing the *likelihood of diagnosis*;

ii. an optional negative relationship of succession between *asthma* treatment and *PNDS* treatment, so before treating *PNDS* the patient cannot be treated for *asthma* and after treating for *asthma* the patient cannot be treated for *PNDS*. This constraint has associated the value of minimizing the *cost of treatment*;

iii. similarly to R2)ii. define an optional negative succession constraint between *GERD* and *PNDS*.

Note that the translation of the preferred treatment plan from recommendation (R2) to a declarative language is straightforward.

4 SEMANTIC MODEL CHECKING

The analysis and interpretation of natural language medical recommendations can not be automated and therefore can be error prone. In Section 2 we disambiguated the recommendations R1) and R2) from the chronic cough guideline based on our interpretation of the medical evidence provided in the guideline. In this section we explain that once the medical recommendations are modeled in a formalism that provides a precise semantic, as CIGDec, it is possible to use the formal model to automatically check if an arbitrary CIG complies with the medical recommendations.

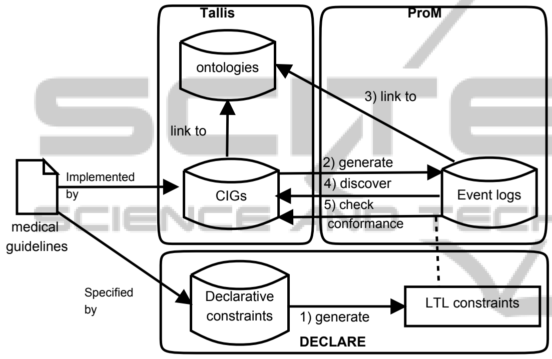


Figure 3: Steps required by the methodology proposed here: 1) Generate LTL constraints from the DECLARE model, 2) Generate logs by enacting the CIG in the Tallis engine, 3) Link ontologies, 4) Discover the model mined from the event logs using ProM, 5) Check conformance using ProM.

The methodology proposed here to check the conformance in a CIG of the DECLARE specification of medical recommendations is illustrated by Figure 3 and explained in detail below:

1) Generate LTL properties from the DECLARE model: the DECLARE tool automatically generates the LTL properties from the constraint model of the medical recommendations explained in Section 3.

2) Enact the CIG to generate event logs: in Mor Peleg et al. work (Peleg et al., 2003) the developers of Asbru (Shahar et al., 1998), GLIF, GUIDE, EON and PRO*forma* (Fox et al., 1997) languages were asked to specify CIGs for similar recommendations to the ones we explained in Section 3. But the developers of the CIGs did not have any access to the chronic cough medical guideline from (Irwin et al., 1998). The recommendations on which the study (Peleg et al., 2003) was based on and the repository of the resulting CIGs is available at the Open Clinical repository (http://www.openclinical.org). None of the CIGs from the repository have been used in any real medical environment. Therefore, we decided to explain the methodology with the PRO*forma* CIG from the Open Clinical repository which has been enacted using the Tallis engine (http://www.cossac.org/tallis). But the strategy explained here requires the execution history (event logs) of the CIG and therefore it is independent of the language used for the specification of the CIG. An event log contains the executions of one or more processes. To construct such log is the required that each event in the log (e.g. an X-ray) can be mapped to a single case or process instance (e.g. a patient treated for cough) and that each process instance can be mapped to a single process (e.g. the process for treating chronical cough). The more information is available in the log, the better the quality of the results and the larger the number of questions that can be answered. For instance, event time stamps can be used to do performance analysis. Similarly, every process instance has zero or more tasks. Every task or audit trail entry must have at least a name and an event type. The event type determines the state of the tasks. Timestamps can be used for capturing timing information and to analyze performance related aspect. The originator element records the person/system that performs the task. Because the Tallis CIG specified for the chronic cough recommendations has not been used in a real medical environment, the event logs have to be generated considering generic patient cases. According to the DECLARE constraints from Section 3 six different patient medical conditions are significant: (1) if the cough is persistent, (2) if the result of the X-ray is normal, (3) if the patient is pregnant, (4) if the patient is less than 18 years old, (5) if the patient is a smoker, (6) if PNDS is presumed. But the enactment of the Tallis CIG showed that the developers of the CIG also considered two additional medical conditions: (7) if the patient has ACE-related cough and (8) if the cough is productive. Therefore firstly generic patient cases were created based on all the possible combinations of the eight mentioned medical conditions. Secondly the generic patient cases were used to enact the Tallis CIG. This is a standard practice in software engineering when system's correctness needs to be tested before the system's release. Finally the events generated during the CIG enactment were recorded as event logs using the MXML (Mining eXtensible Markup Language) format. The schema for the MXML format is available at www.processmining.org. If the considered Tallis CIG would have be running in a real medical context then so-called Process-Aware Information Systems (PAISs) (Dumas et al., 2005) could have been used to automatically generate the corresponding event logs from anonymized real patient's data.

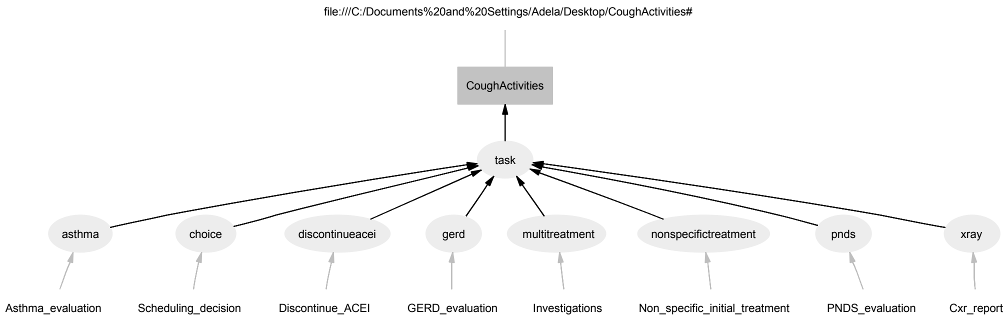


Figure 4: Ontology of activities, where the leafs are the annotated Tallis activities.

3) Link ontologies by semantically annotating the event logs generated by the CIG: this requires mapping the concepts used in the CIG with the concepts from the DECLARE specification. For instance as Figure 4 shows the tasks *Scheduling_decision* and *Cxr_report* from the Tallis implementation presented here can be mapped into the semantically equivalent concepts *choice*, *xray* from the DECLARE model from Section 3.

Only after mapping concepts it is possible to verify if the DECLARE constraints are satisfied in the mined model. For example the DECLARE constraint that specifies that task *choice* has to be enacted exactly once is verified in the model mined from the Tallis implementation as the constraint that the Tallis task *Scheduling_decision* has to be enacted exactly once.

The same DECLARE model used to perform the conformance checking of the Tallis implementation can be reused to check the conformance of any other implementation from the Open Clinical repository on which Mor Peleg et al. work (Peleg et al., 2003) is based on. The only requirement is that previously the corresponding mapping between the DECLARE model and the new implementation has been done.

In particular from the DECLARE model from Section 3 we can create ontologies for data, activities, event types, process instances and originators (actors who enact activities). We have called the ontologies of data and activities *CoughData* and *CoughActivities* respectively. Figure 4 shows the graphical representation of the ontology *CoughActivites*, where the leafs correspond to the annotated activity instances from the Tallis CIG. To perform the ontology linking the MXML file generated in 2) is annotated with concepts from these created ontologies. The resulting semantic annotation is expressed in the SA-MXML (Semantically Annotated MXML) format used by the ProM framework. The SA-MXML format is available at www.processmining.org and it is an extension of the MXML formant where all elements (except for audit trail entries and time stamps) have an optional extra attribute that links to a list of concepts in the ontologies. For instance in the SA-MXML log shown in Figure 5 for the process instance with identifier ”Case0100a” the variables *young* and *old* from the DECLARE ontology *CoughData* are linked by an attribute to the variables *younger* and *older* from the data ontology in the Tallis CIG.

4) Discover the PRO*forma* model from the semantically annotated event logs: using the alpha ++ algorithm that is provided as a plug-in of the ProM framework it is possible to extract (mine) the PRO*forma* model based on the dependency relations that could be inferred among the activities in the Tallis event logs from the SA-MXML file generated in 3). Figure 6 shows the resulting mined PRO*forma* process for generated event logs for the 64 combinations of medical conditions.

5) Perform semantic conformance checking of the discovered PRO*forma* model: using the semantic LTL checker plug-in from the ProM framework it is possible to perform semantic conformance checking of the PRO*forma* semantically annotated model discovered in 4) and the DECLARE model from Section 3.

In Section 3 we explained that it is possible to differentiate in DECLARE between medical recommendations that are mandatory or optional.

Only if a DECLARE constraint is mandatory and it is not satisfied in all the event logs generated from the CIG, a warning should be given during conformance checking. Otherwise if the DECLARE constraint is optional and not satisfied in all the generated event logs it is considered that the CIG complies with the recommendation though it is not optimal. The level of satisfaction of a property can be seen as the percentage of event logs that satisfy the property. This information is provided by the semantic LTL checker as two lists: the first list corresponds to the event logs that satisfy the property and the second list contains the event logs that do not satisfy that property.

In Table 2 we show the results of performing se-

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

```
        <Source program="Rapid Synthesizer" >
                <Data>
                        <Attribute name="program" >Rapid Synthesizer</Attribute>
                </Data>
        </Source>
        <Process id="log2" description=""
modelReference="file:///C:/Documents%20and%20Settings/Adela/Desktop/ProcessOntology.wsml#Proce
ss">
<ProcessInstance id="Case0100a" description=""
modelReference="file:///C:/Documents%20and%20Settings/Adela/Desktop/ProcessInstanceOntology.ws
ml#ProcessInstance" >
<Data>
<Attribute name="Pregnant"
modelReference="file:///C:/Documents%20and%20Settings/Adela/Desktop/CoughData.wsml#Pregnant">
no  </Attribute>
<Attribute name="Smoker"
modelReference="file:///C:/Documents%20and%20Settings/Adela/Desktop/CoughData.wsml#Smoker"> no
</Attribute>
<Attribute name="Younger"
modelReference="file:///C:/Documents%20and%20Settings/Adela/Desktop/CoughData.wsml#Young"> yes
</Attribute>
<Attribute name="Older"
modelReference="file:///C:/Documents%20and%20Settings/Adela/Desktop/CoughData.wsml#Old"> no
</Attribute>
<Attribute name="PressumedPNDS"
modelReference="file:///C:/Documents%20and%20Settings/Adela/Desktop/CoughData.wsml#PressumedPN
DS"> no </Attribute>
</Data>
<AuditTrailEntry>
        <WorkflowModelElement
modelReference="file:///C:/Documents%20and%20Settings/Adela/Desktop/CoughActivities.wsml#choic
e">Scheduling_decision</WorkflowModelElement>
                                <EventType
modelReference="file:///C:/Documents%20and%20Settings/Adela/Desktop/Events_Ontology.wsml#Compl
ete" >Complete</EventType>
                                <Timestamp>2010-04-17T09:00:02.968+00:00</Timestamp>
                                <Originator>UNDEFINED</Originator>
</AuditTrailEntry>
<AuditTrailEntry >
        <WorkflowModelElement
modelReference="file:///C:/Documents%20and%20Settings/Adela/Desktop/CoughActivities.wsml#nonsp
ecifictreatment">Non_specific_initial_treatment</WorkflowModelElement>
                                <EventType
modelReference="file:///C:/Documents%20and%20Settings/Adela/Desktop/Events_Ontology.wsml#Compl
ete" >Complete</EventType>
                                <Timestamp>2010-04-17T09:00:06.968+00:00</Timestamp>
                                <Originator>UNDEFINED</Originator>
</AuditTrailEntry>
        <AuditTrailEntry >
<WorkflowModelElement
modelReference="file:///C:/Documents%20and%20Settings/Adela/Desktop/CoughActivities.wsml#xray"
>Cxr_report</WorkflowModelElement>
                                <EventType
modelReference="file:///C:/Documents%20and%20Settings/Adela/Desktop/Events_Ontology.wsml#Compl
ete" >Complete</EventType>
```


SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

Figure 5: SA-MXML file of annotated event logs from the Tallis CIG for chronic cough.

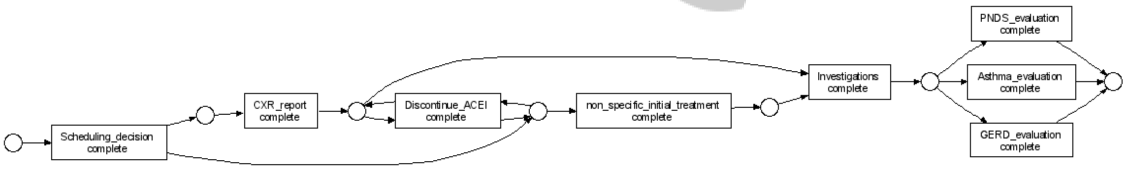


Figure 6: Model of the PRO*forma* CIG discovered by the ProM framework from the Tallis event logs.

mantic conformance checking as explained in Section 4 over the Tallis CIG.

For instance the recommendations R1)(a)ii. and R1)(a)vi. for the case of pregnant women are related to the decision of performing a chest X-ray and they have associated the value of minimizing the *risk of damaging the embryo*. This value is critical and therefore was modeled in Section 4 as mandatory DECLARE constraints. We have shown that all the event logs that we generated with the Tallis CIG satisfy these properties.

We did not find any property that is fully unsatisfied by the generated event logs. For fully unsatisfied properties the semantic LTL model checker returns an empty list of event logs satisfying the property.

For the rest of the properties we showed that they are partially satisfied by the Tallis CIG, it means that some of the generated traces of the event logs are in the first list generated by the semantic LTL checker (traces satisfying the property) and some others are in the second list (traces not satisfying the property). For example this was the case of the non critical mandatory chronic cough medical recommendations R2)i. which specifies that the diagnosis by multi-treatment consists on evaluating for Asthma, GERD and PNDS. Therefore a notification should be given in order to inform that this recommendation has been violated though it was mandatory. From the partial satisfaction of this property and from the observation of the event logs contained in each list computed by the semantic LTL model checker we can infer that the PRO*forma* CIG was designed to ask the user to choose to evaluate only one of the mentioned possible causes of chronic cough.

Table 2: Results of the semantic conformance checking of the medical constraints from the chronic cough guideline in the Tallis CIG. A property can be unsatisfied (-), fully satisfied (+) or partially satisfied (+/-).

Medical recommendation	**DECLARE constraint**	**Natural Language**	**Semantic checking**
RG1) exactly1(choice)	Non critical mandatory	Only once it is possible to decide if a X-ray is performed	+
RG2) exactly1(asthma)	Non critical mandatory	The diagnosis of asthma as cause of cough is evaluated only once	+/-
RG2) exactly1(GERD)	Non critical mandatory	The diagnosis of gerd as cause of cough is evaluated only once	+/-
RG2) exactly1(PNDS)	Non critical mandatory	The diagnosis of PNDS as cause of cough is evaluated only once	+/-
RG2) exactly1(multitreatment)	Non critical mandatory	The diagnosis by multi-treatment is evaluated only once	+
R1)(a)i. init(choice)	Non critical mandatory	Initially for any patient decide if X-ray should be made	+
R1) (a)ii. absence2(xray)	Critical mandatory	For pregnant women or young non smoker with presumed PNDS X-ray is optional	+
R1) (a)iii. response(xray, multitreatment)	Non critical mandatory	After X-ray multi-treatment should be started	+
R1) (a)iv. response(xray, multitreatment)	Non critical mandatory	For pregnant women or young non smoker X-ray can be skipped	+
R 2)i. succession(mutitreatment, asthma) succession(mutitreatment, PNDS) succession(mutitreatment, GERD)	Non critical mandatory	The muti-treatment consists on the treatment of asthma, GERD and PNDS	+/-
R 2) ii. not succession(asthma, PNDS) not succession(GERD, PNDS)	Non critical optional	PNDS can not be preceded by Asthma or GERD	+

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

5 SCENARIO-BASED ANALYSIS

Semantic conformance checking allows to automatically show that certain properties are not satisfied by a CIG. Here we propose a novel methodology to allow further automatic analysis over a CIG once the semantic conformance checking was performed as explained in Section 4. Our methodology takes advantage of: 1) the DECLARE specification of the medical recommendation on which the semantic conformance checking is based on, and 2) results computed by the semantic LTL checker enacted in the ProM framework.

The methodology is based on the scenarios generated by the DECLARE model which can be used to 1) suggest event logs corresponding to unexplored critical behaviors of the CIGs, and 2) analyze if the CIG is more or less restricted than the medical recommendations that were used as starting point.

DECLARE models can be used to suggest significant testing scenarios because they are provided with algorithms to generate: the finite automata that corresponds to the scenarios that do not violate any constraint, and the finite automata that represents all the possible scenarios that violate same of the model constraints (see (Pesic, 2008)).

The information provided by the automaton could be used to:

1. Suggest classes of scenarios that comply with/violate the medical recommendations. For instance these scenarios can be used to generate test cases to verify properties that are satisfied because no event log could be use to check its satisfaction.
2. Provide classes of scenarios that are generated by the DECLARE model and therefore comply with the medical recommendations but can not be generated by the CIG. The generated scenarios can be used to verify if the CIG is more restricted than the medical recommendations on which it is based.

   For instance we can construct the automata that generates all the scenarios which do not violate the constraints defined by the DECLARE model from Figure 2. According to this automata there is a direct transition between the activities *choice* and *multitreatment* for the class of patients corresponding to pregnant women or non-smokers with presumed PNDS. This means that it is possible to chose to start the muti-treatment of asthma/PNDS/GERD without need to perform before a chest X-ray. So in the mined PRO*forma* model (Figure 6) there should be a direct transition between the semantically equivalent activities *Scheduling_decision* and *Investigations*. But this is not the case, the enactment of activity *Scheduling_decision* is only possible if the activity *Cxr_report* has been enacted before. From this analysis we deduce that the mined PRO*forma* model is more restricted that the medical recommendations from the DECLARE model: for the class of pregnant women or non-smokers with presumed PNDS the mined PRO*forma* model obliges to perform a X-ray, instead of offering it as an option.

6 RELATED WORK

In this paper, we focussed on the problem of checking a posteriori whether a CIG satisfies a set of medical recommendations specified in a declarative formal notation. While in (Mans et al., 2009) they have addressed this problem a priori by first specifying the medical recommendations as a Coloured Petri Net (CPN) and then mapping the CPN into CIGs specified in different workflow-based language. It seems that using this mechanism it should be possible to obtain CIGs that are closer to the behavior described by the medical recommendations modeled by the CPN, but there is no claim in (Mans et al., 2009) that the obtained CIG preserves all the behavior from the CPN used as starting point.

An approach close to ours has been proposed in (Stegers et al., 2006) as a strategy to identify and formalize in the Asbru (Shahar et al., 1998) language goals (intentions) identified in clinical guidelines described in natural language. Once the intentions are formalized in the Asbru language they are susceptible of verification. Verification by theorem proving has been proposed in (Teije et al., 2006) (Gendt et al., 2005) to prove that protocols defined in the Asbru language satisfy a set of desirable properties (medical goals and quality indicators) and to find out ambiguities, incompleteness (gasp in the information or insufficient information), inconsistencies or redundancies. By theorem proving it is possible to guaranty the absence of errors, though this strategy is highly costly because it requires to provide a mapping of the language on which the CIG is implement into a theorem prover. So far from the available languages for specification of CIG only the Asbru language provides a mapping to a theorem prover. In contrast an advantage of the semantic conformance checking based approach presented here is that it only requires the process history and therefore it can be applied over any CIG independently of the language used for its implementation. This feature is very important, considering that currently there are numerous incompatible languages available for the specification of medical guidelines (Peleg et al., 2003). Nevertheless, using the LTL constraints that can be generated by DECLARE also conventional model checking could be used without needing event logs.

Semantic process mining techniques have been successfully applied in the area of business process managing (Casati and Shan, 2002) (Kharbili and Stein, 2008) for all kinds of analysis of event logs, like auditing, performance analysis, process discovery, etc. In (Kharbili and Stein, 2008) a theoretical framework (not yet a detailed technical architecture) has been proposed for verifying that a company complies with with external and internal regulations and policies. While more sophisticated than our approach, their work remains as a theoretical proposal. The idea of using semantic process mining techniques to analyze healthcare processes is not new. In (Mans et al., 2008a) (Mans et al., 2008b) they extract relevant event logs from real applications running in hospitals' information systems and they analyze these logs using the ProM framework. Their results show that process mining can be used to provide new insights that facilitate the improvement of existing careflows. Their work differs from us on the fact that they do not use semantic conformance checking to verify if the medical recommendations on which the care process is based are satisfied/violated.

7 CONCLUSIONS

In this paper we propose a technique for checking the compliance of CIGs with respect to medical recommendations specified using a precise declarative language. The technique is based on the execution history of the CIGs and therefore can be applied at post-execution time. Because the Tallis CIG that we used as case study has not been used in a real medical environment we have manually generated the event logs by: 1) creating significant classes of patient cases, 2) enacting the CIG with the classes from (1), 3) recording the resulting events as event logs. But the event logs could be automatically generated from real medical applications by using available PAISs (Dumas et al., 2005). PAIs have been successfully used before for analyzing non-trivial careflow processes, where the patient's real data has been previously anonymised. For instance in (Mans et al., 2008b) they have analyzed a group of 267 gynecological oncology patients treated in 2005 and 2006 in the AMC hospital in Amsterdam, covering diagnostic and treatment activities.

Taking in to account that existing medical recommendations are periodically revised/updated and new medical recommendations can be added, the run-time prevention and detection of violations to medical recommendations and policies is an important problem to consider. Some formal frameworks like (Grando et al., 2010) are available in the field of health informatics for specifying exception managers to detect and recover from undesirable states happening during the enactment of CIGs. But so far no strategy has been proposed to automatically suggest the scenarios or states that violate the medical policies and recommendations. As we explained in Section 5, from DECLARE models it is possible to automatically generate the automaton that describes all the scenarios that violate the model constraints. In the future we are interested on considering the incorporation of the scenario-based information provided by the DECLARE models into exception manager systems like (Grando et al., 2010).

ACKNOWLEDGEMENTS

The authors would like to thank the LOIS initiative at TU/e and the NWO project "MinAdept" for their support.

REFERENCES

Black, E., Glasspool, D., Grando, M. A., Patkar, V., and Fox, J. (2009). Goal-based decisions for dynamic planning. In *AIME*, pages 96–100.

Casati, F. and Shan, M. C. (2002). Semantic analysis of business process executions. In *EDBT*, pages 287–296.

Dumas, M., van der Aalst, W., and ter Hofstede, A. H. (2005). *Process-Aware Information systems: Bridging People and Software Through Process Technology*. Wiley.

Fox, J., Johns, N., Lyons, C., Rahmanzadeh, A., Thomson, R., and Wilson, P. (1997). PROforma: a general technology for clinical decision support systems. *Computer Methods and Programs in Biomedicine*, 54(1-2):59–67.

Gendt, M. V., Teije, A. T., Serban, R., and Harmelen, F. V. (2005). Formalising medical quality indicator to improve guideline. In LNAI, editor, *Proceedings AIME conference*, volume 3581, pages 201–220.

Grando, M. A., Peleg, M., and Glasspool, D. (2010). A goal-oriented framework for specifying clinical guidelines and handling medical errors. *J Biomed Inform*, 43(2):287–299.

Irwin, R. S., Boulet, L. S., Cloutier, M. M., and et al (1998). Managing cough as a defense mechanism and as a symptom, a consensus panel report of the american college of chest physicians. *Chest*, 114(2):133–181.

Kharbili, M. E. and Stein, S. (2008). Policy-based semantic compliance checking for business process management. In *MobIS Workshops*, pages 178–192.

Mans, R., Schonenberg, H., Leonardi, G., Panzarasa, S., Cavallini, A., Quaglini, S., and van der Aalst, W. M. P. (2008a). Process mining techniques: an application to stroke care. In *MIE*, pages 573–578.

Mans, R. S., Schonenberg, H., Song, M., van der Aalst, W. M. P., and Bakker, P. J. M. (2008b). Application of process mining in healthcare - a case study in a dutch hospital. In *BIOSTEC (Selected Papers)*, pages 425–438.

Mans, R. S., van der Aalst, W. M. P., Russell, N. C., Bakker, P. J. M., Moleman, A. J., and et al (2009). From requirements via colored workflow nets to an implementation in several workflow systems. *Transactions on Petri Nets and Other Models of Concurrency*, 3:25–49.

McCartney, J. J. (2005). Values based decision making in helathcare: Introduction. *Journal HEC Forum*, 17(1):1–5.

Mulyar, N., Pesic, M., van der Aalst, W., and Peleg, M. (2007). Declarative and Procedural Approaches for Modelling Clinical Guidelines: Addressing Flexibility Issues. In *Informal Proceedings of the International Workshop ProHealth 2007*, pages 17–28. QUT, Brisbane, Australia.

Peleg, M., Tu, S. W., Bury, J., Ciccarese, P., Fox, J., and et al (2003). Comparing computer-interpretable guideline models: A case-study approach. *J Am Med Inform Assoc*, 10(1):52–68.

Pesic, M. (2008). *Constraint-based workflow management systems: Shifting controls to users*. PhD thesis, Beta Research School for Operations Management and Logistics, Eindhoven.

Pesic, M. and van der Aalst, W. M. P. (2006). A declarative approach for flexible business processes management. In *Business Process Management Workshops*, pages 169–180.

Shahar, Y., Miksch, S., and Johnson, P. (1998). The Asgaard Project: A task-specific framework for the application and critiquing of time-oriented clinical guidelines. *Artificial Intelligence in Medicine*, 14(1-2):29–51.

Stegers, R., Teije, A. T., and van Harmelen, F. (2006). From natural language to formal proof goal, structured goal formalisation applied to medical guidelines. In *Proc Int Conf on Knowledge Engineering and Knowledge Management*, volume 4248, pages 51–58. LNCS.

Teije, A. T., Marcos, M., Blaser, M., and et al (2006). Improving medical protocols by formal methods. *AI in Medicine*.

van der Aalst, W. and Pesic, M. (2006). DecSerFlow: Towards a truly declarative service flow language. In *International Conference on Web Services and Formal Methods 2006*, volume 4184 of *LNCS*, pages 1–23. springer.

van Dongen, B. F., de Medeiros, A. K. A. A., Verbeek, H. M. W., Weijters, A. J. M. M., and van der Aalst, W. M. P. (2005). The ProM framework: a new era in process mining tool support. volume 3536. LNCS.