SEMANTIC PROCESS MINING TOOLS: CORE BUILDING BLOCKS

Alves de Medeiros, Ana Karla and Van der Aalst, Wil, Eindhoven University of Technology, P.O. Box 513, 5600MB, Eindhoven, Netherlands, {a.k.medeiros,w.m.p.v.d.aalst}@tue.nl

Pedrinaci, Carlos, Knowledge Media Institute, The Open University, Milton Keynes, UK, c.pedrinaci@open.ac.uk

Abstract

Process mining aims at discovering new knowledge based on information hidden in event logs. Two important enablers for such analysis are powerful process mining techniques and the omnipresence of event logs in today's information systems. Most information systems supporting (structured) business processes (e.g. ERP, CRM, and workflow systems) record events in some form (e.g. transaction logs, audit trails, and database tables). Process mining techniques use event logs for all kinds of analysis, e.g., auditing, performance analysis, process discovery, etc. Although current process mining techniques/tools are quite mature, the analysis they support is somewhat limited because it is purely based on labels *in logs. This means that these techniques cannot benefit from the actual* semantics *behind these labels which could cater for more accurate and robust analysis techniques. Existing analysis techniques are purely syntax oriented, i.e., much time is spent on filtering, translating, interpreting, and modifying event logs given a particular question. This paper presents the core building blocks necessary to enable semantic process mining techniques/tools. Although the approach is highly generic, we focus on a particular process mining technique and show how this technique can be extended and implemented in the ProM framework tool.*

Keywords: Semantic Process Mining, Semantics-Supported Business Intelligence, Semantic Business Process Management, Semantic Auditing.

1 INTRODUCTION

Nowadays companies usually have some information system to support the execution of their business processes. Common examples are ERP, CRM or Workflow systems. These information systems typically support the creation of event logs that register what happens within companies while executing business process. These event logs normally have data about which tasks have been executed for a given process instance, the order in which these tasks have been performed, by whom and at which times. Additionally, some logs also show which data fields were modified by these tasks. Process mining targets the automatic discovery of information from event logs (cf. Figure 1). The discovered information is used to analyze how the systems that generate these logs are actually being used.

Techniques provided by current process mining approaches can be classified into three perspectives: *discovery*, *conformance* and *extension* (cf. Figure 1). The techniques that focus on *discovery* mine information based on data in an event log only. This means that these techniques do not assume the existence of pre-defined models to describe some aspect of processes in the organization. Examples of such techniques are *control-flow mining* algorithms (Aalst et al. 2004, Greco et al. 2006) that extract a process model based on the dependency relations that can be inferred among the tasks in the log. The algorithms for *conformance* verify if logs follow *prescribed* behaviors or rules. Therefore, besides a log, such algorithms also receive as input a model that captures the desired property or behavior to check. An example of such algorithms is the one used for auditing of logs (in this case, the model is the property to be verified) (Aalst et al. 2005). The *extension* algorithms enhance existing models based on information discovered from event logs. For instance, algorithms that automatically discover business rules for the choices in a given model (Rozinat et al. 2006).

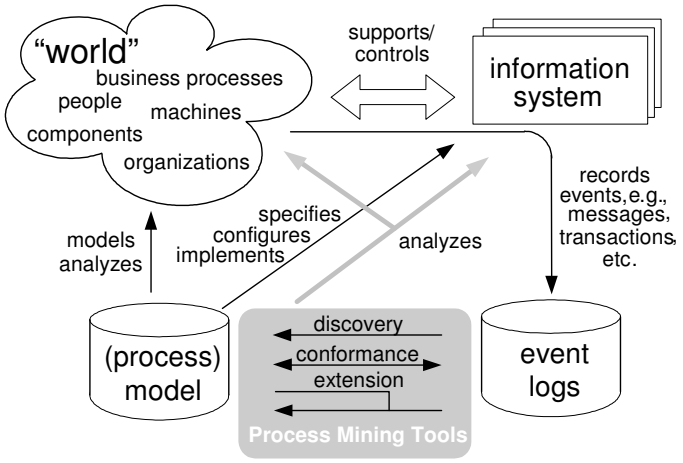

Figure 1: Sources of information for process mining techniques. The discovery *plug-ins use only an event log as input, while the* conformance *and* extension *techniques also need a (process) model as input.*

Current discovery, conformance and extension process mining techniques are already quite powerful and mature. *However, the analysis they provide is purely syntactic.* Note that event logs contain activity names. However, these activity names are strings that typically do not have any semantics attached to them. We have encountered logs from multinationals where depending on the country involved different names were used for the same activity. Moreover, some activities can be seen as special cases of other activities. From the viewpoint of existing process mining techniques, all of these activities are different and unrelated. This example illustrates that these mining techniques *are unable*

to reason over the concepts behind the labels in the log, thus the actual semantics behind these labels remain in the head of the business analyst who has to interpret them. A natural step in the evolution of process mining research is the integration of semantic processing capabilities, leading to what we refer as *semantic process mining*. Leveraging process mining to the conceptual layer can enhance state-of-the-art techniques towards more advanced, adaptable and reusable solutions which can be more easily grasped by business analysts. This is in fact inline with the recent trend in making use of semantics within BPM (Casati et al. 2002, Grigori et al. 2004, Hepp et al. 2005, O'Riain et al. 2006, Sell et al. 2005)[1]. Actually, the European project SUPER (Su06) (within which the work presented in this paper is being developed) "aims at providing a semantic-based and context-aware framework, based on Semantic Web Services technology that acquires, organizes, shares and uses the knowledge embedded in business processes within existing IT systems and software, and within employees' heads, in order to make companies more adaptive". This semantic framework will support the four phases of the BPM life-cycle (Modeling, Deployment, Execution and Analysis). In this context, process mining techniques are been developed to provide for *semantic analysis* of the business processes. *This paper discusses the core elements necessary to perform semantic process mining and illustrates how these elements have been used to extend ProM's LTL Checker (Aalst et al. 2005) to perform semantic auditing of logs.* This new type of semantic analysis is available via www.processmining.org. Moreover, the ideas presented in this paper are not limited to the LTL Checker and can be applied to most other types of process mining.

The remainder of this paper is organized as follows. Section 2 introduces a running example based on our experiences with the SUPER use case partners, which we use for illustration purposes throughout this paper. Section 3 presents how current process mining techniques can be used to check properties in an event log for the running example. Section 4 explains the core elements necessary to implement semantic process mining tools and how current process mining techniques could make use of them. Section 5 describes a concrete semantic process mining algorithm that has been developed based on the approach explained in this paper. Section 6 reviews the related work. Section 7 concludes this paper and points out future directions in semantic process mining.

2 RUNNING EXAMPLE

The running example is based on a real-life scenario taken from the SUPER project. A telephony company - which we will refer to as TelCom - that provides VoIP telephony to small and medium size enterprises (SME). TelCom acts as an intermediary company that connects the SMEs to the big VoIP providers. Its main business is to find out which providers are suitable for fulfilling SME's Quality of Service (QoS) requirements. Finding the appropriate providers is crucial to TelCom's business since the added-value they provide is that they guarantee the required QoS. To support its business TelCom has several processes that allow to create new VoIP accounts to SMEs, update existing ones or close them. In this paper we will focus on the *process to update VoIP accounts of existing customers*. This process starts when a customer informs TelCom that it needs an account update. After receiving this notification, TelCom sends a form where the customer can specify the desired service characteristics. Once TelCom gets back the form filled in by the customer, it checks whether the request fits into one of the pre-defined bundles or it prepares a customized bundle to the customer. TelCom currently contemplates two pre-defined bundles: *Silver* and *Gold*. The *Silver* bundle fits situations where the QoS requested is lower than or equal to 80% (i.e. QoS $\leq$ 80%). The G*old* bundle guarantees a QoS requirement between 80% and 90% (i.e., 80% < QoS $\leq$ 90%). Both bundles have a predefined set of suitable providers, one of which is automatically selected on the basis of the desired QoS. Whenever a customer has a request for which the corresponding QoS is higher than 90%, a new provider (not in the list of pre-defined ones) has to be selected. If such provider cannot be found, the request is aborted

[1] Section 6 provides more details on these works.

and the customer's account is not updated. If a provider is found, then the offer has to be approved by one of the directors in the company before this offer is made to the customer. Furthermore, if the QoS is higher than 96%, this approval has to be given by the CEO of TelCom. The reason is that TelCom reimburses customers whenever the provided services do not meet the requirements, and such high QoS rates are more difficult to meet. If the director does not approve the offer (possibly because it would be too risky for TelCom), the request is archived and the customer's account is not updated. Once a provider is selected (and the necessary authorizations are in place), a contract is sent to the customer with the new specifications. When the customer returns the signed contract, its account is updated and this update is confirmed.

3 PROCESS MINING IN PRACTICE

As explained before, process mining provides objective feedback about actual process executions (registered in event logs). In this section we illustrate how process mining could be used to analyze the TelCom process (cf. Section 2). We have chosen to focus on a particular type of process mining: conformance checking based on LTL. However, we would like to stress that the ideas presented in this paper are generic and can be applied to other types of process mining. For example, semantic annotations could be used to discover high-quality process models, organizational models, simulation models, etc.

Process ID	Task Name	Event Type	Originator	Timestamp	Extra Data
1	Start Request	Completed	Anne	20-07-2006 14:00:00	customerID = 1
1	Send Form	Completed	Anne	20-07-2006 15:05:00	…
1	Receive Form	Completed	John	24-07-2006 10:05:00	QoS = 74%
3	Start Request	Completed	Mary	20-07-2006 15:00:00	customerID = 30
5	Start Request	Completed	Anne	18-06-2006 12:30:00	customerID = 31
5	Send Form	Completed	Anne	18-06-2006 16:00:00	…
3	Send Form	Completed	Mary	22-07-2006 15:30:00	…
1	Silver	Completed	Arthur	14-07-2006 11:05:00	providerID = 45
1	Send New Contract	Completed	Rose	25-07-2006 14:05:00	…
2	Start Request	Completed	John	20-07-2006 17:00:00	customerID = 1025
2	Send Form	Completed	John	21-07-2006 09:00:00	…
1	Receive Contract	Completed	Rose	18-08-2006 14:05:00	…
1	Update Account	Completed	Paul	25-08-2006 16:00:00	…
1	Confirm Request	Completed	System	25-08-2006 16:15:00	…
2	Receive Form	Completed	Mary	25-07-2006 10:05:00	QoS = 98%
2	Custom	Completed	Laura	25-07-2006 11:05:00	providerID = null
3	Receive Form	Completed	Anne	24-07-2006 10:05:00	QoS = 85%
5	Receive Form	Completed	Mary	28-06-2006 12:05:00	QoS = 95%
5	Custom	Completed	Paul	15-07-2006 17:15:00	providerID = 350
3	Gold	Completed	Laura	25-07-2006 08:05:00	providerID = 100
4	Start Request	Completed	John	30-10-2006 08:30:00	customerID = 105
3	Send New Contract	Completed	Marc	28-07-2006 14:05:00	…
5	Get Approval	Completed	Jack	17-07-2006 17:15:00	approved = true
5	Send New Contract	Completed	Marc	25-07-2006 10:05:00	…
3	Receive Contract	Completed	Rose	26-08-2006 09:00:00	…
4	Send Form	Completed	John	01-11-2006 09:00:00	…
4	Receive Form	Completed	Mary	15-11-2006 10:05:00	QoS = 98%
4	Custom	Completed	Arthur	15-11-2006 17:15:00	providerID = 205
5	Receive Contract	Complete	Marc	02-08-2006 14:05:00	…
5	Update Account	Completed	Laura	05-08-2006 10:15:00	…
3	Update Account	Completed	Arthur	26-08-2006 16:00:00	…
3	Confirm Request	Completed	System	26-08-2006 17:30:00	…
2	Abort Request	Completed	System	25-08-2006 16:15:00	…
5	Confirm Request	Completed	System	05-08-2006 17:15:00	…
4	Get Approval	Completed	Patrick	17-11-2006 17:15:00	approved = false
4	Abort Request	Completed	System	18-12-2006 09:00:00	…

Table 1: Example of an event log for the running example introduced in Section 2.

Based on the description of the process in Section 2, three possible analysis questions are: *(Q1) How many requests involve **pre-defined** bundles?; (Q2) How many requests involve **customized** bundles?; and (Q3) Is the rule that "all **confirmed** requests for **custom** bundles have been **checked** by a **director**" indeed being obeyed?*. Current process mining techniques can be used to answer these questions. Actually, all these questions can be answered by conformance algorithms like the LTL Checker (Aalst et al. 2005). However, because the analysis provided by current process mining algorithms is purely syntactic, the end user has to apply her domain knowledge in order to translate the concepts used to formulate this general analysis questions to the *actual labels* contained in the execution log. This is obviously not desirable since it is not realistic nor is it reasonable to expect or require business analysts to go down to such a fine-grained level of detail. For instance, let us consider the log in Table 1 which contains the execution of five process instances (cf. column "Process ID") of the TelCom process to update customer's accounts. For every instance, it is possible to see which tasks were executed, by whom and at which times (cf. the respective columns "Task Name", "Originator", and "Timestamp"). Additionally, it is possible to know at which state a certain task was by analysing the kind of event generated (cf. column "Event Type") and the data fields involved in the execution of this task (cf. column "Extra Data"). For instance, by inspecting the log, one could see that the process instance 4 illustrates the situation in which a request for a customized bundle was rejected by "Patrick". For this event log, the previous analysis questions *Q1*, *Q2* and *Q3* translate to: *(Q1') How many requests involve **Silver or Gold** bundles?; (Q2') How many requests involve **Custom** bundles?; and (Q3') Is that true that "whenever the task **Custom** and the task **Confirm Request** are executed in a process instance, the task **Get Approval** is also executed by **Jack or Patrick**"?*[2]. Note that the *use of actual labels* in these analysis questions makes things over-specific and unnecessarily detailed, and, therefore, hinders their re-use and intelligibility. For instance, think of situations in which a process is re-designed. In this case, any change in the task labels or addition of tasks requires an update of the analysis questions. For instance, if a new pre-defined *Bronze* bundle is included, the question *Q1'* needs to be updated to also include this bundle. It is not difficult to imagine the problems that could arise when dealing with domains characterised by their large size, their complexity or their constant evolution. In order to effectively support this we need to leverage mining techniques to the conceptual level where automated reasoning techniques can be applied. The next section explains the approach we propose to capture this conceptual view into process mining techniques.

4 SEMANTIC PROCESS MINING

The aim of semantic process mining is to make use of the semantics of the data captured in event logs to, on the one hand, create new techniques or enhance existing ones to better support humans in obtaining more detailed and accurate results, and on the other hand, to provide results at the conceptual level so that they can more easily be grasped by business analysts. To cater for this our approach is based on three basic building blocks: *ontologies*, *references from elements in logs/models to concepts in ontologies* and *ontology reasoners* (cf. Figure 2). *Ontologies* (Gruber 1993) define the set of shared concepts necessary for the analysis, and formalize their relationships and properties. We consider in this concern both generic ontologies, e.g., TOVE (Fox et al. 1998), and domain specific ones. The *references* associate meanings to labels (i.e., strings) in event logs or models by pointing to concepts defined in ontologies. The *reasoner* supports reasoning over the ontologies in order to derive new knowledge, e.g., subsumption, equivalence, etc. In a nutshell, our approach consists in feeding the semantic process mining algorithms with: (i) logs/models which elements have references to concepts in ontologies; and (ii) reasoners that can be invoked to reason over the ontologies used in these logs/models. Note that the link to concepts in ontologies and the use of reasoners allows for the development of process mining algorithms with more robust analysis of business processes.

[2] The answers for these questions are (cf. Table 1): (*Q1'*) Two process instances (1 and 3); (*Q2'*) Three process instances (2, 4 and 5); and (*Q3'*) Yes, it is true.

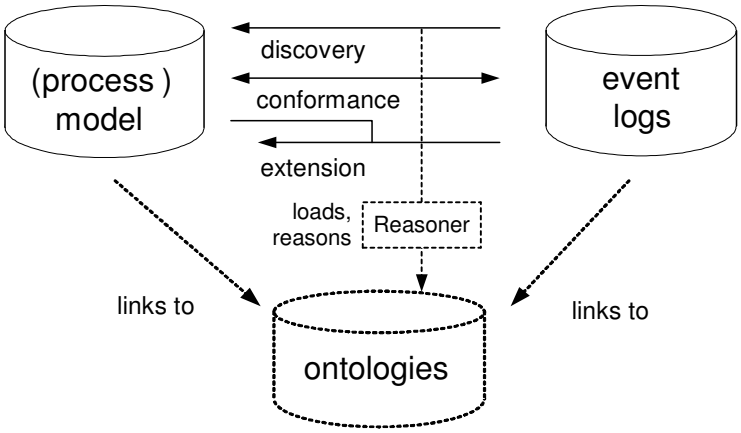

Figure 2: Basic building blocks to support the development of semantic process mining algorithms.

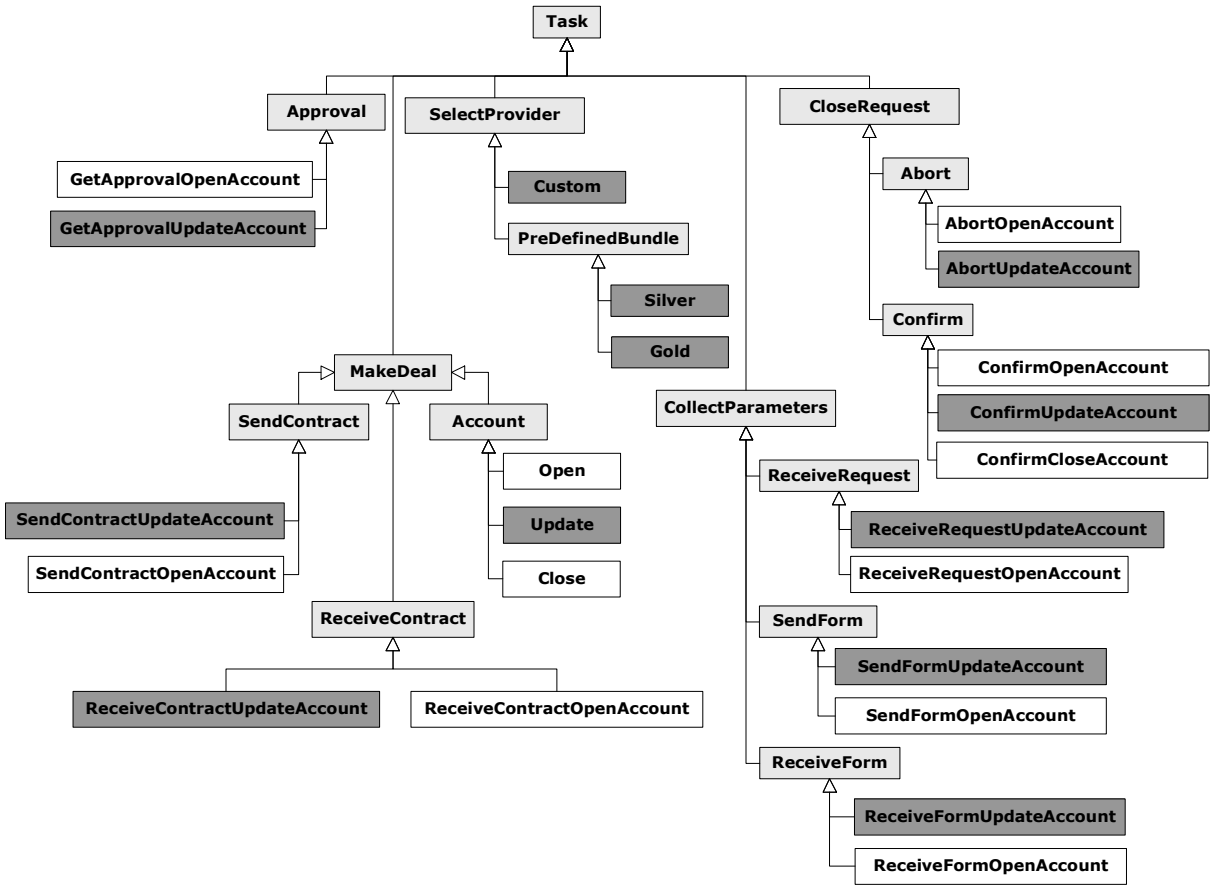

Figure 3: TelComActivities Ontology as a UML Class Diagram. The highlights show the projected view of this ontology based on the relations in Table 2. The concepts in dark grey are directly linked to task names in the log in Table 1. The concepts in light grey are superconcepts of the directly linked concepts.

As an illustration, consider the analysis questions *Q1*, *Q2* and *Q3* in Section 3. These questions are based on concepts that link to tasks and performers of these tasks. Actually, the ontologies depicted in figures 3 and 4 can be used to respectively formalize the concepts for tasks and originators. Additionally, based on these ontologies and the event log in Table 1, the relations in tables 2 and 3 can be determined. Note that elements in the log can link to one or more elements in ontologies. For

instance, most of the originators in Table 3 are associated to two concepts. Provided these references, semantic process mining techniques could use reasoners to identify the concepts that are directly mapped to labels in logs/models (cf. elements in dark grey in figures 3 and 4) and their superconcepts (cf. elements in light grey). It is important to identify the superconcepts because they provide for a higher abstraction level. For example, remark that, based on these concepts, the three questions *Q1*, *Q2* and *Q3* in Section 3 could be defined as: *(Q1") How many requests involve* ***TelComActivityOntology#PreDefinedBundle***[3] *bundles?; (Q2") How many requests involve* ***TelComActivityOntology#Custom*** *bundles?; (Q3") Is that true that "whenever the task* ***TelComActivityOntology#Custom*** *and the task* ***TelComActivityOntology#ConfirmUpdateAccount*** *are executed in a process instance, the task* ***TelComActivityOntology#GetApprovalUpdateAccount*** *is also executed by* ***TelComOrganizationalOntology#Director****"?* Note that these questions are defined in terms of concepts mapped to elements in the log. Actually, although the answers for these questions are exactly the same as for questions *Q1'*, *Q2'* and *Q3'* in Section 3, the approach to find these answers is different. In this case, semantic process mining techniques would use the *ontologies*, the *reasoner,* and the *provided references* to discover the labels that bind to the concepts used in these questions. For instance, consider the first question *Q1*". This question uses the concept *PreDefinedBundle* in the ontology *TelComActivity* ontology. By using the reasoner, it is possible to infer that all process instances with labels referring to any of the concepts *PreDefinedBundle*, *Silver* and *Gold* refer to a pre-defined bundle request. Based on the references in Table 2, these labels are *Silver* and *Gold*.

Task Name	Concepts
Start Request	ReceiveRequestUpdateAccount
Send Form	SendFormUpdateAccount
ReceiveForm	ReceiveFormUpdateAccount
Silver	Silver
Gold	Gold
Custom	Custom
Get Approval	GetApprovalUpdateAccount
Send New Contract	SendContractUpdateAccount
Receive Contract	ReceiveContractUpdateAccount
Update Account	Update
Confirm Request	ConfirmUpdateAccount
Abort Request	AbortUpdateAccount

Table 2: Model references from the elements in the column "Task Name" in Table 1 to the concepts in the "TelComActivity Ontology" in Figure 3.

Originator	Concepts
Anne	SalesPerson, SalesDepartment
Mary	SalesPerson, SalesDepartment
John	SalesPerson, SalesDepartment
Arthur	Engineer, NetworkOperationalCentre
Laura	Engineer, NetworkOperationalCentre
Paul	Engineer, NetworkOperationalCentre
Jack	Director, TechnicalDepartment
Patric	CEO
Rose	Lawyer, ContractManagementDepartment
Marc	Lawyer, ContractManagementDepartment
System	-

Table 3: Model references from the elements in the column "Originator" in Table 1 to the concepts in the "TelComOrganization Ontology" in Figure 4.

The use of ontologies, model references, and a reasoner makes it possible to define more general analysis questions and automatically find the answer for these questions. Furthermore, because the analysis is performed at the conceptual level, it is closer to human understanding, and the addition of new elements in the ontologies or changes of the labels does not necessarily require updating the analysis questions. For instance, for *Q1"*, one could easily include more pre-defined bundles, e.g., bronze and best-effort, without requiring updating the question. This brings much more flexibility to the whole analysis process. The next section shows a concrete implementation that makes use of these core building elements. The next section shows a concrete implementation that makes use of these core building elements.

[3] In this paper, we use the notation *ontology_name#ontology_concept* while referring to a concept in a certain ontology.

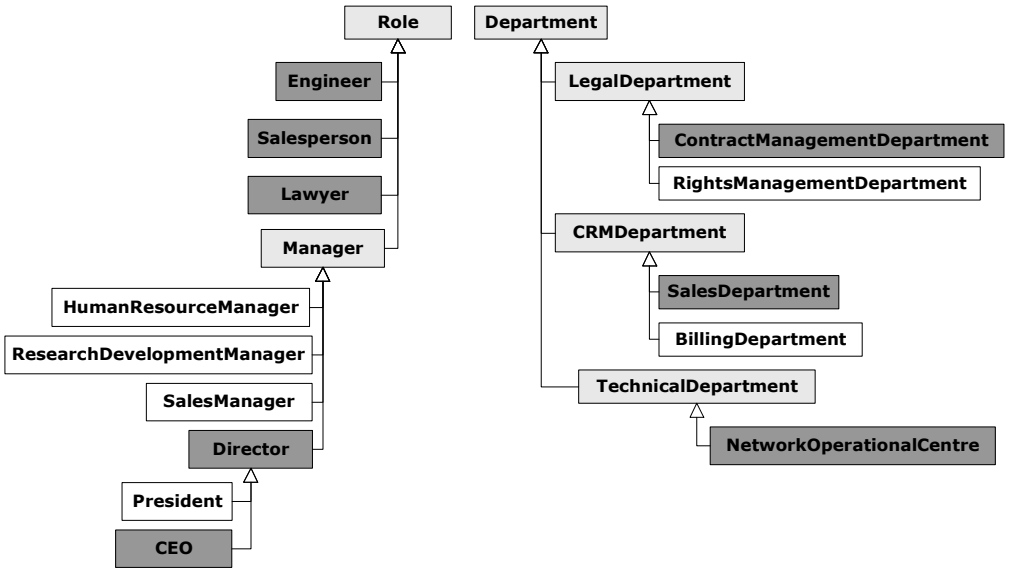

Figure 4: TelComOrganization Ontology as a UML Class Diagram. The highlights show the projected view of this ontology based on the model references in Table 3.

5 CONCRETE IMPLEMENTATION

The approach described in Section 4 has been used to develop semantic process mining plug-ins in the ProM framework tool. ProM is the only open-source framework (Dongen et al. 2005) supporting the development of process mining algorithms. The ProM framework is available via www.processmining.org and it is currently being used by many research groups working in the process mining field. In order to support using semantic information within this framework, we have modified it in the following way: (i) its input format has been extended to support semantic annotations, paving the way for further development of semantic process mining techniques in this tool. This format is explained in Subsection 5.1; (ii) it has been integrated with the WSML2Reasoner framework (W2RF). This reasoner has been chosen because our work is part of the SUPER European project, in which ontologies are defined in the WSML (Lausen et al. 2005) . However, our approach is completely independent from the ontology language and reasoner used, although they obviously determine the level of reasoning we can benefit from within our mining algorithms. Based on these extensions, a semantic version of the conformance analysis plug-in LTL Checker (Aalst et al. 2005) has been developed. This plug-in is explained in Subsection 5.2.

5.1 SA-MXML

The Semantically Annotated Mining eXtensible Markup Language (SA-MXML) format is a *semantic annotated version* of the MXML format used by the ProM framework. In short, the SA-MXML incorporates the *model references* (between elements in logs and concepts in ontologies) that are necessary to implement our approach. However, before explaining the SA-MXML, let us first briefly introduce the MXML format.

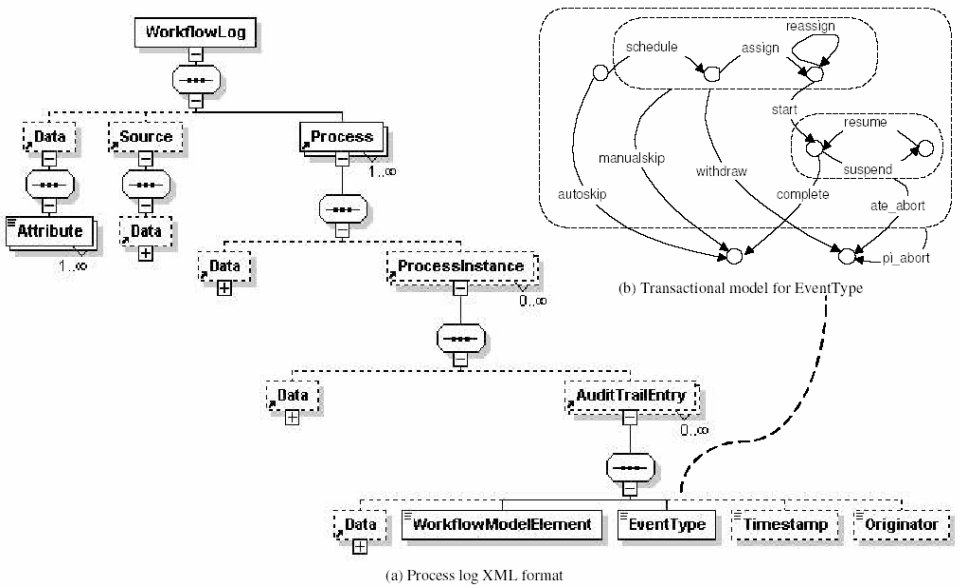

Figure 5: The visual description of the schema for the Mining XML (MXML) format.

The Mining XML format (MXML) started as an initiative to share a common input format among different mining tools (Aalst et al. 2003). This way, event logs could be shared among different mining tools. The schema for the MXML format (depicted in Figure 5) is available at is.tm.tue.nl/research/processmining/WorkflowLog.xsd. As can be seen in Figure 5, an event log (element *WorkflowLog*) contains the execution of one or more processes (element *Process*), and optional information about the source program that generated the log (element *Source*) and additional data elements (element *Data*). Every process (element *Process*) has zero or more cases or process instances (element *ProcessInstance*). Similarly, every process instance has zero or more tasks (element *AuditTrailEntry*). Every task or audit trail entry must have *at least* a name (element *WorkflowModelElement*) and an event type (element *EventType*). The event type determines the state of the tasks. There are 13 supported event types: schedule, assign, reassign, start, resume, suspend, autoskip, manualskip, withdraw, complete, ate_abort, pi_abort and unknown. The other task elements are optional. The *Timestamp* element supports the logging of time for the task. The *Originator* element records the person/system that performed the task. The *Data* element allows for the logging of additional information.

The SA-MXML format is an extension of the MXML format whereby *all elements (except for AuditTrailEntry and Timestamp) have an optional extra attribute* called *modelReference*. This attribute links to a *list of concepts* in ontologies and, therefore, supports the necessary *model references* for our approach. The concepts are expressed as URIs and the elements in the list are separated by blank spaces. Actually, the use of *modelReference* in the SA-MXML format is based on the work for the semantic annotations provided by SAWSDL (Semantically Annotated Web Service Definition Language) (Sa06). The schema for the SA-MXML format is available at is.tm.tue.nl/research/processmining/SAMXML.xsd. The SA-MXML provides the necessary support to capture the correspondence between labels in logs and concepts in ontologies. Furthermore, because the SA-MXML format is *backwards compatible* with MXML format, process mining techniques that do not support semantic annotations yet can also be directly applied to SA-MXML logs.

5.2 Semantic LTL Checker

To illustrate how our approach supports the development of semantic process mining algorithms, we have extended the existing LTL Checker (Aalst et al. 2005) analysis plug-in in ProM to exploit semantic annotations. The LTL Checker can be used to verify properties defined in terms of Linear

Temporal Logic (LTL). This tool is especially useful when auditing logs. The original LTL Checker works only over labels in the log. In other words, setting values for the parameters in the LTL Checker interface is similar to the translation shown from the questions *Q1*, *Q2* and *Q3* to the questions *Q1'*, *Q2'* and *Q3'* in Section 3. The *Semantic LTL Checker*[4] we have developed extends the original LTL Checker by adding the option to provide concepts as input to the parameters of LTL formulae. This way, questions like *Q1"*, *Q2"* and *Q3"* (cf. Section 4) defined at the conceptual level can be answered. Actually, the settings to answer *Q3"* are shown in Figure 6. Note that the parameters "A", "B", "C" and "D" in the formula "activity_A_and_activity_B_implies_activity_C_performedBy_D" can be set to actual labels (option "Instance") or to concepts (option "Ontology"), as shown in the highlighted area in Figure 6. In the latter situation, the user can specify if the subsumption relations should also be used. For instance, for the parameter "D" we have set that the tool should consider elements of the concepts *Director* or any of its subconcepts. In this case, the *Semantic LTL Checker* will consider the process instances that contain links to the concepts *Director* and *CEO* (cf. Figure 4 and Table 3). Behind the scenes this plug-in is using the WSML2Reasoner to infer all the necessary subsumption relations about these concepts.

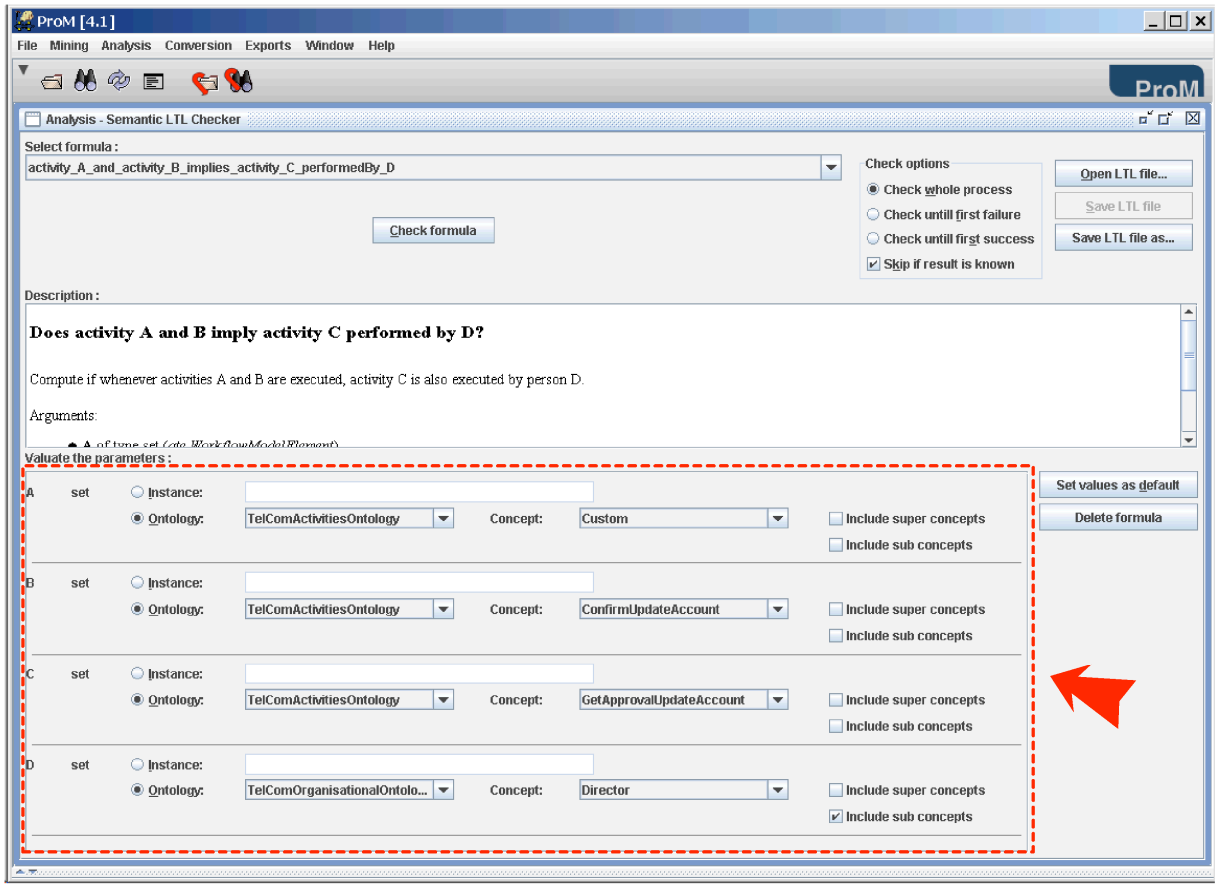

Figure 6: Screenshot of the main interface of the Semantic LTL Checker plug-in.

[4] All the logs, ontologies and LTL formulae used in this section are available at is.tm.tue.nl/research/processmining/TelCom.zip. The semantic LTL Checker plug-in can be started by clicking the menu option "Analysis->Semantic LTL Checker" in the ProM tool.

6 RELATED WORK

The idea of using semantics to perform process analysis is not new (Casati et al. 2002, Grigori et al. 2004, Hepp et al. 2005, O'Riain et al. 2006, Sell et al. 2005). In 2002, Casati et al. (Casati et al. 2002) introduced the *HPPM intelligent Process Data Warehouse (PDD)*, in which taxonomies are used to add semantics to process execution data and, therefore, support more business-like analysis for the provided reports. The work in (Grigori et al. 2004) is a follow-up of the work in (Casati et al. 2002). It presents a complete architecture for the analysis, prediction, monitoring, control and optimization of process executions in Business Process Management Systems (BPMS). This set of tools suite is called *Business Process Intelligence (BPI)*. The main difference of these two approaches to ours is that (i) taxonomies are used to capture the semantic aspects (in our case, ontologies are used), and (ii) these taxonomies are flat (i.e., no subsumption relations between concepts are supported). Hepp et al. (Hepp et al. 2005) proposes merging Semantic Web, Semantic Web Services (SWS), and Business Process Management (BPM) techniques to build Semantic BPMS. This visionary paper pinpoints the role of ontologies (and reasoners) while executing semantic analysis. However, the authors do not present any concrete implementations for their ideas. The works by Sell et al. (Sell et al. 2005) and O'Riain et al. (O'Riain et al. 2006) are related to ours because the authors (i) also use ontologies to provide for the semantic analysis of systems and (ii) have developed concrete tools to support such analysis. The main differences are the kind of supported analysis. The work in (Sell et al. 2005) can be seen as the extension of OLAP tools with semantics. The work in (O'Riain et al. 2006) shows how to use semantics to enhance the business analysis function of detecting the core business of companies. This analysis is based on the so-called Q10 forms. Our paper is the first one to lay down the pillars for semantic process mining tools and to show concrete implementations in this direction.

More from an event log point of view, Pedrinaci et al. (CD07) have defined the Event Ontology and the Process Mining Ontology. These two ontologies can be used to give semantics to the event types and the process instances in logs. For instance, it is possible to say that a process instances was successfully executed.

7 CONCLUSION AND FUTURE WORK

This paper proposes a solid foundation for the development of *semantic* process mining techniques/tools. This foundation consists of three building blocks: *ontologies*, *model references from elements in logs/models to concepts in ontologies*, and *reasoners*. The ontologies formally define the shared concepts (and their relationships) to be used during the semantic analysis. The model references associate meanings to labels in logs/models. The ontology reasoners provide for the inference of subsumption relations between the concepts in ontologies. *Semantic process mining techniques based on these three elements are more accurate and robust than conventional ones because they also take the semantic perspective into account.* Therefore, they are able to provide for analysis at different abstraction levels. The approach based on these three building blocks was concretely illustrated by extending the ProM tool to read *semantically annotated logs* (via the use of the newly defined SA-MXML format) and allow for the *semantic verification of properties* in these logs (via the *Semantic LTL Checker* plug-in).

Future work will focus on three aspects. First of all, we are applying the approach to *other types of process mining*. Conformance checking based on LTL is just one of many process mining techniques that could benefit from the approach presented in this paper. Semantic annotations can also be used for process discovery, the discovery of organizational structures, decision mining, etc. The goal is to cover the whole spectrum shown in Figure 1. Second, we are working on the *discovery of semantical annotations*. Unfortunately, few systems are actually recording semantical information in their logs. Therefore, we need to extract this information from event logs. Therefore, it is vital to provide better support for ontology learning and the automatic insertion of semantic annotations. Third, from a reasoning perspective more complex inferencing, i.e., beyond subsumption reasoning, could also be envisaged so as to benefit further from the inclusion of semantic annotations. In this sense we have

already been working on the development of an ontology-based interval temporal reasoning module that will support integrating the analysis of temporal relationships between activities and processes with a fully-fledged ontology reasoner.

Acknowledgements

This research is supported by the European project SUPER (www.ip-super.org). Furthermore, the authors would like to thank all ProM developers for their on-going work on process mining techniques. More specifically, the authors would like to thank Peter van den Brand for his efforts in implementing some of the ideas presented in this paper in the ProM tool (www.processmining.org).

References

Aalst, W.M.P. van der, H.T. de Beer, and B.F. van Dongen (2005). Process Mining and Verification of Properties: An Approach Based on Temporal Logic. In R. Meersman et al., editors, OTM Conferences, LNCS, 3760 (1): 130-147.

Aalst, W.M.P. van der, B.F. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A.J.M.M. Weijters (2003). Workflow Mining: A Survey of Issues and Approaches. Data and Knowledge Engineering, 47(2):237-267.

Aalst, W.M.P. van der and A.J.M.M. Weijters (2004). Process Mining. Special Issue of Computers in Industry, 53 (3).

Casati, F. and M.-C. Shan (2002). Semantic Analysis of Business Process Executions. In EDBT'02: Proceedings of the 8th International Conference on Extending Database Technology, 287-296, London, UK.

Dongen, B.F. van, A.K. Alves de Medeiros, H.M.W. Verbeek, A.J.M.M. Weijters, and W.M.P. van der Aalst (2005). The ProM Framework: A New Era in Process Mining Tool Support. In P. Darondeau et al., editors, ICATPN, LNCS, 3536:444-454.

Fox, M.S. and M. Grüninger (1998). Enterprise modeling. AI Magazine, 19(3):109–121.

Grigori, D., F. Casati, M. Castellanos, U. Dayal, M. Sayal, and M.-C. Shan (2004). Business Process Intelligence. Computers in Industry, 53(3):321-343.

Greco, G., A. Guzzo, L. Pontieri and D. Saccà (2006). Discovering Expressive Process Models by Clustering Log Traces. IEEE Transactions on Knowledge and Data Engineering, 18(8): 1010-1027. IEEE Computer Society.

Gruber, T.R. (1993). A translation approach to portable ontology specifications. Knowledge Acquisition, 5(2):199–220.

Hepp, M., F. Leymann, J. Domingue, A. Wahler, and D. Fensel (2005). Semantic Business Process Management: a Vision Towards Using Semantic Web services for Business Process Management. In IEEE International Conference on e-Business Engineering, 535-540.

Lausen, H., J. de Bruijn, A. Polleres and D.Fensel (2005). The WSML Rule Languages for the Semantic Web. W3C Workshop on Rule Languages for Interoperability, W3C.

O'Riain, S. and P. Spyns (2006). Enhancing the Business Analysis Function with Semantics. In R. Meersman et al., editors, OTM Conferences, LNCS, 4275(1):818-835.

Pedrinaci, C. and J. Domingue (2007). Towards an Ontology for Process Monitoring and Mining. In Proceedings of Semantic Business Process and Product Lifecycle Management in conjunction with the 3rd European Semantic Web Conference, Innsbruck, Austria.

Rozinat, A. and W.M.P. van der Aalst (2006). Decision Mining in ProM. In S. Dustdar et al., editors, Business Process Management, LNCS, 4102:420-425.

(Sa06) Semantic Annotations for Web Service Description Language (SA-WSDL). *http://www.w3.org/TR/2006/WD-sawsdl-20060630/*.

Sell, D. , L. Cabral, E. Motta, J. Domingue, and R. Pacheco (2005). Adding Semantics to Business Intelligence. In DEXA Workshops, 543-547. IEEE Computer Society.

(Su06) SUPER - Semantics Utilised for Process Management within and between Enterprises. Integrated European Project. *http://www.ip-super.org/*.

(W2RF) WSML 2 Reasoner Framework (WSML2Reasoner). *http://tools.deri.org/*