Towards an Ontology for Process Monitoring and Mining

Carlos Pedrinaci and John Domingue

Knowledge Media Institute, The Open University, Milton Keynes, UK.
(c.pedrinaci,j.b.domingue)@open.ac.uk

Abstract. Business Process Analysis (BPA) aims at monitoring, diagnosing, simulating and mining enacted processes in order to support the analysis and enhancement of process models. An effective BPA solution must provide the means for analyzing existing e-businesses at three levels of abstraction: the business level, the process level and the IT level. BPA requires semantic information that spans these layers of abstraction and which should be easily retrieved from audit trails. To cater for this, we describe the Process Mining Ontology and the Events Ontology which aim to support the analysis of enacted processes at different levels of abstraction spanning from fine grain technical details to coarse grain aspects at the business level.

1 Introduction

Business Process Management (BPM) intends to support "business processes using methods, techniques, and software to design, enact, control, and analyze operational processes involving humans, organizations, applications, documents and other sources of information" [16]. As opposed to so-called Workflow Management Systems (WFMS), BPM acknowledges and aims to support the complete life-cycle of business processes which undoubtedly involves post-execution analysis and reengineering of process models. However, by doing so BPM has made more evident the existing difficulties for obtaining automated solutions from high-level business models, and for analyzing the execution of processes from both a technical and a business perspective.

The fundamental problem is that moving between the business-level and the IT-level is hardly automated. In fact, reusing the words from [2], business modeling is not process modeling. Deriving an IT implementation from a business model is particularly challenging and requires an important and ephemeral human effort which is expensive and prone to errors. Conversely analyzing automated processes from a business perspective, e.g., calculating the economical impact of a process or determining the performance of different departments in an organization, is again an expensive and difficult procedure which typically requires a human in the loop.

In this paper we shall focus on the transition from the IT perspective into the business level. First we introduce Semantic Business Process Management and present our approach to overcoming BPM limitations. Next, we focus on the mining and monitoring of processes. In particular we present initial work on the Process Mining Ontology (PMO) which aims to capture events taking place during the execution of business and IT processes and combine it with additional mining information in order to support the analysis of enacted processes at different levels of abstraction spanning from fine grain technical details to coarse grain aspects at the business level. Finally, we summarize and identify future research that will be carried on in this context.

2 Semantic Business Process Management

So far BPM has focussed mainly on supporting the graphical definition of business processes and on the derivation of skeletal executable definitions that could automate them. From the modeling perspective, notable examples are Event-driven Process Chains (EPC) [7] and the Business Process Modeling Notation (BPMN) [13]. On the technical side, the so-called Service-Oriented Architecture and related technologies such as Web Services, WS-BPEL [11] or Message-Oriented Middleware are perhaps the main enabling technologies [6].

Current approaches to BPM suffer from a lack of automation that would support a smooth transition between the business world and the IT world [5]. On the one hand current technologies only support the derivation of partial definitions of executable processes and still require an important human effort in order to obtain robust deployable solutions. On the other hand, once deployed these automated processes need to be continuously monitored, analyzed, enhanced and adapted to meet evolving (business or technical) requirements and to accommodate ever-changing (business or technical) environments.

In [5] the authors argue that the difficulties for automating the transition between both worlds is due to a lack of machine processable semantics. Often business modeling is in fact approached as process modeling [2, 3], and process modeling mainly focusses on the graphical representation of processes using modeling languages, e.g., BPMN, which cannot capture domain specific semantics. As a result, processes definitions do not provide machine processable semantics that could support business practitioners in the analysis and reengineering of processes and executable processes definitions, e.g., WS-BPEL, are bound to inflexible syntactic definitions which pose important technical difficulties.

Semantic Business Process Management that is, the combination of Semantic Web and Semantic Web Services technologies with BPM, has been proposed as a solution for overcoming these problems [5]. SBPM aims at accessing the process space of an enterprize at the Knowledge Level so as to support reasoning about business processes, process composition, process execution, etc. SBPM builds upon the use of ontologies as a core component providing the required semantic information and enhances the composition, mediation and discovery of Web Services by applying Semantic Web Services techniques.

2.1 The SUPER Approach

Major efforts are currently devoted to pursuing the SBPM vision in the context of the European project SUPER which stands for Semantic Utilized for Process Management within and between Enterprises[1]. The fundamental approach is to represent both the business perspective and the systems perspective of enterprises using a set of ontologies, and to use machine reasoning for carrying out or supporting the translation tasks between the two worlds. An initial version of a comprehensive framework conceptualizing the relevant aspects for the automation of Business Process Management tasks has been devised, see Figure 1.

The stack of ontologies builds upon the use of WSMO [14] as the core Semantic Web Services conceptualization and WSML [1] as the representation language supporting the specification of Ontologies, Goals, Web Services and Mediators. The integration between the different conceptualizations is provided by the Upper-Level Process Ontology which captures general concepts such as Process, Activity, Actor or Role which are extensively reused across the ontologies. In order to enhance the overall coherence it is envisioned that the UPO will be refined using DOLCE [9] as its foundational ontology.

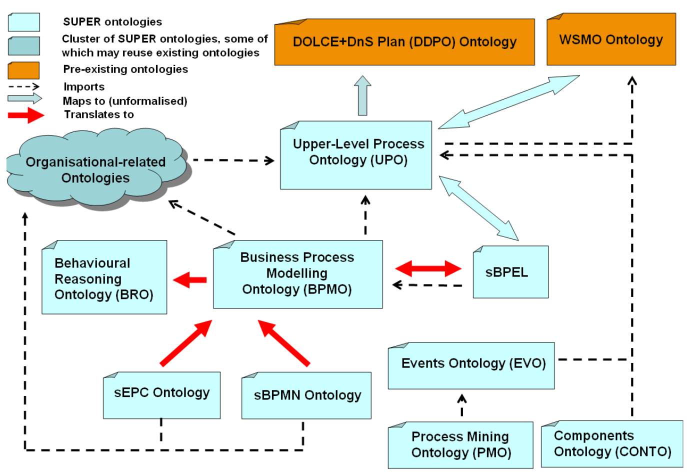

Fig. 1. SUPER Ontology Stack.

The Semantic EPC (sEPC) and Semantic BPMN (sBPMN) ontologies conceptualize EPCs [7] and BPMN [13] respectively incorporating the appropriate links to WSMO concepts. These ontologies therefore provide support for two of the main modeling notations currently used in BPM. The Business Process Modeling Ontology (BPMO) provides a common layer over both sEPC and SBPMN and links them to the rest of the ontologies from the SUPER stack. BPMO links process models to organizational information as conceptualized in the Organizational Ontologies which represent concepts like Organization, Department, Team or Employee and the relationships between them. It is also linked to the

[1] More information at http://www.ip-super.org

Behavioral Reasoning Ontology (BRO) whose aim is to support the composition of processes by reasoning about their behavior. Finally, BPMO enables the transformation of business processes modeled using different notations into their executable form. To support the execution of business processes, BPEL [11] has been chosen as the representation language for its extensive support and use. The Semantic BPEL (sBPEL) ontology formalizes BPEL and includes additional constructs linked to WSMO so as to support the mediation between heterogeneous data or processes, or the invocation of Goals as opposed to explicitly specified Web Services.

Different transformations between these different conceptualizations have been defined, see red arrows in Figure 1. An additional transformation, although not shown in the figure, has been defined for transforming sBPEL into a serialization format, BPEL4SWS, for executing processes on extensions of existing workflow engines. BPEL4SWS is an XML serialization format that is mainly an extension from sBPEL with typical SWS features, e.g. including support for goals instead of predefined activities and use of mediators.

The ontology stack also identifies the Components Ontology which aims to support the conceptualization of IT-level aspects, such as software components and systems. This ontology will be based on previous research on the semantic management of middleware [12]. Finally, because having semantics at the level of processes but not at the level of monitoring and mining defeats to an important extent the benefits that can be obtained from SBPM, the ontology stack includes two ontologies, the Process Mining Ontology and the Events Ontology which are the focus of this paper.

3 Semantic Process Monitoring and Mining

One of the distinguishing characteristics of BPM solutions with respect to traditional WFMS is commonly referred to as Business Process Analysis(BPA) [16]. In a nutshell, BPA aims at monitoring, diagnosing, simulating and mining enacted processes in order to support the analysis and enhancement of process models. The main goal pursued by BPA are on the one hand the verification or validation of the actual execution with respect to prescribed or expected processes, and on the other hand the identification, in a more or less timely manner, of potential improvements of business processes. The knowledge gained in this phase is thus employed for reengineering and fine tuning existing process definitions.

[3] identifies three main levels for the analysis of e-businesses information systems, as shown in Figure 2: the business level, the process level and the IT level. The first level is concerned about the value exchanges between the different actors involved (e.g., companies) and is therefore of particular relevance for business practitioners. The second level considers the process point of view (e.g., BPEL level) and is usually the focus of process architects. Finally, the third level is concerned about technical details such as the decomposition of a process into Web Services. An effective BPA solution must therefore provide the means for analyzing existing e-businesses at these three layers.

This layering is even more complex since there can be, and there usually are, nested layers and different perspectives that can be adopted within each of these layers. For instance the business analyst could focus on individuals, departments or the whole organization. The process execution view might involve several (sub) processes, i.e., what appears to a process as a simple atomic task might in fact be supported by another process as is often the case for complex processes. Finally the process execution will rely on some actual IT infrastructure which will follow some algorithm–a process in itself–which we might need to analyze.

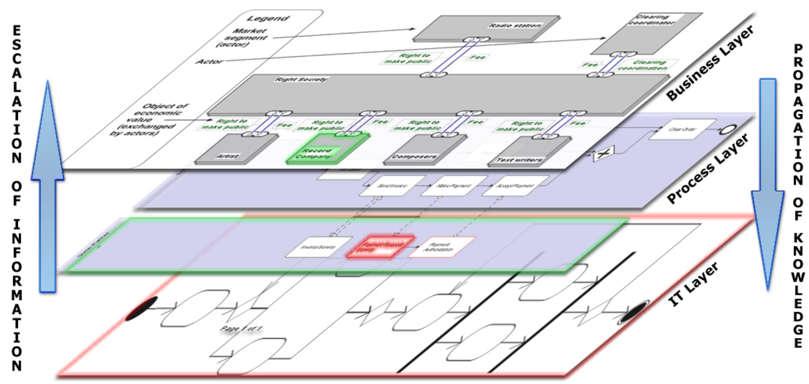

Fig. 2. Business Process Management layers.

Further complications come from the fact that although these layers are clearly distinguished, there exists an inherent intertwining between them. On the one hand, decisions at the business level have implications at the process level which might in turn affect the IT level. On the other hand, the execution of some activity by some system, e.g., a Web Service, affects the process level and this might escalate to the business level. It is worth noting that this propagation between layers takes place both at design time and runtime. In fact, in some domains like telecommunications where for example quality of service is crucial, the technical details regarding the process execution are of particular relevance at the process and even at the business levels. Being able to properly correlate the data between layers at runtime can therefore be of particular importance. Automating this, as necessary for what is commonly referred to as Business Process Intelligence, requires semantic information that spans these layers of abstraction and which should be easily retrieved from audit trails.

3.1 On the Need For a Process Mining Ontology

BPA is mainly targeted at business users and process architects, although it is also concerned about the technical details since automated processes execution eventually depend on the underlying IT infrastructure. In fact, Business Activity Monitoring (BAM), one of the main areas in BPA, uses data logged by the underlying IT support in order to monitor, diagnose and mine executed processes.

In this paper we shall use *audit trail* as commonly adopted in the Workflow and Business Management communities to refer to this data.

So far, Extensive work has been devoted to the definition of mechanisms for the communication of events or notifications between systems. Among this work we can mention CORBA, JMS, WS-Eventing, WS-Notification or Message-Oriented Middleware in general [6, 10]. These technologies, although not uniquely devoted to supporting monitoring, provide the necessary technical support for communicating monitoring information at runtime. What remains to be defined is an appropriate format for capturing audit trails in a way that would support the creation of fully-fledged generic BPA solutions. In fact, as often happens in IT, every specific system provides its own level of detail in heterogeneous formats making it particularly difficult to integrate the audit trails generated by diverse systems as well as it complicates the creation of general purpose BPA solutions.

Perhaps the main effort in defining a common format for storing audit trails has been undertaken in the context of the ProM framework, a pluggable environment for process mining [17]. The ProM framework is able to apply a wide range of process mining algorithms over log data stored in MXML [18], an XML-based format that captures the necessary information for audit trails to be useful in the context of process mining. In a nutshell, MXML establishes that each audit trail should be an event happened at a specific point of time during the execution of an uniquely identifiable activity. The events should specify what actually happened (e.g., start or end of an activity) and they should refer to a concrete process instance belonging to a specific process. The reader is referred to [18] for more details about the format.

MXML has proven to be suitable for capturing diverse audit trails. In particular there currently exists support for importing logs from PeopleSoft, Staffware or FLOWer to name a few [18]. Still, MXML is not all there needs to be to support SBPM. Audit trails, although generated by general purpose software, obviously concern domain specific resources. That is, work may be performed by a specific person, belonging to a specific department from a concrete office of a given company. An MXML log is only able to refer to a *label* identifying the name of this person. The actual *semantics*, i.e., who that person is, where he or she works and other related information are not available. Indeed, in many cases it may be possible to create ad-hoc solutions for retrieving this knowledge but this clearly defeats the very purpose of defining a generic format. In other words, MXML suffers from a lack of machine processable semantics as we previously identified for BPM in general. As a result navigating through the levels of abstraction required for analyzing e-business solutions requires a human in the loop capable to identify the links and relations across layers.

Semantic Web technologies, in particular ontologies [4] for they are formal, sharable and extensible representations, together with the related tooling such as repositories and reasoners, offer a suitable framework on which to build upon generic BPA solutions. First, they are particularly well-suited for defining shared conceptualizations in order to support the integration of heterogeneous sources of information. Second, having a formal definition they are directly amenable to automated reasoning, providing the flexibility required for navigating through different levels of abstraction and querying the overall body of knowledge about the business processes. Finally, ontologies are a step forward towards Business Intelligence, since they provide a natural means for defining reaction rules or applying knowledge-based techniques like Problem-Solving Methods [15], in order to intelligently adapt the behavior of business processes.

In the next two sections we describe the Process Mining Ontology, that aims to enhance the state-of-the-art in BPA by semantically capturing audit trails and process mining details. In order to do so, we first focus on the Events Ontology which provides the core framework for capturing events generated by IT systems. Then we present the Process Mining Ontology which builds upon the former, see Figure 1, and enhances it with mining specific definitions.

3.2 Events Ontology

The Events Ontology (EVO) supports capturing events taking place during the life-cycle of both business and IT processes. It is based on MXML which is in turn based on the analysis of different types of logs in real systems [18]. Doing so ensures (i) that we capture the required information for applying several process mining algorithms;(ii) that we can import logs from some of the main existing systems, and (iii) that we can reuse and enhance ProM for mining EVO logs.

Figure 3 shows the EVO as a UML class diagram[2]. The main concept in EVO is *Monitoring Event*. This concept represents events generated at a specific point on time by an IT system. Monitoring Events are therefore different from "clicking a button" as usually understood in computer science. Events are generated by an *Actor* which is defined in UPO and therefore allows for reusing the rest of the ontologies defined in the SUPER stack. In particular, Actor is refined both in the Organizational Ontology, where Companies, Departments, and Individuals are defined, and in the Components Ontology which describes software components.

Time is one of the main characteristics in Event processing [8]. As a consequence, each Monitoring Event has both a *creation timestamp* based on the clock of the system where the Event was generated, and a *reception timestamp* which captures the actual moment in which the Event was received. The former accommodates pre-existing logging mechanisms and supports performance analysis at the level of every specific component since it is not influenced by external aspects such as the network latency. The latter should be introduced by the events propagation infrastructure accordingly. The reception timestamp is particularly useful for monitoring distributed systems since it supports establishing a global order among all the events without the need for clock synchronization mechanisms and also supports detecting network malfunctions. We believe this attribute will be of most relevance given that nowadays more and more business processes are interorganizational, making the application of clock synchronization techniques particularly challenging, if even realistic.

[2] The complete ontology represented in WSML can be found at http://kmi.open.ac.uk/people/carlos/ontologies/PMO/evo.wsml

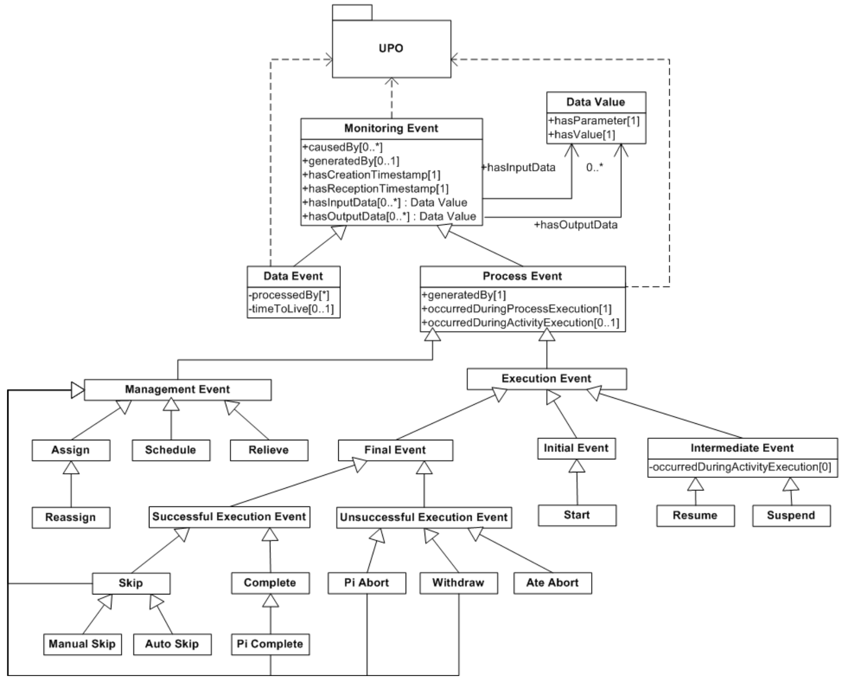

Fig. 3. Events Ontology as a UML Activity Diagram.

Events may have a set of inputs and outputs which are specified as *Data Value* instances which identify the parameter affected and the value given. Finally, an Event may have an associated *causality vector* indicating other Events which caused the generation of the former. This type of causality information is particularly relevant for processes monitoring and mining although it is rarely propagated by the runtime infrastructure excepting some Event-Based infrastructures [8]. Additionally the causality vector attribute can be used for post-execution analysis derived information if necessary.

Monitoring Events are further refined into *Data Events* and *Process Events*. Data Events accommodate Event-Based Systems so that their execution can also be traced. These events are therefore placeholders for information interchanged between actors when it is not directly related to a specific process execution. In addition to the attributes inherited from Monitoring Event, Data Events can also capture a set of actors that have already processed the event, and can include a *Time-To-Live* parameter which basically sets an expiry date for the event.

Process Events represent events that can take place during the life-cycle of a business or IT process, thus encompassing its scheduling and execution. Every event will affect a specific *Process* and might concern some *Activity* belonging to this Process. Both Process and Activity are concepts defined in the UPO and refined in the BPMO and the Components Ontology. Thus, by abstracting away from the specificities of the process being logged, EVO provides the means for capturing audit trails at different levels of abstraction, i.e., the process and the IT level. It is worth noting that Process Events can refer to the whole Process by omitting the Activity, e.g., “a Process is suspended”, or to a specific Activity within the Process, e.g. “Activity Y was correctly executed”.

The EVO refines Process Events into *Management Events* and *Execution Events*. Management Events are Process Events that are not directly caused by the execution of some Process or Activity but rather generated by the action of some external actor, typically a human but possibly an automated management system. These are usually work distribution events such as *Assign*, *Re Assign*, *Schedule*, *Relieve*, and may also affect the eventual execution of the Process, e.g., *Skip*, *Pi Complete*, *Pi Abort* and *Withdraw*.

Execution Events are Events that concern the actual execution of some Process or Activity. They are further refined into *Initial Events*, *Intermediate Events* and *Final Events*. Initial Events indicate the start of the execution of a process. Intermediate Events are those Execution Events which affect the actual execution but are not Final nor Initial. This is the case for *Resume* and *Suspend*. Activities within a Process are considered atomic thus, Intermediate Events cannot refer to an Activity. Should the Activity have a complex implementation which could yield Intermediate Events, these should be logged as part of the subprocess triggered by the Activity. Last, Final Events are those that conclude the execution of a process and are categorized as *Successful* or *Unsuccessful*, which are self-describing.

Among the remaining concepts, we next describe briefly some of the main ones captured in the ontology. The correct completion of some Process or Activity is captured by means of *Complete* events. Conversely, *Ate Abort* events indicate that the execution of the Activity or Process was aborted. The reason for the execution being aborted originates from the execution itself (e.g. software exception, unexpected result). Both *Pi Complete* and *Pi Abort* are the managerial counterpart of Complete and Ate Abort, i.e., some management action has marked the Process or Activity as *completed* or *aborted*. Furthermore, the ontology includes the typical events related to the management and scheduling of processes such as *Resume*, *Suspend*, *Assign*, *Reassign* or *Schedule* which we believe are self-describing. Finally, *Skip* events indicate that the Process or the Activity has been skipped, i.e. will not be executed, and is considered as being properly executed.

3.3 Process Mining Ontology

The Process Mining Ontology formalizes different aspects relevant for mining and analyzing business or IT processes[3]. The PMO integrates all the diverse knowledge required for mining processes by reusing additional ontologies from the SUPER ontology stack such as the EVO, the Organizational Ontology, the BPMO and the Components Ontology, see Figure 1. PMO depends directly on

[3] The complete ontology represented in WSML can be found at http://kmi.open.ac.uk/people/carlos/ontologies/PMO/pmo.wsml

the EVO since it reuses the concepts defined therein, although they remain as separate ontologies for modularity and reusability reasons. Indirectly it integrates through the UPO, the conceptualizations captured in the Organizational Ontology, the BPMO and the Components Ontology.

The PMO, shown in Figure 4, defines containers for workflows, processes and data events audit trails so as to provide the appropriate perspective for workflow, process or Event-Based Systems monitoring and mining respectively. These containers are the *Data Events Log*, the *Process Execution Log* and the *Workflow Log* concepts. It is worth noting that although it would be possible to obtain the containers by means of queries, an explicit definition allows for attaching any analysis result obtained to them for future reuse. The PMO includes a default instance both for capturing Data Events Logs and Workflow Events Log, and includes an axiom for the automated creation of instances of Process Execution Log whenever events concerning a new process are received. Further axioms ensure the events are automatically added to the corresponding containers.

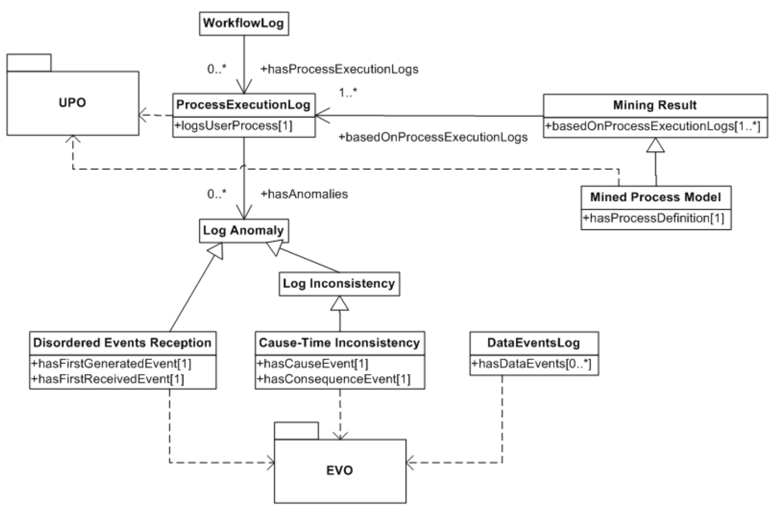

Fig. 4. Process Mining Ontology as a UML Activity Diagram.

The current version of the PMO includes an initial set of mining specific axioms in an attempt to enhance current mining support with automated detection of some anomalies. Currently, the ontology supports detecting the disordered reception of events and inconsistent cause-time relationships between events. The former anomaly, which is typically due to network problems, is detected based on the use of the two timestamps captured in Monitoring Events. Basically, an axiom checks wether the events generated by some actor are received in the same order. The detection of inconsistent cause-time relationships is particularly relevant for event analysis and is based on the causality vector captured in Monitoring Events. The relationship is considered inconsistent when the cause for an event is received after the consequent event. Again, the most likely situation where this could happen is due to network communication problems between the system being monitored and the monitoring infrastructure.

It is important to note that instead of including these axioms as constraints that would basically avoid the existence of such anomalies, we define them as logic programming rules that detect and capture the anomalies as part of the conceptualization, see Figure 4. In fact, it is well known that often systems logs present anomalies but eliminating these would also get rid of important information. For instance, the mere fact of knowing that the reception of events is disordered is relevant information from the IT perspective. Thus, detecting and capturing these avoids using anomalous data for some analysis processes while it provides and maintains valuable information for others.

Finally, the current version of the PMO defines a placeholder for capturing mining results. In particular we currently identify the super concept *Mining Result* which represents the results obtained from applying mining algorithms over the set of logs identified by the attribute *Based On Process Execution Logs*. *Mined Process Model* is the only Mining Result defined so far although we foresee concepts like the *Mined Organizational Model* or performance related results. Future research will be devoted to defining these concepts and to refining the anomalies conceptualization.

4 Conclusions

In this paper we have introduced the lack of automation existing in state-of-the-art BPM solutions. This drawback has been attributed to a lack of machine processable semantics and we have presented Semantic Business Process Management as approached in the SUPER project as a solution. We have focussed in particular in the challenges for the monitoring and mining of processes from a semantics perspective and we have argued for the need of ontologies to support these tasks. Finally we have described in depth the Events Ontology and the Process Mining Ontology that aim to support the analysis of enacted processes at different levels of abstraction spanning from fine grain technical details to coarse grain aspects at the business level.

The ontologies, although still subject of research and improvement are built upon solid bases stemming from one of the most complete general purpose mining solutions. The ontologies are part of an extensive formalization of the BPM domain and therefore allow accessing the whole body of knowledge about processes, organizations or IT systems in order to support making queries at different levels of abstraction. Future research will be devoted to applying these ontologies for the monitoring and mining of the various use-cases of SUPER. Out of this experiments, we expect to be able to assess the suitability and benefits from using EVO and PMO for monitoring and mining as well as we might identify potential improvements for the ontologies during this process.

References

1. Jos de Bruijn. D16.1v0.21 the web service modeling language wsml. http://www.wsmo.org/TR/d16/d16.1/v0.21/, October 2005.
2. J. Gordijn, H. Akkermans, and H. van Vliet. Business modelling is not process modelling. In *Conceptual Modeling for E-Business and the Web, LNCS 1921 (ECOMO 2000)*, pages 40–51, Salt Lake City, USA, 2000. Springer-Verlag.
3. Jaap Gordijn and Hans Akkermans. E3-value: Design and evaluation of e-business models. *IEEE Intelligent Systems*, 16(4):11–17, 2001.
4. T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.
5. Martin Hepp, Frank Leymann, John Domingue, Alexander Wahler, and Dieter Fensel. Semantic business process management: A vision towards using semantic web services for business process management. In Francis C. M. Lau, Hui Lei, Xiaofeng Meng, and Min Wang, editors, *ICEBE*, pages 535–540. IEEE Computer Society, 2005.
6. Gregor Hohpe and Bobby Woolf. *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2003.
7. G. Keller, M. Nüttgens, and August-Wilhelm Scheer. Semantische prozessmodellierung auf der grundlage "ereignisgesteuerter prozessketten (epk)". Arbeitsbericht Heft 89, Institut für Wirtschaftsinformatik Universität Saarbrücken, 1992.
8. David C. Luckham. *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.
9. Claudio Masolo, Stefano Borgo, Aldo Gangemi, Nicola Guarino, Alessandro Oltramari, and Luc Schneider. Wonderweb deliverable d17. the wonderweb library of foundational ontologies and the dolce ontology, August 2002.
10. Gero Mühl, Ludger Fiege, and Peter Pietzuch. *Distributed Event-Based Systems*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
11. OASIS Web Services Business Process Execution Language (WSBPEL) TC. Web services business process execution language version 2.0 committee specification. http://docs.oasis-open.org/wsbpel/2.0/CS01/wsbpel-v2.0-CS01.pdf, January 2007.
12. Daniel Oberle. *Semantic Management of Middleware*, volume I of *The Semantic Web and Beyond*. Springer, New York, FEB 2006.
13. Object Management Group. Business process modeling notation specification - final adopted specification. Available at http://www.bpmn.org, February 2006.
14. Dumitru Roman, Holger Lausen, and Uwe Keller. D2v1.3. web service modeling ontology (wsmo). http://www.wsmo.org/TR/d2/v1.3/, October 2006.
15. R. Studer, R. Benjamins, and D. Fensel. Knowledge engineering: Principles and methods. *Data Knowledge Engineering*, 25(1-2):161–197, 1998.
16. Wil M. P. van der Aalst, Arthur H. M. ter Hofstede, and Mathias Weske. Business process management: A survey. In *Business Process Management*, pages 1–12, 2003.
17. B.F. van Dongen, A.K.A. de Medeiros, H.M.W. Verbeek, A.J.M.M. Weijters, and W.M.P. van der Aalst. The prom framework: A new era in process mining tool support. In *Applications and Theory of Petri Nets 2005. 26th International Conference, ICATPN 2005*, pages 444–454, Miami, USA, June 2005. Springer-Verlag.
18. Boudewijn F. van Dongen and Wil M. P. van der Aalst. A meta model for process mining data. In *EMOI-INTEROP*, 2005.