An Ontology For Workflow Organizational Model Mining

Rami Sellami†, Walid Gaaloul* and Samir Moalla†
*Institut TELECOM, CNRS UMR Samovar, Evry, France
Email: {rami.sellami, walid.gaaloul}@it-sudparis.eu
†Faculty of Sciences of Tunis, University Tunis EL Manar, Tunisia
Email: samir.moalla@fst.rnu.tn

Abstract—Continuous and unforeseeable evolution of business rules, processing logics, and organizational structures within enterprises, requires from business process management systems to integrate continuous design. Supporting business process rediscovery based on workflow logs analysis, workflow mining gathers retroactive (re)design techniques necessary to understand business process execution reality. Most of the works in this area focus on the control flow perspective, while very few of them address the organizational aspect. In addition, the used workflow mining techniques suffer from the lack of automation due to the purely syntactic logs. In this paper, we propose an Organizational Ontology (OrO) specifying the organizational model. This ontology is used to semantically annotate log files and establish an organizational knowledge base that will be used to discover relationships between performers in a workflow. Our approach has been implemented within the ProM framework.

Index Terms—workflow mining, organizational model, organizational ontology, SBPM, ProM.

I. INTRODUCTION

During the business process execution, Business Process Management (BPM) systems generate business event and record them in event logs [1]. An event log contains data about what are the performed tasks and their sequencing, who are the performers of these tasks and when they are executed. This information can be discovered automatically through workflow mining techniques [2]. These techniques allow the discovery of business process models in order to enhance and correct initial models which are handmade.

Although the workflow mining is important for maintaining the workflow life cycle, it still suffers from a lack of automation in its life cycle [3], [4], [5]. Indeed, there are some substantial difficulties in ensuring the transition between the business and IT worlds. One of the major problems is the transition of the high-level business process models, which are modeled by a business analyst, to workflow models, which are executable IT representations of the business processes. This problem causes a high time delay. This conflict is due to the lack of understanding business needs by IT experts and technical details by business experts. Thus, we understand that these problems are due to the lack of semantics in the BPM systems.

To cover this gap, Hepp et al. suggest in [6] the creation of Semantic Business Process Management (SBPM) systems based on the semantic Web technologies and ontologies. The main purpose of SBPM systems is to provide an automatic access to information describing the business process life cycle [7]. Hence, a high level of automation is ensured such as reasoning about business process, their modeling, their execution, etc. SBPM is also important for the workflow mining. In fact, van der Aalst et al. [8] see that semantic annotation of event log files is necessary in order to provide more powerful and reliable workflow mining techniques.

Different workflow mining approaches and techniques are focusing on discovering the control flow model (that is the process activity sequencing) [9]. Nevertheless, very few works deal with the discovery of workflow organizational model (that describes how the workflow users communicate in purpose of sharing their skills and knowledge). In this paper, we are interested in the organizational model mining. In our previous work, we proposed some techniques for organizational structure discovery. However, these techniques are purely syntactic and suffer from the lack of automation. Hence, in order to ensure semantic organizational mining, we propose first a generic ontology for the workflow organizational model. This ontology describes the organizational structure and the communication between performers in the workflow. Second, we enrich event logs through semantic annotation from an organizational perspective. The semantic annotation of the event logs allow us to create a knowledge base by populating the ontology with process instances. Therefore, we discover relationship between performers in a workflow by specifying some rules on this knowledge base.

The rest of this paper is organized as follows. Section II presents the semantic annotation of event log. In section III, we present our organizational ontology for modeling the workflow organizational model. In Section IV, we give a concrete implementation in order to demonstrate the applicability of our approach. Section V reviews the related works. Section VI provides a conclusion and future work.

II. AUTOMATIC SEMANTIC ANNOTATION

BPM systems log the external events that capture the activities life cycle. XES1 [10], [11], which stands for eXtensible Event Stream, is an XML-based standard for event logs. This standard has been adopted by the IEEE Task Force on process

1http://www.xes-standard.org/
mining as the default interchange format for event logs. It replaces the MXML [12] (Mining XML) as the new event log format for process mining tools. The Listing 1 shows the structure of an XES log file. The root node of each XES document is the log element (see Line 1) and it represents a process log. This latter is the container of a process instances execution set. Each process instance is represented by the trace element (see Lines 9 and 31). These instances contain the events that have been observed during a process execution. Similarly, each process instance has a set of events represented in the XES event log by the event element (see Lines 11, 17 and 23). To store any data in the XES log file, we use attributes which have a key, a known type (for example string, date, etc.) and a value of that type (see Lines 8, 10, 12–15, 18–21 and 24–27). In addition, every attribute is defined by standard extensions (which are concept, lifecycle, organizational, time and semantic extensions) or user defined extensions (see Lines 2–7). In this XES event log, we give an example of fines management process (see Line 8). This process has a process instance called Instance-1 (see Line 10). We present three events in this process instance. The event File-Fine is executed by Lucas on March, 03, 2012 at 15:30pm CET (see Lines 11–16). Whereas, the event Send-Bill means that Mary sends a bill on March, 05, 2012 at 09am CET (see Lines 17–22). Finally, the event Payment denotes that John pays a fine on March, 09, 2012 at 15pm CET (see Lines 23–28).

However, the XES log file suffers from a lack of information about the organizational model and its semantics. In fact, this is due to the absence of information in the event log file reflecting the different policies for allocating activities and interactions between performers of a business process. In the log file, we did not find information about for example failure, how performers are interacting, which protocol performers are using to communicate, etc. Concretely, in the event log example (see Listing 1), we do not have information about the receiver of the bill that Mary sent (see Lines 17–22) or about the beneficiary when John pays the bill (see Lines 23–28). The lack of information about the organizational model in the event log presents an obstacle to the workflow organizational model mining. In addition, event logs are purely syntactic. Hence, they expose some ambiguities in discovering relationships between performers. This lack represents also an obstacle in sharing organizational knowledge.

We propose a multi-agent organizational event log. This enrichment is ensured by the use of the communication language FIPA-ACL\(^2\) [13]. This language is based on the usage of a performatives set that is a standard vocabulary. Performatives give the communication facility among performers by exchanging messages in various protocols. We enrich the log file by adding three specific data elements (see Listing 2) in order to give more information about the organizational model. The first element is the Initiator which represents the performer desiring to communicate with a receiver using performatives (see Line 10). The second element is the Receptor (see Line 11) and it is a performer who waits for active communications with the sender. The third element is performatives (see Line 12). This element describes the communication intention expressed using the FIPA-ACL communication language. In Listing 2, we enrich the example given in Listing 1. Indeed, the payment is executed (the performatives Execute) by John (the initiator) to Mary benefit (the Receptor).

```
1 <log>
2  <extension name="Lifecycle" prefix="lifecycle" src="http://www.xes-standard.org/lifecycle.xesext"/>
3  <extension name="Time" prefix="time" src="http://www.xes-standard.org/time.xesext"/>
4  <extension name="Concept" prefix="concept" src="http://www.xes-standard.org/concept.xesext"/>
5  <extension name="Semantic" prefix="semantic" src="http://www.xes-standard.org/semantic.xesext"/>
6  <extension name="Organizational" prefix="org" src="http://www.xes-standard.org/org.xesext"/>
7  <extension name="Instance" prefix="instance" src="http://www.xes-standard.org/instance.xesext"/>
8  <extension name="LifeCycle" prefix="lifecycle" src="http://www.xes-standard.org/lifecycle.xesext"/>
9  <extension name="Trace" prefix="trace" src="http://www.xes-standard.org/trace.xesext"/>
10 <event>
11  <string key="concept:name" value="Instance-1"/>
12  <string key="concept:name" value="File-Fine"/>
13  <string key="lifecycle:transition" value="complete"/>
14  <string key="resource" value="Lucas"/>
15  <date key="time:timestamp" value="2012-03-03T15:30:00.000+01:00"/>
16 </event>
17 <event>
18  <string key="concept:name" value="Send-Bill"/>
19  <string key="lifecycle:transition" value="complete"/>
20  <string key="resource" value="Mary"/>
21  <date key="time:timestamp" value="2012-03-05T15:00:00.000+01:00"/>
22 </event>
23 <event>
24  <string key="concept:name" value="Payment"/>
25  <string key="lifecycle:transition" value="complete"/>
26  <string key="resource" value="John"/>
27  <date key="time:timestamp" value="2012-03-09T11:15:00.000+01:00"/>
28 </event>
29 </log>
```

Listing 1. Example of an XES log file

We also semantically annotate event logs in order to remove ambiguities and enable knowledge sharing. Concretely, we annotate the three specific elements (see Listing 3) Initiator, Receptor, and Performatives, by adding the attribute modelReference (see Line 3, 6 and 9). To do so, we propose an ontology specifying the organizational model. This ontology is used to semantically annotate log files and establish an organizational knowledge base that will be used to discover relationships between performers in a workflow. We present this ontology in the following Section III.

```
1 <log>
2  <string key="performative" value="execute"/>
3  <string key="concept:modellingReference" value="http://www.fipa.org/repository/aclspecs.html"/>
4  <string key="modeller" value="example"/>
5  <string key="initiator" value="John"/>
6  <string key="receiver" value="Mary"/>
7  <string key="modelReference" value="example"/>
8  <string key="intent" value="order"/>
9  <string key="performative" value="execute"/>
```

Listing 3. Example of an annotated log file

\(^2\)http://www.fipa.org/repository/aclspecs.html
III. ONTOLOGICAL MODELING OF THE ORGANIZATIONAL STRUCTURE IN A WORKFLOW

In this section, we present an organizational ontology, called \textit{OrO}. This ontology provides knowledge sharing in the organizational model through semantically annotated event file. For this purpose, we start by defining the workflow organizational meta-model on which the proposed ontology is based (Section III-A). Then, we introduce an ontology for describing FIPA-ACL messages (Section III-B). Finally, we present the \textit{OrO} ontology which integrates the two elements described above: the workflow organizational meta-model and the FIPA-ACL ontology (Section III-C).

A. Workflow organizational meta-model

In [14], Ferber et al. define the AALAADIN meta-model as a generic meta-model for multi-agent systems. Based on this, we define the workflow organizational meta-model (see Fig.1). It is based on three main concepts which are:

- **Performer**: It is defined as an independent communicative entity that plays roles in some different organizational units.
- **Organizational Unit**: It is a set of performers and roles. Every performer can be a member of \(N\) organizational units at the same time and must request its membership to every unit. Every unit has its own responsible who is its creator.
- **Role**: It is an abstract representation of a performer function, service or identification within an organizational unit. Every performer can handle several roles and the same role can be handled by other performers.

![Fig. 1. The workflow organizational meta-model](image)

B. The FIPA-ACL ontology

We use the communication language FIPA-ACL [13] to describe the interaction between performers and the exchanged messages among them. In this context, I. Dickinson et al.\(^3\) propose an ontology for exchanging messages between performers based on the FIPA-ACL language. Hence, we adopt the same logic and we propose an ontology (see Fig.2) describing the communication in the workflow organizational model.

The main class in the FIPA-ACL ontology is the \textit{FipaAclMessage}. It represents a message in a FIPA-ACL communication. This class is related to two classes. The first one is the \textit{Performer} class. It denotes performers in the organizational model. The \textit{FipaAclMessage} class is related to this class by three properties which are \textit{sender}, \textit{receiver} and \textit{replyTo}. These properties map respectively the message sender in an interaction, its receiver, and the performer to whom the reply of the message should be sent. The second class is called \textit{SpeechAct}. It represents the performatives set such as: call for proposal, propose, execute, etc. It is related to the class \textit{FipaAclMessage} by the \textit{performative} property that maps a message to its performative. Every performative type is illustrated by a subclass of the \textit{SpeechAct} class in this ontology. Through these subclasses, we cover a heterogeneity problem. In fact, if we have two performatives expressing the same intention (e.g. \textit{call for proposal} and \textit{cfp}) and they are individuals of the class \textit{SpeechAct}, we do not easily realize that they have the same semantic. In our ontology, we consider them as individuals of the same subclass (see Fig.3).

![Fig. 2. FIPA-ACL ontology](image)

![Fig. 3. Example of overcoming the heterogeneity problem](image)

In addition, the class \textit{FipaAclMessage} is characterized by two data type properties: \textit{content} which denotes the message content and \textit{conversation-id} which denotes an identifier that identifies a conversation.

C. Organizational Ontology (\textit{OrO})

Based on the workflow organizational meta-model and the FIPA-ACL ontology, we define an Organizational Ontology (\textit{OrO}) so that we can use it to semantically annotate event logs (see Fig.4). It describes the whole workflow organizational model. To introduce this ontology, we rely on Definition 1:

\textbf{Definition 1}: Our Organizational Ontology \textit{OrO} is defined by the 4-tuples \(<C_{OrO}, DTP_{OrO}, OP_{OrO}, A_{OrO}>\):

- \(C_{OrO} \): Concepts defining the organizational model;
- \(DTP_{OrO} \): Data providing additional information about concepts in the organizational model;
- \(OP_{OrO} \): Relationships between different concepts in the organizational model;
- \(A_{OrO} \): Annotations of the concepts and relationships.

![Fig. 4. Organizational Ontology](image)
- $A_{OrO}$: A set of evident truths used to enrich the organizational model.

In the following, we define each element of the 4-tuples separately. We start by the first element $C_{OrO}$. On the one hand, we have concepts defining the organizational structure (illustrated by $Process\_Organizational\_Structure$). This first part of the ontology is based on the workflow organizational meta-model (see Section III-A). Hence, an organizational structure is defined by actors (illustrated by $Performer$), roles (illustrated by $Role$) and groups of actors (illustrated by $Organizational\_Unit$). On the other hand, we link the FIPA-ACL ontology (see section III-B) with the $Performer$ concept.

So, we describe the interaction between performers and the exchanged messages between them. Indeed, it is provided by the classes: $FipaAclMessage$ and $SpeechAct$.

The second tuple in Definition 1 is $DP_{OrO}$. It provides additional information about organizational concepts. Indeed, the $unitName$, the $roleName$, the $actorName$ and the $performativename$ denote respectively the names of organizational unit ($OU$), performer role, performer and performative. In addition, $messageContent$ defines the content of a message during a communication.

The third tuple is called $OP_{OrO}$. It defines relationships between organizational concepts. We introduce these relationships in two times. We start by defining the first set by Table I. For instance, the two relationships which are $isDefinedByPerformer$ and $isDefinedByRole$ denote respectively that an $OU$ is defined by a set of performers and a set of roles.

The second part of $OP_{OrO}$ consists on describing communication between performers in the organizational model. In this context, we define a list of relationships (see Table II). For instance, the $communicateWith$ relationship enables us to represent performers’ interaction. Whereas the $superior$ relationship allows us to discover hierarchical relationship between performers in a workflow.

The final tuple in the Definition 1 is the axioms set $A_{OrO}$. This set provides evident truths used to enrich $OP_{OrO}$ and $C_{OrO}$ sets. For instance, a performer must belong to at least one $OU$ and must play at least one role. Furthermore, a role is played at least by one performer. Finally, the relationships ($isPlayedBy$/$plays$) and ($superior$/$subordinate$) are inverse relationships.

Besides, we can enrich our ontology by defining some rules. These rules are an alternative way of defining new relationships between performers in order to complement the ontology. It helps mainly in discovering relationships between performers in an organizational model. Rule’s definition is based on $OP_{OrO}$ and $C_{OrO}$ sets and is written according to the Definition 2:

**Definition 2:** A rule is defined as: $A(x, y) \land B(x) \land C(x, z) \land D(z) \land \ldots \Rightarrow E(y, z)$ where:

- $A(x, y)$, $C(x, z)$, $E(y, z)$: predicates defining relationships ($OP_{OrO}$) such as: $superior$, $communicateWith$, etc.;
- $B(x)$, $D(z)$: predicates defining Concepts ($C_{OrO}$) such as: $Performer$, $FipaAclMessage$, etc.;
- $x$, $y$, $z$: variables, literals, individuals, etc.

According to the Definition 2, we give examples of some rules inferring properties in our ontology:

**Rule$_{OrO1}$:** $FipaAclMessage(z) \land Performer(x) \land Performer(y) \land receiver(z, y) \land sender(z, x) \Rightarrow communicateWith(y, x)$: This rule defines the $communicateWith$ relationship. It means that if the performer $x$ sends a FIPA-ACL message, called $z$, to the performer $y$, then $x$ communicates with $y$.

**Rule$_{OrO2}$:** $Performer(x) \land Performer(y) \land Performer(z) \land superior(x, y) \land superior(y, z) \Rightarrow superior(x, z)$: This rule means that if $x$ is the $superior$ of $y$ and $y$ is the $superior$ of $z$ then $x$ is the $superior$ of $z$ ($x$, $y$ and $z$ are performers).

**Rule$_{OrO3}$:** $Performer(x) \land Performer(y) \land superior(x, y) \Rightarrow subordinate(y, x)$: This rule implies
that if the performer $x$ is a superior of the performer $y$ then $y$ is a subordinate of $x$.

IV. IMPLEMENTATION

In this section, we present a concrete implementation of the different steps of our approach (see Fig.5). First of all, we collect XES event log files by simulating already designed workflow processes (Section IV-A). After that, we semantically annotate log files using the OrO ontology (Section IV-B) so that we can populate the ontology and obtain a new ontology called OrO$_p$ (Section IV-C). Finally, we show how we can discover the relationship between performers based on OrO$_p$ ontology.

![Fig. 5. Different steps of our approach](image)

A. Simulating event log file

To test the feasibility of our approach, we have used simulated logs because getting real logs from big size workflow examples turns out to be a difficult task. The advantage in using simulated logs is that is easier to fix and vary external factors ensuring a better diversity of the examples and more accurate validation. In order to do this, we use a log simulating tool [15] which creates random XML logs by simulating already designed workflow processes based on CPN tools. These tools support the modeling, the execution and the analysis of colored Petri nets. They enable to create simulated logs conform XES. Modifications were brought to these tools to call predefined functions that create enriched logs with respect to our multi-agent organizational log events.

B. Semantic Annotation of event logs

The second step in our approach (see Fig.5) is the semantic annotation of event logs based on OrO ontology. For this purpose we implement a semantic log enricher plug-in in the ProM framework tool [16] which is a "plugable" environment for process mining. The Semantic annotator plug-in which is purely syntactic gets as input an event log and returns a semantically annotated event log based on the OrO ontology as an output.

C. Ontology population (OrO$_p$)

The third step in our approach is the population of the OrO ontology in order to create an organizational knowledge base. So, we add data that are stored in the semantically annotated elements in the event logs as individuals to the OrO ontology. The resulting ontology is called OrO$_p$ and represents the organizational knowledge base. In this context, we implement a conversion plug-in in the ProM framework tool. Fig.6 shows a screenshot of this plug-in. This latter takes the annotated log file and the OrO ontology as input and returns a populated ontology OrO$_p$.

![Fig. 6. Ontology population plug-in](image)

D. Relationships discovery

The final step in our approach is the relationships discovery. For example, we show how we use the organizational knowledge base OrO$_p$ to discover hierarchical relationships in the workflow organizational structure. The discovery is ensured by executing queries on the organizational knowledge base (OrO$_p$). These queries are based on the superior relationship (see Table II). In this context, we develop a discovery plug-in. This plug-in enables the performers’ hierarchy structure discovery by executing SPARQL queries on the OrO$_p$ ontology. In Fig.7, we show a screenshot illustrating hierarchical relationships between five performers and the SPARQL query allowing this discovery.

![Fig. 7. Hierarchical relationships discovery plug-in](image)

V. RELATED WORK

There are already some substantial works in the field of workflow organizational model mining. Ly et al. [17] propose an organizational mining technique based on a decision tree learning approach to discover staff assignment rules which

4http://www.w3.org/TR/rdf-sparql-query/
define the profile of agents capable of and/or eligible for performing an activity. In addition, Van Der Aalst et al. [18] propose mining techniques for social networks based on event logs in order to analyze relationships between performers involved in a business process. We do not deny the importance of these works, but compared to our work, they remain limited in discovering the organizational model due to the lack of semantic description and the limited information contained in event logs. In fact, we focus on the event log which is the main support of workflow mining and we enrich it with additional semantic information about organizational model. This enrichment is ensured by adding three new concepts to the event log: performative, initiator, and Receptor.

Several works propose ontologies specifying organizational model in a workflow. In [19], Filipowska et al. propose organizational ontologies supporting the lifecycle of BPM systems. However, this set of ontologies do not specify communication between performers in a workflow. In [20], Van Der Aalst et al. propose a set of ontologies based on the MXML event log file in order to enrich the event log and support the BPM lifecycle. Among these ontologies, they define an ontology called OriginatorOntology for the organizational model. But, this ontology is restricted only on performers who performed tasks with success. Although these ontologies are interesting, they still unsubstantial to specify the whole workflow organizational model. In addition, they do not cover the lack of automation facing most of workflow mining techniques. Our approach proposes the ontology OrO. This ontology describes the organizational structures and the communication between performers using performatives. Besides, it allows removing ambiguities in event log files by semantic annotations and sharing organizational knowledge in order to discover relationships between performers especially hierarchical relationship.

VI. CONCLUSION

In this paper, we proposed to semantically annotate event log based on the OrO ontology. This allowed us to remove many ambiguities about data in event log and to share organizational knowledge. This latter is ensured by populating the OrO ontology with semantically annotated elements in event log files. Hence, we obtained an organizational knowledge base (OrO, ontology). Based on this knowledge base, we executed queries in order to discover new hierarchical relationships between performers. Our approach was implemented within the ProM framework.

Actually, we are working on testing our plug-ins using qualitatively and quantitatively various set of event logs. Besides, we target applying our techniques to other use cases so that we identify possible discrepancies and make our approach more reliable. Our second perspective consists on providing new organizational mining techniques based on our OrO ontology that are more complex (for example new mining techniques to discover federal and coalition relationships).

REFERENCES