

# An agent-based approach for organizational structures and interaction protocols mining in workflow

Lotfi Bouzguenda<sup>1</sup> · Mahdi Abdelkafi<sup>1</sup>

Received: 26 June 2014/Revised: 3 March 2015/Accepted: 9 March 2015/Published online: 21 March 2015  
© Springer-Verlag Wien 2015

**Abstract** This paper deals with organizational perspective mining in workflow. First, it exposes a comparative study of three representative workflow mining systems: InWolve, WorkflowMiner and ProM. To perform our comparative study, we propose the following quality criteria: the capacity to filter and to convert an events log, the capacity to discover workflow perspectives and the capacity to support their analysis. The main limit of these systems is their inability to deal with organizational perspective mining issue. By organizational perspective, we mean, the social structures defining the activity distribution among actors involved in the workflow (namely federation, coalition, market or hierarchy), as well as the interaction protocols ruling the communications between them (such as contract net, auction or vote). To compensate the previous drawback, the paper proposes an agent-oriented approach to support the organizational perspective mining. More precisely, it proposes an events log model, which extends existing models by taking into account the interactions among actors, thanks to the performative-based agent communication language FIPA-ACL. It also presents a concrete case study to better illustrate the organizational perspective mining and some algorithms for organizational structures and interaction protocols mining. Finally, we describe in detail our disocflow tool to validate our solution.

**Keywords** Organizational perspective mining · Agent approach · Events log model · Case study

## 1 Introduction

The workflow technology provides an appropriate platform to define, implement and execute the coordination of business process activities. This platform is called Workflow Management System (WfMS for short). Most of the WfMSs only focus on design, configuration and enactment steps (Van der Aalst 2005). These tools do not take really the diagnosis step although it is useful for the design step. Very few WfMSs propose a simulation to check and validate the design step and facilitate analysis of data through execution traces. Even if the majority of actual WfMSs [for instance Bonita (Miguel 2003), YAWL (Van der Aalst et al. 2004) and FlowMind (<http://www.flowmind.org>)] collect execution traces of business process instances, they do not offer a support to exploit this important information. However, the Workflow Mining (WM) field has recently emerged as a hot research topic. It is initiated by Wil Van der Aalst (<http://www.processmining.org>). More precisely, the purpose of WM is to analyze the workflow execution traces (or events log) to discover the key workflow perspectives such as the Organizational Perspective (OP), the Informational Perspective (IP) and the Process Perspective (PP), which help the designer to improve or propose a new workflow (Van der Aalst et al. 2003b). These perspectives will be detailed later (see Sect. 2.2).

In the literature, the most of existing workflow mining systems such as InWolvE (Herbst and Karagiannis 2004), WorkflowMiner (Gaaloul et al. 2009) and ProM (Verbeek et al. 2010) only concentrate on the process perspective mining. Mining the OP was neglected in existing WM systems except for ProM system. This latter supports only

---

✉ Lotfi Bouzguenda  
lotfi.bouzguenda@isimsf.mu.tn  
Mahdi Abdelkafi  
mahdi.abdelkafi@hotmail.fr

<sup>1</sup> Miracl laboratory/ISIM/University of Sfax, Route de Tunis, km 10. BP 242, 3021 Sakeit Ezzit, Sfax, Tunisia

the mining of actors, their roles and few kind of social networks based on specific metrics such as handover of work, subcontracting, working together, reassignments and doing similar tasks (Van der Aalst et al. 2009). We later show (see Sect. 4) that mining the organizational perspective could not be limited to the previous elements but should mine more sophisticated organizational structures and interaction protocols as those deployed in real inter- and intra-business processes (Hanachi and Sibertin-blanc 2004). Typically, social networks can appear under execution of activity allocation rules. Besides, many works aim at extending the organizational meta-model associated with the business process design to give a high level of granularity to the staff assignment rules. This is generally performed to enhance the business process performances (Muehlen 2004; Linh et al. 2006). Extending the log files by saving the interactions between performers during the allocation process is neglected. We assume that saving such interactions is a promising field to make the coordination process clearer and more expressive.

In our work, we mean by organizational structures, the social networks or social structures made up of a set of individuals (i.e., the workflow actors involved in the business processes execution) and a set of relations between them (delegation for instance). We also use the typology of social structures (federation, coalition or hierarchy) and interaction protocols (contract net, vote or negotiation) as defended in multi-agent system.

Multi-agent systems are largely used at some phases of the workflow life cycle. Some anterior works use agents to perform some functions in each phase of the workflow life cycle. Agents can make decision with autonomy when they judge it is necessary to perform activity. An agent can receive a signal from other agents or workflow users, and gathers information from data bases or audit trail (Huang 2006). Otherwise, agent systems are largely used in the context of inter-organizational workflow to gather information from distributed environments. This information is required for the decision-making at different phases of the workflow life cycle such as design, execution and monitoring phase (Bui and Lee 1999).

We believe that the reason why the existing propositions only deal with process perspective mining is that they use an events log limited to activities execution and to the actors performing these activities. There is no trace about the interactions among actors.

Giving the previous observations, the problem being addressed in this paper is: “*how to mine sophisticated organizational structures (OS) and interaction protocols (IPr) from events log that integrate interactions between actors?*”

To do this, we will first give a critical and comparative study of WM systems to underline their limits for

organizational perspective mining. Then, we will show how the agent approach provides the appropriate abstractions to model OS and IPr and to easily mine them through enriched events log.

A good application domain justifying the interest of the organizational aspect is the Crisis Management Processes (CMP for short). Indeed, the CMP require cooperation and coordination of several actors distributed and structured as federations or hierarchies for instance. Also, these processes need the use of sophisticated interaction protocols to support consistency of communications between participants and to act effectively to the crisis. In this area, the discovery of the organizational aspect and its analysis seems useful for understanding and explanation of such a crisis and notably the behavior and ability of involved actors. For instance, (Muehlen 2004) gives an organizational meta-model that extends the classical one by a newly defined organizational entity such as the ability of the human resources and their positions to belong to an organizational units. Authors believe that this extension can enhance the staff assignment rules and avoid failure execution results due to the incapacity of the performers. Giving more organizational details with assignment rules can, therefore, better affect activity of the appropriate performer.

In addition, we assist today to the actors’ behavior-driven business process design (known the human perspective in Business Processes) in several conferences such as the Business Process Modeling, Development, and Support, BPMDS’ 2009–2014 (<http://www.bpmds.org/>).

This paper is organized as follows. Section 2 justifies the interest of using WM systems, provides a framework for studying them, and compares three representative systems of the state of art to underline their organizational perspective limitations. Section 3 proposes a new events log model which extends existing models. It starts by justifying the use of agent approach to enrich the log file structure. Then, we underline the limitations of events log models of the previously studied systems before revealing our proposed events log model. The Sect. 4 exposes our running example to illustrate the organizational perspective mining and the proposed algorithms. Section 5 presents the implantation of our discopflow and its evaluation. Section 6 concludes the paper with an outline of the main perspectives in this work.

## 2 Background and motivations

### 2.1 Workflow mining rationale

Workflow mining is mainly advocated under the following considerations:

- *Reverse engineering of workflow perspectives.* The idea is to improve the workflow design based on the mining of workflow perspectives such as the informational, organizational and process perspectives. In the literature, we distinguish several approaches. The Petri net theory approach consists in comparing the WF-net (Workflow net) constructed from events log based on techniques mining with the Wf-net that generates the events log. The goal of this comparison is to adapt/adjust the prescribed processes. In this approach, we can mention the EMiT tool implementing the well-known algorithm alpha (Boudewijn 2004). The heuristic approach deals with noise and incomplete logs. It is based on three mining steps: (a) the construction of a dependency/frequency table (D/F-table), (b) the mining basic relations out of the D/F-table (R-table) and the reconstruction of the WF-net out of the R-table. In this approach, we can quote the Little Thumb mining tool (Weijters et al. 2003). The inductive approach supports the workflow processes mining based on duplicate tasks. It consists of two steps. The induction step that permits to build a stochastic task graph from the events log. The transformation step realizes three sub-steps: (a) the synchronization structures of the workflow instances in the events log, (b) the generation of the synchronization structure of the workflow model and (c) the generation of the model. The proposed mining tool in this approach is the InWolve tool (Herbst and Karagiannis 2004). The *data* mining approach supports the block-structured workflow mining. The tool supporting this approach is called Process miner (Verbeek et al. 2010).
- *Performance analysis of a workflow component* such as activity, actor, etc. For instance, four performance metrics have been proposed for process perspective namely flow time, waiting time, processing time and synchronization time (der Aalst 2005). These metrics permit to enhance the existing processes in terms of complexity, cost, resources and execution.
- *Delta analysis* consists in comparing prescribed models/perspectives (informational, organizational and process) against the deployed to adapt and/or enhance these models. This analysis of differences also allows a comparison of different implementations of a model among various organizations.

The interested reader can find more information about motivations for using workflow mining in Van der Aalst (2011) and Van der Aalst (2012).

## 2.2 Workflow mining systems and quality criteria

We compare some existing and representative workflow mining systems, according to predefined quality criteria.

More precisely, we consider the following WM systems: InWolve (Herbst and Karagiannis 2004), WorkflowMiner (Gaaloul et al. 2009) and ProM (Verbeek et al. 2010; <http://www.promtools.org/prom6/>).

*Remark* Roughly speaking, very few WM systems have been proposed in the literature. It is to note that the EmiT (Boudewijn 2004), Thumb (Weijters and Van der Aalst 2003), MinSocN (der Aalst 2005) and MiMo (Van der Aalst 2003) systems are merged within the ProM system (Van der Aalst 2005). ProM is the well-known Workflow mining system which has several versions. In our study, we are based on the last version of ProM.

It is important to note that the studied systems represent the main existing approaches as presented previously.

Quality measures criteria of a WM system depend on its expectations. Some of the most important related functions are the following:

- Pre-handling of events log;
- Mining of workflow perspectives;
- Analyzing of workflow perspectives.

Let us elaborate each of the above functions.

*Pre-handling of events log* When necessary, WM system must allow the filtering of an events log. The role of this operation is twofold. First, it reduces the noise involved in the separation between the principal and the optional activities and focuses only on completed activities. Often, the noise arises in ad hoc workflow systems and groupware products, which may be based on unstructured process activities. Second, the filtering operation of a log allows the conversion of the log to known format such as XML to facilitate further automated operations.

*Mining of workflow perspectives* Such Workflow mining system must allow the mining of the three interacting workflow perspectives. The Organizational Perspective (OP) has two objectives. First, it structures actors in classes, where each actor plays a specific role (for instance subordinate and chief roles in a strict hierarchical structure). A class is also called an organizational structure defining the interaction space between actors. Second, OP describes the activities allocation policy between actors thanks to the interaction protocols. The Informational Perspective (IP) describes the structure of forms, documents and data, which are consumed and produced by processes. The Process Perspective (PP) defines components activities, their coordination, information and actors involved in each activity.

*Analyzing of workflow perspectives* As mentioned previously, WM system is expected to support various analysis of Workflow components (processes, actors, organizational units, informational resources, etc.), namely the delta and performance analysis;

The functions presented above provide a global view of what should be a WM system. Next we discuss possible quality criteria of a WM system.

Regarding the first function, we have defined criteria to measure the capacity to filter and to convert an events log. The goal of filtering is to decrease the noise elements mentioned above. Concerning the conversion of an events log from given format to other (for instance from text to XML), it permits to facilitate the extraction process of information and then to analyze it.

For the workflow perspectives mining function; we have defined the following criteria to measure the capacity to discover informational, organizational and process perspectives. By informational perspective, the ability to discover the consumed documents and produced documents by a workflow. By process perspective, the ability to discover the activities and their coordination patterns (sequential, parallel, iterative, etc.). Regarding, the organizational perspective, the ability to discover the actors, their roles, their organizational units and the policy of activities allocation (i.e., the employed interaction protocols between actors and social networks describing the nature of collaborations between actors such as federation, coalition, hierarchy, market, etc.).

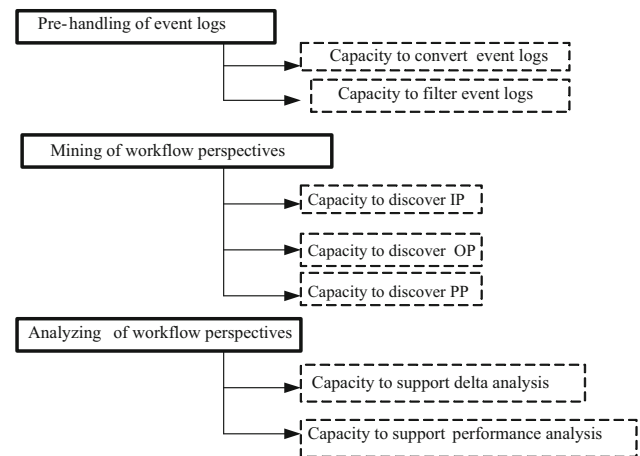
Finally, about the workflow perspectives analysis function, we have defined the criteria to measure the capacity to support workflow perspectives multi-analysis. These include the possibility for designers to make various analyses such as delta analysis and performance analysis to further adjust the components of these perspectives (processes, actors, organizational units and so on).

Besides the above criteria which are not obviously exhaustive, others related to the quality of software in general could be considered, such as usability of interfaces, portability and extensibility. Usability of interfaces measures the easy-to-use feature of the system (a) to design clearly the workflow perspectives with graphic representation (Petri Nets for processes for instance), and (b) to make simulations and animations for detecting some errors and ambiguities.

The functions and the quality measures criteria of a WM system that we have proposed are summarized in Fig. 1.

### 2.3 A comparative study

Due to limitations of space, we examine briefly, in this section, three systems, representative of the existing workflow mining approaches. More precisely, we describe the InVolve (Herbst and Karagiannis 2004), WorkflowMiner (Gaaloul et al. 2009) and ProM (Verbeek et al. 2010) systems, and we compare them according to the previously defined criteria to give a general idea about their abilities in terms of workflow perspectives mining and also their strengths and weaknesses.



**Fig. 1** Functions and quality measures criteria of a WM system

#### 2.3.1 InVolve system

InVolve (Inductive Workflow learning via Examples) system has been developed by Joachim Herbst with C++ (Herbst and Karagiannis 2004). Figure 2 shows the InVolve system architecture.

The InVolve system supports three formats for events log: XML, ASCII and APF (Original format of Adonis WfMS). The proposed mining process by InVolve is composed of two steps:

- The induction step which consists in analyzing the events log to produce a stochastic activities graph. It is performed by the splitSeq or the spliPar components.
- The transformation step which consists in refining the obtained graph according to the following sub-steps: (a) analysis of processes' synchronization, (b) generating the process synchronization model structure and (c) the production of the process model with ADL (Adonis Description Language). It is performed by the SFAtoADL or the SAGtoADL components.

*From a pre-handling of events log point of view:* InVolve system supports the filtering and conversion (for instance from XML to APF) of events log.

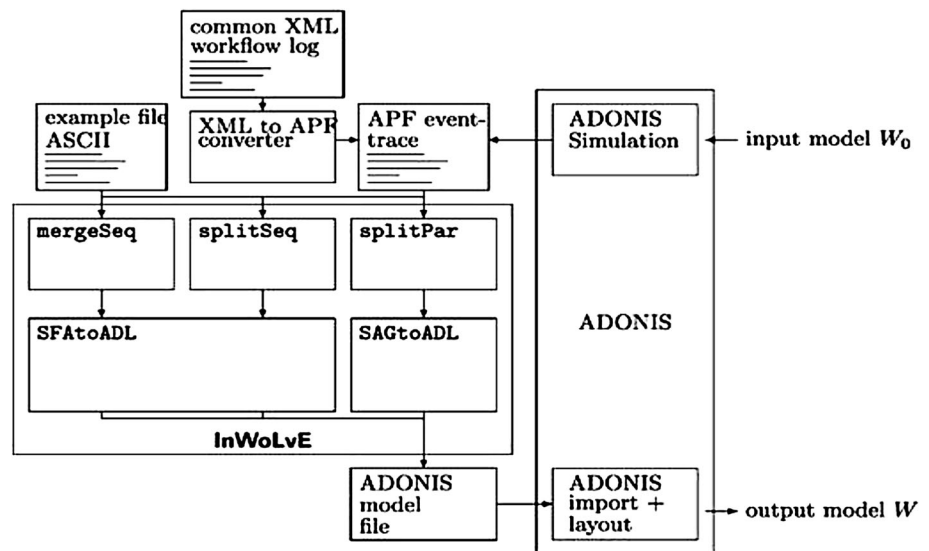
*From a capacity to discover workflow perspectives point of view:* InVolve supports only the mining of process perspective. More precisely, InVolve system creates in the first step a stochastic activity graph from events log and in the second step; it transforms the activities graph into a well-defined process workflow model.

*From a capacity to support multi-analysis of workflow perspectives point of view:* InVolve does not offer any analysis technique for workflow perspectives.

*From a system point of view:* InVolve is not portable since it has been developed with C++ and it does not offer usable interfaces, and finally it is not considered as an



**Fig. 2** Architecture of the InWolve system (Gaaloul et al. 2009)



extensible system since it is not developed around of plugins as the case of ProM system.

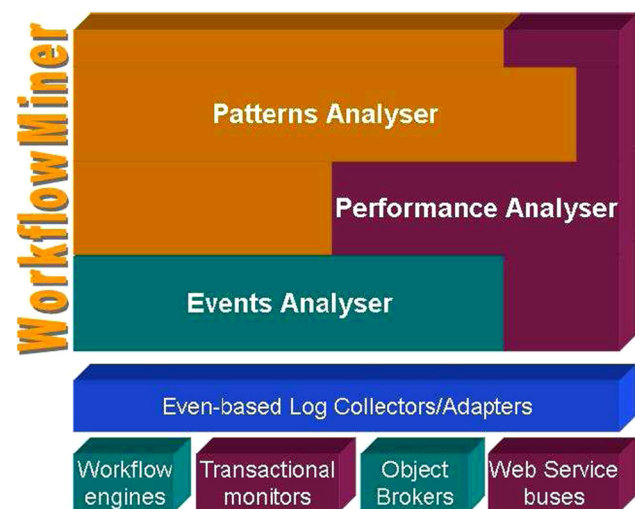
### 2.3.2 WorkflowMiner system

WorkflowMiner system has been developed by Walid Gaaloul in ECOO team of University of Nancy in France (Gaaloul et al. 2009). The aim of this system is to allow the mining of Workflow control patterns from an events log. This is generated by the Bonita WfMS (<http://www.bonitasoft.com/>) to ensure continuous designs that satisfy business processes flexibility issues. Figure 3 shows the architecture of WorkflowMiner system which includes the following components:

- Events-based Log Collectors/Adapters module which consists in loading the execution traces from XML files and adapting them into Prolog predicates format;
- Event Analyzer module which runs the analysis engine of Prolog facts and infers through a statistic technique the casuals' dependencies between events. It constructs the dependencies table, which helps the construction of the workflow graph;
- Performance analyzer module which uses the casuals' dependencies and discovered workflow patterns to measure the performance of workflow.
- Patterns analyzer module which uses the predefined rules that support the discovery of workflow patterns. These rules are reproduced as first-order logic predicates.

*From a pre-handling of events log point of view:* WorkflowMiner supports only the conversion of events log from XML to Prolog events.

*From a capacity to discover workflow perspectives point of view:* WorkflowMiner supports only the discovery of



**Fig. 3** Architecture of WorkflowMiner System (Hanachi and Sibertin-blanc 2004)

process perspective using a statistic technique. More precisely, it allows (a) the “local” workflow patterns mining which covers partial results and (b) the composition of those local discovered workflow patterns iteratively until mining the “global” workflow model.

*From a capacity to support multi-analysis of workflow perspectives point of view:* WorkflowMiner offers only the analysis of performance of those discovered processes. Finally, *from a system point of view:* WorkflowMiner is portable since it has been developed in Java but it does not propose useable interfaces as defended previously and it is not considered as extensible.

*From a pre-handling of events log point of view:* ProM system supports both filtering (LogFilter) and conversion (Conversion plug-in) functions.

Regarding the mining capacity aspect, ProM system ensures the discovery of organizational and process perspectives. With respect to organizational perspective, it leads to discover some elements such as actors, roles and a limited number of social networks. The process perspective is very powerful. It offers a lot of mining algorithms (alpha algorithm, genetic-based algorithm, etc.,) which could be visualized according to different formats (Petri Nets, EPC, etc.,).

2.3.3 ProM system

Process mining (ProM) (Verbeek et al. 2009, 2010) is an extensible system developed with Java at Eindhoven University of Technology. The purpose of this system is to support the maximum number of mining techniques to elucidate workflow perspectives using the offered plug-ins. It also ensures certain flexibility on the input and output formats, thanks to the embedded conversion tools. Figure 4 shows the architecture of the ProM system.

ProM takes as input an events log issued from several actual workflow management systems such as YAWL (Van der Aalst et al. 2004) or Flower (Verbeek et al. 2002). The events log is often represented in XML format and filtered using the component LogFilter to reduce the noise as defined previously. The main plug-ins are:

- Import plug-ins to support a variety of graphic representation such as Petri Net, Social network, etc.
- Mining plug-ins to allow the discovery of workflow models regarding the different perspectives.
- Analysis plug-ins to analyze the discovered models. We can mention the delta and performance analysis.

- Conversion plug-ins to convert a given log file from one format to another. For instance from EPC (Event-driven Process Chain) to a Petri Net format.

Considering the workflow perspectives multi-analysis point of view: ProM offers mainly two types of analysis: delta and performances analysis.

Finally, from a software quality point of view: ProM is portable since it has been developed with Java. It is also extensible since it is possible to add a new plug-in without changing the framework and offers sophisticated interfaces to allow simulation, checking and validation of business processes models.

2.3.4 Comparative table

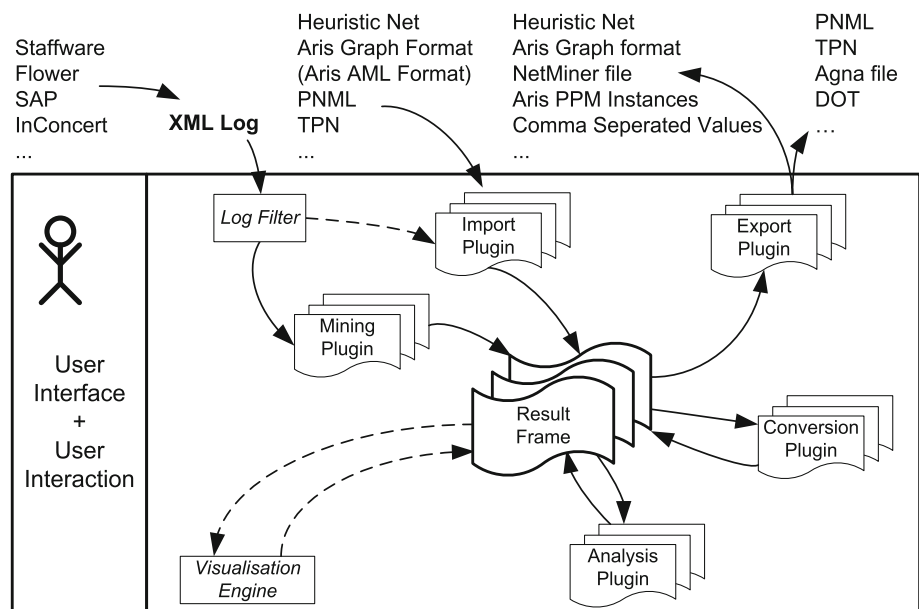
We have compiled our findings in a comparative study shown in Table 1, of the three previously described systems. The comparison is made according to the quality criteria that we have defined such as the capacity to filter and to convert an events log, the capacity to discover workflow perspectives and the capacity to support multi-analysis of workflow perspectives.

Table 1 summarizes the strengths and the weaknesses of each studied Workflow mining system.

According to this table, we can emphasize the following points:

- All the workflow mining systems support the conversion of events log to ease the mining process,
- All the workflow mining systems support the process perspective discovery,

Fig. 4 Architecture of the ProM system (Herbst and Karagiannis 2004)



**Table 1** Comparison of WM systems

Existing WM systems	InWolve	WorkflowMiner	ProM
Pre-handling of events log			
Filtering	+	–	LogFilter
Conversion	From XML to APF or ASCII	Adapters	From EPC to Petri Nets
Capacity to discover workflow perspectives			
Informational perspective (IP)	–	–	–
Organizational perspective (OP)			
Actors	–	–	Plug-in Originator by task matrix
Roles	–	–	+
Org units	–	–	+
Organizational structures	–	–	–
Interactions protocols	–	–	Plug-in social network miner
Process perspective (PP)	Induction and transformation methods	Patterns analyser	Plug-ins alpha, genetic
Capacity to analyze workflow perspectives			
Delta			
IP	–	–	–
OP	–	–	+
PP	–	–	Plug-in Trace comparison
Performance			
IP	–	–	–
OP	–	–	+
PP	–	Performance Analyser	Plug-in a basic statistical analysis
Particularities of system			
Usability	–	–	+
Portability	–	Java	Java
Extensibility	–	–	Framework

- Only the ProM system ensures the mining and the analyzing of organizational perspective (actors, organizational units and social networks),
- The performance analysis of processes is guaranteed by the ProM and WorkflowMiner systems while the delta analysis of processes is only supported by the ProM system.
- The ProM system has usable interfaces and it is the only extensible system.
- None WM system allow the mining of organizational structures and interaction protocols as defended in our work.

Although the effectiveness of these systems (many problems have been addressed and that many solutions have been proposed) are still challenges in the discovery of the organizational perspective as we have defined it. We believe that this mining is very interesting especially in the crisis management processes area. More precisely, this discovery assists crisis managers to (a) build an exact view of reality (i.e., issues, deviations or weaknesses of involved

participants), (b) take suitable decisions and (c) understand and learn from the crisis.

In conclusion, the main limitation raised by the studied existing approaches is the non-possibility to deal with organizational structures and interactions protocols mining except for ProM system. This latter ensures the discovery of actors, their roles and social networks without categorizing them in fine manner (i.e., ProM is not able to identify the different kinds of social networks such as federation, hierarchies, coalition and so on). Indeed, to support this drawback, we need to revisit the log file structure since it is only based on activities traces and their performers. Our idea is to assume that the events log also includes, besides the previous elements, information about exchanges/interactions between actors that are outside the control of workflow management systems. Regarding the others limits, they can be supported without extending the events log. By looking at the agent approach, the organizational structures and interactions protocols are considered as first class entities and several works have been proposed to

**Table 2** Some examples of FIPA-ACL performatives

Performative	Passing information	Requesting information	Negotiation	Performing actions	Error handling
Accept-proposal			*		
cfp			*		
Confirm	*				
Propose			*		
Reject-proposal			*		
Agree				*	
Cancel		*		*	
Failure					*
Refuse				*	
Subscribe		*			

The use of performative facilitates the design of agents because it structures their behavior by distinguishing within the communication, the message of its intention and, therefore, the information, respectively, of the activity to be implemented to make and interpret this information

address the engineering issue such as specification, validation and implementation of these entities.

We believe that we can benefit from this approach for the following reasons: the agent approach considers every workflow actor as a representative entity of workflow process and responsible for a mission that is able to perform independently and in cooperation with other actors. It also provides a natural framework for modeling the interaction (cooperation, coordination and communication) at a high level between the different entities like communication language FIPA-ACL.

In the following sections, we present our approach to deal with discovery of organizational structures and interaction protocols applied to the crisis management field.

### 3 Our proposed events log model

#### 3.1 Motivations for using agent approach

We believe that the agent approach helps the organizational perspective mining in workflow thanks to the following high-level features.

- Natural abstractions to deal with cooperation: a lot of sophisticated protocols like contract net protocols, auctions and negotiation mechanisms are available and could be used to coordinate processes (Bouzguenda et al. 2008). Agent approach also provides organizational concepts to abstract and structure a system as a computational community made of groups, roles and interactions.
- Social abilities of agents also facilitate the cooperation needed to enact complex workflow and to provide an abstraction to high-level concepts like commitments, reputations and so on.

In our work, we concentrate on the social abilities and notably on the three following multi-agent concepts:

- Performative-based FIPA-ACL (<http://www.fipa.org>) that defines clearly the semantic of messages and namely the agents' intentions (delegate, subcontract, negotiate...) (Searle 1975, 1969; Kaplan 1990; Winograd 1988) (see Table 2).
- Interaction protocols that permit the identification of some dynamic coordination structures where each actor plays a role to obey to precise rules (auctions, contract nets, and so on) (see Table 3);
- Organizational structures (hierarchy, coalition, market, etc..) (see Table 3) that model the behavior of the actors' group, i.e., they describe the macro-level of the coordination among actors in terms of externally observable behavior independently from the internal features of each participating component.

To the best of our knowledge, the agent approach has been widely used to study and implement business processes but has never been used in workflow mining. The interested reader can find more information about our previous contribution reflecting agent approach in Workflow mining in (Abdelkafi and Bouzguenda 2010).

#### 3.2 Key requirements for an events log model

The purpose of this section is to justify the key requirements that we propose for describing the events log model. Indeed, according to the business process perspective definition (given in Sect. 2.2), a business process perspective refers to perspectives such as the informational perspective and the organizational perspective (who does what and how). For that purpose, we have chosen the requirement "refer to the workflow



**Table 3** Some Organizational structures (OS) and interaction protocols (IPro) in the multi-agent field

OS/ IPro	Definition	Graphic representation
Strict Hierarchy	It's a flat hierarchy at single level. A single agent has the authority to catch decision while all other members are his subordinates.	
	A federation is a set of groups. Each one is represented by a delegate. This latter plays the role of mediator between its group and the exterior.	
Contract Net Protocol, CNP	In the CNP, one agent takes the manager role which desires to have one or more agents (the participants) to execute some activities. Each participant submits its bid. The manger selects the well offer and rejects the others bids.	

perspectives” to have a comprehensive and complete description of business processes. Second, it is interesting that the events log model is simple with minimal concepts to reduce the pre-handling of the events log (as mentioned in Sect. 2.2) and consequently to facilitate the mining process of workflow perspectives. Third, the events log model must answer to the two main questions: What is the average time for performing an activity? How many activities are performed without interruptions? By given these considerations, the events log must elicit the TimeStamp and the EventStream for each activity. Finally, the events log model must be

described with a standard representation to ensure the usability of the solution. Next, we present further details about these requirements.

- Usability: an events log model must not only be easy to understand but also easy to exploit by the user (i.e., based on standard format like XML).
- Refer to the three workflow perspectives: an events log model must refer clearly the three complementary workflow perspectives as defined above.
- Simplicity: an events log model must define the core concepts of the three workflow models. In other words,

it must not be presented at a very detailed level (fine-grained) or too specific to a given process.

- **TimeStamp**: an events log model must provide a start time and an end time for each activity and as a consequence it is possible to measure service times and the utilization of the workforce.
- **EventStream**: an events log model must precise the event type for each activity such the aborted, failed, completed, compensated, etc.

### 3.3 A comparative study of existing events log models

Because of space constraint, we only give in this sub section a comparative study of these events log models according to the previous requirements (see Table 4).

This comparative study of events log models of studied WM systems calls for five remarks. First, the usability criterion is guaranteed by the all events log models since they all described with XML standard. Second, the process perspective is supported by all the models while the organizational perspective is only supported by ProM. Third, the organizational structures and interaction protocols mining, as defined in Sect. 3.1, are not supported by all the models. Fourth, all the log models are simple. Fifth, the two last requirements EventStream and TimeStamp are included in the majority of events log except in the InWolve system.

In conclusion, we noted that the existing events log models do not allow in any manner the discovery of the organizational perspective. Even if the XES events log model, used by the ProM 6 (Verbeek et al. 2010), is powerful, it is not, in our opinion, described with the adequate attributes to deal with organizational structures (e.g., federation, coalition) and sophisticated interaction protocols discovery (e.g., voting, auctions). Thus, a new

model of events log is proposed to fulfill the requirements described previously.

### 3.4 The proposed events log model

This model is visualized in the UML diagram of Fig. 5. According to this UML model, a Process is composed of one (or several) Process Instance(s) (or cases). Each Process Instance is composed of one (or several) Event(s).

Each Event is composed of one (or several) EventLine(s) in which one makes references to the following elements:

- An activity: it is described through three attributes—Act-Name, EventStream and TimeStamp,
- A document: it is described through the Doc-Name attribute,
- An actor: it is described by two attributes Act-ID and Act-Name. Each actor is a member of the organizational unit and plays a specific role,
- A role: it is described through the Name attribute,
- An organizational unit: it is described by the Org-Unit-Name attribute,
- A performative which is described through the Perf-Name attribute: this element is widely used by multi-agent community and notably in agent communication language (ACL, <http://www.fipa.org>). This latter is used for exchanging information, intentions or goals. ACL is composed of a set of performatives (e.g., propose, accept, inform, etc.) and a set of associated parameters that specify how to react after sending or receiving a message.

In this model, the consumed and produced documents are, respectively, described by the Has\_consumed\_doc and Has\_produced\_doc relationship. The initiator actor and the receiver actor are described, respectively, by the Has\_initiator\_Actor and Has\_receiver\_Actor relationship.

**Table 4** A comparative study of events log models of studied WM systems

Events log of studied WM systems	InWolve	WorkflowMiner	ProM
Usability	+	+	+
Referencing the three workflow perspectives			
IP	-	-	-
OP			
Org structures	-	-	±
Interaction protocols	-	-	-
PP	+	+	++
Simplicity	+	±	+
Event stream	-	+	+
TimeStamp	-	+	+

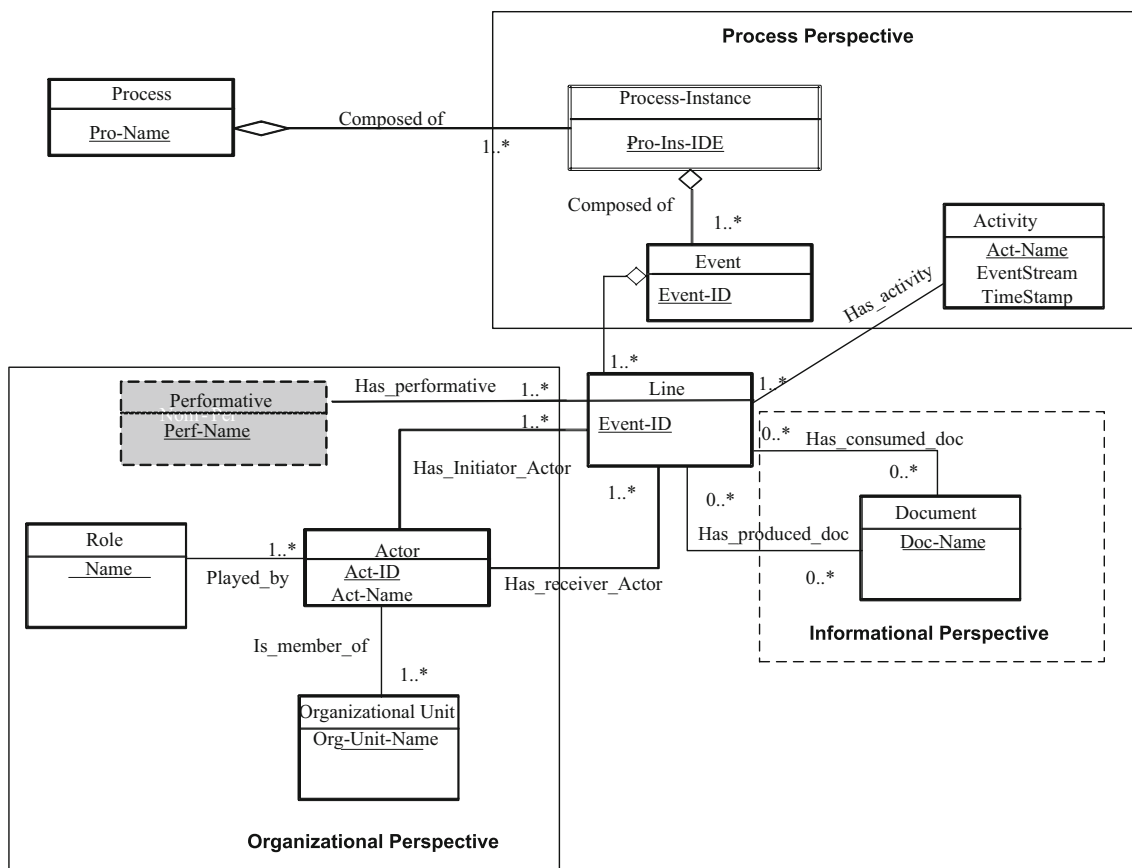


Fig. 5 The proposed events log model

Table 5 A simplified extract of the log file

Cases (process instances)	Performative	Activity	Initiator	Receiver
1. C1	Execute	Release secure perimeter	Mahdi	System
2. C1	Delegate	Clear trees	Mahdi	Salim
3. C1	Delegate	Clear trees	Mahdi	Walid
4. C1	Inform	Clear trees	Salim	Mahdi
5. C1	Inform	Clear trees	Walid	Mahdi
6. C1	Cfp	Hospital support	Malik	Sami
7. C1	Cfp	Hospital support	Malik	Amal
8. C1	Propose	Hospital support	Amal	Malik
9. C1	Propose	Hospital support	Sami	Malik
10. C1	Accept proposal	Hospital support	Malik	Amal
11. C1	Reject proposal	Hospital support	Malik	Sami
12. C1	Execute	Hospital support	Amal	System

Remark. In our context, each event refers to the performed activity with achieved state. More precisely, the same activity can be described around several lines and according to a given allocation protocol. For instance, as shown in Table 5, the activity “Hospital support” is repeated in seven lines describing the contract net protocol.

#### 4 Organizational perspective mining

The objective of this section is to introduce a running example “management process of forest fire crisis” to better illustrate our approach in dealing with mining of organizational structures and interaction protocols. Then, it exposes a simplified version of our events log allowing the

mining of some organizational structures (strict hierarchy for instance) and interaction protocols (contact net for instance). Finally, it provides some algorithms devoted to discover these elements (OS and IPr).

#### 4.1 Running example

In our work, we consider the well-known crisis scenario called, “forest fire”. In this scenario, the authorities set up a crisis unit consisting of DFR representatives (the Department of Fire and Rescue), ambulance and emergency service, police force, DDE (Directorate Department of Equipment) and hospitals. Assume the occurrence of forest fires after a strong wind episode impacting the forest (F). The following ad hoc process is established to deal with this crisis. Police is involved first to *release secure perimeter* to allow others to act. DDE then comes to *clear trees* that obstruct traffic. DFR sends three teams of firefighters (STF : E1, E2 and E3): the first *evacuates some people* from the village of Tabarka, Tunisia threatened by the fire, while the second and third teams involved deal with the two identified houses. DFR informs the crisis unit of the presence of 10 badly burned persons. The crisis unit launches a tender to area hospitals H1 and H2 to request one with available beds and burn specialists. It selects the hospital H1 and sends the ambulance to transport the wounded. Crisis unit coordinates the crisis resolution by submitting electronic records to each representative mission (e.g., firefighter, ambulance, police force). Electronic record indicates a geographical area to be treated, the action to be taken and the final state of the action (“success”, “Failed”). The process described above is experiencing a rebound to take into account: the E2 team is declared “Failed” and the crisis unit decides that team E1 has a new mission to help team E2 after evacuating the people of the village.

The major reason why we chose this case scenario is as follows. We believe that the management of this crisis requires the cooperation and coordination of several distributed participants (DFR, DDE, Polices,...) working in dynamic and open environment organizations. Also, new participants (i.e., volunteer organizations) can dynamically join existing teams; organizations already deployed can restructure and adapt some of their activities.

#### 4.2 Examples of organizational structures and interaction protocols mining

The Table 5 presents an extract of log file of running example and shows how we can discover easily the strict hierarchy structure and the contract net protocol for instance.

More precisely, the white part of the log file (line 2 → line 5) describes the strict hierarchy structure between

the actors Mahdi, Salim and Walid. In the same case C1, the grayed part of this events log shows (line 6 → line 12) the employed contract net protocol among the actors Malik, Amal and Sami. Indeed, Malik actor proceeds by a Call for proposal (Cfp) to deal with the “Hospital accommodation” activity. While the other actors Amal and Sami propose their bid. Finally, Malik notifies each participant either by acceptance (accept proposal) or by rejection (reject proposal).

#### 4.3 Algorithms for organizational structures and interaction protocols mining

The originality of our work is the mining of organizational structures and interaction protocols in contrast to the approaches we have discussed above. Indeed, (Van der Aalst et al. 2009) means by social network, a graph whose arcs represent role relations which are based on different metrics like causality, handover of work, subcontracting and so on. To infer these metrics from a given events log, authors propose mathematic formulations to address the problem of social network’s mining (Linh et al. 2006) uses delta analysis to compare the staff assignment rules prescribed under the design step to the staff derived from audit trail. This comparison is based on the different entities of organizational model. The principle is to accord for each performed activity a growing decision tree. The root of the tree shows two performer sets: the first one represents the class of potential activity performers and the second represents the class of non-performers. To each tree level, we find one entity from the organizational model which is selected as a testing attribute. To verify the performance of the staff rule assignments, authors infer all activity performers from audit trail and then compare them to each attribute testing from the decision tree.

In our work, we propose some algorithms to discover social networks from audit trail. We mean by social networks the organizational structures (OS) and interaction protocols (IPr) of multi-agents society. OS and IPr can describe the actor behaviors during the fulfillment of the staff rule assignments and they will explain many deviations on the schema prescribed by the designers. For example, the absence of such performers requires the re-assignment of the task to other performers. This fact is approved by the discovery of hierarchy structure. Besides, the coordination of complex activities requires additional staff assignments and cooperation with different organizations. This fact is explained by the discovery of federation structure or contract net protocol. In what follows, we give the three algorithms that we have proposed to deal with organizational structures and interaction protocols mining.

### 4.3.1 Strict hierarchy mining algorithm

```

ALGORITHM Mining_Strict_Hierarchy
INPUT
L: LogFile
OUTPUT
List_Strict_Hierarchy: GraphList % Strict Hierarchies
List of G.
Local Variables:
S: List of Heads % List of source Heads in G (Heads
without fathers)
G (N, A): graph % partial graph from L limited to
relations of delegation.
Begin
List_Strict_Hierarchy  $\leftarrow \emptyset$ 
Step 1: Build G which contains only relations of
delegation of L
Partial Graph P (L, "delegate", G (N, A))
Step 2: Checking
If Arborescence (G)) then
    List_Strict_Hierarchy  $\leftarrow G$ 
Else
    S  $\leftarrow$  Sources (G) %computes of sourcesHeads list of G
    For each Head s of S Loop
        Extract the connex component CC of G from source S
        If Arborescence (CC)) Then
            Add CC to List_Strict_Hierarchy
        End If
    End Loop
End If
End Algorithm

```

This algorithm can be resumed according to the three steps. The first one consists in extracting from the events log the partial graph  $G$  limited to the relations of delegation. The second one consists in checking if  $G$  is a tree structure. If it is the case, it is then about a strict hierarchy and we turn  $G$ . The third step corresponds to extract each connected component  $CC_i$  from  $G$ , and to turn over the whole of those which are tree structures and thus hierarchies.

Algorithm complexity can be measured according to the two criteria: time and space. The complexity of Strict\_Hierarchy\_Mining Algorithm "SHMA" should be measured at each step of the algorithm. The time or space complexity will also be the sum of complexities of the three steps. To measure time complexity of SHMA, two constants "a" and "b" are required. These constants depend on the elementary operations execution time. We start the complexity measure by the step 1 of the SHMA algorithm, which takes as input a log file "L" of size "N". Note that in all cases one should browse  $N$  elements. To each of these elements, we should do at maximum one or two elementary operations, that are first the comparison of each

performative of  $L$  to the delegate performative and under comparison we add or not the dataset to the partial graph  $G$ . Consequently, the time complexity of step1 is  $Ct(S1, LF) = 2aN + b$  and we can conclude that the complexity is linear of order  $O(N)$ . Two elements can increase or decrease complexity: the size of  $L$  and the result of the comparison operation. The second step consists of browsing all nodes of the partial Graph  $G(LF, Delegate)$  and executing two elementary operations. For a list of nodes " $S_n$ " of size ' $M$ ', the time complexity is linear (like step 1) and given by  $Ct(S2, S_n) = 2aM + b$ . Finally, as mentioned above, the total complexity will be given by  $Ct(SHMA) = Ct(S1, LF) + Ct(S2, S_n)$ . The measure of space complexity is based on the affectation number of local variables. At each elementary operation, we should perform the memory allocations that are required. Indeed, to an input of size  $N$ , a complexity space of step1 will be  $Cs(S1, LF) = (N - 1) + k$  where  $k$  is a constant depending on the successful condition. In the same manner, the complexity space of step2 will be  $Cs(S2, S_n) = 3(M - 1) + k$  and finally the complexity of the algorithm at space will be  $Cs(SHMA) = N + 3M + 2k - 4$ .



### 4.3.2 Federation mining algorithm

#### ALGORITHM Mining\_Federation

##### INPUT

L: LogFile

##### OUTPUT

FederartionG: Boolean % to check if G corresponds to federation.

##### Local Variables

Fed, Rep: Heads % Fed is a federate of the structure and rep is a representing

ER: HeadList % the set of representing of federations

EF: HeadList % the set of federates

G(N, A):graph % graph of interactions from L

Gd(Nd, Ad): graph % LE Partial Graph of interactions

Prop2, Prop3: Booleans

L: Line % line of log file

##### Begin

federationG  $\leftarrow$  False

Step 1: Build the Graph G from L without restrictions

PartialGraphP(L, "delegate", G(N, A))

Step 2: g is connected and symmetric

If (Connected (G) and Symmetric (G)) then

Step 3: Restriction of G to delegations arcs

PartialGraphP (L, "delegate", Gd(Nd, Ad))

Step 4: Check that G is not Hierarchy

If (Arborescence (Gd)= False) then

Step 5: Build the representing and the federates

Check P2 and P3

ER  $\leftarrow$  set of articulation points in G

EF  $\leftarrow$  N

If (ER  $\neq$   $\emptyset$ ) then

Prop2  $\leftarrow$  true

While (ER is not entire explored)  
and(Prop2=True) Loop

Rep  $\leftarrow$  new Head of ER

Each representing has a federate

If (List Children (Rep, G)  $\cap$  EF  
 $\neq$   $\emptyset$ ) then

Prop2  $\leftarrow$  False

End If

End Loop

Prop3  $\leftarrow$  true

While (EF is not entirely explored)  
and(Prop3=True) Loop

Fed  $\leftarrow$  new head of EF

Each federate has one father  
which is its representing

If (ListFathers(Fed, G)  
ListFathers(Fed, G)|=1) then

Prop3  $\leftarrow$  False

End If

End loop

If (Prop2 and Prop3) then

Federation  $\leftarrow$  True

End if

End if

End if

End if

**END Algorithm**

$\subset$  EF and

We describe the principle of the federation mining algorithm according to the five steps. The first one consists in extracting from the events log the partial graph  $G$  of interactions. In the second step, if  $G$  is connected and symmetric then to carry out step 3, if not it is not about organization of federation and thus to turn over “false”. The third step consists in extracting from the events log the partial graph  $G_d$  limited to the relations of delegation. The fourth step corresponds to verify if  $G_d$  is not a tree structure then to carry out step 5 else it is not about organization of federation and thus to turn over “false”. The last step consists in building the whole of the representatives and federated.

According to the manner followed to measure the complexity of SHMA, we also can find the same results on the measure of Mining\_Federation algorithm ‘MFA’ complexity. To deal with the time and space complexity,

we should know the size of the input and the number of elementary operations to each step. According to step1, to each event of  $L$  we have just one iteration to do consequently one affectation to the Graph  $G(N, A)$ . Therefore, the time complexity is defined as  $Ct(S1, LF) = aN + b$ . In the same way, we can infer the complexity of the other steps, and finally the time complexity of FMA will be the sum of all complexity steps. To measure the space complexity of FMA we need to get the number of variables for each step of the algorithm. To an input of size  $M$ , the space complexity will, therefore, be the number of variables multiplied by “ $(M - 1) + b$ ”.

#### 4.3.3 Contract net mining algorithm

---

```

ALGORITHM Mining_Contract_Net_Protocol
INPUT
C: Case
GM(R, M0): Graph
OUTPUT
Compatible: Boolean
Local Variables
L: Line % Line of Case C
S: ListofPerformatives % List of Performatives extracted
from List of performatives extracted from C
CurrentInitialization: Heads % current Heads in GM
A: Arc % an arc from CurrentInitialization
BEGIN
S ← ∅
For each line L of C Loop
    Add to S(L.performative, L.Sender, L.receiver)
End Loop
CurrentInitialization ← M0
APerformative ← First(S)
Compatible ← True
While ((Compatible=True) and (APerformative =
∅) and (Not(TerminalNode(CurrentInitialization))) Loop
    If (an arc has initial limit CurrentInitialization
labelled as APerformative) Then
        APerformative ← goto the next performative in S
        Current Initialization ← Head with terminal
limit of arc a
    Else
        Compatible ← false
    End if
End loop
If ((Compatible=True) and (APerformative ← ∅) and
((TerminalNode(Current Initialization))) Then
    Compatible ← True
End if

End Algorithm

```

---

According to this algorithm, we build in the order of their execution the sequence **S** of performative present of case **C**. Then, we parse in parallel the **GM** graph starting from **M0** (we assume that we dispose the Petri net describing the contact net protocol and its corresponding marking graph) and the sequence **S** as long as the performative one considered in **S** is in conformity with the label of the arc traversed in **GM**. Finally, if we reach a head terminal node in **GM** at the same time we have exhausted **S**, then **C** is considered to conform to protocol **P** (i.e., contract net).

We can measure the complexity of this algorithm according to the principle used to measure the two first algorithms' complexity. We note that the complexity function in the worst case of the three algorithms will be  $F_{max}: N \rightarrow R$  defined by  $F_{max}(N)$  which is the maximum elementary operations required to execute a program on an input of size  $N$ . Our algorithms have a linear complexity and will finish at order  $O(N)$ . No better performance can be reached since we cannot operate dichotomy search. It is necessary to browse all the events of the log file that has amounts of at least  $N$  elementary operations. In theory, limits of our algorithms may occur when they turn on an input that has a large size also small-capacity computers. Input of large size increases the memory allocation and the response time. Typically, our algorithms can burn normally on an input which does not exceed 1000 events. Our algorithms are simplified and they contain minimum possible affectations and iterations. In other words, noise and redundant datasets can affect the performance of our algorithms by increasing response time and allocation memory space.

## 5 The discopflow tool

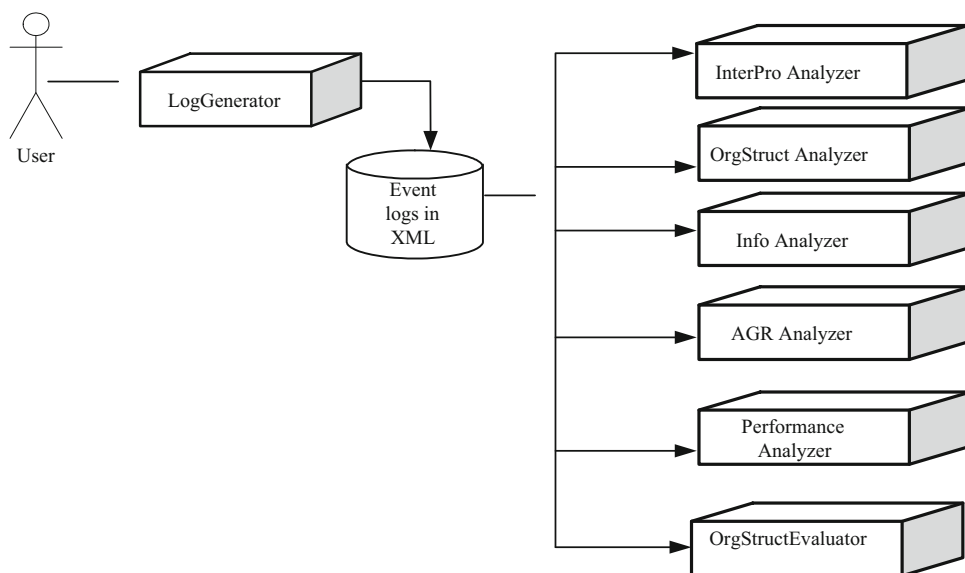
The purpose of this section is to present our discopflow tool for supporting organizational structures and interaction protocols mining in workflow. More precisely, we first present its general architecture and then we expose its interfaces. Finally, we give its evaluation according to the two points of view: quantitative and qualitative.

### 5.1 General architecture

The Fig. 6 visualizes the modular architecture of the discopflow tool. It is structured around several modules and one events log described in XML compliant to the proposed events log model. More precisely, the developed modules with Eclipse platform are LogGenerator, InterPro Analyzer, OrgStruct Analyzer, Info Analyzer, AGR Analyzer, Performance Analyzer and OrgStructEvaluator. Next we discuss in detail each module.

- LogGenerator. It generates in automatic way the events log according to the proposed events log model under the XML format.
- OrgStruct Analyzer. It mines the organizational structures such as strict hierarchy, relaxed hierarchy, federation, coalition and so on.
- InterPro Analyzer. It supports the interaction protocols mining such as contract net, auction, vote and so on.
- Info Analyzer. It mines the consumed and produced documents.
- AGR Analyzer. It gives a graphic representation of each workflow actor according to AGR model. The Agent, Group and Role (AGR) model is one of the

**Fig. 6** The modular architecture of discopflow tool



frameworks proposed to define the organizational dimension of a multi-agent system. According to this model, the organization of a system is defined as a set of related groups, agents and roles.

- Performance Analyzer. It gives a statistical data about the average execution time of an activity, the number of suspended/achieved activities by a given actor, the correlation between interaction protocols and event stream of an activity and so on. This module is in progress.
- OrgStructEvaluator. It evaluates the discovered organizational structures according to three specific quality proprieties namely robustness, flexibility and efficiency. This module is in progress.

## 5.2 Implementation

This work has been implemented as part of the discopflow project, whose objective is to support the mining, in conjoint manner, of the three complementary workflow perspectives: the organizational, informational and process perspectives (Abdelkafi et al. 2011).

The current version of discopflow consists in mining the organizational structures and interaction protocols in workflow.

Discopflow has been developed using Eclipse platform, which allows the development of extensible applications using free plug-ins (<http://www.eclipse.org>). We have tested another running example “water distribution crisis management process” (Abdelkafi and Bouzguenda 2010). Figure 7 shows the general interface of our discopflow tool. This interface helps the user to (a) open the enriched

events log file and (b) have an idea about the main functionality of the discopflow tool. This interface also provides some icons and buttons to facilitate the organizational mining process.

Before starting the organizational perspective mining, the user can see the general data about the events log (business processes and instances/cases), the organizational perspective (actors, performatives, organizational units and roles), the informational perspective (consumed documents and produced documents) and the process perspective (activities, timestamp and event stream).

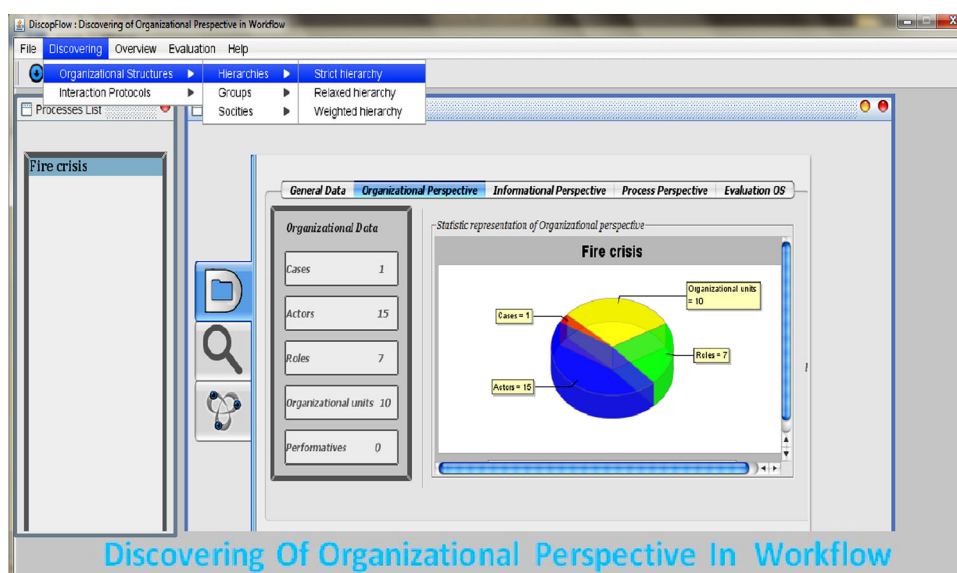
We have also developed a LogGenerator which generates automatically process cases (or instances) conforming to the proposed events log model under the XML format and according to the two modes: random (see Fig. 8a) and driven by the user (see Fig. 8b).

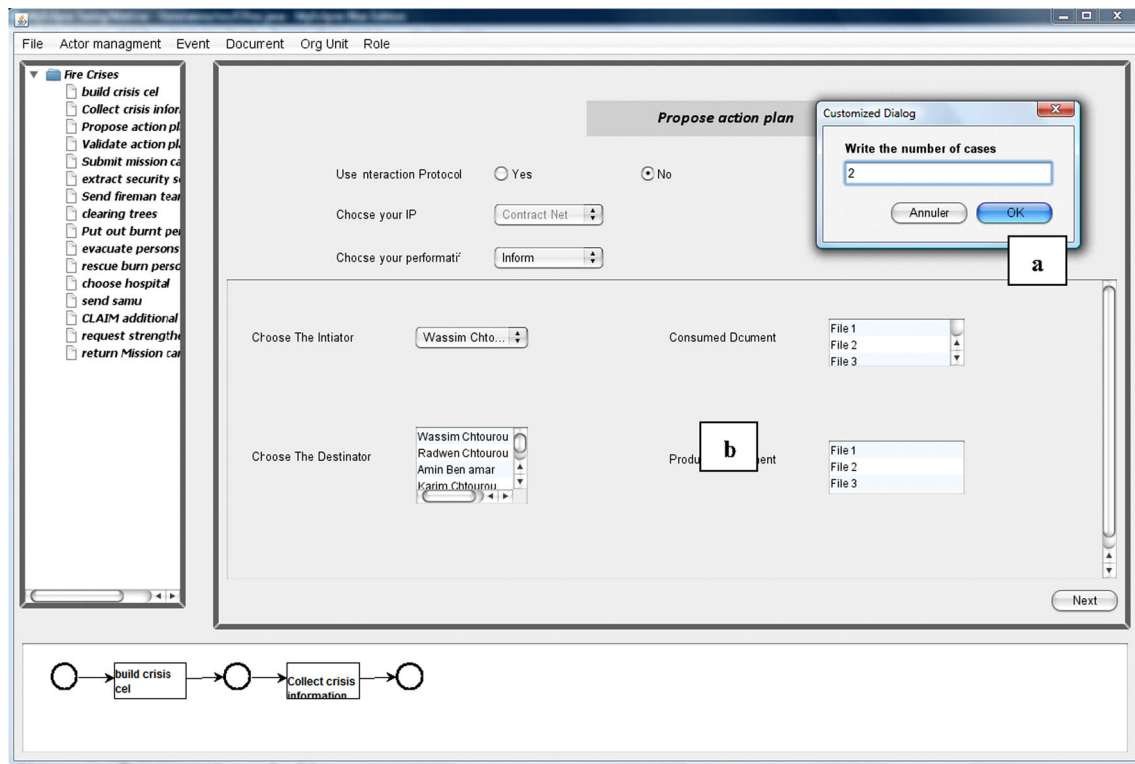
Besides the synthetic data generated by our LogGenerator tool, we have also used data from real-life application about the resolution process of water distribution crisis. The goal of this experimentation is to validate our approach for organizational structures discovering and evaluating. Thus, we have noted the following observations:

*Regarding the discovering of organizational structures*, we have noted the lack of federation structure which is, in our opinion, very useful to better respond to the crisis. More precisely, the actors group specialized in investigation about the incident has no relation with the second actors’ group specialized in reducing the risk of contamination. There is no interaction/exchange between the delegates representing the groups.

*Regarding the evaluation of discovered organizational structures*, even not addressed in this paper, we have also noted that all the deployed OS are not flexible due to the

**Fig. 7** The general interface of Discopflow





**Fig. 8** Overview of LogGenerator tool

lack of coordination relationship between actors. We believe that the OS must have the capacity to adapt to the unexpected executions by adding a new interaction links between actors. This growth avoids any kind of blockage. Based on this evaluation, we have noted that it is very important to address some activities by priority. (a) Reduce the risk of contamination, (b) dealing with contaminating persons, (c) avoid the risk of panic and (d) find more ways to decontaminate all people. Of course, these observations can bring some adjustments or modification on the schema of the prescribed processes.

To conclude, both experimentations (from synthetic data and real-life application) are interesting but we would like to underline first that the real data cannot contain all possible organizational structures and interaction protocols and consequently, we cannot test our algorithms that we have proposed. Second, the collection of execution traces can cause legal problems. The access to the data from real-life application is still difficult because some data are considered confidential (like the origin of the crisis, the number and identity of victims and so on) and that often the authorities refuse to disclose them for political reasons. In one word, our main objective is to offer authorities a system of crises analysis through the discovery and evaluation of organizational structures to help them to understand the

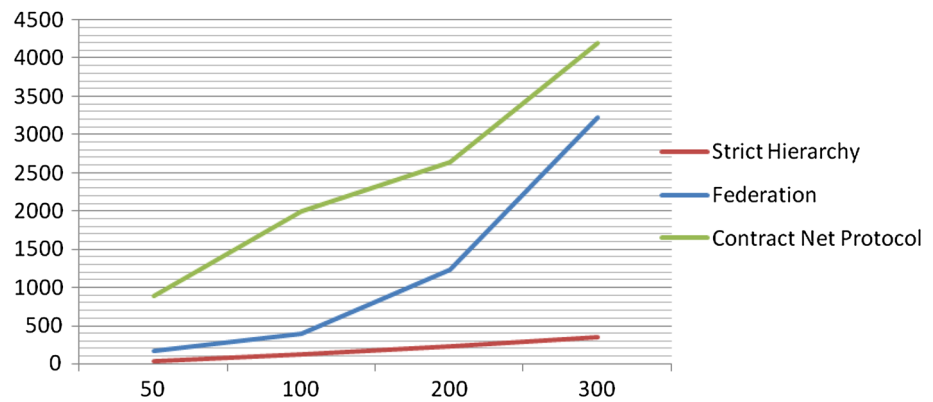
crisis, explain it and take the appropriate response mechanisms.

### 5.3 Quantitative evaluation

The performance of the discopflow is not our goal to be reached since it depends on the particularity of case study and experimentation material. The tests were performed under Windows 7 with a microcomputer having micro-processor dual core t5700 with 2 gega of ram. More precisely, we have realized some measurements of performances, non-exhaustive, but sufficient to give an order of the response time of the discopflow for significant values of the parameters involved in the discovering process. We have varied two parameters: the size of log file (50, 100, 200 and 300) and the type of OS or IPr discovery (strict hierarchy, federation and contract net protocol). The obtained results are rather reasonable since they are measured in milliseconds (see Fig. 9).

The differences in response time between the organizational structures and interaction protocol is explained by the particularity of our case study. Also, we have noted that the strict hierarchy discovery time takes less time than the federation and contract net protocol since the number of performative “delegate” is limited.



**Fig. 9** Response time of discopflow

#### 5.4 Qualitative evaluation

Regarding the qualitative evaluation, our tool presents the following features:

*Usability:* discopflow offers its users a simple and ergonomic environment for discovering and analyzing discovered organizational structures and interaction protocols.

*Portability:* discopflow is portable with any execution platform (Windows, Unix, MAC, etc.,) since it is developed using JAVA language.

*Extensibility:* discopflow is extensible because it is developed as a plug-in under the Eclipse development environment.

## 6 Discussions and conclusion

This paper has presented a comparative study of few existing workflow mining systems according to the quality criteria that we have identified. Although these systems are powerful, they suffer, however, from the organizational perspective discovery as defined previously. More precisely, this paper has:

- defended the use agent technology and notably the organizational dimension in the multi-agent systems to deal with the organizational structures (OS) and interaction protocols (IPr) mining issue;
- defined an events log model having a multi-perspectives view to allow discovering conjointly the three workflow perspectives;
- exposed the proposed algorithms and their complexity dealing with OS and IPr mining;
- presented a case study to better illustrate the proposed solution;
- shown the modular architecture of our discopflow (Discovering of organizational perspective in Workflow) prototype and its some interfaces developed under the eclipse platform.

In the literature, we distinguish three main approaches that address the workflow mining issue.

*The statistical approach.* It focuses on the process perspective mining. The works under this approach adopt the statistical paradigm. The main contribution of this approach is probably made in the WorkflowMiner system (Gaaloul et al. 2009). More precisely, this system allows, at the first time, the workflow control patterns mining which covers partial results. At the second time, it composes them in iterative way to build the global workflow model.

*The inductive approach.* It also concentrates on the process perspective mining. The main contribution that we can mention is the InWolve system (Herbst, J., Karagiannis, D.: Workflow mining with InWoLvE. Computers in Industry—Special issue: Process/Workflow mining, Volume 53 Issue 3, Elsevier Science Publishers B. V. Amsterdam, April 2004). This latter proposes two steps for discovering of process: induction and transformation. The first one consists in analyzing the events log to produce a stochastic activities graph. The second step consists in refining the obtained graph thanks to the transformation rules.

*The Petri net theory and heuristic approach.* Several systems adopting this approach have been proposed to deal with workflow perspective mining (Boudewijn 2004; Weijters and Van der Aalst 2003; Van der Aalst 2005; Van der Aalst 2011). Among them, we can mention the EmiT, Thumb, MinSocN and MiMo systems but the main contribution in this approach is the ProM system. This latter offers several plug-ins allowing the discovery of workflow perspectives, the graphic representation of discovered models (Petri Net, Social network,...) and the analyzing of them according to the Delta and performance analysis. For instance, regarding organizational perspective, ProM provides three methods: default mining, mining based on the similarity of activities and mining based on the similarity of cases. On the one hand, this solution only supports the classical elements such as roles, organizational units and social networks. On the other hand, it does not exploit the agent approach and as hence it does not support the

**Table 6** Comparison between disopflow and studied systems

Existing WM systems	InWolve	WorkflowMiner	ProM	disopflow
Pre-handling of events log				
Filtering	+	-	+	+
Conversion	+	+	+	-
Capacity to discover workflow perspectives				
Informational perspective (IP)	-	-	∓	+
Organizational perspective (OP)				
Actors	-	-	+	+
Roles	-	-	+	+
Org Units	-	-	+	+
Org Structures	-	-	-	++
Interaction Protocols	-	-	+	++
Process perspective (PP)	+	+	+	+
Capacity to analyze workflow perspectives				
Delta				
IP	-	-	-	-
OP	-	-	+	-
PP	-	-	+	-
Performance				
IP	-	-	-	-
OP	-	-	+	+
PP	-	+	+	-
Particularities of system				
Usability	-	-	+	+
Portability	-	+	+	+
Extensibility	-	-	+	+

interaction protocols and organizational structures as defended in this paper.

Our work is distinguished from the existing proposals by:

- the extension of the classical events log model by adding concepts issued from organizational dimension in multi-agent systems and mainly the performative FIPA-ACL,
- the discovery of the organizational perspective and more precisely, the organisational structures (hierarchy, federations, coalitions, etc.) and the interaction protocols (contract nets, auctions, votes, etc.),
- an AGR representation for each workflow actor constituting an organizational structure (Ferber and Gutknecht 1998). It permits to represent graphically traditional elements of organizational perspective such as the roles, the organizational units and the performers.

The Table 6 compares our disopflow tool with the other existing systems.

Our work opens the following perspectives.

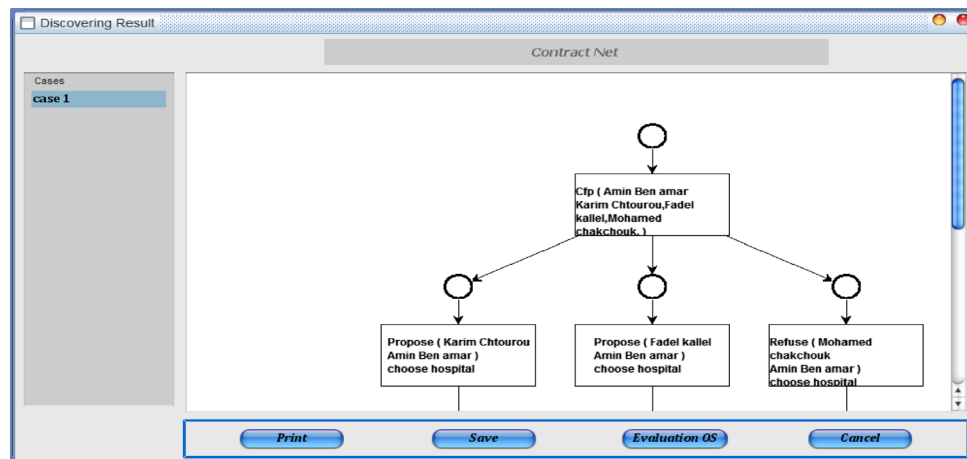
- *The evaluation of the discovered organizational structures and interaction protocols.* Actually, we focused on the evaluation of organizational structures in terms

of flexibility, efficiency and robustness using the Grossi method (Grossi et al. 2006). This latter is based on the graph theory proprieties and assumes an organization structured around actors and three types of relations between them (coordination, control and power) that can be described, thanks to the use of FIPA-ACL performatives. We are also looking for a method to evaluate the quality of discovered interaction protocols.

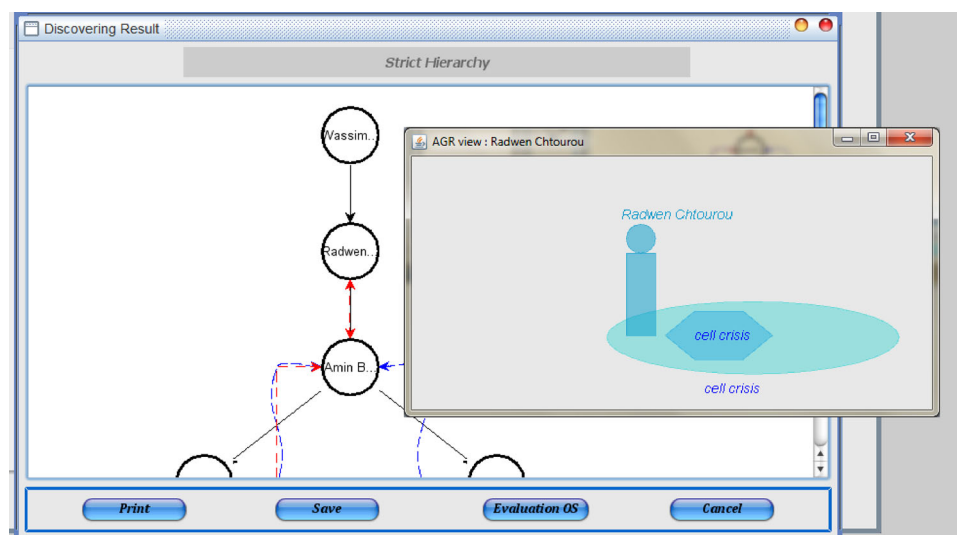
- *The discovery of the tree workflow perspectives.* The idea is to discover conjointly the three workflow perspectives to have a complete description of the deployed processes and being able to compare them to the prescribed processes.
- *The reverse engineering of processes.* We plan to exploit the evaluation result of the discovered organizational structures and interaction protocols to improve the design of the prescribed workflow processes.
- *The disopflow tool development.* We also plan to complete the development of others modules such as “OrgStructEvaluator”, “PerformanceAnalyser” and integrate it as a plug-in in the ProM platform.

**Acknowledgments** The authors would like to thank the professor Chihab Hanachi for its precious collaboration regarding the organizational structures and interaction protocols mining in Workflow.

## Appendix 1: Interface showing the discovered contract net protocol



## Appendix 2: Interface showing the AGR representation of workflow actor



## References

- Abdelkafi M, Bouzguenda L (2010) Discovering Organizational Perspective in Workflow using Agent Approach: an illustrative case study. EOMAS in Conjunction with CAISE, Hammamet, Tunisia
- Abdelkafi M, Bouzguenda L, Gargouri F (2011) Discopflow: a new tool for discovering organizational structures and interaction protocols in workflow. in international conference on information technology and e-services, ICITeS'11, Sousse-Tunisia, 10–12
- Boudewijn F, Dongen V, Van der Aalst WMP (2004) EMiT: a process mining tool. ICATPN pp 454–463
- Bouzguenda L, Bouaziz R, Andonoff E (2008) Dynamic plugging of business processes in cross organizational Workflow. Int J Comp Sci Appl Technomath Res Found, Bombay India 5(3b):141–164
- Bui T, Lee J (1999) An agent-based framework for building decision support systems. Decis Support Syst 25(3):225–237
- Ferber J, Gutknecht O (1998) A meta-model for the analysis and design of organizations in multi-agent systems. In: Proceeding of the 3rd International Conference on Multi-Agents Systems (ICMAS'98). EEE CS Press
- Gaaloul W, Gaaloul K, Bhiri S, Haller A, Hauswirth M (2009) Log-based transactional Workflow mining. Distr Parallel Databases 25(3):193–240
- Hanachi C, Sibertin-blanc C (2004) Protocol moderators as active middle-agent in multi-agent system. Int J Auton Agents Multiagent Syst (JAAMAS) 8(2):131–164
- Herbst J, Karagiannis D (2004) Workflow mining with InWoLvE. Computers in industry—special issue: process/Workflow mining, vol 53 Issue 3, Elsevier Science Publishers, B. V. Amsterdam, pp 245–264

- Huang C, Trappey JC, Yao H (2006) Developing an agent-based workflow management system for collaborative product design. *Indust Manag Data Syst* 106(5):680–699
- Kaplan SM (1990) Conversation builder: an open architecture for cooperative work. *Human-computer interaction, interact '90, IFIP*, pp 917–922
- Linh Thao LY, RINDERLE S, DADAM P (2006) Mining staff assignment rules from event-based data. In *business process management workshops*, pp 177–190
- Grossi, D.; Dignum, F.; Dignum, V.; Dastani, M.; & Royakkers, L. (2006). Structural aspects of the evaluation of agent organizations. *COIN@ECAI*
- Miguel V, Charoy F (2003) Bonita. Workflow cooperative system. <http://bonita.objectweb.org>
- Muehlen ZM (2004) Organizational management in workflow applications—issues and perspectives. *Inf Technol Manage* 5(3–4):271–291
- Searle JR (1969) *Speech acts*. Cambridge University Press
- Searle JR (1975) A taxonomy of illocutionary acts. K. Gunderson (ed) *Language, mind and knowledge*, University of Minnesota Press, pp 334–369
- Van der Aalst WMP (2005) Process mining in cscw systems. In *Proceedings of the 9th IEEE International conference on computer supported cooperative work in design (CSCWD)*, Coventry University/IEEE Computer Society Press, London, UK, pp 1–8
- Van der Aalst WMP (2011) *process mining: discovery, conformance and enhancement of business processes*. Springer-Verlag ISBN 978-3-642-19344-6 pp I–XVI, 1–352
- Van der Aalst WMP (2012) Process mining: overview and opportunities. *ACM Trans Manag Inf Syst* 3(2):7
- Van der Aalst WMP, Boudewijn F, Dongen V, Herbst J, Maruster L, Schimm G, Weijters AJMM (2003a) Workflow mining: a survey of issues and approaches. *Data Knowl Eng* 47(2):237–267
- Van der Aalst WMP, Ter Hofstede AHM, Kiepuszewski B, Barros A (2003) Workflow patterns. *Int J Distr Parallel Databases* 34(1):5–51
- Van der Aalst WMP, Aldred L, Dumas M, Ter Hofstede AHM (2004) Design and implementation of the YAWL system. *CAiSE* pp 142–159
- Van der Aalst WMP, Reijers HA, Song M (2005) Discovering social networks from events log. *Computer Supported Cooperative work* 14(6):549–593
- Van der Aalst WMP, Van Dongen BF, Günther CW, Rozinat A, Verbeek E, Weijters T (2009) *ProM: The Process Mining Toolkit*. BPM (Demos)
- Verbeek HMW, Hirschall A, Van der Aalst WMP (2002) XRL/flower: supporting inter-organizational workflows using XML/Petri-net technology. *WES* 93–108
- Verbeek HMW, Buijs Joos CAM, Boudewijn F, Dongen V, Van der Aalst WMP (2010) XES, XESame, and ProM 6. *CAiSE Forum* 60–75
- Weijters AJMM, Van der Aalst WMP (2003) Rediscovering workflow models from event-based data using little thumb. *Integrated Computer-Aided Engineering* 10(2):151–162
- Winograd T (1988) A language/action perspective on the design of cooperative work. *Human Computer Inter* 3(1):3–30