EFFECTIVE ANALYSIS OF FLEXIBLE COLLABORATION PROCESSES BY WAY OF ABSTRACTION AND MINING TECHNIQUES

Alfredo Cuzzocrea[1], Francesco Folino[2] and Luigi Pontieri[2]
[1]*ICAR-CNR and University of Calabria, Calabria, Italy*
[2]*ICAR-CNR, Rende, Italy*

Keywords: Knowledge Representation and Management, Complex Information Systems, Process Mining.

Abstract: A knowledge-based framework for supporting and analyzing *loosely-structured collaborative processes* (LSCPs) is presented in this paper. The framework takes advantages from a number of knowledge representation, management and processing capabilities, including recent *process mining techniques*. In order to support the enactment, analysis and optimization of LSCPs in an *Internet-worked virtual scenario*, we illustrate a *flexible integration architecture*, coupled with a knowledge representation and discovery environment, and enhanced by *ontology-based knowledge processing capabilities*. In particular, an approach for *restructuring* logs of LSCPs is proposed, which allows to effectively analyze LSCPs at varying abstraction levels with process mining techniques (originally devised to analyze well-specified and well-structured workflow processes). The capabilities of the proposed framework were experimentally tested on several application contexts. Interesting results that concern the experimental analysis of collaborative manufacturing processes across a distributed CAD platform are shown.

1 INTRODUCTION

Emerging *work models* are taking the form of networks of nimble, often self-organizing and cross-organizational, teams performing *loosely-structured processes*. A clear evidence of this trend is given by recent *virtual workspaces* (O. Anya et al., 2007; Experts Group, Next Generation Collaborative Working Environments 2005-2010, 2004), which put emphasis on the novel notion of *collaborative e-work environments*. Complexity and dynamicity that characterize such collaborative work scenarios pose new research challenging that are not addressed by traditional *Workflow Management Systems* (WfMS). This because traditional WfMS assume a rigid structure of the work model in order to control and monitor *Business Processes* (BP), with the aim of optimizing work distribution and resources allocation and usage. From a pertinent computer science perspective, this means that processes and tasks are rigorously modeled and represented according to fixed *structured* (yet *hierarchical*) *models*.

Looking at technological details, several *Enterprise Application Integration* (EAI) solutions (Hohpe & Bobby, 2003) can be exploited to build a flexible *collaborative environment* where existing systems and software components are re-used in order to provide a large spectrum of functionalities, such as content management, communication (e.g., e-mails, chats, forums), user management (e.g., user profiling, group management), inventorying of available technical resources, project management, and so forth.

Clearly, in order to achieve a full interoperability between components even at a semantic level, beyond to a pragmatic level, and to provide both workers and decision makers with a unified and high-level view over the underling organizational structure, collaborative processes and IT infrastructure, the need for a suitable representation and sharing model of information and knowledge is mandatory. In addition to the adaptation of conventional *Knowledge Management* (KM) solutions and strategies, some recent works (e.g., (O. Anya et al., 2007; Biuk-Aghai et al., 1999)) have pointed out the opportunity of extracting novel and useful knowledge from work models and schemes,

Cuzzocrea A., Folino F. and Pontieri L. (2010).
EFFECTIVE ANALYSIS OF FLEXIBLE COLLABORATION PROCESSES BY WAY OF ABSTRACTION AND MINING TECHNIQUES.
In *Proceedings of the 12th International Conference on Enterprise Information Systems - Artificial Intelligence and Decision Support Systems*, pages 157-167
DOI: 10.5220/0003020501570167

possibly by means of consolidated *Knowledge Discovery* (KD) techniques. Among the latter class of techniques, *historical log data* gathered during the execution of collaborative processes are exploited to discover *new process models* by means of *Process Mining* (PM) techniques (van der Aalst, 2003). This approach can well help understanding and analyzing collaborative work schemes actually performed by the target collaborative processes, as well as in determining and optimizing future work via possibly supporting the (re-)design of explicit and reusable process models.

Despite this, traditional process mining approaches are tailored to analyze logs of business processes executed by WfMS, which enforce strict *behavioral rules* along the enactment phase. As a consequence, these approaches are likely to yield knotty (i.e., "spaghetti-like" (Gunther & van der Aalst, 2007)) process models when applied to the collaborative processes arising in collaborative work scenarios outlined above. A major reason of this critical drawback of process mining techniques is represented by the incapability of traditional approaches to view event logs at some suitable application-independent and abstracted level. To the best of our knowledge, the latter research issue was only partially taken into account in very recent process mining literature (e.g., (Greco et al., 2008; Gunther & van der Aalst, 2007; Pedrinaci & Domingue, 2007)).

Starting from these considerations, in this paper we address the problem of supporting and analyzing the enactment of *loosely-structured collaborative processes* (*LSCPs*) by means of innovative knowledge representation, management and discovery tools. Particularly, in our research, LSCPs are viewed as *collaborative processes enacted in an Internet-worked virtual enterprise* that are not necessarily provided with a fully-specified model ruling execution and assignment of process tasks. All considering, this turns out to the definition of a *knowledge-based framework for processing LSCPs in collaborative e-work environments*, which should be considered as the prominent contribution of our research. It should be noted that, apart from addressing an important context of processes mining research that lacks from actual literature, our proposed framework *naturally* captures fundamental models and instances of next-generation business organizations, which more and more act in a virtual, collaborative and loosely-structured manner. More precisely, on the basis of some core ideas traced in (Basta et al., 2008), the following major contributions are provided in this paper.

- We devise a knowledge-based framework for supporting LSCPs by means of a flexible and lightweight *message-oriented architecture*, where a variety of systems and services can be easily integrated to support different kinds of collaborative tasks; here, a number of *ontology-based capabilitie*s are provided to represent and query organizational information and knowledge, while a separate *Data Warehouse* (DW) stores an integrated view of both relevant information along with the history of performed tasks.
- In order to effectively apply process mining techniques to LSPCs, we outline a semantic-aware method for *dynamically restructuring log data in a process-oriented way*, which takes full advantages from the available background knowledge shaped by the main knowledge-based framework for LSCPs introduced in this research.

The whole approach relies on the use of an *ontology-based framework* for representing event logs and associated concepts (e.g., organization structure, application domains, IT infrastructures, and high-level process models), which offers a *semantic substrate* for integrating different kinds of data and applications coming from heterogeneous environments.

The remainder of the paper is organized as follows. In Sect. 2, we first introduce some preliminary concepts on state-of-the-art process mining techniques, while evidencing critical limitations that make these techniques ineffective for analyzing LSCPs. Sect. 3 focuses the attention on the reference architecture implementing the proposed knowledge-based framework for supporting and analyzing LSCPs, by also putting emphasis on *integration issues* that naturally arise in collaborative e-work environments. After introducing the ontology-based framework for modeling event logs and associated organizational concepts/entities, Sect. 4 illustrates the semantic-aware method for process-oriented restructuring of logs, which allows us to straightforwardly apply process mining techniques to LSPCs. In Sect. 5, we discuss some experimentation conducted against a real-life application scenario involving the enactment of LSCPs within a distributed and decentralized collaborative CAD platform. Finally, in Sect. 6 we draw out concluding remarks deriving from our research, along with future research directions in the field of process mining techniques over non-traditionally-modeled business processes.

2 A CRITICAL OVERVIEW OF PROCESS MINING APPROACHES

Process Mining Techniques. Process mining refers to the problem of automatically extracting novel knowledge about the *behavior* of a given process, based on event data gathered in the course of its past enactments, and stored in suitable logs. Notably, such an ex-post analysis of process executions makes these techniques quite different from other business process analysis approaches, which mainly focus the attention on performance monitoring and reporting issues (e.g., (zur Muehlen, 2001; Jeng & Schiefer, 2003; McGregor & Schiefer, 2003)). Several process mining techniques have been defined in literature during past research campaigns. Each technique is indeed tailored to extract different types of (mining) models via capturing different aspects of the underlying process, such as the flow of work/data (e.g., (Folino et al., 2008; Greco et al., 2006; Greco et al., 2008)), or social relationships (e.g., (van der Aalst et al., 2005)).

Traditional process mining techniques mainly address the issue of discovering a graph-based *workflow model* (the so-called process-mining *control-flow* perspective), which describes both process activities and routing constraints that coordinate their execution. In order to describe complex processes in a more precise and modular way, the approach proposed in (Greco et al., 2006; Greco et al., 2008) exploits a *hierarchical clustering procedure* for recognizing different behavioral classes of process instances, and modeling them through *separated workflow schemas*. In particular, these schemas are restructured in (Greco et al., 2008) into a *taxonomical form*, which represents the process behavior at *different abstraction levels*. The resulting model is a *tree of workflow schemas*, where leaves stand for *concrete* usage scenarios, and any other internal node provides a *unified and generalized* representation for the sub-tree rooted in that node.

More recent process mining proposals try to take into account other (useful) information available in real-life logs rather than considering the mere execution of process tasks (e.g., activity executors, parameter values, and performance data) only. As a meaningful case, the approach introduced in (van der Aalst et al., 2005) supports the analysis of process logs according to an "organizational" perspective, in that it extracts different kinds of *social networks* modeling users' interaction. It is worth noticing that approach in (Greco et al., 2006; Greco et al., 2008) has been extended in (Folino et al., 2008) by means of the amenity of supporting the discovery of a *decision-tree model* relating the discovered behavioral classes with other data registered in the log. On the whole, the new research contribution in (Folino et al., 2008) allows us to achieve a *more powerful* and *richer* process behavior discovery model with respect to the one introduced in (Greco et al., 2006; Greco et al., 2008). Notably, the *predictive* capability of such a model ((Folino et al., 2008)) can effectively support different kinds of decisional tasks, which may involve both the design and the enactment of collaborative business processes.

Typical Representation of Log Data and Its Inefficacy for LSCPs Settings. Independently of their specific goals and approaches, the great majority of classical process mining techniques founds on quite a rigid and workflow-oriented conceptualization of process logs, like that underlying the log format *MXML*, an XML dialect used in the process mining framework *ProM* (Dongen et al., 2005). Due to the success of *ProM*, MXML is indeed widely diffused in the process mining community. A typical MXML document contains an arbitrary number of Process elements, each of them collecting a series of ProcessInstance elements. A ProcessInstance element consists of a number of log events (AuditTrailEntry elements), which are mandatorily associated with a process task (WorkflowModelElement element) and a running state (EventType element), which describes the execution state of the task (e.g., *scheduling*, *completion*, *suspension*). Optional elements contained in an AuditTrailEntry element represent its occurrence time (Timestamp element), and the resource (e.g., person or software component) that has triggered it (Originator element). Finally, an arbitrary number of Data elements allow to associate both events and process instances with further information, in the form of attribute-value pairs.

Two critical assumptions made in the *MXML* model risk to undermine the effectiveness of process mining techniques in analyzing LSCPs considered in our research. First, according to the model MXML, each event log must explicitly refer to a well-specified and well-structured workflow process, and to some high-level tasks within this workflow. Contrary to this, LSCPs can spontaneously arise without a well-specified neither well-structured model, via composing elementary, general-purpose

functions. The alternative solution of regarding these functions as high-level process activities may yield very intricate (i.e., "spaghetti-like" (Gunther & van der Aalst, 2007)) process models that are finally useless to analysis purposes.

Second, the model MXML does not encode any semantic information on the different types of entities that event logs may link to (e.g., human resources, software tools, data parameters), but simply models them by means of (elementary) labels. Beside preventing a semantic, high-level analysis of collaborative processes, and, more prominently, LSCPs, the drawback above may lead to a poor integration effect in a decentralized and multi-organization e-work scenario, thus inducting in several inconsistencies and redundancies of information representation.

Our knowledge-based framework and our approach to restructuring logs in a process-oriented way, described in Sect. 4, are just meant to overcome the limitations sketched above, and to allow for defining a mapping between basic log events and MXML elements in a flexible way, by possibly exploiting available high-level background knowledge to analyze the execution of LSCPs at some proper abstraction levels.

3 A REFERENCE ARCHITECTURE FOR MANAGING, TRACING AND ANALYZING LSCPS

Fig. 1 depicts the logical architecture of a comprehensive toolkit able of providing users of loosely-structured work environments with high level knowledge management functionalities, which can well enhance the enactment and the analysis of collaboration processes. The architecture is devised to fit Internet-worked scenarios where a variety of operational systems (making available different heterogeneous services) can be used by workers, possibly belonging to different organizations and located in different places.

In particular, the proposed architecture is hierarchically organized in four main layers,: (*i*) *Operational Systems* (OS), (*ii*) *Data & Application Integration* (D&AI), (*iii*) *Knowledge Management & Discovery* (KM&D); (*iv*) *Decision & Work Support* (D&WS). Principles, structures and functionalities of these layers are detailed next.

Operational systems are located in the OS layer of the architecture. Each operational system may

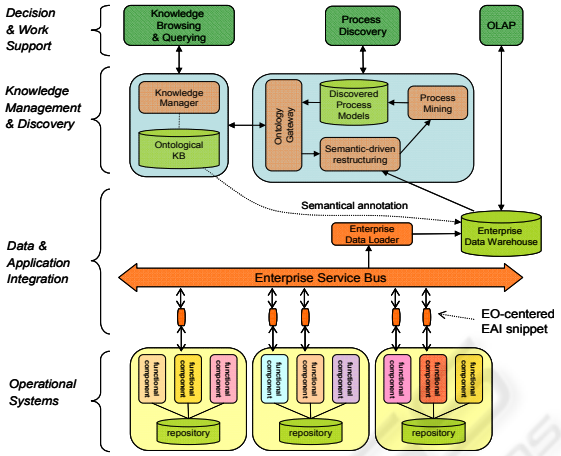

Figure 1: Reference architecture for supporting and analyzing LSCPs.

operate independently of the others, and keeps its own data repository, by also processing data stored in such a repository via a large variety of services, modeled according to a *functional-oriented approach* that is typical of Data Warehousing paradigms. In order to have these systems work congruously, and to prevent information inconsistency and redundancy, a flexible integration strategy is adopted. The latter is the main task of the D&AI layer of the architecture, which we describe in the following.

At the D&AI layer, operational systems, services and their associated data are conceptually integrated on the basis of a *shared conceptualization* of typical organizational and information resources, named as *Enterprise Knowledge Model* (EKM), which is presented and discussed in detail in Sect. 4. Notably, the execution of every operation affecting some entities in the EKM is modeled and regarded in terms of a so-called *Enterprise Event* (EE). Basically, the integration approach we propose is inspired to models and paradigms developed in the context of well-known event-based and service-based infrastructures (e.g., (Papazoglou & Georgakapoulos, 2003; Papazoglou & van den Heuvel, 2007)), like *Data and Knowledge Grids* (Cannataro & Talia, 2003), enriched by a prominent knowledge-oriented flavor. EEs play a key role in our proposed knowledge-based framework for LSCPs, as it will be made clearer later on.

In more details, the D&AI layer is based on a set of lightweight integration components, named *EAI snippets*, which communicate with the remaining components of the D&AI layer (including themselves) and operational systems of the OS layer throughout the so-called *Enterprise Service Bus*

(ESB). ESB essentially acts as a backbone providing high-level and reliable message-exchange services, and transparently handles mediation of endpoint heterogeneities and physical details during component communication. Every time an EE is produced by some functional component located in the OS layer, the associated EAI snippet reactively sends a message throughout the ESB. The message encodes ad-hoc information, including the kind of event occurred, the person or system that has originated the event itself, the work context (e.g., the actual project) of the event. Throughout the ESB, the message is forwarded to any other EAI snippet that subscribed that kind of EE. In turn, the latter EAI snippet will consequently update its data repository at the OS layer.

In addition to providing support for the coordination of operational systems and data integration tasks, in our proposed knowledge-based framework EEs are also treated and traced as *basic units of work* for the ex-post analysis of LSCPs, being this analysis based on process mining and *OnLine Analytical Processing* (OLAP) techniques. The latter mining functionalities are supported by the *Process Discovery* (PD) module, located at the D&WS layer, and the *OLAP* module, still located at the D&WS layer of the architecture, respectively. Furthermore, D&WS layer also supports advanced knowledge browsing, visualization, analysis, and querying services, which are definitely able of enabling effective decision making and collaborative work tasks based on data and knowledge stored and elaborated in the D&AI and KM&D layers of the architecture. The latter functionalities are fulfilled by the *Knowledge Browsing and Querying* (KB&Q) module, located at the D&WS layer.

In our proposed knowledge-based framework for LSCPs, data and knowledge are thus distributed across the D&AI and KM&D layers of the architecture, in order to augment the synergy among all the components of the framework. More specifically, for what regards data, the *Enterprise Data Warehouse* (EDW), located at the D&AI layer, contains snapshots of relevant enterprise data and historical EE logs, which are represented in an integrated and consolidated way according to the EKM. In order to populate the EDW, the *Enterprise Data Loader* (EDL) module, still located at the D&AI layer, continuously elaborates all messages exchanged throughout the ESB with the goal of extracting data and storing them in the underlying warehouse. To this end, canonical *Extraction-Transformation-Loading* (ETL) primitives can be advocated.

For what instead regards knowledge, the *Knowledge Manager* (KM) module, located at the KM&D layer, is in charge of maintaining a series of *interrelated ontologies* (which are modeled according to the ontology-based framework we describe in Sect. 4) within an appropriate *Ontological Knowledge Base* (OKB), still located at the KM&D layer. Beside constituting a semantic background that turns to be very useful for data integration purposes, ontologies stored in the OKB also enable a *meaningful semantic annotation of data stored in the EDW*, while also nicely supporting a semantic-aware access to them. More precisely, the system allows to introduce meaningfully interrelated ontologies for the EEs occurring in the target (virtual) organization, which also capture organizational concepts/entities associated with the EEs themselves. Several studies have already evidenced that collaborative processes clearly benefit from the introduction of knowledge management approaches and strategies. This because the latter can effectively and successfully support the management of knowledge that is created, stored, shared and delivered along the execution of collaborative processes. Therefore, making use of a suitable ontology-based framework within the knowledge-based framework we propose (particularly, at the KM&D layer) is completely reasonable, while it embeds several points of research innovation. From a technical point of view, the proposed reference architecture relies on the ontology-based modeling framework presented in (Gualtieri & Ruffolo, 2005), which both provides a semantic infrastructure for the management of organizational knowledge and supports interoperability among existing operational systems.

The availability of semantical information is fully-exploited by the *Semantic-driven Restructuring* (SdR) module, located at the KM&D layer, which supports selection and manipulation of basic EEs in order to dynamically restructure them prior to the application of process mining algorithms executed by the *Process Mining* (PM) module of the KM&D layer. As mentioned in Sect. 1, this strategy is meant with the aim of straightforwardly applying process mining techniques over EEs, while taking advantages from the available background knowledge. The latter restructuring approach is illustrated in detail in Sect. 4. On the other hand, novel pieces of knowledge, possibly captured in models and patterns extracted by the PM module and stored in the *Discovered Process Models* (DPM) repository of the KM&D layer, can be further integrated in the actual OKB by means of the

creation/modification of ontologies about organizational structures and collaborative processes. In our architecture, the latter functionality is fulfilled by the *Ontology Gateway* (OG) module, located at the KM&D layer.

As a final remark concerning ontology-based knowledge representation and management aspects incorporated by our proposed knowledge-based framework for LSCPs, here we highlight that the architecture above might clearly be enhanced by means of additional capabilities focused to construct and maintain ontologies in a *distributed and collaborative manner*, like recent research results (Kozaki et al., 2007; Pinto et al., 2004) suggest. However, this issue is outside the scope of this paper, and thus left as future work.

4 ONTOLOGY-BASED REPRESENTATION AND RESTRUCTURING OF EE LOGS

Representation of EEs and of Associated Entities. As mentioned in Sect. 1 and Sect. 3, our ontology-enhanced EKM relies on the modeling framework in (Gualtieri & Ruffolo, 2005). By the way, other approaches could have been adopted as well, such as, for instance, the ontological framework presented in (Pedrinaci & Domingue, 2007), which leverages on *Semantic Web* technologies to represent event logs, process mining tasks, and various kinds of associated information, and indeed represents a valid yet reliable alternative to the proposal in (Gualtieri & Ruffolo, 2005).

Briefly, the framework (Gualtieri & Ruffolo, 2005) hinges on two modeling levels: (*i*) the *Core Organizational Knowledge Entity* (COKE) ontology, where the EKM is expressed in terms of the so-called core organizational knowledge entities; (*ii*) a top-level ontology for representing more general organizational knowledge of the target organization, which consists of a structured collection of concepts that can be used to annotate COKE elements. In our context, the central COKE entity corresponds to EEs, which are distinguished in three functional sub-classes: (*i*) *TaskRun*, which pertains the execution of project activities/tasks; (*ii*) *HumanManagement*, which concerns to managerial functions; (*iii*) *ContentManagement*, which is primarily focused on the manipulation of information. Other COKE entities an EE can be related to include: the software component (*Tool*) originating the EE and/or the person (*Agent*) that performed it, which can be suitably in two different sub-ontologies: the *Human Resource* (HR) ontology and the *Technical Resource* (TR) ontology, respectively. Furthermore, an EE can also be associated to a series of parameters modeled as instances of the *Knowledge Object* (KO) ontology, and, most importantly, to instances of the *Business Process* (BP) ontology.

BP ontology allows us to characterize the work context of the event (i.e., the context within which the event has been performed). While *Process* and *Task* are well-known concepts in traditional WfMS, thus they do not deserve additional details, the entity *Project* plays a key role in our proposed knowledge-based framework for supporting and analyzing LSCPs. In fact, in loosely-structured collaborative e-work scenarios, this entity turns to be extremely useful for monitoring and analysis purposes. This because, in these scenarios a project can be intended as a *bunch of tasks* that can be *not a-priori completely-specified and well-structured*, each of these tasks being possibly associated to specific (project) goals and (project) constraints, as well as to a series of human, computational and information resources. Different types of association may link projects and processes. At one end, a project could be carried out according to some well-defined workflow models (which could be used by other (similar) projects as well). At the other end, a project could be accomplished with no explicit process models at all, but rather it could be only based on completely-spontaneous or tacit cooperation schemes (like it happens in collaborative e-work scenarios). In our knowledge-based framework for LSCPs, BP ontology allows to capture these particularities of business processes in loosely-structured collaborative e-work scenarios, thus overcoming limitations of traditional business process models.

Notably, all of the organizational entities mentioned above can be semantically annotated with concepts coming from some suitable domain ontology, possibly organized according to taxonomical or partonomical structures, which can be well exploited for abstraction purposes.

Process-Oriented Restructuring of EE logs. In order to effectively apply process mining techniques to LSCPs, our proposed knowledge-based framework incorporates a semantic-aware approach that allows us to dynamically restructure basic EE logs in a process-oriented way, while effectively exploiting the available background knowledge. The final goal of the proposed restructuring approach

consists in finally producing a *workflow-oriented process log* from actual EE logs, such that this process log can be easily represented via the model MXML (see Sect. 2). To this end, the following three logically-distinct steps are introduced by our EE restructuring approach:

(a) Select a subset of suitable EEs.
(b) Arrange the selected EEs in meaningful process instances.
(c) Map EE attributes into MXML-formatted workflow nodes.

In step (*a*), the analyst is allowed to choose a suitable subset of EEs among those stored in the EDW, by possibly specifying selection conditions on properties of these events, such as the execution date or the kind of event, as well as on properties of other entities associated with these events, such as tools, actors and projects.

The goal of step (*b*) consists in partitioning the previously-selected EE set into a number of *sequences*, *each of which will be regarded as a distinct process instance*. It should be noted that, in a conventional process mining context, each event log already specifies which process instance it refers to. Conversely, in our loosely-structured collaborative e-work scenario such information is not available whenever process activities are not performed following a well-specified and well-structured workflow model. Despite this intrinsic characteristic of loosely-structured collaborative e-work scenarios, in a suitable ex-post analysis session of conventional business processes the analyst could be still interested in re-organizing (well-specified) EEs into workflows different from the actual one modeling the execution of the target process.

For instance, the latter re-organization could be achieved by grouping EEs into separate process instances. In this respect, good grouping alternative could be the following ones: (*i*) *grouping by project* within which EEs have been originated; (*ii*) *grouping by knowledge object* on which EEs have been performed. In more detail, the first case explores the application scenario in which *multiple* projects are carried out by the (virtual) collaborative organization, thus each project is indeed seen as a *distinct* project instance originating EEs. This approach, named as *Project-Centric Enterprise Event Restructuring* (PCEER), can be exploited to extract a global process model that describes work patterns characterizing all the available project instances, or a class of them. The second EE grouping alternative, named as *Knowledge-Object-Centric Enterprise Event Restructuring* (KOCEER), can be instead exploited to analyze the typical life-cycle of a specific class of knowledge objects, such that deliverables, documents, and so forth. Clearly, many other EE grouping options exist. For instance, one can define each process instance in such a way as to assemble all events originated by a single actor on a certain suitable time basis (e.g., all the operations performed by code developers during each week), in order to eventually capture their modus operandi.

Step (c) is devoted to determine the mapping from low-level EEs to the logical tasks (WorkflowModelElement elements in the MXML format) constituting the process. It is easy to understand how step (*c*) plays a critical role within the knowledge-based framework for LSCPs we propose. In fact, the phase associated to step (*c*) *finally determines which abstraction level will be used for analyzing the execution of LSCPs*. In particular, in the most detailed case, all the information content conveyed in each EE is mapped to a *single* WorkflowModelElement node that corresponds to the execution of a *certain* event, performed by a *certain* actor, throughout a *certain* tool, over a *certain* set of knowledge objects, and so forth. Since such an approach is likely to yield rather cumbersome and sparse models, which would turn to be useless for analysis purposes, *some less-detailed representations* of EEs should be achieved. This allows us to capture the execution of the process in a more concise and meaningful way. To this aim, the analyst can decide to focus only on some *dimensions of analysis* associated to EEs (e.g., the tool employed), in an OLAP-like manner, while possibly exploiting suitable concept taxonomies to represent EEs themselves in a more abstract way.

5 CASE STUDY AND EXPERIMENTAL ASSESSMENT: COLLABORATIVE WORK ACROSS A CAD PLATFORM

In order to explore the potentialities of our approach to the restructuring and mining of EE logs, in this Section we illustrate its application to a real-life collaboration scenario, which has been addressed in the context of the research project "TOCAI.it". The target scenario concerns the collaborative development of artifacts within a real-life manufacturing enterprise, by way of a distributed CAD platform. All the experimental results described next have been carried out on a collection

of logs gathered by the CAD system during a two-month period, which regard the development of 5,675 artifacts.

As a preliminary step, the application of process discovery techniques to these data requires some proper recognition of what pieces of EE log data represent the process and its tasks. In this respect, different perspectives according to which EE logs are restructured can be advocated. In our experimental analysis, we have considered the following restructuring perspectives:

- *Operation-Centric Perspective*: Here, we focus on the application of CAD operations on the artifacts developed throughout the CAD platform. In order to support such a kind of analysis, the original EE log data have been restructured into an MXML log, named *OPN-RAW,* where each process instance corresponds to a distinct artifact, and each event trace is labeled with the name of the CAD operation applied to the artifact. A variant of this approach, named *OPN-CAT,* has been also defined via replacing CAD operation names with higher-level operation categories, based on a given ad-hoc taxonomy. The lowest level concepts in the taxonomy are shown in Table 1, along with their associated instances.
- *User-Centric Perspective*: In this case, the focus of the analysis in on the ways workers co-operate across the development of CAD artifacts. In this regard, we have introduced a novel process log, named *USR-RAW*, where each CAD artifact still corresponds to a distinct process instance, while each EE log is labeled with the ID of the user that originated the event itself. Again, a more abstract variant of this approach, named *USR-ROLE*, has been defined via replacing user IDs with three high-level functional roles, still taken from a given taxonomy (which is omitted for space reasons): general users (category: *User*), administrators (category: *SysAdmin*), and checkers (category: *View*).

In our experimental analysis and evaluation, we model the knowledge discovered from EE logs in terms of (collaborative) work models implemented by means of workflows, so that the main emphasis is on stressing the *workflow discovery capabilities* of our proposed framework. To this end, restructured EE logs above have been analyzed by means of several process mining techniques, and a cross-validation has been conducted. For the sake of conciseness, we next focus only on the application of the algorithm *DWS Miner* (Greco et al., 2006), which runs as a plug-in available in ProM. DWS Miner combines a classical workflow discovery tool, namely *HeuristicMiner*, with an innovative *process clustering approach* (Greco et al., 2006).
As demonstrated in (Greco et al., 2006), DWS Miner is particularly suitable to discovery workflow schemas in the context of LSCPs, mainly thanks to the capabilities of its embedded clustering tool in recognizing different behavioral classes among processes. At the end, the latter amenity allows a critical improvement of the mining-from-logs capabilities of DWS Miner (Greco et al., 2006) to be achieved.

Table 1: Excerpt of the taxonomy of CAD operations: leaf concepts and associated instances.

Operation Category	***Operation Instance***
Construction	Creation::WorkInProgress
Modify	CF2T::TechAend, F2T::TechAend, Modify::UnderModify
Test	K2AVP::AV Pruefung, M2AVP::AV Pruefung, T2AVP::AV Pruefung
CVS	Auschecken, Auschecken annullieren, Einchecken, Get, Importieren, Lokale Kopie
Release	AVP2F::Freigabe, CK2F::Freigabe, CT2F::Freigabe, K2F::Freigabe, M2F::Freigabe,RecoverModify::Released, T2F::Freigabe, Wip2F::Freigabe
Delete	CancelModify::Cancelled
Build-Up	AVP2K::Konstruktion, Wip2K::Konstruktion
Model-Definition	AVP2M::Musterbau, K2M::Musterbau, Wip2M::Musterbau
Null Series	Wip2N::Nullserie

Effectiveness of the Clustering Phase. The first study we have conducted concerns the study of the effectiveness of the clustering phase embedded in DWS Miner, based on which different behavioral classes among processes are discovered (and clustered), with respect to the four different restructuring perspectives of EE logs considered in our analysis. In order to carefully assess the capability of discovered (respectively, clustered) workflow models in adequately capturing behaviors recorded in the underlying EE logs, some *conformance metrics* are necessary. To this end, we adopt the idea of resorting the following measures introduced in (Rozinat & van der Aalst, 2008) (all these measure range over the continuous interval [0:1]):

- *Fitness*: a sort of *completeness measure* which evaluates the compliance of EE log traces w.r.t. a

given workflow model. Roughly speaking, this measure considers the number of mismatches that occur when performing a non-blocking replay of all EE log traces through the model. The more are the mismatches the lower is the measure. In a sense, fitness quantifies the ability of the workflow model to parse all the traces from the EE logs.

- *Advanced Behavioral Appropriateness* (*BehAppr* for short): a precision measure able to express the amount of *workflow model flexibilities*, such as alternative or parallel behaviors, that has not been exploited to produce real executions registered in the EE logs.
- *Advanced Structural Appropriateness* (*StrAppr* for short): a measure which assesses the capability of a workflow model in describing EE logs in a maximally-concise way.

Table 2 summarizes some experimental results. Here, for each restructuring perspective, we report the number of clusters found, and the associated conformance scores. At a first glance, we note that, surprisingly, high effectiveness results have been achieved, and, particularly, workflow models collectively attain a maximal score under the metric *StrAppr*. Also, from table 2 it should be noted that a noticeable gain is achieved over *all* the performance metrics when moving from detailed logs (i.e., *OPN-RAW* and *USR-RAW*) to their corresponding *abstract views* (i.e., *OPN-CAT* and *USR-ROLE*). Yet being such a result tightly linked to the particular application scenario considered in our experiential assessment, it gives us a hint of the benefits deriving from synergistically combining process mining techniques and abstraction-oriented restructuring methods that leverage on background knowledge, possibly encoded in suitable domain ontologies.

Table 2: Clustering results and associated conformance scores for different EE log restructuring perspectives.

Restructured Log	***Clusters***	***Fitness***	***BehAppr***	***StrAppr***
OPN-RAW	4	0.6585	0.9164	1.0000
OPN-CAT	4	0.8013	0.9975	1.0000
USR-RAW	4	0.6712	0.7974	0.9200
USR-ROLE	2	0.8033	1.0000	1.0000

Effectiveness of the Workflow Discovery Phase. In a different experimental session, we have focused the attention on the specific workflow discovery capabilities of our framework, based on the preliminary behavior-aware clustering phase. In particular, as suggested by our previous analysis on the effectiveness of the clustering phase, in this second analysis we focus the attention on the (restructured) logs *OPN-CAT* and *USR-ROLE*, respectively, as the latter expose the better conformance scores (see Table 2). Fig. 2 show four workflow schemas discovered from the operation-centric log *OPN-CAT*. Discovered workflow schemas meaningfully allow us to get some insights on the typical life-cycle of artifacts developed across the target CAD platform. As a matter of fact, these workflows reveal that different work models have been followed during the development of artifacts, with respect to sequences of CAD operations applied to those artifacts (see table 1). For instance, activities DELETE and MODIFY only appear in the execution instances modeled by the workflow shown in Fig. 2(*a*). Similarly, workflow in Fig. 2(*b*) distinguishes itself for the presence of the activity NULLSERIE, while the one in Fig. 2(c) is the only containing the activity MODEL-DEFINITION. In addition to this, workflow in Fig. 2(*c*) reveals that the activity MODIFY is always executed after the activity RELEASE, differently from the case of workflow in Fig. 2(*a*). Finally, a very simple workflow schema is the one depicted in Fig. 2(*d*), which yet covers a significant number of process instances. In fact, this somewhat-unexpected work model corresponds to a number of enactments of the manufacturing process, which have not been carried out completely.

As described in beginning of this Section, the aim of the user-centric restructuring perspective is to analyze the cooperative schemes followed by workers across the execution of CAD-based manufacturing operations. Fig. 3 shows the two distinct *usage scenarios* (respectively, workflow schemas) discovered from the user-centric log *USR-ROLE* for two distinct cases represented by the majority of manufacturing cases (Fig. 3(*a*)) and the remaining cases (Fig. 3(*b*)). Despite the simplicity of these models, which mainly descends from the fact that only three distinct user roles appear in the log *USR-ROLE*, it is still possible to appreciate the capability of clearly identifying two well-distinct role-based (collaborative) work models, yet discovering interesting knowledge. For instance, from Fig. 3(*a*), it should be noted that, in the great majority of manufacturing cases, no intervention of users with administration role *SysAdmin* has been required. On the other hand, workflow schema in Fig. 3(*b*) shows that, in the remaining cases where some administrators have been involved, no direct interaction between checkers (*View*) and general users (*User*) have occurred.

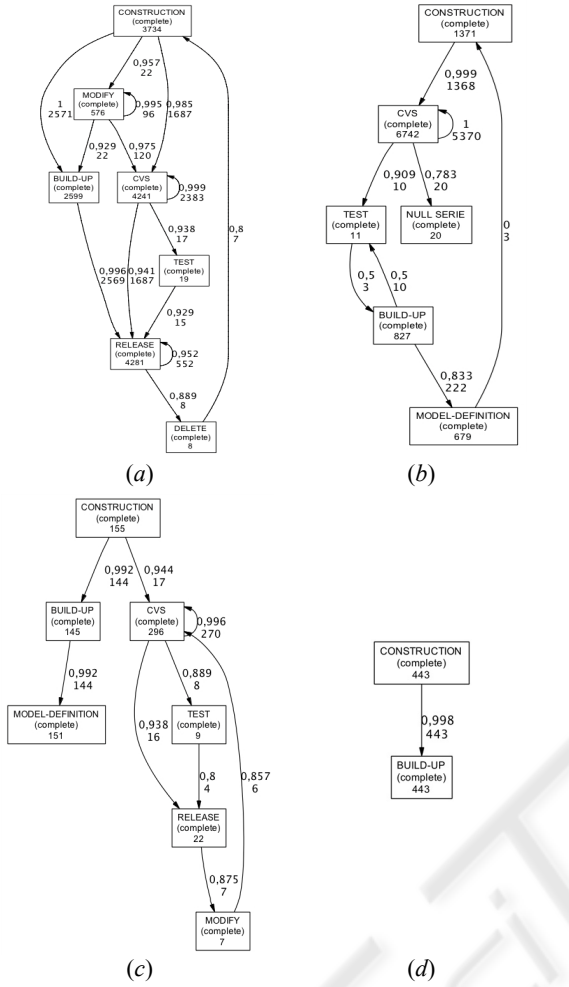

Figure 2: Operation-centric work models discovered from the restructured log *OPN-CAT*.

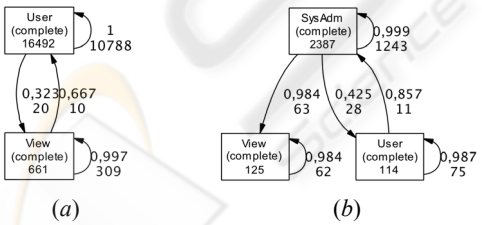

Figure 3: User-centric work models discovered from the restructured log *USR-ROLE*.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we have described a knowledge-based framework for supporting and analyzing LSCPs, and the meaningful construction of extensible and distributed cooperative systems. Our proposed framework fully exploits advanced capabilities for representing, managing and discovering organizational and process knowledge.

A prominent feature of the proposed framework is represented by the definition of an interactive restructuring method for flexibly applying process mining techniques to LSCPs. The framework has been tested against some real-life applications, investigated by a research project concerning the exploitation of advanced knowledge management models and methodologies in inter-organizational collaborative environments. In particular, a comprehensive experimental assessment of the proposed process mining analysis approach over LSCPs has been conducted for a case study represented by the collaborative development of artifacts within a real-life manufacturing enterprise, by way of a distributed CAD platform. Obtained experimental results have clearly demonstrated the effectiveness of the approach in providing analysts and workers with insightful views over the execution of processes, and in supporting the *consolidation* of knowledge across the whole collaborative e-work environment.

As future work, we are investigating on extending our proposed approach with mechanisms supporting a collaborative and distributed construction of ontologies.

REFERENCES

W. M. P. van der Aalst, H.A. Reijers, and M. Song, 2005. Discovering Social Networks from Event Logs. *Computer Supported Cooperative Work*, 14(6): 549—593.

W. M. P. van der Aalst, B.F. van Dongen, J. Herbst et al., 2003. Workflow Mining: A Survey of Issues and Approaches. *Data & Knowledge Engineering*, 47(2): 237—267.

O. Anya, H. Tawfik, and A. Nagar, 2007. A Conceptual Design of an Adaptive and Collaborative E-Work Environment. In *Proc. of 1st Asia Intl Conf on Modeling & Simulation*, pp. 148—154.

R. P. Biuk-Aghai, and I.T. Hawryszkiewycz, 1999. Analysis of Virtual Workspaces. In *Proc. of 1999 Intl Symp on Database Applications in Non-Traditional Environments*, pp. 325—332.

B. F. van Dongen, A.K.A. de Medeiros, H.M.W. Verbeek et al., 2005. The ProM Framework: A New Era in Process Mining Tool Support. In *Proc. of 26th Intl Conf on Applications and Theory of Petri Nets*, pp. 444—454.

Experts Group, 2004. Next Generation Collaborative Working Environments 2005-2010, EUROPEAN COMMISSION Information Society Directorate-General, Report of the Experts Group on Collaboration @ Work, Brussels.

F. Folino, G. Greco, A. Guzzo, and L. Pontieri, 2008. Discovering Multi-Perspective Process Models. In *Proc. of the 10th Intl Conf on Enterprise Information Systems*, pp. 12—16.

G. Greco, A. Guzzo, and L. Pontieri, 2008. Mining Taxonomies of Process Models. *Data & Knowledge Engineering*, 67 (1): 74—102.

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

S. Basta, F. Folino, A. Gualtieri, M.-A. Mastratisi, and L. Pontieri, 2008. A Knowledge-Based Framework for Supporting and Analysing Loosely Structured Collaborative Processes. In *Proc. of the 12th East European Conf on Advances in Databases and Information Systems*, pp. 140—145.

M. Cannataro, and D. Talia, 2003. The Knowledge Grid: An Architecture for Distributed Knowledge Discovery. *Communications of the ACM*; 46(1): 89—93.

A. Gualtieri, and M. Ruffolo, 2005. An Ontology-Based Framework for Representing Organizational Knowledge. In *Proc. of 5th Intl Conference on Knowledge Management*.

C. W. Gunther, and W.M.P. van der Aalst, 2007. Finding Structure in Unstructured Processes: The Case for Process Mining. In *Proc. of 7th Intl Conf on Application of Concurrency to System Design*, pp. 3—12.

G. Hohpe, and W. Bobby, 2003. *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Addison-Wesley Longman Publishing Co.

M. zur Muehlen, 2001. Process-Driven Management Information Systems - Combining Data Warehouses and Workflow Technology. In *Proc. of the 4th Intl Conf on Electronic Commerce Research*, pp. 550—566.

J. J. Jeng, and J. Schiefer, 2003. An Agent-Based Architecture for Analyzing Business Processes of Real-Time Enterprises, In *Proc. of the 7th Intl Enterprise Distributed Object Computing Conf*, pp. 86—97.

K. Kozaki, E. Sunagawa, Y. Kitamura, and R. Mizoguchi, 2007. Distributed and Collaborative Construction of Ontologies Using Hozo. In *Proc. of the 2007 Workshop on Social and Collaborative Construction of Structured Knowledge*.

C. McGregor, and J. Schiefer, 2003. A Framework for Analyzing and Measuring Business Performance with Web Services. In *Proc. of 2003 IEEE Intl Conf on E-Commerce*, p. 405.

C. Pedrinaci, and J. Domingue, 2007. Towards an Ontology for Process Monitoring and Mining. In *Proc. of the 2007 Workshop on Semantic Business Process and Product Lifecycle Management*, pp. 76—87.

H. Pinto, C. Tempich, and Y. Sure, 2004. DILIGENT: Towards a Fine-Grained Methodology for Distributed, Loosely-controlled and evolvInG Engingeering of oNTologies. In *Proc. of the 16th European Conference on Artificial Intelligence*, pp. 393—397.

M. P. Papazoglou, and G. Georgakapoulos, 2003. Service-Oriented Computing. *Communications of the ACM*, 46(10): 24—28.

M. P. Papazoglou, and W.-J. van den Heuvel, 2007. Service-Oriented Architectures: Approaches, Technologies and Research Issues. *VLDB Journal*, 16(3): 389—415.

A. Rozinat, and W.M.P. van der Aalst, 2008. Conformance Checking of Processes based on Monitoring Real Behavior. *Information Systems*, 33(1): 64—95.