Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Supporting and Analyzing Loosely-Structured Collaborative Processes by means of Knowledge Representation, Management and Mining Models

Alfredo Cuzzocrea[1], Francesco Folino[1],
Luigi Pontieri[1]

RT-ICAR-CS-09-09 **Ottobre 2009**

Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR)
– Sede di Cosenza, Via P. Bucci 41C, 87036 Rende, Italy, URL: *www.icar.cnr.it*
– Sezione di Napoli, Via P. Castellino 111, 80131 Napoli, URL: *www.na.icar.cnr.it*
– Sezione di Palermo, Viale delle Scienze, 90128 Palermo, URL: *www.pa.icar.cnr.it*

Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Supporting and Analyzing Loosely-Structured Collaborative Processes by means of Knowledge Representation, Management and Mining Models

Alfredo Cuzzocrea[1], Francesco Folino[1],
Luigi Pontieri[1]

Rapporto Tecnico N.: **RT-ICAR-CS-09-09**

Data: **Ottobre 2009**

---

[1] Istituto di Calcolo e Reti ad Alte Prestazioni, ICAR-CNR, Sede di Cosenza, Via P. Bucci 41C, 87036 Rende(CS)

I rapporti tecnici dell'ICAR-CNR sono pubblicati dall'Istituto di Calcolo e Reti ad Alte Prestazioni del Consiglio Nazionale delle Ricerche. Tali rapporti, approntati sotto l'esclusiva responsabilità scientifica degli autori, descrivono attività di ricerca del personale e dei collaboratori dell'ICAR, in alcuni casi in un formato preliminare prima della pubblicazione definitiva in altra sede.

Supporting and Analyzing Loosely-Structured Collaborative Processes by means of Knowledge Representation, Management and Mining Models

Alfredo Cuzzocrea[1], Francesco Folino[2], Luigi Pontieti[3]

[1]*ICAR-CNR and University of Calabria*
Via P. Bucci, 41C – I-87036 Rende, Italy
cuzzocrea@si.deis.unical.it

[2]*ICAR-CNR*
Via P. Bucci, 41C – I-87036 Rende, Italy
{ffolino,pontieri}@icar.cnr.it

***Abstract*—A knowledge-based framework for supporting and analyzing *loosely-structured collaborative processes* (LSCPs) is presented in this paper. The proposed framework is based on a number of knowledge representation, management and mining models that meaningfully exploit recent *process mining techniques* of traditional *Workflow Management Systems* (WfMS). 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. This approach allows to effectively analyze LSCPs at varying abstraction levels via process mining techniques, originally devised to analyze well-specified and well-structured workflow processes. Finally, the capabilities of the proposed knowledge-based framework are experimentally tested against several settings focusing on knowledge management models and methodologies in real-life large organizations, even in an inter-organizational manner.**

I. 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* [3,6], 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 [13] can be exploited in order to build a flexible *collaborative environment* where existing systems and software components may be re-used 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 and counting 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., [3,4,10]) have pointed out the opportunity of extracting novel and useful knowledge from work models and schemes, 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 in [10] in order to discover *new process models* by means of *Process Mining* (PM) techniques [2]. As demonstrated in [10], this approach can help in 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 [10].

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" [12]) 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 challenge issue has been partially taken into account by very recent process mining literature (e.g., [8,12,18]).

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 mining models. 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, the following three major contributions are introduced in this paper.

- We devise a flexible and lightweight *message-oriented architecture* for supporting LSCPs – In this architecture, a variety of systems and services can be easily integrated in order 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.
- We introduce a fundamental *ontology-based framework* that allows us to represent event logs and associated concepts, such as structure of involved organizations, application domains, IT infrastructures, and high-level process models – In principle, this framework offers a *semantic substrate* for integrating different kinds of data and applications, possibly coming from heterogeneous platforms and organizations.
- Finally, 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 remaining part of the paper is organized as follows. In Section II, 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. Section III focuses the attention on the conceptual 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. Section IV describes the ontology-based framework for modeling event logs and associated organizational concepts/entities, which plays a leading role in our proposed framework. In Section V, we illustrate the semantic-aware method for process-oriented restructuring of logs, which allows us to straightforwardly apply process mining techniques to LSPCs. In Section VI, we present several tests and experimental results on the capabilities of the proposed knowledge-based framework against the scenarios drawn by several research projects focused on knowledge management models and methodologies in real-life large organizations, even in an inter-organizational manner. Finally, in Section VII 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.

II. Preliminaries: Process Mining and Process Logs, and their Limitations in Dealing with LSCPs

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., [14,15,17]).

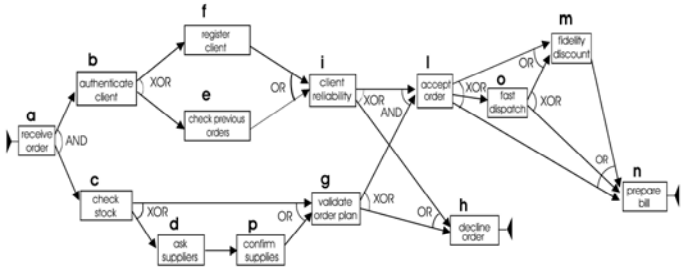

Fig. 1 Workflow schema for the sample process HANDLEORDER

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., [7,8,9]), or social relationships (e.g., [1]). Traditional process mining techniques mainly address the issue of discovering a *workflow model* (the so-called process-mining *control-flow* perspective), which describes both process activities and routing constraints that coordinate their execution. A beginner's picture of such a simple-yet-powerful model is given in Figure 1 that shows an hypothetical order management process, called HANDLEORDER. Here, edges represent precedence relationships, while additional constraints are associated to activity nodes. For instance, task l is an AND-join activity, meaning that task l can be activated only *after* it has been notified that both the client is reliable (task i) and the order can be supplied correctly (task g). Conversely, the XOR-split

activity modeled by task b can activate just one of its adjacent activities (i.e., task f and task e), once enacted.

In order to describe complex processes in a more precise and modular way, the approach proposed in [8,9] 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 [8] into a *taxonomical form*, which represents the process behavior at *different abstraction levels*. The resulting process 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 [1] 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 [8,9] has been extended in [7] 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 [7] allows us to achieve a *more powerful* and *richer* process behavior discovery model with respect to the one introduced in [8,9]. Notably, the *predictive* capability of such a model ([7]) can effectively support different kinds of decisional tasks, which may involve both the design and the enactment of collaborative business processes.

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. As an instance, Figure 2 illustrates the model *MXML*, an XML-based format used by the framework *ProM* [5] for representing event logs. Due to the popularity and the success of *ProM*, MXML is indeed widely diffused in the process mining community, with dignity in both the academic and industrial research communities. As shown in Figure 2, in a typical MXML document WorkflowLog modeling a log file the root node WorkflowLog contains an arbitrary number of elements Process, each of them collecting a series of elements ProcessInstance. Furthermore, the root node WorkflowLog can contain an element Source, which specifies the system/component from which the log file has been imported. A process instance element ProcessInstance consists of a number of log events modeled by elements AuditTrailEntry, which mandatory refer to a process task, modeled by the element WorkflowModelElement, and are associated to a running state, modeled by the element EventType. The running state describes the state of the target business process at the time the log file has been gathered. Possible running state instances are: *scheduling*, *completion*, *suspension*, and so forth. Optional elements contained by an audit trail entry element AuditTrailEntry represent its occurrence time, modeled by the element Timestamp, and the resource (e.g., person or software component) that has triggered it, modeled by the element Originator. Finally, an arbitrary number of attribute-value pairs modeling useful information available in real-life logs can be associated to both events and process instances. This information is captured and modeled by the element Data.

Two critical assumptions made in the *ProM*'s event log 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" [12]) process models that are finally useless to analysis purposes.

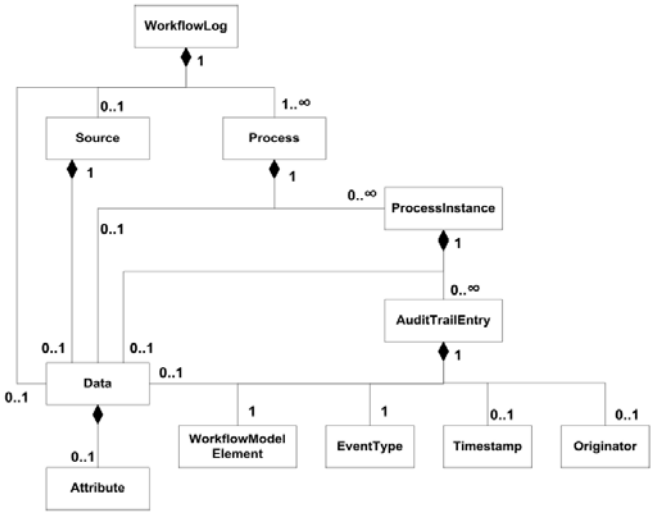

Fig. 2 The MXML format: a quasi-standard for modeling process logs

Second, the model MXML does not encode any semantic information on the different types of entities that event logs may be connected with (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.

The knowledge-based framework for supporting and analyzing LSCPs and the approach for restructuring logs in a process-oriented way, which are described in Section IV and Section V, respectively, are meant to overcome limitations above, and to allow us to define mappings 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.

III. A Conceptual Architecture for Supporting and Analyzing LSCPs

Figure 3 depicts the conceptual architecture implementing our knowledge-based framework for supporting and analyzing LSCPs, enriched by integration, tracking and reporting capabilities. The proposed architecture is devised with the goal of fitting an Internet-worked scenario, where a variety of operational systems (which made available different and heterogeneous services) can be used by workers, who possibly belong to different organizations and are located in different (perhaps geographically-distributed) places.

In particular, the architecture we propose 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). In the following, we detail principles, structures and functionaries of these layers.

Operational systems are located in the OS layer of the architecture. Each operational system may 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 Section IV. 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., [20,21]), like *Data and Knowledge Grids* [22], 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 in the following part of the paper.

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 operational systems of 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 subscribes that kind of EE. In turn, the latter EAI snippet will consequently update contents of its associated data repository at the OS layer. This way, actual knowledge are processed, and new knowledge is created.

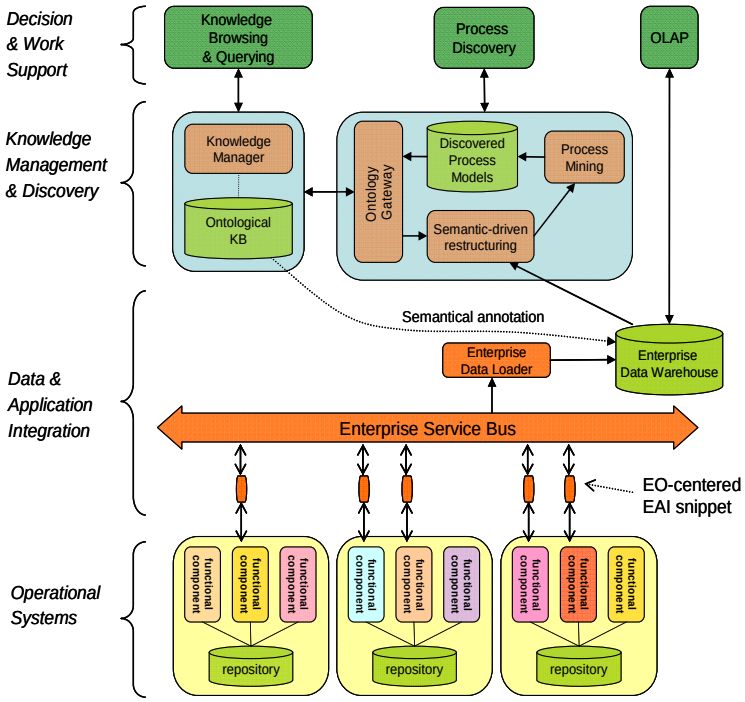

Fig. 3 The reference conceptual architecture implementing the proposed knowledge-based framework

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 Section IV) 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. In particular, such a capability 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 Section I, 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 Section V. 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 concluding 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 [16,19] suggest. However, this issue is outside the scope of this paper, and thus postponed as future work.

IV. Improved Representation of Enterprise Events via a Suitable Ontology-based Framework

From Section III, where the reference architecture that implements our proposed knowledge-based framework for supporting and analyzing LSCPs, recall that, at the KM&D layer, we introduce meaningfully interrelated ontologies on EEs occurring in the target (virtual) collaborative organization. In particular, as highlighted in Section I, at this stage our main research innovation is represented by the fact that these ontologies, beyond proper EEs, also capture organizational concepts/entities associated to EEs.

Several previous studies have already recognized the evidence stating 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.

To this end, in our proposed knowledge-based framework we exploit the ontology-based modeling framework presented in [11], which provides a semantic infrastructure for the management of organizational knowledge, yet supporting interoperability among existing operational systems populating the target (virtual) collaborative organization. Briefly, the framework [11] 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.

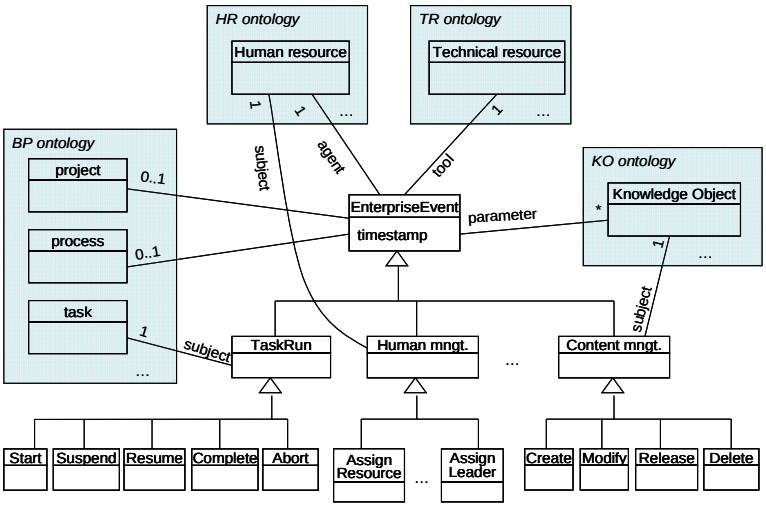

Fig. 4 Improved ontology-based modeling of EEs

Figure 4 provides a (incomplete and approximated) view of an EE and related organizational concepts/entities with regard to the context of a typical IT project, modeled by means of the approach [11]. Here, EEs are distinguished in three sub-classes, according to a functional perspective: (*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. Each one of the three main EE

sub-classes above is in turn hierarchically organized in other sub-sub-classes, whose meaning and semantics are both clearly intuitive. Furthermore, the correlation of EEs with some major concepts of the EKM is remarkably emphasized in Figure 4. Each EE refers to the software (represented by the association *Tool*) that has originated and/or to the person (represented by the association *Agent*) that has performed the event itself, which are modeled by 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, which we describe next.

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 permits us 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 shown in Figure 4 can be semantically annotated with concepts coming from some suitable domain ontology. As an example, consider Figure 5. Here, an ISA taxonomy for project tasks is intuitively depicted. More specifically, all internal nodes of the taxonomy illustrated in Figure 5 represent high-level concepts of a typical IT project, while each directed edge denotes a generalization link between the associated concepts – e.g., the class *Testing* is modeled as a sub-class of the class *Development*. Contrary to this, all leaf nodes of the taxonomy illustrated in Figure 5 represent "real" task instances (i.e., coming from effective, real-life tasks performed in the target project), which are modeled according to the EE conceptual model of Figure 4, and stored in the EDW (see Figure 3). Particularly, each of these task is semantically annotated by one of the concepts in the project taxonomy (dashed edges in Figure 5). Coming again to our reference architecture (see Figure 3), the semantic annotation task is fulfilled by the KM module across the KM&D and the D&AI layer (see Section III).

As a final concluding remark concerning the ontology-based modeling of EEs and associated knowledge, here we notice that other approaches alternative to [11] could have been adopted. For instance, a valid alternative is represented by the ontological framework presented in [18]. This framework, which is mainly focused on emerging challenges that arise in actual *Semantic Web* research, allows us to represent event logs, process mining tasks, and various kinds of knowledge about a wide spectrum of application domains, organizations and IT infrastructures.

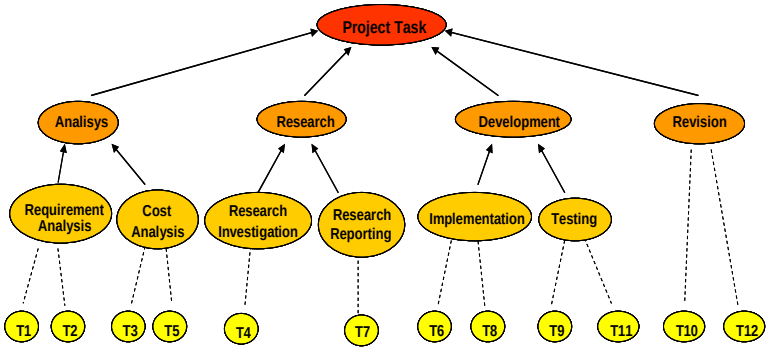

Fig. 5 An example IT project taxonomy

V. Restructuring Enterprise Event Logs for Effective Process Mining Analysis of LSPCs

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 Section II). 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 a series of 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. Contrary

to this, in our loosely-structured collaborative e-work scenario such information is not available whenever process activities are not performed with respect to a well-specified and well-structured workflow process. 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 high-level WorkflowModelElement elements of the MXML model capturing the target (restructured) process log (see Section II). The main assertion of step (*c*) of our EE restructuring approach views WorkflowModelElement elements in terms of basic logical tasks that constitute 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.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <WorkflowLog xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="WorkflowLog.xsd" description="Created by KMS+ ProM
  framework">
  <Source program="ATM" />
- <Process id="project_task" description="">
  - <ProcessInstance id="KMS-plus" description="">
    - <AuditTrailEntry>
        <WorkflowModelElement>RequirementAnalysis;\n</WorkflowModelElement>
        <EventType>Complete</EventType>
        <Timestamp>2007-01-20T14:00:00.000+01:00</Timestamp>
        <Originator>Frank</Originator>
      </AuditTrailEntry>
    - <AuditTrailEntry>
        <WorkflowModelElement>CostAnalysis;\n</WorkflowModelElement>
        <EventType>Complete</EventType>
        <Timestamp>2007-03-31T14:00:00.000+02:00</Timestamp>
        <Originator>Thomas</Originator>
      </AuditTrailEntry>
    - <AuditTrailEntry>
        <WorkflowModelElement>ResearchInvestigation;\n</WorkflowModelElement>
        <EventType>Complete</EventType>
        <Timestamp>2007-04-02T14:00:00.000+02:00</Timestamp>
        <Originator>Lewis</Originator>
      </AuditTrailEntry>
    </ProcessInstance>
    ...
  </Process>
</WorkflowLog>
```

Fig. 6 A fragment of the MXML-formatted process log obtained by restructuring the EEs coming from the research projects KMS-plus and PROMIS

In order to illustrate our EE restructuring approach with the help of a practical (toy) case study, here we refer to a selection of EEs of kind *TaskRun* (see Section IV) concerning the enactment of two research projects developed at ICAR-CNR and University of Calabria, Italy, namely *KMS-plus* and *PROMIS*, which both focus on knowledge management models and methodologies in the context of real-life large organizations acting in a wide spectrum of application domains ranging from manufacturing enterprises to agro-alimentary sectors and logistic companies, and so forth. The IT project taxonomy used to semantically annotate organizational entities of the example EEs is again the one shown in Figure 5.

Figure 6 shows the MXML log which has been obtained by applying the EE restructuring approach described so far to the running case study. For the sake of simplicity, restructured EEs are shown in the form of a table, whose columns correspond to some of the concepts embedded in the model MXML (see Section II). In particular, the restructuring task has been carried out as follows. After selecting only EEs of class *Complete*, which captures EEs related to completed project tasks (see Figure 4), we grouped them based on their associated projects (*KMS-plus* or *PROMIS*), while defining the WorkflowModelElement nodes to simply being correspondent to the executed tasks. These latter, however, have been represented in an as-more-as-possible abstract way, by replacing each of the task label occurring in the EE log with the *most specific concept* associated to the task, according to the IT project taxonomy shown of Figure 5. As a consequence, the resulting process log just consists of two

process instances (see Figure 6), each of which describes the sequence of tasks executed during one of the two projects, at an abstraction level higher than the original process log. This clearly confirms to us the effectiveness and the benefits deriving from our EE restructuring approach.

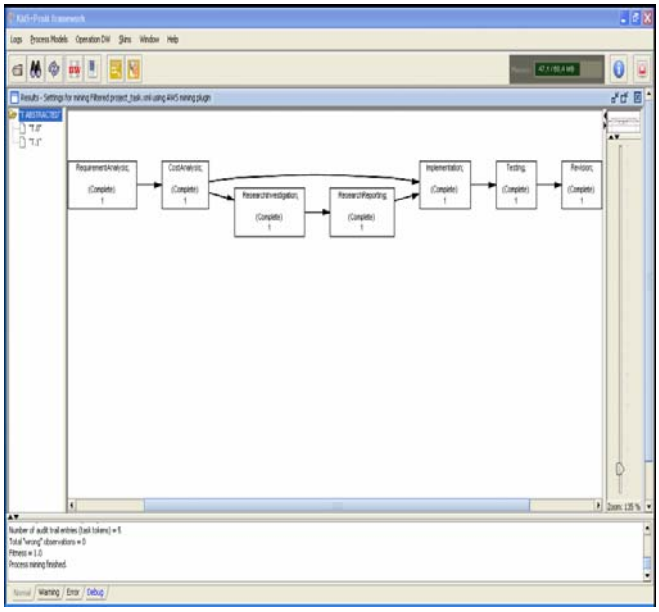

Fig. 7 A taxonomical process model discovered from the restructured process log of Figure 6

VI. Analysis of the Mining Effectiveness of Restructured Enterprise Event Logs

In order to further explore potentialities and possible limitations of our EE restructuring approach, in this Section we propose two simple-yet-effective analysis where we test the effectiveness of restructured process logs under the execution of two different mining methods, namely the *algorithm for mining hierarchical models* [8] and the *clustering-based process mining technique* [9]. We name as *mining effectiveness* the capability above.

Figure 7 reports the hierarchy of process schemas obtained by applying the algorithm for mining hierarchical model [8] to the restructured process log of the running case study whose fragment is depicted in Figure 6. In particular, the structure of the hierarchy is shown on the left side, while the root schema is illustrated in the right side in terms of a workflow graph whose nodes correspond to abstract tasks. Despite the simplicity of this analysis (each leaf schema in the hierarchy just models a single project between the two possible ones, *KMS-plus* and *PROMIS*), it is worth to appreciate the potentialities of such a taxonomical model for supporting the design/refinement of process ontologies, as well as for consolidating the representation of the different collaborative schemes occurring in the virtual organization. On the other hand, discovering a hierarchal model that contains two classes exactly again confirms to us the merits of our EE restructuring approach even from the mining perspective proposed in [8].

Figure 8 shows instead three different work models discovered by means of the clustering-based process mining technique [9] for the case of a real-life collaborative manufacturing scenario studied in another research project developed by ICAR-CNR and University of Calabria, Italy, namely *TOCAI.it*, which focuses on ontology-based knowledge management models and methodologies in inter-organizational environments. In order to discover such models, we again applied our restructuring approach to EEs originated by the executions of processes of the target (virtual) collaborative organization. In more detail, we preliminary selected a log of the operations performed in a two-month period over a distributed CAD platform. Among these operations, we considered the following ones: *Creation*, *Construction*, *Modification*, *Test*, *Release*, *Deletion*, with obvious meaning and semantics We then transformed these data into a workflow-oriented log by considering each artifact as a distinct process instance and, in order to label workflow nodes, we only considered the information about the person that executed the actual operation. In particular, in this analysis, we represented these persons in an aggregated way, by replacing each of them with the team she/he belonged to, based on a given organizational model representing the partitioning of workers in teams. Interestingly, models in Figure 9 collectively capture three different working scenarios occurring throughout the life-cycle of a collection of analyzed artifacts, and help in recognizing some typical collaborative work patterns that relate different teams among each others. As an instance, the model in Figure 9 (a) evidences the fact that, during the development of a number of artifacts, only three teams have been involved, and that these teams have worked in a “pipeline” fashion. Note that, for privacy reasons, real group names have been hidden in Figure 9, and replaced with fictitious labels. Notably, the effectiveness and the benefits due to our EE restructuring approach have been again confirmed even from the mining perspective proposed in [9].

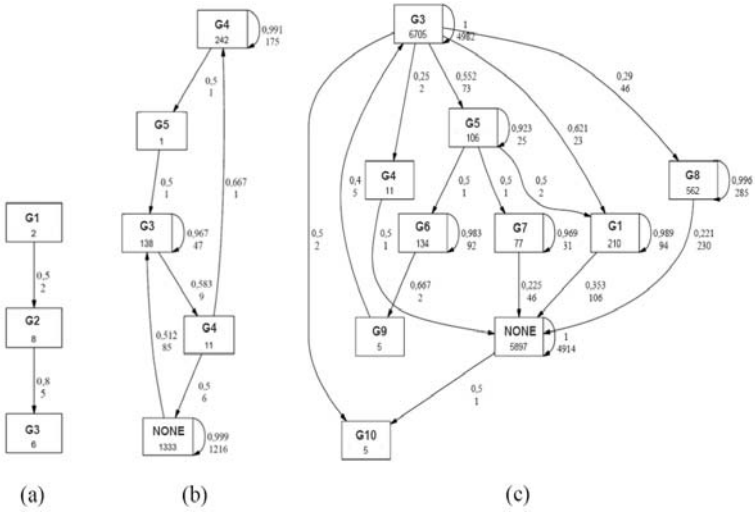

Fig. 8 Collaborative work models discovered by means of the clustering-based process mining technique [9] over restructured EE logs

VII. 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 proposed framework has been also tested against real-life application scenarios investigated by several research projects focused on knowledge management models and methodologies in real-life large organizations, even in an inter-organizational manner. Preliminary results obtained from these real-life scenarios evidence the capability of our process mining analysis approach over LSCPs 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 (*i*) extending our proposed approach by means of mechanisms able of supporting a collaborative and distributed construction of ontologies, and (*ii*) adopting modern Semantic Web technologies within the core layer of our proposed knowledge-based framework for LSCPs.

REFERENCES

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

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

[3] O. Anya, H. Tawfik, and A. Nagar. A Conceptual Design of an Adaptive and Collaborative E-Work Environment. In *Proc. of 1st Asia International Conference on Modelling & Simulation (AMS'07)*, pages 148—154, 2007.

[4] R.P. Biuk-Aghai, and I.T. Hawryszkiewycz. Analysis of Virtual Workspaces. In *Proc. of International Symposium on Database Applications in Non-Traditional Environments (DANTE'99)*, pages 325—332, 1999.

[5] B.F. van Dongen, A.K.A. de Medeiros, H.M.W. Verbeek et al. The ProM Framework: A New Era in Process Mining Tool Support. In *Proc. of 26th International Conference on Applications and Theory of Petri Nets (ICATPN'05)*, pages 444—454, 2005.

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

[7] F. Folino, G. Greco, A. Guzzo, and L. Pontieri. Discovering Multi-Perspective Process Models. In *Proc. of the 10th International Conference on Enterprise Information Systems (ICEIS'08)*, pages 12—16, 2008.

[8] G. Greco, A. Guzzo, and L. Pontieri. Mining Hierarchies of Models: From Abstract Views To Concrete Specifications. In *Proc. of the 3rd International Conference on Business Process Management (BPM'05)*, pages 32—47, 2005.

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

[10] A. Gualtieri, F. Folino, G. Greco et al. Knowledge Discovery and Classification of Cooperation Processes for Internetworked Enterprises. In *Proc. of 4th Italian Conference of the Italian Chapter of AIS (itAIS'07)*, 2007.

[11] A. Gualtieri, and M. Ruffolo. An Ontology-Based Framework for Representing Organizational Knowledge. In *Proc. of 5th International Conference on Knowledge Management (I-KNOW'05)*, 2005.

[12] C.W. Gunther, and W.M.P. van der Aalst. Finding Structure in Unstructured Processes: The Case for Process Mining. In *Proc. of 7th International Conference on Application of Concurrency to System Design (ACSD 2007)*, pages 3—12, 2007.

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

[14] M. zur Muehlen. Process-Driven Management Information Systems - Combining Data Warehouses and Workflow Technology. In *Proc. of the 4th International Conference on Electronic Commerce Research (ICECR-4)*, pages 550—566, 2001.

[15] J.J. Jeng, and J. Schiefer. An Agent-Based Architecture for Analyzing Business Processes of Real-Time Enterprises, In *Proc. of the 7th International Enterprise Distributed Object Computing Conference (EDOC'03)*, pages 86—97, 2003.

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

[17] C. McGregor, and J. Schiefer. A Framework for Analyzing and Measuring Business Performance with Web Services. In *Proc. of 2003 IEEE International Conference on E-Commerce (CEC'03)*, page 405, 2003.

[18] C. Pedrinaci, and J. Domingue. Towards an Ontology for Process Monitoring and Mining. In *Proc. of the 2007 Workshop on Semantic Business Process and Product Lifecycle Management (SBPM'07)*, pages 76—87, 2007.

[19] H. Pinto and C. Tempich and Y. Sure, 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* (ECAI'04), pages 393—397, 2004.

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

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

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