

Enabling Support of Collaborative Cross-enterprise Business Processes for Legacy ERP Systems

Gundars Alksnis¹, Ērika Asņina¹, Mārīte Kirikova¹ and Egils Meiers²

¹Institute of Applied Computer Systems, Riga Technical University

²Visma Enterprise Ltd.

{Gundars.Alksnis, Erika.Asņina, Marite.Kirikova}@rtu.lv, egils.meiers@visma.lv

Abstract. In order to create innovative business products, share knowledge between people and businesses, or increase the control and quality of services, more and more often enterprise business processes involve in collaborations by delegating or providing some pieces of work to other enterprises. Necessity to cooperate in the cross-enterprise setting leads to Collaborative Business Processes (CBPs). The difference between CBPs and Business Processes (BPs) is in the decentralized coordination, flexible backward recovery, participants notification about the state, efficient adaptability to changes, presence of multiple information systems, and individual authorization settings. In the paper we consider a specific case of CBPs where multiple collaborating partners use Enterprise Resource Planning (ERP) system of the same vendor. The vendor can see (e.g., monitor) the changes of data elements, but does not have explicit process awareness in the ERP system to support flow of activities in the cross-enterprise setting. The paper also discusses different settings of cross-enterprise CBP and shows simplified enterprise models behind the vendor possibilities to positively impact collaborative processes. The restrictions of the vendor are implicit information flows in BP, diversity of ERP integrations with third party Information Systems (IS), the lack of mechanisms for monitoring BP instances, backward recovery, user notification about the current state and tasks, and inability to make explicit changes in customers' ISs.

Keywords: Cross-enterprise, collaborative business process, business process monitoring, legacy ERP system, leading system architecture.

1 Introduction

In order to create innovative business products, share knowledge between people and businesses, or increase the control and quality of services, enterprises need to collaborate with each other, thus delegating or providing some pieces of work to other enterprises [1]. First introduced in [2], such processes are named as Collaborative Business Processes (CBP). The specifics of CBPs is that they are not coordinated by a single workflow (WF) engine, but rather by multiple engines collaboratively. Research on CBPs has two aspects. The first is personal collaboration because it includes support for personal negotiations, personal knowledge sharing as well as coordination and planning of activities (e.g., in the so-called industrial clusters [3]) by using networks, voice, video and audio IT solutions (e.g., [3]–[6]). The second, let us call it “computer system driven” collaboration, includes support of a business-to-business (B2B) Workflow Management System (WMS) that “defines, creates and manages the execution of workflows through the use of software, running on one or more WF engines, which is able to interpret the process definition,

interact with workflow participants and, where required, invoke the use of IT tools and applications” [7], like FITMAN [8]. CBPs are more dynamic than sequential processes without collaboration and involve more complex communications between enterprises, especially in relation to non-functional aspects [1].

CBPs are also identified as cross-enterprise [9], cross-organizational [10], multipartner [11], and intra-enterprise [2], [12] processes. The main distinction from inter-enterprise CPSs is that they involve several autonomous enterprises in the business process (BP) execution and without a single centralized management entity. The process choreography is not trivial, since business independency and sometimes unwillingness/inability to introduce changes in the already implemented IT solutions are at the fore for our business case [2].

Among others, Enterprise Resource Planning (ERP) systems also support cross-enterprise CBPs [13]. It might have specific restrictions (discussed further) that make it hard to control and monitor such processes, e.g., private/public and explicit/implicit information flows, diversity of ERP integrations with customers’ information systems (IS), a lack of mechanisms for collaborative process monitoring and backward recovery, and inability to make changes in customers’ ISs.

In this paper we provide extended information from the [39] and focus on the following research question: “Can ERP system vendor support collaborative cross-enterprise processes for its customers?” Such question arises because, on one hand, a big amount of information is processed with the help of ERP systems during the collaboration, however, on the other hand, the cross-enterprise process itself as a whole is hidden from the particular ERP vendor. There can be situations that the process could be documented and the ERP vendor thus would be able to follow in the process enhancement services. In most cases the CBPs still are not or even cannot be predefined as they include spontaneous behavior. In order to answer our research question, we have carried out a review of techniques and mechanisms which are used for cross-enterprise CBPs. The results were used during experimental development activities to define Leading System architecture techniques and mechanisms for practical implementation of BP monitoring and user guidance.

The paper is organized as follows. Section 2 provides a brief discussion of the cross-enterprise CBPs and their management. Section 3 illustrates the context of the ERP business case and its restrictions. Section 4 presents a view on existing solutions within the CBP management regarding the ERP business case. Sections 5 outlines our experience with practical applications of CBP management. In section 6 we conclude with the summarization of main results.

2 Cross-enterprise Collaborative Business Processes

Usually B2B collaboration systems are statically configured. As a consequence, it is not possible to adjust BPs dynamically according to the constantly changing needs and constraints of the businesses. Instead processes are described in some hardcoded notations, e.g., Business Process Model and Notation (BPMN). Some researchers try to tackle this issue by applying semantic business process management (BPM) approaches, e.g., hierarchical task networks and Web ontologies [9], [14]. However, they are dealing rather with the task of generation of Business Process Execution Language (BPEL) specifications, but not with the task of running and monitoring execution processes thereafter.

BP monitoring has been widely applied in ISs [15]. Usually it involves some form of event-based processing, when it is possible to attach some action(s) before/after an activity that is triggered by an event. Such events are mostly used for monitoring the compliance within the enterprise, but they also can be used to send notifications to the external endpoints, e.g., for informing other systems/enterprises. Such systems usually utilize some form of a rule/condition engine, so that the events are fired only when the condition is met. However, in order to make a decision regarding particular implementation, it is important to evaluate its appropriateness in the

context of the used WF engine, e.g., whether addition of event processing involves changes in the BP definitions or can be applied transparently to it.

As mentioned in the introduction, in the cross-enterprise processes non-functional requirements, like privacy and confidentiality, might play even more crucial role than functional requirements. For managing interests and protecting privacy of the involved enterprises, the CBP management system should provide different perspectives/views for a particular party. However, such adoption of views inevitably increases the complexity of overall process representation and maintenance of ever changing BPs.

We are mostly concerned adding event processing and data state identification capabilities to the legacy/non-process oriented ISs. Namely we need to know how to ensure monitoring and runtime user guidance without influencing legacy applications and still be able to apply some form of BP modeling and integration with third party workflows. Currently there are at least four paradigms to BPM [16]:

- Activity-flow oriented, when the activities and their predefined sequences are used at firsthand, but the data that must be processed are perceived as second-class citizen [1, 2, 17];
- Document oriented, when it is important to finish some document using a strict authorization mechanism, document partitioning and templates [18]–[20];
- Case handling, when it is important to resolve a case without necessarily specifying the order of activities, obtaining some minimum amount of data or making predefined number of decisions [21], [22];
- Business artifact-centric (a special case of data-centric paradigm), which combines and models both data (information model) and process (lifecycle model) aspects as a single unit [16], [23]–[28].

The first paradigm is quite mature already and implemented in many enterprise ISs. Several WMSs offer general modeling and execution of structured BPs. Besides pure WMSs, ERP systems such as SAP, Oracle, Baan, and PeopleSoft have also adopted this technology [21].

The second and the third paradigms are dedicated for knowledge-intensive applications, where decision making could be based on unspecified data and facts, and usually involves mental activities or where a WF highly depends on the runtime context [21], [29]. The document-oriented paradigm provides both a flexible decision making and a strong authorization mechanisms. The case handling aims to provide rather full data than partial one, allows editing data before and after an activity is executed, and decides about availability of activities based rather on the current information than on the previously executed activities.

The last paradigm, originally proposed in [23], initiated many further studies and applications because of “a natural modularity and componentization of business operations and varying levels of abstraction” and the familiarity of the artifact concept to the business people [16]. Business artifacts can be mapped onto WF engines, thus enabling attachment of other BPM capabilities, including specification and monitoring of business rules and key performance indicators (KPIs) [16]. Artifact-centric research continues developments, especially in the context of the declarative style for lifecycle specification and their tool support [24].

WF monitoring starts with the execution of a BP instance in the WF engine. Usually it is implemented in a form of logging more or less complete event information. Another form that must be mentioned is process mining from event logs [30] or execution records [29]. Modern process mining tools use the control-flow dimension of a WF by using Workflow Petri Nets [29]. Within ERP.Bi-partial monitoring agents (probes) can be used [31], [32]. They contain a memory for BP allocation and a logic for BP monitoring. However, deployment of agents over a business process may be hard to automate.

3 The ERP Vendor's Business Case

ERP systems can be divided into two groups, namely ERP I and ERP II [13]. While ERP I systems focus on integration of back-office ISs, ERP II systems (which we consider here) focus on collaboration among companies and their customers, serve all sectors and have open web-based architectures. ERP II are complex systems that have more difficulties in coordination among partners as well as in management and assessment, and, thus, can lead to more frequent failures [13].

Let us illustrate the general features of the business case we plan to discuss. An ERP vendor has many customers, mostly, small and medium size enterprises (see Figure 1). Each customer uses the vendor's ERP solution for his specific business needs, integrates it with its own and third party ISs and ERPs, and collaborates with other enterprises. The complexity of the case is that a single BP may involve several participants from different enterprises, which collaborate to obtain some business value, as well as may involve functions provided by other vendors' ISs. However, the ERP vendor has the ability to change/enhance only the functionality of his own product.

Let us look at the simplified example of the cross-enterprise CBP (see Figure 2). Participant 1 initiates the business process, executes own activities, and then requires collaborative activities from other enterprises, e.g., agreement of Participant 2 and approval of Participant 3. In the worst case, Participant 1 might not even know about the necessity of approval by Participant 3, since this could be Participant's 2 confidential information. Besides, Participant 1 cannot monitor other participants' activities and own documents until the final status of them will be available for all allowed participants.

From the ERP vendor's viewpoint, this problem looks even more complicated. User tasks may be automated in the ERP and other ISs. There is no automatic coordination between tasks, i.e., only the experienced users might know about the correct order of tasks and how to conduct them. The participant who initiated the WF (Participant 1) does not know what happens within the business process until the last "link" in the task chain is completed (e.g., registering the completed document in the database by Participant 3). Therefore, there is the necessity to know how the collaborative business process is executed in reality. In our case, it could be some external monitoring process that follows changes of data and BP states.

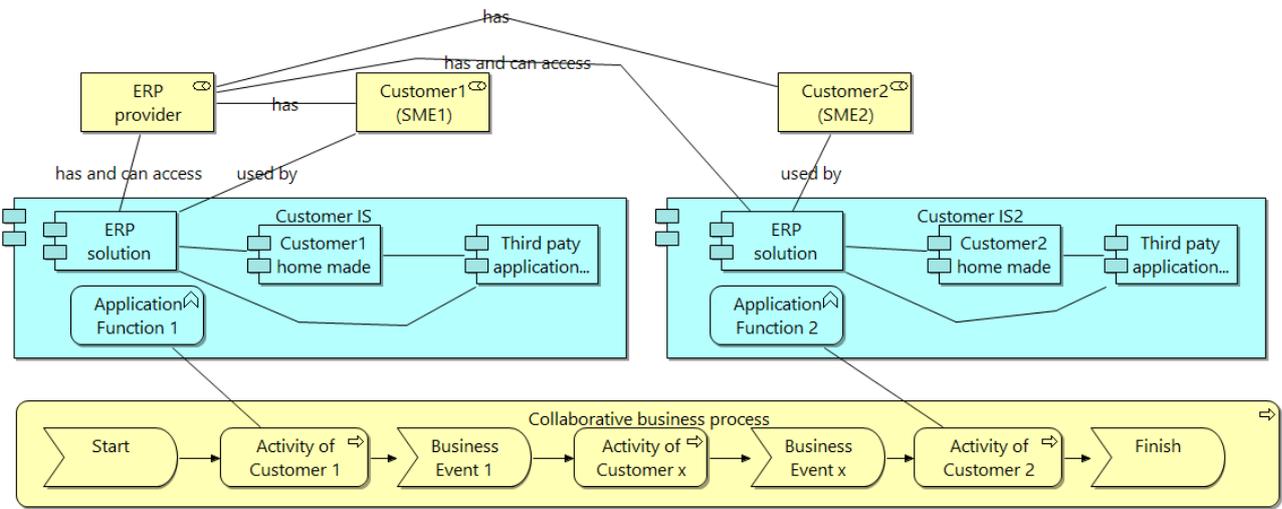


Figure 1. ERP vendor and its customers which are involved in cross-enterprise business process

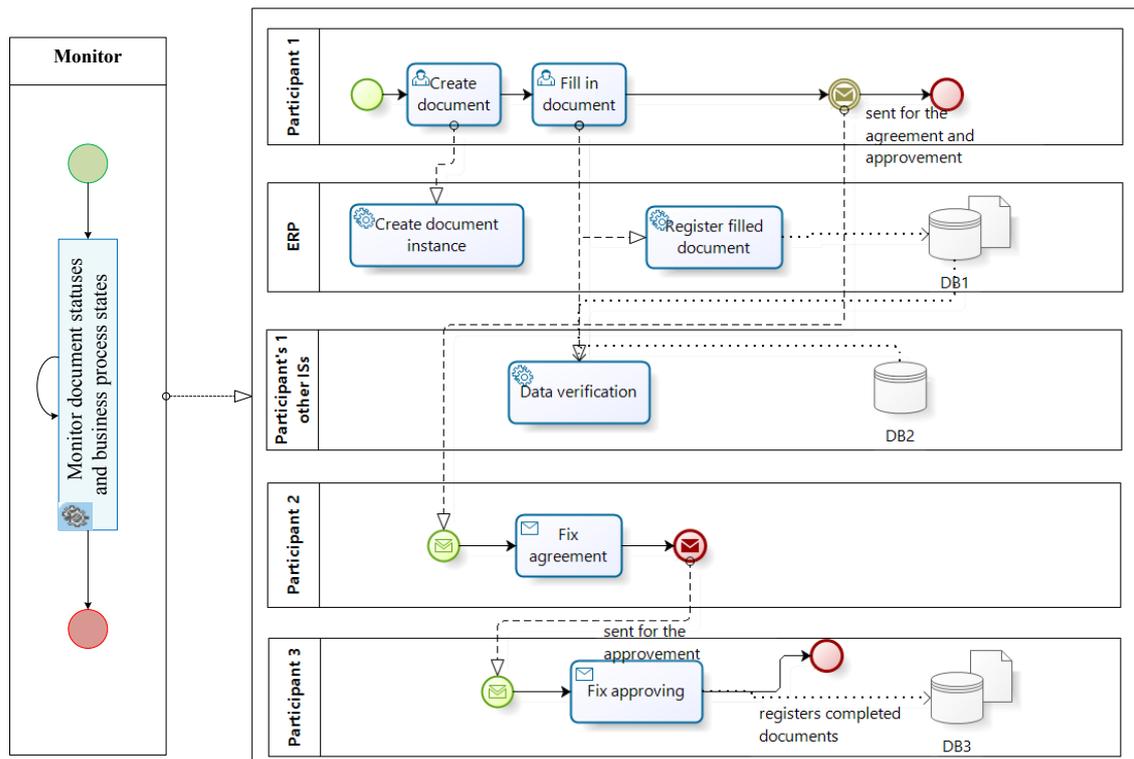


Figure 2. The simplified CBP from the ERP vendor's viewpoint

In the context of legacy systems, when WF engine might not even exist, or when the knowledge about business processes are only in the users' heads or in some decoupled descriptions, knowing how a process is being used in reality becomes even more crucial. If it would be possible to detect changes in the business objects or obtain information about ongoing/completed activities, this information could be used to obtain some form of user guidance; for example, by listing the activities a particular user should perform. We should note once more, that the particular ERP vendor's customers mostly are small and medium size enterprises that might be not able to make substantial investments into adapting customized BPM products.

To sum up, the main issues of the business case of the ERP vendor that we are analyzing further in this paper are partly in the field of information logistics:

- Each participant has its own ERP solution, integrated with other ISs. The ERP vendor has no access to these systems' internals. It is not welcomed to change the existing IT solutions, and it is not possible to change other ISs.
- There is no mechanism to monitor the current state of the business processes and data, if they are under other participant's or system's control. Only the direct user of the corresponding system is able to find out this information by querying the data. Other participants don't have such rights due to access control restrictions.
- There is no notification mechanism about consequent participant's tasks even when the document or activity is completed by another participant, and the participant has access to the data.
- Information flows are not transparent for participants. Experienced users do know their own tasks and task execution order, but are not able to see the whole picture. Thus, a participant who has initiated the business process might not know about all other participants which are involved as well as about the results of execution of their functions or tasks within the business process.
- There is no informational support regarding the required actions within the task execution, only experienced users' knowledge.

Our idea is to introduce a runtime service for user guidance to support participants' work with the vendor's ERP by monitoring the current state of BPs based on the data in the ERP system. One approach to tackle this case is to use some form of monitoring data changes in the data store (database). However, usual approaches that use trigger or transaction log analysis mechanisms might not be sufficient, because it is not possible to detect all kinds of data usage with them. For example, selection queries are not logged in the transaction logs. Instead we must use specialized database activity monitoring modules (e.g., Microsoft's SQL Server database Change Tracking or Oracle's Audit Vault and Database Firewall).

Typically ERP systems provide a registry of accumulated business information. Business processes in such systems are "encoded" in the form of information entries and their validation. On the one hand, this enables various scenarios for information entry. On the other hand, however, implementation of guided user experience might not be a trivial task. Another requirement related to the backward recovery is in the context of compensation activities that must be performed during the lifecycle of the business object in order to restore its state to some point in history.

Next sections are devoted to the review of existing solutions in the field of CBPs and analysis of their suitability for our business case.

4 Existing Solutions in the Context of Collaborative Business Processes

Accordingly to the issues and corresponding constraints mentioned in the previous section, we have defined the following criteria that we will use further for paradigm analysis and solution selection: maturity of the implementation technique, specification languages, execution principles, monitoring mechanisms (active or passive), easiness of both current state and task identification, flexibility of decision making mechanisms, communication and BP coordination/synchronization mechanisms, authorization and privacy, integration with other ISs, and a compensation mechanism (transaction backward recovery).

4.1 Activity-flow Oriented Paradigm

Regarding the activity-flow paradigm, we discuss three techniques. After discussing the techniques we present a fragment of the generic enterprise model which depicts the main entities involved in monitoring activities according to the activity-flow paradigm.

The first technique is peer-to-peer collaborative business process management [2], [33], where a role [17] or an agent [2] can create an instance of the CBP, initiate own "peer-side" instance, and then notify a peer to instantiate the peer's side instance. In case of roles, the workflow must be well-understood, specified and shared by all participants. Holding privacy of enterprise activities within the workflow is the challenge in this case. In agent-based processes, an agent must know its communication paths with other agents and a corresponding part of the workflow state space. For roles, the workflow could be specified using Collaborative Process Definition Language (CPDL) in an XML document, then compiled to Data Object Model (DOM) tree of Java objects, then to Java classes. For agents, it could be any executable workflow specification. The process is executed as a set of peer process instances (which share the same process definition and may have private data and sub-processes) run by process management systems of the participants [2]. Monitoring is realized as a collaboration among multiple engines which share one common predefined workflow specification. Besides, each agent has its own engine and querying server which collaborates with other agents' querying servers, thus allowing the agent to obtain data about the current process state and data statuses. Another way is implementation of monitoring agents within the ERP system that collect metrics from the ERP database and log files, perform an analysis and inform the ERP users about the results [31]. The decision making is decentralized and could be based on predefined roles within the shared

workflow [2] or on a set of predefined links to partners related to the possible process states [17] which contain also data about requested and suggested tasks. Communication is implemented as the peer-to-peer messaging mechanism [2]. However simple message interchange (at the conversational level) is not enough when a workflow is complex and includes intra- and inter-enterprise collaborations. In such case process-level coordination is critical. Authorization can be role-based [2], when each participant has its own role within the shared workflow and the rights to manage access permissions within the role. If the multipartner collaboration uses the peer-to-peer messaging mechanism, then the solution could be embedded as a set of separate external WMSs [2], or completely implemented in the single WMS [17] that controls and coordinates all process invocations.

The second technique is workflow-view-based process collaboration [1], where processes are views on the workflow from the participants' viewpoint, and their instances may form a network. Still, the common synchronization means is defined by the workflow specification. The specification could be written in BPMN, Web Services BPEL (WS-BPEL), Electronic Business using eXtensible Markup Language (ebXML), or RosettaNet, and then it can be run in WMS [1]. Unlike [2], the specification is implemented and executed by a single WMS engine, and communication is controlled and managed by this engine. The open challenges are analysis of visibility constraints between entities, process view coordination, and deployment of process views for partner organizations [1]. Monitoring uses tracking of a network of process instances that have multiple collaborative relations with each other. Unlike [2], in a variety of workflow views, each process has its own view on the common workflow and may change authorization settings. Tracking in a network of process instances could be a useful compensation mechanism for backward recovery, notifying corresponding partners about failures.

Another kind of views on the workflow are web services (WS). Their implementation, monitoring and coordination is well elaborated during previous years. However, currently service networks might not correspond to business services and their combination requires good understanding of their business properties (ontology based semantic services is the promising answer to this issue), and their coordination is not flexible. Specifications, like WS-BPEL and Web Service Choreography Description Language (WS-CDL), use predefined processes or rules to solve questions on inputs/outputs, message correlations, etc., but do not take into account dynamics of business processes in collaboration. They require more transactional support because standards and protocols like BPEL4WS, WS-Coordination, WS-Transaction, Web Service Choreography Interface (WSCI) and WS-CDL have too inflexible compensation mechanisms [1].

Looking to all three techniques, we can see that in each case a bit different elements of the enterprise model may be involved, e.g., one or several process definitions (views) may be exploited, or communication paths between actors can be directly considered or not considered. However, depending on the situation it is possible to follow changes in the states of all or part of elements (as shown in Figure 3).

The main difference in the extent to which the ERP vendor can participate in CBP monitoring depends on willingness of participants to share the CBP specification with the vendor. In case the specification is shared, the vendor can implement the service that watches the entire process. In case the specification is not shared, the only way to participate in the monitoring is informing participants about some unusual state changes or delays in state changes of the specific elements.

4.2 Document Oriented Paradigm

The document-oriented paradigm is dedicated for processes, where a document is a “product” of the collaboration and its main element (or the context). Document-centric situational collaboration processes [18] foresee document producing among different departments of an enterprise as well as collaboration with external participants. Control flows specify collaboration

activities, which form iterative and incremental evolution of the document, where each part of the document may be completed by different participants or groups of participants [18]. Documents are specified in document models that should have general and extensible semantics and layout and support reuse and composition of parts as building blocks.

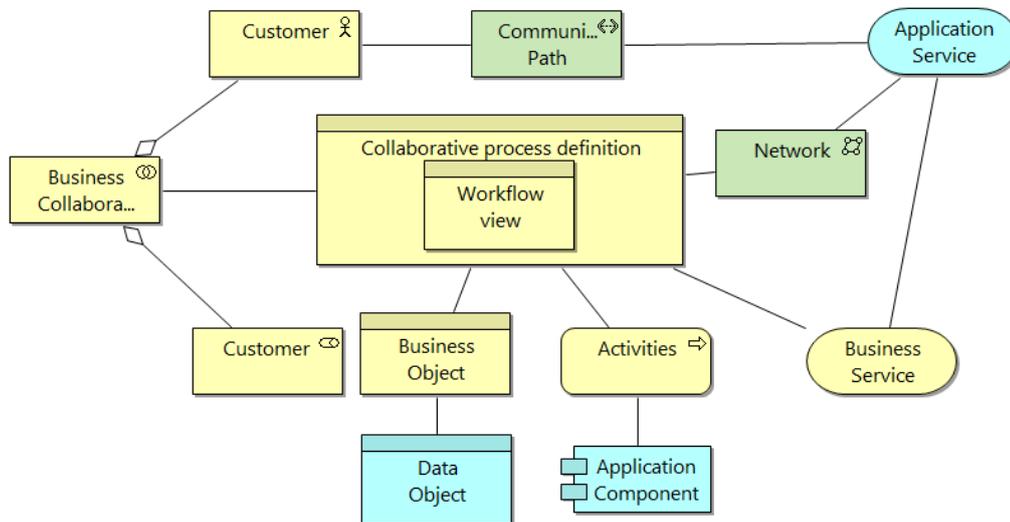


Figure 3. Enterprise model elements for monitoring CBPs (the activity-flow paradigm)

A decision mechanism is flexible. It is based on authoring rights, which are flexible, transparent and usable, and document-related events from participants (persons and systems). The collaboration process itself must be manageable and controllable. In order to control “production” of the document, document services may be used [18]. In such case, the process may be controlled using document service mashups.

Authors in [18] suggest using a Darwin Information Typing Architecture (DITA) model that is specified in XML and dedicated for authoring; and a document (software) service model can be specified by RESTful Services. Collaboration is modeled using an asynchronous event-based communication model, where communication consists of document interaction patterns. Execution of this approach is possible by using web service mashups with Wiki tools. Coordination is implemented using service mashups. The compensation mechanism is not mentioned in [18], but is possible in practice, since collaboration steps are predefined and can be recorded.

In contrast to the communication model in [18], authors of [19] and [20] suggest using agent collaborations borrowed from artificial intelligence. Authors in [19] enhance WFs with “commitments,” thus coordination is implemented using procedural specifications and commitments models. Each commitment between a task sender (creditor) and a task executor (debtor) is specified as inputs, outputs, preconditions and results of the activity. Execution is implemented using Microsoft BizTalk (and BPEL) and BizTalk Orchestrations [19], [20]. Each participant (agent) has one or more roles within the WF (as well as each role may be assigned to different participants), the behavior of roles is specified as an orchestrated WF. However, communication among participants is coordinated using choreography of multiple WFs without a centralized engine and monitoring. Similar agent WF choreography is suggested in [20]. Notification can be implemented using Microsoft SharePoint lists or document libraries [20].

In the document oriented paradigm it is necessary to know what are the authoring rules for the document. These rules can be known from normative documents or can be derived from the process execution history. The same applies to document partitioning and usage of specific document templates.

Figure 4 illustrates what enterprise model elements are relevant for the document oriented paradigm. By monitoring changes in the states of business and data objects these elements with respect to changes of states in related elements can reveal some patterns of states that are not valid or point to the inefficiency of processes. In such cases the document monitoring service may signal possibility to provide guidelines to process executors or warn them regarding particular inconsistencies, time constraints, or other issues.

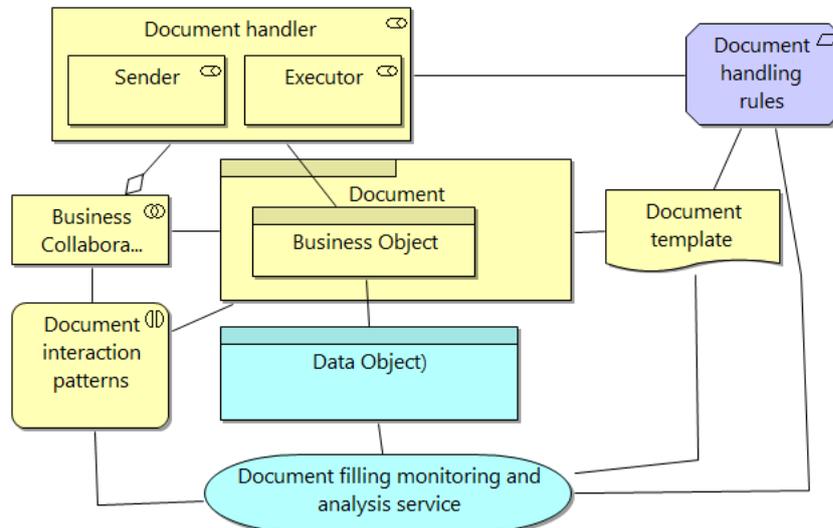


Figure 4. Enterprise model elements to be monitored and vendor's service (document oriented paradigm)

4.3 Case Handling Param

The case handling paradigm [21] is an attempt to solve inflexibility of the standard WF implementations that make knowledge-intensive BP support senseless. This is an attempt to provide a balance between data and processes. As mentioned in [21], limitations of imperative WF implementations are that work needs to be straight-jacketed into WF activities, routing is implemented for both work distribution and authorization, focus on control flows sets aside the context (context tunneling), and WFs have push-oriented perspective, i.e., prescribe what a participant has to do not what he can do.

Case handling have features that mitigate these limitations. They are providing all available information for the case as a whole (avoids context tunneling), enabling activities rather on the assumption of this exhaustive information not on the history of executed activities, separating work distribution from authorization and suggesting types of roles additional to the execute role, and what is very important, allowing workers to view and modify data before or after execution of the corresponding activities [21].

The case handling paradigm is implemented in such tools as FLOWer [34] created by Pallas Athena, as well as mentioned in [21] Staffware Case Handler and COSA Activity Manager. The latter one can be used also as “add-on” application [35].

In the essence, the core element in this paradigm is “a case” [21], [36], e.g., the evaluation of a job application, and the outcome of a tax assessment. This element forms the main context of WF execution and is the “product” which is made. A case has a particular structure and a state based on a collection of data objects which determine its logistical state. A case is produced by

executing “activities.” Control flows between activities are specified as precedence relations (the less number of relations, the better) and modeled, for example, using state-transition diagrams.

In contrast to the activity-flow paradigm where activities are atomic and either executed completely or canceled, here an activity may be executed partially by one participant and then carried out by another participant. Thus, it is a simple chunk of work, where activities (work) are separated by points of “work transfer.” A point of “work transfer” from one participant to another is likely or possible [21]. Activities may have the following states: initiated, ready (for execution), running, interrupted, saved, completed, skipped, and bypassed. The interesting point is that activities and cases follow typical patterns – types, while in the activity-flow paradigm WFs follow the simplest case structure. The process is a method (recipe) for handling cases of a given type.

Coordination is carried out by providing participants with insight in the whole case when they are executing activities. Forms are used to show relevant viewpoint on data objects of the case for a participant. Rights to enter data objects depends on their types: free data objects don't depend on the activities (could be considered as global data objects with the case context) and could be changed during case handling, mandatory data objects are assigned to activities and must be completed in order to execute the activities (before or in the activity), as well as restricted data objects that could be entered only in the corresponding activities [21]. Data objects serve as post-conditions on activities. Activities are completed only if and when all mandatory data objects are entered. Data object life-cycles also may be described by state transition diagrams. A limitation of the case handling is inability to modify case types on the fly, however this could be overcome by creating a new case type based on the type that requires modification [36]. Decision making is based on using Event-Condition-Action (ECA) rules assigned to some event triggered by a user action.

The case handling paradigm foresees grouping and hierarchical organization of roles. Each role is specific to processes, thus allowing multiple roles with the same name assigned to different processes. Participants work as actors and may have multiple roles. For each process and activity three types of roles need to be specified, i.e., the execute role, redo role and skip role. Such separation of role types allows flexible modeling of exceptions as well as serves as a flexible compensation mechanism for cases. Notification of the data and case state to actors is provided by a query mechanism – filtering available information using corresponding data objects and roles [21], [36].

Enterprise elements to be taken into consideration for monitoring purposes are depicted in Figure 5. If cases are known to the ERP vendor, it is possible to establish services that can monitor each case according to the predefined input and output states of data objects as well as chunks of work in the activities according to case definitions, if object lifecycles and ECA rules are given.

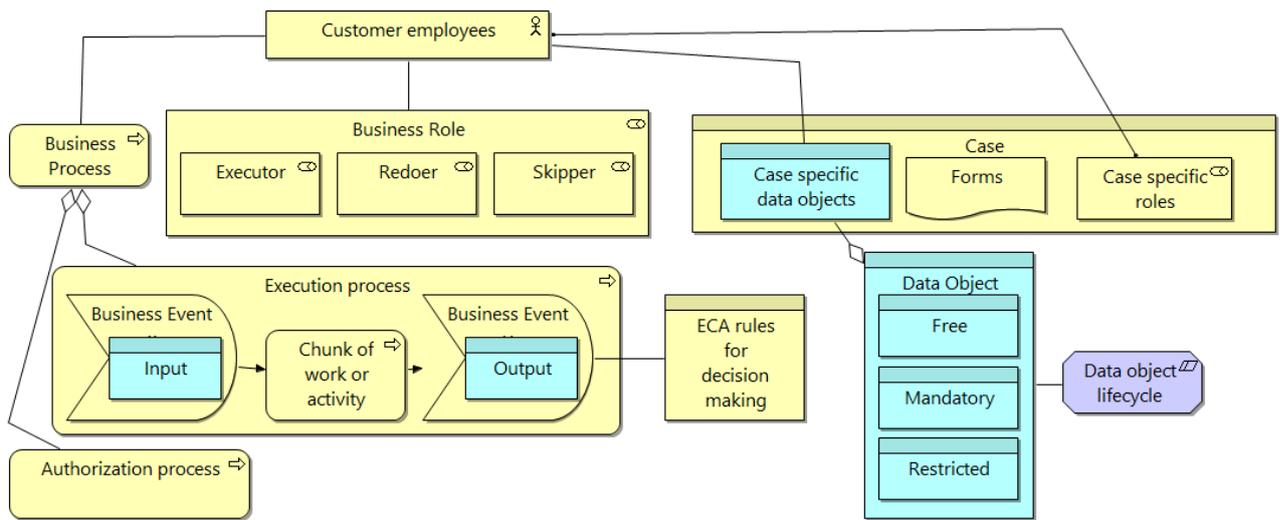


Figure 5. Enterprise model elements for monitoring CBP (the case handling paradigm)

4.4 Business Artifact-centric Paradigm

While the activity-flow oriented business process modeling is oriented on the activity/task flows and considers data flows as secondary, but the document/case oriented modeling is centered on the data. Another very promising data-centric paradigm is the business artifact paradigm that uses a combination of data and process flows as a single building block called (business) artifact [16]. “Artifacts are business-relevant objects that are created, evolved and archived as they pass through a business” [16].

Artifacts are specified by two models (or schemas [26]), namely an information/data model and a lifecycle model [25], [27]. The information model specifies data about the business objects during their lifetime. The lifecycle model specifies tasks and their order (way) in time, when they must be invoked on the business objects in the state-based form. There are two notations for the lifecycle specification [26]. The first one is Finite-State-Machine (FSM) for lifecycles without multiple state paths. The second one is Guard-Stage-Milestone (GSM) for lifecycles with complex branching. Decision making is implemented by rule definitions in form of ECA, where artifact data serve as business terms in the rules. The current state and data statuses are kept in the artifact and business rule instances and are available on demand (Figure 6). Thus, monitoring could be provided in a simple way – by inspecting artifact instances directly [27].

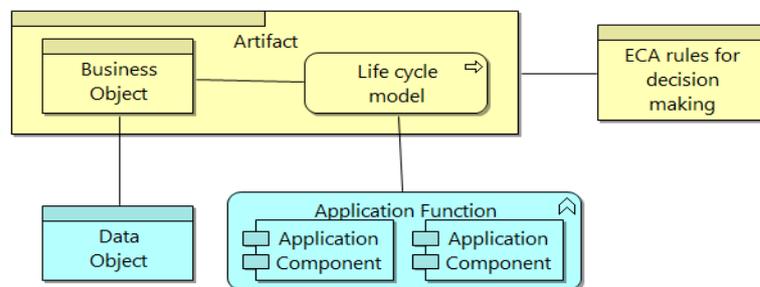


Figure 6. Enterprise model elements for monitoring CBP (business artifact-centric paradigm)

Participants might use only a part of artifact’s lifecycle, however they will be familiar with other stakeholders’ needs and their own needs [25]. In this case coordination could be implemented by using log files and controllers of processes, artifacts, and business rules [27].

Artifact specifications may be mapped into workflow designs and executed by a workflow engine, but this leads to losing information about business rules, since they are degraded to control flows and become hardly manageable [27]. Some authors encourage to use special artifact-oriented management systems [25]–[27]. In such systems business rules may be modified/removed/added at run-time. Authors suggest using two kinds of management systems: procedural and declarative. The imperative (Siena for FSM models [28] and Barcelona for GSM models [24]) systems are similar to the WMS. It is passive, since allows querying and retrieving artifact information, invoking business events and notifying about the pre-subscribed events [26]. Passive systems could be integrated easier than imperative ones. However, current artifact-centric management systems mostly have only prototype implementations. At the present realization of this paradigm in WMS is faster and more mature, however requires additional efforts in monitoring and business rule flexibility [27].

The artifact model is represented as an XML document and interpreted directly [16]. All these systems foresee participants' access control over artifact data [16, 26] using the Create, Read, Update, Delete, Add, Edit (CRUDAE) mechanism. Also the compensation mechanism could be implemented in the context of business artifacts, since all states are known and historical data are logged.

In overall, we see that this paradigm is at the early stage of elaboration and its practical implementation may require additional research, e.g., in the context of databases, concurrency control of artifact instances, and data integrity [16].

4.5 Paradigm Comparison

Since ERP systems are mostly process and data oriented, but quite often also require backward recovery, only first and last paradigms are considered relevant to our business case and will be compared against each other.

By summarizing the feature characteristics (see Table 1), we conclude that conceptually the business artifact paradigm has almost all features of the activity-flow oriented paradigm and also is more flexible. It has a different viewpoint concerning the WF activities and data, but may be successfully mapped to the already matured activity-flow oriented paradigm and existing tools. However, in such case the implementation will lose separate logic of business rules, since it will be distributed within control flows. Another advantage of the business artifact-centric paradigm is in its potential implementation as a passive (e.g., monitoring) system that is one of our business case needs. Still this paradigm is new and its practical realization in an artifact management systems has various technical challenges (we discuss about this in Section 6).

Table 1. Characteristics of the activity-flow and business artifact oriented paradigms [39]

Feature	Activity-flow oriented	Artifact oriented
Technique	mature	immature
Specification (standardization, tool support, levels of abstraction)	standardized, broad WMS support, levels of abstraction (conceptual, design, executable)	not standardized but could be mapped to WMS supported standards, tool support only by prototypes, levels of abstraction (conceptual, design, executable)
Execution (engines, collaboration, contexts)	XML + Java, one or several WEs; process instance or agent collaboration; the WF/process context	the artifact or WF engine; artifact collaboration; the artifact/process context
Monitoring	active within WMSs	active within WMSs or active/passive within artifact management systems
State and task notification	kept implicitly within WF instances (requires additional calculations)	kept explicitly within artifact instances (fast access)
Decision making	glued with control flows (more static)	at the conceptual level is separated; at other levels is glued only if WMS are used
Communication	peer-to-peer messaging, notification by a WE	notification by a WE or artifact engine (in passive

		systems after the request or after artifact state updates)
Coordination	agent/process choreography, shared WF specification, single WF	artifact lifecycle choreography, shared artifact specifications, and, in case of WMSs, process orchestration or choreography
Authorization	roles, agents; private/public processes and data	roles; private/public processes and data
Integration	WE must be tightly integrated with other ISs in order to manage WFs	- if WE, then must be tightly integrated with other ISs in order to manage WFs; - if artifact engine, then may be realized as a passive system.
Compensation mechanism	too fixed for dynamic collaboration if standardized, otherwise requires additional effort	direct retrieval of information from the artifact's historical data

5 Paradigm Usage in Practice

In this section we discuss lessons learned from practical use of two paradigms, namely the business artifact-centric one and the activity-centric one.

5.1 Business Artifact-centric Paradigm Usage

In this sub-section we discuss lessons learned from the modeling sessions that have been done by using ACSI BizArtifact project's I-Hub system [37] and suitability of this solution for our business case.

I-Hub system provides tools to model actionable business artifacts for later execution in the model driven execution engine. Business process definitions are specifications encoded in XML and XSD schemas. Modeling of business processes are done via Web interface. I-Hub composite application consists of a set of business artifacts, an authorization model and a set of external services. Business artifacts are the fundamental building blocks of the composite application. An artifact contains an Information Model, a Lifecycle Model, and a set of Artifact Services.

Despite the information that this tool should be regarded only as a prototype, we carried out a set of modeling sessions, during which tried to model a specific business process of customer debt handling (not discussed here). Initial expectations were high, however during the sessions we encountered various issues that were not described in the tool's documentation or did not work as described. Our modeling sessions stopped at the inability to implement transitions between stages in the Artifact's GSM model, because from the tool itself and provided documentation it was not possible to conclude how events are connected and where milestones are reached. It is important to note that we do not conclude that therefore the business artifact-centric paradigm is not suitable for our business case. It is rather a matter of tool and supporting documentation maturity as well as adoption time, especially in the context of its integration with legacy ERP systems.

As an advantage for the business artifact-centric modeling we can say that it helps to understand lifecycle stages of a business entity, what states entity can enter, basic dependencies between states and what are the preconditions for state activation. Problematic, however, is the definition of relations (dependencies) between stages, because we must test them for all preconditions in order to explicitly prohibit transition cases to the unwanted stages. During the modeling of business artifacts, such tests should be done by the tool in an automated manner.

However the most restrictive was the requirement for storing all business artifact data in the runtime engine itself which complicates implementation in a cross-enterprise setting, because of the need for synchronization of data between I-Hub runtime and external ERP systems. Also at the present, user interface generation and output is possible only via the Web.

Thus at the current stage of the research we decided to postpone further developments in this direction by trying to implement integration with activity-centric WEs first. The proposed conceptual architecture is discussed in the next sub-section.

5.2 Activity-centric Paradigm Usage

In this sub-section we explain why we decided to remain with the activity-centric paradigm and give a conceptual architecture that satisfies most of our business case requirements.

As was already mentioned, due to the specific ERP system requirements, we discarded the document/case centric paradigms. Although the business artifact-centric paradigm looks promising, after modeling sessions we concluded that at this stage of research we should postpone it and focus only on the activity-centric paradigm. It is because of multiple tools availability, both for process modeling and execution in the process/WF engines.

Most activity-centric tools assume the presence of Process Aware Information System (PAIS), which, especially in the context of legacy systems, might not be an option. As a result we need to extract information from the legacy systems that don't know anything about the business processes in which they are used. Only by interpreting directly extracted information we should be able to infer about the states which business process instances are currently in and possibly were in.

By taking into account the fact that for legacy ERP system integration all business information primarily should be stored in them, the restriction that a process engine therefore must store only references to actual data is put forward. The case when we store a copy of ERP system's data in the process engine complicates both the implementation and maintenance because of data synchronization issues. Let us call this the Leading System paradigm. In this case an ERP system and a process engine should take the role for a leading process instance in a stepwise manner – when the process instance needs to wait for an event that is related to the changes in the ERP system, the leading entity becomes the ERP system, but when the process definition requires execution of some service activity (that eventually might lead to the changes in the ERP system itself), then the leading entity becomes the process engine. In this Leading System scenario we propose to integrate process engines with the legacy (i.e., non-PAIS) ERP systems.

There is the well-known workflow reference model that was originally proposed back in 1995 [38]. Although it might seem outdated compared to today's situation, we believe it is not. That is because the workflow reference model was defined abstractly without referencing to particular implementation technologies [38 (Fig 6)]. In the context of legacy systems, the most crucial (and complex) then becomes Interface3 between Workflow API and Invoked Applications. It must ensure activation of the legacy system user interface and report back user activity outcomes to the workflow engine in order to advance process instances.

In situations, when WF engine is not able to communicate with the legacy systems directly (e.g., because of a proprietary/closed communication protocols), there can be used various software proxy agents, that translate communication between the WF engine interface and the legacy system's interface in both directions. If we assume the presence of such agents, then we can propose the following conceptual architecture for integration of the process engine with legacy ERP systems as shown in Figure 7 below.

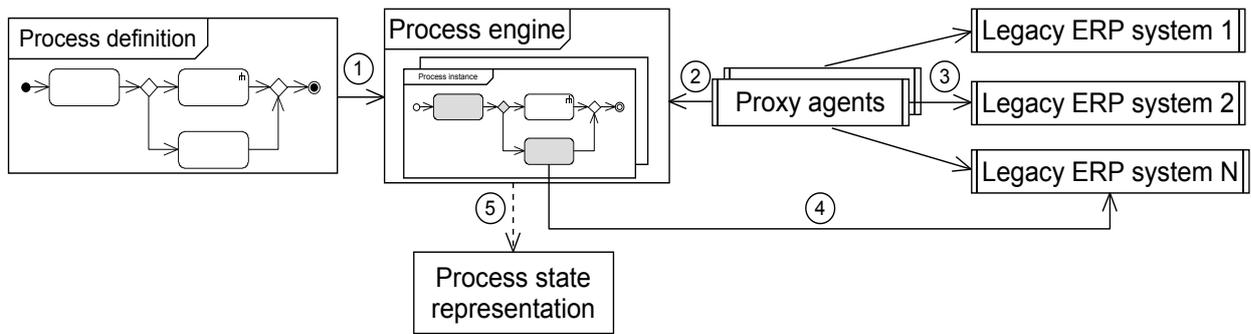


Figure 7. Conceptual architecture for process engine integration with multiple legacy ERP systems in the CBP setting

During the process analysis and definition within this architecture (Figure 7, left side) we propose to use various activity-centric process modeling tools in order to create process definitions that satisfy customer needs but at the same time are not overloaded with technical elements of notation required only for successful import and execution in the process engine.

In the process engine layer, successfully imported process definitions (link 1) are then used to create, advance and terminate process instances (Figure 7, the middle). The restriction on such processes is that no business data should be stored in the process engine itself – only references to the data that reside in the legacy ERP systems are allowed. Process instances are advanced either upon arrival of events from proxy agents (link 2) or based on the results of the service activity call to the ERP system (link 4).

A set of software proxy agents communicates with one or more ERP systems (Figure 7, the right side and link 3) in order to monitor predefined data state changes and to generate corresponding events for the process engine layer (link 2). In order to simplify agent definitions, we assume that a particular agent does not need to know anything about the business process it is related to. Only the queries on the business data should be defined and events in the case of match should be generated. Each agent should execute queries and monitor only a single business rule, with one or more events associated with each matched result set item. Such business rules are based on the facts about data in the legacy ERP systems, and usually are the cause for the state transition in the business process instances. We also assume that rules do not have dependencies (i.e., rule sequences) – we are interested only in the fact of a match against the predefined query. We call such rules a contract between data and process layers.

When, according to the rule, an agent detects the presence of a particular data state in an ERP system, it generates an event for the process engine. Depending on an event, the process engine reacts either by creating a new instance or advancing or terminating the existing instance(s) of the process. The additional restriction on query definitions and their execution schedules is that proxy agents must not degrade overall performance of legacy ERP systems.

If there are agents that interact with multiple heterogeneous legacy systems, then it is possible to support also cross-enterprise CBPs. Finally, we assume that the agent definition ensures that only one event is generated and sent to the process engine for a particular matched result set item. The main problem we foresee in this scenario is the possibility for an agent to miss a particular data state, therefore, the event will not be generated at all. We are also working on restrictions that must be placed on process definitions so that process instances do not halt but advance despite the missing event(s). For example, in the process definition this might be supported by activities with associated timeout elements.

Finally, legacy ERP systems are responsible mainly for presentation of the user interface and interaction with the business data. In this layer we do not assume the presence of business process knowledge according to which data are created, modified, or deleted. The only changes that could be made in the legacy systems is the provision of some kind of a dashboard (Figure 7,

bottom and link 5), so that users could see the process instances they are currently associated with (e.g., as process owners or just participants) and at what state each instance is. This should help them to identify active tasks that must be performed in the ERP systems.

This architecture has been tested for simple sequential processes that do not have branching or looping elements. Initial results are promising, however, further research are required in order to apply and test such an architecture for real life CBP scenarios. We foresee that the most suitable implementation would be such that the business process definitions are not too overloaded with technical elements, but at the same time it is possible to run them in the process engines, based only on the data change events from the legacy ERP systems.

6 Conclusions

In this paper we have discussed current collaborative cross-enterprise BPM issues in the context of ERP vendor participation in such processes. We focus on cases where several collaborating enterprises use the same ERP system. Small and medium businesses, which the ERP vendor targets, are not able to implement expensive WF solutions, as well as adaptation of legacy ERP systems for WF based execution usually is not possible without completely redesigning the architecture of the legacy ERP system. As a compromise, BP monitoring and user guidance during cross-enterprise CBP execution could be introduced. However, there are various technological issues to solve, e.g., state change detection in the process and data flows, access control, and determination and notification about subsequent tasks.

The review of the available CBP solutions revealed that each paradigm has its own challenges. Nevertheless, the ERP vendor can support cross-enterprise processes to a smaller or larger extent according to all four BPM management paradigms. Simplified enterprise model fragments behind this possible support were presented. The models were created on the basis of literature review. Analysis of the available techniques for managing cross-enterprise CBP let to conclude that for the business case discussed in this paper we should focus on the activity-centric and artifact-centric approaches. The former is mature and has broad tool support, but is imperative and less flexible. The latter might be more promising, but its practical application and tool support are not mature enough and have to be studied in more detail. Therefore we chose the activity-flow oriented approach and, by using Leading System paradigm, tested it on the business case for the ERP system vendor. Initial results are promising, however we will do further experimental development in order to apply this paradigm in real life cross-enterprise CBP scenarios.

Acknowledgment

The research has been conducted within the framework of European Regional Development Fund's project "Information and communication technologies competence center" Nr. KC/2.1.2.1.1/10/01/001 (Contract No. L-KC-11-0003) activity 1.2. "Technologies for managing services in workflows."

References

- [1] C. Liu, Q. Li, X. Zhao. "Challenges and opportunities in collaborative business process management: Overview of recent advances and introduction to the special issue," in *Information Systems Frontiers* 11(3), Springer US, 2009, pp. 201-209.
- [2] Q. Chen, M. Hsu. "Inter-enterprise collaborative business process management," in *Proc.17th International Conference on Data Engineering, IEEE, 2001, pp. 253-260. Available: <http://dx.doi.org/10.1109/ICDE.2001.914836>*

- [3] M. Pouly, C. Huber. "Collaborative networks: Human aspects and corresponding IT support," in 13th International Conference on Computer Supported Cooperative Work in Design, IEEE, 2009, pp. 576-581. Available: <http://dx.doi.org/10.1109/CSCWD.2009.4968121>
- [4] X. Feng, Y. Peng, H. Xie, Z. Yan. "Role-Based Learning Path Discovery for Collaborative Business Environment," in International Conference on Control Automation and Systems Engineering (CASE), IEEE, 2011, pp. 1-4. Available: <http://dx.doi.org/10.1109/ICCASE.2011.5997835>
- [5] Citrix GoToMeeting™, Available: <http://www.citrix.com/products/gotomeeting/>
- [6] Q. Li, M. Abel, J.-P. Barthes. "Sharing working experience: Using a model of Collaborative Traces," in 16th International Conference on Computer Supported Cooperative Work in Design, IEEE, 2012, pp. 221-227. Available: <http://dx.doi.org/10.1109/CSCWD.2012.6221822>
- [7] Workflow Management Coalition: Terminology & Glossary. WFMC-TC-1011, Issue 3.0, February 1999, Available: <http://www.wfmc.org/resource/>
- [8] Future Internet Technologies for MANufacturing industries, Available: <http://www.fitman-fi.eu/>
- [9] V. M. Hoang, H. H. Hoang. "An Ontological Approach for Dynamic Cross-Enterprise Collaboration," in 26th International Conference on Advanced Information Networking and Applications Workshops, IEEE, 2012, pp. 1355-1360. Available: <http://dx.doi.org/10.1109/WAINA.2012.128>
- [10] P. Walter, D. Werth, P. Loos. "Peer-to-Peer-Based Model-Management for Cross-Organizational Business," in 15th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, IEEE, 2006, pp. 255-260. Available: <http://dx.doi.org/10.1109/WETICE.2006.53>
- [11] I. Khriess, E. Levesque, G. Tremblay, A. Jacques. "Towards Adaptability Support in Collaborative Business Processes," in International MCETECH Conference on e-Technologies, IEEE, 2008, pp. 34-45. Available: <http://dx.doi.org/10.1109/MCETECH.2008.25>
- [12] C. P. de Laborda, C. Popfinger, S. Conrad. "Dynamic intra- and inter-enterprise collaboration using an enhanced multidatabase architecture," in Proc. 16th International Workshop on Database and Expert Systems Applications, IEEE, 2005, pp. 626-631. Available: <http://dx.doi.org/10.1109/DEXA.2005.79>
- [13] S. Eckartz, M. Daneva, R. Wieringa, J. van Hillegerberg. "Cross-organizational ERP management: how to create a successful business case?" in Proc. the ACM symposium on Applied Computing, USA: ACM, 2009, pp. 1599-1604. Available: <http://dx.doi.org/10.1145/1529282.1529641>
- [14] R. K. L. Ko, S. G. Lee, E.W. Lee, A. Jusuf. "Dynamic Collaborative Business Process Formulation via Ontologised Hierarchical Task Network (HTN) Planning," in IEEE International Conference on Web Services, IEEE, 2009, pp. 959-967. Available: <http://dx.doi.org/10.1109/ICWS.2009.77>
- [15] R. Thullner, S. Rozsnyai, J. Schiefer, H. Obweger, M. Süntinger. "Proactive Business Process Compliance Monitoring with Event-Based Systems," in 15th IEEE International Enterprise Distributed Object Computing Conference Workshops, IEEE, 2011, pp. 429-437. Available: <http://dx.doi.org/10.1109/EDOCW.2011.22>
- [16] D. Cohn, R. Hull. "Business Artifacts: A Data-centric Approach to Modeling Business Operations and Processes," IEEE Data Eng. Bull., vol. 32(3), 2009, pp. 3-9.
- [17] X.-Q. Zheng, M. Liu, J.-W. Yan. "Design of the executable business state-model process based on finite state machine," in 16th IEEE International Conference on Computer Supported Cooperative Work in Design, IEEE, 2012, pp. 207-213. Available: <http://dx.doi.org/10.1109/CSCWD.2012.6221820>
- [18] N. Schuster, C. Zirpins, S. Tai, S. Battle, N. Heuer. "A Service-Oriented Approach to Document-Centric Situational Collaboration Processes," in 18th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, IEEE, 2009, pp. 221-226. Available: <http://dx.doi.org/10.1109/WETICE.2009.23>
- [19] J. Dang, C. Toklu, K. Hampel, U. Enke. "Human Workflows via Document-Driven Process Choreography," in International MCETECH Conference on e-Technologies, IEEE, 2008, pp. 25-33. Available: <http://dx.doi.org/10.1109/MCETECH.2008.15>
- [20] L. Zhao, J. Xing, L. Meng. "The Research and Realization of a New Workflow Model with Step-Task Two Layers Based on Document," in IEEE Asia-Pacific Conference on Services Computing, IEEE, 2006, pp. 285-292. Available: <http://dx.doi.org/10.1109/APSCC.2006.103>
- [21] W. M. P. van der Aalst, M. Weske, D. Grünbauer. "Case handling: a new paradigm for business process support," Data Knowl. Eng., vol. 53(2), 2005, 129-162. Available: <http://dx.doi.org/10.1016/j.datak.2004.07.003>
- [22] Case Management Model and Notation (CMMN) version 1.0, May 2013, Object Management Group, Available: <http://www.omg.org/spec/CMMN/1.0/>
- [23] A. Nigam, N.S. Caswell. "Business artifacts: An approach to operational specification," IBM System Journal, vol. 42(3), 2003, pp. 428-445. Available: <http://dx.doi.org/10.1147/sj.423.0428>
- [24] F. (T.) Heath, D. Boaz, M. Gupta, R. Vaculín, Y. Sun, R. Hull, L. Limonad. "Barcelona: A Design and Runtime Environment for Declarative Artifact-Centric BPM," in Service-Oriented Computing. LNCS, S. Basu, C. Pautasso, L. Zhang, X. Fu. (Eds.), vol. 8274, Heidelberg: Springer, 2013, pp. 705-709.
- [25] A. Russo. "Data-aware and Artifact-centric Business Process Management," in Seminars in Software and Services for the Information Society 2013-2014 (slides). Sapienza Università di Roma, 2014. Available: <http://www.dis.uniroma1.it/~arusso/teaching.html>

- [26] L. Limonad, D. Boaz, R. Hull, R. Vaculin, F. Heath. "A Generic Business Artifacts Based Authorization Framework for Cross-Enterprise Collaboration," in Annual SRII Global Conference, IEEE, 2012, pp. 70-79. Available: <http://dx.doi.org/10.1109/SRII.2012.19>
- [27] K. Ngamakeur, S. Yongchareon, C. Liu. "A Framework for Realizing Artifact-Centric Business Processes in Service-Oriented Architecture," in Database Systems for Advanced Applications, S.-G. Lee, Z. Peng, X. Zhou, Y.-S. Moon, R. Unland, J. Yoo. Eds. LNCS, vol. 7238, Berlin Heidelberg: Springer, 2012, pp. 63-78. Available: http://dx.doi.org/10.1007/978-3-642-29038-1_7
- [28] D. Cohn, P. Dhoolia, F. Heath III, F. Pinel, J. Vergo. "Siena: From PowerPoint to Web App in 5 Minutes," in International Conference on Service Oriented Computing, A. Bouguettaya, I. Krueger, T. Margaria. (Eds.), LNCS, vol. 5364, Heidelberg: Springer, 2008, pp. 722-723. Available: http://dx.doi.org/10.1007/978-3-540-89652-4_63
- [29] C. Di Ciccio, A. Marrella, A. Russo. "Knowledge-intensive Processes: An Overview of Contemporary Approaches," in Proceedings of the 1st International Workshop on Knowledge-intensive Business Processes, CEUR Workshop Proceedings, 2012, pp. 33-47. Available: http://ceur-ws.org/Vol-861/KiBP2012_paper_2.pdf
- [30] Process Mining - Research, Tools, Application. Available: <http://www.processmining.org/>
- [31] F. Mulazzani, B. Russo, G. Succi. "ERP Systems Development: Enhancing Organization's Strategic Control through Monitoring Agents," in 8th IEEE/ACIS International Conference on Computer and Information Science, IEEE, 2009, pp. 535-542. Available: <http://dx.doi.org/10.1109/ICIS.2009.143>
- [32] E. Damiani, F. Mulazzani, B. Russo, G. Succi. "SAF: Strategic Alignment Framework for Monitoring Organizations," in Business Information Systems. W. Abramowicz, D. Fensel, Eds. LNBIP, vol. 7, Berlin Heidelberg: Springer, 2008, pp. 213-226. Available: http://dx.doi.org/10.1007/978-3-540-79396-0_19
- [33] Q. Chen, M. Hsu, V. O. Mehta. "Dynamic Interface between BPSS Conversation Management and Local Business Management," US Patent 8301573, 2010.
- [34] Perceptive Software. Available: <http://www.perceptivesoftware.com/products/perceptive-process>
- [35] I. Vanderfeesten, H. Reijers, W. M. P. van der Aalst. "Case Handling Systems as Product Based Workflow Design Support," in: J. Filipe, J. Cordeiro, J. Cardoso, Eds. Enterprise Information Systems, LNBIP, vol. 12, Berlin Heidelberg: Springer, 2008, pp. 187-198. Available: http://dx.doi.org/10.1007/978-3-540-88710-2_15
- [36] C. W. Guenther, M. Reichert, W. M. P. van der Aalst. "Supporting Flexible Processes with Adaptive Workflow and Case Handling," in Proceedings 3rd IEEE Workshop on Agile Cooperative Process-aware Information Systems, IEEE, 2008. Available: <http://www.wis.win.tue.nl/~wvdaalst/publications/p494.pdf>
- [37] Artifact-Centric Service Interoperation (ASCI) project public deliverables. Available: <http://www.acsi-project.eu/pd.html>
- [38] Workflow Management Coalition. Reference model - The Workflow Reference Model. Technical Report WFMC-TC-1003, version 1.1, 1995. Available: <http://www.wfmc.org/standards/docs/tc003v11.pdf>
- [39] Ē. Asjina, G. Alksnis. "Survey on Information Monitoring and Control in Cross-enterprise Collaborative Business Processes," in Proc. the 7th International Workshop on Information Logistics and Knowledge Supply. CEUR Workshop Proceedings, 2014. Available: <http://ceur-ws.org/Vol-1246/paper-01.pdf>