Cooperative Process Coordination

Julia Bitcheva, Olivier Perrin, Claude Godart

COOPERATIVE PROCESS COORDINATION

Julia Bitcheva*
France Télécom R&D
2, av. Pierre Marzin
22307 Lannion, France
bitcheva@loria.fr

Olivier Perrin, Claude Godart
LORIA
Campus Scientifique, BP 239
54506 Vandœuvre-lès-Nancy, France
{operrin, godart}@loria.fr

ABSTRACT

Virtual enterprise organization regroups partners distributed in space, time and organization, in order to achieve a common goal. Thus, their business process realization needs the coordination of distributed partners' interactions. This paper presents the "synchronization point" concept, which provides support for the cooperative process coordination. It provides partners pertinent information about their work progress while maintaining the privacy of information. Moreover, it supports both the long-time transactions and the dynamic process definition as requested for cooperative process management. Finally, its data repository and action manager helps human interactions in cross-organizational applications.

KEY WORDS

cooperation process, collaborative activities, coordination, virtual enterprises

1. Introduction

In the very competitive and expanding global marketplace, different organizations need to form alliances to achieve a common business goal. Business pressures (development costs, time-based competition...) are placing increased emphasis on how organizations operate and interoperate with other enterprises. The B2B interactions should take place simply; the organizations should work more directly with their suppliers and customers, to respond more quickly to changes. The rapid growth of web technologies is beginning to transform traditional inter-enterprise business models and allows virtual enterprise creation.

To enable organizations to adapt to this new business environment, a middleware is required to provide dynamic and flexible integration between partners in the value chain. Although new technologies will be needed to enable such integration, they will have to work seamlessly with existing inter-enterprise business processes. In this paper, we propose a concept that tries to answer these features.

In section 2, we give a definition of virtual enterprises and awaited features of cross-organizational processes. Our proposal, the synchronization point concept, is presented extensively in the next sections. Section 3 presents the process services. The SP functionalities are presented in section 4, while its components are defined in section 5, and its management system in section 6. Section 7 gives a short comparison to related work, and we conclude in section 8.

2. Problem statement

2.1. Virtual enterprise

A virtual enterprise (VE) is an organization that allows enterprises to create a partnership for a specific project. To work on this project, the partners' enterprises have to share competences and resources, and make their members work together. But these members belong to different physical organizations and they could be in different places, and even in different time zones.

Contrary to classical organizational structure based on long-standing business partnerships, VE organizational models have to be more dynamic, and loosely coupled with the specific business partner and the partner's physical location. Another difference is that a VE achieves its objectives by defining and implementing processes, but these processes are distributed across several organizations [1]. Each partner implements a subset of activities of the overall process and the same activity can be implemented conjointly by several enterprises. This means that we have to coordinate and synchronize all activities. Another requirement is that each member's contribution is specified by a contract. The inter-enterprise contract definition specifies the deliverables, the person in charge, and the contract terms (deadline, quality, guarantees, etc.).

To summarize, the virtual enterprise realization needs to take into account the resources, the organizational models, the contracts between participants, and the process models. Our virtual enterprise model is described in [2]. In this paper, we focus on the collaborative process model.

2.2. Motivating example

Let's take, as an example, the activity representing the design of a new PC motherboard integrating different functionalities: network, audio, and video. Each functionality corresponds to an integrated component on the motherboard and each component will be designed by a specific designer. The whole design activity requires the cooperation of each designer: the motherboard designer, the network designer, the audio designer, and the video designer.

The motherboard design activity can be divided into several phases. Firstly, the project manager chooses the partners. Then, each of them proceeds to the electric layout design where they specify which electronic components

will be used and their theoretical connections. The next phase consists of the component layout specification. In fact, the distance between components is a primal concern (mainly in high frequency) for their communication performance. Finally, all ports (serial, parallel, keyboard, sound, video, network, etc.) have to take place on the same motherboard site. Consequently, the component layout specification is a major and very complex phase, requiring all partners' cooperation. The component layout specification proceeds in three steps: electronic component placement, bus placement, and port placement. Each of these steps needs all partners' participation. Every designer makes a blueprint proposal for his relevant part. Since those parts must be assembled in the same motherboard, the blueprint acceptance depends on the whole set of proposals. An incompatible proposal may provoke work cancellation by some partners, and the revision of previous steps.

As an example, for the realization of the first step, each designer will propose one of its off-the-shell solutions, but it may be impossible to assemble the shape of those propositions into the restricted motherboard space. Thus, the partners will have to collaborate on shape modifications. In the next step, partners should agree on the placement of the buses, so that the different parts can be linked and communicative. This may require deep modification in these parts, calling for a new first step iteration. Similar problems can arise in the last step. This is why the partners' collaboration is essential for the whole design activity. This is necessary for realization of the common goal, but also for cost optimization and for a better work performance.

2.3. Cooperative processes

A process definition consists of a network of activities and their relationships [3,4]. In a cooperation process, partners from different enterprises realize both atomic and composite (sub-processes) activities. In fact, a cooperative process definition is similar to traditional workflows in the sense that it describes the flow of the composed process. However, if they could be an acceptable solution for sequential activities, workflow systems are not adequate for the coordination of collaboration activities. As a matter of fact, the major characteristic of collaboration activities is that they are realized by parallel flows of execution, modeling the activity of each contributor. When using a traditional workflow system to model such a collaboration activity, time efficiency could be impaired: partners may have to wait for termination of all activities before being able to estimate result compatibility and before making a decision. For faster reaction to the changes, collaboration partners have to exchange intermediate results, before the end of their contributions. Some 'flexible' workflow systems can manage the intermediate results' exchange, if all partners' exchanges are anticipated and modeled before the process execution. But, human collaboration activities are too unpredictable and could not be totally anticipated. As the considered collaborations are not fully automated, and the human reaction is uncertain, a VE requires the ability for dynamic process definition change. Another consequence of the human participation is the relatively long time of their reactions. The use of traditional ACID transactions is inappropriate for the long-running activities because of resource locking. Instead, such activities require *long-time transactions*: isolation and durability levels should be relaxed and intermediate results can be released before transaction's end. The consistent state of the system is guaranteed by compensating activities (typically application specific). Once again, traditional workflow systems do not support long-time transactions.

On the other hand, groupware tools provide implicit coordination means. They support multiple partners' parallel work by managing divergence through version storage, but the global version integration is delayed until the last phase. Then the problem is that divergence could then be so high that the global integration is impossible. To manage this problem, they provide group awareness means, which allow the participants' auto-coordination [5].

In either case, the problem remains "how to coordinate collaboration activities?" The strong interdependency of partners' parallel work requires intermediate results exchange and process-progress synchronization. Our approach tries to combine the advantages of workflow and groupware tools. By adding the flexibility of group awareness implicit coordination to the explicit coordination of workflow systems, we provide a tool that allows partners to coordinate themselves during the work progress. Moreover, we support both the long-time transactions and the dynamic process changes.

In the next sections, we describe our proposal for supporting cooperation process modeling and activities' coordination.

3. Process services

To take part in cooperation projects, an enterprise should declare what it can offer to partners, thanks to the description of offered products and services. On the other hand, external processes will need event feedback, in order to control their own work progress.

Since all these outcomes are the result of an internal business process, we named the description of the offers, a *process service*. Besides the outcomes (products and events) description, a *process service* definition contains the conditions surrounding the offers (inputs, guarantees, etc.), as well as the provider's information (access information, communications modes, etc.). Our process service definition is compliant to the WSDL standard [6], but includes information dedicated to the synchronization, the transactional management and the retrieval of the service's state [7].

Business processes are part of the enterprise strategic core, as they represent the organization know-how and contain a lot of proprietary information. Partners should not have direct access to this information. So, our process service is an abstract representation of the enterprise business process. The *process service model* is a layered model,

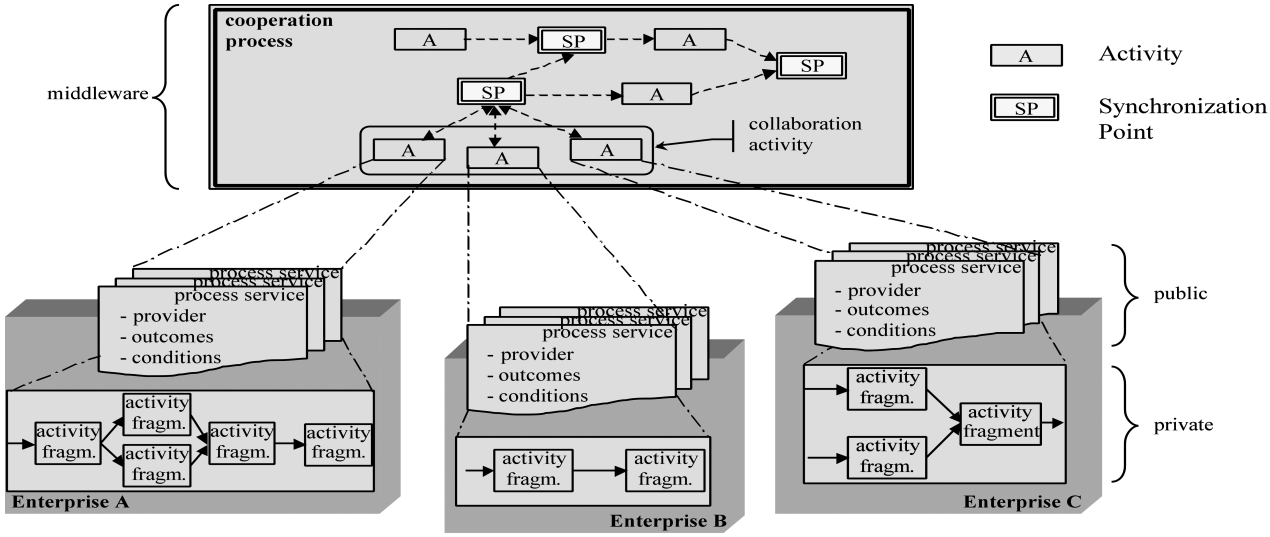

Figure 1. Inter-enterprise cooperation process

where the process service layer is a public abstract definition of the outcomes the enterprise is able to deliver, and the business process layer is the internal process flow in the enterprise. The model clearly separates the enterprise offer from its implementation in order to respect the enterprises privacy needs: no direct access to internal processes and no restriction for their implementation.

Using this abstraction level, the cooperation process realization boils down to the problem of process services' composition and integration. The interactions between process services have to be coordinated.

4. Synchronization Points

As described above, cooperation processes need a tool to allow partners' work coordination. In this aim, we propose the synchronization point concept.

The Synchronization Point (SP) is, as its name suggests, a process component where two or several partners' activities will be coordinated. This requires the definition of coordination (i.e. synchronization) criteria at the beginning of each cooperation. This definition should be done in a flexible way, in order to allow the synchronization point's revision (addition, suppression or modification) as the work progresses.

Figure 1 provides an overview of the proposed approach. A process definition consists of a network of activities and their relationships. In an inter-enterprise cooperation process, an activity realization can be either done by a single enterprise or by partners belonging to different enterprises (collaboration activity). The SP aims to coordinate all activities. It manages the inter-enterprise exchange and controls the contract terms: deadlines, outcomes' guarantees, etc. Moreover, in the case of collaboration activities, the SP also provides tools allowing partners' synchronization for the realization of the common goal. Thus, a cooperative process definition becomes a set of activities and synchronization points that coordinate partners' work.

To achieve this coordination, the SP implements several functions described in the following sections.

Control. To be able to synchronize the partners, the SP needs information on their work progress. To respect enterprise privacy needs, it does not ask information on their internal processes, but uses only the outcomes that are proposed by the corresponding process services. The implementation of this function consists of outcome-checking. First, the SP associates to each activity a list of expected outcomes, corresponding to the inter-enterprise contract. This list contains the description of intermediate and final results, and some predefined events (for example: end of specific task, document reception, etc.) Then, the SP compares and evaluates the activity outcomes regarding a set of predefined criteria - time schedule, quality criteria, etc.

Act. The SP actions depend on a set of rules predefined by the partners. These *action rules* map event notifications to a set of actions and to a person in charge of its realization. The events can be either the result of the normal execution of the process, or an exception. For example, we can describe the actions to be taken if one partner does not respond. The people in charge are also defined for all non-expected events. The realization of the action rules is described in a following section.

Plan. This SP function supports dynamic process changes. It allows the SP revision (e.g. deadline change or action rule addition), as well as the process adjustment - with the help of the SPMS described below - by addition or suppression of SP and activities. A list of available changes is proposed for all running SPs. Some SP revisions require the current SP execution suspension.

Recovery. Each finishing activity may prescribe a compensating activity in order to revert any changes made permanent by the first one. Compensating activities are used to recover from activities that cannot be rolled back

(e.g. sending an e-mail or shipping a package). The SP invokes the compensating activities for all the changes carried out.

Communication. Communication is one of the essential functions for coordination achievement. It allows participants to exchange information concerning their results or the firing of some events. The SP supports different communication policies (request/response, solicit response, one-way, notification) and provides structures to information storing and secure sharing.

Awareness. Being aware of the partners' work is a good way to help coordination of people in the realization of a common objective. To achieve this, the SP notifies partners when a change is available. In order to provide only pertinent information and avoid overload, SP allows each participant to create a subscribed events list, so that he/she will be notified only for a kind of event. For example, a participant can subscribe to a shared document state event. Of course, SP allows also the possibility to check directly the current work progress.

At the beginning of each collaboration, the partners agree on a set of rules for their common work coordination. These rules (action rules) are stored and managed by a SP, and could be changed on the fly if needed. SP executes them automatically if possible, but leaves the responsibility of main actions and decisions to human partners. Humans are also in charge of all non-expected situations.

To preserve the independence of partners, the SP considers their contribution as parts of autonomous work, that we call *activity fragment*s. The fragment outcomes - (intermediate) results or process events - are sent to the synchronization point and can be shared between partners. This sharing is made possible thanks to a shared data repository, managed by the SP. The intermediate results are used by the SP to provide awareness information to partners, and the final results to guarantee the output's conformity to the inter-enterprise contract.

Providing the intermediate results' exchange and current changes recovery, the SP supports the long-time transactions that are essential for the realization of the collaboration activities. Moreover, the use of the *coo* operator [8] allows for managing the access to resources (e.g. intermediate results) shared by two collaborative partners' processes. This operator defined by the *coo*-serializability correctness criterion helps to ensure the correct execution (wrt transactional meaning) of two partners' processes.

5. Synchronization Points Components

The synchronization point includes three principal components: partners identification module, data repository and action manager.

5.1. Identification module

This module contains all SP participants' relevant information: roles, the corresponding characteristic information (identification and access information), and, if available, a specific information like awareness preferences.

5.2. Data repository

This component manages the synchronization point's information storage. It consists of two data structures, according to the data's type:

Version graph. The SP uses a DAG (Directed Acyclic Graph) of versions to store all activity's results (figure 2). It allows the secure intermediate or final results sharing. The files are available for all synchronization point's participants, according to their access rights. Each participant, when publishing a result, has to choose a label between "early access", "draft", "release" or "validated", which will mark the result status. He can also add, as a comment, a description of the changes carried out. If there is a control (e.g. quality) criteria, they will also be stored in the version graph.

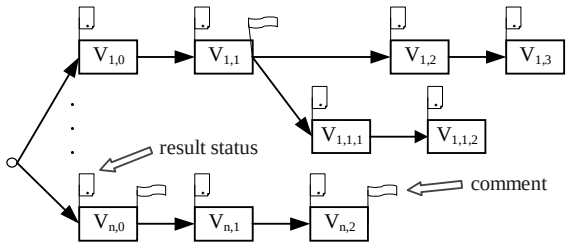

Figure 2. Version graph

Events store. This structure contains received event's identifiers and parameters.

5.3. Action manager

This component manages all information relating to the SP's actions. It contains the descriptions of expected information exchanges, the predefined SP's action rules and, if available, the user-defined restrictions.

5.3.1. Information exchanges

This component contains the expected information inputs and outputs. Its description has six parts:

The first part is a flag mentioning if it is an input or output. The second one specifies the exchanged information type – event or result. The third part is the specification:

- for events – identifier, parameters and access mode (automatic or function-call).
- for results – file identifier and type,

The fourth part defines the actor:

- for inputs – the sender,
- for outputs - the person in charge.

The fifth part describes the time of exchange:

- for inputs – the reception deadline,
- for outputs – time of sending: immediately, date/hour or condition.

The last part is optional and contains the textual description of the exchange meaning.

5.3.2. Action rules

This component contains a set of predefined actions. It specifies the SP's *action rules* in the form of tuples *<event, condition, action, in-charge>* [9]. These rules are used to manage events checking, exception management, awareness actions, and, if available, activity recovery actions.

Event. The SP manages both primitive events and composite events.

Primitive events are atomic, low level events that can be SP events, temporal events, or external events. The *synchronization point's events* correspond to SP internal operations, such as data storing, sending information, or SP state changing. A *temporal event* can be specified with an absolute time value. It could also be defined as a delay relative to a reference point, which may be any event apparition. *External events* correspond to partners' process progress. They are produced by external applications (i.e. outside the SP) and are only managed by the SP as primitive events.

Composite events allow SPs to detect the combinations of different events as a single event. A composite event is composed of two or more primitive or composite events connected through event operators. The event operators can be both logical operators (OR, AND, NOT) and specific operators like ANY (a subset) or SEQ (specified sequence) [10].

Condition. This part is optional. It specifies a set of conditions that has to be true to proceed to the execution of the action.

Action. An action can be specified on primitive events as well as on composite events and there can be several actions specified on the same event. An action can be any executable program.

In-charge. This part defines the person in charge of the action execution. If there are several actions, a different person can be specified for each one.

5.3.3. Rules management

For the event detection the system use notification graph as shown in figure 3. Primitive events are represented as leaf nodes (*e1*, *e2*, *e3*). The non-leaf nodes are composition operators that represent the corresponding composite events.

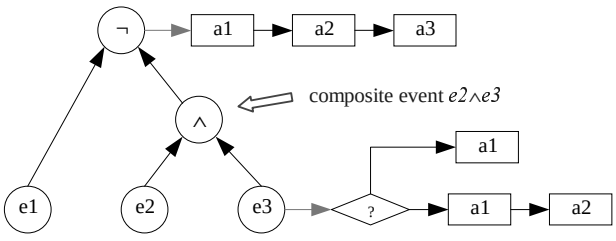

Figure 3. Event notification graph

When an event occurs, the corresponding nodes in the graph are activated and the notification propagates upwards in the graph. A couple condition-action can be associated with both primitive and composite events. A person in charge is defined for each action.

The activation of a node triggers the evaluation of the associated condition. If the condition applies than the actions are sequentially executed.

This component manages also the data about information exchanges. They can be considered as a special type of action rules that could only be triggered by temporal events.

6. Synchronization Points Management

The Synchronization Point Management System (SPMS), not described in this article due to lack of space, manages SP instances' life cycle. It is responsible for the new SP creation and manages all SP state transitions.

One of the SPMS main advantages is the dynamic process changes management, allowing the SP creation, modification or suppression during the process execution. The SPMS contains the system units for the main types of possible changes and it manages the set of available changes in order to assure the process definition correctness. Furthermore, the SPMS controls the past executions' history management. All SP information is saved before a SP state transition. As a result, we are able to start a new instance of terminated SP, taking into account the reasons of the abort. All this allows for a successful cooperation process reengineering.

7. Related Work

Current work on process management is not directly applicable in the virtual enterprise domain. In fact, there is no common shared middleware that could be used by several enterprises or that could fit in the needs for spreading across enterprise boundaries. Current propositions lead to tightly coupled enterprises, not only at an architectural level, but also at the process level. Moreover, transaction models and coordination needs are not accurate. For instance, access to shared resources is very important, and locking these resources is not desirable as one organization could lose its autonomy. Then, recovery operations could not be under the responsibility of only one enterprise. To summarize, in virtual enterprises, there are two contradictory needs that are the autonomy of partners and the need to get some information about processes held by the others. We will now present some works that are trying to answer this challenge.

VanDerAalst [11] proposes a model that is compliant to the WfMC model. It is a global model that could be split into different parts, but there is no communication between the partners. The execution of this model is under the responsibility of centralized architecture. However, the use of Petri nets makes possible the verification of the process consistency.

In Weske [12], there are some ideas to resolve flexibility in workflow management systems but no proposals for multi-

enterprise processes. One interesting idea is the definition of a meta-schema for workflows and the use of graphs for defining workflows.

The work of Casati [13] describes data exchange and process interoperability, but in a B2B context, where exchanges are limited to peer-to-peer conversations. The concept of traces is very interesting and similar to our proposition.

Georgakopoulos [14] presents in CMI the concept of window of opportunity that allows for conciliating prescribed activities inherited from WfMS and optional activities inherited from groupware applications. Our synchronization point concept helps to provide the same kind of feature.

The CrossFlow [15] project aims at providing high-level support for workflows in dynamically formed virtual organizations through service abstraction, cooperation support, and contract management. They use a proprietary format for service abstraction, while our proposition is WSDL compliant. Moreover, they support cooperation by using a traditional workflow system, and we have already shown the problems of this approach. Finally, we added data filtering capabilities in our contract management.

Then, propositions from HP (e-speak), OASIS (ebXML) and Microsoft (.NET) answer some issues but fail to provide solutions for long term transactions, collaboration phases, or binding to internal processes.

8. Conclusions and Future Work

This paper presents a coordination tool that provides a flexible solution for cooperation process modeling and execution. We assume that each enterprise describes their offers as an abstract process service definition. Thus arises the problem of different enterprises' services integration and composition. Furthermore, when several partners have to realize a common objective, they need coordination means.

This is the role of the synchronization points. When different partners have to work together, they only need to define the cooperation rules (corresponding to contract specification). Then, the SP coordinates their progress and provides participants pertinent information of the work evolution. All information is adapted to partners' specific needs.

Our choices of event detection and action management result in cooperation rules that respect each partner's autonomy. Moreover, by including data and control flow management functions in the SP, we allow a flexible process definition - we are able to adjust the cooperation process definition by updating the SP during the work progresses.

Our future work aims to *analyze* function realization that will analyze the history of past process executions in order to help the decision-making. The aim is, using the knowledge of the current situation and of past execution history, to detect the reason for the problem and to suggest solutions to the person in charge of the decision. Using past experience knowledge, we can use data-mining methods to detect the reasons for problem occurrence, and to provide helpful information for decision-making [16].

References

[1] H. Schuster, D. Georgakopoulos, A. Cichocki, and D. Baker, Modeling and Composing Service-Based and Reference Process-Based Multi-enterprise Processes, *CAiSE 2000*, LNCS 1789, 2000.

[2] J. Bitcheva, Virtual Enterprise Meta Model, Technical Report, Vision e-company Research Program, France Télécom R&D, April 2002.

[3] WfMC, Interface 1: Process Definition Interchange, Process Model, WfMC TC-1016-P, Version 1.1, October 1999.

[4] WfMC, Terminology & Glossary, WfMC-TC-1011, Version 3.0, February 1999.

[5] P. Dourish, and V. Bellotti, Awareness and Coordination in Shared Workspaces, *Proceedings of the ACM Conference on Computer-Supported Cooperative Work*, ACM Press: Toronto, Ontario,1992, 107-114

[6] Web Services Description Language (WSDL) 1.1, W3C, March 2001.

[7] J. Bitcheva, O. Perrin, C. Godart, Cross-Organizational Processes Coordination, Technical Report A02-R-046 LORIA, Nancy, May 2002.

[8] C. Godart, O. Perrin, and H. Skaf, COO: a workflow operator to improve cooperation modeling in virtual enterprises, *9th IEEE RIDE, International Workshop on Research Issues in Data Engineering*, Sydney, March 1999.

[9] J. Widom and S. Ceri, Active Database Systems, *Triggers and rules for advanced database processing* (Morgan Kaufmann Publishers, Inc., San Francisco, CA, 1996).

[10] S. Chakravarthy, V. Krishnaprasad, E. Anwar, and S.-K. Kim, Composite Events for Active Databases: Semantics, Contexts and Detection, *Proceedings of the 20th VLDB Conference*, Santiago, Chile, 1994.

[11] W. Van Der Aalst, and M. Weske, The P2P Approach to Interorganizational Workflows, *CAiSE 2001*, LNCS 2068, 2001.

[12] M. Weske, Formal Foundation and Conceptual Design of Dynamic Adaptations in a Workflow Management System, *Proceedings of the 34th Hawaii International Conference on System Sciences*, Ð 2001.

[13] F. Casati, M. Sayal, U. Dayal, and M.C. Shan, Integrating Workflow Management Systems with Business-to-Business Interaction Standards, *ICDE 2002*, February 2002.

[14] D. Georgakopoulos, Collaboration Management Infrastructure (CMI): Advancements in Coordination, Awareness, and Integration, November 2001.

[15] P. Grefen, K. Aberer, H. Ludwig, Y. Hoffner, CrossFlow: Cross-Organizational Workflow Management for Service Outsourcing in Dynamic Virtual Enterprises, *IEEE Data Engineering Bulletin 24* (1), 2001, 52-57.

[16] D. Grigori, F. Casati, U. Dayal, M.C. Shan, Improving Business Process Quality through Exception Understanding, Prediction, and Preventing, *VLDB*, Roma, Italy, 2001, pp. 10.