Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Data & Knowledge Engineering 56 (2006) 139–173

DATA & KNOWLEDGE ENGINEERING

www.elsevier.com/locate/datak

The view-based approach to dynamic inter-organizational workflow cooperation

Issam Chebbi [a], Schahram Dustdar [b,*], Samir Tata [a]

[a] Department of Computer Science, GET/INT (Institut National des Télécommunications), 9, Rue Charles Fourier, 91011 Evry, France

[b] Institute of Information Systems, Distributed Systems Group, Vienna University of Technology, Argentinierstrasse 8/184-1, A-1040 Wien, Austria

Received 8 March 2005; received in revised form 8 March 2005; accepted 24 March 2005
Available online 25 April 2005

Abstract

This paper presents a novel approach to inter-organizational workflow cooperation. Our goal is to provide support for organizations which are involved in a shared but not pre-modeled cooperative workflow across organizational boundaries. Our approach allows for partial visibility of workflows and their resources, thus providing powerful ways for inter-organizational workflow configuration. Varying degrees of visibility of workflows enable organizations to retain required levels of privacy and security of internal workflows. Our presented view concept provides a high degree of flexibility for participating organizations, since internal structures of collaborative workflows may be adapted without changes in the inter-organizational workflows. Furthermore, we provide workflow participants with the freedom to change their workflows without changing their roles in the cooperation. This increases flexibility and is an important step to increase efficiency as well as reduction in costs for inter-organizational workflows. The presented approach is inspired by the Service-oriented Architecture (SOA). Accordingly, our approach consists of three steps: workflow advertisement, workflow interconnection, and workflow cooperation.

Keywords: Inter-organizational workflow; Service-oriented Architecture; Workflow views

[*] Corresponding author. Tel.: +43 1 58801 18414; fax: +43 1 58801 18491.
E-mail address: dustdar@infosys.tuwien.ac.at (S. Dustdar).

0169-023X/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.datak.2005.03.008

1. Introduction

In context of globalization, a high competitive pressure characterizes the general situation on businesses. Competition is a dilemma many organizations face every day. It can lead to intensive re-structuring of organizational structures and processes to make production and services more efficient and less expensive. Additionally, new forms of inter-organizational collaboration between organizations may emerge. In this case organizations especially Small and Medium sized Enterprises (SMEs), cooperate to fulfill conditions of complex, often concurrent projects.

Parallel to this evolution, organizations are increasingly utilizing process-aware information systems to perform their workflows in an automated way. Based on such information systems, organizations focus on their core competencies and access other competencies through cooperation, moving towards a new form of network known as virtual organization.

There is still no agreed-upon definition of virtual organizations. Broadly speaking, a virtual organization is often defined as a temporary organization formed from strategic alliances or partnerships ("real organizations") that can be dissolved when the common business or the common project is finished. For Davidow and Malone [1], the word "virtual" comes from the idea of "potential" meaning "excellent", "high quality" or "advanced". A virtual organization is considered as the modern organizational form, and hence, as being the more advanced and the most efficient one [2]. Byrne defines a virtual organization as a temporary network of independent companies, suppliers, customers, and even rivals, linked by information technology to share costs, skills and access to markets. It will have neither central offices nor organizational charts, nor hierarchies, and no vertical integration [3].

As for us, we define a virtual organization as a set of partners ("real organizations") distributed in time and in space sharing resources and competencies (similar or dissimilar) and cooperating to reach some shared objectives using information technologies. Thus, partners with complementary competencies and knowledge can be gathered to carry out projects, which are not within the range of only one organization: cooperation allows each partner to benefit from knowledge of the other partners in the virtual organization. With this intention, partner workflows are not carried out in an isolated manner, but interact during their execution, while sharing data in a coordinated way [e.g., 28]. Coordination brings a synergy that contributes to the improvement of each partner work performances.

The rest of this paper is organized as follows. Section 2 identifies the requirements our contribution aims to meet. Section 3 presents the existing approaches for inter-organizational workflows. Section 4 proposes a three steps approach for inter-organizational workflow cooperation: workflow identification and advertisement, workflow interconnection, and workflow cooperation. All steps are discussed, and the second step is presented in more detail in Section 5. Section 6 provides a brief overview on the cooperation prototype platform we are developing. Section 7 concludes and presents our future work.

2. Requirements for inter-organizational workflows

2.1. Flexibility support

Cooperation between partners within a virtual organization is established according to needs for businesses and their competencies and roles. This leads to a dynamic character of cooperations. Indeed, the set of partners is in constant evolution. It is possible for organizations composing the virtual organization to be geographically distributed. It is also possible for organizations to join and leave the virtual organization as its state changes over time. That is, the composition of the virtual organization may be dynamic. Furthermore, its shared net behavior implies that there is some dynamic structure to the interactions between the organizations composing the virtual organization.

On the one hand, interactions can be relevant and constructive if they complement each other and guide the work toward the objective of the virtual organization. On the other hand, they can cause adverse effects if they are not well coordinated or if participants do not follow their roles in the cooperation. An action's effect can unintentionally remove the effect of another one. The effective result of the cooperation and the desired objective are then likely to diverge.

Since there is some dynamic structure to the interactions between the organizations composing the virtual organization, we think that interactions between workflows in virtual organizations cannot be specified in advance. In [12] the author states that "there are numerous situations where the organizations participating in a shared workflow processes feel the need to specify the coordination structure explicitly". In [29] the author mentions that "in many cases, where the coordination structure and the interaction between the business partners are not specified explicitly, this is not a realistic assumption". Nevertheless, we think that since interactions are dynamic their specification is difficult (if not impossible). Our intention and contribution of this paper is to describe (without explicit specification) a set of accepted interaction scenarios rather than one (which is the case of existing approaches since they use a workflow to specify interactions between workflows, see [15], for example).

2.2. Privacy respect principle

On one hand, cooperation needs a certain degree of workflow inter-visibility in order to perform interactions and data exchange. On the other hand, cooperation may be employed as a cover for organizations to internalize the know-how of their partners. The question here is how to best preserve the know-how of each partner and capitalize on the accumulated experience and knowledge to allow cooperation and to improve productivity.

In order to preserve privacy and autonomy of workflow of participants, we must reduce workflow inter-visibility to be as little as the cooperations need.

2.3. Established workflow preservation

For enabling cooperative organizations to integrate their disparate workflows it is necessary to allow them to use established workflows.

When planning projects, it is important to note that any changes to established workflows (even clear improvements) will cost money and time. Therefore, if organizations are to achieve the efficiencies and reduction in costs that the cooperation promises especially if they are SMEs, approaches for workflow cooperation must fully integrate pre-established workflows.

3. Related work

The last years have have witnessed the development of many inter-organizational workflow approaches, allowing organizations with complementary skills to cooperate and carry out works that are not within the range of only one organization. Despite their existence, these proposals are facing many problems vis-à-vis inter-organizational workflows requirements, in particular flexibility, privacy and workflow preservation respect. In fact, many existing approaches are not very flexible and interactions are specified in advance. Moreover, almost all solutions currently developed for the management of cooperation in the virtual organizations suppose that the homogeneity of the partners in terms of data structures, business logic, for example, which is restrictive and limits the cooperation. Besides, although some solutions provide means to preserve privacy and established workflows, inter-visibility is either very tiny and the collaborating partners act as black boxes or very open. In both cases, it is not convenient in the context of inter-organizational workflows, where we need a certain degree of workflow inter-visibility in order to perform interactions and data exchange without revealing partners know-how. The inter-visibility must be as tiny as the cooperation needs.

For a number of years research on workflow management has focused on inter-organizational issues and much has been achieved so far. In [4], the author presents some forms of workflow-interoperability and focuses on capacity sharing, chained execution, subcontracting, (extended) case transfer, loosely coupled, and public-to-private architectures. A workflow is seen as an "automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules" [5].

The inter-organizational cooperation problem has also been addressed by using the notion of agreements and contracts to define the business relationships between organizations. An example is the *CrossFlow* approach [6,7].

Moreover, the *WISE* project addressed the issue process crossing the organizational boundaries and developed an architecture to model a virtual enterprise process [8,9]. Besides, a model to support collaborative work in virtual enterprises based on process services and contracts has been proposed in [10]. Virtual enterprises are also addressed by Business Process Execution Language for Web Services (BPEL) [11] which provides an XML notation and semantics for specifying business process behavior based on Web services.

In the following we present a brief survey of the previously mentioned work that have been done in the field of inter-organizational workflows.

3.1. Approaches based on workflow interoperability

Various forms of workflow interoperability are defined in the literature, which we briefly summarize: capacity sharing, chained execution, subcontracting, (extended) case transfer, loosely coupled, public to private approach.

3.1.1. Capacity sharing

In the capacity sharing approach, tasks are executed by external resources under the control of one workflow manager.

3.1.2. Chained execution

In the chained execution approach, the process is divided into subsequent phases and each business partner takes care of one phase. The workflow process is split into a number of disjunctive sub processes executed by different business partners in a sequential order [12]. This form of interoperability is only useful for applications where the process is composed of sequentially ordered parts. Nevertheless, it was generalized into an approach to distributed workflow execution where parts are inter-mixed [13]. However, this last approach is static since it starts from a global centralized workflow where all activities are known a priori and assumes that for each activity there exists an assignment to a department or business unit of the enterprise.

3.1.3. Subcontracting

In the subcontracting approach, a sub-process is executed by another organization. There is one business partner, which subcontracts sub processes to other business partners.

3.1.4. Case transfer

In the case transfer approach, each partner uses the same workflow process and cases (i.e., workflow instance) are transferred from one partner to another. If at a specific location the process is extended with additional tasks, then this form is called extended case transfer. Cases can be transferred among partners. At any time, each case resides at exactly one location [14].

3.1.5. Loosely coupled

In the loosely coupled approach, each partner takes care of a specified part of the process which may be active in parallel.

3.1.6. Public-to-private

In the public-to-private approach, a common public workflow is specified and partitioned according to the organizations involved by private refinement of the parts based on a notion of inheritance. Each partner has a copy of the workflow process description. The public-to-private approach consists of three steps. Firstly, the organizations involved agree on a common public workflow, which serves as a contract between these organizations. Secondly, each task of the public workflow is mapped onto one of the domains (i.e., organization). Each domain is responsible for a part of the public workflow, referred to as its public part. Thirdly, each domain can now make use of its autonomy to create a private workflow. To satisfy the correctness of the overall inter-organizational workflow, however, each domain may only choose a private workflow which is a subclass of its public part [15].

Problems to be encountered on the way to workflow interoperability include mainly autonomy of local workflow processing, confidentiality that prevents complete view of local workflow [16], and especially flexibility that needs no definition of a global workflow that defines cooperation between local workflows.

In [10], the authors present an approach for process management and coordination based on synchronization points between process services. This approach provides more flexibility in order to allow partners to personalize their internal processes without affecting the cooperation.

3.2. Contracting of workflows

The inter-organizational cooperation problem has been addressed by approaches using the notion of agreements and contracts. We cite among others *COSMOS* [17], *TOWEC* [18], *CrossFlow* [19] and *WISE* [8,9]. In the following two sections we briefly discuss the *CrossFlow* and *WISE* approaches.

3.2.1. CrossFlow

The CrossFlow [19] project investigates some issues which are concerned with business processes crossing organizational boundaries. A contract-based approach is used to define the business relationships between the organizations. Within this contractual basis, inter-organizational processes can be defined and performed. However, the approach does not support arbitrary public processes and no standard definition language and semantics is provided for the enforcement of contracts between two enterprises. In addition, all enterprises involved are required to use the same software for contract enforcement.

3.2.2. WISE

The *WISE* [8,9] (Workflow-based Internet SErvices) project aims at designing, building, and testing commercial infrastructures for developing distributed applications over the Internet. It proposed a framework to compose a virtual business process through process interfaces of several enterprises. This architecture provides means to define, enact, and monitor virtual enterprises business processes, as well as to manage context aware communication among process participants. It includes an Internet workflow engine to control the business process execution, a process modeling tool to define and monitor processes, and a catalog tool to find the building blocks for the processes. A workflow engine based on the Internet is supposed to overcome the shortcoming of other workflow systems by providing workflow functionality for heterogeneous, distributed applications. WISE is platform independent. The accessibility over the Internet makes this solution scalable and open but service descriptions and the service catalog are not in line with general standards. Moreover, the centralized workflow engine inhibits dynamic selection and exchange of partners since all participants have to comply with stipulated interfaces.

3.3. Workflow specification languages

To specify inter-organizational workflows, big efforts have been made during recent years and many languages have been proposed. In the following we present a very brief survey of some proposed languages.

3.3.1. Business process execution language for web services

The Business Process Execution Language for Web Services (BPEL4WS or BPEL for short) [11] is a language for specifying business processes behavior based on Web services and business interaction protocols. It merges and extends the *WSFL* concepts of *IBM* (control structures of *WSFL* as the sequence, parallel, and loops structures) and those of *XLANG* of Microsoft (instantiation-correlation, compensation). A *BPEL* process allows the definition of two types of business processes, abstract process and executable process. The first type defines the business protocol role and describes its public aspects. The second one, defines the logic and state of the process by providing sequence of the Web service interactions conducted at each business partner. Moreover, *BPEL* defines a set of primitive activities, such as *invoke*, to invoke Web service operations. These primitive activities can be combined into more complex primitives using any of the structure activities provided such as *sequence*, *flow*, and *while*. However, the Business Process Execution Language does not support many concepts that are paramount for inter-organizational collaboration. First, it does not profit of the rich concepts of exiting workflow management systems as the notion of manual activities, applications, nor addresses the integration with them, since it uses Web services exclusively which represent a limit to call other types of services like XML services, databases, etc. Second, in the context of collaboration, it does not support the partners' heterogeneity. Incoming messages must be validated as well as transformed and enriched with additional data which represent a rigid constraint in the context of collaboration where partners are supposed to manipulate different kind of structures and process heterogeneous soft and hard infrastructures. Moreover, the collaboration description that consists of linking roles to ports is limited. Besides, *BPEL* does not cater for non-Web service interactions and the notion of independent activities, everything in a *BPEL* process is a Web service operation. Finally, *BPEL* does not provide yet a standard way to specify how flows in the same process send messages to each other. Indeed, it is possible and critical that flows in the same process be able to send messages to each other but there is no standard way to specify that.

3.3.2. WSDL

WSDL [20] is an XML-based language for locating and describing *Web services*, and how to access them. It offers four ways of message transmissions, whereby today mostly two of them are supported: one-way and request/response messages. It includes also a set of protocol bindings like *SOAP*, *MIME* and *HTTP GET/POST*.

3.3.3. WSFL

WSFL [21] is built on top of *WSDL* and can be used to refine a *WSDL* specification or compose workflow fragments. It supports workflows fragment integration with heterogeneous data structures by using *XPath* expressions.

3.3.4. XLANG

XLANG [22] refines *WSDL* service specification with behavior and allows the *WSDL* services composition specification. It uses the notion of blocks and supports message handling, timing and execption handling. It also supports ACID transactions and open nested transactions with compensation. But transactions are not allowed to span workflow fragments.

3.3.5. BPML

BPML [23] presents some similarity with *XLANG* and provides additional concepts like executable specifications, transactions spanning workflow fragments, and dynamic participation. It also offers a visibility mechanism for information hiding.

3.3.6. WSCL

WSCL [24] aims at defining the minimal set of concepts necessary for the conversation specification. This minimalism makes the language simple but restricts the expressiveness of its specifications. It does not support parallel activities, and timing constraints, for example.

3.3.7. ebXML

ebXML [25] aims at providing a framework for business to business transactions. ebXML can be seen as a global electronic market place where enterprises of any size, anywhere can find each other electronically and conduct business through exchange of XML-based business messages. It supports re-usable data types, inter-organizational transactions, and profiles and agreements. It offers interaction primitives to support timing, security, and atomicity properties. Capabilities of an individual party is described in term of Collaboration Protocol Profile (CPP) which is stored in ebXML registry (i.e., Business partners can find each other's CPP through registry). Capabilities that trading partners have agreed to use to perform a particular business collaboration s described in term of Collaboration Protocol Agreement (CPA).

3.3.8. WPDL

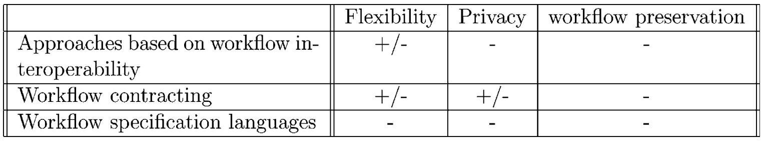
WPDL [26] is intended for the exchange of workflow types between workflow management systems but not for inter-organizational workflow specification. It also lacks interaction support.

3.4. Summary of related work vis-à-vis inter-organizational workflows requirements

Fig. 1 summarizes related work vis-à-vis inter-organizational workflows requirements: flexibility support, privacy respect principal and established workflow preservation.

In the approaches based on workflow interoperability, it is clear that support for inter-organizational workflows can be improved substantially. As an example, the approach presented by [15], describe cooperation between workflows by specifying an inter-organizational workflow and partitioning it according to the organizations involved by private refinement of the parts based on rules of inheritance. A drawback of this approach is the lack of the preservation of pre-established workflows. In fact, in this approach, one has to look for which rules, in what order and how many times one has to apply in order to match the pre-established workflow with the public part which is deduced from partitioning of the public workflow. If not impossible, this is hard to do. Moreover, there is no defined procedure to do that.

In contracting approaches, flexibility and workflow preservation requirements are not met. In the CrossFlow approach, for example, all enterprises involved are required to use the same software for contract enforcement. Moreover, the approach does not support arbitrary public processes.



	Flexibility	Privacy	workflow preservation
Approaches based on workflow interoperability	+/-	-	-
Workflow contracting	+/-	+/-	-
Workflow specification languages	-	-	-

Fig. 1. Overview of related work.

Despite their diversity, no single workflow specification language fulfills all requirements identified for specifying inter-organizational workflows [27]. In *BPEL*, for example, the collaboration description that consists of linking roles to ports is limited and limits the use of pre-established workflows. Besides, *BPEL* does not cater for non-Web service interactions and the notion of independent activities, everything in a *BPEL* is a Web service operation.

The study of the existing approaches shows that no single solution fulfills all requirements identified for the inter-organizational workflow collaboration. In consequence, the absence of end-to-end process control, which is often related to the absence and/or limitation of a single overall process ownership, has led workflow research to reexamine and to find new ways for workflow composition. One of the basic obstacles is the lack of a comprehensive model for inter-organizational workflows as a basis for contracting and standardization.

4. Steps for inter-organizational workflow cooperation

To meet the requirements we have presented above, we propose a novel approach to inter-organizational workflow cooperation. This is motivated by the idea that an inter-organizational workflow can be considered as a cooperation of several pre-established workflows of several organizations.

The approach is inspired by the Service-oriented Architecture (SOA). That architecture requires three fundamental operations: publish, find, and bind. Service providers publish services to a service broker. Service requesters find required services using a service broker and bind to them. Accordingly, our approach consists of three steps: workflow advertisement, workflow interconnection, and workflow cooperation. In the following we present these steps. Section 5 focuses on the second step of this approach, as outlined in Section 4.2.

4.1. Step 1: workflow identification and advertisement

For building an inter-organizational workflow, each organization has to advertise its offered and required activities within their workflows. Each organization identifies its partners with complementary competencies and knowledge that can be gathered to carry out projects which are not within the range of only one organization. Partner identification is based on a (semi) automated search of the new organizations and potential partners, looking for joining a virtual organization. Research will be based on the semantic description of services (i.e., workflow activities), which the organization requires, and the level of the cooperation that it wishes to establish. In other words, the profiles of the workflow activities to be interconnected. The various profiles published can be managed within an accessible registry on the Web.

Each organization does not know a priori which partners to cooperate with. The registry role provides an organization with searching and publication capabilities, which allow the organization to get partners with useful skills. In addition, it gives organizations the ability to share workflow semantic information and workflow resources. Indeed, a semantic registry is the key foundation block upon which inter-organizational workflow cooperation can be built. Registry technology enables trading partners to identify common data sets, data structures, and workflows.

4.2. Step 2: workflow interconnection using cooperation policies

Identified partners negotiate their roles within the virtual organization as well as the coordination of their workflows. The result of this step is a set of cooperation policies that describe especially the responsibilities and the roles played by the partners in the cooperation. For each partner, cooperation policies define the visibility levels of its workflows for its partners.

The policies describe (without explicit specification) a set of accepted interaction scenarios rather than one (which is the case of existing approaches since they use a workflow to specify interactions between workflows, see [15], for example). To do so, cooperation policies are defined between some virtual activities belonging to workflows of the virtual organization. A virtual activity can be connected to one or several activities belonging to one workflow and represents the level of the visibility used to preserve privacy and its know-how as well as to allow interactions with cooperating partners. The inter-visibility is reduced to be as little as is required for the cooperation (see [30–32] for more details on cooperation policies).

Connections between virtual activities and "real" ones can be changed without changing the coordination (or control flow) between virtual activities. This allows an organization to adapt and/or change its workflow internal structure without changing its role in the cooperation. Section 5 gives examples for connection between virtual and "real" activities.

4.3. Step 3: workflow cooperation and monitoring using trusted third party

Trust is a very important condition to guarantee when one would support electronic cooperation between workflows in virtual organizations. Indeed, [33] quote that the underlying basis for exchange in Internet marketplaces is trust among market participants. Significant efforts have been made in this area. For example, [34] propose models of a trusted third party in electronic commerce based on certification technologies. The idea is that a trusted third party acts as a certification authority, providing validation of market participants using public key encryption technology. We call this trusted third party a contracting authority. This authority will assume the monitoring and the control of cooperation between workflows.

4.4. Cooperation sequence diagram

Fig. 2 depicts a sequence diagram showing the different steps described above as well as their interactions. The diagram is composed of a set of partners who will cooperate, a registry containing the participants profiles, and a contracting authority ensuring the cooperation monitoring and control. We identify three logical blocks. In the first block, partners publish some of their activities into the registry (publish(...)). Then, to carry out a work that is not with the range of only one organization, a partner begins by searching organizations with complementary skills via the activities they published (find(...)). When these organizations are found, the registry informs them that they will be partners in an inter-organizational workflow (partner(...)). This constitutes the workflow identification step.

In the second block, identified partners negotiate their roles and responsibilities as well as the coordination of their cooperation (negotiate(...)), and then interconnect to each other (connect(...)). This forms the workflow interconnection step.

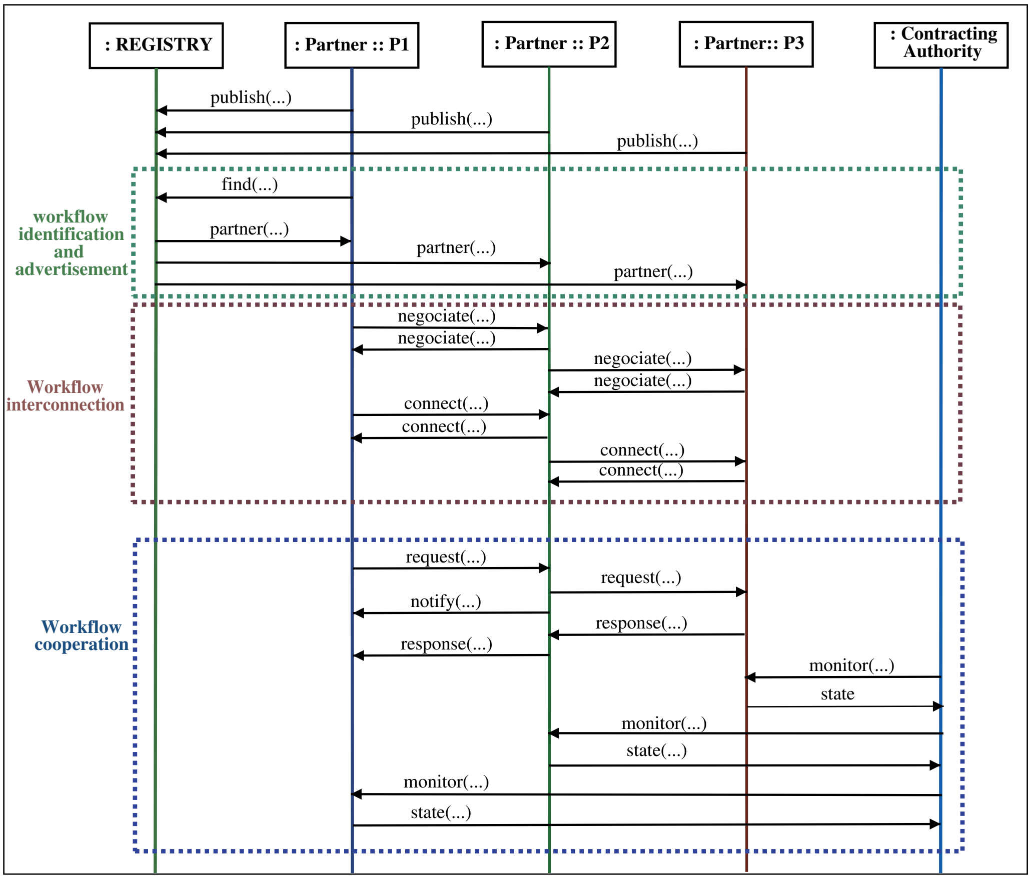


Fig. 2. Sequence diagram of the workflow interaction.

After their interconnection, partners are now ready to cooperate and communicate by sending data (request(...)) and/or receiving data (response(...)) and/or being notified (notify(...)). Their communication is monitored and controlled by the contracting authority (monitor(...), state(...)).

5. Describing cooperation in inter-organizational workflows

In this section we discuss how to describe workflow cooperation in order to meet the requirements we have presented in Section 2, including their drawbacks.

5.1. Running example

To illustrate the problem, consider the example presented by Fig. 3, involving four business partners (a customer, a producer and two suppliers) illustrating the three cooperation phases mentioned above. The Customer sends an order for a product. Then it receives a notification

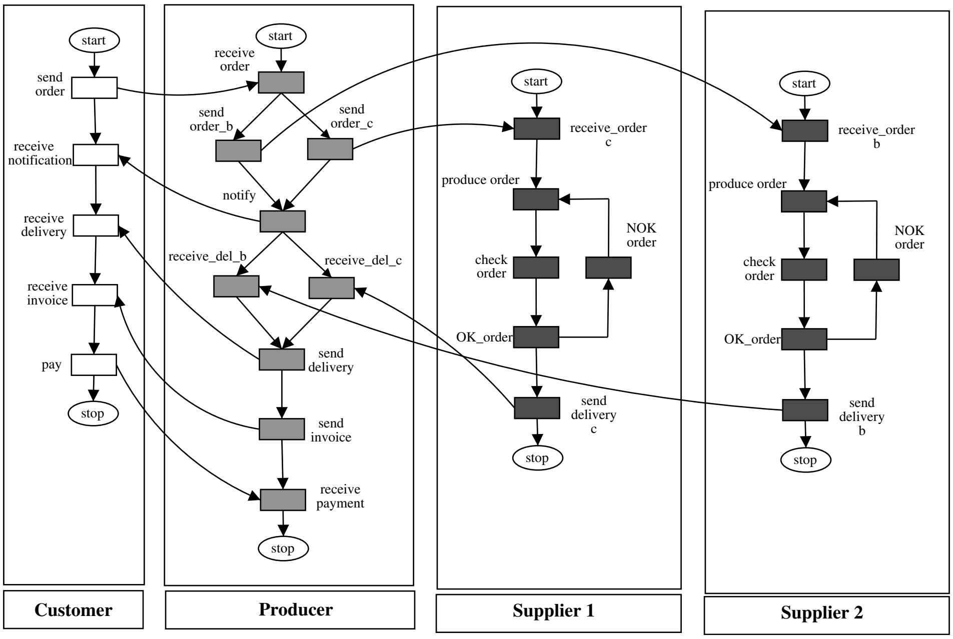


Fig. 3. An inter-organizational workflow.

announcing that the product has been taken into account by the producer. When the product is ready, the customer receives the delivery and then the invoice. Finally, he pays for the product he has ordered. The producer, waits for an order request. Then he searches for two suppliers to provide him with needed components in order to satisfy the received order. After that, he notifies the customer that his order is taken into account and waits for the suppliers' response. When he receives the requested components, he assembles them and delivers the product to the customer. Finally, he sends the invoice and waits for the payment. The last partners are the two suppliers which, in our example and for simplicity purposes, have exactly the same workflow. When a supplier receives an order, he begins by producing it and then checking it. If the product conforms to the specification, it will be sent to the requester, otherwise another product must be produced and the process is repeated until the order is satisfied.

As it is modeled, this inter-organizational workflow does not allow any participating workflow (of a customer, the producer or the supplier) to change its internal control flows even though the role played by this participating workflow is still the same. Hence, one could argue that the inter-organizational workflow is hard-wired and public. As a consequence, changing the workflow is cost-intensive.

For preserving privacy, one of our objectives is to describe the cooperation in an inter-organizational workflow without specifying the internal structure participating workflows.

5.2. Definitions

A cooperation within an inter-organizational workflow is considered to be the exchange between participating workflows. We use dataflows as the vehicle for providing cooperation. Two types of cooperation called Produce and Consume, are used here. A Produce is used by a workflow (A) to initiate an activity in a second workflow (B), and/or to provide input to that activity. A Consume is used by a workflow (B) to send the results of an activity (or a set of activities) to its requesting workflow (A), providing output. Although the Produce and Consume cooperation types are clearly complimentary, there is no requirement that they always be used in conjunction.

Produce and Consume cooperation types among workflows are defined by means of cooperative activities. Cooperative activities denote points in the workflow where a dataflow is produced or consumed, thereby allowing synchronization and data exchange with other workflows, as well as notification of state changes or requests of activity execution. Cooperative activities can be part of the flow structure of a workflow, just like ordinary activities, and can be of two types: producing activities or consuming activities.

5.2.1. Producing cooperative activity

We call a producing cooperative activity each activity that produces a dataflow for an external activity that belongs to another workflow. In Fig. 3, the activity *send_order* is a producing cooperative activity.

5.2.2. Consuming cooperative activity

We call a consuming cooperative activity, each activity that consumes a dataflow from an external activity that belongs to another workflow. In Fig. 3, the activity *receive_notification* is a consumer cooperative activity.

5.2.3. Cooperative activity

We call a cooperative activity a cooperative consuming and/or producing activity.

5.3. Internal, cooperative, and public processes

The important things in the example of Fig. 3 are the cooperative activities and the interactions (control flow) between them rather than the non-cooperative activities or the control flow between them. Many items depicted in Fig. 3 are not important (necessary) vis-à-vis to the description of cooperation. We quote among others the following information in workflow of the *supplier* 1:

- the *produce_order*, *check_order*, *OK_order* and *NOK_order* activities,
- the control flows between these activities.

If we hide this type of information, we can provide more flexibility to the *supplier* 1 to perform the activities *produce_order*, *check_order*, *OK_order* and *NOK_order* separately or within one or two activities without changing the interactions with the *producer*. Moreover, this preserves privacy since it hides the internal structure of the workflow.

In order to be able to hide internal structures of participating workflows, we propose to use the notion of public process representing one view, among others, of a participating workflow. As an example, Fig. 4 depicts public processes of the workflows *supplier* 1 and *producer*. The public process of the *producer* is composed of two virtual activities connected to two cooperatives activities (*send_order_c* and *receive_del_c*). The public view of *supplier*_1 is also composed of two virtual

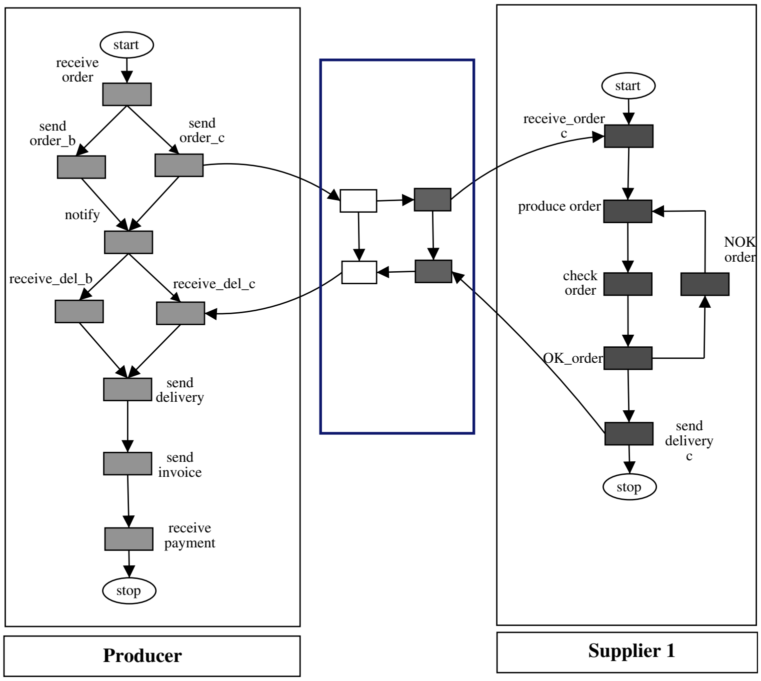


Fig. 4. Public vs. internal processes.

activities connected to two cooperative activities (*receive_order_c* and *send_delivery_c*). The connection between the virtual activities describe the cooperation between these two workflows (i.e., the cooperation policy).

By representing a public process, we hide their internal structures and we focus on cooperation between the workflows. In fact, by Fig. 4, only virtual activities and the control flows between them are visible. Connections between virtual and cooperative activities are not visible. Given a virtual activity, partners cannot identify which cooperative activity will be performed. It is up to the workflow owner to decide which cooperative activity will eventually be performed.

As a result, inter-organizational workflow cooperation presented in Fig. 4, composed of virtual activities and their connection, represents a cooperation contract between the participating workflows. This contract allows several cooperation scenarios, in which internal workflow structures are changed, and privacy is preserved. It allows participating organizations to adapt and modify their workflows without changing their cooperation.

For that reason the *producer* can change its internal structure (see left part of Fig. 5) without changing its role in the cooperation (the public process is the same). In the same way, the *supplier* can change its internal structure (see right part of Fig. 5).

For the three different scenarios presented in Figs. 4 and 5, participating workflows have the same public processes and the same interconnection between them, representing the *cooperation contract*.

Fig. 6 illustrates the cooperation architecture involving partners cooperating via their public processes, where the communication is ensured by a middleware.

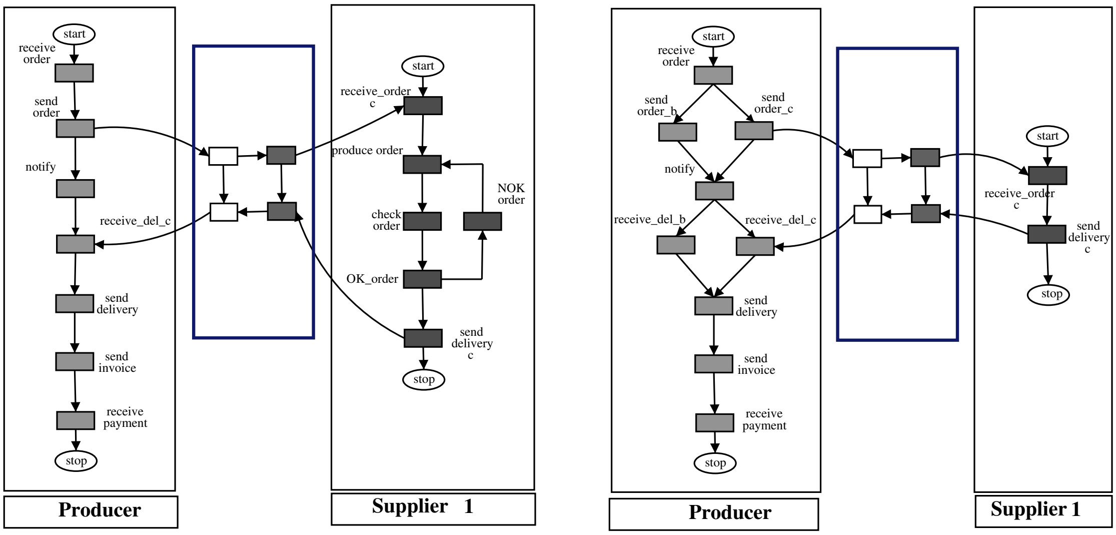


Fig. 5. Producer and Supplier change their internal processes.

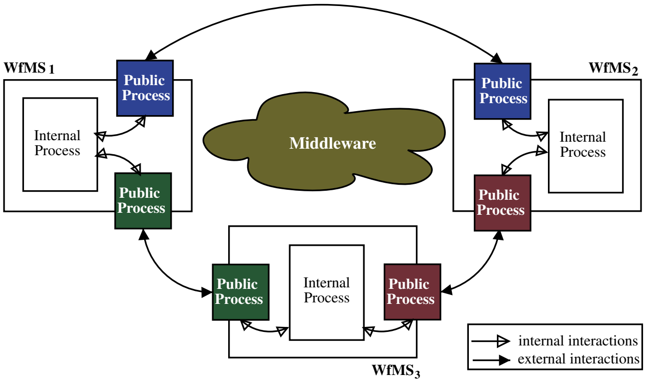


Fig. 6. Cooperation architecture.

In Fig. 6, several organizations with complementary competencies want to cooperate and carry out a work that is not within the range of only one of them without revealing their proper skills. To do so, these organizations hide their internal processes and generate one or more public processes to communicate with the other partners with an inter-visibility as tiny as cooperation needs. Finally, the communication will be ensured by a middleware.

We have introduced by this example the notion of public processes. In the following section we present its design process.

5.4. Definition process of public processes

In this section, we present how the cooperative process can be deduced from the internal one as well as how a public process can be defined in order to depict a selected view of the cooperative one.

5.4.1. Cooperative processes

First, each partner defines its internal process including its know-how where all dataflows and internal (cooperative and non-cooperative) activities as well as their control flows are specified (or identified). An internal process can abstract or specialize a pre-established process.

In our example, at the beginning, none of the business partners knows the other partners it will collaborate with. Thus, each of them models its proper internal process specifying all steps needed to accomplish its services independently of the others. The customer, producer, first and second supplier's internal processes are illustrated in Fig. 7.

Then, for each internal process, we can identify a sub-process where all activities that will produce and/or consume dataflows (cooperative activities) as well as their coordination are extracted and all extra and unnecessary information are hidden. We call such a sub-process a cooperative process. The cooperative process is the "minimal" connected and compacted sub-process that contains all cooperative activities.[1]

To pass from an internal workflow to a cooperative one, we begin with finding the minimal connected workflow and then transform it into a cooperative workflow using a compacting procedure. First, we determine a minimal connected workflow composed of all the cooperative activities as well as the internal activities between them ensuring the connectivity of the whole workflow. Second, we use a compacting procedure to transform the the "minimal" connected process into a cooperative one in which non-cooperative activities act just as connection activities.

In our example, the producer can cooperate with three partners: a partner that requests an order (customer) and two others to construct the received order components (two suppliers). At this step, he extracts the different cooperative activities to form the corresponding cooperative process allowing the cooperation with the other partners. The resulting process is composed of the following activities: the activities receive_order, notify, send_delivery, send_invoice, receive_payment to allow cooperation with the customer, and the activities $send\_order_{b/c}$, $receive\_del_{b/c}$ to permit cooperation with the two suppliers.

The customer needs only one partner to cooperate with: a partner for the production of a specified product. He begins by building his internal process and then extracts all cooperative activities to form the cooperative process which, in our case, is composed of five activities: send_order, receive_notification, receive_delivery, receive_invoice and the pay activities.

Finally, the same rules are applied to the suppliers's workflows. In our example, the suppliers have only two cooperative activities allowing them to receive and then deliver an order: receive_order and send_delivery in order to cooperate with the producer.

Fig. 8 presents the cooperative workflows corresponding to the four partners.

5.4.2. Public processes

After determining the internal and cooperative processes, public processes are created for one of the following purposes: A workflow wants to expose some activities or it wants to access activities of workflows of other organizations.

A public process is a public view of the cooperative process. A cooperative process can expose many public processes at the same time. Each one is used to define the cooperative with one part-

[1] See Appendix A for formal definition of a cooperative process.

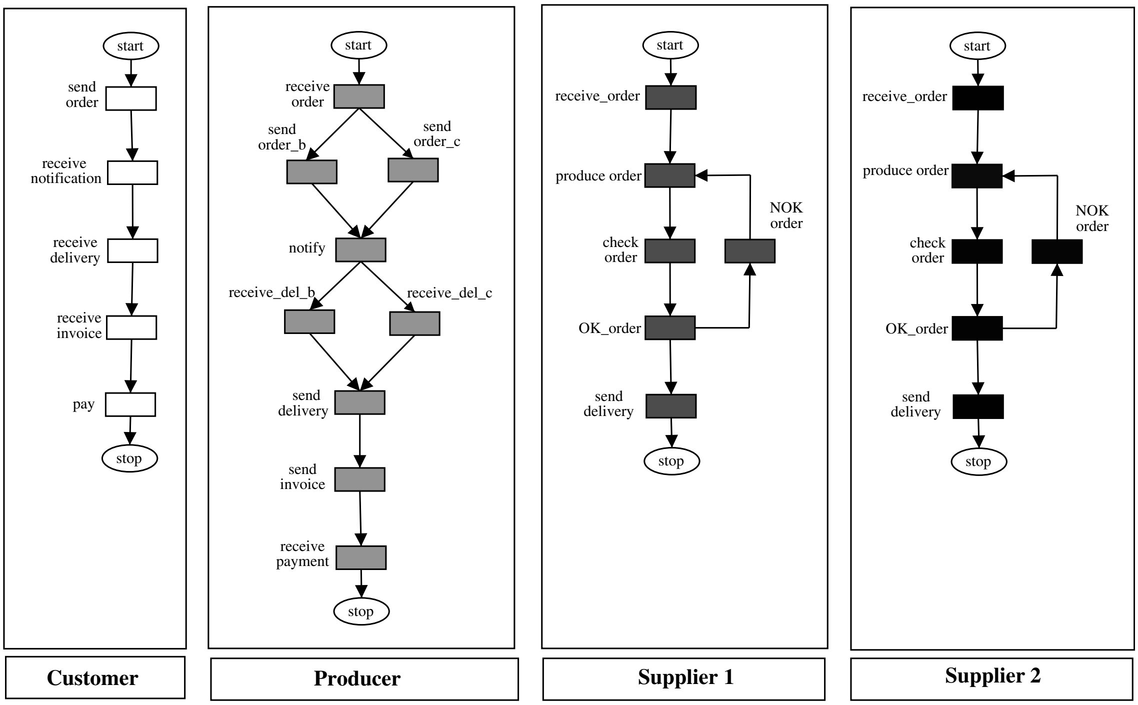


Fig. 7. Customer, producer and suppliers's internal processes.

ner. A public process consists of virtual activities which represent a subset of the cooperative process activities. Virtual activities are not supposed to produce or consume output/input. Nevertheless, they are supposed to transfer output/input data to/from other workflows. Similarly they are not meant to be executed by local role.

In the following we present how public processes can be deduced from the cooperative ones using the notion of cooperation policy. The cooperation policy establishment process we propose in this paper integrates the participants' roles during cooperation, the dataflow that will be passed from one partner to another and the participant's public processes that will be exposed and accessed by external organizations. Given a set of organizations with complementary competencies, we describe their interactions in terms of a *cooperation policy*, which is a set of rules between participants defined in terms of dataflow and access contracts, and workflows public process definition.

In order to establish a cooperation policy, we propose a three phases process: a dataflow contract establishment to express dataflows to be exchanged between partners, an access contract establishment to express allowed activities one partner can execute on its behalf by an external partner and workflow public process definitions.

After determining the different activities participants' workflow can be executed on its behalf within an external workflow, each partner specifies the different dataflows to exchange with the other partners. The result of this step is a set of rules associating, in a peer to peer manner, the partners with the data they can send to each other. We call this set of rules a *dataflow contract*. In our example, the dataflow contracts are illustrated by Fig. 9.

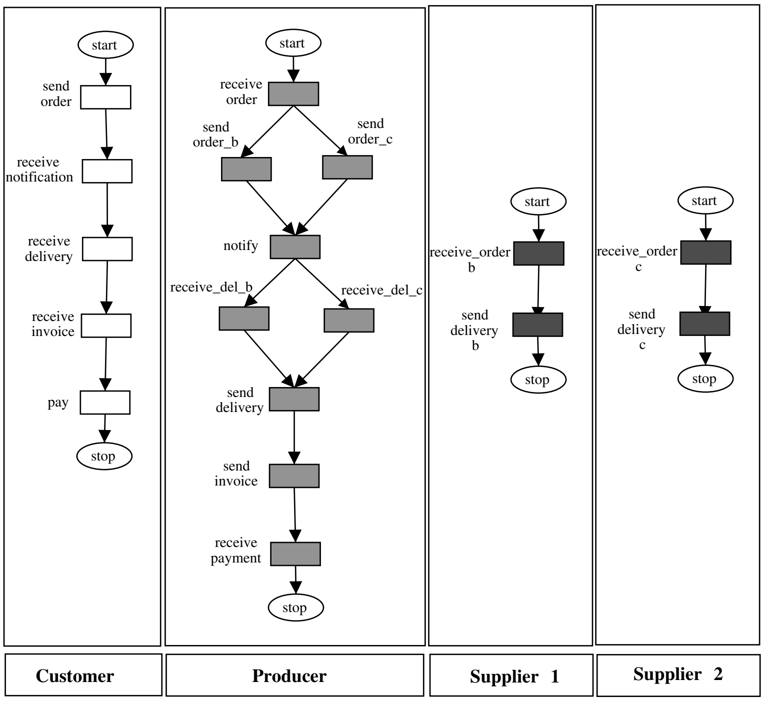


Fig. 8. Partners' cooperative workflows.

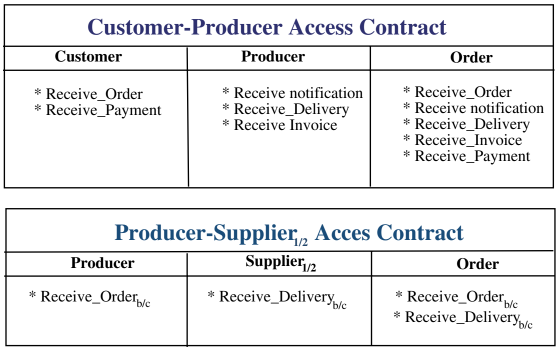
Customer-Producer Dataflow Contract	
Customer	**Producer**
* order_a * payment	* notification * delivery_a * invoice

Producer-Supplier $_{1/2}$ Dataflow Contract	
Producer	**Supplier$_{1/2}$**
* Order$_{b/c}$	* Delivery$_{b/c}$

Fig. 9. Partner's dataflow contracts.

The objective of the access contract establishment is to determine, for each partner, the set of external activities (activities executed by external workflows on its behalf) it can execute as well their coordination. This consists of identifying the roles of the different identified partners, as well as the coordination of their cooperative activities. Hence, the result is a set of rules describing the responsibilities and the roles played by each partner in the cooperation. We call this set of rules an *access contract*. In Fig. 10, we give the access contracts between the producer and the customer, and between the producer and the two suppliers.

Based on the access contracts, each partner defines the visibility levels of its workflow to the other partners. Hence, many views permitting to cooperate without revealing its know-how are generated. This allows providing each partner with only information it needs to know by concealing all details and extra-information. This reduces the inter-visibility to be as little as required for the cooperation.



Customer-Producer Access Contract

Customer	Producer	Order
* Receive_Order * Receive_Payment	* Receive notification * Receive_Delivery * Receive Invoice	* Receive_Order * Receive notification * Receive_Delivery * Receive_Invoice * Receive_Payment

Producer-Supplier$_{1/2}$ Acces Contract

Producer	Supplier$_{1/2}$	Order
* Receive_Order$_{b/c}$	* Receive_Delivery$_{b/c}$	* Receive_Order$_{b/c}$ * Receive_Delivery$_{b/c}$

Fig. 10. Partner's access contracts.

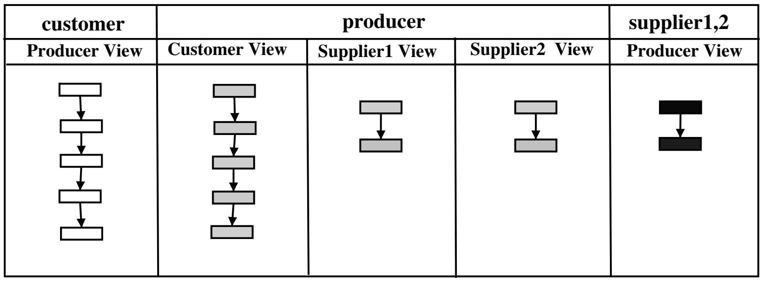


Fig. 11. Partner's public views.

In our example, the customer cooperates only with the producer, hence it generates only one public process composed of five virtual activities. The producer generates three public processes, the first is composed of five virtual activities to interconnect to the customer and the two other ones are identical and composed of two virtual activities in order to permit the sent of an order and the reception of the correspondent delivery. Fig. 11 shows the partner's public views workflows.

After defining internal, cooperative, and public processes, the four partners are ready to interconnect to each other and then cooperate via their public processes (see Fig. 12). The communication between them will be ensured by a middleware.

The interconnection of the different partners is illustrated in Fig. 13, where every partner is composed of three main components: the public processes for the cooperation with other partners, the internal process containing the know-how of the organization and the partner *WfMS* that will execute the corresponding workflow.

6. Implementation issues

Discussing details of implementation is not in the intended scope of this paper. Currently we are implementing a prototype of our workflow cooperation platform (see Fig. 14). Existing workflow management systems (*WfMS*) can plug into our platform. This operation can be done if the

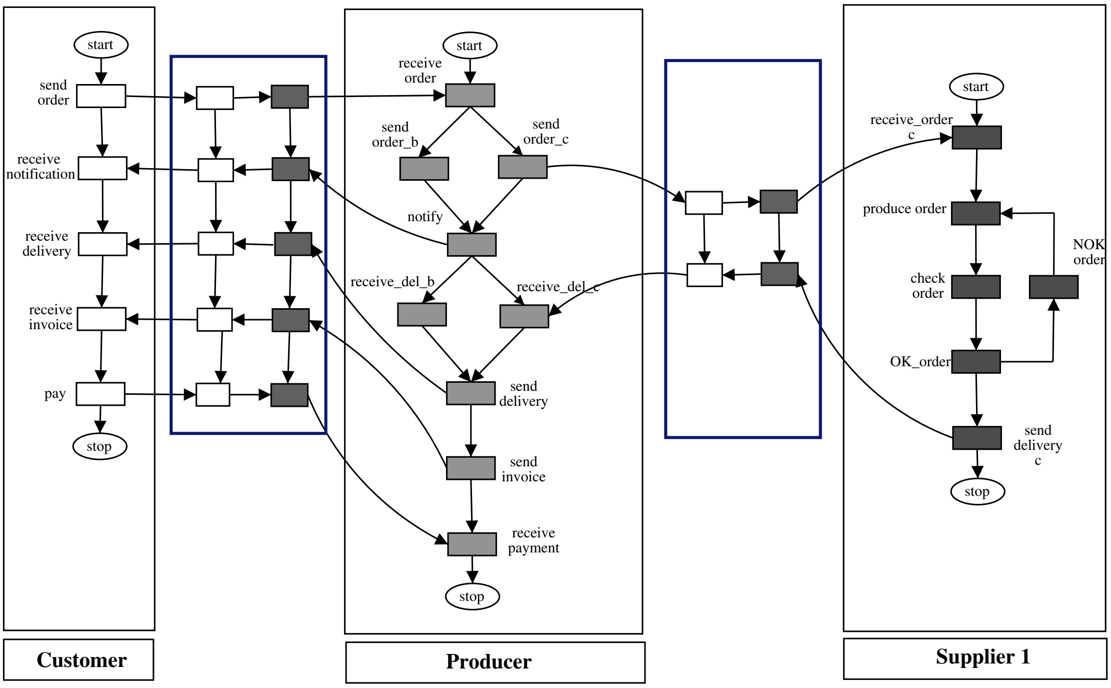


Fig. 12. Partners' interconnection.

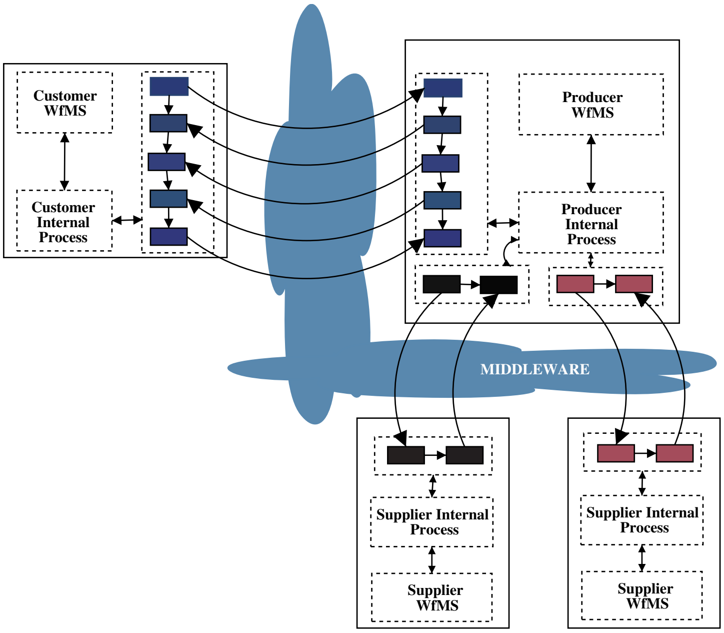

Customer WfMS
Customer Internal Process
Producer WfMS
Producer Internal Process
MIDDLEWARE
Supplier Internal Process
Supplier WfMS
Supplier Internal Process
Supplier WfMS

Fig. 13. Customer, producer and Suppliers interaction scenario.

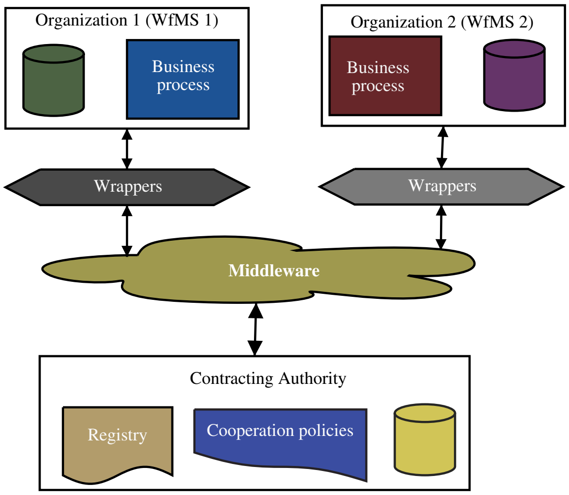


Fig. 14. A platform for inter-organizational workflow.

WfMS fulfills two conditions. First, the *WfMS* can call external applications (e.g., programs, Web services, etc.). Second, the *WfMS* allow external applications to invoke any step within a workflow it manages. To adapt incoming and outgoing invocations, we use wrappers.

Our implementation relies on the development of patterns for cooperation. Actually, a workflow may be related to proxy and adapter patterns. The wrappers shown in Fig. 14 represent one proxy and one adapter for each workflow. While the proxy pattern limits accesses to workflow resources (data access rights, method visibility, visibility of business process events), the adapter pattern provides a new interface to the adapted workflow (ability to exchange data, coordination ability, ability to control interaction consistency).

The role of wrappers is interfacing of a workflow with the registry and the contracting authority. Namely, the wrapper serves a workflow by adapting its actions (which depend on the characteristics of the system it holds). The actions performed can be data management operations and coordination actions. Interactions between workflows are controlled by the contracting authority. Only interactions that satisfy negotiated cooperation policies are accepted.

One of the goals of our work is to allow workflows to cooperate through partial visibility of their resources (e.g., data sharing, group awareness). In addition, it is necessary to provide to cooperating workflows means to coordinate their actions and to work in an autonomous way while being aware of what occurs in the virtual organization. For this reason, each workflow has to provide events allowing other services to know the state of its execution and its private data. Since each workflow is interested in specific information, it is beneficial to control the visibility of workflow events. This control is provided by the adapter.

So far, we have developed two wrappers for two existing workflow management systems: *XFlow* [2] and *OSWorkflow*.[3] In the following, we present both of the *WfMS*'s and then describe

[2] XFlow: http://xflow.sourceforge.net/.

[3] OSWorkflow: http://www.opensymphony.com/osworkflow/.

the cooperation scenario between them. *XFlow* is a pure J2EE platform for building, executing and managing business processes and workflows. It runs within an EJB and servlet container. JBoss 4.0 (with bundled Tomcat) is the container used in our implementation. The architecture of *XFlow* supports distributed and parallel processes within an organization's firewall as well as across organizational boundaries.

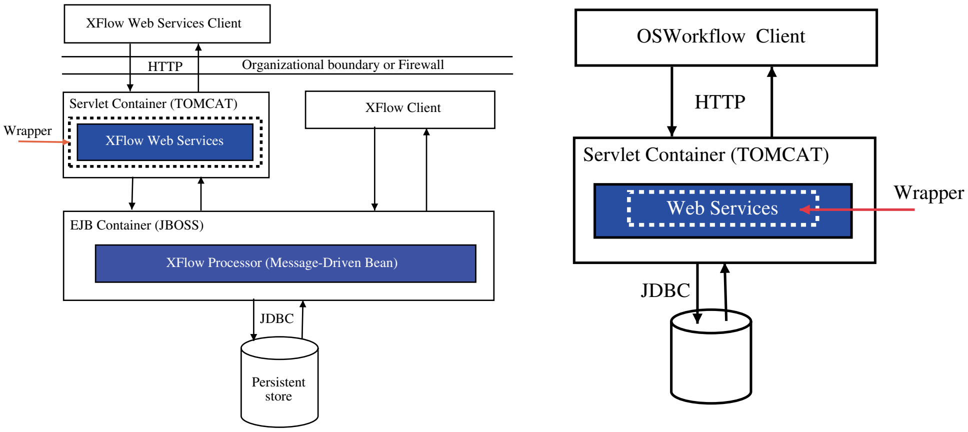

XFlow Web Services Client
HTTP
Organizational boundary or Firewall
Servlet Container (TOMCAT)
Wrapper
XFlow Web Services
XFlow Client
EJB Container (JBOSS)
XFlow Processor (Message-Driven Bean)
JDBC
Persistent store
OSWorkflow Client
HTTP
Servlet Container (TOMCAT)
Web Services
Wrapper
JDBC

Fig. 15. XFLOW and OSWorkflow deployment.

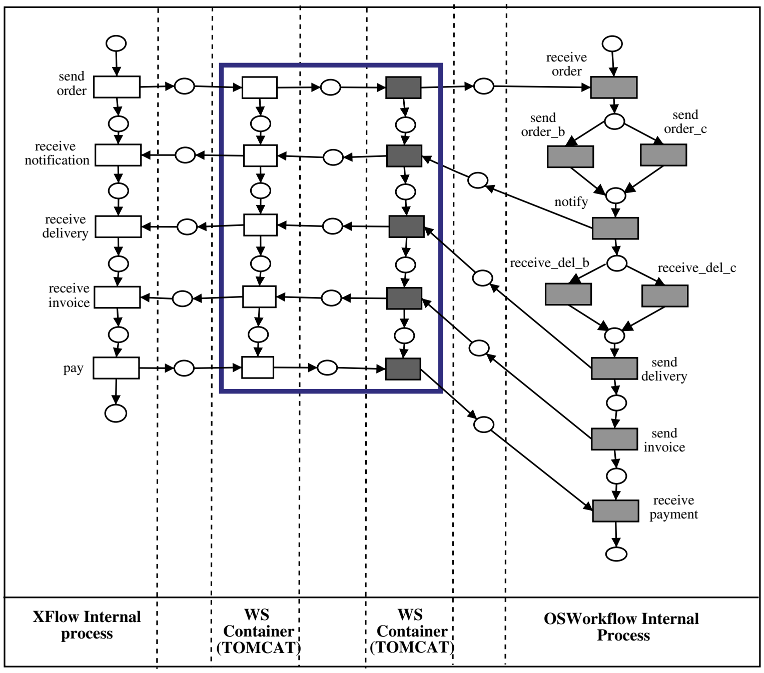


Fig. 16. Communication scenario.

The second workflow management system we have used is called *OSWorkflow* and is a Java-based open source project.

We mention here that both of the *WfMS*'s used fulfill the two conditions of cooperation which are the capability of calling external applications and allowing these ones to invoke any step within a workflow they manage.

The communication between the two systems is supported by a Web service middleware. And in order to preserve the system's privacy and limit accesses to workflow resources (data access rights, method visibility, etc.), we have developed two wrappers that play the proxy role for both of them. These wrappers are in the form of two simple web services, deployed in WS containers (Tomcat), that launch some activities of the workflows. Fig. 15 shows an overview of the *WfMS's* deployment.

Fig. 16 shows the communication scenario between the two *WfMS's* we have interconnected. The figure is composed of three major blocks. The first contains the internal workflow of the producer, the second contains the internal workflow of the customer, and finally, we find the public views used by the partners to cooperate. In order to run the example, the first step consists in launching the workflow management systems on the both sides. Then, the producer and the customer cooperate and each one calls the activities it is authorized to invoke via Web services.

After cooperating, both of the workflow systems progress independently.

7. Conclusion and future work

In this paper we presented important steps to provide support for inter-organizational workflows. The relevance of inter-organizational workflows is best seen when considering emerging virtual organizational forms, consisting of geographically dispersed teams and their respective business processes. Our approach allows for partial visibility of workflows and their resources, thus providing powerful ways for inter-organizational workflow configuration. Varying degrees of visibility of workflows enable organizations to retain required levels of privacy and security of internal workflows. Furthermore, the view concept provides a high degree of flexibility for participating organizations, since internal structures of collaborative workflows may be adapted without changes in the inter-organizational workflows. Currently we are in the process of finalizing building a prototype system where our ideas are implemented, validated, and tested.

This paper presents an approach to inter-organizational workflows. Many open issues we are addressing and we have to deal with in our future work. For example, the question here is what are the necessary properties local workflows have to satisfy in order to be consistent with the cooperation? Examples of these properties are soundness and correctness criteria. What are the mechanisms utilized for semantic registries? More research is requited on those issues.

Consistency of inter-organizational workflows, as introduced in [14], is defined as soundness of the global (i.e., inter-organizational) workflow and the local workflows. In addition, [35,36] has proposed an approach of a decentralized solution that relies the soundness of a constructed view based on the local workflow combined with the party's view on the the global workflow, which is not specified explicitly. In particular, the local workflows are extended by bilateral interactions they are involved in. With our approach, we are now improving the minimalization technique of the internal workflow to preserve four kinds of behaviors on the cooperative activities: sequential behaviors, alternative behaviors, parallel behavior, and divergent behavior. To accomplish this task, we are basing our work on the exploitation of linear invariants in Petri nets.

Appendix A

In this appendix we present how to pass from an internal workflow to a cooperative one in two steps. We begin with finding the minimal connected workflow and then transform it into a cooperative workflow using a compacting procedure.

In the first step, we start from an internal workflow. Based on an incremental construction, we determine a minimal connected workflow composed of all the cooperative activities as well as internal activities between them ensuring the connectivity of the whole workflow.

The next step consists of hiding all extra activities that will not play a connection role in the workflow. This step will keep only internal activities that will act just as connectors between cooperatives activities that are called *connection activities*.

To illustrate how to pass from an internal workflow to a cooperative one, consider the example presented by Fig. 17, showing three workflows: an internal workflow, the corresponding minimal connected wokflow and finally the compacted workflow. The first workflow presents the know-how of the organization and contains all required activities as well as their coordination. We distinguish two kinds of activities: internal ones presented by white rectangles (t_1, t_2, t_4, t_5, t_9, t_{12}, t_{13} and t_{14}), and cooperative ones presented by black rectangles (t_3, t_6, t_7, t_8, t_{10} and t_{11}) (Fig. 17(b)). The second workflow presents the minimal connected workflow where we only retain the cooperative activities and internal ones that ensure the connectivity of the workflow (t_1, t_2, t_4, t_5 and t_9). This workflow is then compacted in order to eliminate all extra information and activities.

A.1. Minimal connected workflow

Definition 1. A Workflow $W(P, T, F)$ is determined by:

- a finite set $P = \{p_1, p_2, \ldots, p_n\}$ of places,
- a finite set $T = \{t_1, t_2, \ldots, t_m\}$ of transitions ($P \cap T = \emptyset$),
- a set of arcs $F \subseteq (P \times T) \cup (T \times P)$.

The set of input (output) places for a transition t is denoted $\bullet t$ ($t\bullet$). The set of transitions sharing a place p as output (input) place is denoted $\bullet p$ ($p\bullet$).

The set $T = T_{\text{coop}} \cup T_{\text{int}}$ where T_{coop} is the set of cooperative activities and T_{int} is the set of internal activities.

Definition 2. A petri net $W(P,T,F)$ is a WF-net (Workflow net) if and only if [14]:

1. W has one source place $i \in P$ such that $\bullet i = \emptyset$ and one sink place $o \in P$ such that $o\bullet = \emptyset$, and
2. Every node $x \in P \cup T$ is on a path from i to o.

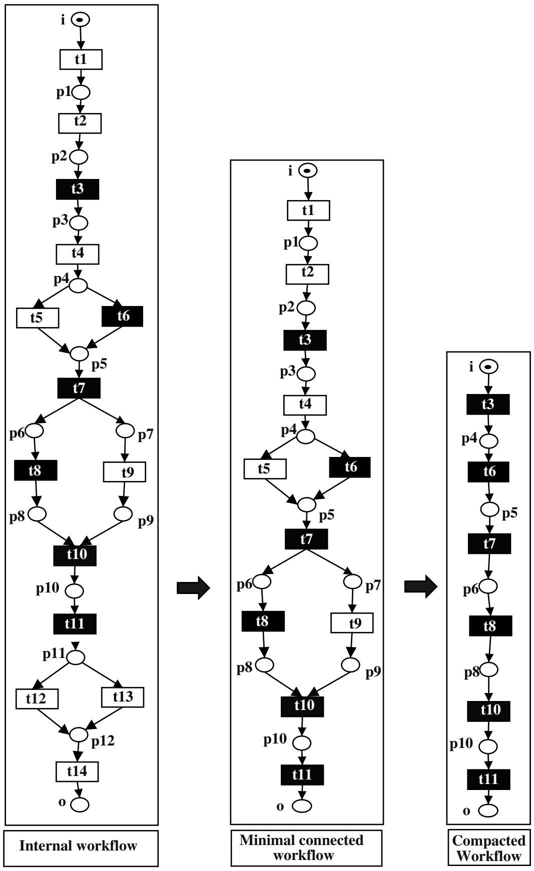


Fig. 17. Internal-minimal-compacted workflows.

Definition 3. Let $W(P,T,F)$ be a WF-net, $T = T_{\text{coop}} \cup T_{\text{int}}$. A minimal cooperative connected workflow $W_c(P_c,T_c,F_c)$ is determined by:

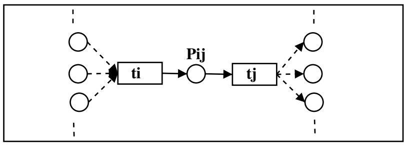
- $T_c \subseteq T$,
- $T_{\text{coop}} \subseteq T_c$,
- $P_c \subseteq P$,
- W_c is minimal connected (i.e., $\nexists W'_c(P'_c,T'_c,F'_c) \neq W_c$ where W'_c is connected, $T_{\text{coop}} \subseteq T'_c$, $T'_c \subseteq T_c, F'_c \subseteq F_c$ and $P'_c \subseteq P_c$) ($W_1(P_1,T_1,F_1) = W_2(P_2,T_2,F_2)$ iff $P_1 = P_2$, $T_1 = T_2$ and $F_1 = F_2$).

A.2. Cooperative workflow

After determining the minimal connected workflow, the next step consists of simplifying the workflow by removing all unnecessary information and hiding activities and flows that do not play any role in the cooperation or the maintaining of the workflow connectivity.

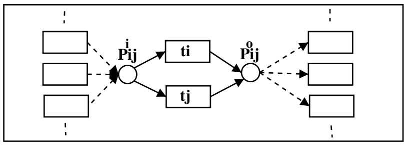
Definition 4. Let t_i, $t_j \in T$ and $p_{ij} \in P$. t_i and t_j are sequent and denoted seq(t_i, t_j,p_{ij}) if and only if

- $\bullet p_{ij} = \{t_i\}$,
- $p_{ij}\bullet = \{t_j\}$,
- $t_i\bullet = \{p_{ij}\}$,
- $\bullet t_j = \{p_{ij}\}$.



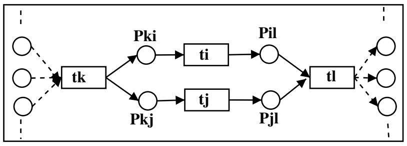
Definition 5. Let t_i, $t_j \in T$ and p^i_{ij}, $p^o_{ij} \in P$. t_i, t_j are alternative and denoted alt(t_i, t_j, p^i_{ij}, p^o_{ij}) if and only if

- t_i, $t_j \in p^i_{ij}\bullet$,
- t_i, $t_j \in \bullet p^o_{ij}$,
- $\bullet t_i = \bullet t_j = \{p^i_{ij}\}$,
- $t_i\bullet = t_j\bullet = \{p^o_{ij}\}$.



Definition 6. Let t_i, t_j, t_k, $t_l \in T$ and p_{ki}, p_{il}, p_{kj}, $p_{jl} \in P$. t_i, t_j are synchronized and denoted sync(t_i, t_j, t_k, t_l,p_{ki},p_{il},p_{kj},p_{jl}) if and only if

- $\bullet p_{ki} = \{t_k\}$, $p_{ki}\bullet = \{t_i\}$,
- $\bullet p_{kj} = \{t_k\}$, $p_{kj}\bullet = \{t_j\}$,
- $\bullet p_{il} = \{t_i\}$, $p_{il}\bullet = \{t_l\}$,
- $\bullet p_{jl} = \{t_j\}$, $p_{jl}\bullet = \{t_l\}$,
- $\bullet t_i = \{p_{ki}\}$, $t_i\bullet = \{p_{il}\}$,
- $\bullet t_j = \{p_{kj}\}$, $t_j\bullet = \{p_{jl}\}$.



A.3. Definition procedure of cooperative workflow

Given a minimal cooperative connected workflow $W_c(P_c, T_c, F_c)$, the W_c activities are compacted based on the following rules:

1. *Rule*$_1$

$\exists t_i, t_j \in T, p_{ij} \in P$ where

$P = \{p_1, \ldots, p_{ij}, \ldots, p_n\}$

$P' = P - \{p_{ij}\}$

$T = \{t_1, \ldots, t_i, t_j, \ldots, t_m\}$

$T' = T - \{t_i\}$

$F = \{f_1, \ldots, (t_i, p_{ij}), (p_{ij}, t_j), \ldots, f_k\}$

$F' = F - (\{(t_i, p_{ij}), (p_{ij}, t_j)\} \cup \{(p_k, t_i)/p_k \in \bullet t_i\}) \cup \{(p_k, t_j)/p_k \in \bullet t_i\}$

$C = \mathrm{seq}(t_i, t_j, p_{ij}), t_i \in T_{\mathrm{int}}$ and $t_j \in T_{\mathrm{int}}$

$Rule_1\colon \dfrac{(P, T, F), C}{(P', T', F')}$

This rule shows that if we dispose of an internal activity t_i followed by another internal activity t_j and linked to it via p_{ij}, then we eliminate t_i and p_{ij} as well as all the flows between t_i and t_j. Moreover, we get rid of all the flows coming from $\bullet t_i$ to t_i and create new flows linking the places belonging to $\bullet t_i$ to t_j. This case is illustrated in Fig. 18.

In our example, t_1 and t_2 are sequent activities and both of them is internal. By applying *rule*$_1$ we eliminate the activity t_1, the place p_1 and the flows (t_1, p_1) and (p_1, t_2) that link t_1 to t_2 and create the new flow (i,t_2) that links the source i to t_2 (Fig. 19).

2. *Rule*$_2$

$\exists t_i, t_j \in T, p_{ij} \in P$ where

$P = \{p_1, \ldots, p_{ij}, \ldots, p_n\}$

$P' = P - \{p_ij\}$

$T = \{t_1, \ldots, t_i, t_j, \ldots, t_m\}$

$T' = T - \{t_i\}$

$F = \{f_1, \ldots, (t_i, p_{ij}), (p_{ij}, t_j), \ldots, f_k\}$

$F' = F - (\{(t_i, p_{ij}), (p_{ij}, t_j)\} \cup \{(p_k, t_i)/p_k \in \bullet t_i\}) \cup \{(p_k, t_j)/p_k \in \bullet t_i\}$

$C = \mathrm{seq}(t_i, t_j, p_{ij}), t_i \in T_{\mathrm{int}}$ and $t_j \in T_{\mathrm{coop}}$

$Rule_2\colon \dfrac{(P, T, F), C}{(P', T', F')}$

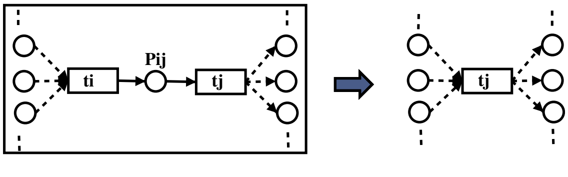


Fig. 18. *Rule*$_1$: compaction of two sequent internal activities.

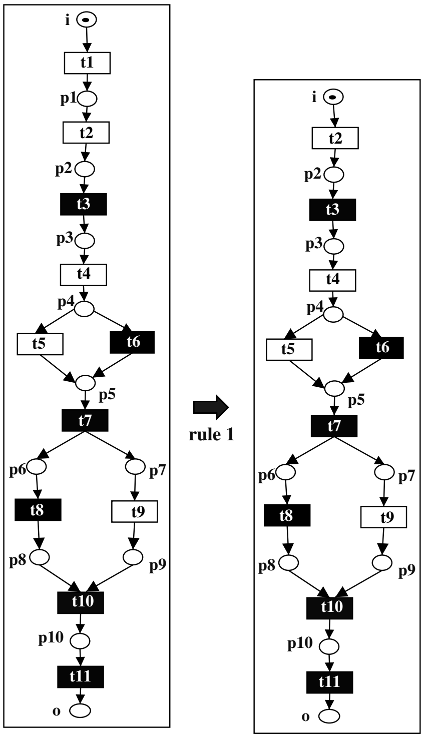


Fig. 19. Result of *rule*$_1$ application.

Rule$_2$ shows that if we dispose of an internal activity t_i followed by a cooperative activity t_j and linked to it via p_{ij}, then we eliminate t_i, p_{ij} and all the flows existing between t_i and t_j (i.e. (t_i,p_{ij}) and (p_{ij}, t_j)). In addition, we eliminate all the flows coming from $\bullet t_i$ to t_i and create new ones linking the places belonging to $\bullet t_i$ to t_j. This case is illustrated in Fig. 20.

In Fig. 21, t_2 is an internal activity followed by a cooperative activity t_3 and linked to it via p_2, then *rule*$_2$ can be applied and as a result we eliminate the activity t_2, the place p_2 and the flows (i, t_2), (t_2,p_2) and (p_2, t_3) and create a new flow linking the source i to t_3 (Fig. 21).

3. *Rule*$_3$

$$\exists t_i, t_j \in T, p_{ij} \in P \text{ where}$$
$$P = \{p_1, \ldots, p_{ij}, \ldots, p_n\}$$
$$P' = P - \{p_{ij}\}$$
$$T = \{t_1, \ldots, t_i, t_j, \ldots, t_m\}$$
$$T' = T - \{t^j\}$$
$$F = \{f_1, \ldots, (t_i, p_{ij}), (p_{ij}, t_j), \ldots, f_k\}$$
$$F' = F - (\{(t_i, p_{ij}), (p_{ij}, t_j)\} \cup \{(t_j, p_k)/p_k \in t_j\bullet\}) \cup \{(t_i, p_k)/p_k \in t_j\bullet\}$$
$$C = \text{seq}(t_i, t_j, p_{ij}), t_i \in T_{\text{coop}} \text{ and } t_j \in T_{\text{int}}$$
$$Rule_3\text{: } \frac{(P, T, F), C}{(P', T', F')}$$

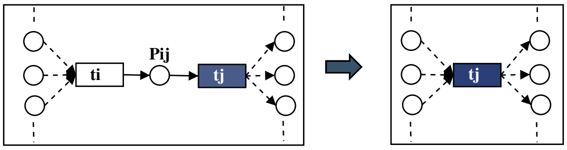


Fig. 20. *Rule*$_2$: compaction of an internal activity followed by a cooperative activity.

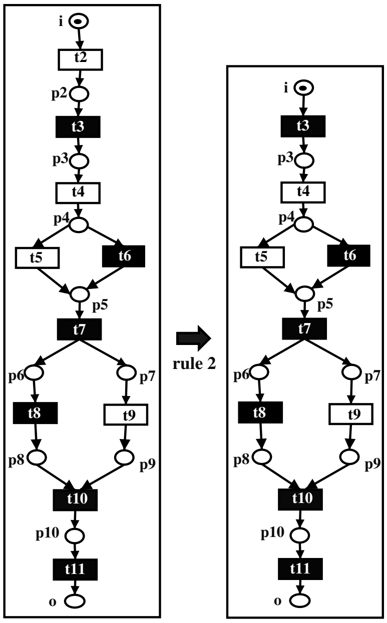


Fig. 21. Result of *rule*$_2$ application.

Here we show another case of sequent activities compaction where we dispose of a cooperative activity t_i followed by an internal one t_j and linked to it via p_{ij}. In this case, we remove the internal activity t_i, the place p_{ij} and all the flows belonging to $\bullet t_j$. Besides, we create new flows linking t_i to all the places belonging to $t_j \bullet$. This scenario is illustrated in Fig. 22.

The application of *rule*$_3$ to t_2, t_3 of Fig. 21 results in removing the internal activity t_4, the place p_3 as well as the flows (t_3,p_3), (p_3, t_4) and (t_4,p_4), and creating the new flow that relies t_3 to p_4 (Fig. 23).

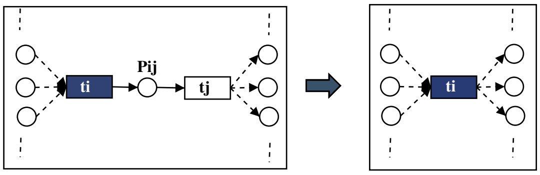


Fig. 22. *Rule*$_3$: compaction of a cooperative activity with an internal activity.

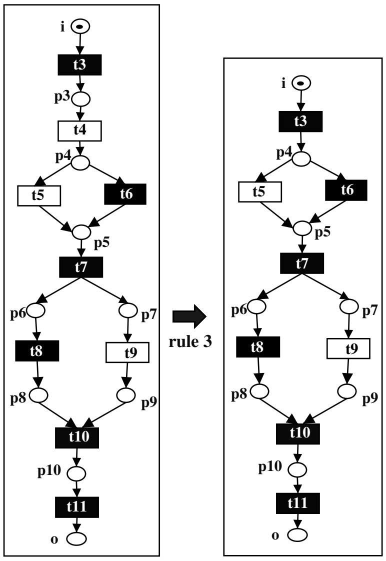


Fig. 23. Result of *rule*$_3$ application.

4. *Rule*$_4$

$\exists t_i \in T_{int}, t_j \in T, p^i_{ij}, p^o_{ij} \in P$ where

$T = \{t_1, \ldots, t_i, t_j, \ldots, t_m\}$

$T' = T - \{t_i\}$

$F = \{f_1, \ldots, (p^i_{ij}, t_i), (t_i, p^o_{ij}), \ldots, f_k\}$

$F' = F - \{(p^i_{ij}, t_i), (t_i, p^o_{ij})\}$

$C = \text{alt}(t_i, t_j, p^i_{ij}, p^o_{ij})$ and $t_i \in T_{\text{int}}$

$$Rule_4\text{:}\ \frac{(P, T, F), C}{(P, T', F')}$$

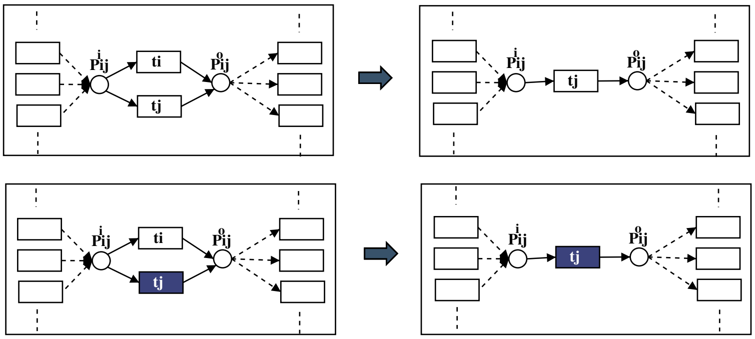


Fig. 24. *Rule*$_4$: compacting alternative activities.

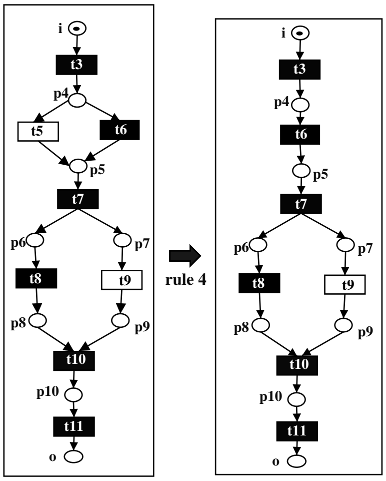


Fig. 25. Result of *rule*$_4$ application.

In this rule we show that if we dispose of two alternative activities t_i and t_j where t_i is internal and t_j can either be internal or cooperative, then we remove t_i as well as all the flows that are directly linked to it. This case is illustrated in Fig. 24.

Rule$_4$ can be applied to the activities t_5 and t_6 of Fig. 23 that fulfills all its conditions and as a result we eliminate t_5 and the flows (p_4, t_5) and (t_5, p_5). The obtained workflow is shown in Fig. 25.

5. *Rule*$_5$

$\exists t_i, t_j \in T, p_{ki}, p_{il}, p_{kj},$
$p_{jl} \in P$ where
$P = \{p_1, \ldots, p_{ki}, p_{il}, \ldots, p_n\}$
$P' = P - \{p_{ki}, p_{il}\}$
$T = \{t_1, \ldots, t_i, t_j, \ldots, t_m\}$
$T' = T - \{t_i\}$
$F = \{f_1, \ldots, (t_k, p_{ki}), (p_{ki}, t_i), (t_i, p_{il}), (p_{il}, t_l), \ldots, f_k\}$
$F' = F - \{(t_k, p_{ki}), (p_{ki}, t_i), (t_i, p_{il}), (p_{il}, t_l)\}$
$C = \text{sync}(t_i, t_j, p_{ki}, p_{il}, p_{kj}, p_{jl})$ and $t_i \in T_{\text{int}}$

Rule$_5$: $\dfrac{(P, T, F), C}{(P', T', F')}$

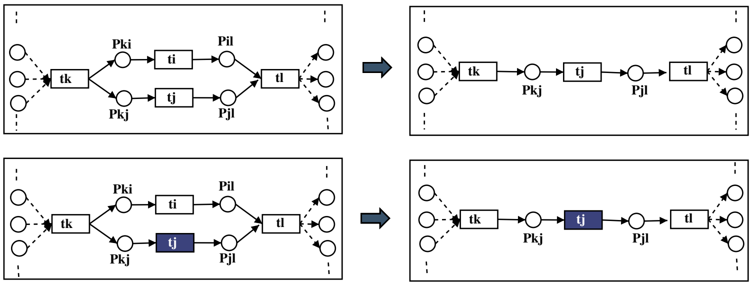


Fig. 26. *Rule*$_5$: compacting parallel activities.

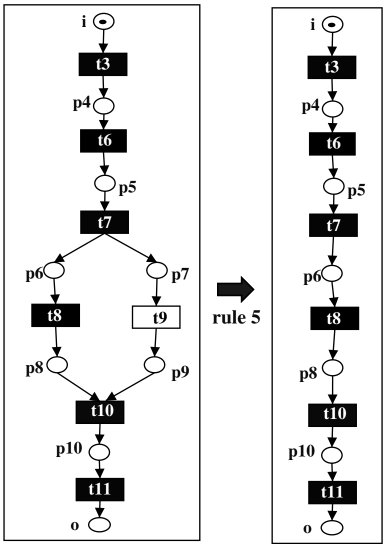


Fig. 27. Result of $rule_5$ application.

The objective of this rule is to remove internal activities in a synchronized flow. Hence, if an internal activity t_i is synchronized with an activity t_j that can be either internal or cooperative, then t_i as well as all the places and flows directly linked to it are removed. This case is illustrated in Fig. 26.

In our example, the activities t_8 and t_9 of Fig. 25 fullfills all the conditions of this rule and applying $rule_5$ results in removing the internal activity t_9, the places p_7 and p_9, and the flows (t_7,p_7), (p_7, t_9), (t_9,p_9) and (p_9, $t_1$0). The obtained workflow is illustrated in Fig. 27. Finally, we notice that none of the five rules could be applied any more. As a consequence, the compacting procedure is

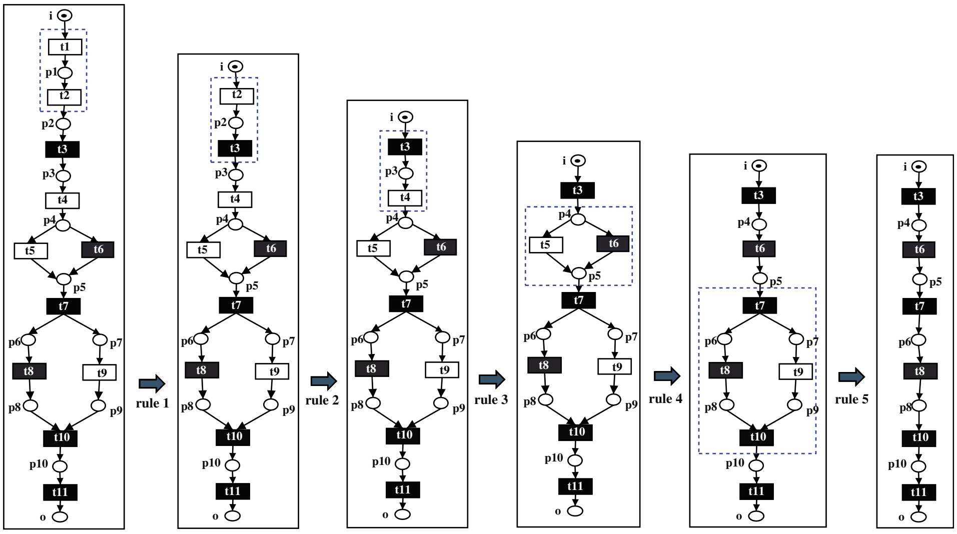


Fig. 28. Workflow compacting.

stopped and the resulting workflow (Fig. 27) is the cooperative workflow corresponding to the internal workflow of Fig. 17.

The resut of the five rules' application on the minimal connected workflow of Fig. 17 is summarized in Fig. 28.

References

[1] W. Davidow, M. Malone, The Virtual Corporation, Harper Business, 1992.

[2] V. Gornev, V. Rarassov, R. Soenen, K. Tahon, Virtual enterprise: reasons, sources and tools, in: MCPL'97, Campinas, Brazil, 1997, pp. 53–58.

[3] J.A. Byrne, The virtual corporation, Business Week 31 (1993) 36–39.

[4] W.M.P. van der Aalst, Loosely coupled interorganizational workflows: modeling and analyzing workflows crossing organizational boundaries, Information and Management 37 (2000) 67–75.

[5] R. Allen, Workflow: an introduction, in: L. Fisher (Ed.), Workflow Handbook, Future Strategies, Lighthouse Point, FL, 2001, pp. 15–38.

[6] P. Grefen, K. Aberer, Y. Hoffner, H. Ludwig, Crossflow: cross-organizational workflow management in dynamic virtual enterprises, International Journal of Computer Systems Science & Engineering 15 (2000) 277–290.

[7] A.V. Dijk, Contracting workflows and protocol patterns, in: Proceedings BPM, Eindhoven, The Netherlands, 2003.

[8] G. Alonso, U. Fiedler, C. Hagen, A. Lazcano, H. Schuldt, N. Weiler, Wise: business to business e-commerce, in: Proceedings of 9th International Workshop on Research Issues on Data Engineering: Information Technology for Virtual Enterprises, 1999.

[9] A. Lazcano, G. Alonso, H. Schuldt, C. Schuler, The wise approach to electronic commerce, International Journal of Computer Systems Science & Engineering 15 (2000), Special issue on Flexible Workflow Technology Driving the Networked Economy.

[10] O. Perrin, C. Godart, A model to support collaborative work in virtual enterprises, Data & Knowledge Engineering 50 (2004) 63–86.

[11] T. Andrews, F. Curbera, H. Dholakia, Y. Goland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic, S. Weerawarana, Business process execution language for web services, version 1.1, 2003.

[12] W.M.P. van der Aalst, Process-oriented architectures for electronic commerce and interorganizational workflow, Information Systems 24 (1999) 639–671.

[13] P. Muth, D. Wodtke, J. Weissenfels, G.W. Angelika Kotz Dittrich, From centralized workflow specification to distributed workflow execution, Journal of Intelligent Information Systems 10 (2) (1998) 159–184.

[14] W.M.P. van der Aalst, Interorganizational workflows: an approach based on message sequence charts and petri nets, Systems Analysis-Modelling-Simulation 34 (1999) 335–367.

[15] W.M.P. vanderAalst, M. Weske, The p2p approach to interorganizational workflows, in: Proceedings of the13th International Conference on Advanced Information Systems Engineering, Springer-Verlag, 2001, pp. 140–156.

[16] J.L. Zhao, Workflow management in the age of e-business, in: Tutorial at the 35th Hawaii International Conference on System Sciences, Waikoloa, Hawaii, 2002.

[17] F. Griffel, M. Boger, H. Weinreich, W. Lamersdorf, M. Merz, Electronic contracting with cosmos—how to establish, negotiate and execute electronic contracts on the internet, in: C. Kobryn, Z.M. Atkinson (Eds.), 2nd International Enterprise Distributed Object Computing Workshop (EDOC '98), 1998, p. 10.

[18] E. Verharen, M. Papazoglou, Introduction to contracting in distributed transactional workflow, in: 31st Annual Hawaii International Conference on Sys. Sci., 1998.

[19] P. Grefen, K. Aberer, Y. Hoffer, H. Ludwig, Crossflow: Cross-organizational workflow management for service outsourcing in dynamic virtual enterprises, IEEE Data Engineering Bulletin 24 (2001) 52–57.

[20] E. Christensen, F. Curbera, G. Meredith, S. Weerawarana, Web services description language (wsdl) version 1.1. Technical report, W3C Note, 2001.

[21] F. Leymann, Web services flow language (wsfl 1.0). Technical report, IBM, 2001 (cf. http://www-4.ibm.com/software/solutions/webservices/pdf/WSFL.pdf).

[22] S. Thatte, Xlang-web services for business process design, 2001.

[23] A. Arkin, Business process modeling language (bpml), Working draft 0.4, BPMI, 2001 (cf. http://www.bpmi.org).
[24] A. Banerji et al., Web services conversation language (wscl) 1.0, 2002.
[25] ebXML: http://ebxml.org, 2002.
[26] Workflow Management Coalition: Interface 1: Process definition interchange-process model. doc. no. wfmc tc 1016-p, version 1.1 (official release), Technical report, 1999.
[27] K. Gerhard, W. Retschitzegger, M. Bernauer, G. Kappel, Specification of interorganizational workflows—a comparison of approaches. Technical Report 08/02, Institute of Software Technology and Interactive Systems, Business Informatics Group, Vienna University of Technology, 2002.
[28] N. Russell, A. ter Hofstede, D. Edmond, W. van der Aalst, Workflow data patterns. Technical Report FIT-TR-2004-01, Queensland University of Technology, Brisbane, 2004.
[29] M. Mers, B. Liberman, K. Muller-Jones, W. Lamersdorf, Interorganisational workflow management with mobile agents in cosm, in: Proceedings of PAAM96 Conference on the Practical Application of Agents and Multiagent Systems, 1996.
[30] S. Tata, Policies for cooperative virtual teams, in: International Conference on Coordination Languages and Models, COORDINATION'02, York, UK, 2002.
[31] K. Baina, S. Tata, K. Benali, A model for process service interaction, in: W.M.P. van der Aalst, A. ter Hofstede, M. Weske (Eds.), Business Process Management: International Conference, LNCS 2678, Eindhoven, The Netherlands, Springer-Verlag Heidelberg, 2003, pp. 261–275.
[32] S. Tata, I. Boughzala, Modeling contribution for virtual enterprise support, in: 12th IEEE International Workshops on Enabling Technologies, Linz, Austria, IEEE Computer Society, 2003, pp. 165–170.
[33] R. Kauffman, E. Walden, Economics and electronic commerce: survey and research directions, 2001.
[34] S. Ba, M. Stinchcombe, A.B. Whinston, H. Zhang, Trusted third parties in the electronic marketplace, in: D. Haseman, D. Nazareth (Eds.), Proceedings of the Fifth Americas Conference on Information Systems, Milwaukee, WI, 1999.
[35] A. Wombacher, K. Aberer, Requirements for workflow modeling in p2p-workflows derived from collaboration establishment, in: DEXA Workshops, 2004, pp. 1036–1041.
[36] A. Wombacher, Overview on decentralized establishment of multi-lateral collaborations, in: From Integrated Publication and Information Systems to Virtual Information and Knowledge Environments, 2005, pp. 11–20.

Issam Chebbi is currently working toward the Phd degree at *Institut National des Telecommunications* in France (INT). His research interests include interoperability within inter-organizational workflows, workflows description, discovery and interconnection.

Schahram Dustdar is an Associate Professor at the Distributed Systems Group, Vienna University of Technology. His previous work experience includes several years as the founding head of the Center for Informatics (ZID) at the University of Art and Industrial Design in Linz (1991–1999) and as the director of Coordination Technologies at the Design Transfer Center in Linz (1999–2000). While on sabbatical leave he was a post-doctoral research scholar at the London School of Economics (Information Systems Department) (1993 and 1994), and a visiting research scientist at NTT Multimedia Communications Labs in Palo Alto, USA during 1998. Since 1999 he works as the co-founder and chief scientist of Caramba Labs Software AG (CarambaLabs.com) in Vienna, a venture capital co-funded software company focused on software for collaborative processes in teams. More information can be found on: http://www.infosys.tuwien.ac.at/Staff/sd.

Samir Tata is an Associate Professor at the Institut National des Télécommunications, Evry France. He was an Research and Teaching Assistant in the University of Nancy I and in the ECOO team of the LORIA/INRIA research laboratory. He previously was a postdoctoral researcher at the Aalborg University Esbjerg in Denmark. His current research area is Computer-Supported Cooperative Work and Workflow with emphasis in description and support of process interaction in virtual enterprises. More information can be found on: http://www-inf.int-evry.fr/tata.