

## Steering dynamic collaborations between business processes

**Citation for published version (APA):**

Zhao, X., & Liu, C. (2010). Steering dynamic collaborations between business processes. *IEEE Transactions on Systems, Man and Cybernetics. Part A, Systems and Humans*, 40(4), 743-757.  
<https://doi.org/10.1109/TSMCA.2010.2044409>

**DOI:**

[10.1109/TSMCA.2010.2044409](https://doi.org/10.1109/TSMCA.2010.2044409)

**Document status and date:**

Published: 01/01/2010

**Document Version:**

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

**Please check the document version of this publication:**

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

[www.tue.nl/taverne](http://www.tue.nl/taverne)

**Take down policy**

If you believe that this document breaches copyright please contact us at:

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.

# Steering Dynamic Collaborations Between Business Processes

Xiaohui Zhao and Chengfei Liu

**Abstract**—Under the background of business globalization nowadays, many organizations are connecting their business processes into complex collaborative business processes to facilitate business collaboration. To adapt to the changing requirements and market opportunities, collaborative business processes have to evolve all the time. Such dynamics brings challenges to the modeling and tracking of collaborative business processes. In addition, these issues can be further complicated by the requirements of privacy protection and information openness in the interorganizational context. Aiming to tackle these problems, this paper proposes a comprehensive framework to model and track dynamic collaborative business processes with our extended relative workflow model. This framework helps each participating organization derive its collaborative business processes and update the processes at run time. A process visibility control mechanism is developed to enforce such derivation and extend organizations' process perception via process visibility transitivity. The whole framework is formalized with matrices and is theoretically proven to be privacy safe. Corresponding algorithms are developed for generating and tracking collaborative business processes. A prototype is also implemented for the proof-of-concept purpose.

**Index Terms**—H.4.1.g workflow management, J.7.e process control.

## I. INTRODUCTION

**I**N THE current business globalization setting, organizations merge their business processes together into an interconnected network for sharing each other's speciality and seeking business synergy. In this network, organizations can respond to customer requirements and market opportunities faster and provide better service [1]. As this network is subject to the dynamic supply/demand relations between organizations, the participating organizations may join or leave this network at any time. This makes the network constitution and the underlying collaborative business processes highly dynamic [2]–[6]. Such complex collaboration requires organizations to manage not only their own business processes but also the collaboration with partners' business processes. In addition, the diverse authorities and relationships between organizations result in different views of the same business process [7], [8], and

therefore, a good process visibility mechanism is needed to prevent authority and privacy violations. These requirements bring challenges to collaborative business process modeling, execution, and tracking of [9]–[13].

Most traditional workflow monitoring approaches and standards, e.g., Workflow Management Coalition (WfMC) Monitor and Audit specifications [14], [15], adhere to a fixed process/workflow model, disregarding the requirements for workflow flexibility or privacy. Most commercial workflow products, such as BEA Weblogic Integration [16] and IBM WebSphere MQ Workflow [17], have no visibility control at all. Some recent research works, e.g., agent-based workflow monitoring [18] and customizable workflow monitoring [19], attempt to improve the flexibility of workflow tracking by increasing user customizability. These works merely support the workflow tracking for process users with different authority levels, but fail to characterize the run-time dynamics of collaborative business processes.

With the aim of supporting the process flexibility and privacy protection, this paper proposes a comprehensive framework for modeling and tracking collaborative business processes. In particular, this framework adopts an organization-oriented perspective to specify the diverse process visibilities of different organizations and establish a process visibility control mechanism. A set of representation matrices and matrix operations are developed to enhance the precision and automation of the process modeling and tracking. In summary, this framework contributes to current collaborative business process management in the following aspects.

- 1) Privacy protection. Our framework allows each organization to restrict the perception level of its business processes to its partner organizations.
- 2) Scalable and extensible process structure. The transitivity of process visibility enables organizations to observe the business processes of nonneighboring organizations. Also, a collaborative business process is updated dynamically as it evolves, e.g., whenever an organization interacts with a new partner, the partner's involved business process(es) will be updated into the collaborative business process.
- 3) Flexible tracking. According to the changing structure of an evolving collaborative business process, the process tracking dynamically detects the instances of newly involved business processes.

The rest of this paper is organized as follows. Section II reviews the related work on process modeling and tracking and classical workflow/process view approaches. Section III discusses the requirements for the modeling and tracking of

Manuscript received February 10, 2009; revised May 13, 2009. Date of publication April 26, 2010; date of current version June 16, 2010. This paper was recommended by Associate Editor M. Mora.

X. Zhao is with the Information Systems Group, Department of Industrial Engineering and Innovation Sciences, Eindhoven University of Technology, 5600 Eindhoven, The Netherlands (e-mail: x.zhao@tue.nl).

C. Liu is with the Centre for Complex Software Systems and Services, Faculty of Information and Communication Technologies, Swinburne University of Technology, Melbourne, Vic. 3122, Australia (e-mail: cliu@groupwise.swin.edu.au).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSMCA.2010.2044409

collaborative business processes with a motivating example. Section IV presents the framework for steering collaborative business processes with the extended relative workflow model, its operational mechanism, and the instance correspondence issue in tracking. Section V introduces a matrix-based formalization for the proposed framework and the related algorithms for generating and tracking collaborative business processes. A Web-service-based prototype implementation is given in Section VI. Section VII discusses the advantages and limitations of the proposed framework. Finally, Section VIII concludes this paper and outlines the direction of future work.

## II. RELATED WORKS

### A. Process Modeling and Tracking, From Intraorganizational to Interorganizational

In the 1990s, the WfMC established a series of specifications for workflow management. Among them, the Interface 5 specification [15] formally defined the functionalities of workflow tracking and logging for the first time, which was regarded as a milestone of workflow tracking technology development. Subsequently, many commercial workflow management systems emerged, such as Action Technology's Action Workflow [20], TIBCO's InConcert [21], DEC's Object Flow, Hewlett Packard's Process Manager etc. Most of these workflow management systems integrated the tracking function and used it to monitor the execution status of workflow instances and evaluate their performance. However, these products mainly focused on intraorganizational applications, and their tracking function was limited to local workflow instances.

When entering the business globalization era, business process management evolved to support interorganizational business processes. IBM MQSeries Workflow and its successor WebSphere MQ Workflow [24] were two typical products for interorganizational workflow management, which both deployed message queues to enable the communication between collaborating organizations. As the main component for process tracking, WebSphere Business Integration Monitor provided a series of tools for workflow simulation and performance analysis. Studio Workflow Monitoring of BEA WebLogic Integration [25] supported users to view the execution status and interrupt the execution of workflow instances. Microsoft's first workflow product, Windows Workflow Foundation [26], supported interorganizational workflows by means of Web service invocations and used the Workflow Tracking Service Database to provide SQL tracking services, which enabled users to view tracking information about workflows and their associated activities.

Most of these workflow management systems concentrated on the communication between heterogeneous systems, while workflow tracking was simply developed as a viewing tool or an analysis tool to handle fixed workflows. Many important issues, such as privacy control, flexible modification, scalability, etc., were neglected or insufficiently considered by these products [27].

The mentioned limitations were mainly caused by the "public-view" nature of these products. In this public-view perspective, all organizations see the same content of a collaborative business process. In addition, a collaborative busi-

ness process is often defined by a third-party designer or a main contractor (focal organization) of a virtual organization alliance. This public-view modeling scheme works well with the assumption that such third-party designer or main contractor exists and that it can see certain details of all participating organizations. However, this assumption clearly does not apply in many practical cases. The public view neutralizes the diverse partnerships and authorities of participating organizations [28]. Consequently, the public-view perspective fails to distinguish the process perceptions of different organizations and lacks effective mechanisms for protecting organizations' business privacy. In this perspective, each participating organization follows the same prefixed collaborative business process to conduct the collaboration. Due to the consistency requirements [29], [30], the prefixed process is hard to change, and consequently, the public-view modeling scheme has a difficulty in capturing the dynamics of collaborative business processes [31] and supporting the collaboration flexibility.

### B. Workflow View Research

To support workflow privacy and interoperability, many researchers attempted to apply the workflow/process-view technology in the interorganizational collaboration environment. A workflow view carries a partial representation of the real workflow process and thereby enables the process visibility control for workflow representation. Schulz and Orłowska [32] proposed coalition workflows to combine private workflows and workflow views together for improving workflow interoperability. Issam *et al.* [33] extracted an abstract workflow view to describe the choreography of a collaboration scenario and compose individual workflows into a collaborative business process. Preuner and Schrefl [34] investigated the generation of collaborative business processes from the perspective of service composition, where services were modeled as complex processes. Their approach distinguished the observable and invocable activities and thereby improved the interoperability granularity in collaborative business process tracking and execution.

In regard to the structural consistency during process view transformations, Liu and Shen [35] proposed an order-preserving approach for deriving a structurally consistent process view from a base process. In their approach, the generation of "virtual activities" (compound tasks) followed their proposed membership, atomicity, and order-preservation rules. Recently, Eshuis and Grefen [36] formalized the operations of task aggregation and process customization, and they also proposed a series of construction rules for validating the structural consistency.

We also established a "relative workflow" approach [29], [30], which modeled collaborative business processes from the perspective of each individual organization. Compared with other workflow/process-view approaches, our relative workflow approach looked into the diversity of organizations' process perceptions and their purposes, such as for process interaction and tracking, and explained the perception derivation from commercial contracts. Therefore, we built up our framework for steering collaborative business processes on the foundation of our relative workflow work.

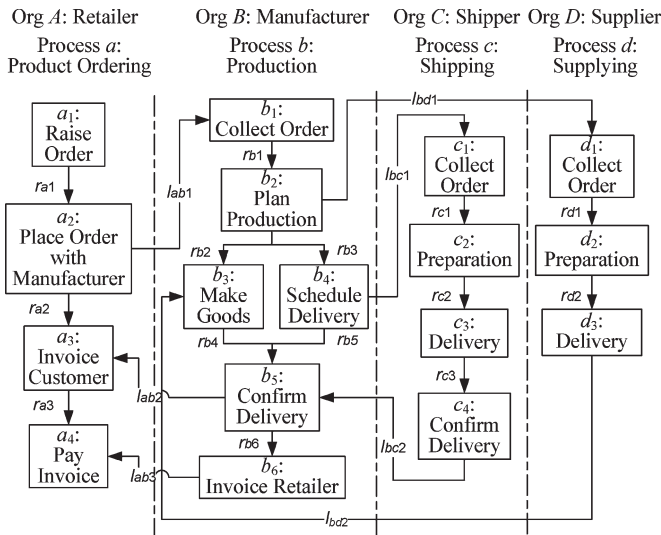


Fig. 1. Collaboration scenario where a retailer collects orders from customers and then purchases products from a manufacturer. The manufacturer may contact a shipper for delivering the ordered products and make goods with the necessary supplies from a supplier.

### III. REQUIREMENT ANALYSIS WITH MOTIVATING EXAMPLE

Fig. 1 shows a collaboration scenario, where a retailer creates and updates its collaborative business process as follows: At first, the collaborative business process has only the product ordering process. By placing an order with a manufacturer, the retailer initiates the collaboration with the manufacturer, and thereafter, the manufacturer’s production process is included into the retailer’s collaborative business process. From then on, the retailer can contact the manufacturer and inquire about the execution status of the production process by referring to some identifiers, e.g., the order number. After the manufacturer arranges with a shipper for the product shipping, the shipper is involved in the collaboration. Thus, the shipper’s shipping process is included into the retailer’s collaborative business process. The retailer can also contact the shipper via the manufacturer and inquire about the shipping status. This progressive updating procedure reflects the dynamic growth of the retailer’s collaborative business process. However, the retailer may not know anything about the supplying process, as the collaboration between the manufacturer and the supplier may be confidential to the retailer.

For the manufacturer, it may build its collaborative business process differently. Aside from the retailer’s product ordering process and the supplier’s supplying process, the manufacturer can also include the shipper’s shipping process after ordering supplies from the supplier. Similarly, the shipper and the supplier may create their own collaborative business processes, respectively.

From the aforementioned discussion, we can summarize the following features of dynamic collaboration.

- 1) For the same collaboration, different organizations may build different collaborative business processes. Thus, the tracking should be varied, too.

- 2) The structure of a collaborative business process may expand on the fly according to the organization’s dynamic process perception.
- 3) An organization may track the execution status of some business processes, which are included in its collaborative business process while actually belonging to partner organizations. Such tracking is conducted through the inquiry/reply communication between the host organization and the partner organization.

These features indicate that the modeling and tracking of collaborative business processes are organization-dependent and structurally dynamic. However, due to the limitations of the public-view nature, these features cannot be fully supported by traditional process approaches [37]–[41]. Aiming to tackle these issues, this paper proposes a novel framework for collaborative business process modeling and tracking. This framework inherits the process visibility control mechanism from our *relative workflow model*[29] and extends it with dynamic modeling and tracking supports. The research reported in this paper is based on a preliminary version of our workflow tracking work [42], with significant improvement and extension on model formalization, system design, and analysis.

### IV. FRAMEWORK FOR STEERING DYNAMIC COLLABORATIONS

#### A. Assumptions and Objectives

Our view is that different organizations may see different pictures of the same collaborative business process and may need to know and be only allowed to know certain details of the collaboration. With this belief, we have proposed our *relative workflow model* in [29]. This model treats a collaborative business process as an aggregation of several business processes belonging to different organizations plus the interactions between these processes.

Based on this model, our framework aims to support the dynamic collaboration with the following conditions.

- 1) Peer-to-peer collaboration. This allows organizations to find and choose collaboration partners by themselves rather than through brokers, thus supporting collaboration flexibility.
- 2) Process perception transitivity. This enables organizations to perceive the processes of partners and even partners of partners under the restriction of privacy protection.
- 3) Process tracking across organizational boundaries. With the process perception transitivity, organizations can check the execution progress of their own processes and their perceived processes from partners.

To support the steering of dynamic collaborations with the aforementioned conditions, we detail the objectives of our framework as follows:

- 1) emphasize the autonomy of participating organizations in the peer-to-peer collaboration and support organizations to join or leave the collaboration in a flexible manner;
- 2) devise a visibility control mechanism to restrict the process perceptions of different organizations and thereby protect the information privacy;



- 3) support the process perception of nonneighboring processes to broaden the collaborative business process tracking;
- 4) provide related operations to generate and track collaborative business processes;
- 5) formalize the related notions and operations to consolidate the proposed framework.

The following sections present the framework in detail.

### B. Extended Relative Workflow Model

In the relative workflow model, a participating *organization* can wrap its *local processes* into a series of *perceivable processes* for different partner organizations, according to the *visibility constraints* defined in the corresponding *perceptions*. A *relative process* is generated by linking an organization's local processes with the perceivable processes of its partner organizations. Different from public-view approaches, a relative process restricts an organization's interactions within the scope of its local processes and perceivable processes of partner organizations to prevent excessive information disclosure. By creating a relative process for each partner organization, a macroview business collaboration process can be distributed into several interactions among neighboring organizations, where each participating organization acts as an autonomous entity with the total control of its local processes. Some key definitions are given next.

**Definition 1—Local Process:** A local process  $lp$  represents the structure of a workflow process.  $lp$  can be defined as a directed graph  $(\mathcal{T}, \mathcal{R})$ , where  $\mathcal{T}$  is the set of nodes (representing tasks) and  $\mathcal{R} \subseteq \mathcal{T} \times \mathcal{T}$  is the set of arcs (representing the execution sequence between tasks).

**Definition 2—Organization:** An organization  $g$  is the host of a set of local processes  $\{lp^1, lp^2, \dots, lp^m\}$ . An individual local process  $lp^i$  of  $g$  is denoted as  $g.lp^i$ ,  $1 \leq i \leq m$ , where  $m$  is the number of  $g$ 's local processes.

According to the two most important behaviors in the context of collaborative business processes, i.e., process interaction and process tracking, we define the following three values for the visibility of tasks as listed in Table I.

**Definition 3—Visibility Constraint:** Visibility constraints specify the visibility values of the tasks belonging to a local process. A visibility constraint  $vc$  is defined as tuple  $(t, v)$ , where  $t$  denotes a task and  $v \in \{\text{Invisible}, \text{Trackable}, \text{Contactable}\}$ . A set of visibility constraints  $\mathcal{VC}$  defined on a business process  $lp$  is represented as set  $\{vc : (t, v) | \forall t (t \in lp.\mathcal{T})\}$ .

**Example 1:** Based on the aforementioned motivating example, a set of visibility constraints is given as follows:

$$\begin{aligned} \mathcal{VC} = \{ & (\text{'Collect Order'}, \text{Contactable}), \\ & (\text{'Plan Production'}, \text{Invisible}), \\ & (\text{'Make Goods'}, \text{Trackable}), \\ & (\text{'Schedule Delivery'}, \text{Trackable}), \\ & (\text{'Confirm Delivery'}, \text{Contactable}), \\ & (\text{'Invoice Retailer'}, \text{Contactable}) \}. \end{aligned}$$

TABLE I  
VISIBILITY VALUES

Visibility value	Explanation
Invisible	A task is said invisible to an external organisation, if it is hidden from that organisation.
Trackable	A task is said trackable to an external organisation, if that organisation is allowed to trace the execution status of the task.
Contactable	A task is said contactable to an external organisation, if the task is trackable to that organisation and the task is also allowed to send/receive messages to/from that organisation for the purpose of business collaboration.

*Due to the high diversity of business collaborations, these three values may hardly cover all visibility scenarios. In this paper, we use these three values to create a fundamental visibility control mechanism, and this visibility value table is open for future extension.*

This set is defined on the manufacturer's 'Production' process from the retailer's view.

**Definition 4—Perception:** A perception  $p_{g_0}^{g_1.lp}$  contains the visibility constraints and message descriptions that are defined for local process  $g_1.lp$  from another organization  $g_0$ . Perception  $p_{g_0}^{g_1.lp}$  can be defined as tuple  $(\mathcal{VC}, \mathcal{MD}, f)$ , where the following are defined.

- 1)  $\mathcal{VC}$  is set of visibility constraints defined on  $g_1.lp$ .
- 2)  $\mathcal{MD} \subseteq \mathcal{M} \times \{\text{in}, \text{out}\}$  is set of the message descriptions that contains the messages and the passing directions, where  $\mathcal{M}$  is the set of messages.
- 3)  $f : \mathcal{MD} \rightarrow g_1.lp_{g_0}.\mathcal{T}$  is mapping from  $\mathcal{MD}$  to  $g_1.lp_{g_0}.\mathcal{T}$ , where  $g_1.lp_{g_0}$  is the *perceivable process* of  $g_1.lp$  from  $g_0$ . Here, a perceivable process represents the perceivable form of a local process for a partner organization. The generation of  $g_1.lp_{g_0}$  from  $g_1.lp$  will be discussed in the next section.

**Example 2:** Based on the aforementioned motivating example, the perception of the manufacturer's 'Production'(tm) process from the retailer is given as follows:

$$\begin{aligned} p_{\text{retailer}}^{\text{Manufacturer.productionProcess}} &= (\mathcal{VC}, \{(\text{'Order of Products'}, \text{in}), \\ & (\text{'Confirmation of Delivery Date'}, \text{out}), \\ & (\text{'Invoice'}, \text{out})\}, \\ & \{(\text{'Order of Products'}, \text{in}) \rightarrow \text{'Collect Order'}, \\ & (\text{'Confirmation of Delivery Date'}, \text{out}) \\ & \rightarrow \text{'Confirm Delivery'}, (\text{'Invoice'}, \text{out}) \\ & \rightarrow \text{'Invoice Retailer'}\}), \end{aligned}$$

where  $\mathcal{VC}$  is defined in Example 1.

**Definition 5—Relative Process:** A relative process denotes an organization  $g_0$ 's view of a business collaboration, which includes  $g_0$ 's local process(es) and the involved perceivable process(es) of  $g_0$ 's partners. The structure of relative process  $rp$

perceivable from organization  $g_0$  is defined as a directed graph  $(\mathcal{T}, \mathcal{R})$ , where the following are described.

- 1)  $\mathcal{T}$  is the set of the tasks perceivable from  $g_0$ , which includes the following two unions:
  - a)  $\cup_k g_0.lp^k.\mathcal{T}$ , the union of the task sets of all  $g_0.lp^k$  ( $1 \leq k \leq m_0$ , and  $m_0$  is the number of  $g_0$ 's involved local processes);
  - b)  $\cup_{i,j} g_i.lp^j_{g_0}.\mathcal{T}$ , the union of the task sets of perceivable processes of all  $g_i.lp^j$  from  $g_0$  ( $1 \leq i \leq n$  and  $1 \leq j \leq m_i$ , while  $n$  is the number of  $g_0$ 's partner organizations and  $m_i$  is the number of  $g_i$ 's involved perceivable processes for  $g_0$ ).
- 2)  $\mathcal{R}$  is the set of arcs perceivable from  $g_0$ , which is a union of the following four parts, where  $i, j$ , and  $k$  are defined the same as in  $\mathcal{T}$ :
  - a)  $\cup_k g_0.lp^k.\mathcal{R}$ , the union of the arc sets of all  $g_0.lp^k$ ;
  - b)  $\cup_{i,j} g_i.lp^j_{g_0}.\mathcal{R}$ , the union of the arc sets of perceivable processes of all  $g_i.lp^j$  from  $g_0$ ;
  - c)  $\mathcal{L}_{intra}$ , the set of intraorganizational messaging links that connect tasks belonging to different local processes, which is defined in  $\cup_{i,j}(g_0.lp^i.\mathcal{T} \times g_0.lp^j.\mathcal{T})$ , where  $i \neq j$ ;
  - d)  $\mathcal{L}_{inter}$ , the set of interorganizational messaging links that connect tasks between a local and a perceivable process, which is defined in  $\cup_{i,j,k}(g_0.lp^k.\mathcal{T} \times g_i.lp^j_{g_0}.\mathcal{T} \cup g_i.lp^j_{g_0}.\mathcal{T} \times g_0.lp^k.\mathcal{T})$ .

Besides the involved local processes, a relative process can only contain the perceivable processes that have messaging links with the host organization's local process(es), i.e., the perceivable processes with "contactable" tasks to the host organization. To support interorganizational process tracking, we extend the relative workflow model to cover proper trackable perceivable processes of nonneighboring organizations. To help determine whether a business process is trackable, we first give two important definitions.

**Definition 6—Visible Messaging Link:** A messaging link  $l$  connecting two perceivable processes is said *visible* to organization  $g$  if and only if the two tasks connected by  $l$  are both visible (contactable or trackable) to  $g$ .

**Definition 7—Reachability:** For organization  $g$ 's relative process  $rp$ , a perceivable process  $pp$  is said *reachable* if  $pp$  has at least one visible messaging link connecting a visible (contactable or trackable) task of  $rp$  and a visible task of  $pp$ .

Via visible messaging links, an organization may extend its relative process by including the perceivable processes of nonneighboring organizations. Formally, we define such an extended relative process as a *collaborative business process*.

**Definition 8—Collaborative Business Process:** A collaborative business process  $cbp$  for organization  $g$ 's relative process  $rp$  represents  $g$ 's view for the business collaboration in which  $rp$  is involved. The structure of  $cbp$  can be defined as a directed graph in the form of tuple  $(\mathcal{T}, \mathcal{R})$ , where the following are described.

- 1) Task set  $\mathcal{T}$  includes the following:
  - a)  $rp.\mathcal{T}$ , the tasks of relative process  $rp$ ;

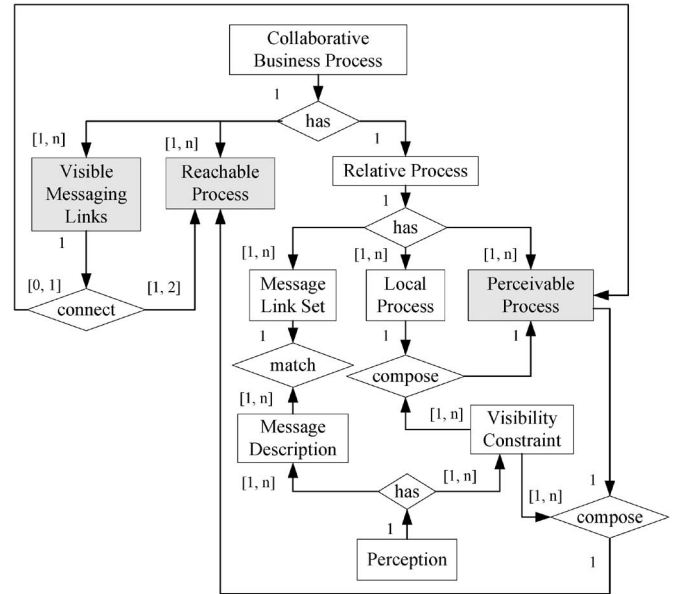


Fig. 2. Relations between the notions in the extended relative workflow model.

- b)  $\cup_{i,j} g_i.lp^j_{g_0}.\mathcal{T}$ , the union of the task sets of reachable perceivable processes of all  $g_i.lp^j$  from  $cbp$  ( $0 \leq i \leq n$  and  $0 \leq j \leq m_i$ , while  $n$  is the number of  $g$ 's nonneighboring organizations and  $m_i$  is the number of  $g_i$ 's perceivable processes reachable from  $cbp$ ).
- 2) Arc set  $\mathcal{R}$  includes the following:
  - a)  $rp.\mathcal{R}$ , the arcs of relative process  $rp$ ;
  - b)  $\cup_{i,j} g_i.lp^j_{g_0}.\mathcal{R}$ , the union of the arc sets of reachable perceivable processes of all  $g_i.lp^j$  from  $cbp$ ;
  - c)  $\mathcal{L}_{per} = \cup_{i,j,k,m} g_i.lp^j_{g_0}.\mathcal{T} \times g_k.lp^m_{g_0}.\mathcal{T}$ , the union of the visible messaging links connecting the perceivable processes in  $cbp$ , where  $\forall l \in \mathcal{L}_{per}$ ,  $l$  is visible from  $g$  ( $0 \leq i, k \leq u$ ,  $0 \leq j, m \leq v_i$ , while  $u$  is the number of  $g$ 's partners (neighboring and nonneighboring) and  $v_i$  is the number of  $g_i$ 's reachable perceivable processes in  $cbp$ ).

From the definition, we can see that a collaborative business process is defined in a recursive manner to capture the run-time dynamics.

Fig. 2 shows the relationship between these notions, where the white and shadowed boxes represent the local and foreign parts of a collaborative business process, respectively.

For each partner organization, the host organization creates a perception to define the visibility constraints and message descriptions for a local process. By hiding "invisible" tasks defined in the perception, an organization can wrap a local process into a perceivable process. By matching the message descriptions defined in proper perceptions, the organization can connect its local processes and the perceivable processes from its partners. These interconnected processes construct the main skeleton of a relative process. Based on such a relative process, a collaborative business process can be derived by including the reachable processes via visible messaging links.

As an organization may set various visibility constraints for different partners, the relative processes generated by different

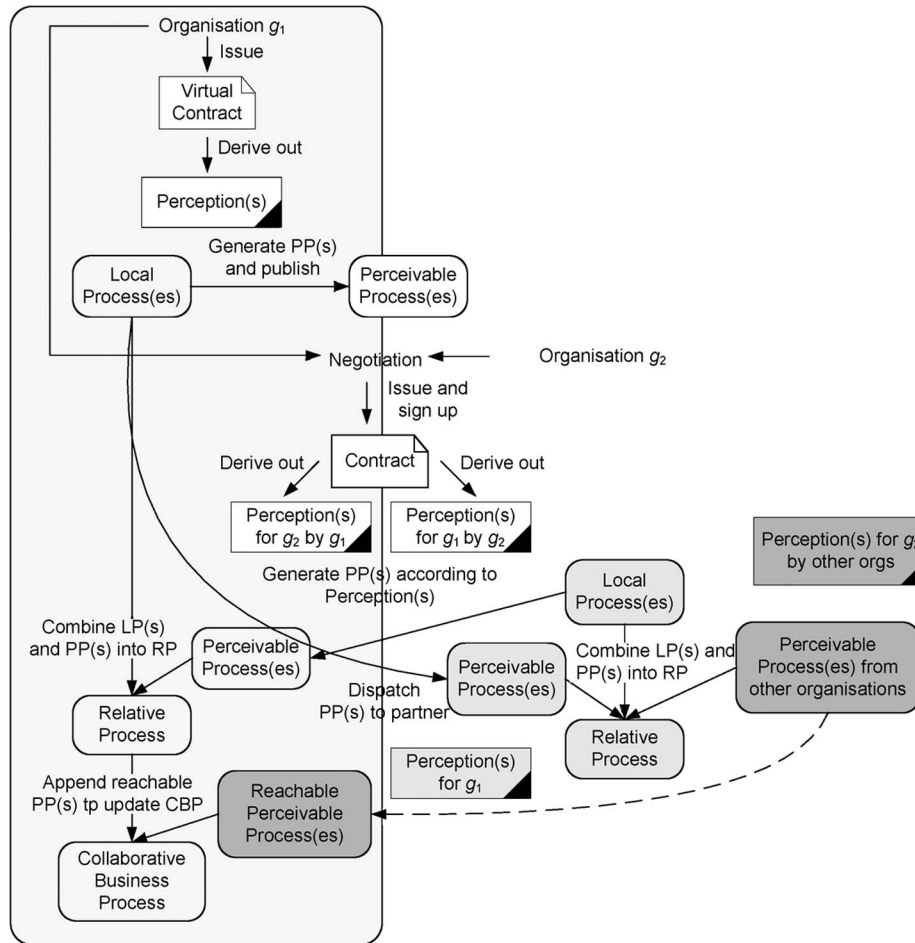


Fig. 3. Illustration of open contracting mechanism.

partners could be different, too. This feature reflects the characteristics of relativity. Once the perceivable processes are published, an organization can choose and assemble proper perceivable and local processes into a relative process. This feature reflects the characteristics of autonomy in the collaboration. A collaborative business process can be extended by including reachable processes as the collaboration proceeds. This feature caters for the collaboration flexibility and scalability.

### C. Operational Mechanism

This extended relative workflow model particularly fits into the peer-to-peer collaboration. Two typical peer-to-peer collaboration examples are virtual organization alliance and business ecosystem. In such collaboration scheme, the partnership is usually decided by means of price matching, bidding, or auctions, and it may terminate as soon as the trading accomplishes. Different from traditional collaborations, organizations in peer-to-peer collaboration mainly follow an open contracting mechanism. In this mechanism, the host organization, for example,  $g_1$ , as shown in Fig. 3, first lists the basic supply/demand requirements in a virtual contract. With this virtual contract,  $g_1$  can derive the corresponding perception(s) and generate the perceivable process(es) for its involved local process(es). These perceivable processes are then published for advertise-

ment. Interested organizations, for example,  $g_2$ , may check  $g_1$ 's published perceivable processes before determining the collaboration. Once the collaboration is determined,  $g_2$  can negotiate with  $g_1$  and sign up a concrete contract.

From the concrete contract,  $g_1$  and  $g_2$  will derive the perceptions for their participated local processes, respectively. For the details of this derivation, please refer to our previous work [29]. Thereafter,  $g_1$  and  $g_2$  will create corresponding perceivable processes and pass them to each other. Upon receiving these perceivable processes from  $g_2$ ,  $g_1$  can connect them with its own local process and create the relative process for the collaboration. Similarly,  $g_2$  can generate its own relative process.

As  $g_2$  may collaborate with other organizations at the same time,  $g_2$ 's relative process may include the perceivable processes from other organizations. Under the visibility constraints of the perceptions by  $g_2$  and other organizations, if some perceivable processes are reachable from  $g_1$ 's relative process, they will be appended into  $g_1$ 's relative process. By continuously appending the found reachable processes,  $g_1$  can generate its collaborative business process.

### D. Running Examples

This section is to further illustrate the mentioned concepts and operations with the motivating example discussed

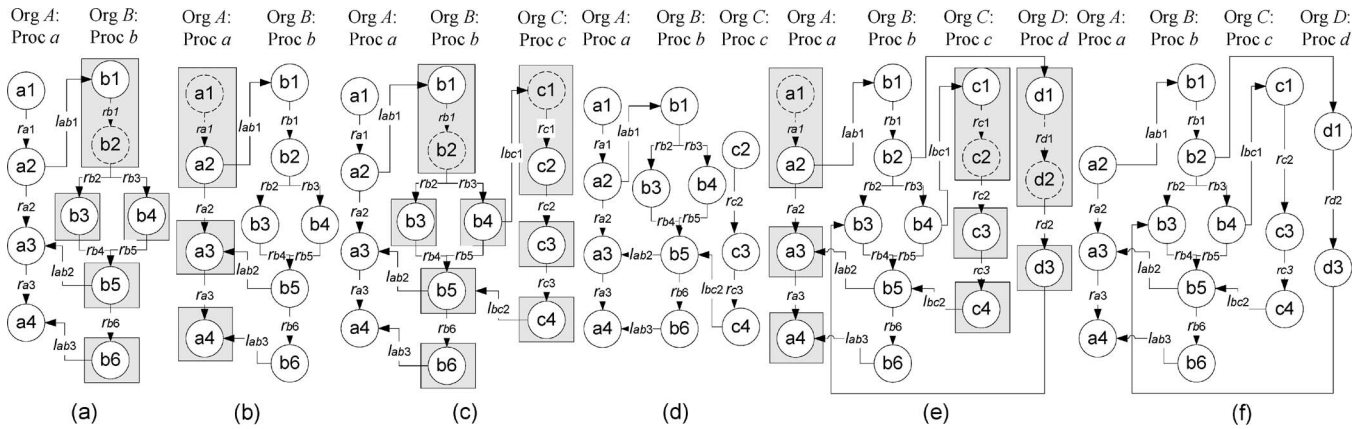


Fig. 4. Evolving relative workflows and collaborative business processes.

in Section III. First, we suppose that the involved perceptions use the following visibility constraints in this collaboration scenario:

$$\begin{aligned}
 &P_{\text{Retailer}}^{\text{Manufacturer.productionProcess}}.VC \\
 &= \{(\text{'Collect Order'}, \text{Contactable}), \\
 &\quad (\text{'Plan Production'}, \text{Invisible}), \\
 &\quad (\text{'Make Goods'}, \text{Trackable}), \\
 &\quad (\text{'Schedule Delivery'}, \text{Trackable}), \\
 &\quad (\text{'Confirm Delivery'}, \text{Contactable}), \\
 &\quad (\text{'Invoice Retailer'}, \text{Contactable})\}.
 \end{aligned}$$

$$\begin{aligned}
 &P_{\text{Manufacturer}}^{\text{Retailer.ProductOdering}}.VC \\
 &= \{(\text{'raise order'}, \text{Invisible}), \\
 &\quad (\text{'place order with manufacturer'}, \text{Contactable}), \\
 &\quad (\text{'invoice customer'}, \text{Contactable}), \\
 &\quad (\text{'pay invoice'}, \text{Contactable})\}
 \end{aligned}$$

$$\begin{aligned}
 &P_{\text{Manufacturer}}^{\text{Shipper.Shipping}}.VC \\
 &= \{(\text{'collect order'}, \text{Contactable}), \\
 &\quad (\text{'preparation'}, \text{Invisible}), \\
 &\quad (\text{'delivery'}, \text{Trackable}), \\
 &\quad (\text{'confirm delivery'}, \text{Contactable})\}
 \end{aligned}$$

$$\begin{aligned}
 &P_{\text{Retailer}}^{\text{Shipper.Shipping}}.VC \\
 &= \{(\text{'collect order'}, \text{Invisible}), \\
 &\quad (\text{'preparation'}, \text{Trackable}), \\
 &\quad (\text{'delivery'}, \text{Trackable}), \\
 &\quad (\text{'confirm delivery'}, \text{Trackable})\}
 \end{aligned}$$

$$\begin{aligned}
 &P_{\text{Manufacturer}}^{\text{Supplier.Supplying}}.VC \\
 &= \{(\text{'collect order'}, \text{Contactable}), \\
 &\quad (\text{'preparation'}, \text{Invisible}), \\
 &\quad (\text{'delivery'}, \text{Contactable})\}.
 \end{aligned}$$

These visibility constraints restrict that each organization can only see a partial view of the collaborative business process.

Since the retailer and the supplier have no partner relationship in the collaborative business process, they do not define any perception for each other. This section will take the retailer and the manufacturer as examples to show how organizations generate relative processes and collaborative business processes, respectively.

Fig. 4(a) and (b) shows the partial views upon the whole collaborative business process from the perspectives of the retailer and the manufacturer, respectively. For simplicity, Fig. 4 renames the organizations, processes, and tasks with letters and numbers.

According to the visibility constraints defined in  $P_{\text{Retailer}}^{\text{Manufacturer.productionProcess}}$ , we can see that task  $b_2$  is set invisible to the retailer by the manufacturer. Therefore, task  $b_2$  will be composed into task  $b_1$ , when generating the perceivable production process for the retailer.

As shown in Fig. 4(a), this perceivable process can be connected to organization  $A$ 's process  $a$ , and thereby,  $A$  can generate a relative process. In Fig. 4(a), the shadowed part denotes the perceivable process and the task circled with a dotted line denotes an invisible task. The technical details of the mentioned composition operation and connection operation will be presented in Section V.

At the site of the manufacturer, the retailer can compose its local process into a perceivable process, as shown in the left part of Fig. 4(b), according to perception  $P_{\text{Manufacturer}}^{\text{Retailer.ProductOdering}}$ . The whole part of Fig. 4(b) shows the manufacturer's relative process at this stage.

Although the shipper has no direct interactions with the retailer, it still sets up the visibility constraints for the retailer in perception  $P_{\text{Retailer}}^{\text{Shipper.Shipping}}$ . This perception enables the reachability of the shipping process from the retailer, and therefore, the retailer can append the shipping process into its collaborative business process. This reachability is also subject to perceptions  $P_{\text{Manufacturer}}^{\text{Shipper.Shipping}}$  and  $P_{\text{Retailer}}^{\text{Manufacturer.productionProcess}}$ . To determine the reachability, we first need to check whether any visible messaging link exists between processes  $b$  and  $c$  from organization  $A$  in Fig. 4(c). For messaging link  $l_{bc1}$ , since task  $c_1$  is set invisible to  $A$  in  $P_{\text{Retailer}}^{\text{Shipper.Shipping}}$ ,  $l_{bc1}$  is invisible to  $A$  according to Definition 6. For  $l_{bc2}$ , it is a visible messaging link from  $A$ , and process  $c$  is therefore reachable from  $A$ . After appending the perceivable process  $c$  and the visible messaging



link  $l_{bc2}$  to its relative process, organization  $A$ 's relative process evolves into the collaborative business process, as shown in Fig. 4(d), according to Definition 8.

As the collaboration proceeds, more organizations and processes may join in, and organization  $A$ 's collaborative business process may grow accordingly. To support such dynamic expansion, the collaborative business process generation runs by continuously detecting reachable processes and propagating the detection procedure. The details of this appending operation and the algorithm of generating collaborative business processes will be presented in Section V.

At the site of the manufacturer, it can include the perceivable shipping and supplying processes, according to perceptions  $p_{\text{Manufacturer}}^{\text{Shipper.Shipping}}$  and  $p_{\text{Manufacturer}}^{\text{Supplier.Supplying}}$ , as shown in Fig. 4(e). Fig. 4(f) shows the final relative process of the manufacturer, where it has no reachable processes.

From this example, we can see that the visibility control mechanism prevents the privacy disclosure. Once a task is set invisible to an organization, the task can never be disclosed to that organization. An organization can only see what they are allowed to see. Even with the process visibility transitivity, organizations cannot obtain excessive process information through intermediate organizations. These features guarantee the privacy protection during collaborations.

### E. Instance Correspondence in Collaborative Business Process Tracking

At the instance level, the tracking over a collaborative business process relies a lot on instance correspondence. This issue results from the phenomenon that multiple instances of the same business process may exist in the same collaboration. Therefore, an instance of a collaborative business process  $p$  may contain several instances of  $p_1$ , several instances of  $p_2$ , etc., where  $p_1, p_2, \dots$ , are  $p$ 's constituent processes. Each instance of  $p_1$  may interact with none, one, or multiple instances of  $p_2$ , which results in a complex correspondence relation. For this issue, how to keep the process cardinality and the correspondence between these involved instances at run time is the key problem. More details on instance correspondence can be found in our previous work on the correspondence Petri net [43].

To record the instance correspondence in a collaborative business process, we designed a data structure as shown in Fig. 5. This data structure consists of several lists, each of which stores the instances of a constituent process of the collaborative business process. Each element on the list records the execution status of a task belonging to a process instance. The links between the elements of different lists represent the correlation between these process instances. The algorithms for generating and tracking a collaborative business process will be given in Section V.

## V. METHODOLOGY FORMALIZATION

### A. Representation Matrices

To precisely describe the structure of a business process, we define the following matrices.

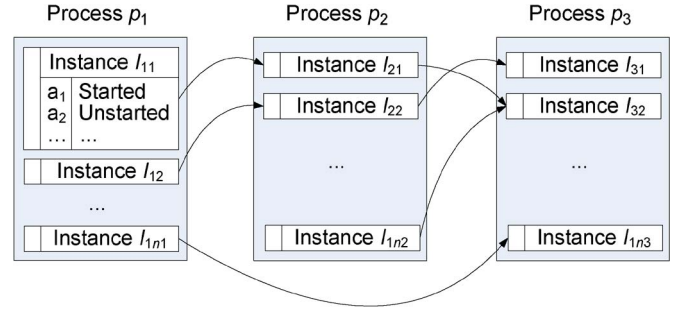


Fig. 5. Data structure for collaborative business process tracking.

**Definition 9—Self-Adjacency Matrix:** A self-adjacency matrix represents the structure of either a local, perceivable, or collaborative business process. For an  $n$ -task business process  $p$  of organization  $g$ , it can be represented as an  $n \times n$  self-adjacency matrix  $D_g^p = [d_{ij}]$

$$d_{ij} = \begin{cases} r, & \text{if there exists arc } r \text{ linking task } t_i \text{ and task } t_j, \\ & \text{where } i < j \\ 0, & \text{otherwise.} \end{cases}$$

Each element of a self-adjacency matrix denotes an arc between tasks, such as  $r_{a1}$  and  $r_{b2}$  in Fig. 1. An arc connecting tasks  $t_i$  and  $t_j$  is put in  $d_{ij}$ , not  $d_{ji}$ , where  $i < j$ . Thus,  $D_g^p$  is always an upper triangular matrix. According to the relative workflow model, a self-adjacency matrix can represent a local, perceivable, or relative process.

**Definition 10—Transformation Matrix:** A transformation matrix represents the procedure of creating a perceivable process by hiding the invisible tasks. For composing a local process  $p$  into a perceivable process for organization  $g$ , this procedure can be represented as an  $n \times n$  triangular 0–1 matrix  $T_g^p = [t_{ij}]$ , where

$$t_{ij} = \begin{cases} 1, & \text{if task } t_j \text{ is composed into task } t_i \text{ (} j \neq i \text{)} \\ & \text{or not composed (} j = i \text{)} \\ 0, & \text{otherwise.} \end{cases}$$

This matrix can be directly derived from the visibility constraints defined in the corresponding perception, following the task composition algorithm discussed in [29]. Note that each column has only one element with value “1,” which indicates that each task can be composed only once at most.

**Definition 11—Boundary Adjacency Matrix:** A boundary adjacency matrix represents the messaging links between two business processes. From the perspective of organization  $g$ , the messaging links between processes  $p_1$  and  $p_2$  can be represented as an  $m \times n$  matrix  $B_g^{p_1|p_2} = [b_{ij}]$ , where

$$\begin{aligned} m & \text{ number of } p_1 \text{'s tasks;} \\ n & \text{ number of } p_2 \text{'s tasks;} \\ b_{ij} & = l \text{ if there exists a messaging link } l \text{ connecting } p_1.t_i \text{ and } p_2.t_j; \text{ otherwise, } b_{ij} = 0. \end{aligned}$$

For a collaborative business process, it can be represented by a composite self-adjacency matrix. For example, suppose organization  $A$  has a collaborative business process containing

three processes  $a$ ,  $b$ , and  $c$ . This collaborative business process can be represented as

$$D_A^{(a|b)|c} = \begin{pmatrix} D_A^a & B_A^{a|b} & B_A^{a|c} \\ 0 & D_A^b & B_A^{b|c} \\ 0 & 0 & D_A^c \end{pmatrix}$$

which contains proper self-adjacency matrices and boundary adjacency matrices inside.

### B. Matrix Operations

To facilitate the generation and tracking of collaborative business processes, we define the following matrix operations.

- **Composition Operation:** In a transformation matrix, each element with value “1” at a nondiagonal position  $(i, j)$  stands for a procedure of composing task  $t_j$  into task  $t_i$ . The composition is subject to the following rules.
  - 1) A link connecting  $t_j$  and  $t_k$  ( $k \neq i$ ) should be changed to connect  $t_i$  and  $t_k$ .
  - 2) A link connecting  $t_i$  and  $t_j$  should be discarded.
  - 3) A link connecting  $t_i$  and  $t_k$  ( $k \neq j$ ) should remain.

The first rule corresponds to a modification operation over the elements of the self-adjacency matrix for the business process. In this modification, the elements on row  $j$  will be added to the corresponding elements on row  $i$ , and afterward, all the elements on row  $j$  will be set zero. In our framework, this modification operation can be formalized as matrix multiplication between the transformation and self-adjacency matrices.

With respect to the second rule, we need to check whether there exists a link connecting  $t_i$  and  $t_j$ , i.e., whether, in the corresponding transformation matrix, there exists a row that has a value “1” at both columns  $i$  and  $j$ . Here, we represent this checking as Boolean expression  $|f_{\text{row}}(i) = f_{\text{row}}(j)|$ , where  $f_{\text{row}}(x)$  defines a function that returns the row where column  $x$  has value “1.”

The third rule identifies the case that needs no action. In this case, we see that the composing tasks all have value “1” on the diagonal line of the transformation matrix, whereby they take no effect in the matrix multiplication.

Considering all these three rules, we formalize the operation that organization  $g_1$  composes process  $p$  for organization  $g_2$  as  $D_{g_2}^{g_1 \cdot p} = f_{\text{reshape}}(T_{g_2}^{g_1 \cdot p} \otimes D_{g_1}^{g_1 \cdot p})$ , where the following are defined.

- 1) Operator  $\otimes$  is defined as

$$T_{n \times n} \otimes D_{n \times n} = \left[ |f_{\text{row}}(i) \neq f_{\text{row}}(j)| \cdot \sum_{x=1}^n t_{ix} \cdot d_{xj} \right]_{n \times n}.$$

- 2) Function  $f_{\text{reshape}}$  is used to adjust the result matrix into an upper triangular form. For input matrix  $M_{n \times n}$ ,  $f_{\text{reshape}}$  is defined as

$$f_{\text{reshape}}(M_{n \times n}) = M_{n \times n}^{\circ} = [m_{ij}^{\circ}], \text{ where } m_{ij}^{\circ} = \begin{cases} m_{ij} + m_{ji}, & i < j \\ 0, & \text{otherwise.} \end{cases}$$

- **Connection Operation:** Suppose organization  $g_3$  has a relative process comprising its local process  $p_3$  and perceivable process  $p_2$  from its neighboring organization  $g_2$ . From  $g_3$ 's perspective, when process  $p_1$  of its nonneighboring organization  $g_1$  is connected to  $p_2$ ,  $g_3$ 's knowledge about the visible messaging links between  $p_1$  and  $p_2$  is subject to perceptions  $p_{g_3}^{g_1 \cdot p_1}$  and  $p_{g_3}^{g_2 \cdot p_2}$ . In the matrix context, we define the operation of connecting  $g_1 \cdot p_1$  and  $g_2 \cdot p_2$  for  $g_3$  as  $B_{g_3}^{g_1 \cdot p_1 | g_2 \cdot p_2} = (f_{\text{diag}}(T_{g_3}^{g_2 \cdot p_2}) \cdot (f_{\text{diag}}(T_{g_3}^{g_1 \cdot p_1}) \cdot B_{g_1}^{g_1 \cdot p_1 | g_2 \cdot p_2})^T)^T$ , where the following are defined.

- 1)  $B_{g_3}^{g_1 \cdot p_1 | g_2 \cdot p_2}$  is the result boundary adjacency matrix that represents the visible messaging links between  $g_1 \cdot p_1$  and  $g_2 \cdot p_2$  from  $g_3$ 's perspective.
- 2)  $T_{g_3}^{g_2 \cdot p_2}$  and  $T_{g_3}^{g_1 \cdot p_1}$  are the transformation matrices for perceptions  $p_{g_3}^{g_1 \cdot p_1}$  and  $p_{g_3}^{g_2 \cdot p_2}$ , respectively.
- 3)  $B_{g_1}^{g_1 \cdot p_1 | g_2 \cdot p_2}$  is the boundary adjacency matrix that represents the visible messaging links between  $g_1 \cdot p_1$  and  $g_2 \cdot p_2$  from  $g_1$ 's perspective.
- 4) Function  $f_{\text{diag}}$  is used to diagonalize a transformation matrix  $T$  into a diagonal matrix  $T^{\circ}$ . Function  $f_{\text{diag}}$  is defined as

$$f_{\text{diag}}(T_{n \times n}) = T_{n \times n}^{\circ}, \quad t_{ij}^{\circ} = \begin{cases} 1, & \text{if } t_{ij} = 1 \text{ and } i = j \\ 0, & \text{otherwise.} \end{cases}$$

In this connection operation,  $g_1$  first diagonalizes  $T_{g_3}^{g_1 \cdot p_1}$  and performs a matrix multiplication on the diagonalized  $T_{g_3}^{g_1 \cdot p_1}$  and  $B_{g_1}^{g_1 \cdot p_1 | g_2 \cdot p_2}$ . Subsequently,  $g_2$  uses the diagonalized matrix  $T_{g_3}^{g_2 \cdot p_2}$  to multiply the result matrix from  $g_1$ . In this procedure, matrices may be transposed to align the columns of the left-hand matrix with the rows of the right-hand matrix for the purpose of matrix multiplication.

- **Expansion Operation:** The expansion operation of a relative process means to append proper boundary adjacency matrices and self-matrices to an existing self-adjacency matrix. Accordingly, we define the operation of expanding process  $p_1$  of organization  $g_1$  by appending perceive process  $p_2$  of organization  $g_2$  as  $D_{g_1}^{g_1 \cdot p_1 | g_2 \cdot p_2} = \begin{pmatrix} D_{g_1}^{g_1 \cdot p_1} & B_{g_1}^{g_1 \cdot p_1 | g_2 \cdot p_2} \\ 0 & D_{g_1}^{g_2 \cdot p_2} \end{pmatrix}$ , where the following are defined.
  - 1)  $D_{g_1}^{g_1 \cdot p_1 | g_2 \cdot p_2}$  is the result self-adjacency matrix that represents the extended process.
  - 2)  $D_{g_1}^{g_1 \cdot p_1}$  and  $D_{g_1}^{g_2 \cdot p_2}$  represent the processes  $p_1$  and  $p_2$ , respectively, from  $g_1$ 's perspective.
  - 3)  $B_{g_1}^{g_1 \cdot p_1 | g_2 \cdot p_2}$  represents the visible messaging links between  $g_1 \cdot p_1$  and  $g_2 \cdot p_2$  from  $g_1$ 's perspective.

### C. Privacy Protection in Generating Collaborative Business Processes

This section is to check the privacy protection property of the proposed operations. As the visibility constraints of tasks are explicitly given in perceptions, it is easy to guarantee the privacy protection at task level. Therefore, we concentrate on the handling of arcs and messaging links during the generation

of collaborative business processes. Technically, this can be interpreted as follows: No arcs connect invisible tasks, and all invisible arcs are hidden in the result process of the composition operation; no messaging links connect invisible tasks, and all invisible messaging links are hidden in the result process of the connection operation.

*Property 1—No Privacy Disclosure in Composition Operation:*

*Proof:* This operation applies to transformation and self-adjacency matrices for handling the arcs between tasks. According to the definition of a transformation matrix, for each visible task  $t_i$ , “1” will be put on position  $(i, i)$  on the diagonal, while for each invisible task  $t_j$ , “1” will be put on position  $(k, j)$ , where task  $t_k$  is visible ( $i, j$ , and  $k$  denote the sequential numbers of tasks of a process). The condition  $|f_{\text{row}}(i) \neq f_{\text{row}}(j)|$  of operator  $\otimes$  in composition operation indicates the removal of self-linking arcs, because in the business process context, such a self-linking arc, i.e., an arc links a task to the task itself, is redundant. (Note that we assume that a meaningful loop must contain at least two tasks, one of which is particularly used for deciding the loop exit condition.)

For the matrix multiplication of the composition operation, we discuss the arc handling in the following five cases.

- 1) *An arc links two visible tasks.* This arc will remain, which is guaranteed by the “1’s” on diagonal positions of the transformation matrix.
- 2) *An arc links a visible task  $t_x$  to an invisible one  $t_y$ .* If  $t_y$  is to be composed into  $t_x$ , this arc will be treated as a self-linking arc and then removed in the end. This is because the transformation matrix must have “1’s” on positions  $(x, y)$  and  $(x, x)$ , which sets condition  $|f_{\text{row}}(x) \neq f_{\text{row}}(y)|$  false. If  $t_y$  is to be composed into  $t_z$  instead of  $t_x$ , this arc will remain but will be redirected to link  $t_z$  to  $t_y$ , as the “1” on position  $(z, y)$  will enforce this redirection during the matrix multiplication.
- 3) *An arc links an invisible task  $t_y$  to a visible one  $t_x$ .* This is similar to case 2).
- 4) *An arc links two invisible tasks  $t_x$  and  $t_y$ , both of which will be composed into the same visible task  $t_z$ .* In this case, the arc is treated as a self-linking one and will be removed, since the transformation matrix must have “1’s” on positions  $(z, x)$  and  $(z, y)$ , as discussed in the first part of case 2).
- 5) *An arc links two invisible tasks  $t_x$  and  $t_y$ , which will be composed into different visible tasks  $t_p$  and  $t_q$ , respectively.* This arc will remain and be directed to link  $t_p$  to  $t_q$ , as discussed in the second part of case 2).

From the aforementioned discussion, we can see that, in all cases, no arcs will connect with any hidden tasks, and only self-linking arcs will be removed in the composition operation. These findings prove the privacy protection of the composition operation. ■

*Property 2—No Privacy Disclosure in Connection Operation:*

*Proof:* Basically, the connection operation handles the messaging links that are defined in a boundary adjacency matrix through two matrix multiplications with different transforma-

tion matrices. Since a boundary adjacency matrix may have different numbers of rows and columns, the intermediate and result matrices need to transpose to adapt to the matrix multiplication. The  $f_{\text{diag}}$  of connection operation only keeps the “1’s” on the diagonal while changes “1’s” on other positions to “0,” for the purpose of removing all invisible tasks in messaging-link handling.

Now, let us discuss the handling of messaging links in the following cases.

- 1) *A messaging link connects an invisible task to a visible task or connects a visible task to an invisible task.* This messaging link will be hidden. Suppose this messaging link connects an invisible task  $t_j$ . In the boundary adjacency matrix, this messaging link should stand on row  $j$ , according to the definition of a boundary adjacency matrix. In the corresponding transformation matrix, there must be a “0” on position  $(j, j)$  and a “1” on position  $(i, j)$  if  $t_j$  is to be composed into visible task  $t_i$ . However, after the diagonalization, the “1” on  $(i, j)$  will be set to “0.” In the matrix multiplication, we know that only the “1’s” on column  $j$  in the transformation matrix can preserve the messaging link on row  $j$  in the boundary adjacency matrix. Yet, the transformation matrix has no “1’s” on column  $j$ , and therefore, the messaging link will disappear during the multiplication.
- 2) *A messaging link connects an invisible task to another invisible task.* This link will be hidden, as discussed in case 1).
- 3) *A messaging link connects a visible task to another visible task.* This messaging link will remain, since the transformation matrix will have “1’s” on the diagonal positions, which can preserve the messaging link.

From the aforementioned discussion, we can see that only messaging links connecting two visible tasks will remain, while all other messaging links will be hidden. This proves the privacy protection of the connection operation. ■

For expansion operation, it operates with the results of the aforementioned two operations, and therefore, its privacy protection property can be guaranteed by the property of those two operations.

#### D. Related Algorithms

1) *Generating a Collaborative Business Process:* The generation of a collaborative business process is a continuous updating process. Technically, it means to append a new generated column to the underlying self-adjacency matrix once a reachable business process is detected. This new generated column consists of a new self-adjacency matrix and a series of new boundary adjacency matrices. The new self-adjacency matrix denotes the inner structure of the detected reachable business process, while the new boundary adjacency matrices denote the interaction relationships between the detected business process and the constituent business processes of the collaborative business process.

Fig. 6 shows how a collaborative business process evolves from a single self-adjacency matrix to a composite one by including other matrices. At the starting point, the collaborative



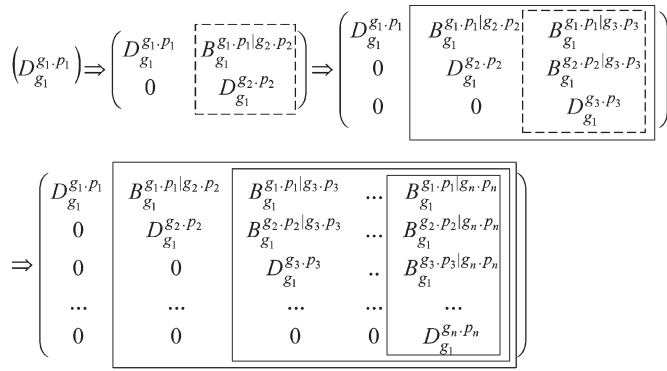


Fig. 6. Evolution of a collaborative business process.

business process contains only  $D_{g_1}^{g_1 \cdot p_1}$ , which means that only  $g_1 \cdot p_1$  is included. Afterward, organization  $g_1$  detects that perceivable process  $g_2 \cdot p_2$  is reachable from  $g_1 \cdot p_1$  and then appends a column containing  $B_{g_1}^{g_1 \cdot p_1 | g_2 \cdot p_2}$  and  $D_{g_1}^{g_2 \cdot p_2}$  to the right side. Likewise, when organization  $g_2$  detects that process  $g_3 \cdot p_3$  is reachable from  $g_1 \cdot p_1$  via  $g_2 \cdot p_2$ ,  $g_2$  may append a column containing  $B_{g_1}^{g_1 \cdot p_1 | g_3 \cdot p_3}$ ,  $B_{g_1}^{g_2 \cdot p_2 | g_3 \cdot p_3}$ , and  $D_{g_1}^{g_3 \cdot p_3}$ . This appending process continues until no more reachable processes can be detected.

For each expansion step, the interprocess interaction relations are identified by the organization (*context organization*) that owns the “bridging” business processes (by which the expansion proceeds). As such, the expansion spreads as a stepwise propagation, and the context organization changes from step to step. Organization  $g_1$  is called the *original context organization* of the generation of the collaborative business process.

Algorithm 1 details the generation procedure. First, we define some involved functions: Function  $\text{relatedProc}(p)$  returns a set of local and perceivable processes that have direct interactions with process  $p$ ; function  $\text{includedProc}(cbp)$  returns all included business processes at that moment in collaborative business process  $cbp$ , which initially contains a self-adjacency matrix defined on a local process of the original context organization; function  $\text{BAM}(p_1, p_2, g)$  returns the boundary adjacency matrix between processes  $p_1$  and  $p_2$  from the view of organization  $g$  using the connection operation; function  $\text{SAM}(p, g)$  returns the self-adjacency matrix of process  $p$  from the view of organization  $g$  using the composition operation; and function  $\text{genOrg}(p)$  returns the owner organization of process  $p$ .

**Algorithm 1** *genCBPStruc*—Generation of a collaborative business process

**Input:**

$cbp$ —A self-adjacency matrix for  $cbp$ ;

$ctxProc$ —A local process of the context organization;

$origCxtOrg$ —The original context organization that starts the generation;

**Output:**

$cbp$ —The expanded self-adjacency matrix.

1  $detectedProcSet = \text{relatedProc}(ctxProc)$ ;

2  $includedProcSet = \text{includedProc}(cbp)$ ;

```

3  $detectedProcSet = detectedProcSet -$ 
   $includedProcSet$ ;
4  $appendedProcSet = \emptyset$ ;
5 for each process  $p_i \in detectedProcSet$ 
6    $tempB = \text{BAM}(ctxProc, p_i, origCxtOrg)$ ;
7   if  $tempB$  is a nonzero matrix then
8      $newColumn = \text{NULL}$ ;
9     for each process  $p_j \in includedProcSet$ 
10       $B = \text{BAM}(p_j, p_i, origCxtOrg)$ ;
11      append  $B$  to  $newColumn$ .
12    end for
13     $D = \text{SAM}(p_i, origCxtOrg)$ ;
14    append  $newColumn$  and  $D$  to  $cbp$ , using expansion
    operation.
15     $includedProcSet = includedProcSet \cup \{p_i\}$ ;
16     $appendedProcSet = appendedProcSet \cup \{p_i\}$ ;
17  end if
18 end for
19 for each process  $p_i \in appendedProcSet$ 
20    $targetOrg = \text{genOrg}(p_i)$ ;
21    $cbp = targetOrg.\text{genCBPStruc}(cbp, p_i,$ 
     $origCxtOrg)$ ;
22 end for
23 return  $cbp$ ;
```

The generation process starts from a local process of the original context organization and then spreads to all reachable business processes of the involved organizations by recursively invoking this algorithm. When this generation process comes to an organization, this organization becomes the context organization of this algorithm. Technically, lines 1–3 detect the business process. Lines 4–18 expand the collaborative business process, where line 10 generates related boundary adjacency matrices of the new column and line 13 generates the self-adjacency matrix of the new column. Lines 19–22 propagate the detection procedure, and line 23 returns the expanded collaborative business process.

*Tracking a Collaborative Business Process:* Tracking a collaborative business process works like traversing a graph, whereby the related business process instances represent the nodes of the graph and their messaging links represent the arcs. The correlation between two process instances is determined when the visible messaging link is activated for the first time.

Algorithm 2 details the procedure for tracking a collaborative business process instance. First, we define some involved functions. Function  $\text{addInstance}(p, i)$  inserts instance  $i$  to the list of processes  $p$  in the tracking data structure given in Section IV-E. Function  $\text{addLink}(i_1, i_2)$  creates a link between instances  $i_1$  and  $i_2$  in the tracking data structure. Function  $\text{linkedInstances}(i, cbp)$  returns the instances linked to instance  $i$  in the tracking data structure according to collaborative business process  $cbp$ . Function  $\text{relatedBAMs}(p, cbp)$  returns the set of boundary self-adjacency matrices related to process  $p$  defined in  $cbp$ . Function  $\text{partnerProc}(B, p)$  returns the partner process of  $p$  defined in boundary self-adjacency matrix  $B$ . Function  $\text{genOrg}(p)$  returns the organization of process  $p$ . Function  $\text{genProc}(i)$  returns the process of instance  $i$ .



**Algorithm 2** *trackProc*—Tracking Process**Input:**

*Cbp*—The matrix for the collaborative business process to track;

*origInstance*—An instance of the original context organization's initial local process defined in *cbp*;

*DS*—The tracking data structure;

**Output:**

*DS*—The updated tracking data structure.

```

1  trackInstanceSet = ∅;
2  stacks = newstack();
3  s.push(origInstance);
4  while s is not empty do
5    cxtInstance = s.pop();
6    foundInstanceSet = linkedInstances
   (cxtInstance, cbp) – trackInstanceSet;
7    for each i ∈ foundInstanceSet
8      s.push(i);
9      cxtProc = genProc(cxtInstance);
10     BAMset = relatedBAMs(cxtProc, cbp);
11     for each link l of each boundary adjacency matrix
       B ∈ BAMset
12       partnProc = partnerProc(B, cxtProc);
13       partnOrg = genOrg(partnProc);
14       if cxtInstance.l is newly fired then
15         newInstanceSet = ∅;
16         Ask partnOrg to check any new participating instances
           of partnProc and set the instances to newInstanceSet.
17         newInstanceSet = newInstanceSet –
           trackInstanceSet;
           /*filter the previous discovered instances*/
18         for each i ∈ newInstanceSet
19           addInstance(partnProc, i);
20           addLink(cxtInstance, i);
           /*update the tracking data structure*/
22         s.push(i);
           /*add the newly discovered instance to the stack*/
24       end for
25     end if
26   end for
27   trackInstanceSet = trackInstanceSet ∪
     {cxtInstance};
28 end for
29 end while
30 for each instance i ∈ trackInstanceSet
31   p = genProc(i);
32   targetOrg = genOrg(p);
33   Inquire targetOrg for the execution status of i and
     then update the status of i in DS.
34 end for

```

This algorithm starts from a local process instance of the original context organization. Following the corresponding collaborative business process, this algorithm searches along visible messaging links and propagates the execution status

queries to all reachable business process instances. The corresponding collaborative business process records the interaction relationship between the processes of these reachable business process instances. When an interorganizational interaction occurs, the algorithm will check whether any new business process instances join the business collaboration. If so, these business process instances will be added to the tracking data structure. Technically, lines 4–29 discover the participating business process instances, where lines 11–26 search related instances by following each visible messaging link. Lines 30–34 update the execution status of participating business process instances.

## VI. SYSTEM IMPLEMENTATION

Web services are now becoming a popular platform for enterprise software development [44]–[46]. To prove the feasibility and applicability of the presented framework, we have developed a prototype based on Sun Microsystem's Java Web Service Application Programming Interfaces (API) stack. This API stack comprises Java API for XML Web Services [47], Java Architecture for XML Binding 2.0 [48], and SOAP with Attachments API for Java 1.3 [49]. The core part of the prototype is the process steering component, which interacts with the workflow engine and partner organizations for modeling and tracking collaborative business processes. Fig. 7 shows the architecture of the process steering component.

The process steering component handles both local and collaborative business processes. Particularly, components *perceivable process locator* and *collaborative business process assembler* are responsible for modeling and updating collaborative business processes, while components *monitoring manager* and *status assembler* are employed to propagate status inquiries to partner organizations and derive the execution progress of collaborative business process instances. In addition, this monitoring manager also handles partner organizations' inquiries by reporting the execution status of requested local process instances with the help of a *local process locator* and a *local monitor*.

The major components of process steering service are listed next.

- 1) The *perceivable process locator* searches for reachable perceivable processes for modeling collaborative business processes.
- 2) The *collaborative business process assembler* drives the generation and updating of collaborative business processes. It is responsible for appending the perceivable processes found by the *perceivable process locator* to proper collaborative business processes. Algorithm 1 details this appending procedure.
- 3) The *monitoring manager* propagates status inquiries to partner organizations and asks the *status assembler* to incorporate the retrieved status information to the process instances stored in the *collaborative business process instance database*. Algorithm 2 details this tracking procedure. On the other hand, when receiving the status inquiries from partner organizations, the monitoring manager calls the *local process locator* to map the requested

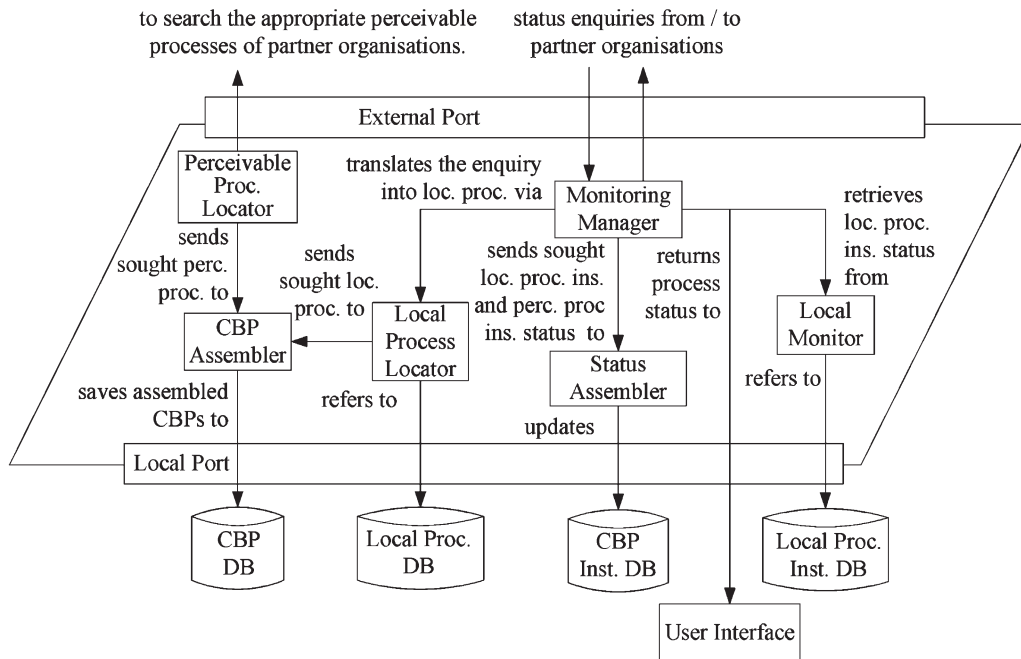


Fig. 7. Architecture of the process steering component.

perceivable process instance to the corresponding local process instance. After that, the monitoring manager commands the *local monitor* to check the execution status of this local process instance and finally returns the status information back.

- 4) The *local process locator* searches the *local process database* for the local processes that are specified by the status inquiry from partner organizations. After that, the found local processes will be sent to the *monitoring manager*.
- 5) The *local monitor* is responsible for monitoring the execution of local process instances.
- 6) The *status assembler* is particularly used to incorporate the retrieved execution status into the status of collaborative business process instances.

See [50] for more details about the system architecture.

## VII. DISCUSSION

The proposed framework dealt with the issues of flexibility, privacy protection, and scalability in modeling and tracking dynamic collaborative business processes. Technically, this framework employed the relative workflow model to support the organization-oriented modeling. The established process visibility mechanism enabled the visibility transitivity via “trackable” tasks, and thereby, an organization’s collaborative business process can go beyond neighboring organizations to cover the processes of nonneighboring organizations. Based on the matrix formalization, a set of matrix operations were developed to derive the structure of a collaborative business process on the fly. A particular data structure and corresponding algorithm were introduced to support the collaborative business process tracking, with emphasis on workflow cardinality and instance correspondence.

Compared with other collaborative business process approaches, our framework brought the following appealing features.

- 1) *Privacy protection.* The applied process visibility mechanism enables organizations to hide different process details from different partners according to actual partnerships and authority levels. Once the tasks are set invisible for a partner, they will never be released to the partner, and thereby, the privacy protection during the collaboration can be guaranteed.
- 2) *Scalable process structure.* Some approaches, like the Web Services Business Process Execution Language (WS-BPEL), model collaborative business processes from the perspective of a pivotal organization. Although it follows an organization-oriented observation perspective, the generated collaborative business process can only include the invocation interfaces of partner organizations’ services. This scheme restricts the host organization’s perception within its neighboring organizations. On the contrary, our framework enables the process visibility transitivity via “trackable” tasks, whereby an organization’s collaborative business process can go beyond neighboring organizations to cover the processes of non-neighboring organizations.
- 3) *Flexible derivation.* The structure of a collaborative process can be dynamically derived from related visibility perceptions. Before a new process joins the collaboration, corresponding visibility perceptions will be generated. If this new process is detected and found reachable by other organizations, it will be appended to the collaborative business processes of these organizations. Similarly, the deletion of a business process will be dynamically updated if an organization leaves the collaboration. Due to the diversity of process visibilities,

for organization  $g$ , only the changes to  $g$ 's reachable and perceivable processes will be updated to  $g$ 's collaborative business process, and therefore, extra updating loads can be avoided.

- 4) *High autonomy*. In our framework, each organization can customize the visibility constraints for its local business processes to blind off private information from partners, while it can also freely explore all the visible and reachable processes from partners to update its collaborative business process. Therefore, organizations own high autonomy for process modeling and tracking.

However, the migration to our framework may bring some tradeoffs, which can be potential limitations. Some tradeoffs and consequent limitations are summarized as follows, although they may be outweighed by many advantages offered by our approach.

- 1) Due to the organization-oriented observation perspective, different organizations may generate different collaborative business processes for the same collaboration. The inconsistency between these business processes inevitably results in the complex coordination between participating organizations. In the WS-BPEL environment, to handle this issue, we have extended WS-BPEL partner links with a new attribute "invokeType" to differentiate the function invocations, i.e., task execution requests from the host organization and partner organizations. For the latter, the invocations have to be redirected to actual Web service operations by mapping the perceivable process to its corresponding local process. In the developed prototype, the status assembler and local workflow locator are particularly responsible for maintaining this correspondence. For more details, please refer to [50] and [51].
- 2) This framework employs matrices to represent business process structures. Yet, the matrices may contain quite a few zeros, which affect the space efficiency of data storage. This problem may be erased by adopting some matrix compression techniques in our future work.

### VIII. CONCLUSION AND FUTURE WORK

This paper has focused on business process modeling and tracking in the interorganizational collaboration scenario. A matrix-based framework has been presented and illustrated to support the modeling and tracking of dynamic collaborative business processes with developed algorithms. The modeling and tracking firmly follow the process visibility constraint and transitivity and, therefore, can guarantee the privacy protection for participating organizations. The modeling and tracking is conducted in a continuous propagating manner and thereby supports the flexibility of dynamic collaborative business processes well. A prototype system has been implemented for proof-of-concept purpose.

In the future, we are to refine the applied data structure to minimize the mentioned tradeoffs. Another direction for future development is to monitor the performance of partners, i.e., evaluate the conformity between the results achieved by each

partner and the contracted tasks. The result of such performance monitoring and evaluation will be useful to guide the subsequent processes of partner search and selection.

### REFERENCES

- [1] C. Liu, Q. Li, and X. Zhao, "Challenges and opportunities in collaborative business process management," *Inf. Syst. Frontiers*, vol. 11, no. 3, pp. 201–209, Jul. 2009.
- [2] H. Osterle, E. Fleisch, and R. Alt, *Business Networking—Shaping Collaboration Between Enterprises*. New York: Springer-Verlag, 2001.
- [3] H. Smith and P. Fingar, *Business Process Management—The Third Wave*. Tampa, FL: Meghan-Kiffer Press, 2003.
- [4] M. M. Cunha and G. D. Putnik, "On the dynamics of agile/virtual enterprise reconfiguration," *Int. J. Netw. Virtual Organisations*, vol. 3, no. 1, pp. 102–123, Jan. 2006.
- [5] M. M. Cunha and G. D. Putnik, "Identification of the domain of opportunities for a market of resources for virtual enterprise integration," *Int. J. Prod. Res.*, vol. 44, no. 12, pp. 2277–2298, Jun. 2006.
- [6] R. T. By, "Organisational change management: A critical review," *J. Change Manage.*, vol. 5, no. 4, pp. 369–380, Dec. 2005.
- [7] P. Radha Krishna, K. Karlapalem, and D. K. W. Chiu, "An EREC framework for e-contract modeling, enactment and monitoring," *Data Knowl. Eng.*, vol. 51, no. 1, pp. 31–58, Oct. 2004.
- [8] R. Singh and A. F. Salam, "Semantic information assurance for secure distributed knowledge management: A business process perspective," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 36, no. 3, pp. 472–486, May 2006.
- [9] M. zur Muehlen and M. Rosemann, "Workflow-based process monitoring and controlling—Technical and organizational issues," in *Proc. 33rd Hawaii Int. Conf. Syst. Sci.*, Maui, HI, 2000, pp. 6032–6042.
- [10] O. Perrin and C. Godart, "A model to support collaborative work in virtual enterprises," *Data Knowl. Eng.*, vol. 50, no. 1, pp. 63–86, Jul. 2004.
- [11] M. Laguna and J. Marklund, *Business Process Modeling, Simulation, and Design*. Upper Saddle River, NJ: Pearson, 2005.
- [12] M. Rouached, O. Perrin, and C. Godart, "A contract layered architecture for regulating cross-organisational business processes," in *Proc. 3rd Int. Conf. Bus. Process Manage.*, Nancy, France, 2005, pp. 410–415.
- [13] B. T. Clegg, "Building a Hierarchy using business process-oriented holonic (ProH) modeling," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 37, no. 1, pp. 23–40, Jan. 2007.
- [14] WfMC, *The Workflow Reference Model*, [WfMC 1003], 1995.
- [15] WfMC, *Workflow Management Coalition Audit Data Specification* [WfMC 1015], 1998.
- [16] *Using BEA WebLogic Components*, 2000.
- [17] *BPEL4WS Business Processes With WebSphere Business Integration*, IBM, Armonk, NY, 2004.
- [18] M. Wang and H. Wang, "Intelligent agent supported flexible workflow monitoring system," in *Proc. 14th Int. Conf. Adv. Inf. Syst. Eng.*, 2002, pp. 787–791.
- [19] W. Hur, H. Bae, and S.-H. Kang, "Customizable workflow monitoring," *Concurrent Eng.—Res. Appl.*, vol. 11, no. 4, pp. 313–325, Dec. 2003.
- [20] Action Technologies, Inc.
- [21] TIBCO Software Inc.
- [22] Digital Equipment Corporation.
- [23] Hewlett-Packard Company: Process Manager.
- [24] *IBM WebSphere Business Integration Handbook*, IBM, Armonk, NY, 2005.
- [25] *WebLogic Integration 7.0 Documentation*, 2005.
- [26] [Online]. Available: <http://msdn.microsoft.com/en-us/netframework/aa663328.aspx>
- [27] L. Shostack, "Designing services that deliver," *Harvard Bus. Rev.*, vol. 62, no. 1, pp. 133–139, Jan. 1984.
- [28] S. K. Katsikas, J. Lopez, and G. Pernul, "Trust, privacy and security in digital business," *Comput. Syst.: Sci. Eng.*, vol. 20, no. 6, pp. 391–400, 2005.
- [29] X. Zhao, C. Liu, and Y. Yang, "An organisational perspective on collaborative business processes," in *Proc. Int. Conf. Bus. Process Manage.*, Nancy, France, 2005, pp. 17–31.
- [30] X. Zhao, C. Liu, Y. Yang, and W. Sadiq, "Aligning collaborative business processes: An organisation-oriented perspective," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 39, no. 6, pp. 1152–1164, Nov. 2009.
- [31] D. Chen and N. Daclin, "Barriers driven methodology for enterprise interoperability," in *Proc. 8th IFIP Working Conf. Virtual Enterprises*, Guimarães, Portugal, 2007, pp. 453–460.

- [32] K. A. Schulz and M. E. Orłowska, "Facilitating cross-organisational workflows with a workflow view approach," *Data Knowl. Eng.*, vol. 51, no. 1, pp. 109–147, Oct. 2004.
- [33] C. Issam, D. Schahram, and T. Samir, "The view-based approach to dynamic inter-organizational workflow cooperation," *Data Knowl. Eng.*, vol. 56, no. 2, pp. 139–173, Feb. 2006.
- [34] G. Preuner and M. Schrefl, "Requester-centered composition of business processes from internal and external services," *Data Knowl. Eng.*, vol. 52, no. 1, pp. 121–155, Jan. 2005.
- [35] D. R. Liu and M. Shen, "Workflow modeling for virtual processes: An order-preserving process-view approach," *Inf. Syst.*, vol. 28, no. 6, pp. 505–532, Sep. 2003.
- [36] R. Eshuis and P. Grefen, "Constructing customized process views," *Data Knowl. Eng.*, vol. 64, no. 2, pp. 419–438, Feb. 2008.
- [37] P. Grefen, K. Aberer, H. Ludwig, and Y. Hoffner, "CrossFlow: Cross-organizational workflow management for service outsourcing in dynamic virtual enterprises," *Data Eng.*, vol. 24, no. 1, pp. 52–57, 2001.
- [38] W. M. P. van der Aalst and M. Weske, "The P2P approach to inter-organizational workflows," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.*, 2001, pp. 140–156.
- [39] A. Lazcano, H. Schuldt, G. Alonso, and H. J. Schek, "WISE: Process based e-commerce," *IEEE Data Eng. Bull.*, vol. 24, no. 1, pp. 46–51, 2001.
- [40] E. Colombo, C. Francalanci, and B. Pernici, "Modeling coordination and control in cross-organizational workflows," in *Proc. DOA/CoopIS/ODBASE*, 2002, pp. 91–106.
- [41] I. Wetzel and R. Klischewski, "Serviceflow beyond workflow? IT support for managing inter-organizational service processes," *Inf. Syst.*, vol. 29, no. 2, pp. 127–145, Apr. 2004.
- [42] X. Zhao and C. Liu, "Tracking over collaborative business processes," in *Proc. Int. Conf. Bus. Process Manag.*, Vienna, Austria, 2006, pp. 33–48.
- [43] X. Zhao, C. Liu, Y. Yang, and W. Sadiq, "Handling instance correspondence in inter-organisational workflows," in *Proc. 19th CAiSE*, Trondheim, Norway, 2007, pp. 51–65.
- [44] J. Zhang, C. K. Chang, L.-J. Zhang, and P. C. K. Hung, "Toward a service-oriented development through a case study," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 37, no. 6, pp. 955–969, Nov. 2007.
- [45] Z. Maamar, D. Benslimane, G. K. Mostéfaoui, S. Subramanian, and Q. H. Mahmoud, "Toward behavioral web services using policies," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 6, pp. 1312–1324, Nov. 2008.
- [46] P. Xiong, Y. Fan, and M. Zhou, "QoS-aware web service configuration," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 4, pp. 888–895, Jul. 2008.
- [47] JAX-WS, *Java API for XML Web Services*.
- [48] JAXB, *Java Architecture for XML Binding*.
- [49] SAAJ, *SOAP with Attachments API for Java*.
- [50] X. Zhao and C. Liu, "Supporting relative workflows with web services," in *Proc. 8th Asia Pacific Web Conf.*, Harbin, China, 2006, pp. 680–691.
- [51] X. Zhao, "On supporting collaborative business processes—An organisation-oriented perspective," Ph.D. dissertation, Swinburne Univ. Technol., Melbourne, Australia, 2007, in Faculty of ICT.



**Xiaohui Zhao** received the B.S. and M.S. degrees in computer science and technology engineering from the Harbin Institute of Technology, Harbin, China, in 2001 and 2003, respectively, and the Ph.D. degree in information technology from Swinburne University of Technology, Melbourne, Australia, in 2007.

Before coming to The Netherlands, he participated in the Australian Research Council Linkage Project "Kaleidoscope" with SAP Research, Australia. He is currently a Research Fellow in the Information Systems Group, Department of Industrial Engineering and Innovation Sciences, Eindhoven University of Technology, Eindhoven, The Netherlands. He has published around 30 peer-reviewed papers in international journals and conferences, and he has also organized the international workshop series on collaborative business processes. His research interests include business process management, pervasive computing, service-oriented computing, etc.



**Chengfei Liu** received the B.S., M.S., and Ph.D. degrees in computer science from Nanjing University, Nanjing, China, in 1983, 1985, and 1988, respectively.

He is currently a Full Professor and the Head of the Web and Data Engineering Research Group, Centre for Complex Software Systems and Services, Faculty of Information and Communication Technologies, Swinburne University of Technology, Melbourne, Australia. Prior to joining Swinburne University of Technology in 2004, he taught with the University of South Australia, Adelaide, and the University of Technology Sydney, Sydney, Australia, and was a Research Scientist with the Cooperative Research Centre for Distributed Systems Technology (DSTC), Brisbane, Australia. He also held visiting positions at The Chinese University of Hong Kong, Shatin, Hong Kong, The University of Aizu, Aizuwakamatsu, Japan, and the IBM Silicon Valley Laboratory, San Jose, CA. He has published more than 120 peer-reviewed papers in various journals and conference proceedings. He was a Guest Coeditor for three special issues for the *Journal of Database Management*, the journal *Information Systems Frontiers*, and the *World Wide Web Journal*. His research interests include database systems, XML data management, business process management, workflows, and web information systems.

Dr. Liu has served on technical program committees and organizing committees of about 70 international conferences or workshops in the areas of database systems, Web information systems, and workflow systems. He was a recipient of the Vice Chancellor's Award for Research Excellence at Swinburne University of Technology in 2007.