On Supporting Collaborative Business Processes — an Organisation-Oriented Perspective

by

Xiaohui Zhao

B.E. (HIT, China)
M.E. (HIT, China)

A thesis submitted to

Faculty of Information and Communication Technologies

Swinburne University of Technology

for the degree of

DOCTOR OF PHILOSOPHY

November, 2007

Abstract

Current globalisation of economy drives organisations to streamline their business processes and to quickly form collaboration for responding to fast changing market opportunities. This thesis is dedicated to proposing a novel organisation-oriented view methodology for collaborative business process management. This methodology observes a collaborative business process from the perspective of individual organisations, and provides comprehensive solutions to the privacy, autonomy and openness issues in business collaboration.

In this thesis, we first propose a relative workflow model to formalise the business process modelling from the organisation-oriented perspective. This model deploys a constraint-based visibility control mechanism to protect business privacy, and to distinguish the partnerships between collaborating organisations. At the instance level, we investigate the instance correspondence in a collaborative business process, and characterise the instance correspondence in terms of the workflow cardinality and instance correlations. An extended Petri net model formalises the representation and dynamic tracing of the instance correspondence. On this basis, we utilise the instance correspondence to perform inter-organisational workflow tracking from each participating organisation point of view. In addition, we develop a series of representation matrices and matrix operations that allow dynamic tracking and monitoring on collaborative business processes. To demonstrate the applicability of the organisation-oriented view methodology, we conduct two case studies on a virtual organisation alliance and a transient supply chain, respectively. Finally, a system design is introduced to demonstrate the feasibility and deployment of our methodology in the Web service environment.

The research reported in this thesis provides a comprehensive solution for collaborative business process management. The reported research puts forward an innovative idea and constructs a solid foundation for future research towards collaborative business process management.

The Author's Publications

Journal papers

[1] X. Zhao and C. Liu, "Process Integration Based on Interoperability across Multiple Workflow Domains", *International Journal of Internet and Enterprise Management*, inderScience publisher, Vol.3, No.2. pp.199-213.

Conference papers

[2] X. Zhao and C. Liu, "Version Management in the Business Process Change Context". *Proceedings of the Fifth International Conference on Business Process Management* (BPM 2007), Brisbane, Australia, Lecture Notes in Computer Science Vol. 4714, pp.198-213, 2007.

[3] X. Zhao, C. Liu, Y. Yang and W. Sadiq, "Handling Instance Correspondence in Inter-Organisational Workflows". *Proceedings of the Nineteenth International Conference on Advanced Information Systems Engineering* (CAiSE'07), Trondheim, Norway, Lecture Notes in Computer Science Vol. 4495, pp.51-65, 2007.

[4] X. Zhao, "On Supporting Collaborative Business Processes". *Proceedings of the Nineteenth International Conference on Advanced Information Systems Engineering* (CAiSE'07), Doctoral Consortium, Trondheim, Norway, pp.881-888, 2007.

[5] X. Zhao, C. Liu and Y. Yang, "Specifying Instance Correspondence in Collaborative Business Processes", *Proceedings of the Fifteenth International Conference on Cooperative Information Systems* (CoopIS'06), Montpellier, France, Lecture Notes in Computer Science Vol. 4277, pp.1-2, 2006.

[6] X. Zhao and C. Liu, "Tracking over Collaborative Business Processes", *Proceedings of the Fourth International Conference on Business Process Management* (BPM 2006), Vienna, Austria, Lecture Notes in Computer Science Vol. 4102, pp.33-48, 2006.

[7] X. Zhao, C. Liu and Y. Yang, “Supporting Virtual Organisation Alliances with Relative Workflows”, *Proceedings of the Fourth Asia Pacific Conference on Conceptual Modelling* (APCCM 2006), Hobart, Australia, pp.115-124, 2006.

[8] X. Zhao and C. Liu, “Supporting Relative Workflows with Web Services”, *Proceedings of the Eighth Asia Pacific Web Conference* (APWeb'06), Harbin, China, Lecture Notes in Computer Science Vol. 3841, pp.680-691, 2006.

[9] X. Zhao, C. Liu and Y. Yang, “An Organisational Perspective on Collaborative Business Processes”, *Proceedings of the Third International Conference on Business Process Management* (BPM 2005), Nancy, France, Lecture Notes in Computer Science Vol. 3649, pp.17-31, 2005.

[10] X. Zhao, C. Liu and Y. Yang, “A Web Services Based Architecture for Workflow Management Systems”. *Proceedings of the Fifteenth International Conference on Database and Expert Systems Applications* (DEXA 2004), Zaragoza, Spain, Lecture Notes in Computer Science Vol. 3180, pp.34-43, 2004.

[11] X. Zhao, “Workflow Simulation across Multiple Workflow Domains”. *Proceedings of the Thirteenth International Conference on Database and Expert Systems Application*s (DEXA 2002), Aix en, France, Lecture Notes in Computer Science Vol. 2453, pp.50-59, 2002.

Acknowledgements

There are lots of people I would like to thank for a variety of reasons.

It is difficult to overstate my gratitude to my supervisor, Associate Professor Chengfei Liu. With his enthusiasm, his inspiration, and his great efforts to explain things clearly and simply, he helped me initiate and forward this research. I deeply appreciate his considerable effort and invaluable comments in earlier drafts of this thesis. I also thank my associate supervisor, Prof. Yun Yang, for his support and valuable guidance, especially for his time in helping me decide my research topic.

I would like to thank my colleagues in the Web and Data Engineering Program of the Centre for Information Technology Research. Their kindness and help contributed to the completion of this thesis. My appreciation also goes to Faculty of Information and Communication Technologies, Swinburne University of Technology, for their financial support throughout my research, and their assistance in various administrative issues.

I wish to thank Associate Professor, Yusheng Ji of the National Institute of Informatics in Japan, for her support and encouragement. Without her help, there would be difficulties for me to accomplish my PhD.

Leaving the best to last, I wish to thank my parents, Liankui Zhao and Shuhua Wang. They bore me, raised me, supported me, taught me, and loved me. To them I dedicate this thesis.

Declaration

This thesis contains no material which has been accepted for the award to the candidate of any other degree or diploma, except where due reference is made in the text of the thesis. To the best of the candidate's knowledge, this thesis contains no material previously published or written by another person except where due reference is made in the text of the thesis.

Xiaohui Zhao

November 2007

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis addresses business process management in the loosely coupled collaboration environment. The research reported in this thesis particularly focuses on supporting autonomy, privacy protection and flexibility of organisations during inter-organisational collaborations. These features are highly required nowadays by B2B eCommerce applications. To provide comprehensive supports to these features, an organisation-oriented workflow model, known as relative workflow model, is presented in this thesis to describe a collaborative business process from individual organisations' perspective. In addition, the phenomenon of multiple instance correspondence in a collaborative business process is represented by Petri net based solution for multiple instance representation and execution. In this multiple instance correspondence scenario, workflow tracking issue is also investigated in the context of the proposed relative workflow model. Moreover, Web service based architecture for managing collaborative business processes is presented in this thesis. This research provides a comprehensive solution for collaborative business process management from a new perspective.

This chapter introduces the background and key issues of this research. First, a brief introduction to workflow management is given in Section 1.1. Then, Section 1.2 outlines the key issues of this research. Finally, Section 1.3 presents the structure of the remainder of this thesis.

1.1 History of Business Process Management

Workflow Management Coalition (WfMC), an independent workflow standard consortium, has defined business processes as follows [1]:

"*A set of one or more linked procedures or activities which collectively realise a business objective or policy goal, normally within the context of an organisational structure defining functional roles and relationships*."

In practical cases, business processes combine information flow, task flow, finance flow, material flow and related restrictions to provide a comprehensive behaviour script to achieve a specific business goal. A well defined business process precisely formalises organisational business behaviours using business planning, predicating, and reuse of previous knowledge. This formal process shifts business management from an art level to an engineering level. Organisations are seeking appropriate technology solutions to define reasonable business processes and align their business behaviours according to these business processes. The technology is now commonly referred to as Business Process Management (BPM).

Business process management has been defined as follows [2]:

"*Supporting business processes using methods, techniques, and software to design, enact, control, and analyse operational processes involving humans, organisations, applications, documents and other sources of information.*"

The scope of business process management covers from business process design and execution to business process monitoring. Business process management technology enables organisations to effectively design, control and implement their business processes. Historically, the advance of business process management can be classified with three major stages so far [2, 3].

(1) The first stage began in 1920, where non-automated processes were implicitly put in work practice.

(2) The second stage occurred during the 90s of last century. In this stage, there emerged many enterprise integration methodologies and approaches,

represented by Enterprise Application Integration (EAI) solutions such as Enterprise Resource Planning (ERP) etc.

In this period, workflow technology was widely adopted as the standard enabling technology for intra-organisational business process management. Typically, in 1999, WfMC has defined a workflow as [1]:

"*The 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*".

While, a Workflow Management System is defined as [1]:

"*A system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications.*"

Workflow technologies for business process management progressed fast to a relatively matured level during this period. Many classical workflow research were extensively applied, such as workflow reference model from WfMC and Petri net based WF-Net model [4], along with many commercial workflow software products, such as Staffware, MQSeries [5], COSA and other workflow engines from SAP [6], Oracle [7], PeopleSoft etc.

However, most of the work at this stage roughly focused in an intra-organisational scope. Business processes mainly describe the business behaviours of a single organisation, and workflows are merely used to integrate internal business processes within the organisation's boundary.

(3) Current days, the exploding trade globalisation for rapidly changing market opportunities drives the third surge of business process management. This stage is charactered with organisations' urgent requirements for prompt collaborations across organisational boundaries.

To support global collaborations, business processes expand to include internal and external partners, systems, and resources. According to the market forecasts from Forrester Research [8], American eCommerce market is expected to grow from $172 billion in 2005 to $329 billion in 2010. Moreover, it has been envisaged that the European eCommerce market may surge to €263 billion in 2011, and the total spending for business process outsourcing will rise from €11.0 billion in 2006 to €18.9 billion in 2011 with a compound annual growth rate at 11.5%. In this scenario, a business process is expected to conduct collaborative business across multiple organisations. Consequently, relevant business processes across participating organisations are integrated for efficient functioning of business in the global market. This kind of integrated business process is called *collaborative business process*. Inevitably, a collaborative business process will definitely confront more dynamics from the contained external business processes, and more complicated relationship between participating organisations. Therefore, for the organisations of this period, business process management is expected to offer a breakthrough that makes end-to-end, dynamic, expanding, contracting and ever changing business processes manageable.

The notion of business process in this period transformed to a complete and dynamically coordinated set of collaborative and transactional activities that deliver value to customers. Such business processes are characteristically [3]:

- Large and complex, involving the end-to-end flow of materials, information and business commitments.
- Dynamic responding to demands from customers and to changing market conditions.
- Widely distributed and customised across boundaries within and between businesses, often spanning multiple applications on disparate technology platform.
- Long running – a single instance of a process such as “order to cash” or “develop product” may run for months or even years.
- Automated – at least in part. Routine or mundane activities are performed by computers wherever possible, for the sake of speed and reliability.

Corresponding requirements for the third stage business process management will be extensively discussed in the next section.

1.2 Issues of Business Process Collaboration

To stay competitive, nowadays organisations must be agile in adapting their business processes to market dynamics. The adaptive business process based organisations should look beyond the traditional organisations and marketplaces through collaborative interactions and dynamic e-business solution bindings. In this revolutionary stage, business process management is required to provide the capability for dynamic discovery of trading partners and service providers for organisations. Moreover, business process management enables federated security mechanisms, solution monitoring and management over collaborating business processes in a loosely coupled collaboration environment. Particularly, business process management are expected to:

- Describe the business services that organisations can provide or they need from partners in service level agreements.
- Harmonise the enactment of collaboration by coordinating the participating organisations according to defined business processes.
- Maximise the autonomy of organisations during collaborations, and thereby ensuring organisations to benefit most from their own business objectives.
- Represent the partnerships between collaborating organisations during collaborations, and update the partnership changes.
- Guarantee the business privacy of organisations during collaborations.
- Allow specialists and other third parties to monitor, measure, and oversee the execution of business processes.

In recent years, previous efforts target at the primitive aspects of business process collaboration management, such as universal business process specification languages, system infrastructures and communication protocols across heterogeneous platforms [5, 9-11]. These achievements currently provide an acceptable foundation for basic collaboration functionalities, such as business service description, partnership representation, etc. Yet, there is still a great demand for further advancement surrounding collaborative business process management domain.

Paybacks from business endeavours inherently drive organisations to join in collaborations, and organisations intentionally go in for their own business goals from collaborations. This intrinsic feature naturally leads to the request for higher *autonomy* of organisations during business collaborations. Organisations are expecting more entitlements for selecting partner organisations, choosing collaborating business processes, changing interaction behaviours, etc., to maximally customise the collaborative business processes.

To maintain the agility towards changing market opportunities, organisations may have to alter partners, change collaboration processes, etc. Thus, the collaborations must be dynamic and flexible. Pre-fixed collaborative business processes are no longer popular, since they fail to quickly adapt the market changes. Business process management is therefore required to allow run time modifications on executing business processes, and dynamic updates of partnerships.

Potential privacy disclosure may arise during business collaborations, as the collaborative business processes are shared by all participating organisations. In the context of processing sensitive information, special restrictions should be applied according to actual partnerships.

The dilemma between privacy protection and flexible collaboration results in the issue of collaboration *openness*, which means the level of exposure of organisations' internal business processes to partner organisations. A customisable perception control mechanism over business processes is mandatory for the proper tuning of the openness feature.

To execute such dynamic, complex, collaborative business processes, it requires good harmonisation of cooperation between participating organisations. For this requirement, the main issue falls in the instance correspondence between business processes. Regarding the instance correspondence in a collaborative business process, the first issue is how to sort out the underlying logical and semantic relation between collaborating business process instances, and follow it to coordinate the *cooperation*. Technically, the correlation between business process instances is the key to this issue.

Furthermore, collaborative business process management necessitates workflow tracing and tracking for organisations to perceive the execution status of the participated collaboration. Unlike intra-organisational workflow tracking, inter-organisational workflow tracking enables organisations to be aware of the business processes execution beyond organisational boundaries. In addition, inter-organisational workflow tracking also needs to cater for the scalability and flexibility of business collaborations.

1.3 Key Issues of This Research

This thesis targets collaborative business process modelling and execution. The body of this research covers a new theory of observing collaboration, a series of supporting mechanisms, and a facilitating infrastructure for collaborative business process management. The pertinent issues for the problems stated in this research, viz., *autonomy*, *openness* and *cooperation*, are all well supported in this thesis.

Comprehensive architecture for this research is as follows:

(a) A formal organisation-oriented relative workflow model specifying what business processes of participating organisations are perceived and how these processes participate in collaborative business processes;

This workflow model applies an organisation-oriented view to model collaboration business processes. Contrary to traditional workflow models, this workflow model defines a collaborative business process from the perspective of each individual organisation. This facilitates different organisations to define the same collaborative business process differently. This feature reflects appropriately the diverse perception ability of collaboration between participating organisations, according to their actual partnerships in a given collaboration.

(b) Mechanisms for supporting such a model, which include modelling, enacting and tracking;

With respect to business process modelling, this organisation-oriented view mechanism secures the autonomy for organisations, since each organisation is empowered the authority to design its own collaborative business process from its own perspective. In addition, an explicit visibility control mechanism is offered to provide a customisable privacy protection to organisations.

As business process execution is concerned, instance correspondence is particularly targeted, since business collaboration always involves complex instance correlations across different business processes from different organisations. To formally address this issue, a Petri net based solution is presented in this thesis, together with corresponding algorithms.

Regarding business process monitoring in a collaboration environment, this research has proposed a workflow tracking solution based on matrices. Corresponding algorithms are developed to precisely describe how to define a tracking structure for a collaborative business process from the perspective of an individual organisation, and how we follow this tracking structure to perform inter-organisational workflow tracking on the fly.

(c) Supporting architecture and infrastructures based on Web service platform.

To demonstrate the feasibility, we design a facilitating prototype system for our organisation-oriented view solution. This system is purely implemented with Web service technology, and is incorporated with an extended version of WS-BPEL business process definition language. This prototype system fully supports our organisation-oriented workflow model and the proposed modelling, enacting and tracking mechanisms.

1.4 Structure of The Thesis

In particular, this thesis is dedicated in proposing an innovative solution for collaborative business process management, which is based on an organisation-oriented perspective to design and manage collaborative business processes. A comprehensive paradigm, including workflow modelling, justification at process level, workflow correspondence and workflow tracking at instance level, together with case studies and a prototype system design, is presented in this thesis.

Chapter 2 briefly introduces the progress track of workflow and business process research. Thereafter, this chapter reviews the major related work in categories of workflow standards, business process modelling, implementation technologies, monitoring and correspondence, etc.

With a motivating collaboration example, Chapter 3 first discusses about the requirements for modern collaborative business process management, where problems of organisation autonomy, openness and flexibility for cooperation evolvement are identified. Aiming to solve these problems, Chapter 3 proposes a relative workflow model which follows an organisation-oriented observation scheme. Moreover, Chapter 3 describes the procedure to derive a relative workflow process according to related commercial contracts. Finally, Chapter 3 presents a proof for the information sufficiency and necessity of our relative workflow model.

With the proposed relative workflow model, Chapter 4 discusses the instance correspondence in context of collaborative business processes. In Chapter 4, cardinality parameter and correlation structure are proposed to represent instance correspondence at process level and instance level. Traditional Petri nets are extended with these cardinality parameter and correlation structure to describe the instance correspondence in collaborative business processes.

By applying the instance correspondence representation in Chapter 4, we develop a workflow tracking approach in Chapter 5. This chapter first analyses the requirements for inter-organisational workflow tracking, and defines a set of rules with respect to workflow tracking in context of relative workflows. Based on the Petri net based instance correspondence framework, this chapter uses matrices to formally specify build time structure generation procedure and run time tracking mechanisms.

In Chapter 6, two case studies are conducted to demonstrate the application of relative workflow approach. This chapter at the outset analyses two typical collaboration scenarios namely virtual organisation alliance and transient supply chain. Subsequently, this chapter illustrates the deployment of relative workflow approach to support these two collaboration scenarios, with emphasises on workflow tracking and dynamic collaborations, respectively.

Chapter 7 presents the design of a prototype system on the basis of Web service platform. This system architecture consists of four services, viz., Agreement Management Service, Workflow Modelling Service, Workflow Engine Service and Workflow Monitoring Service. In addition, WS-BPEL is adopted as the default workflow modelling language in this system, and thereby this chapter includes a mapping from the components of our relative workflow model to WS-BPEL and WSDL elements.

Chapter 8 reviews the organisation-oriented view research presented in this thesis. The advantages of this research and the tradeoffs of the proposed approaches are discussed in this chapter.

The last chapter, Chapter 9, summarises the work presented in this thesis, the major contribution of this research, and further research goals.

Chapter 2

Literature Review

The past decade has witnessed that tremendous efforts were placed to the field of business process management. Due to the high pressure of business automation and globalisation, organisations are intensively re-structuring their organisational structures and business processes to make production and services more efficient and less expensive. Consequently, a lot of research in business process modelling, architectures and implementation techniques for business process management systems, and numerous commercial products have appeared.

The movement towards an architected process management infrastructure that started in the early 1990's has resulted in a number of product offerings, e.g., Action Technology's Action Workflow [12], TIBCO's InConcert [13], DEC's Object Flow [14], IBM's FlowMark (and subsequently, MQSeries Workflow) [5], HP's Process Manager [15]. The Workflow Management Coalition (WfMC) [16] founded in 1993, was the first industrial consortium aimed at promoting frameworks and interoperability for open architected process management. It published a reference model and a set of associated specifications. Besides, Object Management Group (OMG) [17] and other consortiums also established a series of specifications, e.g., Business Process Modelling Notation (BPMN) [18] and Workflow Facility Specification (WfFS) [19] etc., to regulate business process management. In addition, there is also recent work in inter-organisational collaborative business process management, and various consortiums, such as Organisation for the Advancement of Structured Information Standards (OASIS) [20] and Business Process Management Initiative (BPMI) [21] etc, are introducing standards for business process management. In the aspect of business process enabling technologies, several industrial consortiums, e.g., World Wide Web

Consortium (W3C) [22], ebXML [10], RosettaNet [11], Open Buying on the Internet (OBI) [23], etc., advocate Web service technologies with a series of frameworks. RosettaNet's Partner Interface Protocol (PIP) [11] is one of the earlier frameworks that made a significant contribution to the notion of business process based e-commerce. The ebXML consortium [10], which is particularly interesting because it attempts to leverage the pre-existing industry experience in Electronic Data Interchange (EDI) in designing new XML-based B2B frameworks.

In this chapter, a series of workflow standards are addressed in Section 2.1 to present a big picture of workflow glossaries and terminologies, including a brief introduction of workflow reference model and interoperability specification from WfMC; Section 2.2 discusses about conventional business process modelling, including classical Petri net based approaches, process algebra etc; Section 2.3 moves forward to inter-organisational business process modelling, including Public-to-Private, workflow view approaches, and some business-to-business oriented workflow languages and standards; Section 2.4 reviews inter-organisational workflow implementation technologies, which cover protocols and infrastructure technologies, together with typical workflow projects and commercial products; Section 2.5 gives a brief investigation upon workflow monitoring and correspondence.

2.1 Workflow Standards

2.1.1 Workflow glossary

Due to the increasing number of workflow vendors in the middle of 1990s, vendor-specific terminology for workflow constructs had led to an inconsistent vocabulary of workflow terms. In order to counter this trend, the first goal of the WfMC was to establish a common terminology for workflow concepts, which led to the publication of the WfMC Terminology & Glossary [1]. Today, the WfMC Glossary covers most workflow concepts and gives definitions for terms such as activity, workflow management system, and participant. Although not all workflow vendors use standard terminology, the WfMC vocabulary has found widespread acceptance in practice. It is perceived as a valuable aid for the system selection process, since proprietary terms used by different vendors can be transformed to a common standard, thus enabling a comparison of systems on the basis of a single vocabulary [24].

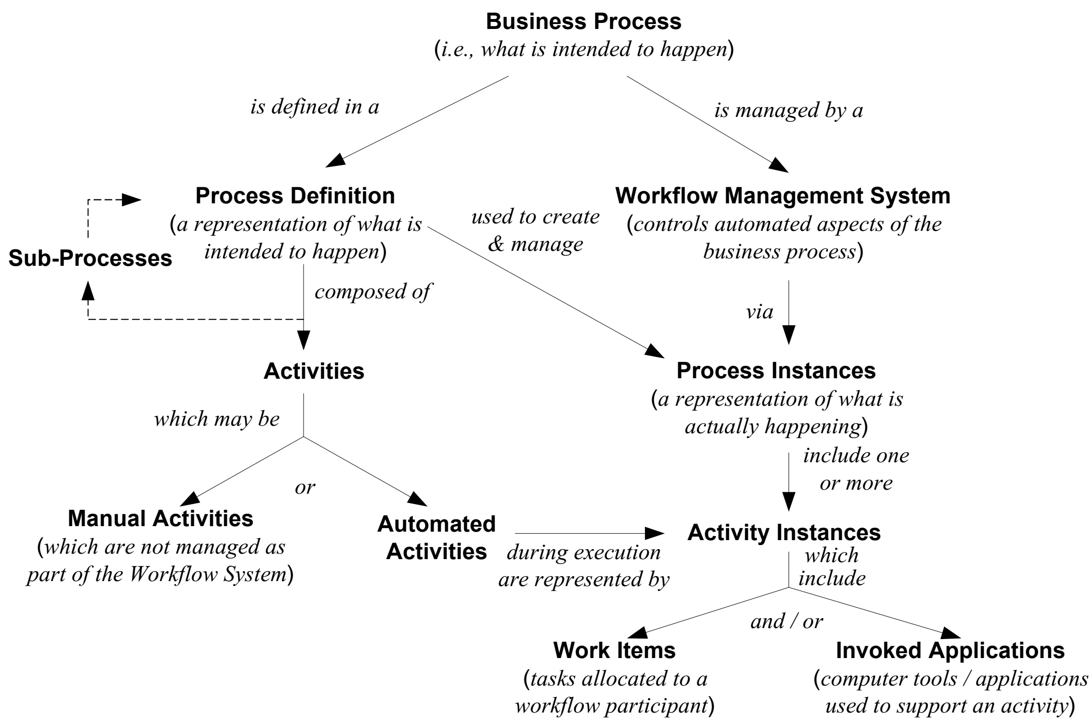

Figure 2.1 Relationships between workflow glossaries

Figure 2.1 illustrates the main terms of the WfMC glossary. The following list gives their definitions according to the WfMC Glossary.

- A *Business Process* is a set of coordinated tasks and activities, conducted by both people and equipment, which will lead to accomplishing a specific organisational goal.
- A *Process Definition* is the representation of a business process in a form which supports automated manipulation, such as modelling or enactment by a workflow management system. The process definition consists of a network of activities and their relationships, criteria to indicate the start and termination of the process, and information about the individual activities, such as participants, associated IT applications and data etc.
- A *Workflow* is the automation of a business process, in whole or part, creates and manages the execution of workflows. The execution and analysis of workflows rely on specialised software applications, which are designed to interpret process definitions, interact with workflow participants and invoke related IT applications.
- An *Activity* is a single piece of work that forms a distinct step within a process. An activity may be a manual activity or an automated, computer-based activity.

- A *Process Instance* is a specific instance of a particular process; more specifically, the representation of a single enactment of a process, or activity within a process, including its associated data. Each instance represents a separate thread of execution of the process or activity, and will have its own internal state and externally visible identity.
- An *Activity Instance* is a representation of an activity within a (single) enactment of a process, i.e., within a process instance."
- A *Workflow Participant* is a user who performs the work represented by an activity. This work is normally manifested as one or more work items assigned to the participant via the worklist.
- A *Work Item* is a representation of the work to be processed (by a workflow participant) in the context of an activity within a process instance.
- The *Worklist* is a list of work items associated with a role or participant. The worklist forms part of the interface between a workflow engine and the worklist handler."

2.1.2 Workflow reference model

Workflow Management Coalition published its reference model [25] in October 1994, identifying the functional area addressed by the workflow management facility and typical usage scenarios. This model defines a workflow management system and the most important system interfaces. Other WfMC standards as well as the OMG standard make reference to this model. Figure 2.2 shows the major components and interfaces of a workflow management system, which are outlined as follows:

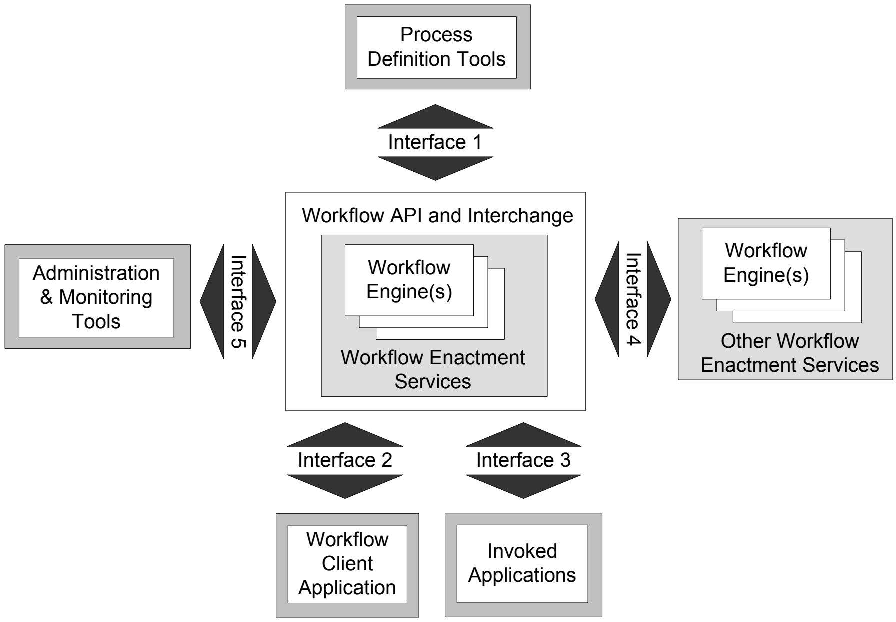

Figure 2.2 Workflow reference model

- *Workflow Engine*: A workflow engine is a software service that provides the run-time environment in order to create, manage and execute workflow instances.
- *Process Definition Tools*: Workflows are represented in form of process definitions, and a process definition toll is designed to edit these process definitions. This class of tools may also include components for workflow analysis, evaluation and simulation.
- *Workflow Interoperability*: Interfaces to support interoperability between different workflow systems.
- *Invoked Applications*: Interfaces to support interaction with a variety of IT applications.
- *Workflow Client Applications*: Interfaces to support interaction with the user interface.
- *Administration and Monitoring*: Interfaces to provide system monitoring and metric functions for facilitating the management of composite workflow application environments.

The release of this reference model is regarded as a milestone in workflow management area. Since its advent, the WfMC's workflow reference model has been widely accepted as the guide to develop workflow systems. Almost all deployable workflow systems are based on and compatible with this reference model.

2.1.3 Workflow interoperability standards

WfMC has defined a set of specifications on workflow models and systems, including workflow interoperability interfaces. The workflow interoperability interface defines the mechanisms that workflow product vendors are required to implement so that one workflow engine may request another workflow engine to affect the selection, instantiation and enactment of known process definitions by that other engine. The requesting workflow engine should (optionally) also be able to receive back status information and the results of the enactment of the process definition.

WfMC identifies eight levels of interoperability [26]. The levels are distinguished by the architectural and consequent operational characteristics of implementations of workflow engines. The levels of interoperability are:

Level 1: No interoperability

Level 2: Coexistence

Level 3: Unique Gateways

Level 4: Limited Common Application Programming Interfaces (API) subset

Level 5: Complete workflow API

Level 6: Shared definition formats

Level 7: Protocol Compatibility

Level 8: Common Look and Feel utilities

These levels are sorted from low interoperability degree to high, according to the aspects of data communication, function accessing, process definition format and visual appearance etc.

In addition, this specification also defines four models of interoperability, i.e., chained processes, nested synchronous sub-process, event synchronised sub-process, nested sub-process (Polling/Deferred Synchronous).

These four models represent the basic workflow interoperability forms.

- Chained processes model allows a workflow engine to trigger the creation and enactment of a sub-process instance on another workflow engine. In this scenario, the process instance on the invoking engine takes no further interest in the newly created sub-process instance.
- Nested synchronous sub-process model allows the process instance initiation triggered by a different workflow engine. This model works in the scenario that the activity on the invoking workflow engine remains active until the sub-process terminates and allows forward enactment of the activity instance. Synchronisation is achieved by notifying specific attributes changes of the process instance or in the state of the sub-process instance.
- Event synchronised sub-process model also allows the process instance initiation triggered by a different workflow engine. Yet, such initiation is limited in the scenario that the triggering of events arises due to a sub-process being aborted by its enacting workflow or as part of defined check-point logic between two process instances being enacted on separate workflow engines.
- Nested sub-process (polling) model allows the process instance initiation triggered by a different workflow engine in the following scenario: the invoking workflow engine carries on with the enactment of the process instance that invoked the sub-process until it reaches a rendezvous point. At this stage, it polls the enacting workflow engine to determine when the sub-process has reached its completion.

These forms may further compose with each other to perform more complicated interpretability cases.

2.2 Conventional Business Process Modelling

2.2.1 Petri net based workflow modelling

Petri nets have been first introduced by Carl Petri in his thesis for modelling concurrent behaviours of a distributed system. A Petri net is a bipartite graph whose nodes can be distinguished in places and transitions, which are graphically represented by circles and rectangles, respectively. A series of variations are derived from the primitive Petri net model. Coloured Petri nets use colours to category tokens, and thereby one coloured Petri net can simulate the execution behaviours of multiple types of instances. Timed

Petri nets assign execution period to each transition and therefore can calculate the period distribution of the whole completion time.

Based on Petri net theoretical foundation, van der Aalst proposed a workflow net model [4]. This workflow net model imports two new places, i.e., the starting place *i* and the ending place *o*, to guarantee that the modelled workflow process has unique starting point and ending point. In addition, a workflow net is required to be strong-connected if a transition is added to connect place *i* and *o*. This condition guarantees that a workflow net does not contain any isolated activity or restriction. In particular, a workflow net is intended to describe the behaviour of a single workflow case in isolation. Any case handled by the procedure represented by the workflow net is created, when it enters a workflow management system and is deleted once it has completed. When a workflow net is executed twice, for two cases, then the second case will run through exactly the same process specification as the first.

2.2.2 Process algebra and calculus

Process algebra or process calculus refers to a family of specification techniques that are designed to describe systems of concurrent, communicating components. Currently there exist many languages and dialects, such as Communicating Sequential Processes (CSP) [27], Calculus of Communicating Systems (CCS) [28], and Language of Temporal Ordering Specification (LOTOS) [29].

All these languages support concealment operators, which are intended to prevent the environment from participating in, or observing actions. These three languages adopt two different mechanisms for the enforcement of their concealment operators. The concealment operators in CSP and LOTOS adopt a concealing-actions model. This model allows the users to explicitly list the events that may occur invisibly, without the participation of the environment. The corresponding operator in CCS adopts a restricting-possible-actions model. In this model, if a process is ready to perform some restricted action, this action cannot occur at all.

2.2.3 Event-driven process chain modelling

Event-driven Process Chain (EPC) [30] diagrams illustrate business process workflows, and are an important component of the SAP R/3 modelling concepts for business engineering. EPC diagrams use graphical symbols to show the control flow structure of a business process as a chain of events and functions. Though EPC is based on the concepts of stochastic networks and Petri nets, it does not rigidly distinguish between output flows and control flows or between places and transitions.

The building blocks used in EPC diagrams are:

- *Functions*, which are the basic building blocks of the diagram. Each function corresponds to an executed activity. In the EPC graph, a function is represented as rounded rectangle
- *Events*, which occur before and/or after a function is executed. Functions are linked by events. In the EPC graph an event is represented as hexagon.
- *Connectors*, which associate activities and events. There are three types of connectors: AND, OR, and exclusive OR (XOR). In the EPC graph, a connector is represented as a circle.
- *Organisation units* determine which person or organisation within the structure of an enterprise is responsible for a specific function. In the EPC graph, an organisation unit is represented as an ellipse with a vertical line.

The EPC chain model simulates the execution of a business process as a series of events, and such an event may trigger a function, as well a function may lead to an event. Meanwhile, an event may involve only one or more processes to fulfil, but process is unique for one event. As for function, its data may be included in one or more information resources, while an organisation unit is only responsible for one specific function.

Due to the simplicity and the easy-to-understand notations of the EPC chain model, it was widely accepted as a technique to denote business processes. In 1990s, an extended version of EPC model was integrated within ARIS toolset.

2.2.4 Pattern based workflow modelling approaches

The workflow patterns aimed at providing a systematic and practical approach to dealing with the diversity of languages for control-flow specification [31]. The initiative took the state of the art in workflow management systems as a starting point and documented a collection of twenty patterns, predominantly derived from constructs supported by these systems. The patterns provided abstractions of these constructs as they were presented in a language-independent format. The patterns consist of a description of the essence of the control-flow dependency to be captured, possible synonyms, examples of concrete business scenarios requiring the control-flow dependency, and, for the more complex ones, typical realisation problems and solutions to these problems.

In this work, totally twenty patterns were proposed in six categories, viz., basic control flow patterns, advanced branching / synchronisation patterns, structural patterns, cancellation patterns, state-based patterns and patterns involving multiple instances.

These workflow patterns formed the starting point for the development of Yet Another Workflow Language (YAWL) [32]. This language extends Petri nets with constructs for dealing with some of the patterns in a more straightforward manner. Though based on Petri nets, its formal semantics is described as a transition system. Beyond a modelling language, now YAWL also encompasses an implementation environment, which is detailed in reference [32].

2.2.5 XPDL and BPMN

XML Process Definition Language (XPDL) (newest version 2.0) [33] is intended to be used as a file format for the Business Process Modelling Notation (BPMN) [18] from the consortium of Business Process Management Initiative (BPMI). XPDL and the BPMN specifications address the same modelling problem from different perspectives. XPDL provides an XML file format that can be used to interchange process models between different tools. BPMN provides a set of graphical notations to facilitate human communication between business users and technical users, of complex business processes.

One of the key elements in XPDL is its extensibility to handle information used by a variety of different tools. In addition, XPDL is designed as a generic construct that support vendor-specific attributes for use within the common representation. Thus, XPDL is tightly coupled with the WfMC workflow reference model Interface I, which is defined as a common interchange format and supports the transfer of process definitions between separate products.

XPDL is conceived of as a graph-structured language with additional concepts to handle blocks. XPDL only allows process definition on the top level. Hence, there are no nested processes. Since workflow relevant data is declared either on the top level, or within a process definition, it is limited to two scope levels. Routing is handled by specification of transitions between activities. The activities in a process can be thought of as the nodes of a directed graph, with the transitions represented by the edges. Conditions associated with the transitions determine at execution time which activity or activities should be executed next.

2.3 Inter-organisational Business Process Modelling

2.3.1 Public-to-Private approach

van der Aalst and Weske have proposed a “top-down” workflow modelling approach, i.e., public-to-private approach [34], to define and coordinate inter-organisational workflows, with the help of their Petri net based WF-net model. In this approach, a collaborative business process is specified and partitioned according to the organisations involved by private refinement of the parts based on a notion of inheritance.

This approach consists of three steps:

- Firstly, the involved organisations agree on a common public workflow, which serves as a contract between these organisations;
- Secondly, each task of the public workflow is mapped onto one of the domains, i.e., organisations. Each domain is responsible for a part of the public workflow, referred to as its public part;
- Finally, each domain can now make use of its autonomy to create a private workflow. To satisfy the correctness of the overall inter-organisational workflow, however, each domain may only choose a private workflow which is a subclass of its public part.

The inter-organisational workflow corresponding to the partitioned public workflow serves only as an agreement, i.e., it is the business-to-business protocol that the business partners agreed upon and not the real workflow as it is executed. The workflow description that is used to actually execute the workflow within one of the domains is called the private workflow. The private workflow typically contains several tasks, which are only of local interest. Each partner has a copy of the workflow process description. This approach guarantees that the private workflows of the participating organisations satisfy the public workflow as agreed upon.

Their previous work of WF-net based on Petri net [4] has been deployed to model both public and private workflows. An inter-organisational workflow is modelled in the form of an inter-organisational WF-net, which consists of a set of WF-nets, a set of channels, a set of methods, and a channel flow relation. This WF-net is a high-level representation of the domains and their dependencies; its semantics are given in terms of a labelled Place/Transition (P/T) net by taking the union of all WF-nets, adding a place for each channel, connecting transitions to these newly added places. Projection inheritance is used as a formal link between the public parts of the domains and the private workflows, which are actually executed. Transformation rules are proposed to create specialisations of a given WF-net, making use of the fact that applying these rules to a given WF-net is guaranteed to create a subclass of that WF-net.

Though this public-to-private approach focuses on supporting organisational autonomy and privacy with public and private views, yet it fails to provide a further refined granularity of privacy protection. The public view provides the same content to all participating organisations, therefore, it neutralises the different perceptions of all participating organisations. In addition, this public-to-private approach follows a top-down modelling scheme, which assumes a pre-existing general diagram of the collaboration and a focal organisation in charge of modelling the collaborative business process. These assumptions are sometimes too restrictive in loosely coupled collaboration environments.

2.3.2 Workflow view models

Chiu et al. [35] adapted the concepts of views from databases to workflows, and employed a virtual workflow view for the inter-organisational collaboration instead of the real instance, to hide internal information. In their workflow view model, only information necessary for process enactment, enforcement and monitoring of the service is made available to both parties, in a fully controlled and comprehensive manner.

Moreover, each party needs only minor, or none, modification to its own workflow to successfully arrive at a commonly agreed and interoperable interface. This kind of adaptation is only required on the first interaction, and is subsequently reusable, unless their respective workflows are changed drastically. Because an organisation is probably interoperating with many other different organisations, different views of a workflow can be presented to different organisations according to their respective requirements. Liu and Shen [36] presented an algorithm to construct a process view from a given workflow, but did not discuss its correctness with respect to inter-organisational workflows.

Schulz and Orlowska [37] have also modelled cross-organisational workflows with a workflow view approach, to tackle the communication between the entities of a view-based workflow model. This workflow view model is provided for a multi-granular privacy for workflows, given they believe that a workflow view needs to be protected from unauthorised interaction. Finally, they used a Petri-net based representation as the basis for consideration of state dependencies between tasks in a workflow and the adjacent task in a workflow view.

Problems to be encountered on the way to workflow interoperability mainly include autonomy of local workflow processing, confidentiality that prevents complete view of local workflow [38], and especially flexibility that needs no definition of a global workflow that defines cooperation between local workflows.

In [39], Perrin and Godart present an approach for process management and coordination based on synchronisation points between process services. This approach provides more flexibility in order to allow partners to personalise their internal processes without affecting the cooperation.

2.3.3 Business Process Modelling Language (BPML)

BPML [40] 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].

In BPML, collaborative business process models can contain as many, or as few, execution details as the process designer considers necessary to share between partners or business units. BPML also supports a high level of abstraction, in which execution details can be hidden. This approach promotes collaboration among business partners. A model of the process can be shared even as certain implementation details can be left to each partner. This is a vital breakthrough for true interoperability, taking integration one-step further, to true process collaboration.

BPML copes with the complexity of collaboration among business partners by permitting arbitrarily large numbers of participants to play a role in end-to-end business process designs, at any level of nesting and concurrency. Participants communicate freely with one another, right across the value chain. BPML achieves this by separating and interleaving control flow, data flow and event flow, while adding complementary and orthogonal design capabilities for business rules, security roles and transactions. Messages exchanged between participants contain the process data that is needed for process collaboration.

2.3.4 Business Process Execution Language for Web Services (WS-BPEL)

The Business Process Execution Language for Web Services (WS-BPEL or previously BPEL4WS) [9] is a language for specifying business processes behaviour 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) [41]. A WS-BPEL process allows the definition of two types of business processes, viz., abstract process and executable process. The first type defines the business protocol role and describes its public aspects. The second type defines the logic and state of the business process by providing the sequence of the Web service interactions, which are conducted at the site of each business partner. Moreover, WS-BPEL defines a set of primitive activities, such as “invoke” to invoke Web service operations, “reply” to send a response, etc. These primitive activities can be combined into more complex primitives using any of the structure activities, such as “sequence”, “flow”, “while” etc.

However, WS-BPEL does not support many concepts that are paramount for inter-organisational collaboration.

First, it does not comprise the rich concepts of existing workflow management systems, such as manual activities and applications, nor addresses the integration with them. Because it uses Web services exclusively, it is limited to call other types of services like XML services, databases, etc.

Second, in the context of collaboration, it does not support the heterogeneity of partners. 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, WS-BPEL does not cater for non-Web service interactions and the notion of independent activities, as everything in a WS-BPEL process is a Web service operation.

Finally, WS-BPEL has not yet provided 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 are able to send messages to each other but there is no standard way to specify those flows.

2.3.5 ebXML and RosettaNet PIP

The Electronic Business XML (ebXML [10]) consortium defined a comprehensive set of specifications for XML document exchange among trading partners. It is motivated to provide a framework in which EDI's substantial investments in business processes can be preserved in an architecture that exploits XML's new technical capabilities.

The technical infrastructure of ebXML encompasses the following major elements:

- *Messaging Service* provides a standard way to exchange business messages between organisations.
- *Registry* is a database that stores information about items, which actually reside in a repository, and thereby supports doing business electronically.
- *The Collaboration Protocol Profile* (CPP) provides the definition (DTD and W3C XML schema) of an XML document that specifies the details of how an organisation is able to conduct business electronically.

- *Business Process Specification Schema* (BPSS) provides the definition (in the form of an XML DTD) of an XML document that describes how an organisation conducts its business.

In the ebXML world, an organisation uses an ebXML Registry to make its services available through the Internet. This ebXML Registry works similar to a UDDI registry but contains much more detailed information about the service being provided. To advertise its services, the organisation has to submit a 'business profile' including implementation details and reference links, as well as the business scenarios that it supports. The ebXML registry verifies the business scenario and then makes the business scenario available as part of the registry. Therefore, potential uses of the business scenario can find information about the scenario by querying the registry [42].

However, a business process in the context of ebXML is different from the context of workflow management. The former focuses on exchange of business documents while the latter focuses on the flow of both control and data.

The Partner Interface Process (PIP) [11] blueprints by RosettaNet specify interactions using UML activity diagrams for the Business Operational View (BOV) and UML sequence diagrams for the Functional Service View (FSV) in addition to DTDs for data exchange. A PIP process can be viewed as a drill-down from the high-level vision of how the entities will interact, to a low-level functional view of the business processes. A PIP process details how to implement a collaborative business process between trading partners, with BOV and IFV from two layers. BOV, also known as the Action Layer, describes the flow of business interactions between business entities; FSV, also known as the Transaction Layer, specifies the business transactions between entities in terms of message exchanges between RossetaNet services; Implementation Framework View (IFV), also known as the Service Layer, defines communication protocol and message format requirements based on BOV and FSV. Once PIP processes have been defined, these processes are collected and published by RosettaNet in a PIP Directory.

PIP processes fit into seven clusters, or groups of core business processes, that represent the backbone of the trading network, such as "Partner Product and Service Review", "Product Information", "Order Management", "Inventory Management" etc.

Each cluster is broken down into segments which are cross-enterprise processes involving more than one type of trading partners. Within each segment are individual PIPs. However, RosettaNet PIP is primarily focusing on electronic markets or supply chains with long-lasting pre-specified relationships between parties with one party (such as the market maker) imposing rigid business rules.

2.4 Inter-Organisational Workflow Implementation Technologies

2.4.1 Workflow coordination structure

van der Aalst has classified the workflow coordination structure into five types, viz. capacity sharing, chained execution, subcontracting, case transfer and loosely coupled [43].

- In the *capacity sharing* type, external resources under the control of one workflow manager execute tasks.
- In the *chained execution* type, the process is divided into subsequent phases and each business partner takes care of one phase. [43]. This form of interoperability is only useful for applications where the process is composed of sequentially ordered parts. Nevertheless, it was generalised into an approach to distributed workflow execution where parts are inter-mixed [44].
- In the *subcontracting* type, a sub-process is executed by another organisation. There is one business partner, which subcontracts sub processes to other business partners.
- In the *case transfer* type, 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*. [4].
- In the *loosely coupled* type, each partner takes care of a specified part of the process that may be active in parallel.

Different from the WfMC interoperability model introduced in Section 2.1.3, these five types of coordination structures focus on the workflow run time execution, especially the workflow control at instance level. These five types list the basic instance coordination scenarios from a relatively high perspective, yet how to realise them through the cooperation of practical workflow management systems is a question of further research and development in this area.

2.4.2 Protocols and infrastructure technologies

2.4.2.1 Wf-XML

The initial mandate of WfMC was to create interoperability between workflow engines of different vendors. Wf-XML [45] is the most recent specification for workflow interoperability, and its functionality extends beyond the coupling of workflow engines to the remote invocation of a process service by other clients. Wf-XML aims to facilitate the remote invocation and manipulation of processes through a lightweight interface that is modelled as a combination of HTTP and SOAP protocols [24].

Wf-XML categories resources into five types, viz., Service Registry, Factory, Instance, Activity and Observer. Well-defined service interfaces permit interaction with these resources and their properties. Wf-XML extends the basic HTTP operators with commands that are more specific to the world of workflows and Web services. In the context of workflow, the main service is a process factory, which can spawn process instances upon requests. Wf-XML describes an interface to an asynchronous process over Web applications.

To summary, Wf-XML represents a lightweight protocol for the discovery and invocation of processes that are provided by a remote process, it is suitable for the interaction with virtually any processes that an organisation might wish to expose, because clients can discover the interaction options available to them as they go along.

2.4.2.2 Web Service description and flow description

The Web services paradigm has emerged as a response to the shift in the IT landscape away from isolated, tightly coupled systems to a highly distributed, heterogeneous environment. Compared with the traditional Client/Server architecture, Web services behave as independent entities, each having a distinct responsibility within the system and each specifying its own internal behaviour [42]. Since Web services have no dependency on each other, we can use them to connect technically diverse systems for the purpose of utilising the functionalities that may already exist either within, or outside an organisation. To specify inter-organisational workflows of the platform of

Web service, lots of efforts have been made during recent years and many languages have been proposed.

Web Service Description Language (WSDL) [46] 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.

Web Service Flow Language (WSFL) [47] is built on top of WSDL and can be used to refine a WSDL specification or compose workflow fragments. It supports workflow fragment integration with heterogeneous data structures by using XML Path Language (XPath) expressions.

XLANG [41] refines WSDL service specification with XLANG service behaviours, and allows the WSDL services composition specification. It uses the notion of blocks to support message handling, timing and exception handling. It also supports transactions with compensation. But transactions are not allowed to span workflow fragments.

2.4.2.3 Web Service choreography interface

Web Service Choreography Interface (WSCI) [48] describes how Web service operations, such as those defined by WSDL can be choreographed in the context of a message exchange in which the Web service participates. Interactions between services, either in a business context or not, always follow and implement choreographed message exchanges (processes). WSCI is the first step towards enabling the mapping of services as the components that realise those processes. WSCI also describes how the choreography of these operations should expose relevant information, such as message correlation, exception handling, transaction description and dynamic participation capabilities. WSCI does not distinguish that Web services are from different companies or not, therefore, it can equally well describe interfaces of components that represent internal organisational units or other applications within the enterprise. Again, WSCI does not address the definition of the process driving the message exchange or the definition of the internal behaviour of each Web service.

2.4.3 Service interaction pattern

The work of service interaction patterns [49] is advertised as a shift towards a reference framework for service based collaborative business processes. This work aims to exploit the service choreography and orchestration behaviours in multi-party collaboration environments. Especially, this work investigates into the nature of service interactions in collaborative business processes, where a number of parties, each with its own internal processes, need to interact with one another according to certain pre-agreed rules.

The collected patterns have been derived and extrapolated from insights into real-scale B2B transaction processing, choreography and orchestration examples including use cases that gathered by standardisation committees. The proposed patterns are structured into four groups, viz., single-transmission bilateral interaction patterns, single-transmission multilateral interaction patterns, multi-transmission interaction patterns and routing patterns, derived from the following dimensions:

- The maximum number of parties involved in an exchange, which may be either two (bilateral interactions, covering both one-way and two-way interactions) or unbounded (multilateral interactions).
- The maximum number of exchanges between two parties involved in a given interaction, which may be either two (in which case we use the term single-transmission interactions) or unbounded (multi-transmission interactions).
- In the case of two-way interactions (or aggregations thereof), it is the receiver of the ”response” is necessarily the same as the sender of the “request” (round-trip interactions) or not (routed interactions).

This whole collection of patterns consolidates the nature of service interactions through generalised functional classification. Especially for incorporating Web service technologies to business process management, the service interaction patterns play a very important role. When these involved Web services belong to different organisations, the orchestration among Web services would particularly benefit from these patterns, in terms of messaging synchronisation.

2.4.4 Security and privacy

Web Service Security (WS-Security) [50] developed by OASIS, is an extension to SOAP that can be used to implement integrity and confidentiality. WS-Security addresses a fundamental need for end-to-end application level security, since there is no way to realise the encryption from the initiating application to the ultimate receiver with the hop-by-hop security mechanism in current HTTPS. From technical point of view, WS-Security defines a SOAP header block (called security) which can carry a signature. This block header includes the security information and indicates what elements can be found there and how they must be processed [42].

Web Service Policy (WS-Policy) [51] defines a framework and a model for the expression of specific policy requirements, such as security, quality of service etc. WS-Policy provides a flexible and extensible grammar for expressing the capabilities, requirements, and general characteristics of entities in an XML Web service based system. In WS-Policy context, a policy is defined as a collection of one or more policy assertions, while assertions express the capabilities and constraints of a particular web service. In addition, some operators, such as ALL, ExactlyOne etc., are used to group these assertions into a comprehensive one.

However, WS-Policy does not define any domain-specific policies. It only provides a framework through which domain-specific policies can be described. The task of describing the actual policies is left to other standards. Thus, at this stage we can anticipate subsequent standards would provide profiles on WS-Policy usage within other common Web service technologies.

2.4.5 Workflow prototypes and projects

Quite a few workflow prototypes and projects are done or being conducted in institutions with regard to inter-organisational business process management. Here, we cite among others COSMOS [52], TOWEC [53], CrossFlow [54], WISE [55, 56], PIEC and MEMO [57].

CrossFlow [54] project covers some issues related with business processes across organisational boundaries. A contract-based approach is proposed to specify the business relationships between the collaborating organisations. Within this contractual basis, inter-organisational processes can be defined and performed. However, this approach does not support arbitrary public processes, and no standard definition language and semantics is provided for the enforcement of contracts between two organisations. In addition, all organisations involved are required to use the same software for contract enforcement.

The WISE [55, 56] (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 organisations. This architecture provides means to define, enact, and monitor business processes within virtual organisation alliances, 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 modelling tool to define and monitor processes, and a catalogue 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. As WISE is platform independent, the accessibility over the Internet makes this solution scalable and open. But service descriptions and the service catalogue are not in line with general standards. Moreover, the centralised workflow engine inhibits dynamic selection and exchange of partners since all participants have to comply with stipulated interfaces.

Similar work has been carried out by van der Huevel and Weigand and is described in Process-Integration for Electronic Commerce (PIEC) project and MEdiating and MOnitoring for electronic commerce (MEMO) ESPRIT project [57]. Heuvel and Weigand propose a contract model to represent formal and informal communication structures that are used to coordinate business workflows. The approach assumes workflows are private within organisations. A contract exposes selected activities, and links them to activities of other participants. The contract then specifies the messages that can be exchanged between them. However, the authors did not argue how fine-grained the communication between participants can be coordinated with this approach.

2.4.6 Commercial workflow products

A lot of workflow products were developed in late 90s, such as IBM FlowMark, ActionFlow, FileNet, HandySoft's BizFlow, Lotus Workflow etc. As today's world embraces the global collaboration age, more collaboration oriented or collaboration-supportive workflow management systems emerge out and are taking over the main market of workflow software products. Among these modern workflow management systems, we cite typical products, viz., IBM WebSphere MQ Workflow, BEA WebLogic Workflow Integration and SAP Webflow, to display the support from industry side for business process management.

2.4.6.1 IBM WebSphere MQ workflow

IBM WebSphere MQ Workflow evolves from two previous products of IBM, i.e., FlowMark and MQSeries. Now, IBM WebSphere MQ Workflow is entirely built on top of WebSphere MQ application and messaging infrastructure, including communications between servers and clients. With a true object-oriented design, WebSphere MQ Workflow offers a high level of re-usability. It excels on process management and organisation modelling. Activities can be defined or implemented with the assistance of ActiveX objects and Java APIs and using Web Client technology for Rapid Application Development (RAD), as well as automated generation of JSP files based on process definitions. With IBM WebSphere Business Integration Modeller and Monitor (former Holosofx BPM tool), MQ Workflow has a strong business process analysis, simulation and development tool in one box, as well as a comprehensive monitoring environment. In addition, it has a user friendly interface on the Web environment [5].

From the inter-organisational business process management perspective, WebSphere MQ Workflow provides a component for WebSphere Partner Gateway to support inter-organisational collaboration. At technical level, Partner Gateway provides the ability to connect to back-end integration systems. Partner Gateway of Enterprise and Advanced Editions provide both file-based integration and integration over HTTP, HTTPS, and Java Messaging Service (JMS) transports. Partner Gateway acts as the entry point for documents coming into the enterprise. It validates, transforms, and processes the documents, based on their type, and passes them on to a back-end system that integrates the information with other applications. The back-end system can be a business process integration server, such as WebSphere Process Server, an integration broker such as

WebSphere InterChange Server or WebSphere Business Integration Message Broker, or it can be a customer-developed system.

Consequently, Partner Gateway empowers an organisation with the flexibility of transport and message formats for collaboration with a number of trading partners. This type of organisation can act as a Community Manager (of its own smaller community) as well as a community participant in a larger community. Furthermore, a component of Process Choreographer is added to coordinate business processes in BPEL format. Yet, MQ Workflow focuses on communication infrastructure more than collaboration management. MQ Workflow does not put much concentration on managing the flexible and dynamic partnership between these community members, and therefore it merely works as a communicating platform for B2B collaborations.

2.4.6.2 BEA WebLogic workflow integration

BEA WebLogic Workflow Integration [58] is a pure java based enterprise level integration software suite. This suite encompasses a Studio for workflow designing, developing, and monitoring, as well as a Worklist for users to interact with running workflows.

With the Studio, business process specialists can develop both public processes (also known as collaborative processes in B2B environment) and private business processes. B2B integration plug-in is particularly responsible for developing public processes with the Studio. Public processes enable the interaction of trading partners in a collaborative, B2B arrangement by choreographing the exchange of business messages between them. Private processes are internal to an organisation. They are not exposed outside of the enterprise, and customers or trading partners do not interact with them directly. In addition, this Studio also enables specialists to monitor the ongoing effectiveness of business processes by viewing processes in real time, and by collecting runtime statistics for reports. These data can be used to evaluate processes, optimise performance and throughput, and increase uptime.

BEA WebLogic Workflow Integration encourages users to interact with running business processes through a Web based Worklist interface. Using the Worklist, users can handle business process tasks assigned to them, such as making a decision about a customer's credit limit, or they can respond to messages from a process. The Studio supports both top-down modelling and bottom-up modelling for workflow design, yet it prefers the former in nature. In this top-down mechanism, the workflow definition process involves moving from mapping out a high-level graphical representation of the basic activities and logic that the application fulfils, to drilling down to deeper levels of detailed specifications.

However, WebLogic Workflow Integration does not emphasise the granularity of organisation's perception. The public process and the private process only provide a primitive privacy protection mechanism. This mechanism fails to characterise the diverse partnership between collaborating organisations, and the different perception of different organisations.

2.4.6.3 Oracle BPEL process manager

Oracle BPEL Process Manager [7] is one component of Oracle Fusion Middleware family of products, which offers a comprehensive solution for business integration and collaboration. Oracle BPEL Process Manager is particularly designed to help organisations model, deploy and manage BPEL processes.

Completely based on Web services, BPEL Process Manager enables organisations to orchestrate and execute Web services and business processes. It comprises a BPEL modeller, a scalable BPEL engine, an extensible WSDL binding framework, a monitoring console and a set of built-in integration services (transformation, user task, java embedding). With the native support for BPEL, Oracle BPEL Process Manager reduces the cost and complexity of integration projects while increasing their strategic value, as BPEL is becoming popular in industry domain.

The BPEL Process Manager administration console allows users to deploy user-defined BPEL processes, test and debug flows, collect data on transactions, and trace both completed histories and those in progress. Users can use the graphical interface to visually inspect a business flow and drill down to the underlying XML document that encapsulates the transaction.

In all, the Oracle Fusion Middleware suite, including Oracle BPEL Process Manager, offers a service-oriented architecture for business integration and collaboration. With these products, Oracle represents a glimpse of next-generation enterprise applications. As an active pioneer of technology revolution, Oracle currently moves forward to develop products for Grid computing platform.

As the nature of BPEL language, a BPEL process models a collaborative business process from the perspective of a pivot organisation. Consequently, such a BPEL process can only represent the interactions of this pivot organisation in the collaboration, yet fails to cover the interactions beyond this pivot organisation. Thus, the scope of a BPEL process is relatively limited. In the research reported in this thesis, standard BPEL is extended to include the interactions beyond neighbouring organisations.

2.4.6.4 SAP business workflow and WebFlow

SAP Business Workflow [6] is the R/3 tool for handling the process automation within R/3 or between R/3 systems and other systems involved in the business process. The different R/3 applications supply standard workflows for the commonly occurring processes. Once these workflow templates are activated, they are ready for immediate use. A complete set of workflow tools, including the workflow editor and workflow generation wizard, are provided to enhance these standard workflows or create new workflows. These tools are complemented with transactions for monitoring, tracking and the statistical analysis of the processes.

WebFlow caters for the e-business functionality of SAP Business Workflow. It is the driving force behind the Internet processes in mySAP.com, ensuring the delivery of the right information to the right person at the right time while enabling the swift change-life-cycle of e-Business processes. With a true object-oriented design, WebFlow offers a high level of re-usability. WebFlow Engine also provides a number of tools, such as Workflow Builder for defining workflows and WebFlow Functions for workflow execution between different organisations, and Business Workflow Explorer for monitoring operations.

However, WebFlow is not a close follower of current Web service trend, since it still sticks to commerce XML (cXML) and EDI data formats. This point limits WebFlow's application and extensibility towards external services or products.

2.5 Workflow Monitoring and Instance Correspondence

Workflow monitoring and analysing are two basic functions as defined in WfMC's workflow management system reference model. The work reported in this thesis also covers the topics of inter-organisational workflow tracking and instance correspondence. The following content mentions some classical methods in related fields.

2.5.1 CPM & PERT charts

Program Evaluation and Review Technique (PERT) and Critical Path Method (CPM) are two classical graphical system analysis approaches, especially for project management and process analysis.

In 1957, CPM was developed as a net model, which models the activities and events of a project as a network. Activities are depicted as nodes on the network, and events that signify the beginning or ending of activities are depicted as arcs or lines between the nodes. While CPM is easy to understand and use, it does not consider the time variations that can have a great impact on the completion time of a complex project. Therefore, CPM fairly yet merely fits routine projects with minimal uncertainty in the project completion times.

PERT is a network model that allows for randomness in activity completion times. PERT was developed in the late 1950's for the US Navy's Polaris project that has thousands of contractors. It has the potential to reduce both time and cost required to complete a project. Compared with CPM, PERT provides expected project completion times, and probability of completion before a specified date. However, the activity time estimates are somewhat subjective and depend on judgement, therefore, could contain bias in the estimate. Even if the activity times are well estimated, PERT assumes a beta distribution for these time estimates, but the actual distribution may be different.

2.5.2 Workflow monitoring

Regarding workflow monitoring, WfMC is the first association to formally define that the monitoring function is an indispensable part of any workflow management system, and specified a set of Application Programming Interfaces (APIs).

Wonchang Hur, Hyerim Bae and Suk-Ho Kang [59] have considered the customisation of monitoring services in a workflow management system. This work focused on providing a monitoring environment fitted to an individual user's requirements, according to the different responsibilities, roles, and preferences of different users. The whole framework encompasses fours components, viz., data object, analysis method, presentation style, and audit event, to support customisable workflow monitoring. A monitoring template is used to combine these components together to create a workflow monitoring service, which can automatically adapt to different users. This work first considers the flexibility and personalisation of workflow monitoring, yet this work is still confined within intra-organisational boundary without concerning inter-organisational scenarios.

Bastin Tony Roy Savarimuthu, Maryam Purvis and Martin Fleurke [60] investigated in monitoring and controlling of a multi-agent based workflow system, namely JBees. In their workflow system, a simulation agent is used to obtain and log the system performance data, while a controlling agent continuously monitors the data, and logs appropriate warning messages, as well as displays warning message to users. The monitoring and controlling agents help in optimising the workflow, as well as improve the system effectiveness. Minhong Wang and Huaiqing Wang [61] also contributed in workflow monitoring in an intelligent agent environment.

2.5.3 Instance correspondence

A complex workflow process may contain several sub workflow processes, and therefore an instance of such a composite workflow process naturally corresponds to multiple instances of the sub workflow processes. Despite a complete workflow management system is supposed to handle multiple workflow instantiation, yet many workflow management systems still lack support for this feature.

2.5.3.1 Pattern based approaches

Guabtni and Charoy have extended the multiple instantiation patterns described in [62], and proposed a set based multiple instantiation approach. The essence of their work is to define sets of tasks in a dynamic workflow process. Each set contains activities that must be executed multiple times, while each set is governed by constraints making it possible to supervise the multiple executions. These sets can be nested or even overlap.

Guabtni and Charoy classified multiple workflow instantiations into two scenarios, viz., parallel and iterative multiple instantiations. According to the two multiple instantiation scenarios, a parallel instance set and an iterative instance set are proposed in their work.

- The parallel instance set is used for activities that are executed multiple times in parallel. This set corresponds to the patterns of multiple instantiation without synchronisation, multiple instantiation with a priori design time knowledge, multiple instances with a priori runtime knowledge, and multiple instantiation without a priori runtime knowledge, which are defined in [31]. The goal of this set is to provide an answer to the problem of activities that have to be executed several times. There is no constraint on the parallel instance set of activities. Any activity that has not been yet started can be selected to participate in a parallel instance set.

- The iterative instance set is used for activities that are executed multiple times in sequence. This set corresponds to the patterns of arbitrary cycles or loops, and implicit termination. The goal of this set is to allow the repetition of a set of activities until some condition is evaluated to true. The circles in a workflow process are handled by re-instantiating activities as long as it is needed. Thus, iteration is not the re-execution of a set of activities but the successive execution of copies of these activities. This feature benefits the data management and simplifies the specification of constraints.

In this way, this method provides an easy to understand solution for multiple instantiation without importing any additional control structures or operators to the workflow model. Moreover, this set based multiple instantiation allows nested and overlapped sets. This work has been integrated with an actual workflow model in their Bonita project. However, their research is conducted at task or sub process level, without concerns of inter-organisational collaborations.

Zhou, Shi and Ye [63] also did pattern based modelling for multiple instances of workflow activities. Dumas and ter Hofstede have discussed this topic using UML activity diagrams [64]. And later they extended their work to service interactions, and gave a set of thirteen service interaction patterns [49], including single-transmission bilateral interaction patterns, single-transmission multilateral interaction patterns, multi-transmission interaction patterns and routing patterns.

2.5.3.2 Message correlation in WS-BPEL

WS-BPEL (previously BPEL4WS) [9] uses a property based message correlation mechanism to handle instance correspondence in collaborative business process environment. This property based message correlation mechanism allows processes to participate in stateful conversations. It can be used to match return or known customers to long-running business processes. When a message arrives for a Web service, which has been implemented using BPEL, that message must be delivered somewhere – either to a new or an existing instance of the process. Here, message correlation exactly focuses on determining to which conversation a message belongs, or locating/instantiating the proper instance in BPEL terms.

Unlike traditional distributed object systems, instances in WS-BPEL are identified not by an explicit "instance ID" concept, but by one or more sets of key data fields within the exchanged messages. For example, an order number may be used to identify a particular instance of a process within an order fulfilment system. In BPEL terms, these collections of data fields which identify the process instance are known as correlation sets.

In Web service environment, each BPEL correlation set has a name associated with it, and is composed of WSDL-defined properties. A property is a named, typed data element which is defined within a WSDL document, and whose value is extracted from an instance of a WSDL message by applying a message-specific XPath expression. In WSDL, a propertyAlias defines each such mapping. The mappings are message-specific; hence a single property can have multiple propertyAliases associated with it. Thus, the properties and propertyAliases together enable a user to refer a single, logical piece of information in a consistent way, even if it might appear in different forms across a set of messages.

Multiple correlation sets, some of which are initialised and some of which are used for comparison, can appear on a single activity. The current BPEL specification does not define the semantics of locating an instance based on multiple correlation sets. Moreover, WS-BPEL defines a business process in terms of a pivot organisation. This results in that a WS-BPEL business process only represents the interaction behaviours of the pivot organisation with its neighbouring organisations. This feature limits its application for complex business collaborations, which are likely to include interactions beyond neighbouring organisations.

2.6 Summary

This chapter has introduced related work from the far past to the recent in detail. From these literatures, it is clear to see that collaborative business process management urgently calls for novel technologies and solutions to achieve efficient and effective process modelling and execution. Autonomy, privacy and cooperation of business collaborations nowadays are continuously pressing, yet there is very limited support from conventional business process management approaches. The later inter-organisational business process management approaches provide some primitive supports, especially on issues like communication protocols and infrastructures. Some approaches apply public and private workflow views to protect the business privacy during collaborations, yet a finer-granulated visibility mechanism is expected to differentiate the perception of different organisations according to the diverse partnerships.

Chapter 3

Relative Workflow Methodology

From the literature review in Chapter 2, we see that business processes form the backbone of organisations. Effective business process management is an eternal research topic for organisations in order to stay competitive. In the last decade, tremendous efforts were placed on research, standardisation and development of workflow systems, which result in a mature business process technology for automating and controlling business processes. However, the traditional workflow management has been emphasising on homogeneous environments within the boundary of a single organisation. In today's business climate, many organisations form dynamic partnerships to effectively deal with market requirements. Different types of collaboration may be required. For examples, an organisation may concentrate on its core business and outsource secondary activities to other organisations in a virtual organisation alliance; growing complexity of products may require co-makership relationships between organisations in collective production; value chains may require a tight cooperation between organisations participating in these chains.

These new requirements drive business process management technology to thrust more supports in cooperation, autonomy and openness. Current business process technology is expected to deal effectively with cooperation among participating organisations in heterogeneous business process environments. Furthermore, it is also expected to support the autonomy and privacy of participating organisations, and flexibility in dynamic formation of new and dismantling of existing collaborations. In this chapter, an organisation-oriented business process model, called relative workflow model is presented. This relative workflow model is based on an organisation-oriented perspective, and uses a visibility control mechanism to enable the organisational autonomy and privacy protection. This relative workflow model establishes the theoretical foundation for the research reported in this thesis. In addition, the procedure of generating a collaborative business process in the form of a relative workflow process from the original business contracts is also presented in this chapter.

The remainder of this chapter is organised as follows: Section 3.1 analyses the requirements for collaborative business process management by identifying three problems, and discusses these problems with a motivating example; Section 3.2 presents the details of our relative workflow model; Section 3.3 portrays the procedures for generating relative workflow processes; Section 3.4 justifies the relative workflow model in terms of information sufficiency and necessity.

3.1 Requirement Analysis

3.1.1 Identifying problems

Traditional inter-organisational workflow design approaches streamline business processes contributing to a common business goal, yet belonging to different organisations, into a public business process. As discussed earlier, this procedure has the following problems.

The first problem is the autonomy of organisations participating in the collaboration. As discussed in Chapter 2, most inter-organisational workflow modelling approaches assign a third party designer or a focal organisation (main contractor) to determine the collaboration choreography and orchestration of all participating organisations. However, this modelling mode places the other minor participating organisations in a relatively passive role, or a mere member of the poll queue. Due to this reason, these organisations behave in the collaboration the same way as workers do in a pipeline workshop. Nevertheless, driven by globalisation nowadays, organisations expect more for process customisation to react with local markets and national regulations. The challenge for any organisation responding to globalisation is the need to strike the appropriate balance between centralised corporate standards and the autonomy needed to serve local markets [3]. Further, these facts result in an organisation's desire for autonomy including choosing its own partner organisations, defining collaborative business process by itself according to its own collaboration objectives and benefits, etc.

Besides, an appropriate third party designer or a main contractor is always not obtainable, especially in loosely coupled collaboration environments.

The second problem is the coarse granularity of openness. Most inter-organisational business process modelling approaches also compulsorily use a common business process, and all participating organisations share this public business process to conduct the collaboration. However, this public business process based collaboration inevitably results in that either excessive information is to be disclosed or required collaboration information is not provided sufficiently. In the former, some private business information may be unwillingly disclosed to an involved organisation with a distant partnership. In the latter, business processes belonging to involved organisations cannot be integrated seamlessly. A customisable visibility control over inter-organisational business processes is therefore required to balance the information openness and privacy prevention elaborately, as well as to guarantee the maximal autonomy of participating organisations.

The third problem is the poor flexibility and adaptability of pre-determined collaborative business processes. Due to the turbulent and rapidly changing environment, organisations always need to modify their business processes to accommodate these variants. According to the market requirements and changing partnerships, both internal and collaborative business processes may transform into an ad hoc manner. Therefore, an all time reconfigurable collaborative business process is expected to substitute a pre-determined collaborative business process.

3.1.2 Motivating example

Figure 3.1 shows business collaboration among a retailer, a manufacturer, a shipper and a supplier, from a public view. Five intra-organisational business processes and their interactions in between them are shown in the figure. When a 'Product Ordering' process of a retailer sends a product order to a manufacturer, the manufacturer's 'Production' process may hold this order until it has collected enough orders from more than one 'Product Ordering' process for the purpose of batch production. Before it starts producing products, the manufacturer needs to order necessary parts from suppliers, therefore, will interact with the manufacturer's 'Inventory Management' process later for arrival checking and invoice/payment processing. Also, the manufacturer needs to contact shippers to book the delivery of products, and simultaneously checks inventory with the 'Inventory Management' process through the corporate database within the same organisation. Finally, the retailer receives the products from the shipper, and pays the manufacturer.

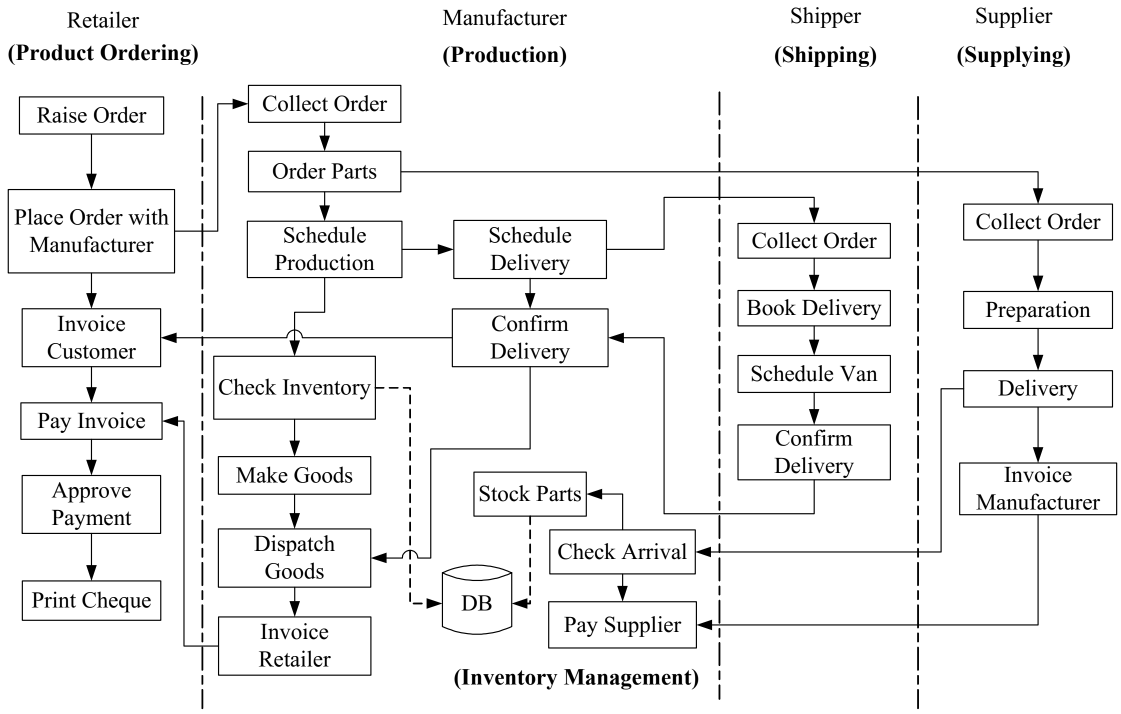

Figure 3.1 Inter-organisational workflow process example (modified from [65])

In this example, all the participating organisations have the global knowledge of the whole collaboration process, which is somehow pre-determined and may be defined by a third party designer or a main contractor such as the manufacturer. Once the collaborative business process has been defined, each participating organisation acts passively and loses more or less its autonomy. It will be difficult for an organisation to change its collaboration structure and behaviours, for instance, to start a new partner relationship or to terminate an existing partner relationship. Besides, the global knowledge of the whole collaboration process gives no chance to define a close or distant partner relationship between participating organisations. For example, from Figure 3.1, we can clearly see that the views from a retailer and a manufacturer on the collaborative process are different. While a manufacturer has a close partner relationship with all other participating organisations, a retailer, however, only has a close partner relationship with a manufacturer via a proper source/supply contract. A retailer may not need to know, and actually should not know the manufacturer's partner relationships, say, with a supplier. At the same time, a retailer may need to have some knowledge about a shipper of the manufacturer so that the tracking on delivery of products may be made possible. We may also need to allow that a manufacturer changes partner relationships with suppliers and shippers for better services. All these are not well supported in the public view approaches.

To address these issues, we believe that business collaboration should be decided from the view of each individual organisation, i.e., an organisation defines its collaboration structure and behaviours by following corresponding contracts with proper partner organisations, and may change them later by updating existing contracts or signing new contracts. In this way, each organisation is empowered with the authority to design its collaboration structure and behaviours in a proactive mode. In addition, the views from different organisations may be different due to the partnerships and privacy reasons.

3.2 Relative Workflow Model

In our context, a collaborative business process consists of several intra-organisational business processes of participating organisations and their interaction. We call these intra-organisational business processes as local workflow processes.

Definition 3.1 *Local Workflow Process.* A local workflow process lp is defined as a directed acyclic graph ($\mathcal{T}$, $\mathcal{R}$), where $\mathcal{T}$ is the set of nodes representing the set of tasks, and $\mathcal{R}\subseteq\mathcal{T}\times\mathcal{T}$ is the set of arcs representing the execution sequence.

Definition 3.2 *Organisation.* An organisation g is defined as a set of local workflow processes $\{lp^1, lp^2, \ldots, lp^m\}$. An individual local workflow process lp^i of g is denoted as $g.lp^i$, $1\leq i\leq m$, m is the number of g's local workflow processes.

In loosely coupled environments, each organisation expects to protect critical or private information of its business processes from disclosing to other organisations. According to the two most important behaviours in the context of collaborative business processes, i.e. workflow tracking and workflow interaction, we define the following three values for the visibility of tasks as listed in Table 3.1.

Table 3.1 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 provide a fundamental visibility control mechanism, and this visibility value table is open for future extension.

Definition 3.3 *Visibility Constraint.* A visibility constraint *vc* is defined as a tuple (t, v), where t denotes a task and $v \in \{$ `Invisible`, `Trackable`, `Contactable` $\}$. A set of visibility constraints $\mathcal{VC}$ defined on a business process *lp* is represented as a set $\{vc{:}(t, v) \mid \forall t\ (t \in lp.\mathcal{T})\}$.

Example 3.1 Based on the aforementioned motivating example, two sets of visibility constraints are given as follows:

$\mathcal{VC}_1$ = { ('Raise Order', `Invisible`), ('Place Order with Manufacturer', `Contactable`), ('Invoice Customer', `Contactable`), ('Pay Invoice', `Contactable`), ('Approve Payment', `Invisible`}), ('Print Cheque', `Invisible`)}.

$\mathcal{VC}_2$ = { ('Collect Order', `Contactable`), ('Order Parts', `Invisible`), ('Schedule Production', `Trackable`), ('Schedule Delivery', `Trackable`), ('Confirm Delivery', `Contactable`), ('Check Inventory', `Invisible`), ('Make Goods', `Trackable`), ('Dispatch Goods', `Trackable`), ('Invoice Retailer', `Contactable`)}.

These two sets are defined on the 'Product Ordering' and 'Production' processes, respectively.

Definition 3.4 *Perception*. A perception $p_{g_0}^{g_1.lp}$ of an organisation g_1's local workflow process *lp* from another organisation g_0 is defined as ($\mathcal{VC}$, $\mathcal{MD}$, f), where

– $\mathcal{VC}$ is a set of visibility constraints defined on $g_1.lp$.

– $\mathcal{MD} \subseteq \mathcal{M} \times$ { `in`, `out` }, is a set of the message descriptions that contains the messages and the passing directions. $\mathcal{M}$ is the set of messages used to represent inter-organisational business activities.

– f: $\mathcal{MD} \rightarrow g_1.lp_{g0}.\mathcal{T}$ is the mapping from $\mathcal{MD}$ to $g_1.lp_{g0}.\mathcal{T}$, and $g_1.lp_{g0}$ is the *perceivable workflow process* of $g_1.lp$ from g_0. Here, a perceivable workflow process represents the perceivable form of a local workflow process for a partner organisation. The generation of $g_1.lp_{g0}$ from $g_1.lp$ will be discussed in the next section.

Example 3.2 Again, based on the aforementioned motivating example, the perception of the retailer's 'Product Ordering' process from the manufacturer, and the perception of the manufacturer's 'Production' process from the retailer are given, respectively, as follows:

$p_{Manufacturer}^{retailer.productOrderingProcess}$ = ($\mathcal{VC}_1$, { ('Order of Products', `out`), ('Confirmation of Delivery Date', `in`), ('Invoice', `in`) },

{('Order of Products', `out`) → 'Place Order with Manufacturer', ('Confirmation of Delivery Date', `in`) → 'Invoice Customer', ('Invoice', `in`)→ 'Pay Invoice'});

$p_{retailer}^{Manufacturer.productionProcess}$ = ($\mathcal{VC}_2$, { ('Order of Products', `in`), ('Confirmation of Delivery Date', `out`), ('Invoice', `out`) },

{ ('Order of Products', `in`) → 'Collect Order', ('Confirmation of Delivery Date', `out`) → 'Confirm Delivery', ('Invoice', `out`) → 'Invoice Retailer'}).

where $\mathcal{VC}_1$ and $\mathcal{VC}_2$ are defined in Example 3.1.

Definition 3.5 *Relative Workflow Process.* A relative workflow process *rp* perceivable from an organisation g_0 is defined as a directed acyclic graph ($\mathcal{T}$, $\mathcal{R}$), where

$\mathcal{T}$ is the set of the tasks perceivable from g_0, which is a union of the following two parts.

– $\bigcup_k g_0.lp^k.\mathcal{T}$, the union of the task sets of all $g_0.lp^k$. Here, $1 \le k \le m_0$ and m_0 is the number of g_0's involved local workflow processes.

– $\bigcup_i \bigcup_j g_i.lp^j_{g_0}.\mathcal{T}$, the union of the task sets of perceivable workflow processes of all $g_i.lp^j$ from g_0. Here, $1 \le i \le n$ and $1 \le j \le m_i$, while n is the number of g_0's partner organisations and m_i is the number of g_i's involved perceivable workflow processes for g_0.

$\mathcal{R}$ is the set of arcs perceivable from g_0, which is a union of the following four parts, where i, j and k are the same as in the definition of $\mathcal{T}$.

– $\bigcup_k g_0.lp^k.\mathcal{R}$, the union of the arc sets of all $g_0.lp^k$.

– $\bigcup_i \bigcup_j g_i.lp^j_{g_0}.\mathcal{R}$, the union of the arc sets of perceivable workflow processes of all $g_i.lp^j$ from g_0.

– $\mathcal{L}_{intra}$, the set of intra-organisational messaging links that connect tasks belonging to different local workflow processes, and is defined on

$\bigcup_i \bigcup_j \left(g_0.lp^i.\mathcal{T} \times g_0.lp^j.\mathcal{T}\right)$, here $i \neq j$.

– $\mathcal{L}_{inter}$, the set of inter-organisational messaging links that connect tasks between a local workflow process and a perceivable workflow process, and is defined on

$\bigcup_i \bigcup_j \bigcup_k \left(g_0.lp^k.\mathcal{T} \times g_i.lp^j_{g_0}.\mathcal{T} \cup g_i.lp^j_{g_0}.\mathcal{T} \times g_0.lp^k.\mathcal{T}\right)$.

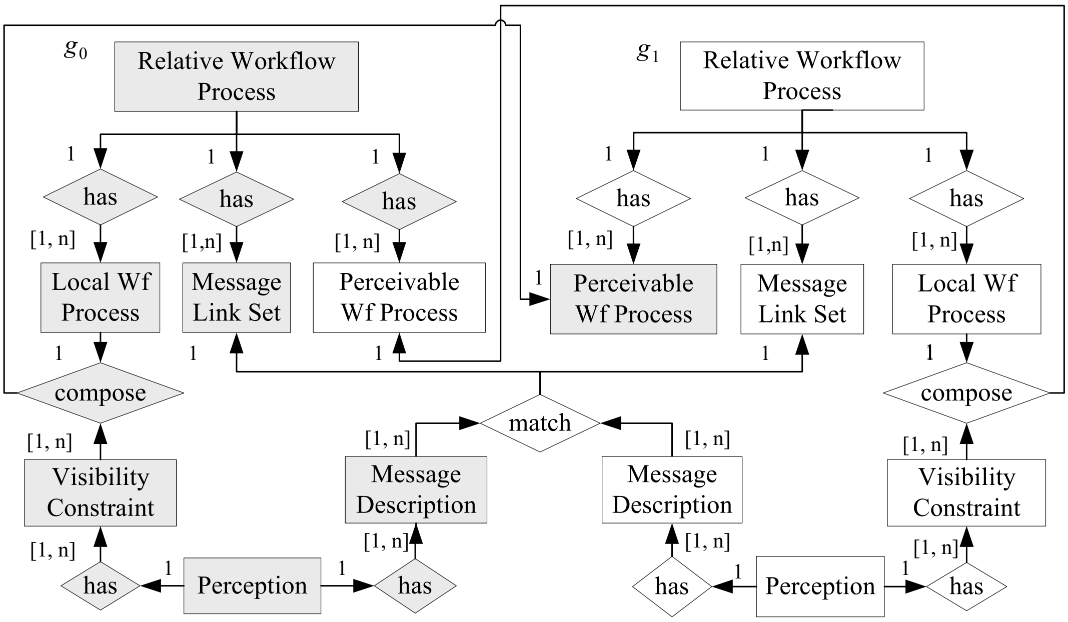

Figure 3.2 Relative workflow model

Figure 3.2 illustrates how the components of the relative workflow model are related across organisations. Given the discussion and definition of the relative workflow process above, a necessary procedure for an organisation, g_0, to generate relative workflow processes is to define the perceptions on local workflow processes of its partner organisations, g_1, g_2, …, g_n. In Figure 3.2, we only show one partner organisation, g_1, for illustration. This step includes defining visibility constraints, messages links and mapping functions. Once the perceptions on local workflow processes of its partner organisations have been defined, a relative workflow process can be generated by other two steps: composing tasks and assembling relative workflow processes.

The purpose of composing tasks is to hide some private tasks of local workflow processes. We choose to merge invisible tasks with the contactable or trackable tasks into composed tasks. According to the perceptions defined by g_1, a local workflow process of g_1 after this step becomes a perceivable workflow process for g_0.

An organisation may assemble relative workflow processes by linking its local workflow processes and the perceivable workflow processes from partner organisations together with messaging links. As shown in Figure 3.2, a relative workflow process g_0 consists of g_0's local workflow processes, the perceivable workflow processes from g_1

and the messaging links obtained by matching the message descriptions defined in the perceptions of g_0 and g_1.

The details are discussed in the following section.

3.3 Generating Relative Workflow Processes

3.3.1 Defining perceptions

A perception can be derived by analysing and decomposing a commercial contract between organisations in connection with certain business collaboration. Griffel et al. [52, 66] proposed a contract model in the Common Open Service Market for SMEs (COSMOS) project, which classifies a contract into four major parts of *Who*, *What*, *How* and *Legal Clauses*, as shown in Figure 3.3. In this paper, we employ this contract model for the visibility analysis of perceptions.

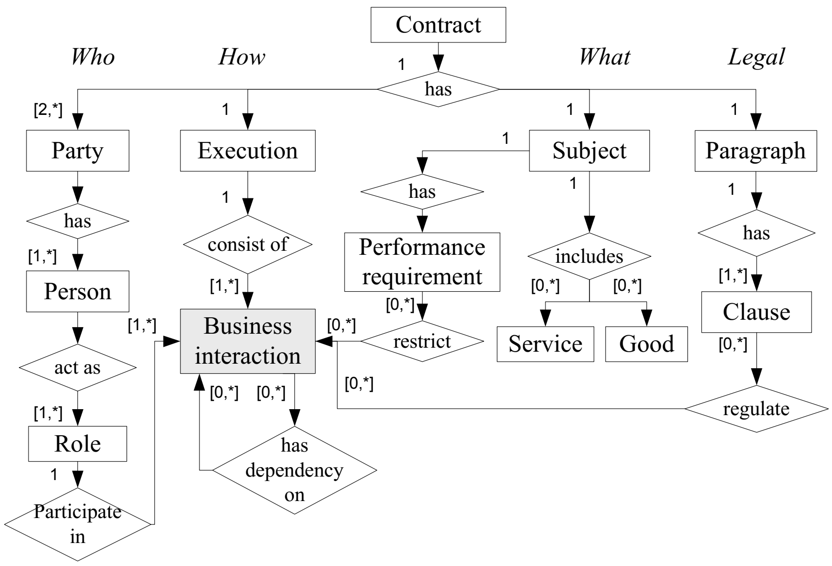

Figure 3.3 Simplified contract model (modified from [52])

As the main part of this contract model, the *How* part defines the execution details for the obligations defined in the *What* and *Legal* parts. The *execution* consists of *business interactions* that describe how the parties defined in *Who* part should interact with to fulfil the collaborations. At the process level, each business interaction is supported by one or more tasks of the involved business processes. Between these business interactions, there may exist *dependencies*, such as the logic relationship or tracking requirements etc., and these dependencies may further complicate the correlation between the supporting tasks at the process level. In the contracting process, we call the organisation that issues a contract a *host organisation*, and the responding organisations *partner organisations*.

Unlike a contract, a perception is defined from the perspective of one organisation on the local workflow processes of other participating organisations. To represent a business collaboration between an organisation g_0 and partner organisations, $g_1, \ldots, g_n$, two sets of such perceptions are required:

$\mathcal{PS}_1$, the set of the perceptions defined on g_0's participating local workflow processes, $g_0.lp_1, \ldots, g_0.lp^{m_0}$, from $g_1, \ldots, g_n$, i.e. $\{ p_{g_1}^{g_0.lp^1}, \ldots, p_{g_1}^{g_0.lp^{m_0}}, \ldots, p_{g_n}^{g_0.lp^1}, \ldots, p_{g_n}^{g_0.lp^{m_0}} \}$;

$\mathcal{PS}_2$, the set of the perceptions defined on all participating local workflow processes of $g_1, \ldots, g_n$ from g_0, i.e. $\{ p_{g_0}^{g_1.lp^1}, \ldots, p_{g_0}^{g_1.lp^{m_1}}, \ldots, p_{g_0}^{g_n.lp^1}, \ldots, p_{g_0}^{g_n.lp^{m_n}} \}$.

Thus, we can see that each workflow process involved in the contracted business collaboration is assigned with a proper perception. To achieve this, we need to derive a business collaboration oriented contract to a business process oriented perception. This derivation involves recognising necessary inter-organisational messages and setting up visibility constraints for tasks etc. As mentioned before, a contract c defines the necessary business interactions to fulfil the collaboration. Algorithm 3.1 gives the detailed steps that how g_0's partner organisation g_1 generates a perception p of g_1's local workflow process lp for g_0, according to the business interactions defined in c.

Algorithm 3.1 Generating perceptions

Input:

c a contract signed by two organisations g_0 and g_1

g_0 the host organisation

g_1 the partner organisation

lp an involved local workflow process of g_1

Output:

p the generated perception of g_1 from g_0

Step 1 *Set all tasks invisible.*

$p.\mathcal{VC} = \emptyset$; $p.\mathcal{MD} = \emptyset$; $p.f = \emptyset$;
for each task $t \in lp${
  $p.\mathcal{VC} = p.\mathcal{VC} \cup \{(t,$ `invisible`$)\}$;
}

Step 2 *Set contactable tasks.*

for each business interaction *bi* defined in contract *c* **{**
  for each task $t \in lp$ **{**
    if task *t* provides necessary functions for *bi* **then {**
      if $\exists(t,$ `invisible` $) \in p.\mathcal{VC}$ **then {**
        $p.\mathcal{VC} = p.\mathcal{VC}$ **-** $\{(t,$ `invisible` $)\}$;
        $p.\mathcal{VC} = p.\mathcal{VC} \cup \{(t,$ `contactable` $)\}$;
      }
      mdSet = {the message descriptions to be used by *t* to support *bi* }
      for each message $md \in mdSet$ **{**
        $p.\mathcal{MD} = p.\mathcal{MD} \cup \{ md \}$;
        $(md \rightarrow t) \rightarrow p.f$;
      }
    }
  }
}

Step 3 *Set trackable tasks.*

for each business interaction *bi* defined in contract *c* **{**
  for each task $t \in lp$ **{**
    if *bi* has status dependency with *t* **then {**
      if $\exists (t,$ `invisible` $) \in p.\mathcal{VC}$ **then {**
        $p.\mathcal{VC} = p.\mathcal{VC}$- $\{(t,$ invisible $)\}$;
        $p.\mathcal{VC} = p.\mathcal{VC} \cup \{(t,$ `trackable` $)\}$;
      }
    }
  }
}

Figure 3.4 shows the 'Product Ordering' process and the 'Production' process in the motivating example, where the dashed arrows denote the message descriptions. To represent the collaboration between these two business processes, we can define the perception $p_{Manufacturer}^{Retailer.productOrderingProcess}$ of the retailer's 'Product Ordering' process from the manufacturer and the perception $p_{Retailer}^{Manufacturer.productionProcess}$ of the manufacturer's 'Production' process from the retailer, which are already given in Example 3.1.

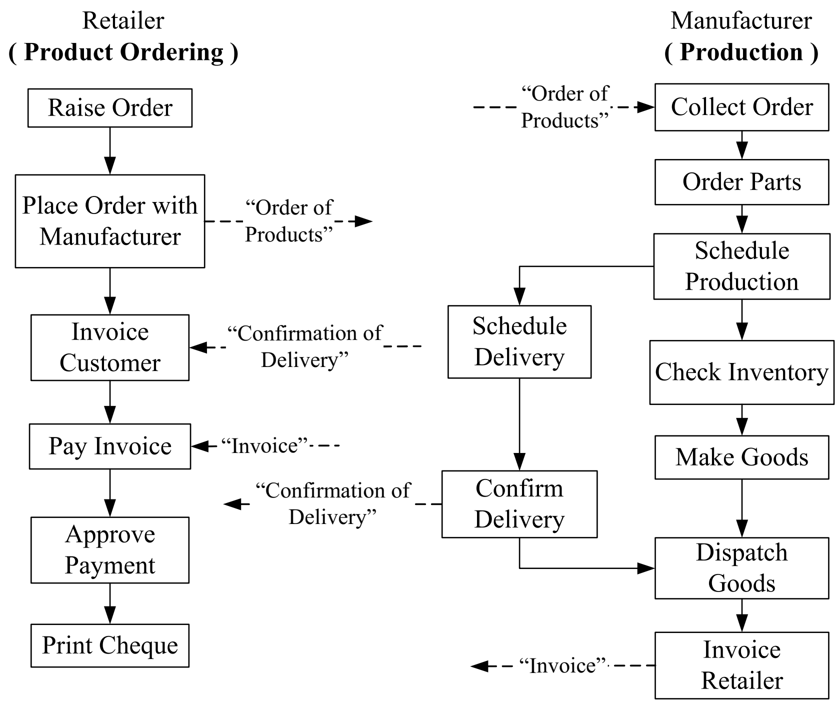

Figure 3.4 Local workflow processes

3.3.2 Composing tasks

In this step, a local workflow process hides its invisible tasks by composing them with proper contactable or trackable tasks to create the corresponding perceivable workflow process. The algorithm is given below.

For simplicity of discussion, we only consider composing partner organisation g_1's local workflow process *lp* from host organisation g_0. Furthermore, we conduct a pre-processing on all split / join structures of *lp* such that for all those branches consisting of only invisible tasks, a dummy task is created to delegate these branches.

Algorithm 3.2 Task Composition

Input:

lp $g_1.lp$, organisation g_1's local workflow process lp before composition

p $p_{g_0}^{g_1.lp}$, the perception of g_1's lp from g_0

Output:

lp' $g_1.lp_{g0}$, the perceivable workflow process composed from lp for g_0, according to $p_{g_0}^{g_1.lp}$

Step 1 *Connect invisible tasks.*

$lp' = lp$;
$\mathcal{VT}$ = { all the visible tasks of lp, defined in p};
while $(\exists t, t' \in (lp'.\mathcal{T} - \mathcal{VT}))\ ((t, t') \in lp'.\mathcal{R}) \wedge \text{seq}(t) \wedge \text{seq}(t'))$
// seq(t)=(indegree(t)=1$\wedge$*outdegree(t)=1)*
{ $t^\circ = t + t'$;
 $lp'.\mathcal{T} = lp'.\mathcal{T} \cup \{t^\circ\} - \{t, t'\}$;
 $lp'.\mathcal{R} = lp'.\mathcal{R} - \{(t, t')\}$;
 replace t, t' in $lp'.\mathcal{R}$ with t° ;
}

Step 2 *Downward composition with incoming interaction tasks.*

while $((\exists t \in \mathcal{VT}(p'.f^{-1}(t) = (m, \texttt{in}) \wedge \text{outdegree}(t) = 1) \wedge (\exists t' \in (lp'.\mathcal{T} - \mathcal{VT}))((t, t') \in lp'.\mathcal{R} \wedge \text{indegree}(t') = 1))$
{ $t^\circ = t + t'$;
 $\mathcal{VT} = \mathcal{VT} \cup \{t^\circ\} - \{t\}$;
 $lp'.\mathcal{T} = lp'.\mathcal{T} \cup \{t^\circ\} - \{t', t\}$;
 $lp'.\mathcal{R} = lp'.\mathcal{R} - \{(t, t')\}$;
 replace t, t' in $lp'.\mathcal{R}$ with t° ;
}

Step 3 *Upward composition with outgoing interaction tasks.*

while $((\exists t \in \mathcal{VT}(p'.f^{-1}(t) = (m, \texttt{out}) \wedge \text{indegree}(t) = 1) \wedge (\exists t' \in (lp'.\mathcal{T} - \mathcal{VT}))((t', t) \in lp'.\mathcal{R} \wedge \text{outdegree}(t') = 1))$
{ $t^\circ = t + t'$;
 $\mathcal{VT} = \mathcal{VT} \cup \{t^\circ\} - \{t\}$;
 $lp'.\mathcal{T} = lp'.\mathcal{T} \cup \{t^\circ\} - \{t', t\}$;
 $lp'.\mathcal{R} = lp'.\mathcal{R} - \{(t', t)\}$;
 replace t, t' in $lp'.\mathcal{R}$ with t° ;
}

Algorithm 3.2 first keeps composing each pair of neighbouring sequential invisible tasks into one invisible task, then downward composes invisible tasks with incoming interaction tasks and upward composes invisible tasks with outgoing interaction tasks.

Figure 3.5 shows the results of task composition: (a) is the perceivable 'Product Ordering' process of the retailer from the manufacturer; and (b) is the perceivable 'Production' process of the manufacturer from the retailer, where the dashed rectangles denote invisible tasks.

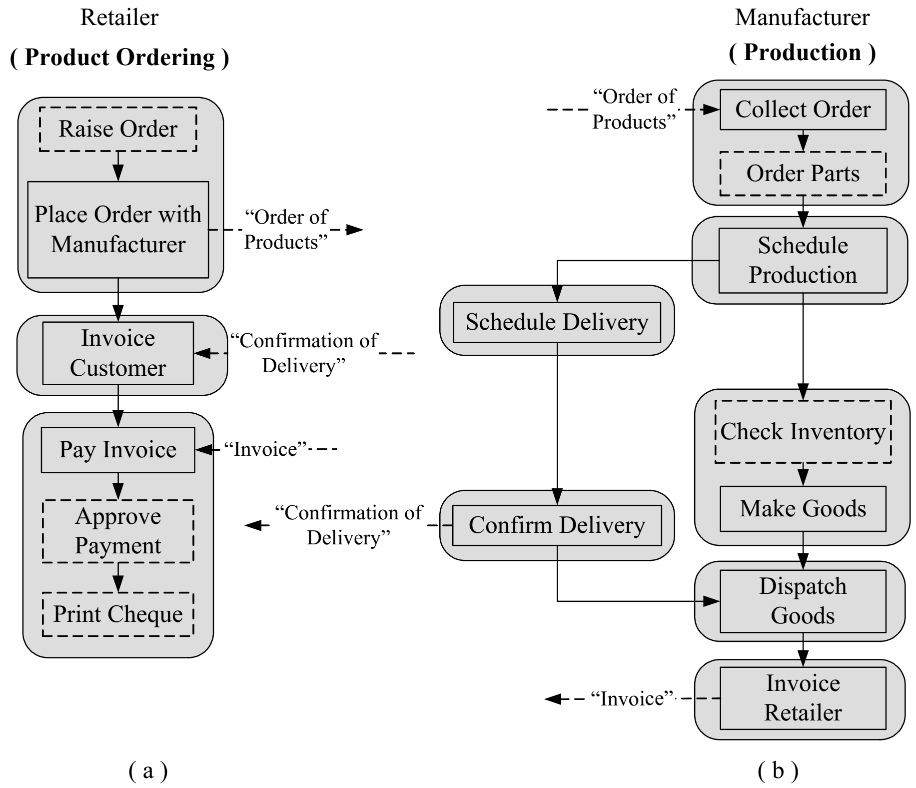

Figure 3.5 Perceivable workflow processes

3.3.3 Assembling relative workflow processes

In this step, proper local workflow processes and perceivable workflow processes are connected together by linking the corresponding interaction operations. Using algorithm 3.3, this linking procedure can be done by matching the message descriptions defined in the perceptions. For simplicity of discussion, we only consider matching one local workflow process *lp* of partner organisation g_1 from host organisation g_0 in the given algorithm.

Algorithm 3.3 Local Workflow Process Matching

Input:

lp' — $g_1.lp_{g0}$, the perceivable workflow process composed from g_1's local workflow process lp

p — $p_{g_0}^{g_1.lp}$, the perception of g_1's lp from g_0

$\mathcal{PS}$ — $\{ p_{g_1}^{g_0.lp^1}, \ldots, p_{g_1}^{g_0.lp^{m0}} \}$, the set of perceptions defined on g_0's perceivable workflow processes from g_1

Output:

$\mathcal{L}$ — the set of generated messaging links.

Step *Generating messaging links to bind workflow processes.*

$\mathcal{L} = \emptyset$;
for each $t \in lp'.\mathcal{T}$ **{**
  if $\exists md(p.f(md)= t)$ **then {**
   $md_1=p.f^{-1}(t)$;
   for each $p° \in \mathcal{PS}$ **{**
     for each $md_2 \in p°.\mathcal{MD}$ **{**
       if md_1 **matches** md_2 **then {**
         $\mathcal{L} = \mathcal{L} \cup \{(t, p°.f(md_2), md_1)\}$;
/ the messaging links are obtained by matching messaging descriptions. */*
       }
     }
   }
  }
}

By a message description md_1 matching another message description md_2 in Algorithm 3.3, we mean that they have the same message, and one has passing direction 'in' while the other has 'out'. With the set $\mathcal{L}$ of generated messaging links, we can now finally assemble relative workflow processes.

Figure 3.6 (a) shows the relative workflow process perceivable from the retailer; and (b) shows the relative workflow process perceivable from the manufacturer, where the dashed connecting arrows denote the generated message links. Different participating organisations may have different views to the same collaborative business process. This reflects the *relativity* characteristic of the model.

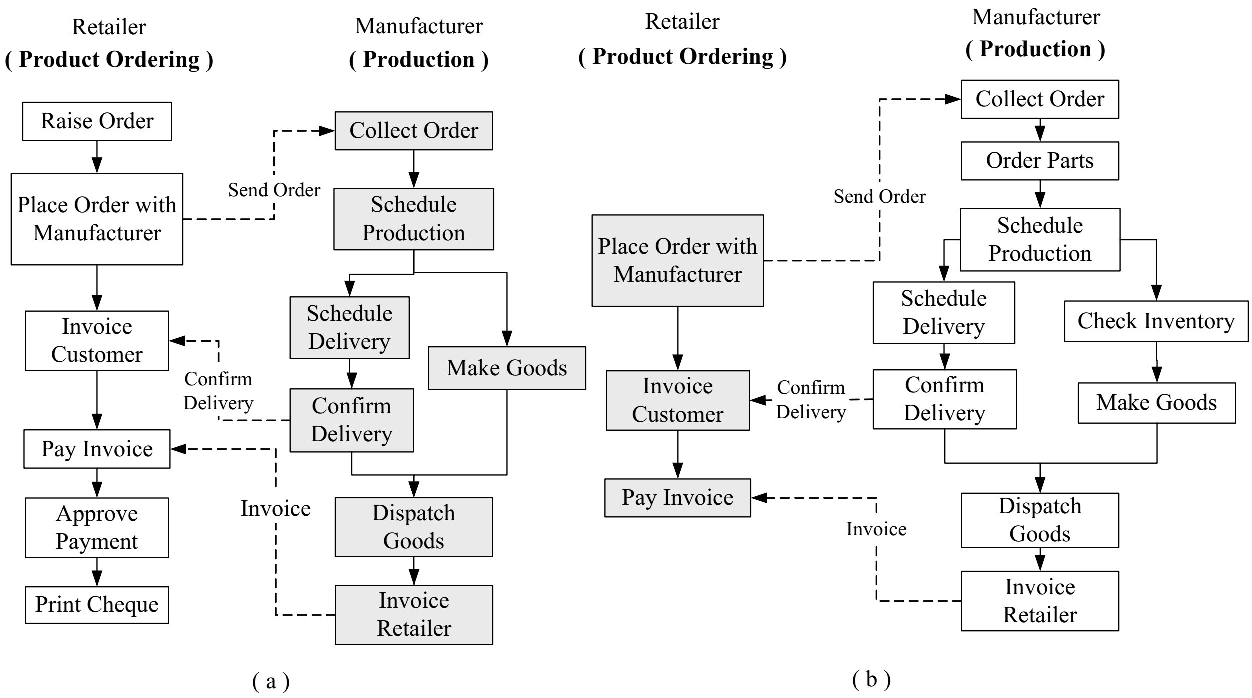

Figure 3.6 Relative workflow processes

3.4 Model Justification

In this section, we justify the relative workflow model from the aspects of information sufficiency and necessity. From the organisation-oriented perspective, we define the information sufficiency and necessity for relative workflows in terms of their partial views over a collaborative business process in a public view. Formally, the following two propositions describe the information sufficiency and necessity, respectively:

Proposition 1. A relative workflow process contains necessary information for the host organisation to accomplish its responsibilities in its participated business collaboration.

Proposition 2. A collaborative business process can be sufficiently represented by a finite number of relative workflow processes defined for participating organisations.

In regard to Proposition 1, the responsibilities of an organisation in its participating collaborative business process are defined in the *What* and *Legal Clause* parts of contracts, according to the contract model of Section 3. Further, the *How* part describes the execution details for the content defined in *What* and *Legal Clause* using business interactions. These business interactions are thereafter converted into messaging interactions between some tasks, which are set "contactable" in proper perceptions. With the perceptions defined for a specific organisation, this organisation can see all the contactable tasks of its partner organisations. The perceptions also provide necessary interface specifications, such as the message descriptions combined with the interfaces. The relative workflow process generated from these perceptions inherits all these information. Therefore, such a relative workflow process includes the necessary information for the host organisation to fulfil its responsibilities in the collaboration. □

Proposition 2 emphasises that a collaborative business process in a public view can be covered by a group of relative workflow processes, which are created for the participating organisations, though each of these relative workflow processes only represents a partial view over the whole collaboration. To prove this, we first represent the structure of a collaborative business process, say *cbp*, as a graph ($\mathcal{N}$, $\mathcal{A}$) in a public view, where

- set $\mathcal{N}$ denotes the set of involved tasks;
- set $\mathcal{A}$ denotes the set of all links.

In addition, set $\mathcal{A}$ contains two kinds of links, viz,. a set of intra process links, say $\mathcal{A}_{intra}$, and a set of inter process links, say $\mathcal{A}_{inter}$.

Based on the definition of relative workflow process, each relative workflow process can also be represented as a graph ($\mathcal{T}$, $\mathcal{R}$), where $\mathcal{T} = \mathcal{T}_L \cup \mathcal{T}_p$, $\mathcal{R} = \mathcal{R}_L \cup \mathcal{R}_p \cup \mathcal{L}_{intra} \cup \mathcal{L}_{inter}$.

- sets $\mathcal{T}_L$ and $\mathcal{T}_P$ denote the tasks belonging to local workflow processes, and the tasks belonging to perceivable workflow processes, respectively;
- sets $\mathcal{R}_L$ and $\mathcal{R}_P$ denote the intra process links inside local workflow processes and the intra process links inside perceivable workflow processes;
- sets $\mathcal{L}_{intra}$ and $\mathcal{L}_{inter}$ denote the intra-organisational links that connect tasks belonging to different local workflow processes, and the inter-organisational links that connects tasks between a local workflow process and a perceivable workflow process.

The tasks of local workflow processes are totally visible to the host organisation, therefore all the tasks of a collaborative business process *cbp* can be obtained from the tasks of local workflow processes belonging to a group of relative workflow processes. This can be formalised below,

$$(\forall cbp)\ \exists RWF\ (\ cbp.\mathcal{N} \subseteq \bigcup_{rwf \in RWF} rwf.\ \mathcal{T}_L\ ) \tag{1}$$

Here, set *RWF* denotes a set of relative workflow processes.

Given the tasks of local workflow processes are available, the intra process links between these tasks are also obtainable from these relative workflow processes, due to the definition of intra process links, i.e., $\mathcal{R}_\mathcal{L} \subseteq \mathcal{T}_\mathcal{L} \times \mathcal{T}_\mathcal{L}$. Here, we formalise this finding as the following expression.

$$(\forall cbp)\ \exists RWF\ (\ cbp.\mathcal{A}_{intra} \subseteq \bigcup_{rwf \in RWF} rwf.\ \mathcal{R}_L\ ) \tag{2}$$

Regarding a specific relative workflow process, say *rwf*, of organisation *g*, it includes the set of inter process links connecting a task of a perceivable workflow process and a task of a local workflow process This means that set $rwf.\mathcal{L}_{inter}$ includes the links that connect the tasks of *g*'s local workflow processes to the tasks of workflow processes belonging to *g*'s neighbour organisations in a public view. As such a finite number of relative workflow processes, at most all the relative workflow processes of all participating organisations, will definitely cover the links between two workflow processes belonging to different organisations in a collaborative business process. Therefore, we can formalise this finding as the following expression,

$$(\forall cbp)\ \exists RWF\ (\ cbp.\mathcal{A}_{inter} \subseteq \bigcup_{rwf \in RWF} rwf.\mathcal{L}_{inter}\ ) \tag{3}$$

Based on (1), (2) and (3), we can finally draw the following conclusion,

$$(\forall cbp)\ \exists \mathcal{RWF}\ (\ cbp.\mathcal{N} \subseteq \bigcup_{rwf \in RWF} rwf.\ \mathcal{T}_\mathcal{L}) \wedge (\ cbp.\mathcal{A} \subseteq \bigcup_{rwf \in RWF} (\ rwf.\mathcal{L}_{inter} \cup rwf.\mathcal{R}_\mathcal{L}\ )\ ) \ \Box$$

3.5 Summary

In this chapter, the problems of current collaborative business process management are first identified, namely the pre-dominant design, the vulnerable visibility control and the poor flexibility of partnership representation. To tackle these problems, this chapter presents a relative workflow model in detail. Different from the other workflow models, this model defines a collaborative business process from the individual view of each participating organisation. Thereby, different organisations may define different collaborative business processes for the same collaboration. A set of key notions have been formally defined, and the meta model of the relative workflow model has been discussed with the motivating example. The generation of a relative workflow process is introduced with algorithms and examples, including procedures for perception generation, perceivable workflow process generation and relative workflow process assembling. A formal justification is also given to prove the information sufficiency and necessity for the relative workflow model.

Chapter 4

Specifying Instance Correspondence in Collaborative Business Processes

Last chapter discusses about the collaborative business process management at process level. This chapter goes forward to investigate the collaborative business process management at instance level. Different from conventional business processes, a collaborative business process involves multiple parties and their business processes, and therefore it inevitably brings new challenges to workflow choreography and orchestration. Especially, the issue of instance correspondence becomes very pressing in the context of collaborative business processes.

In this chapter, we address this issue on the basis of our organisation-oriented view framework. In this chapter, we look into the problem of instance correspondences in terms of cardinality and correlation. As such, the static and dynamic instance correspondence can be represented at build time and run time, respectively.

Some research efforts were put in this field. Multiple workflow instantiations were discussed by Dumas and ter Hofstede [64], using UML activity diagrams. Later they extended their work to service interactions [49]. van der Aalst et al. [31, 67] deployed coloured Petri nets to represent multiple workflow cases in workflow patterns, and implemented it in the YAWL system [32]. Zhou, Shi and Ye [63] also studied pattern based modelling for multiple instances of workflow activities. Guabtni and Charoy [62] extended the multiple instantiation patterns and classified multiple workflow instantiation into parallel and iterative instances. However, most of above research focuses on interaction patterns, and sidesteps the instance correspondence issue in collaborative business processes.

WS-BPEL (previously BPEL4WS) [9] uses its own correlation set to combine workflow instances, which have same values on specified message fields. However, WS-BPEL defines a business process in terms of a pivot organisation. This results in that a WS-BPEL business process only represents the interaction behaviours of the pivot organisation with its neighbouring organisations. This feature limits its application for complex business collaborations, which are likely to include interactions beyond neighbouring organisations.

Aiming to address this issue, this chapter proposes a method to support instance correspondences from an organisation-oriented view. In our method, cardinality parameters are developed to characterise cardinality relationships between collaborating business processes at build time. Besides, a correlation structure is combined with each instance to trace dynamic workflow correlations at run time. In addition, we formalise this method by extending traditional Petri nets to describe instance correspondence precisely.

In this chapter, Section 4.1 analyses the instance correspondence within collaborative business processes with a motivating example. In Section 4.2, we illustrate an organisation-oriented view towards collaborative business processes, and then discuss workflow cardinality and correlation issues in the context of business collaborations. In Section 4.3, we establish a novel correspondence Petri net (CorPN) model with special parameters for workflow cardinality and correlation. In Section 4.4, algorithms are developed to illustrate how we model collaborative business processes and manage run time executions of collaborative business processes with proposed CorPNs.

4.1 Motivating Example

Figure 4.1 shows a collaboration scenario, which is modified from the motivating example in Chapter 3. In this scenario, retailers, manufacturers and shippers participate in one collaboration using their product-ordering, production and shipping processes, respectively. A retailer may initiate a product-ordering process instance that orders products from a manufacturer. The manufacturer may have a production process instance, which keeps receiving orders from retailers. Once it obtains enough orders, the production instance may start making goods in bulk. At the same time, the manufacturer may assign several shippers to handle goods delivery. These shippers get the consignee information from the manufacturer, and arrange their goods transfer according to their transfer capability and route optimisation etc. In this process, via the production instance, each product-ordering instance is correlated with the shipping instances that are responsible for the transfer of goods ordered by this product-ordering instance. Finally, these shipping instances send goods to the proper retailers according to these correlations.

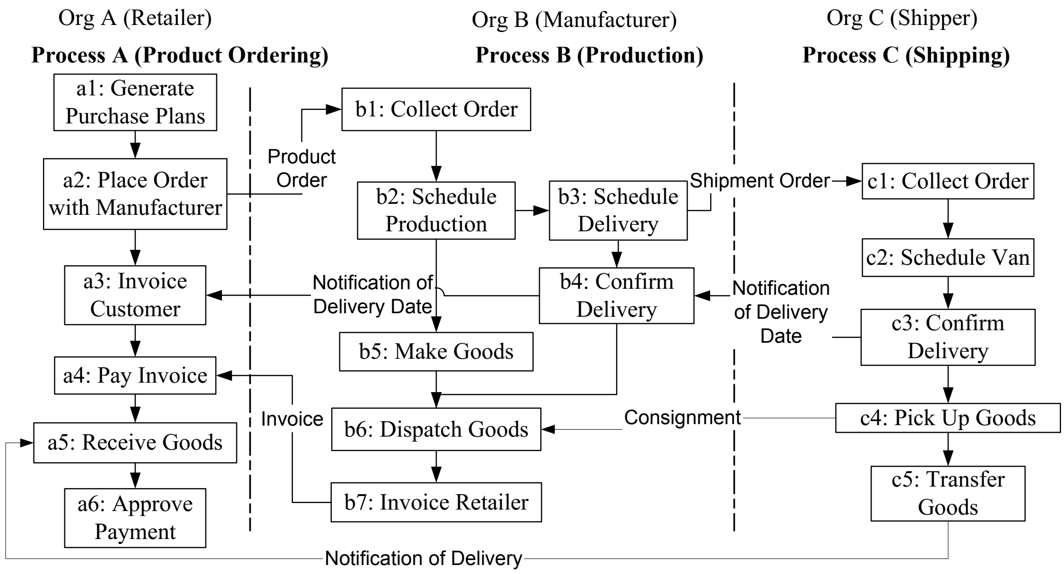

Figure 4.1 Motivating example

From this collaboration scenario, we see that an instance of one business process is likely to interact with multiple instances of another business process. For example, one production instance may correspond to multiple product-ordering instances, and multiple shipping instances may correspond to multiple product-ordering instances. In contrast to such quantitive relationship between business process instances, most current business process modelling approaches simply assume that one instance of a business process interacts with one instance of another business process. To better support instance correspondences, cardinality between different business processes should be particularly considered at build time.

At run time, instance correspondences are subject to the correlations between instances of different business processes. These correlations result from the underlying semantics of business interactions, and therefore these correlations reflect the coupling relationship between business process instances. In real cases, such correlations may be realised by real interactions (direct) or passing unique identifiers (indirect), such as order numbers. Sometimes, real interactions between instances may be triggered by time duration, external events etc. In this example, the manufacturer's production instance is correlated with retailers' product-ordering instances during the real interaction of receiving orders from retailers. Afterwards, the manufacturer contacts shippers to book deliveries. At the same time, the manufacturer also passes the order numbers to proper shippers. With these order numbers, shippers' shipping instances are indirectly correlated with retailers' product-ordering instances. Following these correlations, shippers can pick up produced goods from the manufacturer, and then transfer them to proper retailers.

From the above discussion, we see that workflow correlations combine business interactions into a meaningful collaboration. Some existing approaches provide primitive support for correlation handling, such as message correlations in WS-BPEL. As discussed in Section 1, a WS-BPEL business process generated for a retailer cannot cover the interactions between the manufacturer and shippers, not to mention the correlations between their production and shipping instances.

4.2 Workflow Cardinality and Correlation in Collaborative Business Processes

4.2.1 Organisation-oriented view

In a collaborative business process, each participating organisation may play a specific role and only care about its own interests. For this reason, participating organisations do not wish, and may not be allowed to know the details of their partner organisations. Therefore, each participating organisation only has a partial and restricted view of the whole collaboration [37, 68-71]. Due to diverse partnerships and authorities, different organisations may view the same collaboration differently.

In a collaborative business process, one-to-one correspondence may not be held between instances of different sub business processes. Therefore, it is hard to identify an instance of a collaborative business process. However, such an instance can be defined for individual participating organisation. From each instance ζ of a business process belonging to organisation g, we can derive a so-called *logical instance* ξ. This logical instance includes ζ and all its related instances of business processes belonging to other organisations through the instance correlations at run time. Here, organisation g is called *host organisation* of ξ, and ζ is called *base workflow instance* of ξ. In this example, the logical instance for a product-ordering instance may include related production instance and shipping instances besides itself, since these production instance and shipping instances are responsible for making the ordered products and transferring these goods to the retailer, respectively.

4.2.2 Workflow cardinality

Figure 4.2 shows a possible instance correspondence situation of the collaborative business process discussed in the motivating example.

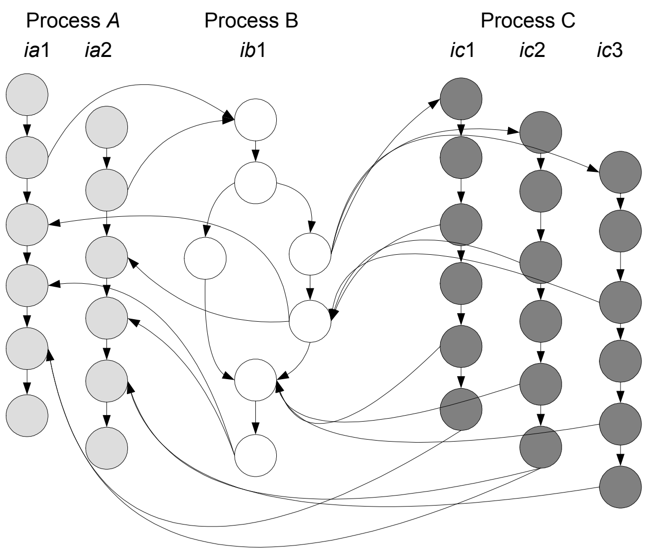

Figure 4.2 Workflow cardinality of motivating example

In general, there are four possible cardinality relationships between a pair of interacting business processes, viz., single-to-single, single-to-many, many-to-single and many-to-many. Three of these relationships can be found in Figure 4.2, since the motivating example does not involve the single-to-single relationship. In the organisation-oriented view, it makes more sense to use unidirectional cardinality specifications, i.e., to-one and to-many. The four bilateral cardinality relationships are therefore represented by the pair of unidirectional cardinality relationships. For example, a single-to-many relationship between workflow processes p_B and p_C can be represented by a "to-many" relationship from p_B to p_C and a "to-one" relationship from

p_C to p_B. A many-to-many relationship between p_A and p_C can be represented by a "to-many" relationship from p_A to p_C and a "to-many" relationship from p_C to p_A. In this chapter, we define these two unidirectional cardinality relationships with two *workflow cardinality parameters*,

[:1], denotes a *to-one* cardinality relationship;

[:n], denotes a *to-many* cardinality relationship.

As process interactions are implemented in the form of messaging behaviours, we incorporate these two cardinality parameters to message modelling. Conceptually, a message type can be defined as follows:

Definition 4.1 *Message type*. A message m is defined as a tuple (α, ϑ, β, f, χ), where

– α is m's messaging direction, '*in*' or '*out*'. These two values denote that m stands for an incoming message or an outgoing message, respectively.

– ϑ is a task of a workflow process. ϑ represents the source task of message m, if m stands for an outgoing message; or it represents the target task.

– β is a set of tasks. This set of tasks represents m's source tasks, if m stands for an incoming message; or it represents m's target tasks. Each task in β is expected to send or receive an instance of m, according to the direction of m.

– $f : \beta \rightarrow \{$ [:1], [:n] $\}$ is a mapping from β to the two aforementioned cardinality parameters.

– χ denotes the specification of the message body.

Here, ϑ and β together represent the cardinality between business processes at type level. Two messages types are said to be a pair if they have complementary source / target tasks and the same message body specification. These paired message types can be used to link corresponding business processes together into a collaborative business process. The details about this linking process via message types will be discussed in Section 4.5.

4.2.3 Workflow correlation

Workflow correlation denotes the coupling relation between workflow instances in the same business collaboration. Two or more workflow instances are directly correlated, when they "shake hands" during run time interactions. In addition, some participating instances may inherit pre-existing workflow correlations from their counterparts during run time interactions. This correlation inheritance indicates that business coupling relation may extend to subsequent workflow instances as the collaboration proceeds.

In the scenario shown in Figure 4.2, firstly instances ia_1 and ia_2 are correlated with instance ib_1, when ib_1 accepts orders from ia_1 and ia_2; Secondly, ib_1 contacts instances ic_1, ic_2 and ic_3 for delivery booking. Here, suppose ib_1 assigns ic_1 and ic_2 to transfer products for ia_1, and assigns ic_2 and ic_3 to transfer products for ia_2. Thereby instances ic_1, ic_2 and ic_3 are directly correlated with ib_1, and they also inherit previous correlations from ib_1. In this example, ic_1 and ic_2 inherit the correlation between ia_1 and ib_1 from ib_1, while instances ic_2 and ic_3 inherit the correlation between ia_2 and ib_1 from ib_1. This correlation inheritance implies that shippers require consignees' information to arrange their shipping schedules. Corresponding shipping instances are therefore indirectly correlated with retailers' product-ordering instances. As discussed before, this inheritance is realised by passing retailers' order numbers from the manufacturer to shippers.

Based on these workflow correlations, a logical instance of a participating workflow instance can be derived in the organisation-oriented view. Here, we define a logical instance of a base workflow instance in terms of workflow correlations.

Definition 4.2 *Logical instance*. In the context of a collaborative business process Λ, the logical instance for a base workflow instance ζ is defined as tuple (ζ, Λ, Δ), where Δ is the set of workflow instances that are correlated with ζ in the context of Λ.

The set of correlated workflow instances evolves during the business collaboration. For example, if we start from instance ia_1, the set of correlated workflow instances Δ for ia_1 contains no instances at the beginning; while it includes instance ib_1 right after ib_1 accepts orders from ia_1, i.e., $\Delta = \{ ib_1 \}$; afterwards instances ic_1 and ic_2 may be added after ib_1 books delivery with ic_1 and ic_2, then $\Delta = \{ ib_1, ic_1, ic_2 \}$.

4.3 Correspondence Representation Methodologies

4.3.1 Introduction of Petri nets

Petri nets were invented by Carl Petri in 1960s [72] for modelling concurrent behaviours of a distributed system. A Petri net is a bipartite graph whose nodes can be distinguished in places and transitions, which are graphically represented by circles and rectangles, respectively. A Predicate / Transition or coloured Petri net can differentiate tokens with unique identifications or a set of colours. Each place can contain tokens of different identifications or colours at the same time. Each arc may be assigned with an expression to restrict what tokens and the amount of tokens that can transfer through. Therefore, a Petri net can represent multiple process executions within one net. Besides a sequential structure, Petri nets also use the following four control structures to coordinate the routing of tokens, viz., AND-Split / AND-Join / OR-Split / OR-Join, as listed in Figure 4.3 [73, 74].

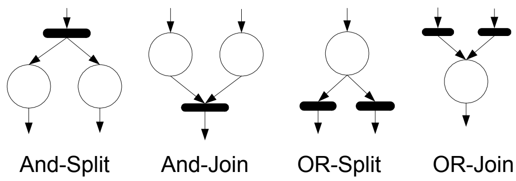

Figure 4.3 Primitive Petri net structures

4.3.2 Extension to Petri nets

To support workflow cardinality and correlation, we extend traditional Petri nets with new parameters and functions together with special places and transitions.

1. Message representation

In our approach, we use an auxiliary place in Petri net to represent a message between two business processes. As shown in Figure 4.4, two collaborating business processes are represented by two sub nets, which are differentiated by white and striped circles. The auxiliary place, which is represented as a shaded circle, stands for a message that is sent from the left transition to the right one.

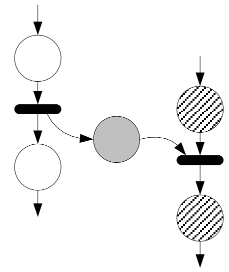

Figure 4.4. Representing messages

2. Cardinality parameters

The two uni-directional cardinality parameters, [:1] and [:n], are now incorporated into the arcs adjacent to auxiliary places, as shown in Figure 4.5.

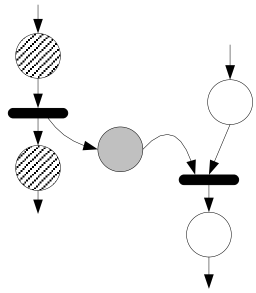

Figure 4.5 Cardinality parameters

The shaded auxiliary place p connects two sub nets A and B, which represent two collaborating business processes, respectively. Transition t_1 of A is an *interaction requesting transition*, while transition t_2 of B is an *interaction responding transition*. Thus, t_2 has input arcs from both A and B. Label "[:1]" on the arc linking t_1 to p denotes that A views this interaction as a "to-one" cardinality. This means each token in A interacts with one token in B from A's view. On the other hand, label "[:n]" on the arc linking p to t_2 denotes that B treats this interaction as a "to-many" cardinality, which indicates that each token in B corresponds multiple tokens in A from B's view. Therefore, we see that an auxiliary place separates the cardinality views from different perspectives.

3. Multiple message senders / receivers

Actually, cardinality parameter "[:n]" already implies the existence of multiple message senders or receivers. A label "[:n]" on an outgoing or incoming arc indicates multiple message receivers or senders, respectively. But label "[:n]" here merely represents a specific scenario that the senders or receivers are instances of the same business process. In some more complex scenarios, senders or receivers may be instances of different business processes, and only part of senders or receivers may be expected to send or receive a message. Thus, we create the following structures in Figure 4.6 to handle these scenarios by composing Petri net primitive AND/OR Join/Split structures. In Figure 4.6, we differentiate business processes with sub nets painted in different circles, and mark all auxiliary places as shaded ones.

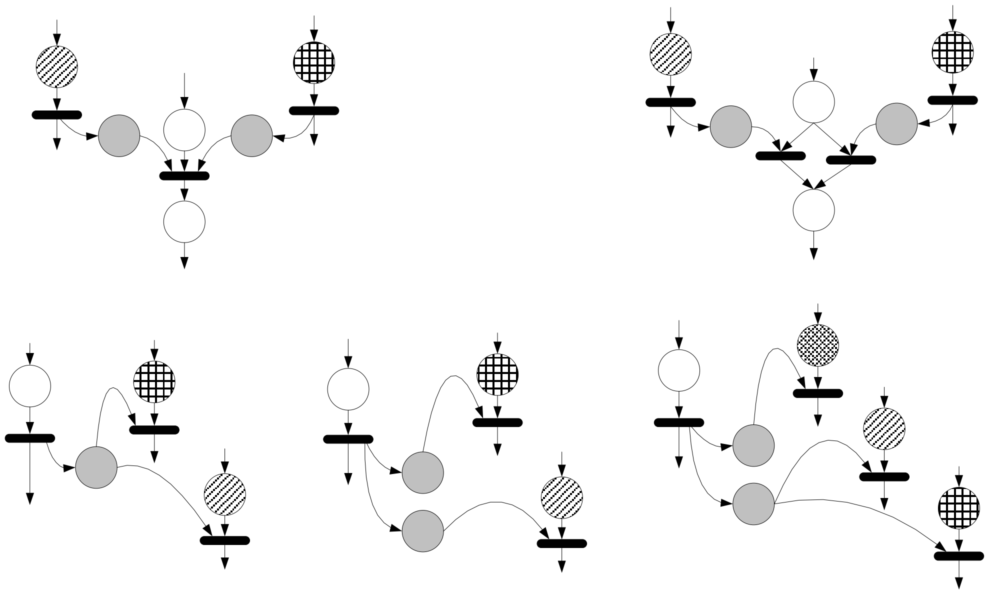

Figure 4.6 Interactions with multiple senders/receivers

In regard to the interactions with multiple senders, Figure 4.6 (a) shows an interaction receiving messages from two senders; while Figure 4.6 (b) shows an interaction receiving one message from two senders. In regard to the interactions with multiple receivers, Figure 4.6 (c) shows an interaction in which one of two receivers is expected to receive the message; while Figure 4.6 (d) shows an interaction that a message is sent to both receivers. By composing these basic interaction schemes, we can represent more complicated interactions. For example, Figure 4.6 (e) shows a scenario that a task sends a message to three receivers, and one of the three will receive it definitely, while only one of the other two is expected to do so.

In this approach, we also note that all multi-lateral interactions are decomposed into a series of bilateral ones at the receiver's side to adapt our unidirectional cardinality representation.

• • •

4. Special transitions

In some cases, an interaction may result in generating new business process instances. For example, when the manufacturer contacts shippers to schedule delivery, a shipper may generate several new shipping instances to handle the delivery. In the corresponding Petri net, this requires the transition for this interaction to be capable of generating new tokens. In this chapter, we category such transitions as *token-generating transitions*. In this way, we represent the book-delivery interaction between manufacturer and shippers using the Petri net segment shown in Figure 4.7. Here, two sub nets A and B are differentiated with different circles, and auxiliary place p is represented as a shaded circle.

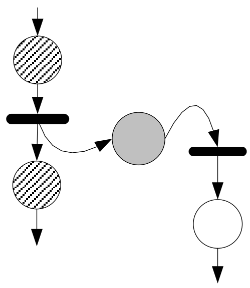

Figure 4.7 Correlation function attached structures

In Figure 4.7, variable x or y is labelled along an arc to denote the type of tokens that may go through this arc. For example, the token that flows from transition t_1 to place p is different from the token that flows out of transition t_2. In addition, t_2 is a token-generating transition, which may generate several tokens when triggered by tokens from A. Here, we note that expression 2^y is labelled along the arc linking t_2 to the adjacent place. This arc allows that more than one token representing instances of the same business process to pass through at one time.

5. Correlation structures

From above discussion, we see that a Petri net can well model the control-flow dimension at the conceptual level. Nevertheless, because tokens of traditional Petri nets do not carry much case related information, it is hard to specify dynamic behaviours of a single token. For this reason, we combine a correlation structure with each token to record run time workflow correlation, and a correlation structure is defined as follows:

Definition 4.3 *Correlation structure*. In a Petri net, the correlation structure for token ς is defined as $r^{\varsigma} = \{ \varsigma, D_1, D_2, \ldots, D_n, \mathcal{R} \}$, where

- each D_i ($1 \le i \le n$) denotes a set of tokens, which represent correlated instances of a business process. All tokens in $D_1, D_2, \ldots, D_n$ are correlated with ς.
- $\mathcal{R}$ is a binary relation defined between tokens in $\bigcup_{i=1}^{n} D_i$. Here, $d_x \mathcal{R} d_y$, (d_x, $d_y \in \bigcup_i D_i$), denotes that tokens d_x and d_y are correlated via token ς.

Here, ς is called *base token* of this correlation structure. For example, we re-depict the collaboration scenario of the motivating example with a Petri net as shown in Figure 4.8. Participating business processes are represented as individual sub net segments, which are distinguished with different circles, and the auxiliary places are marked as shaded circles. The tiny circles within places denote tokens in places. For simplicity, Figure 4.8 only shows part of the Petri net. Here, tokens such as ia_1, ib_1, ic_1 stand for participating business process instances, while transitions such as ta_2, tb_1, tc_1 stand for corresponding tasks in Figure 4.1. When ia_1 and ia_2 flow to transition tb_1 via auxiliary place ap_1, that means the production instances accepts the orders from two retailers. Therefore, correlation structure of r^{ib1} at this moment is $\{ ib_1, \{ ia_1, ia_2 \}, \varnothing \}$. Tokens ia_1 and ia_2 may have correlation structures $r^{ia1}=\{ ia_1, \{ ib_1 \}, \varnothing \}$ and $r^{ia2}=\{ ia_2, \{ ib_1 \}, \varnothing \}$, respectively.

This correlation structure accordingly evolves as the base token flows and interacts with other tokens. When ib_1 contacts ic_1, ic_2 and ic_3 to arrange the goods delivery for ia_1 and ia_2, we suppose that ib_1 assigns ic_1 and ic_2 to serve ia_1, while assigns ic_2 and ic_3 to serve ia_2. Thus, correlation structure r^{ib1} will change to $\{ ib_1, \{ ia_1, ia_2 \}, \{ ic_1, ic_2, ic_3 \}, \{(ia_1, ic_1), (ia_1, ic_2), (ia_2, ic_2), (ia_2, ic_3)\}\}$. Here, the last set denotes the correlated tokens via ib_1. As the consignee information, the order numbers from ia_1 and ia_2 are passed to ic_1 and ic_2, ic_2 and ic_3 by ib_1, respectively. Therefore, r^{ic1} is set as $\{ ic_1, \{ ib_1 \}, \{ ia_1 \}, \varnothing \}$, r^{ic2} is set as $\{ ic_2, \{ ib_1 \}, \{ ia_1, ia_2 \}, \varnothing \}$ and r^{ic3} is set as $\{ ic_3, \{ ib_1 \}, \{ ia_2 \}, \varnothing \}$.

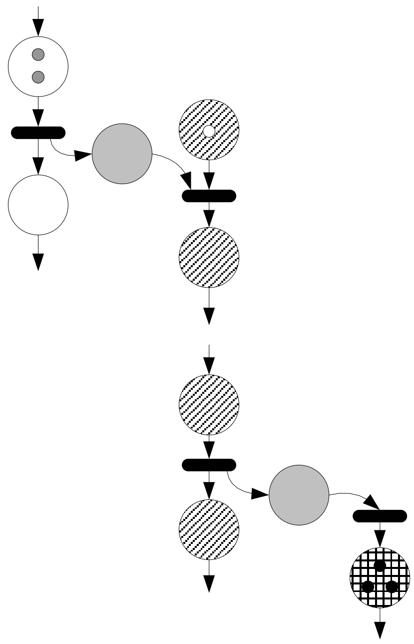

Figure 4.8 Correlation scenario

4.3.3 Correspondence Petri nets

According to the above discussion, we establish a novel correspondence Petri net (CorPN) by extending the traditional Place / Transition Petri net. The definition of this extended Petri net is given below.

Definition 4.4 *Correspondence Petri net*. A correspondence Petri net is represented as tuple $\Sigma = (\mathcal{P}, \mathcal{T}, \mathcal{F}, \mathcal{P}^{\circ}, \mathcal{F}^{\circ}, \mathcal{D}, \mathcal{V}, \mathcal{G}, \mathcal{E}, \mathcal{C}, Q, I)$, where

(*i*) $(\mathcal{P}, \mathcal{T}, \mathcal{F})$ is a directed net, called the base net of Σ. Here, $\mathcal{P}$, $\mathcal{T}$ and $\mathcal{F}$ stand for the sets of places, transitions and arcs, respectively. The sets comply with the following relations:

$\mathcal{P} \cap \mathrm{V} = \varnothing$; $\mathcal{P} \cup \mathcal{T} \neq \varnothing$; $\mathcal{F} \subseteq \mathcal{P} \times \mathcal{T} \cup \mathcal{T} \times \mathcal{P}$.

(*ii*) $\mathcal{P}^{\circ} \subset \mathcal{P}$, is the set of auxiliary places, which represent the messaging relations between component business processes inside a collaborative business process.

(*iii*) $\mathcal{F}^{\circ} \subset \mathcal{F}$, is the set of arcs that connect auxiliary places, i.e., $\mathcal{F}^{\circ} \subseteq \mathcal{P}^{\circ} \times \mathcal{T} \cup \mathcal{T} \times \mathcal{P}^{\circ}$.

(iv) $\mathcal{D}$ is a set of tokens, each of which stands for a possible participating business process instance. Here, $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2 \cup \ldots \cup \mathcal{D}_n$, $\mathcal{D}_i \cap \mathcal{D}_j = \varnothing$, where $1 \le i, j \le n$ and $i \ne j$. Precisely, each $\mathcal{D}_i$ denotes a *token group*, which includes the instances of the same business process. n is the number of token groups.

$\mathcal{V}$ is a set of variables for token groups, and $\mathcal{V} = \{ v_1, v_2 \ldots, v_n \}$. Actually, each element v_i of $\mathcal{V}$ is defined on a token group, i.e., $v_i \in \mathcal{V}$. v_i is defined on $\mathcal{D}_i$, where $1 \le i \le n$ and n is the number of token groups.

(vi) $\mathcal{G}: \mathcal{P} \rightarrow \tau$, where each element τ_i of set τ is a set of possible tokens, i.e., $\tau_i \in \tau$ and $\tau_i \in 2^{\mathcal{D}}$.

(vii)$\mathcal{E}: \mathcal{F} \rightarrow \sigma$, where σ is a set of expressions defined on $\mathcal{V}$.

($viii$)$\mathcal{C}: \mathcal{F}^{\circ} \rightarrow \varepsilon$, where ε is the set of cardinality parameters, i.e., $\varepsilon = \{ [:1], [:n] \}$.

(ix) $\mathcal{Q}: \mathcal{D} \rightarrow \lambda$, where λ is a set of correlation structures.

(x) $\mathcal{I}: \mathcal{P} \rightarrow \theta$, where θ is a set of possible composition of tokens defined in $\mathcal{D}$.

Explanation:

(1) ($\mathcal{P}$, $\mathcal{T}$, $\mathcal{F}$) determines the component net structures of this CorPN.

(2) $\mathcal{P}^{\circ}$ and $\mathcal{F}^{\circ}$ describe the messaging behaviours between the business processes belonging to the underlying collaborative business process.

(3) The variables in $\mathcal{V}$ are defined according to each token group, which represents the instances of a business process. Thus, the variables can be used to differentiate the instances of participating business processes and abstract the common behaviours of each business process.

(4) Mapping $\mathcal{G}$ sets up the capacity of each place defined in $\mathcal{P}$.

(5) Mapping $\mathcal{E}$ sets up the arc expressions to restrict the flowing of tokens.

(6) Mapping C maps a cardinality parameter onto each arc that connects with an auxiliary place.

(7) Mapping Q combines a correlation structure to each token, and this evolving correlation structure is responsible for recording tokens that correlated with the combined token. Actually, the combined token is the base token of this correlation structure.

(8) Mapping I denotes the initial distribution of tokens.

4.3.4 Mapping to relative workflow model

The CorPN can be easily mapped to a business process defined by the relative workflow model. Table 4.1 lists each component of the relative workflow model as well as the corresponding component of the CorPN model.

Table 4.1. Mapping from relative workflow model to CorPN model

Components of the relative workflow model	*Components of the CorPN model*
Instance of a local workflow process	Token
Instance ID of a local workflow process	Token ID
Local workflow process or perceivable workflow process	Sub net
Relative workflow process	Composite net
Activity	Transition
Possible execution state	Place
Messaging behaviour	Auxiliary place
Link	A series of arcs

The CorPN's structural components, such as places, arcs and transitions, represent the process level information of the relative workflow model, while the tokens represent the instance level information. Therefore, a CorPN can work equivalently as a relative workflow process. Moreover, the execution of a business process can be simulated by the token's flowing.

The procedure for generating a relative workflow process from several local workflow processes and perceivable workflow processes corresponds to the procedure for assembling several sub CorPNs into a composite CorPN. The following section discusses this assembling procedure in detail.

4.4 Applying Correspondence Petri Nets

4.4.1 Generating correspondence Petri nets

In this section, we demonstrate how to generate corresponding components of a CorPN to represent the collaborative business process discussed in the motivating example.

First, we need to collect the participating business processes belonging to this collaborative business process, as well as the messages to use. As Figure 4.1 shows, three business processes, viz., product-ordering process, production process and shipping process, are involved in the motivating example. In addition, messages like 'Product Order', 'Shipment Order' etc., are used for business communications across organisational boundaries.

Algorithm 4.1 details the procedure for constructing base net segments, i.e., the separate structures for involved business processes. In Algorithm 4.1, function *insertPN*(Σ, p) first converts a business process p into a place / transition net structure, and then insert this net into CorPN Σ; function *tokens*(p) returns the set of tokens that represent the instances of business process p; function *transitions*(p, Σ) returns the set of transitions that are generated for business process p; function *places*(p, Σ) returns the set of places that are generated for business process p; function *arcs*(p, Σ) returns the set of arcs which are generated for business process p; function *outArcs*(t) returns the set of arcs that are linking out from transition t.

Algorithm 4.1 Constructing base net segments

Input:
WP the set of participating business processes.

Output:
Σ: the CorPN tuple that is updated with base net segments.

```
set Σ = null;
for each business process p ∈ WP {
      insertPN( Σ, p );
      create variable v defined on tokens( p );
      Σ.𝒟 ← tokens( p ); Σ.𝒱 ← v ; // generate token set and variable set
      for each tk ∈ tokens( p ) {
         Σ.𝒬 ←( tk → {tk} );    // initialise correlation structures
      }
      set tempA = null;
      for each transition t ∈ transitions( p, Σ ); {
         if t is a token-generating transition then {
           for each arc a ∈ outArcs( t, Σ ) {
              tempA ← a ;
              Σ.ℰ ←( a→2^v );
              // 2^v denotes this arc allows any tokens that v stands for.
           }
         }
      }
      for each place pl ∈ places( p, Σ ) {
          Σ.𝒢 ←( pl→ v );
      }
      for each arc a ∈ arcs( p, Σ ) – tempA {
         Σ.ℰ ←( a→v );
      }
}
```

This algorithm constructs base net segments by realising the sets defined in CorPN tuple. Firstly, we build up token set $\mathcal{D}$ and variable set $\mathcal{V}$. With $\mathcal{D}$ and $\mathcal{V}$, we set up place capacity expression set $\mathcal{G}$ and arc expression set $\mathcal{E}$ to designate the flowing range of tokens. For each business process, a unique variable v is specified to differentiate instances of this business process from other business processes. In regard to token producible transitions, we mark a variable symbol 2^v to adjacent outgoing arcs to represent the possibility of all available tokens defined for this business process. Correlation set $\mathcal{C}$ has also been initialised in this procedure.

Algorithm 4.2 presents the procedure for assembling these separate segments together into a CorPN for the underlying collaborative business process. As this CorPN is created at process level instead of instance level, messages types are therefore used in this algorithm instead of message instances.

In Algorithm 4.2, function *transition*(Σ, t) returns the transition that stands for task t in CorPN Σ; function *link*(t / p, p / t) creates an arc linking transition t to place p, or place p to transition t, and t or p can also be set null to denote an undetermined transition or place; function *priorP* / *posteriorP* (Σ, tr) returns the prior / posterior place of transition tr in CorPN Σ; function *priorA* / *posteriorA* (Σ, tr) returns the prior / posterior arc of transition tr in CorPN Σ; function *relink*(Σ, a / p, p / a) adjusts a half-determined arc a to connect to / from place p in CorPN Σ.

Algorithm 4.2 Assembling net segments

Input:

MSG the set of unidirectional message types used by business processes in *WP*.

Σ the CorPN tuple obtained by Algorithm 1.

Output:

Σ' the CorPN tuple that is updated with auxiliary places, corresponding arcs etc.

```
set Σ′ = Σ; ∏ = null; Ω = null; sendingArcs = ∅;
for each m∈MSG {
  if m.α = 'out' then { // handling for outgoing message types
    tempT = ∅;   // create a half-determined arc for each outgoing message type
    a = link ( transition( Σ′, m.ϑ ), null ); Σ′.𝓕°← a;
    for each t′∈ m.β {
       Σ′.𝒞°← ( a → m.f( t′ ));
       tempT ← transition( Σ′, t′ ); sendingArcs ← a;
    }
    ∏←( a→ tempT );
  else {            // handling for incoming message types
    tempA = ∅;
    for each task t′ ∈ m.β {  // decompose the message-receiving transition into a
       create transition tr;   // series of transitions, please refer to Figure 6 (b)
       a = link( priorP( transition( Σ′, m.ϑ ), tr ); Σ′.𝓕°←a;
       b = link( tr, posteriorP( transition( m.ϑ ) );
       c = link( null, tr ); Σ′.𝓕°← c; Σ′.𝒞°← ( c→m.f( t′ ) );
//create a half-determined arc for each potential incoming route of this message type
       Ω ←( transition( Σ′, t′ ) → c );
    }
    Σ′.𝒯= Σ′.𝒯- { transition( Σ′, m.ϑ )};
    Σ′.𝓕 = Σ′.𝓕- { priorA( Σ′, transition( m.ϑ )), posteriorA( Σ′, transition( m.ϑ ))};
```

```
22.     }
23. }
24. for each a ∈ sendingArcs {  // link half-determined arcs with proper auxiliary places
25.     create auxiliary place px;
26.     relink( Σ′, a, px );
27.     for each transition tr∈ ∏( a ) {
28.         b = Ω( tr );
29.         ∏←( a→( ∏( a )-{ a })); Ω←( tr→( Ω( tr ) –{ b } ));
30.         relink( Σ′, px, b );
31.     }
32. }
```

In this algorithm, line 4 to line 10 first generates arcs for outgoing message types, and line 12 to line 21 generates arcs for incoming message types. At this stage, these generated arcs are half-determined ones, because we only designate one end of an arc while leave the other end open. To keep the information of multiple receivers or senders of a message, two mapping functions, $\prod$ and Ω, are used to record the correspondence between the interaction participating transitions and the generated half-determined arcs. Based on these two mappings, line 24 to line 32 generates auxiliary places and re-links the open ends of those half-determined arcs to proper auxiliary places. In this way, Algorithm 4.2 can connect the separate segments generated by Algorithm 4.1 together according to the messaging behaviours between participating business processes.

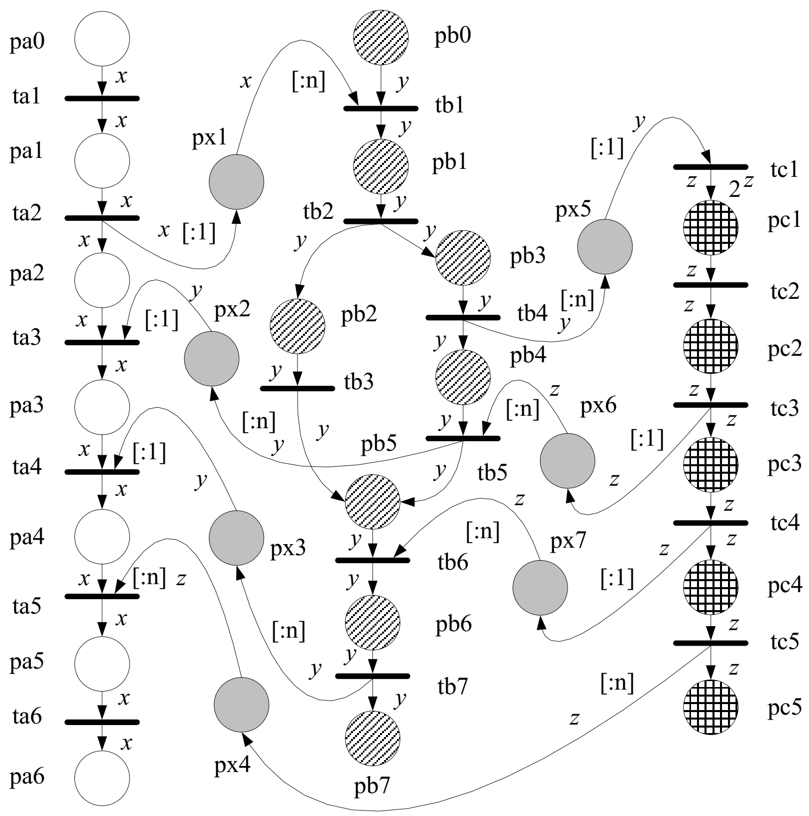

Figure 4.9 CorPN for a collaborative business processes

With Algorithms 4.1 and 4.2, we can generate a CorPN as shown in Figure 4.9 for the collaborative business process of the motivating example. Places and transitions are all labelled in Figure 4.9. The sub net segments for different business processes are distinguished with different circles, and the auxiliary places are marked as shaded circles.

Each sub net segment is also identified by exclusive variables over arcs. Different from traditional Petri net modelling for single business process [75], this CorPN can represent the collaboration between multiple business processes. For this reason, such an extended Petri net may own more than one starting place and ending place.

4.4.2 Run time execution

As discussed in Section 4.3, workflow correlations are initialised when two or more business process instances interact for the first time. During interactions, a participating instance may inherit some existing workflow correlations from its counterparts in case that this interaction relates with previous correlations. To update these correlations, each business process instance needs to modify its correlation structure every time after 'shaking hands' with partner business process instances. For example, when the manufacturer contacts shippers for goods delivery, the manufacturer's production instance may update its correlation structure with the correlations between retailers' product-ordering instances and shippers' assigned shipping instances. In the meantime, these shipping instances also update their correlation structures with the production instance and retailers' product-ordering instances that are to be served.

As for retailers' product-ordering instances, they may not know these new correlations until the manufacturer notifies them of the delivery date after booking deliveries. Actually, to timely update their correlation structures, retailers need to proactively trace such potential correlations rather than passively wait for feedbacks from their partners. To trace potential correlation information, an organisation can propagate enquiries among its partner organisations.

From above discussion, we see that correlation handling comprises two major procedures, i.e., to generate correlations after business process instances interact, and to trace existing correlations through coupled instances. In our CorPN context, two algorithms are proposed to support these procedures, respectively.

Algorithm 4.3 details the procedure for updating correlation structures after two or more business process instances 'shake hands'. Following the organisation-oriented view, we classify participating tokens into local tokens and foreign tokens. Local tokens represent the host organisation's business process instances that participate in this interaction. Foreign tokens represent the instances of partner business processes that participate in this interaction. In this algorithm, function *TYPE*(*setTK*) returns which token group that tokens in *setTK* belong to; function *relatedTK*(*tk*, *setTK*, ψ) returns the set of tokens correlated with token *tk* from set *setTK* during interaction ψ ; function *update*(*tk*, *setTk*) updates the content of token *tk*'s correlation structure with tokens in *setTk*. The details of function *update* are also given at the end of Algorithm 4.3.

Algorithm 4.3 Updating correlation structures

Input:

Σ	A CorPN.
ψ	a real interaction.
localTK	the set of participating local tokens during interaction ψ.
foreignTK	the set of participating foreign tokens during interaction ψ.

Output:

```
        Σ′      the updated CorPN.
 1.  set f = null; set Σ′ = Σ;
 2.  for each tk ∈ localTK {
 3.          setTK′= relatedTK( tk, foreignTK, ψ );
 4.          update( tk, setTK′ );
             // update the correlation structures of local tokens.
 5.          for each tk° ∈ setTK′ {
 6.              f←( tk°, tk );
 7.          }
 8.  }
 9.  for each tk° ∈ foreignTK {
10.          update( tk°, f( tk°));
     // update the correlation structures of foreign tokens.
11.  }


       // function update is given below
     update( tk, setTK )
 1.    r^tk = Σ′.Q( tk );
 2.    if ∃D_i, D_i ∈ r^tk ( TYPE(D_i)=TYPE(setTK)) then {
 3.        r^tk.D_i←setTK;
 4.    }
 5.    else {
 6.        r^tk.D_i←{ setTK };
 7.    }
 8.    for each tk_1 ∈ ∪_i r^tk.D_i, tk_2 ∈ setTK {
 9.        if tk_1 is coupled with tk_2 via tk then {
10.            r^tk.R←(tk_1, tk_2);
11.        }
12.    }
```

Once an interaction occurs between two business processes, each participated business process instance needs to run Algorithm 4.3 to update its correlation structure. For each local token, this algorithm searches all participated tokens for the correlated ones with this local token. This job is done by line 2 to line 8. Line9 to line 11 calls function *update* to update these correlated tokens in the correlation structures of local tokens. In addition, function *update* also generates proper tuples in relation R of each participated local token's correlation structure, if there exist tokens that are correlated via this local token. As discussed for correlation structures in Section 4.3.2, product-ordering instances are correlated with shipping instances via the production instance, when the manufacturer contacts shippers for goods delivery.

Algorithm 4.4 describes the procedure for tracing potentially correlated tokens. An organisation may use this algorithm to proactively detect correlated business process instances for its own business process instance. In this algorithm, function *update*(*tk*, *setTk*) is of the same content with the one in Algorithm 4.3.

Algorithm 4.4 Tracing correlated tokens

Input:

tk° the original token to update correlation structure.
Σ a CorPN.

Output:

Σ' the updated CorPN.

1. **set** $\Sigma' = \Sigma$;
2. $List = \varnothing$; $oldList = \varnothing$;
3. $r^{tk^{\circ}} = \Sigma.Q(tk^{\circ})$;
4. $List \leftarrow \bigcup_i r^{tk^{\circ}}.D_i$; *// List is used to store the tokens to check.*
5. **do while** $List \neq \varnothing$ {
6. **select** $tk \in List$;
7. **remove** tk from $List$;
8. $oldList \leftarrow tk$; *// oldList is used to store the checked tokens.*
9. **for** each $tk' \in \bigcup_i r^{tk}.D_i$ {
10. **if** $\exists(tk^{\circ}, tk') \in r^{tk}.\mathcal{R} \wedge tk' \notin oldList$ **then {**
11. $List \leftarrow tk'$;
12. **}**
13. **}**
14. **}**
15. $update(tk^{\circ}, oldList)$;

This tracing procedure, from line 5 to line 14, follows a depth-first strategy to search for correlated tokens. After finding correlated tokens, the host organisation updates the retrieved tokens to its correlation structure by invoking function *update*. This correlation structure determines the logical instance of the business process instance represented by the token.

This procedure may be called upon request by the host organisation, for example, at a point that a retailer wants to know shippers' details while waiting for goods delivered by several shippers. Therefore, we do not have to derive this correlation structure for all instances involved in a collaborative business process in advance.

4.5 Summary

This chapter has proposed a method to specify instance correspondences in the context of collaborative business processes. In this method, we have developed unidirectional cardinality parameters to characterise correspondences between instances of different business processes at build time. We also have defined workflow correlations to identify actual correspondences between instances of different business processes at run time. For precise representation, we establish a novel CorPN model with the proposed cardinality parameters and correlation structures, as well as auxiliary places and token-generating transitions etc. In this CorPN based approach, particular algorithms have been presented to formalise the procedure for assembling separate business processes into a collaborative business process. Furthermore, the procedures for specifying workflow correlations and tracing workflow correlations on the fly are also formalised by corresponding algorithms.

Chapter 5

Tracking over Collaborative Business Processes

Workflow tracking is defined as the function of monitoring and tracing the execution of a business process instance. Typically, workflow tracking belongs to instance level business process management. In the context of collaborative business processes, workflow tracking may go beyond organisational boundaries to cover the business processes of partner organisations. Therefore, workflow tracking brings challenges to the representation of dynamic structure of collaboration, the awareness beyond neighbouring organisations and well-balanced openness of such awareness for privacy protection etc. Based on the relative workflow model and instance correspondence research discussed before, this chapter proposes a comprehensive solution for inter-organisational workflow tracking.

Most traditional workflow monitoring approaches, such as WfMC Monitor and Audit specification [76, 77], BEA Weblogic Integration [78], IBM WebSphere MQ Workflow [79], the agent based workflow monitoring [61] and the customisable workflow monitoring [59], are mainly applicable either in an intra-organisational environment or in an inter-organisational environment yet without privacy concern. To our best knowledge, there is little discussion on workflow monitoring in a privacy sensitive inter-organisational environment.

Aiming to fill this gap, this chapter presents a matrix based framework for inter-organisational workflow tracking. The relative workflow model is employed to guarantee the business privacy during collaborations. In this framework, each participating organisation may derive tracking structures over its relative workflow processes and the involved relevant business processes of partner organisations. Thus, the organisation can perform inter-organisational workflow tracking with the generated tracking structures.

The remainder of this chapter is organised as follows. Section 5.1 analyses requirements of workflow tracking in a privacy sensitive environment with a motivating example. In Section 5.2, according to our proposed relative workflow model, some fundamental rules for workflow tracking are discussed, and a set of representation matrices are introduced. Based on these rules, several matrix operations are presented in Section 5.3 for tracking structure generation, together with the algorithms for generating tracking structures and performing tracking.

5.1 Requirement Analysis with Motivating Example

Figure 5.1 shows a collaboration scenario, which is simplified from the motivating example in Chapter 3. In this collaboration scenario, a retailer collects orders from customers, and then purchases products from a manufacturer. The manufacturer may contact a shipper for booking product delivery while making goods with supplies from a supplier. In this scenario, a retailer may track the collaborative business process as follows: After placing an order with a manufacturer, the retailer may contact the manufacturer and enquire about the execution status of the production process by referring, say the order number. Furthermore, after the manufacturer organises product shipping for the retailer by a shipper, the retailer may also contact the shipper via the manufacturer and enquire about shipping information. However, the retailer may not be allowed to enquire about the goods supply information, as that could be confidential information of the manufacturer and is therefore hidden from the retailer. For a manufacturer, it may track the same collaborative business process differently. Besides the retailer and shipper, the manufacturer can also track the supplier for goods supply information.

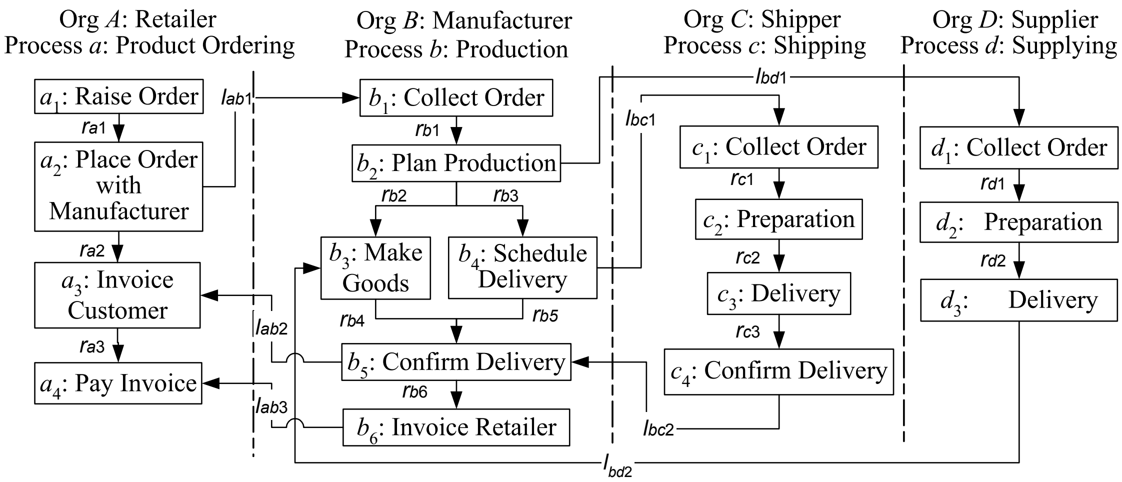

Figure 5.1 Inter-organisational collaboration example

From this scenario, we can see that:

1. A participating organisation may require tracking other organisations for its involved part of a collaborative business process;

2. Each participating organisation may track same collaborative business process differently.

The first point requires collaboration between participating organisations, which is fundamental to inter-organisational workflow tracking. The second point, however, requires that a participating organisation is treated as a fully autonomous entity and can provide different visibilities to different organisations. Obviously, the public view approaches cannot meet the second requirement. Yet, from flowing sections, we can see that our relative workflow approach can well support these two issues.

5.2 Relative Workflows and Tracking Structures

5.2.1 Relative workflow setting up

In the motivating example discussed in last section, we suppose that the following visibility constraints can be inferred from the appropriate perceptions defined in the example.

$p^{Manufacturer.Production}_{Retailer}.\mathcal{VC}$ = {("collect order", `Contactable`), ("plan production", `Invisible`), ("make goods", `Trackable`), ("schedule delivery", `Trackable`), ("confirm delivery", `Contactable`), ("invoice retailer", `Contactable`)}

$p^{Retailer.ProductOdering}_{Manufacturer}.\mathcal{VC}$ = {("raise order", Invisible), ("place order with manufacturer", Contactable), ("invoice customer", Contactable), ("pay invoice", Contactable)};

$p^{Shipper.Shipping}_{Manufacturer}.\mathcal{VC}$ = {("collect order", Contactable), ("preparation", Invisible), ("delivery", Trackable), ("confirm delivery", Contactable)};

$p^{Shipper.Shipping}_{Retailer}.\mathcal{VC}$ = {("collect order", Invisible), ("preparation", Trackable), ("delivery", Trackable), ("confirm delivery", Trackable)};

$p^{Supplier.Supplying}_{Manufacturer}.\mathcal{VC}$ = {("collect order", Contactable), ("preparation", Invisible), ("delivery", Contactable)}.

These visibility constraints only allow a partial view of the collaborative business process from each organisation's individual perspective. Such partial views are subject to the diverse partnerships between participating organisations.

Figure 5.2 (a) and (b) show the partial views upon the whole collaborative business process from the retailer and the manufacturer, respectively. For simplicity, Figure 5.2 renames retailer, manufacturer, shippers and suppliers as org *A*, org *B*, org *C* and org *D*, while marks tasks as numbers in proper business processes.

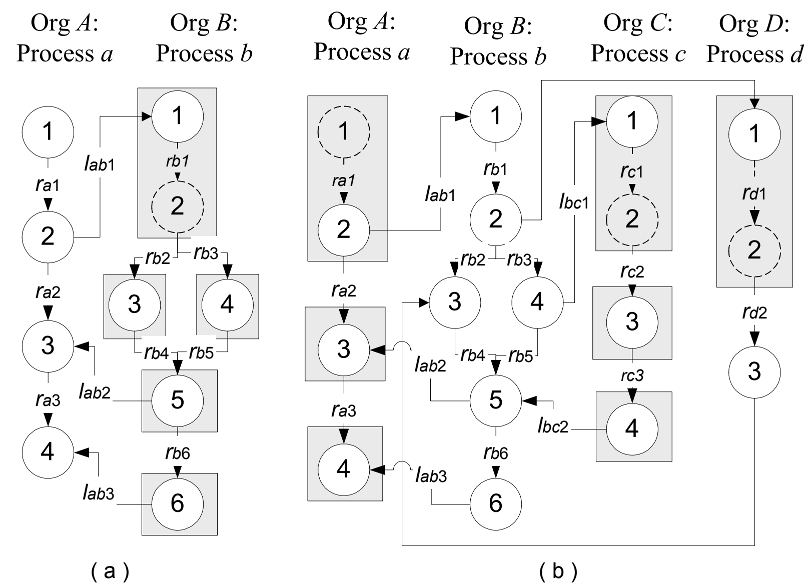

Figure 5.2 Relative workflow and tracking structure examples

Since the retailer and the supplier have no partner relationship in the collaborative business process, they do not define perceptions for each other.

The tasks with dashed circles denote the invisible tasks. These two diagrams clearly illustrate that the relative workflow processes for same collaborative business process may be different from the perspectives of different organisations. This reflects the relativity characteristics of our relative workflow approach.

5.2.2 Representation matrices

To accurately describe our relative workflow model, we establish several matrices to formally represent key concepts of the relative workflow model.

Definition 5.1 *Self Adjacency Matrix*. An n-task business process p of organisation g is represented by a special matrix, called Self Adjacency Matrix (SAM), which is defined as,

$$D_{g\ n\times n}^{p} = [d_{ij}], \text{ where } d_{ij} = \begin{cases} r, \text{ if exists link } r \text{ linking task } t_i \text{ and task } t_j, \text{ where } i < j; \\ 0, \text{ otherwise.} \end{cases}$$

Each element of an SAM denotes an intra process link between tasks, such as r_{a1} and r_{b2} in Figure 5.1. As a link connecting tasks t_i and t_j is put in d_{ij}, not d_{ji}, where $i<j$, D_g^p is always an upper triangular matrix. For example, process a in Figure 5.1 can be represented by SAM $D_A^a = \begin{pmatrix} 0 & r_{a1} & 0 & 0 \\ 0 & 0 & r_{a2} & 0 \\ 0 & 0 & 0 & r_{a3} \\ 0 & 0 & 0 & 0 \end{pmatrix}$. Similarly, $D_B^b = \begin{pmatrix} 0 & r_{b1} & 0 & 0 & 0 & 0 \\ 0 & 0 & r_{b2} & r_{b3} & 0 & 0 \\ 0 & 0 & 0 & 0 & r_{b4} & 0 \\ 0 & 0 & 0 & 0 & r_{b5} & 0 \\ 0 & 0 & 0 & 0 & 0 & r_{b6} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$ and

$$D_C^c = \begin{pmatrix} 0 & r_{c1} & 0 & 0 \\ 0 & 0 & r_{c2} & 0 \\ 0 & 0 & 0 & r_{c3} \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

A self adjacency matrix can be used to represent not only a local workflow process but also a perceivable workflow process, a relative workflow process, or a tracking structure, which will be introduced later.

When composing a local workflow process p into a perceivable workflow process for organisation g, the composition is subject to the visibility constraints defined in proper perceptions. According to the discussion upon this composition in Chapter 3, we formalise the composition process as a particular matrix, called transformation matrix.

Definition 5.2 *Transformation Matrix*. A transformation matrix (TM) is an $n\times n$ triangular 0-1 matrix, for representing the composition of a local workflow process into a perceivable workflow process under visibility constraints, which is defined as,

$$T_g^{p}{}_{n\times n} = [t_{ij}], \text{ where } t_{ij} = \begin{cases} 1, \text{ if task } t_j \text{ is composed into task } t_i \ (j \neq i), \text{ or not composed } (j = i); \\ 0, \text{ otherwise.} \end{cases}$$

This matrix can be directly derived from the visibility constraints defined in the corresponding perception, following the task composition algorithm discussed in Section 3.3.2. Note, each column has only one element with value "1", because each task can be composed only once or may not be composed at all. For example, the procedure for composing local workflow process b into a perceivable workflow process for organisation A can be described by TM $T_A^b = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$. This composing procedure is conducted by the visibility constraints defined in perception $p_A^{B.b}$.

Likewise, we can calculate that $T_A^c = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$.

Finally, the relevant messaging links between two business processes are represented by another matrix, called boundary adjacency matrix.

Definition 5.3 *Boundary Adjacency Matrix*. The mmessaging links between two business processes, p_1 and p_2, from the perspective of organisation g, can be represented by an $m\times n$ matrix called boundary adjacency matrix (BAM), where m is the number of tasks belonging to p_1, and n is the number of tasks belonging to p_2. A BAM is defined as follows,

$$B_g^{p_1|p_2}{}_{m\times n} = [b_{ij}], \text{ where } b_{ij} = \begin{cases} l, \text{ if exists messaging link } l \text{ connecting } p_1.t_i \text{ and } p_2.t_j \\ 0, \text{ otherwise.} \end{cases}$$

For example, the interaction relationship between local workflow process *b* and perceivable workflow process *c* at the site of organisation *B*, can be represented by BAM $B_B^{b|c} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ l_{bc1} & 0 & 0 & 0 \\ 0 & 0 & 0 & l_{bc2} \\ 0 & 0 & 0 & 0 \end{pmatrix}$. Similarly, $B_A^{a|b} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ l_{ab1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & l_{ab2} & 0 \\ 0 & 0 & 0 & 0 & 0 & l_{ab3} \end{pmatrix}$.

5.2.3 Tracking structure

From the discussion in Section 5.1, we see that a tracking structure acts as an organisation's observable view upon the execution progress of a collaborative business process. Technically, a tracking structure is different from a relative workflow process. A relative workflow process is created by messaging links connecting to *contactable* tasks of neighbouring organisations, therefore a relative workflow process only represents the collaboration between neighbouring organisations. Yet, a tracking structure may go beyond neighbouring organisations through *trackable* tasks.

Unlike the "contactable" visibility value defined in Table 3.1 of Chapter 3, the "trackable" value is designed for tracking purpose and can be set on the tasks of the business processes belonging to non-neighbouring organisations. We define a tracking structure for each relative workflow process, and this tracking structure can be created by including trackable tasks from non-neighbouring organisations. The definition of a tracking structure is given below:

Definition 5.4 *Tracking Structure*. A tracking structure *ts* for organisation *g*'s relative workflow process *rp* consists of the following tasks and links.

- The tasks include: (*i*) the tasks of relative workflow process *rp*; (*ii*) the union of task sets of perceivable workflow processes that are *reachable* from *g*. These perceivable workflow processes may belong to *g*'s neighbouring or non-neighbouring organisations. The reachability of a perceivable workflow process from an organisation is to be discussed later.
- The links include: (*i*) the links of relative workflow process *rp*; (*ii*) the union of link sets of perceivable workflow processes that are reachable from *g*; (*iii*) the set of

messaging links between perceivable workflow processes that are *visible* from g. The visibility of a messaging link from an organisation is to be discussed later.

5.2.4 Rules

From the definition of a tracking structure, we need to first define the visibility of a messaging link and the reachability of a perceivable workflow process from an organisation. Basically, these two things both rely on the visibility of tasks. Therefore, we establish a series of rules to determine the visibility of workflow tasks and links, and reachability of perceivable workflow processes.

- Intra Process Visibility Rule:

If a task t in organisation g_1's local workflow process $g_1.lp$ is set invisible to organisation g_2, then t is hidden by composing it into a visible (contactable or trackable) task of $g_1.lp$. The links connecting t will be changed accordingly. The composition procedure will be discussed in the *composition operation* in next section. After composition, $g_1.lp$ becomes a perceivable workflow process $g_1.lp_{g2}$.

- Inter Process Visibility Rule:

A messaging link l connecting two perceivable workflow processes is said *visible* to organisation g, if and only if both tasks connected by l are visible to g.

- Expansion Rule:

Let ts be the tracking structure for a relative workflow process of organisation g. A perceivable workflow process outside ts is said *reachable* and therefore can be included into ts, if and only if it has at least one *visible* messaging link connecting a task inside ts.

Following the Intra Process Visibility Rule, the original link r_{b1} connecting tasks b_1 and b_2 of process b in Figure 5.1 becomes invisible in its perceivable form for organisation A in Figure 5.3. This is because task b_2 is invisible to organisation A. Correspondingly, links r_{b2} and r_{b3} which connect b_2 and b_3, b_2 and b_4 in Figure 5.1, respectively, are now changed to connect b_1 and b_3, b_1 and b_4, in Figure 5.3. Following the Inter Process Visibility Rule, messaging link l_{bc1} connecting task b_4 and task c_1 is not visible while messaging link l_{bc2} connecting task b_5 and task c_5 is visible in Figure

5.1. Following the Expansion Rule, the perceivable workflow processes of process *c* is reachable from organisation *A*, because of the existence of the visible messaging link l_{bc2}. By applying all these rules, we can finally generate a tracking structure shown in Figure 5.3 for *A*'s relative workflow process shown in Figure 5.2 (a).

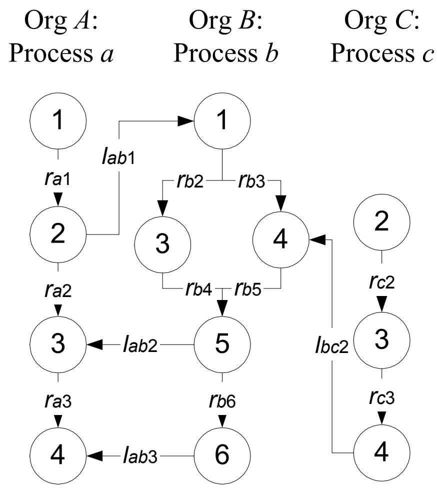

Figure 5.3 Tracking structure from the retailer's perspective

5.3 Generating Tracking Structures

5.3.1 Operations

According to the rules discussed in last section, we define three matrix operations for tracking structure derivation.

- Composition Operation

As defined in the TM for a local workflow process, each element with value "1" in a non-diagonal position (i, j) stands for a procedure for composing the composed task t_j to the composing task t_i. Under the restriction of the Intra Process Visibility Rule, the following sub rules may apply to this composition:

(1) a link connecting t_j and t_k ($k \neq i$) is changed to a link connecting t_i and t_k;

(2) a link connecting t_i and t_k ($k \neq j$) is unchanged;

(3) a link connecting t_i and t_j is discarded.

The first sub rule requires an operation that can be applied to the SAM defined for a local workflow process. This operation first adds the elements in row j to their corresponding elements in row i, and then sets all elements in row j to zero. This can be achieved by applying a matrix multiplication to this TM and the SAM defined for the local workflow process. Function $f_{reshape}$ is assigned to reshape the result matrix into an upper-triangular form.

For input matrix $M_{n\times n}$, function $f_{reshape}$ is defined as

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

The second sub rule identifies the case that needs no action. From the definition of a TM, we can see that the composing tasks of this case all have value "1" on the diagonal line, which takes no effect in the matrix multiplication.

With respect to the third sub rule, we need to check whether there exists a link connecting t_i and t_j in the corresponding TM. This can be easily achieved by checking whether there exists a row that has value "1" at both column i and column j. We can represent the existence of such a link by a boolean expression, i.e. $|f_{row}(i)= f_{row}(j)|$, where $f_{row}(x)$ defines a function that returns the row where column x has the value "1".

Finally, these three sub rules can be merged together as an operation $\otimes$, which is defined on $T_{n\times n}\otimes D_{n\times n}=[\,|f_{row}(i) \neq f_{row}(j)| \cdot \sum_{x=1}^{n} t_{ix} \cdot d_{xj}\,]_{n\times n}$. Hence, organisation g_1 may apply a *Composition Operation* on a local workflow process p to generate a perceivable workflow process for g_2. This composition operation can be defined as

$$D_{g_2}^{g_1 \cdot p} = f_{reshape}(T_{g_2}^{g_1 \cdot p} \otimes D_{g_1}^{g_1 \cdot p})$$

Here $D_{g_1}^{g_1 \cdot p}$ and $T_{g_2}^{g_1 \cdot p}$ are the SAMs of g_1's local workflow process p and the corresponding TM for perception $p_{g2}^{g1.p}$, respectively.

By applying this composition operation, organisations B and C can generate perceivable workflow processes b and c for organisation A in the form of $D_A^b = f_{reshape}(T_A^b \otimes D_B^b) = \begin{pmatrix} 0 & 0 & r_{b2} & r_{b3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & r_{b4} & 0 \\ 0 & 0 & 0 & 0 & r_{b5} & 0 \\ 0 & 0 & 0 & 0 & 0 & r_{b6} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$ and $D_A^c = f_{reshape}(T_A^C \otimes D_C^c) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & r_{c2} & 0 \\ 0 & 0 & 0 & r_{c3} \\ 0 & 0 & 0 & 0 \end{pmatrix}$.

- Connection Operation

According to the aforementioned Inter Process Visibility Rule, we need to identify the visible messaging links between perceivable workflow processes. Afterwards, we can determine which perceivable workflow processes of non-neighbouring organisations can be included in the tracking structure for a given organisation. For this purpose, we need to identify the visible tasks by simply checking elements valued "1" in diagonal positions of the corresponding TM. We use function f_{diag} to diagonalise TM T into a diagonal matrix T°. Function f_{diag} is defined as follows,

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

The visible messaging link between two business processes, for example, g_1's p_1 and g_2's p_2, from the perspective of another organisation, say g_3, can be represented as BAM $B_{g_3}^{g_1\cdot p_1|g_2\cdot p_2}$. The *Connection Operation* connecting $g_1.p_1$ and $g_2.p_2$ for g_3 can be defined as

$$B_{g_3}^{g_1\cdot p_1|g_2\cdot p_2} = (f_{diag}(T_{g_3}^{g_2\cdot p_2}) \cdot (f_{diag}(T_{g_3}^{g_1\cdot p_1}) \cdot B_{g_1}^{g_1\cdot p_1|g_2\cdot p_2})^T)^T$$

This connection operation first requires g_1 to diagonalise TM $T_{g_3}^{g_1\cdot p_1}$, and then perform a matrix multiplication on the diagonalised $T_{g_3}^{g_1\cdot p_1}$ and BAM $B_{g_1}^{g_1\cdot p_1|g_2\cdot p_2}$. g_2 will subsequently use the diagonalised matrix $T_{g_3}^{g_2\cdot p_2}$ to multiply the result matrix from g_1. In the connection operation, proper transposition operations are needed to align the columns of the left hand matrix with the rows of the right hand matrix for matrix multiplication.

Regarding the motivating example given in Section 5.2, organisations B and C can generate matrix $B_A^{b|c}$ for organisation A to provide the visible messaging links between B's process b and C's process c in A's view.

$$B_A^{b|c} = (f_{diag}(T_A^c) \cdot (f_{diag}(T_A^b) \cdot B_B^{b|c})^T)^T = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & l_{bc2} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix};$$

- Extension Operation

The aforementioned Expansion Rule is used for extending the tracking structure to include perceivable workflow processes of both neighbouring and non-neighbouring organisations. Technically, an extension step can be represented as an *Extension Operation*. With a local workflow process p_1 in the tracking structure, organisation g_1 may apply the extension operation to include local workflow process p_2 of organisation g_2 in the tracking structure. This extension operation can be defined as $D_{g_1}^{g_1 \cdot p_1 | g_2 \cdot p_2} = \begin{pmatrix} D_{g_1}^{g_1 \cdot p_1} & B_{g_1}^{g_1 \cdot p_1 | g_2 \cdot p_2} \\ 0 & D_{g_1}^{g_2 \cdot p_2} \end{pmatrix}$.

For example, the tracking structure containing processes a and b from the view of organisation A can be described by a composite SAM $D_A^{a|b} = \begin{pmatrix} D_A^a & B_A^{a|b} \\ 0 & D_A^b \end{pmatrix}$, which is obtainable through this extension operation.

5.3.2 Generation algorithm

The tracking structure generation can be technically considered as a process that appends a new generated column each time that a reachable business process is detected. This new generated column consists of a new SAM and a series of new BAMs. The new SAM describes the inner structure of this detected business process, while the new BAMs describe the interaction relationships between the detected business process and the existing processes of the structure.

As shown in Figure 5.4, at the starting point, the tracking structure contains only $D_{g_1}^{g_1 \cdot p_1}$. This means that only $g_1.p_1$ is included. Afterwards, g_1 detects that perceivable workflow process $g_2.p_2$ is reachable from $g_1.p_1$, 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 tracking structure. Likewise, organisation 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 $D_{g_1}^{g_3 \cdot p_3}$, when g_2 detects that process $g_3.p_3$ is reachable from $g_1.p_1$ via $g_2.p_2$. This appending process continues until all reachable perceivable workflow processes are detected. The inter process interaction relationships can only be identified by the organisation (*context organisation*) that owns the "bridging" business processes, by which the expansion proceeds. Therefore, a propagation mechanism is adopted to spread this detection process over all involved organisations. The context organisation for an appending step may change from time to time. Organisation g_1 is called the *original context organisation* of this tracking structure.

$$
\left(D_{g_1}^{g_1 \cdot p_1} \right) \Rightarrow \begin{pmatrix} D_{g_1}^{g_1 \cdot p_1} & B_{g_1}^{g_1 \cdot p_1 | g_2 \cdot p_2} \\ 0 & D_{g_1}^{g_2 \cdot p_2} \end{pmatrix} \Rightarrow \begin{pmatrix} D_{g_1}^{g_1 \cdot p_1} & B_{g_1}^{g_1 \cdot p_1 | g_2 \cdot p_2} & B_{g_1}^{g_1 \cdot p_1 | g_3 \cdot p_3} \\ 0 & D_{g_1}^{g_2 \cdot p_2} & B_{g_1}^{g_2 \cdot p_2 | g_3 \cdot p_3} \\ 0 & 0 & D_{g_1}^{g_3 \cdot p_3} \end{pmatrix}
$$

$$
\Rightarrow \begin{pmatrix} D_{g_1}^{g_1 \cdot p_1} & B_{g_1}^{g_1 \cdot p_1 | g_2 \cdot p_2} & B_{g_1}^{g_1 \cdot p_1 | g_3 \cdot p_3} & \cdots & B_{g_1}^{g_1 \cdot p_1 | g_n \cdot p_n} \\ 0 & D_{g_1}^{g_2 \cdot p_2} & B_{g_1}^{g_2 \cdot p_2 | g_3 \cdot p_3} & \cdots & B_{g_1}^{g_2 \cdot p_2 | g_n \cdot p_n} \\ 0 & 0 & D_{g_1}^{g_3 \cdot p_3} & .. & B_{g_1}^{g_3 \cdot p_3 | g_n \cdot p_n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & D_{g_1}^{g_n \cdot p_n} \end{pmatrix}
$$

Figure 5.4 Tracking structure evolving process

We note that the evolving process shown in Figure 5.4 starts from g_1's local workflow process $g_1.p_1$ instead of g_1's relative workflow process $g_1.rp$. Actually, $g_1.rp$ can be generated by the first step of the evolving process when g_1 is the context organisation.

Algorithm 5.1 details the generation procedure. In algorithm 5.1, function *relatedProc*(p) returns a set of local workflow processes and perceivable workflow processes that have direct interactions with process p; Function *includedProc*(*trackStruc*) returns all included business processes at that moment in tracking structure *trackStruc*, which initially contains an SAM defined on a local workflow process of the original context organisation; Function *BAM*(p_1, p_2, g) returns the BAM between processes p_1 and p_2 from the view of organisation g, using the connection operation; Function *SAM*(p, g) returns the SAM of process p from the view of organisation g, using the composition operation; Function *genOrg*(p) returns the organisation of process p.

Algorithm 5.1 *genTrackStruc* - Tracking Structure Generation

Input:

trackStruc	-	A tracking structure matrix
cxtProc	-	A local workflow process of the context organisation
origCxtOrg	-	The original context organisation that starts the generation
Output:		
trackStruc	-	The expanded tracking structure matrix

Step 1 *Detect business processes*

detectedProcSet = *relatedProc*(*cxtProc*);
includedProcSet = *includedProc*(*trackStruc*);
detectedProcSet = *detectedProcSet* – *includedProcSet*;

Step 2 *Expand the tracking structure*

appendedProcSet = ∅;
for each process $p_i \in$ *detectedProcSet* **{**
tempB = *BAM*(*cxtProc*, p_i, *origCxtOrg*);
if *tempB* is a non-zero matrix **then {**
newColumn = NULL;
for each process $p_j \in$ *includedProcSet* **{**
B = *BAM*(p_j, p_i, *origCxtOrg*);
Append *B* to *newColumn.*
/* generate related boundary adjacency matrices of the new column*/
}
D = *SAM*(p_i, *origCxtOrg*);
/* generate the self adjacency matrix of the new column */
Append *newColumn* and *D* to *trackStruc*, using extension operation.
includedProcSet = includedProcSet $\cup \{ p_i \}$;
appendedProcSet = *appendedProcSet* $\cup \{ p_i \}$;
}
}

Step 3 *Propagate the detection process*

for each process $p_i \in$ *appendedProcSet* **{**
targetOrg = genOrg(p_i);

```
        /* Ask targetOrg to call genTrackStruc */
        trackStruc = targetOrg.genTrackStruc( trackStruc, p_i, origCxtOrg );
    }
  Step 4              Return the expanded tracking structure
    return trackStruc;
```

The tracking structure generation process starts from a local workflow process of the original context organisation, and then spreads to all reachable business processes of the involved organisations. When this generation process comes to an organisation, this organisation becomes the context organisation of Algorithm 5.1.

For example, if we start from the retailer's product ordering process, i.e., process *a* in the motivating example, this algorithm first detects the business processes having direct interactions with process *a*. Then it checks for each detected business process whether it is reachable from organisation *A*, and if so, the detected process will be included to the tracking structure. In this step, organisation *B*'s process *b* is included, and the tracking structure is expanded to $D_A^{a|b} = \begin{pmatrix} D_A^a & B_A^{a|b} \\ 0 & D_A^b \end{pmatrix}$. After that, this generation process will be propagated to *B*, and *B* repeats the above steps to extend the tracking structure. At this stage, *B* may find process *c* and process *d*, while only process *c* is reachable from organisation *A* and is therefore included. This is because that the retailer and the supplier do not set up perceptions for each other in this example, and hence no transformation matrix is defined for process *d* from *A*. Therefore, the tracking structure is finally expanded to $D_A^{(a|b)|c} = \begin{pmatrix} D_A^a & B_A^{a|b} & B_A^{a|c} \\ 0 & D_A^b & B_A^{b|c} \\ 0 & 0 & D_A^c \end{pmatrix}$, which equals to the diagram shown in Figure 5.3. Here, $B_A^{a|c}$ is a zero matrix because there are no direct interactions between processes *a* and *c*, and the other sub matrices can be found from the former part of this chapter.

5.3.3 Performing workflow tracking

As discussed in Chapter 4, instance correspondences couple the collaborating business process instances together into a logical instance of a collaborative business process. In the solution proposed in Chapter 4, instance correspondences are characterised in terms of workflow cardinality and instance correlation, which are represented by cardinality parameters and correlation structures, respectively. In the inter-organisational workflow tracking context, these correlation structures are also used to record the coupling information for run time workflow tracking.

Figure 5.5 shows the data structure that is designed for workflow tracking. This data structure is constructed by a series of lists. Each list contains the set of instances belonging to a specific local workflow process. Each unit of a list records the execution status of the business process instance. Besides, each unit also maintains a correlation structure for the instance. With the content of these correlation structures, the coupling relation between collaborating business process instances can be identified as links connecting these units.

The tracking process is similar to a graph traversal process, where the nodes represent the related business process instances and the arcs represent their messaging links to be tracked. In addition, new participating business process instances will be identified at the time when visible messaging links are fired.

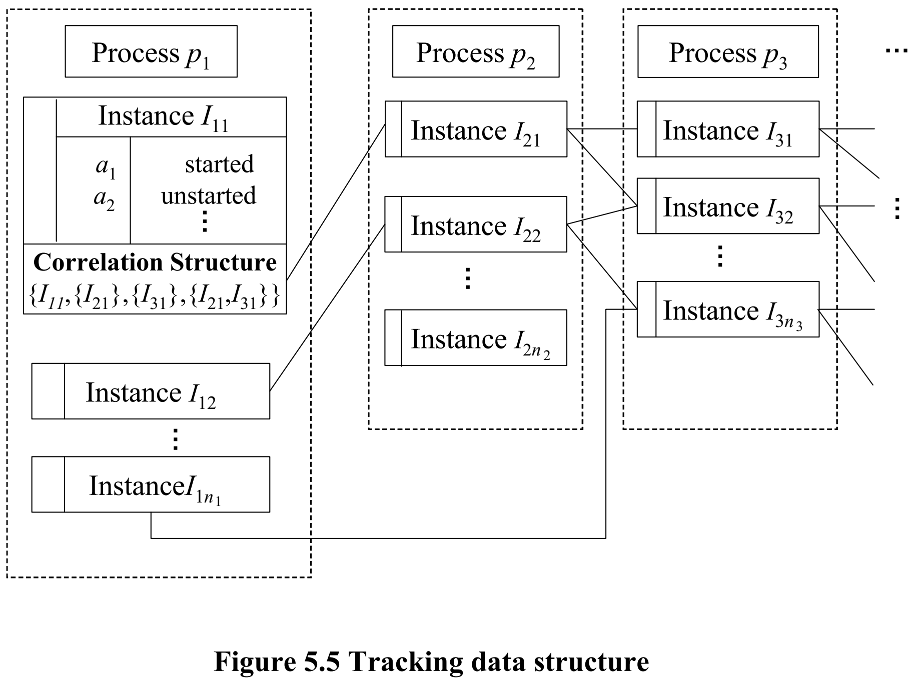

Figure 5.5 Tracking data structure

Algorithm 5.2 details the procedure for performing workflow tracking. In this algorithm, function *addInstance*(p, i) inserts instance i to the list of process p in the tracking data structure; Function *addLink*(i_1, i_2) creates a link between instances i_1 and i_2 in the tracking data structure; Function *linkedInstances*(i, *trackStruc*) returns the instances linked to instance *i* in the tracking data structure, according to the tracking structure *trackStruc*; Function *relatedBAMs*(*p*, *trackStruc*) returns the set of BAMs related to process *p*, defined in *trackStruc*; Function *partnerProc*(*B*, *p*) returns the partner process of *p* defined in BAM *B*; Function *genOrg*(*p*) returns the organisation of process *p*; Function *genProc*(*i*) returns the process of instance *i*.

Algorithm 5.2 *trackProc* - Tracking Process

Input:

trackStruc	-	The tracking structure to conduct the tracking
origInstance	-	An instance of the original context organisation's initial local workflow process defined in *trackStruc*
DS	-	The tracking data structure
Output:		
DS	-	The updated tracking data structure

Step 1 *Initialisation*

trackInstanceSet = ∅;
stack s=new stack( );
s.push(*origInstance*);

Step 2 *Discover the participating business process instances*

while *s* is not empty **{**
  cxtInstance = *s.pop*( );
  foundInstanceSet = *linkedInstances*(*cxtInstance*, *trackStruc*) – *trackInstanceSet*;
  for each *i* ∈ *foundInstanceSet* **{**
    s.push(*i*);
    cxtProc = *genProc*(*cxtInstance*);
    BAMset = *relatedBAMs*(*cxtProc*, *trackStruc*);
    for each link *l* of each boundary adjacency matrix *B*∈ *BAMset* **{**
/* now, start discovering business process instances by following each visible messaging link */
      partnProc = *partnerProc*(*B*, *cxtProc*);
      partnOrg = *genOrg*(*partnProc*);
      if *cxtInstance.l* is newly fired **then {**
        newInstanceSet=∅;
        Ask *partnOrg* to check any new participating instances of *partnProc*, and set the instances to *newInstanceSet*.
        newInstanceSet = *newInstanceSet* – *trackInstanceSet*;
/* filter the previous discovered instances */
        for each *i* ∈ *newInstanceSet* **{**
          addInstance(*partnProc*, *i*);
          addLink(*cxtInstance*, *i*);
/* update the tracking data structure */
          s.push(*i*);
/* and add the newly discovered instance to the stack */

```
            }
        }
    }
trackInstanceSet = trackInstanceSet ∪ { cxtInstance };
/* the set of instances to track */
}
```

Step 3 *Update the execution status of participating business process instances*

```
for each instance i ∈ trackInstanceSet {
   p = genProc( i );
   targetOrg = genOrg( p );
   Enquire targetOrg for the execution status of i, and then update the status of i in DS.
}
```

This algorithm starts from a local workflow instance of the original context organisation. Following the corresponding tracking structure, this algorithm searches along visible messaging links and propagates the execution status queries to all reachable business process instances. The corresponding tracking structure records the interaction relationship between the processes of these reachable business process instances. When an inter-organisational interaction is fired, the algorithm will check whether any new business process instance joins the business collaboration. If so, the algorithm will add these business process instances to the tracking data structure.

5.4 Summary

This chapter contributed to the study of workflow tracking across organisational boundaries. Compared with other workflow tracking solutions, the approach proposed in this chapter not only enables an organisation to track other organisations for its involved parts of collaborative business processes, but also allows different organisations track same collaborative business process differently.

In this chapter, we have deployed a matrix-based framework which comprises three representation matrices and three matrix operations. Algorithms have been presented to illustrate how to use these matrices and operations to generate tracking structures and perform workflow tracking. With the help from the relative workflow model, this framework guarantees the privacy protection during inter-organisational workflow tracking. The framework also supports a tracking structure to evolve dynamically, and therefore adapts the flexibility of collaborative business process management. With the generated tracking structures, an organisation can proactively trace the execution progress of its involved part of a collaborative business process.

Chapter 6

Case Studies

Based on the organisation-oriented view methodology and corresponding mechanisms introduced in Chapters 3, 4 and 5, this chapter presents two case studies of modern business collaboration applications. These two case studies demonstrate the deployment of our organisation-oriented view methodology, and the advantages in supporting business collaboration. Section 6.1 presents a case study on a virtual organisation alliance for tool making, which is featured by low trustiness, uni-directional contracting and agile cooperation. Section 6.2 presents a case study on a transient supply chain for dairy production, which requires supports for highly scalable structure, flexible partnerships, as well as tracking and tracing.

6.1 Case Study 1: Virtual Organisation Alliance

With the trend of booming global business collaborations, organisations are required to streamline their business processes to form a virtual organisation [2, 80]. A virtual organisation defines a trading community as a set of participating organisations for conducting collaborative business processes. Normally, the building blocks of a collaborative business process are the pre-existing business processes of participating organisations. Therefore, it is fundamental that a collaborative business process knows how the business process belonging to different organisations are linked together for cooperation [81, 82]. While this kind of cooperation is a pre-requisite, organisations must act as autonomous entities during business collaboration. Besides, certain levels of privacy of participating organisations have to be guaranteed. Many existing inter-organisational business process approaches align the related business processes of different organisations, into a public view business process [34, 54, 83-85]. This public view neutralises the diversity of the perception on collaborative business processes from different organisations, and fails to support business privacy sufficiently. In this case study, we analyse the feasibility of deploying our organisation-oriented view methodology in a virtual organisation alliance.

6.1.1 Introduction

Two characteristics, i.e. the dynamic structure and the collaboration openness, distinguish virtual organisation alliances from traditional federated organisations. Moreover, these two characteristics also raise challenges to manage the collaborative business processes for virtual organisation alliances, especially at contracting and collaboration design phases. The temporary and dynamic cooperation relationship requires high flexibility in describing and implementing collaboration processes between member organisations. Furthermore, the temporary and dynamic partnership in turn results in the lack of trustiness between member organisations in loosely coupling business collaborations, and therefore complicates the authority control [86, 87]. Here, a case study on a virtual organisation alliance in toolmaking filed is discussed to illustrate how we apply our organisation-oriented view methodology to support these two characteristics.

Australian toolmaking firms are relatively small and specialised, operating with minimal business infrastructure in an attempt to control overhead costs. This specialisation restricts access to additional customers or larger projects. In response to this increasing dilemma, toolmakers need to become effective in engaging and servicing a more geographically disperse clientele, and complementary toolmakers need to pool their resources. Technology-enabled collaboration can assist with dealing with this industry deficiency. [88] In this chapter, we apply our organisation-oriented view methodology to support collaboration behaviours of a virtual organisation alliance for these toolmaking firms.

As Figure 6.1 shows, a virtual organisation alliance in toolmaking field may hold designers, manufacturers, prototypers and marketing companies together to collaboratively work for customer products.

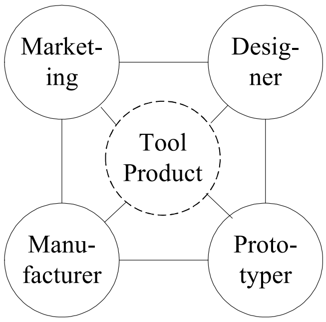

Figure 6.1 Toolmaking VOA

With this background, we narrow our focus down to business scenarios where exist diverse business collaborations between four member organisations, viz. organisation *A*, *B*, *C* and *D*. Figure 6.2 illustrates three business collaboration scenarios between the four member organisations. For simplicity, we only give key tasks of the involved business processes.

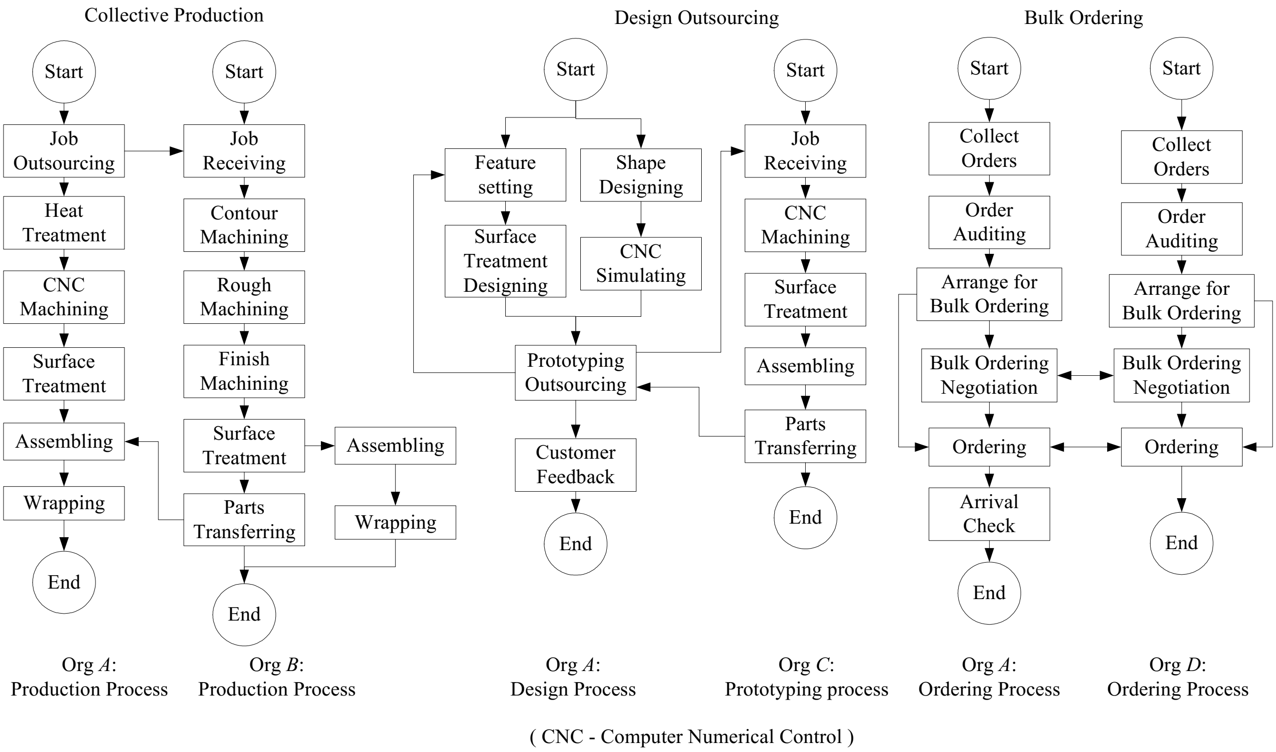

Figure 6.2 Business collaborations

In the scenario of collective production shown in Figure 6.2, organisation *A*'s production process uses organisation *B*'s production service, which is supported by organisation *B*'s production process. Organisations *A* and *B* produce different kinds of parts, respectively, and finally assemble and package them into unitised tools at the site of organisation *A*. This collaboration is motivated by the production capability requirement, and reflects the synergy for small-to-medium sized organisations.

In the scenario of design outsourcing, we suppose that organisation *C* is stronger in prototyping. Thus, organisation *A* outsources its prototyping task to organisation *C*, for the efficiency of time and cost. This collaboration involves the interaction between organisation *A*'s design process and organisation *C*'s prototyping process.

The scenario of bulk ordering reflects the economic of scale production, since the organisations with orders for the same parts or parts from the same supplier batch their orders together for a more economical price. This collaboration involves organisation *A*'s ordering process and organisation *D*'s ordering process.

6.1.2 Supports for virtual organisational alliance

Normally, B2B collaboration originates by contracting, where two or more parties come to an agreement to cooperate for a common objective, and this agreement is regulated by a legal document of contract [89]. As mentioned in Chapter 3, reference [52] has modelled a contract as four major parts of *Who*, *What*, *How* and *Legal clauses*. The *How* part defines the execution details for the obligations: When and which services are to be delivered? What is the deadline? Which clause will apply when a party falls behind its obligation? These details together describe the necessary *business interactions* for the collaboration.

Since a virtual organisation alliance enables the collaborations with a broad range of potential partners, each member organisation is empowered to quickly assemble the resources and expertise to capture emerging opportunities. To keep these options open, the partnerships between organisations are not static, but rather continuously evolve to stay competitive on the market. Correspondingly, this open partnership requires an open contracting mechanism, where an organisation posts the business services that it can offer and it may request to all potential co-operators in the virtual organisation alliance. Thereafter, some organisations with special interests may respond by referring to the business services. Finally, the involved organisations can come to negotiate the details of the contract for the collaboration. We call the organisation that issues the contract a *host organisation*, and the responding organisations *partner organisations*.

Compared with the traditional closed contracting process, this open contracting process has the following features.

- Low trustiness.

Since the contract may be established between parties with no prior partnerships, high trustiness can hardly be granted. In such a low trustiness environment, organisations require more authority control to eliminate the privacy vulnerabilities. Particularly designed for privacy protection, our relative workflow model uses a visibility constraint based visibility control mechanism to guarantee the finer granularity of business process perception between collaborating organisations. With these visibility constraints, participating organisations can intentionally choose which tasks to be hidden or revealed to partner organisations, according to the trustiness level and the necessity of interactions for collaborations.

- Uni-directional contracting.

Traditional contracting process defines concrete parties at the starting time, while the open contracting process only involves a single party at the beginning, i.e. the host organisation. Therefore, a uni-directional contracting process suits the relative workflow generation process. In the uni-directional contracting process, an organisation browses the published perceivable workflow processes from other organisations; then this organisation may contact proper organisation for contract negotiation; and finally, the two organisations sign contracts and create final perceivable workflow processes for partners to generate relative workflow processes.

- Agile collaboration.

Because the collaborating organisations share a loosely coupling relationship, the collaboration is dynamic with low coordination, interdependence, short duration and few transactions, and is therefore called agile collaboration. This agile collaboration requires high flexibility of collaboration structure and behaviours. Our relative workflow model supports a kind of “off-the-shelf” collaboration formation scheme. This scheme empowers organisations to choose partner organisations and define relative workflow processes with their own local workflow processes and perceivable workflow processes from partner organisations. In this scheme, each participating organisation acts as an autonomous entity and each organisation can change its partner organisations or redefine its collaborations dynamically to adapt the fast changing market opportunities.

6.1.3 Support at collaboration design phase

Once a contract is signed by all involved parties, participating organisations may come to the next phase, i.e. collaboration design phase. In this phase, each participating organisation designs and coordinates the business collaborations amongst partner organisations by linking related business processes.

At this stage, each participating organisation may participate in multiple collaborations with different groups of partner organisations at the same time. Furthermore, each participating organisation may choose and combine several collaborations into a comprehensive collaboration according to its own preferences and management. Hence, different participating organisations may own different forms of business collaborations. For this reason, the individual perspective of each participating organisation is better than a public perspective for representing the collaboration of the organisation. In particular, our organisation-oriented view methodology models a collaborative business process from a relative perspective, therefore can explicitly distinguish the perceptions of organisations. In consequence, this relative modelling perspective better supports the complicated partnership among organisations of a virtual organisation alliance.

From above discussion, we see that the relative perspective on collaborative business processes provides stronger representation for complex collaboration scenarios and partnerships. The visibility control mechanism protects the privacy during business collaborations, and the dynamic definition scheme guarantees the flexibility of business collaborations. In summary, the organisation-oriented view methodology well supports B2B collaborations at contracting and collaboration design phases in an open, loosely-coupled and low-trustiness application environment, such as virtual organisation alliance.

6.1.4 Business process setting

Now, we start from the bulk ordering collaboration to demonstrate how our organisation-oriented view methodology supports a virtual organisation alliance. In the scenario of the bulk ordering collaboration, when organisation *A* collects orders from its production department(s), it will consider whether to seek a bulk ordering with potential co-buyers. If needed, it will publish a request for bulk ordering of listed parts or materials, to all other member organisations in this alliance. Suppose that organisation *D* has the same things to buy, and then organisation *D* responds to organisation *A* to further negotiate the details about the amount for bulk ordering, the expected price, etc. Finally, a contract will be signed to regulate the agreement on bulk ordering, and the two organisations can conjoin their orders. In this scenario, the contract is motivated by seeking a more economical price, and the collaboration is supported by the business services of parts ordering of the two organisations. The underlying supporting business processes are organisation *A*'s ordering process and organisation *D*'s ordering process, respectively.

Since this collaboration mainly focuses on the bulk ordering negotiation, some tasks of ordering processes may be set invisible for the collaborating organisation, if these tasks do not directly participate in the bulk ordering negotiation. Here, we suppose that organisation *A* may set up the following visibility constraints on its ordering process for organisation *D*.

VC_1 = { ('Collect Orders', Invisible), ('Order Auditing', Invisible), ('Arrange for Bulk Ordering', Invisible), ('Bulk Ordering Negotiation', Contactable), ('Ordering', Contactable), ('Arrival Check', Invisible) }.

These visibility constraints are stored in perception $p_D^{A.orderingProcess}$, and they prohibit organisation *D*'s cognition on private tasks, such as "Collect Orders", "Order Auditing" and "Arrange for Bulk Ordering". These tasks only handle internal procedures, and do not participate in the bulk ordering collaboration. Therefore, this prohibition does not affect the negotiation with organisation *D*.

Similarly, perception $p_A^{D.orderingProcess}$ which is defined on organisation *D*'s ordering process from organisation *A*'s view, may have the following visibility constraints.

> VC_2 = { ('Collect Orders', Invisible), ('Order Auditing', Invisible), ('Arrange for Bulk Ordering', Invisible), ('Bulk Ordering Negotiation', Contactable), ('Ordering', Contactable) }.

Now, we can generate the relative workflow process for this bulk ordering collaboration from the perspective of organisation *A*, according to the visibility constraints defined in perception $p_A^{D.ordering\,\mathrm{Pr}ocess}$. Figure 6.3 shows the generated relative workflow process.

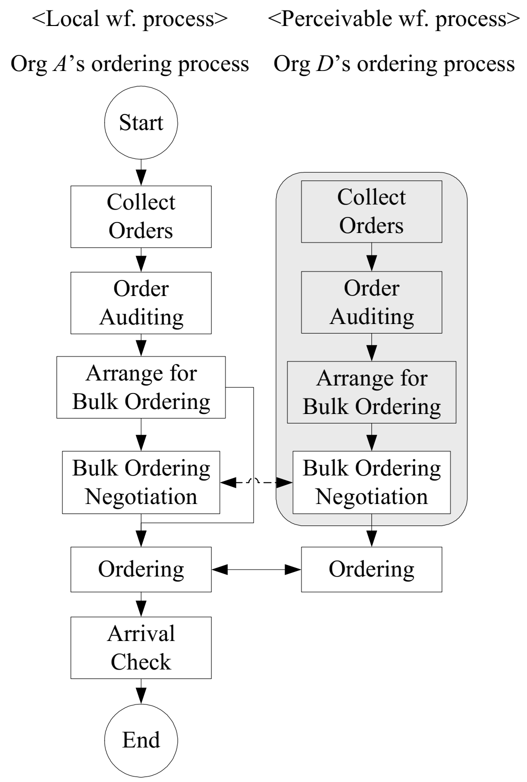

Figure 6.3 Relative workflow process for bulk ordering collaboration

The shadowed tasks of the perceivable workflow process shown in Figure 6.3 denote the invisible tasks to organisation *A*, and the clear tasks are either trackable or contactable ones.

Following this way, organisation *A* may also sign contracts with organisations *B* and *C*, for the collective production and design outsourcing. Therefore, organisation *A* is simultaneously participating in the three collaborations with organisations *B*, *C* and *D*, respectively. These three collaborations together support organisation *A*'s whole process of tools manufacturing. At the site of organisation *A*, it can generate a composite relative workflow process integrating all the three collaborations, to represent organisation *A*'s comprehensive manufacturing business collaboration.

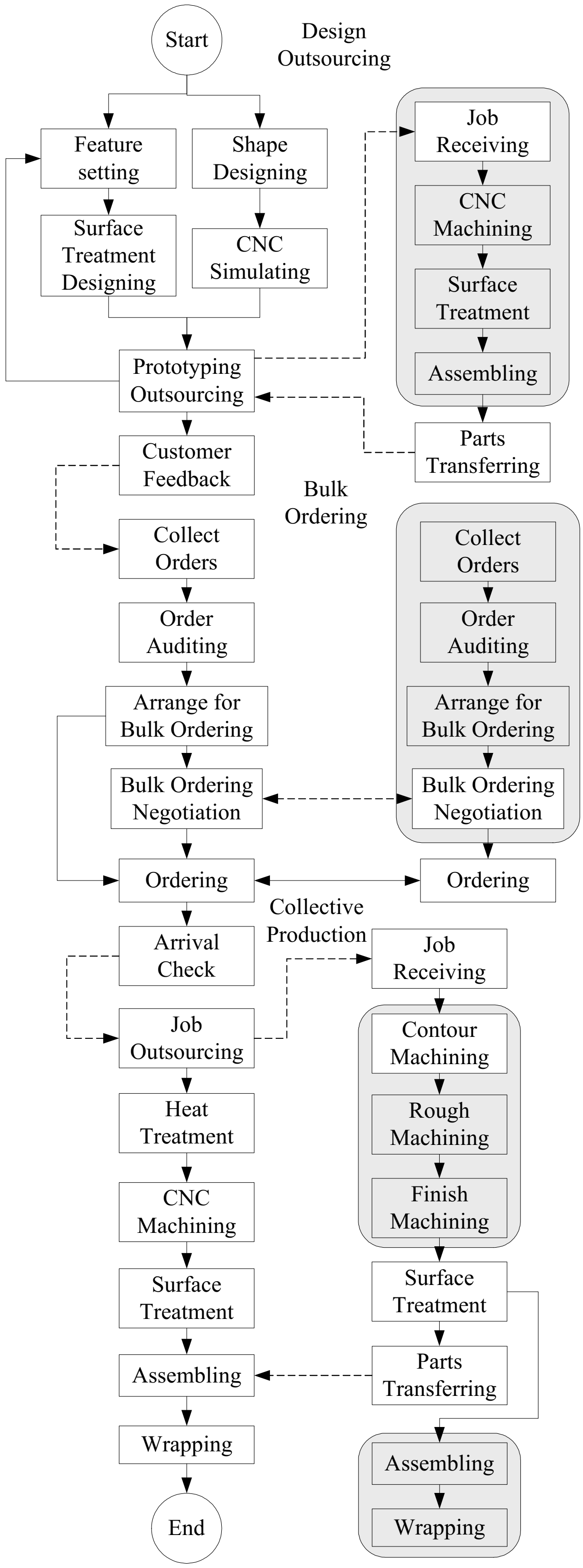

Figure 6.4 Final relative workflow process

Figure 6.4 gives the composite relative workflow process from organisation *A*'s view. This relative workflow process combines organisation *A*'s three local workflow processes, viz., engineering process, ordering process and production process. In addition, this relative workflow process includes three other business processes of its partner organisations, i.e. organisation *C*'s prototyping process, organisation *D*'s ordering process and organisation *B*'s production process, in their perceivable forms.

From the perspective of another participating organisation, say organisation *B*, it may perceive a different picture of this collaboration. As organisation *B* does not participate in the collaborations of bulk ordering or design outsourcing with organisation *A*, organisation *B* therefore may not have authorities to perceive those two collaborations. This means that organisation *B* may even not know the existence of these two collaborations. The relative workflow process generated from organisation *B*'s perspective is given in Figure 6.5, where organisation *D*'s ordering process and organisation *C*'s prototyping process are not visible.

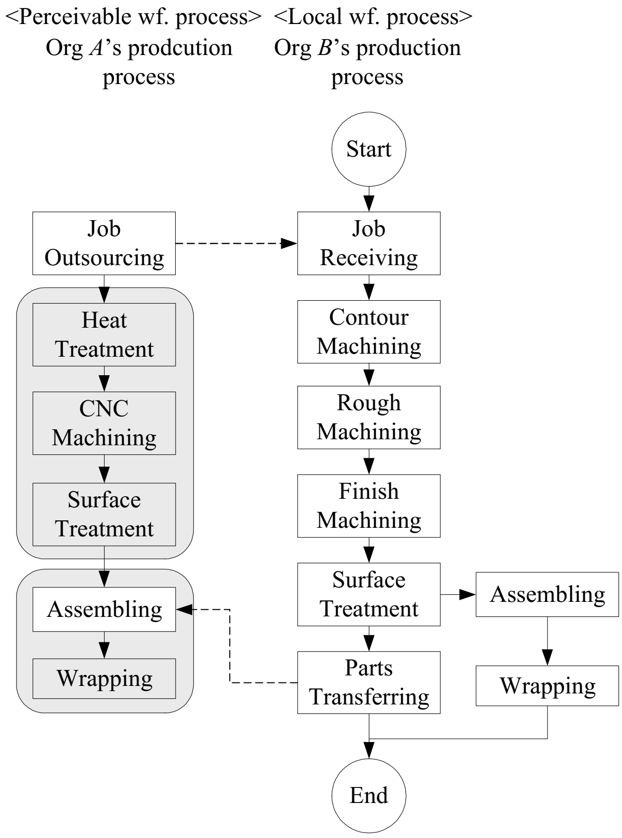

Figure 6.5 Relative workflow process from org B's view

From the relative workflow processes shown in Figure 6.4 and Figure 6.5, we can see that different organisations may hold different views towards the same collaboration. This reflects our relativity characteristics.

With this organisation-oriented modelling perspective, each organisation is in charge of choosing partners by issuing and signing proper contracts. In addition, each organisation is responsible for defining the collaboration structure and behaviours to fulfil its own business plans and objectives. Each organisation acts as an autonomous entity, and can change its partners or redefine its collaborations dynamically, to grasp the fast changing market opportunities. The visibility control mechanism prevents the privacy disclosure at both task level and process level. A participating organisation is now able to control the granularity of partner organisations' perception on its business processes during collaborations.

6.1.5 Summary

This Australian Tool making virtual organisation alliance represents a typical agile collaboration where co-makership is created between organisations on the fly. In this case study, benefits and advantages of our organisation-oriented view methodology are particularly demonstrated.

First, the customisable visibility constraints enable organisations to protect privacy information of their business processes. With these visibility constraints, organisations can establish collaborations safely and promptly in an open, loosely coupled and low-trustiness environment.

Second, the organisation-oriented modelling perspective empowers an organisation with high autonomy. Therefore, a designer company or a manufacturer company can establish a collaboration in accordance with its own business benefits or intention.

Finally, in the form of relative workflow processes, different participating organisations may create different collaborative business processes, which ideally adapt organisations' cognition to diverse partnerships between organisations.

6.2 Case Study 2: Transient Supply Chain

6.2.1 Introduction

Globalisation and technical innovation are driving a particular kind of virtual enterprise, where a dynamic network of inter-connected organisations, from suppliers' suppliers to customers' customers, work collaboratively to bring value to the marketplace. Accordingly, there emerges supply chain technology that combines information flows, product flows and payment flows, and changes the way that organisations produce goods and provide services.

Supply chains are expected by participating organisations to improve outsourcing processes, design smarter interfaces, recognise opportunities for aftermarket services, and develop mutually beneficial, flexible relationships with customers and suppliers. Particularly, current swift marketing opportunities require transient supply chains to hold strong resilience via flexibility for the purpose of adapting to the fast changing partnerships [90].

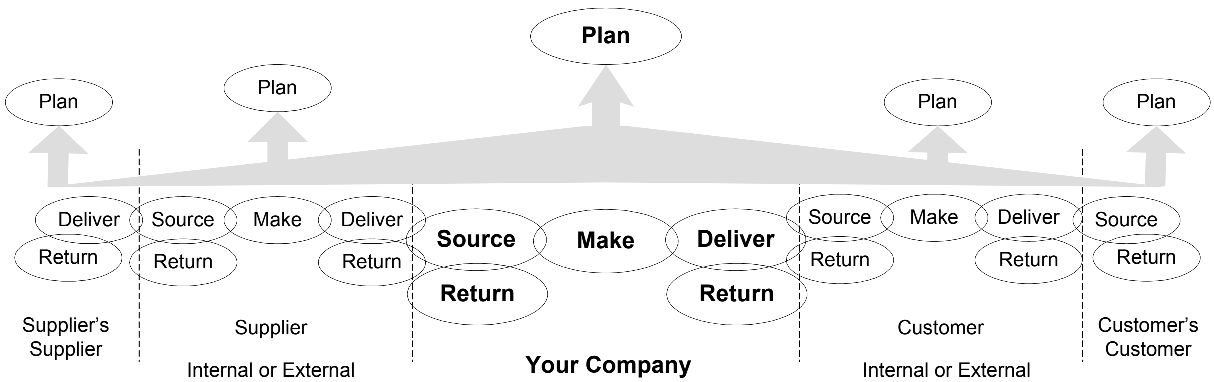

Figure 6.6 SCOR model

In the area of supply chain management, the Supply Chain Council (SCC) [91], an independent not-for-profit corporation, has developed and endorsed a Supply Chain Operations Reference model (SCOR) [92]. This SCOR model has been widely adopted as the cross-industry standard. The SCOR model is based on five distinct management processes: Plan, Source, Make, Deliver, and Return. Figure 6.6 illustrates the SCOR model with a brief overview of each management process. The five management processes are defined as below, respectively.

- *Plan* – The process that balances aggregated demand and supply to develop a course of action that best meets sourcing, production, and delivery requirements.

- *Source* – The process that procures goods and services to meet planned or actual demand.

- *Make* – The process that transforms products to a finished state to meet planned or actual demand.

- *Deliver* – The processes that provides finished goods and services to meet planned or actual demand, typically including order management, transportation management, and distribution management.

- *Return* – The process that is associated with returning or receiving returned products for any reason. These processes extend into post-delivery customer support.

Figure 6.7 shows a typical supply chain collaboration case, where a dairy company is linked with its suppliers, distributors and customers. In this supply chain scenario, the retailer grocer orders packaged milk from a packaging company. The packaging company sources milk and cardboard containers from a milk processing company and a container manufacturer, respectively. A paper mill is responsible for supplying paper to the container manufacturer. In addition, a competition may exist between the cardboard container manufacturer and a cardboard container importer. The packaging company may alternate to buy cardboard containers from the importer instead of the manufacturer, to respond the customers' preferences, or for cost and quality reasons etc.

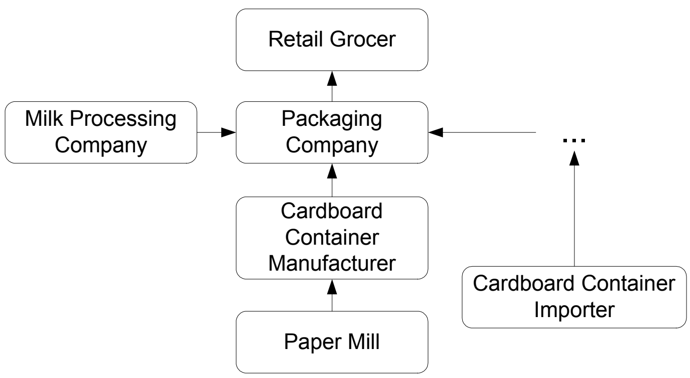

Figure 6.7 Supply chain example

The supply chain amongst these organisations is built up one chain by another. During this building process, new related business processes are included and assembled to the underlying collaborative business process for the supply chain. The business process of each participant conforms to the SCOR model. Instead of distinct processes, a business process comprising “plan”, “source”, “make” and “deliver” activities is used to represent a participating organisation’s internal procedure. For simplicity, the “return” activity is ignored in our analysis.

6.2.2 Supports for transient supply chains

A transient supply chain (or agile supply chain) is featured as a market sensitive supply chain. This means that the supply chain can read real market demands, and thereafter adjust its production plans according to the collected demands. In contrast, a traditional supply chain organises its planning and scheduling in a forecast-driven mode rather than demand-driven mode. In other words, a transient supply chain expects demands according to the feed-forward from the marketplace instead of the feed-back from past sales or shipments [90, 93].

A transient supply chain also tends to be ‘virtual’. This means that a transient supply chain is typified by information shared between upstream and downstream partner organisations. Shared information between partner organisations can only be fully leveraged through process alignments, which denote the collaborative work between buyers and suppliers, such as joint product development, vendor managed inventory, common systems, and synchronous operations. These forms of cooperation in a transient supply chain are highly prevailing, as companies focus on managing their core competencies and outsource all other activities. In summary, we characterise transient supply chain management with the following features:

- High scalable structures.

Since a supply chain covers the suppliers’ suppliers, and the customers’ customers, a supply chain turns to be a very broad network as it grows. This feature passes requirements for high scalability and extensibility from the underlying business process modelling method. The relative workflow model designs a collaborative business process by assembling other workflow processes, and this assembling mechanism theoretically allows unlimited extension of the modelled business process. With the relative workflow model, our organisation-oriented view methodology well supports the scalability and extensibility of supply chains at process level.

A supply chain always reconfigures itself to adapt the changing marketplace. The collaboration between participating suppliers and consumers therefore has to be dynamic and unpredictable. In practical scenarios, a supply chain always has to put up with the dynamic come-in and leave-out of organisations at run time. Different from a static community with a fixed number of fixed organisations, a supply chain handles a dynamic collaboration between a variable group of organisations. Our organisation-oriented view methodology can well adapt such variants by dynamically re-configuring corresponding relative workflow processes.

- Flexible partnerships.

A supply chain links all related partners together as a confederated network. Especially, a transient supply chain preferably constructs a network of smaller, specialised providers of resources, knowledge, and capabilities. Such a network is inherently more agile than its vertically integrated predecessor. Therefore, this network is more easily to reconfigure itself to meet the requirements from a changing marketplace. Due to this high livelihood, such a network often involves flexible partnerships within the partner organisations of this network.

- Tracking and tracing

Tracking and tracing are particularly important to supply chain management. The knowledge of resource dependencies in short or long terms highly facilitates the management of material flows in a supply chain. At technical level, such dependencies are obtainable from the information flow and execution status of the underlying collaborative business process. Our relative workflow research provides a matrix-based inter-organisational workflow tracking framework. This framework enables an organisation to proactively trace the execution status of related business processes of the collaboration, and track the instance correlations in a specific collaborative business process.

6.2.3 Supply chain evolving

As more participating organisations join in on the fly, a transient supply chain evolves dynamically. This evolving process typically reflects the scalable structure of a supply chain. In this section, we use the supply chain for dairy production to illustrate how our organisation-oriented view methodology facilitates the supply chain evolvement.

Here, we start from the collaboration between a packaging company and a retailer grocer. The collaboration is shown in Figure 6.8, where the related business processes are shown in a public view. The packaging company first initialises a production plan, then orders containers and milk. While receiving orders from retailers, the packaging company may redo the production planning and order more milk or containers if needed. The loop of planning, receiving orders and replanning reflects the phenomenon that the packaging company continuously responds to the customers' demand. With received containers and milk, the packaging company continues to package them together and send the products to the retailer.

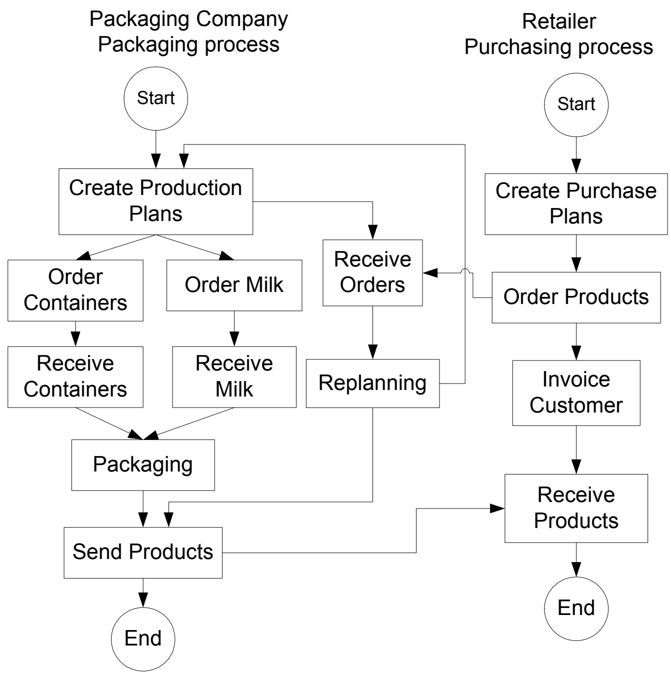

Figure 6.8 Collaborative business process between the packaging company and the retailer

In the relative workflow context, organisations set up visibility constraints to customise the openness level of their internal business processes to others. Here, we suppose the packaging company and the retailer define the following visibility constraints in corresponding perceptions.

$P_{Retailer}^{PakCompany.packagingProcess}.\mathcal{VC}$ = { (“Create Production Plans”, invisible), (“Order Containers”, invisible), (“Order Milk”, invisible), (“Receive Container”, invisible), (“Receive Milk”, invisible), (“Receive Orders”, contactable), (“Replanning”, invisible), (“Packaging”, trackable), (“Send Products”, contactable) };

$P_{PackagingCompany}^{Retailer.PurchasingProcess}.\mathcal{VC}$ = { (“Create Purchase Plans”, invisible), (“Order Products”, contactable), (“Invoice Customer”, invisible), (“Receive Products”, contactable) }.

According to these visibility constraints, the packaging company may generate a relative workflow process as shown in Figure 6.9. In the figure, the shadowed part denotes the perceivable workflow process, while a shadowed rectangle and a clear rectangle of the perceivable workflow process denote an invisible task and contactable task, respectively.

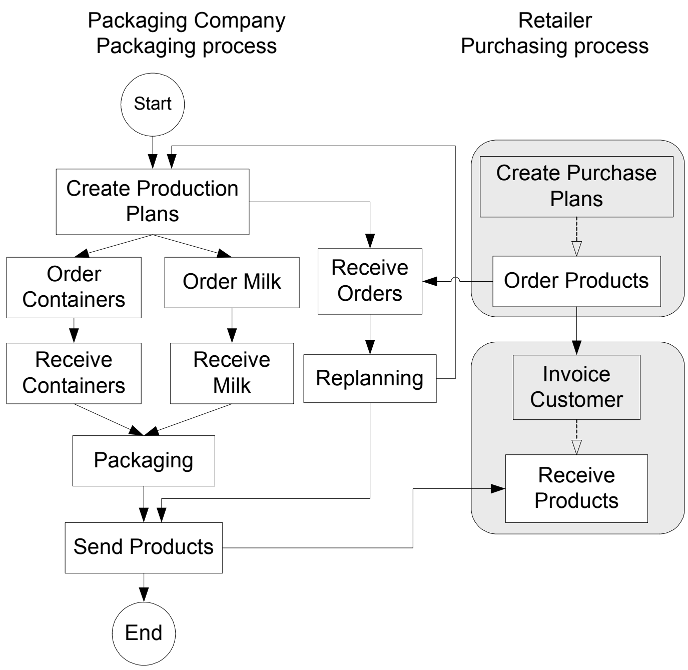

Figure 6.9 Relative workflow process from the packaging company’s view

As the packaging company requires paperboard containers and milk for packaging, it creates the collaboration shown in Figure 6.10, with a container manufacturer and a milk processing company. The milk processing company and the container manufacturer supply the ordered containers and milk to the packaging company. At this stage, we see that the original supply chain between the packaging company and the retailer grocer extends to include two more business processes from upstream organisations.

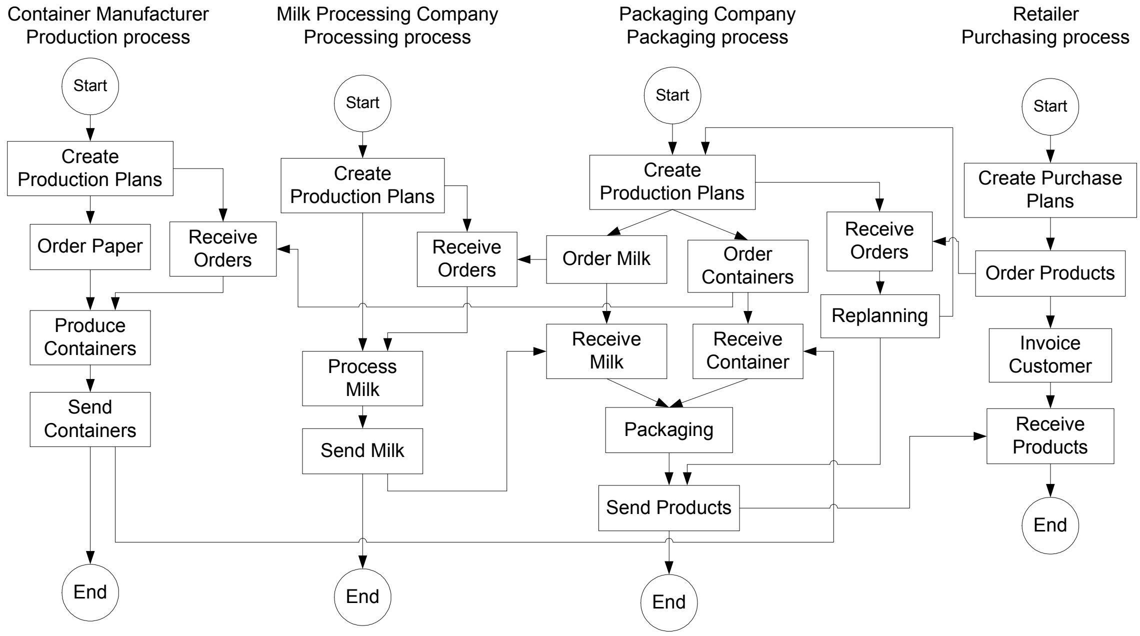

Figure 6.10 Collaborative business process between the four organisations

Here, we suppose that the container manufacturer and the milk processing company define the following visibility constraints for the packaging company in corresponding perceptions.

$P^{MilkProcCompany.processingProcess}_{PackagingCompany}.\mathcal{VC}$ = { ("Create Production Plans", invisible), ("Receive Orders", contactable), ("Process Milk", trackable), ("Send Milk", contactable) };

$P^{ContainerManufacturer.productionProcess}_{PackagingCompany}.\mathcal{VC}$ = { ("Create Production Plans", invisible), ("Order Paper", trackable), ("Receive Paper", trackable), ("Receive Orders", contactable), ("Produce Containers", trackable), ("Send Containers", contactable) };

According to these visibility constraints, the packaging company can include the business processes of the container manufacturer and the milk processing company to its relative workflow process. The relative workflow process from the packaging company's view may evolve to the one shown in Figure 6.11. Here, a clear rectangle of a perceivable workflow process denotes a contactable or trackable task.

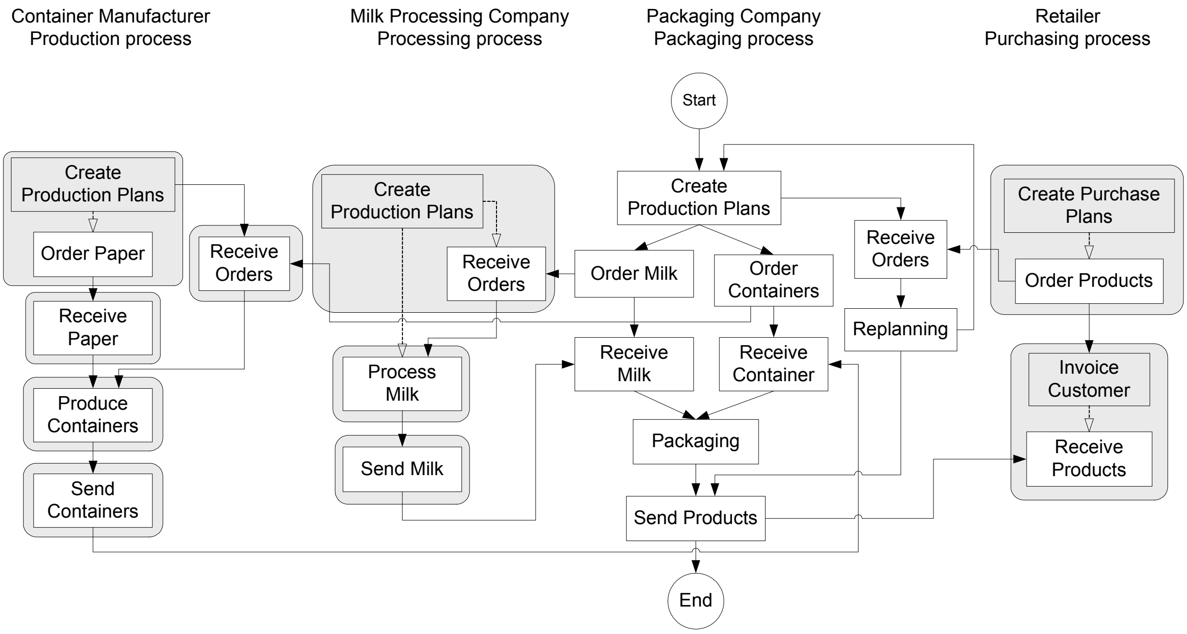

Figure 6.11 Relative workflow process from the packaging company's view, including four organisations

Also, the packaging company and the retailer may define the following visibility constraints for the milk processing company. As the retailer has no direct interactions with the milk processing company, some tasks are set trackable instead of contactable for the milk processing company. Therefore, the milk processing company may trace the terminal distribution of its products.

$P^{PackagingCompany.packagingProcess}_{MilkProcessingCompany}$. $\mathcal{VC}$ = { ("Create Production Plans", invisible), ("Order Containers", invisible), ("Order Milk", contactable), ("Receive Container", invisible), ("Receive Milk", contactable), ("Receive Orders", invisible), ("Replanning", invisible), ("Packaging", trackable), ("Send Products", trackable) };

$P^{Retailer.purchasingProcess}_{MilkProcessingCompany}$. $\mathcal{VC}$ = { ("Create Purchase Plans", invisible), ("Order Products", invisible), ("Invoice Customer", invisible), ("Receive Products", trackable) }.

Finally, the supply chain evolves again to cover the supplier of the container manufacturer, i.e., the paper mill. Figure 6.12 shows the supply chain comprising these five business processes from different organisations in a public view.

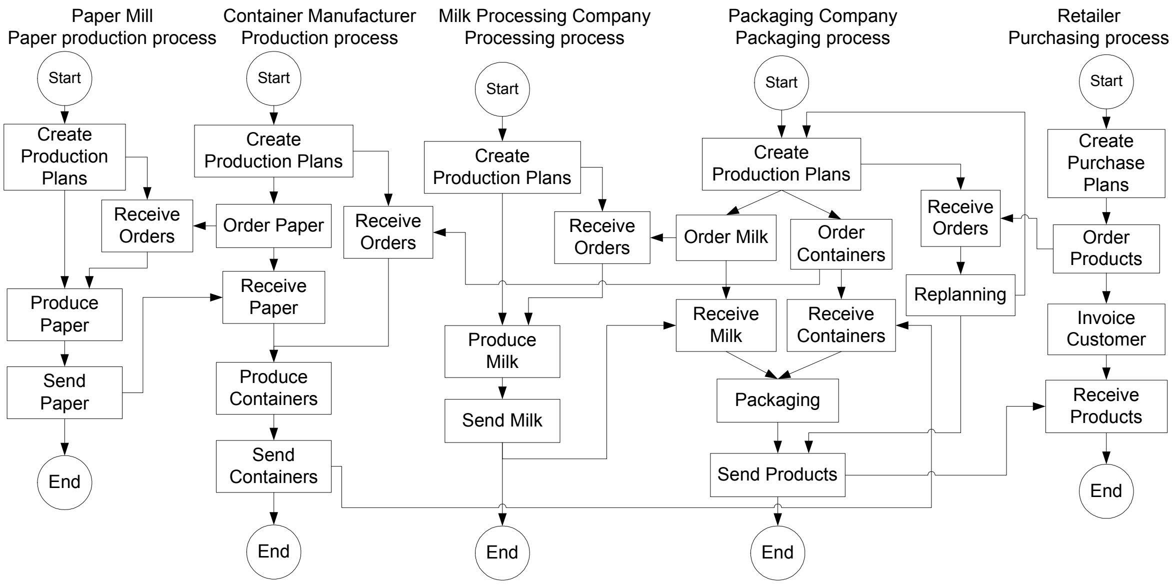

Figure 6.12 Collaborative business process between the total five organisations

Here, we suppose the paper mill defines the following visibility constraints for the packaging company and the container manufacturer in proper perceptions.

$P_{PackagingCompany}^{PaperMill.paperproductionProcess}.\mathcal{VC}$ = { ("Create Production Plans", invisible), ("Receive Orders", trackable), ("Produce Paper", trackable), ("Send Paper", trackable) }.

$P_{ContainerManufacturer}^{PaperMill.paperproductionProcess}.\mathcal{VC}$ = { ("Create Production Plans", invisible), ("Receive Orders", contactable), ("Produce Paper", trackable), ("Send Paper", contactable) };

For privacy protection, the container manufacturer and the packaging company may define the following visibility constraints in their perceptions for the paper mill.

$P_{PaperMill}^{ContainerManufacturer.productionProcess}.\mathcal{VC}$ = { ("Create Production Plans", invisible), ("Order Paper", contactable), ("Receive Orders", invisible), ("Receive Paper", contactable), ("Produce Containers", trackable), ("Send Containers", trackable) };

$P_{PaperMill}^{PackagingCompany.packagingProcess}.\mathcal{VC}$ = { ("Create Production Plans", invisible), ("Order Milk", invisible), ("Order Containers", invisible), ("Receive Milk", invisible), ("Receive Container", trackable), ("Replanning", invisible), ("Packaging", trackable), ("Send Products", trackable) }.

Because the packaging company has no contactable tasks with the paper mill, the packaging company needs not to include the paper mill's business process in its relative workflow process. Therefore, the packaging company's relative workflow process does not change during this supply chain extension. On the other side, the paper mill may create a relative workflow process as shown in Figure 6.13, according to the visibility constraints mentioned above.

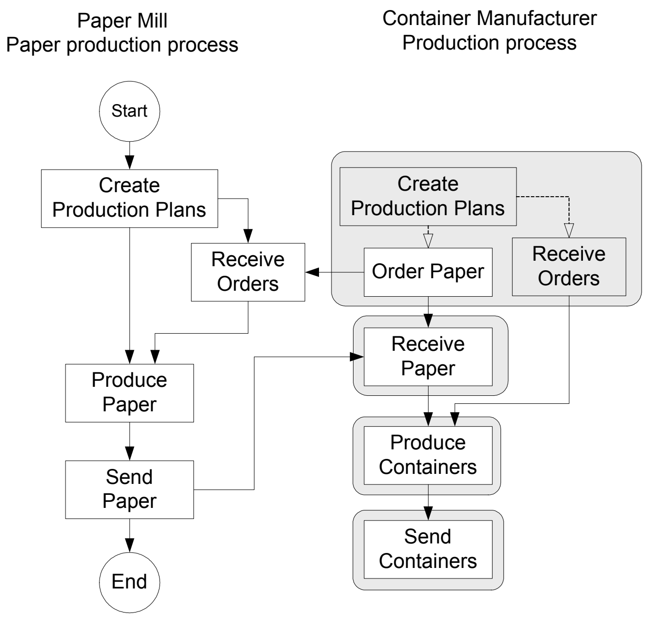

Figure 6.13 Relative workflow process from the paper mill's view

Because the paper mill only has contactable tasks with the container manufacturer, the corresponding relative workflow process merely includes the container manufacturer's business process.

6.2.4 Transient supply chain adaptation

In the aforementioned supply chain for dairy production, the packaging company may order cardboard containers from the importer in an ad hoc mode to cater for occasional demand bursts. This feature results in that the supply chain may change to cover the importer and change back from time to time. The partnerships between organisations of this supply chain are highly dynamic.

The collaboration between the importer and the packaging company is shown in Figure 6.14 in a public view.

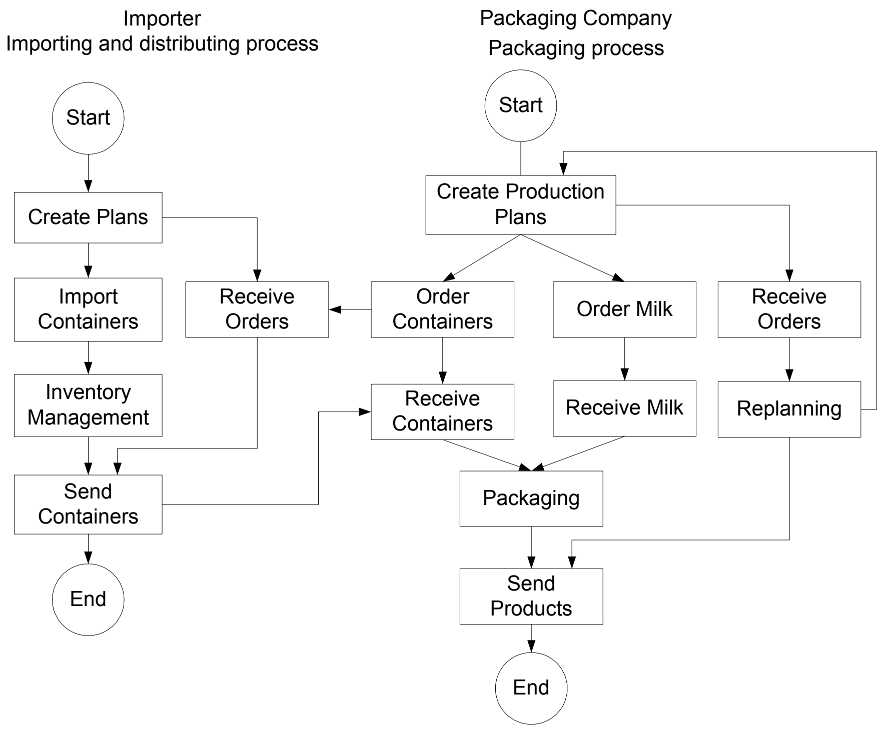

Figure 6.14 Collaboration between the importer and the packaging company

As a competitor to the container manufacturer, the importer may be unwilling to reveal its internal business processes to the container manufacturer or the paper mill, and vice versa. In the relative workflow context, this implies that these two organisations may set their business processes invisible to each other. This results in the following two null sets of visibility constraints in proper perceptions.

$$P_{Container}^{Importer.Import\&distProcess}.\mathcal{VC} = \varnothing;$$

$$P_{Importer}^{Container.productionProcess}.\mathcal{VC} = \varnothing;$$

These two null sets denote that the paper mill or the container manufacturer cannot perceive the importer's involvement. Therefore, the importer's business process does not appear in the container manufacturer's or the paper mill's relative workflow process. Thus, the paper mill's relative workflow process keeps the same as that shown in Figure 6.13.

Here, we suppose that the importer sets up the following visibility constraints in its perception for the packaging company.

$P_{PackagingCompany}^{Importer.Imp\&distProcess}.\mathcal{VC}$ = { ("Create Plans", invisible), ("Import Containers", invisible), ("Inventory Management", invisible), ("Receive Orders", contactable), ("Send Containers", contactable) };

According to these visibility constraints, the packaging company's relative workflow process may change to the one as shown in Figure 6.15, when the packaging company sources containers from both the importer and the container manufacturer.

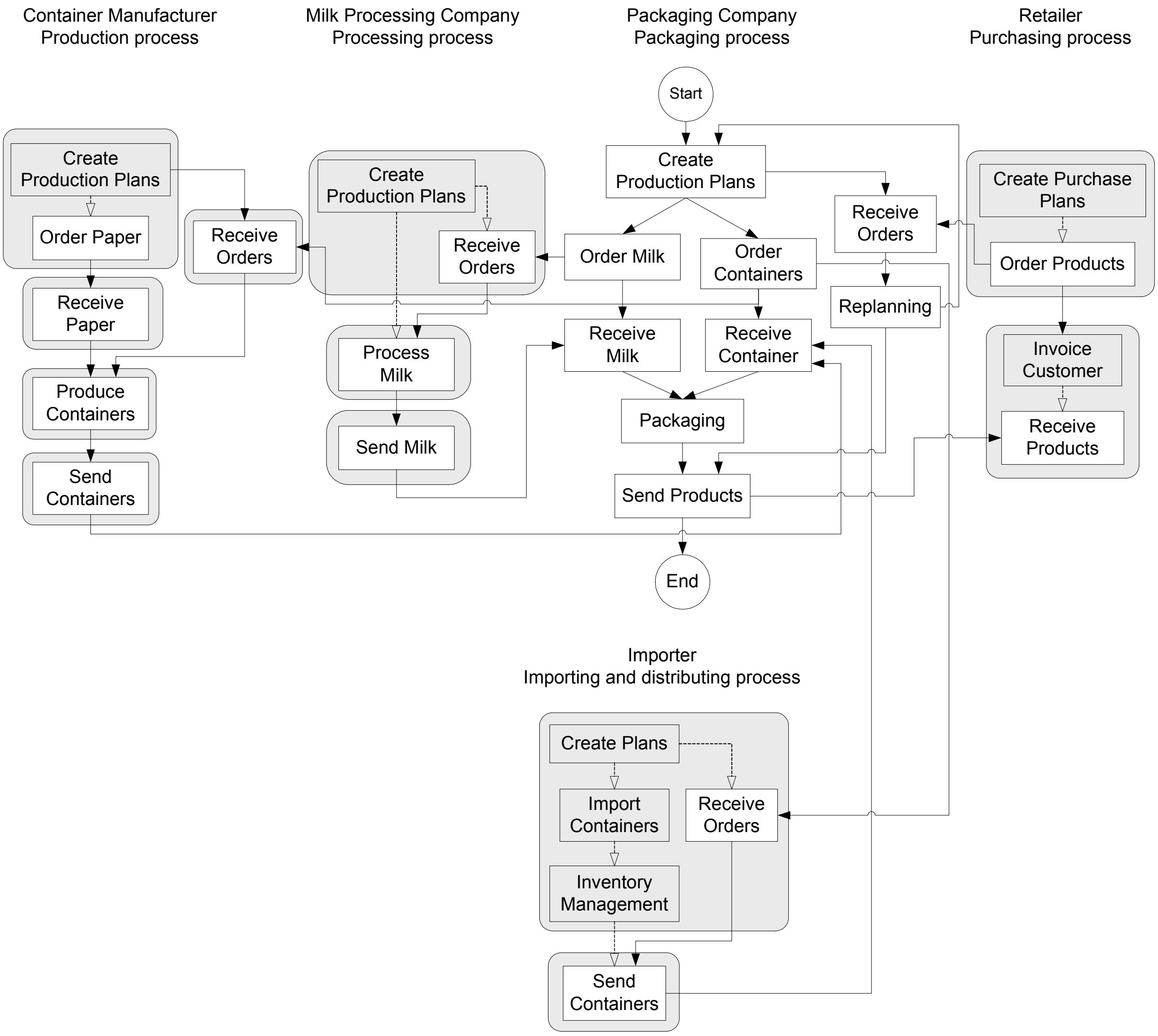

Figure 6.15 The relative workflow process from the view of the packaging company, including five organisations

Form above discussion, we see that a relative workflow process may dynamically change to fit the external trading environment. From the organisation-oriented perspective, such a change occurs individually not as a whole. Only the organisations involved in the change need to update their relative workflow processes, yet the other organisations do not need to change at all.

6.2.5 Supply chain collaboration tracking

In supply chain context, the tracking over collaboration is important for monitoring and management. In the aforementioned supply chain for dairy production, the packaging company may monitor the whole supply chain from the paper mill to the retailer, by tracking the execution status of the business processes of all involved organisations.

Therefore, the packaging company may know the production status of milk and containers, and the distribution of the shipped packaged milk. Yet, such tracking may go beyond the perception scope of a relative workflow process that is limited to the business processes of neighbouring organisations. To break through this limit, this section deploys the inter-organisational workflow tracking approach proposed in Chapter 5 to support supply chain collaboration tracking.

As discussed in Chapter 5, the proposed inter-organisational workflow tracking approach depends on the notion of tracking structure. The following part of this section mainly focuses the generation of the tracking structure.

When the packaging company links its business process and the retailer's business process together, the corresponding tracking structure for the packaging company changes from D_{PC}^{PC} to $\begin{pmatrix} D_{PC}^{PC} & B_{PC}^{PC\,|\,Retailer} \\ 0 & D_{PC}^{Retailer} \end{pmatrix}$, according to the extension operation discussed in Chapter 6.

Here, D_{PC}^{PC} stands for the self-adjacency matrix for the involved local workflow process of the packaging company; $D_{PC}^{Retailer}$ stands for the self-adjacency matrix for the involved perceivable workflow process of the retailer from the packaging company's view; $B_{PC}^{PC|Retailer}$ stands for the boundary adjacency matrix for the business processes of the two organisations.

When the packaging company includes the business processes from the milk processing company and the container manufacturer, the tracking structure grows to be $\begin{pmatrix} D_{PC}^{PC} & B_{PC}^{PC|Retailer} & B_{PC}^{PC|Milk} & B_{PC}^{PC|Container} \\ 0 & D_{PC}^{Retailer} & B_{PC}^{Retailer|Milk}=0 & B_{PC}^{Retailer|Container}=0 \\ 0 & 0 & D_{PC}^{Milk} & B_{PC}^{Milk|Container}=0 \\ 0 & 0 & 0 & D_{PC}^{Container} \end{pmatrix}$. Here, D_{PC}^{Milk} and $D_{PC}^{Container}$ stand for the self-adjacency matrices for the perceivable workflow processes of the two later joined organisations, respectively, from the packaging company's view; and a series of boundary adjacency matrices stand for inter-process links between collaborating business processes. As the retailer has no direct business interactions with the milk processing company or the container manufacturer, matrices $B_{PC}^{Retailer|Milk}$ and $B_{PC}^{Retailer|Container}$ only contain zeros in fact.

As tasks "Receive Orders", 'Produce Paper" and "Send Paper" of the paper mill's business process are set trackable to the packaging company, the packaging company may extend its tracking structure to include the paper mill's business process. Thus, the packaging company's tracking structure at this time evolves to the following matrix,

$$\begin{pmatrix} D_{PC}^{PC} & B_{PC}^{PC|Retailer} & B_{PC}^{PC|Milk} & B_{PC}^{PC|Container} & B_{PC}^{PC|Mill}=0 \\ 0 & D_{PC}^{Retailer} & B_{PC}^{Retailer|Milk}=0 & B_{PC}^{Retailer|Container}=0 & B_{PC}^{Retailer|Mill}=0 \\ 0 & 0 & D_{PC}^{Milk} & B_{PC}^{Milk|Container}=0 & B_{PC}^{Milk|Mill}=0 \\ 0 & 0 & 0 & D_{PC}^{Container} & B_{PC}^{Container|Mill} \\ 0 & 0 & 0 & 0 & D_{PC}^{Mill} \end{pmatrix}.$$

Moreover, the final tracking structure extends to cover the importer's business process as an alternative sourcing destination. Therefore, the final tracking structure evolves to the matrix below,

$$\begin{pmatrix} D_{PC}^{PC} & B_{PC}^{PC|Retailer} & B_{PC}^{PC|Milk} & B_{PC}^{PC|Container} & B_{PC}^{PC|Mill}=0 & B_{PC}^{PC|Importer} \\ 0 & D_{PC}^{Retailer} & B_{PC}^{Retailer|Milk}=0 & B_{PC}^{Retailer|Container}=0 & B_{PC}^{Retailer|Mill}=0 & B_{PC}^{Retailer|Importer}=0 \\ 0 & 0 & D_{PC}^{Milk} & B_{PC}^{Milk|Container}=0 & B_{PC}^{Milk|Mill}=0 & B_{PC}^{Milk|Importer}=0 \\ 0 & 0 & 0 & D_{PC}^{Container} & B_{PC}^{Container|Mill} & B_{PC}^{Container|Importer}=0 \\ 0 & 0 & 0 & 0 & D_{PC}^{Mill} & B_{PC}^{Mill|Importer}=0 \\ 0 & 0 & 0 & 0 & 0 & D_{PC}^{Importer} \end{pmatrix}$$

The full view of this matrix is given in Figure 6.16.

$$
\begin{pmatrix}
\begin{pmatrix}
0 & r & 0 & r & 0 & r & 0 & 0 & r \\
 & 0 & r & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & 0 & 0 & 0 & r & 0 & 0 & 0 \\
 & & & 0 & r & 0 & 0 & 0 & 0 \\
 & & & & 0 & 0 & r & 0 & 0 \\
 & & & & & 0 & 0 & 0 & r \\
 & & & & & & 0 & r & 0 \\
 & & & & & & & 0 & r \\
 & & & & & & & & 0
\end{pmatrix} &
\begin{pmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & l & 0 & 0 \\
0 & 0 & 0 & l \\
0 & 0 & 0 & 0
\end{pmatrix} &
\begin{pmatrix}
0 & 0 & 0 & 0 \\
0 & l & 0 & 0 \\
0 & 0 & 0 & l \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix} &
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & l & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & l \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} &
0 &
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & l & 0 \\
0 & 0 & 0 & 0 & l \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{pmatrix} \\
 &
\begin{pmatrix}
0 & 0 & 0 & 0 \\
 & 0 & 0 & r \\
 & & 0 & 0 \\
 & & & 0
\end{pmatrix} & 0 & 0 & 0 & 0 \\
 & &
\begin{pmatrix}
0 & 0 & 0 & 0 \\
 & 0 & r & 0 \\
 & & 0 & r \\
 & & & 0
\end{pmatrix} & 0 & 0 & 0 \\
 & & &
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & r & r & 0 & 0 \\
 & & 0 & 0 & r & 0 \\
 & & & 0 & r & 0 \\
 & & & & 0 & r \\
 & & & & & 0
\end{pmatrix} &
\begin{pmatrix}
0 & 0 & 0 & 0 \\
0 & l & 0 & 0 \\
0 & 0 & 0 & l \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix} & 0 \\
 & & & &
\begin{pmatrix}
0 & 0 & 0 & 0 \\
 & 0 & r & 0 \\
 & & 0 & r \\
 & & & 0
\end{pmatrix} & 0 \\
 & & & & &
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 \\
 & & 0 & 0 & 0 \\
 & & & 0 & r \\
 & & & & 0
\end{pmatrix}
\end{pmatrix}
$$

Figure 6.16 The full view of the composite matrix

Each row or column of this matrix stands for a workflow task. Table 6.1 lists the mapping between tasks and the rows or columns. For example, the second column of the sub matrix $B_{PC}^{PC|Retailer}$ represents the second task of the packaging company's business process, i.e., "Order Milk" task; The third row of sub matrix $D_{PC}^{Retailer}$ represents the third task of the retailer's business process, i.e., "Invoice Customer" task.

Table 6.1 Legend

No. of row / column	Workflow task
Paper Mill	
1	Create Production Plans
2	Receive Orders
3	Produce Paper
4	Send Paper
Milk Processing Company	
1	Create Production Plans
2	Receive Orders
3	Produce Milk
4	Send Milk
Container Manufacturer	
1	Create Production Plans
2	Order Paper
3	Receive Paper
4	Receive Orders
5	Produce Container
6	Send Container
Packaging Company	
1	Create Production Plans
2	Order Milk
3	Receive Milk
4	Order Containers
5	Receive Containers
6	Receive Orders
7	Packaging
8	Send products
9	Replanning
Retailer	
1	Create Purchase Plans
2	Order Products
3	Invoice Customer
4	Receive Products
Importer	
1	Create Plans
2	Import Containers
3	Inventory Management
4	Receive Orders
5	Send Containers

The corresponding graphical tracking structure is given in Figure 6.17.

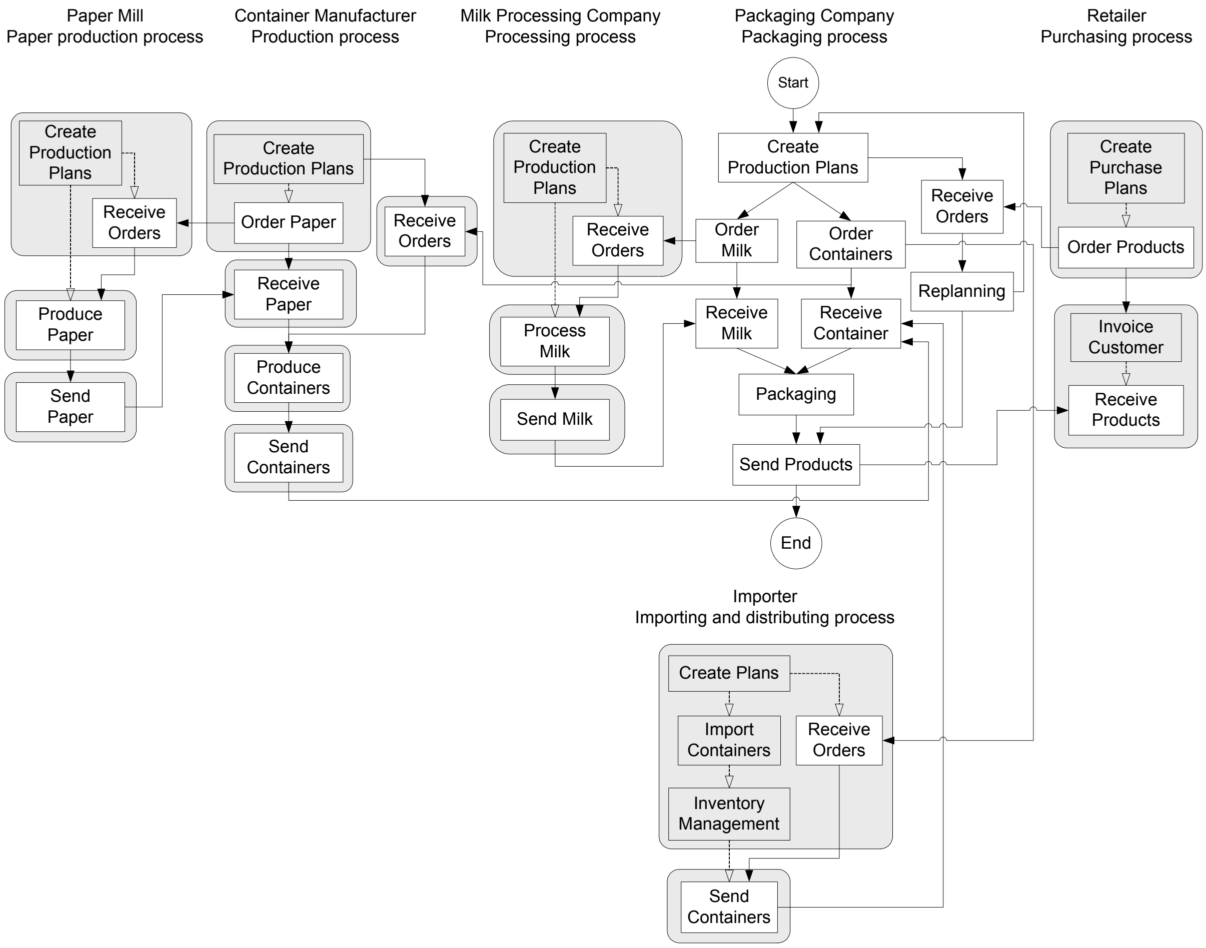

Figure 6.17 Tracking structure from the packaging company

In Figure 6.17, we can see that the packaging company can perceive beyond neighbouring organisations, i.e., the milk processing company or the retailer, to the non-neighbouring organisations, i.e., the container manufacturer and the paper mill. This perception is obtainable via the trackable tasks of related business processes, which can transform the perception to other organisation. With such perception, the packaging company can monitor the execution status of the whole supply chain and the trail of material flows.

From the perspective of the paper mill, some tasks of the business processes belong to the packaging company and the retailer are set trackable. Therefore, the paper mill may extend its tracking structure step by step, as illustrated by Figure 6.18.

$$D_{Mill}^{Mill} \rightarrow \begin{pmatrix} D_{Mill}^{Mill} & B_{Mill}^{Mill|Container} \\ 0 & D_{Mill}^{Container} \end{pmatrix} \rightarrow \begin{pmatrix} D_{Mill}^{Mill} & B_{Mill}^{Mill|Container} & B_{Mill}^{Mill|PC}=0 \\ 0 & D_{Mill}^{Container} & B_{Mill}^{Container|PC} \\ 0 & 0 & D_{Mill}^{PC} \end{pmatrix}$$

$$\rightarrow \begin{pmatrix} D_{Mill}^{Mill} & B_{Mill}^{Mill|Container} & B_{Mill}^{Mill|PC}=0 & B_{Mill}^{Mill|Retailer}=0 \\ 0 & D_{Mill}^{Container} & B_{Mill}^{Container|PC} & B_{Mill}^{Container|Retailer}=0 \\ 0 & 0 & D_{Mill}^{PC} & B_{Mill}^{PC|Retailer} \\ 0 & 0 & 0 & D_{Mill}^{Retailer} \end{pmatrix}$$

Figure 6.18 Generation of the paper mill's tracking structure

The full view of this composite matrix is given in Figure 6.19.

$$\begin{pmatrix}
\begin{pmatrix} 0 & r & r & 0 \\ & 0 & r & 0 \\ & & 0 & r \\ & & & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & l & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & l & 0 & 0 & 0 \end{pmatrix} & 0 & 0 \\
 & \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ & 0 & r & 0 & r & 0 \\ & & 0 & 0 & r & 0 \\ & & & 0 & 0 & 0 \\ & & & & 0 & r \\ & & & & & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & l & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} & 0 \\
 & & \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 0 & 0 & r & r & 0 \\ & & & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & 0 & 0 & 0 & 0 & 0 \\ & & & & & 0 & 0 & r & 0 \\ & & & & & & 0 & 0 & 0 \\ & & & & & & & 0 & 0 \\ & & & & & & & & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & l \\ 0 & 0 & 0 & 0 \end{pmatrix} \\
 & & & 0
\end{pmatrix}$$

Figure 6.19 The full view of the composite matrix for the paper mill

The mapping between tasks and the columns or rows is also listed in Table 6.1. The corresponding graphical tracking structure is given in Figure 6.20.

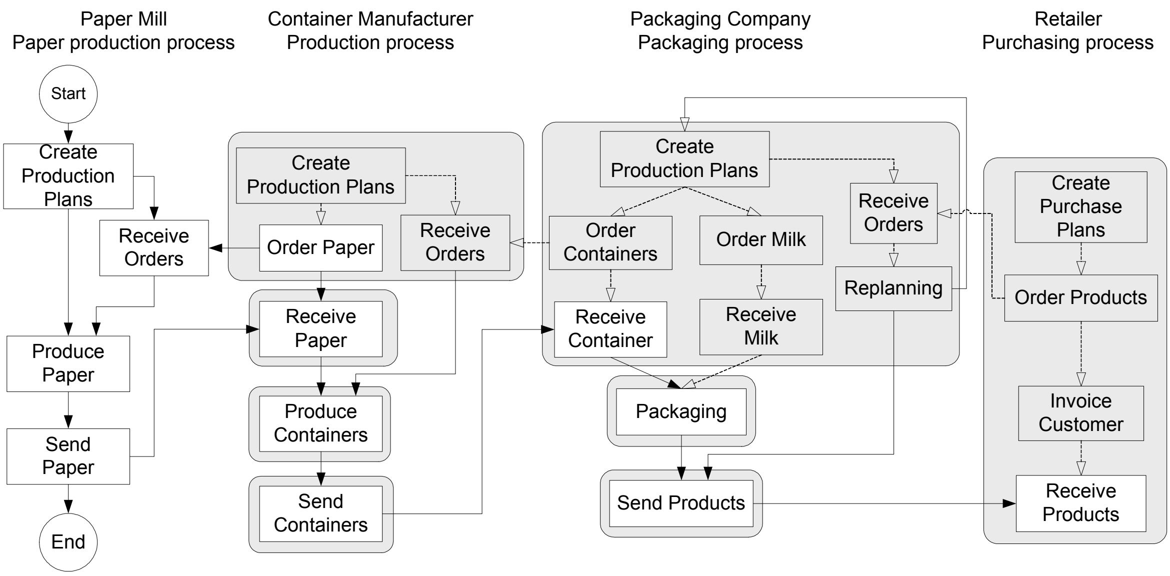

Figure 6.20 Tracking structure from the paper mill's view

The tracking structure shown in Figure 6.20 enables the paper mill's perception upon the business processes of its non-neighbouring organisations, i.e., the packaging company and the retailer. With this tracking structure, the paper mill can trace the flow and use of its paper products. For example, the paper mill can track which batch of containers are made of its papers, which milk products are packaged with the batch of containers, etc. The manageability of tracking and tracing is very important for quality control, after sale service and customer relationship management etc.

6.2.6 Summary

The dairy production collaboration scenario reflects the scalable and flexible features of a transient supply chain. The demand-oriented strategy drives the transient supply chain to transform dynamically. Correspondingly, we conduct an analysis upon the requirements from a transient supply chain and the supports from our organisation-oriented view methodology. Inter-organisational workflow tracking are realised with the approach proposed in Chapter 5. A generated tracking structure empowers an organisation to monitor the business processes of both neighbouring organisations and non-neighbouring organisations. Therefore, an organisation is aware of the flow and use of its products, and the sourcing of its supply.

Chapter 7

Facilitating Relative Workflows in Web Service Environment

To demonstrate and prove the feasibility of the organisation-oriented view methodology presented in this thesis, a system is designed on Web service platform. Web service paradigm recently emerged as a response to the shift in the IT landscape from isolated, tightly coupled systems to a highly distributed, loosely coupled environment. Ideally, Web service technology fits the requirements for business collaboration, and we therefore develop this system on Web service platform. This decision is also resulted from the supplementarity between Web services and our organisation-oriented view methodology. From one side, Web services provide infrastructural supports including communication protocols, universal modelling language and standard invocation interfaces between heterogeneous entities in a loosely-coupled collaboration environment; from the other side, the organisation-oriented view methodology effectively harmonises the functioning of Web services, especially in aspects of privacy and autonomy.

The two major parts of this work are listed below:

- A set of mapping rules for converting relative workflow processes to executable business processes in the format of Business Process Execution Language for Web services (WS-BPEL, formerly BPEL4WS), with an extension to support relative workflows.

- A Web service based system architecture for relative workflow management, including local and inter-organisational modelling, execution, monitoring etc.

Sections 7.1 and 7.2 are to intensively discuss these two parts, respectively.

7.1 Incorporating Relative Workflows to WS-BPEL

In the current Web service protocol stack, Web Service Description Language (WSDL) [94] is used to specify the static interfaces of a Web service. An interface specification includes message types, port types, partner link types, etc. WS-BPEL [9] is used to define the orchestration and choreography of Web services towards specific business objectives. To seamlessly incorporate our organisation-oriented view methodology to the Web service technology, a mapping between relative workflow components and WS-BPEL/WSDL elements is provided and presented in this section.

7.1.1 Mapping rules

WS-BPEL specification applies event-driven mechanism to mediate the interactions between participating organisations. In addition, WS-BPEL architects a collaborative business process from the perspective of a pivot organisation. Therefore, a BPEL business process only describes the pivot organisation's interaction behaviours in the collaboration. Technically, a BPEL business process describes a business process in terms of invocation parameters, variables, messaging behaviours, exception handling behaviours and flow controlling descriptions etc. Corresponding sets of elements are defined in BPEL specification to represent these things [95]. To support business collaborations, <*PartnerLink*> elements are used to represent the messages passing between partner organisations. Each invocation in a BPEL business process denotes an invocation to a Web service via a service operation port. The service operation port is described in WSDL language, which formalises the static service interfaces and message details. The operations and messages are described abstractly, and then bound to a concrete network protocol, currently SOAP 1.1, and message format.

In the relative workflow model discussed in Chapter 3, a complete relative workflow process contains both local workflow processes and perceivable workflow processes, as well as inter process links including the links between local workflow processes, the links between perceivable workflow processes, and the links between these two kinds of workflow processes. For each single workflow process, it contains tasks, intra-process links, and messages. A perceivable workflow process may also have blind tasks resulted from the visibility control mechanism.

7.1.1.1 Incorporating local workflow processes

Regarding the definition of relative workflow processes, the local workflow processes embedded in a relative workflow process must belong to the same organisation, for example g_1. Therefore, there is no visibility restriction between these local workflow processes. In another word, all details of these local workflow processes and the interactions between them are available to g_1. Thus, we can treat these local workflow processes as a whole, say a business process network representing the interaction behaviours between the involved local Web services. With structured activity elements in WS-BPEL, such as <*sequence*>, <*switch*>, <*while*>, <*flow*>, <*pick*> etc., we can describe the structure of a series of networked local workflow processes.

As for the messages passing between these local workflow processes, we keep the corresponding BPEL partner links to indicate the functional invocations, and also use proper structured activity elements in BPEL to explicitly link the processes together. In synchronised messaging cases, <*pick*> and related <*onMessage*> elements may be added to represent instant responses on the event of partner link invocations.

In contrary to conventional business processes, such a business process network may have more than one starting entries, because each local workflow processes may start and be executed independently. To support the multiple stating entries, we can use standard WS-BPEL correlation elements, viz. <*correlationSets*> / <*correlations*> / <*correlation*>, to coordinate the concurrent starting activities.

7.1.1.2 Incorporating perceivable workflow processes

Different from local workflow processes, the perceivable workflow processes defined in a relative workflow process of organisation g_1 are subject to visibility constraints. Moreover, the execution of these perceivable workflow processes is in the hands of g_1's partner organisations, rather than g_1. The inclusion of these perceivable workflow processes is to empower g_1's perception over the collaboration scenario. Therefore, the perceivable workflow processes act as a non-executable part of a WS-BPEL business process.

To differentiate these perceivable workflow processes from local workflow processes, we create a couple of new elements, viz., <*perceivableProcesses*> and <*perceivableProcess*> to indicate the perceivable workflow processes of a relative workflow process. Now, the root element <*process*> of original WS-BPEL is used to represent the composite process constructed by the local workflow processes, while each <*perceivableProcess*> element of the <*perceivableProcesses*> element nested in the <*process*> element, is used to describe an individual perceivable workflow process.

7.1.1.3 Incorporating inter process links

In WS-BPEL specification, the services with which a business process interacts are modelled as partner links, and each partner link is characterised by a <*partnerLinkType*> element. Thus, we map the message links defined in relative workflow model to WS-BPEL <*partnerLinkType*> elements. These <*partnerLinkType*> elements specify the roles that the host organisation and the partner organisation act as during the interaction, using the <*role*> sub element, and specify the service operation ports using the <*portType*> sub element.

Current WS-BPEL partner links do not recognise whether a message link go within or across the organisational boundary. This causes troubles in navigating the execution of a collaborative business process between the workflow engines of participating organisations. To solve this, we introduce a new attribute "*invokeType*" for element <*partnerLinkType*> with a value "*local*" or "*foreign*" denoting whether it is an internal or external message link. According to the given "*invokeType*" and messaging directions, a workflow engine can readily recognise the transfer between the executable part and non-executable part of an extended WS-BPEL business process. Thus, the workflow engine may hand over the execution of a collaborative business process to the workflow engine of the partner organisation, or simultaneously notify the partner organisation's workflow engine to handle the corresponding request. Thereby, the orchestration between workflow engines is achieved.

In Chapter 4, cardinality parameters are proposed to describe the instance correspondence between collaborative business processes at process level. Correspondingly, we add a new attribute "*cardinality*", to the standard <*correlation*> element. Attribute "*cardinality*" can be assigned to a value of either "*single*" or "*multiple*", according to the practical unidirectional cardinality. In addition, the direction can be denoted by the original attribute "*pattern*" of element <*correlation*>.

7.1.1.4 Mapping table

In summary, we list all mapping rules in Table 7.1, from major relative workflow components to the elements defined in WSDL or WS-BPEL. With these rules, a graphical relative workflow process can be mapped to a documental WS-BPEL business process, and therefore can be applied to Web service environment.

Table 7.1 Mapping Rules

Relative Workflow Components	Web Service / BPEL Process Elements	Example
$\bigcup_i lp^i$, all local workflow processes defined in a relative workflow process	business process, in WS-BPEL	<process> <partnerLinks>...</partnerLinks> <variables>...</variables> <correlationSets>... </correlationSets> ... </process>
a perceivable workflow process	perceivable process, in WS-BPEL	<process> <perceivableProcesses> <perceivableProcess... /> ... </perceivableProcess> </process>
$t\in\mathcal{T}$, a task	Web service operation defined in a WSDL portType. The operation will be invoked through a partnerLink defined in WS-BPEL.	<operation name =... > ... </operation>
$r\in\mathcal{R}\text{-}\mathcal{L}$, a connection between tasks	elements for structured activities, in WS-BPEL <link>, <sequence>, <switch>, <while>, <pick>, <flow>	<links> <link name="buyToSettle"/>... </links> <receive name="getBuyerInfo"> <source linkName="buyToSettle"/> </receive>

$l\in\mathcal{L}_{inter}$, an inter-organisational messaging link	partner link, in WS-BPEL	<partnerLink name =...
		partnerLinkType=...>
		    myRole = ...
		partnerRole=...
		    invokeType = “foreign” >
		</partnerLink>

$l\in\mathcal{L}_{intra}$, an intra-organisational messaging link	partner link and event response elements in WS-BPEL	<partnerLink name =...
		partnerLinkType=...
		    invokeType = “local” />
		  ...
		  <pick>
		  <onMessage partnerLink =...>
		  ...
		  </onMessage>
		</pick>

$m\in\mathcal{M}$, a message	message type, in WSDL	<message name = ...> ... </message>
md=(m, in/out) $\in\mathcal{MD}$, a message description	primitive messaging activities, in WS-BPEL <invoke> / <receive> in asynchronised mode; <invoke> / <wait> in synchronised mode.	<invoke name="sendItem" operation="shipItem" partnerLink="shipper"/> <receive name="receiveItem" operation="confirmDelivery" partnerLink="shipper"/>

$cp\in\{[:1], [:n]\}$, cardinality parameter	“cardinality” attribute of the extended BPEL correlation element	<correlation set="Cor_1"
		    initiate="yes"
		    pattern="out"
		    cardinality="single"/>

a blind task	an empty activity, in WS-BPEL	<empty/>

7.1.2 Process mapping demonstration

7.1.2.1 Example business process

In this section, we present an example business process to demonstrate how we convert a relative workflow process to an extended WS-BPEL business process in practice. In the collaboration scenario shown in Figure 7.1, a retailer creates a relative workflow process containing two local workflow processes for product ordering and customer relationship management (CRM), and one perceivable workflow process of a manufacturer for production. The retailer's product ordering process collects orders from customers, and then orders products from the manufacturer. At the same time, the CRM process records all customer related behaviours. The numbers in square brackets denote the unidirectional cardinality between collaborating business processes.

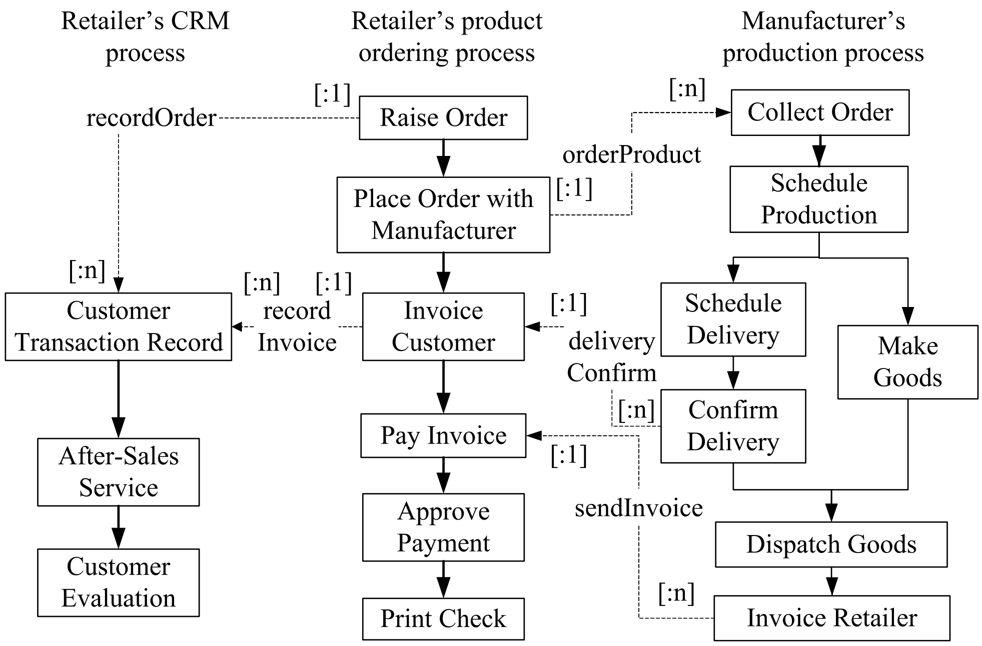

Figure 7.1 Collaborative business process example

In this example, we suppose that the manufacturer applies the following visibility constraints for the retailer. Thus, from the retailer's perspective, a relative workflow process as shown in Figure 7.2 may be created accordingly.

$p_{Retailer}^{Manufacturer.production\Pr ocess}.\mathcal{VC}$ = { (‘Collect Orders’, Contactable), (‘Schedule Production’, Invisible), (‘Schedule Delivery’, Invisible), (‘Confirm Delivery’, Contactable), (‘Make Goods’, Invisible), (‘Dispatch Goods’, Invisible), (‘Invoice Retailer’, Contactable) }

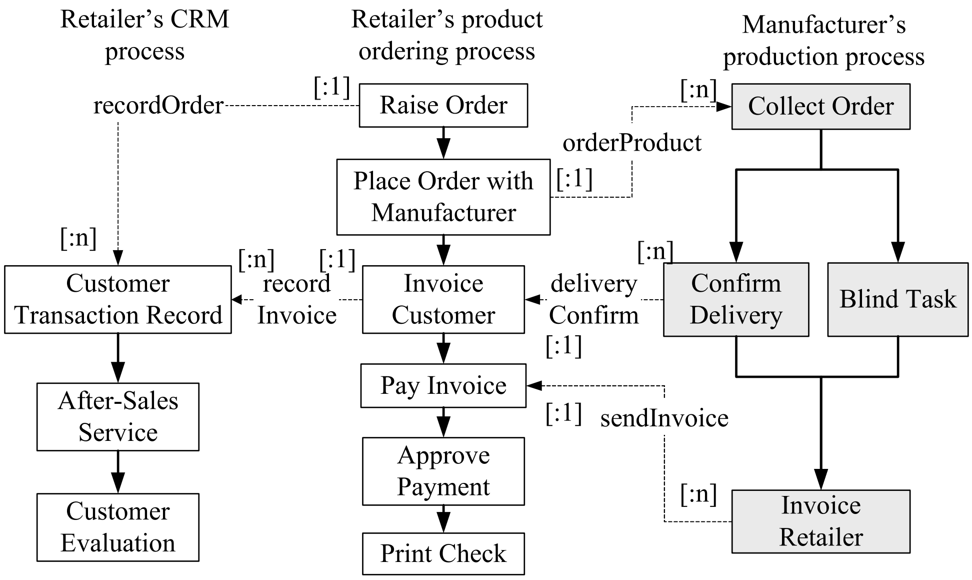

Figure 7.2 Relative workflow process example

As the invisible “Make Goods” task exists in the parallel branch of the manufacturer’s production process, this task is replaced with a blind task. This replacement is used to keep the original branching structure, as mentioned in Algorithm 3.2 of Chapter 3.

7.1.2.2 Example WS-BPEL business process

According to the mapping rules listed in Table 7.1, we convert the graphical relative workflow process shown in Figure 7.2 to a WS-BPEL business process. The corresponding WSDL document and WS-BPEL document for this business process are Figure 7.3 and Figure 7.4, respectively, with comments in the right column.

```
...
<definitions>
<message name=”POMessage”>
  <part name=”customerInfo”
        type=”sns:customerInfo”/>
  <part name=”puchaseOrder”
        type=”sns:purchaseOrder”/>
</message>
...
<portType name=”customerOrderPT”>
  <operation name=”sendOrder”>
    <input message=”pos:POMessage”/>
  </operation>
...
</portType>
...
<plnk:partnerLinkType
       name=”customerOrderLT”>
  <plnk:role name=”Retailer”>
  <plnk:portType name=”customerOrderPT”/>
  </plnk:role>
</plnk:partnerLinkType>
...
</definitions>
```

Defined messages.

Defined port types.

Defined partner link types.

Figure 7.3 WSDL document

```
<process name="RWF-CRM-POrdering-Production">
 <partnerLinks>
   <partnerLink name=”customerOrder”
    partnerLinkType=”lns:customerOrderLT”
    myRole=”Retailer”
    />
   <partnerLink name=”recordOrder”
    partnerLinkType=”lns:recordOrderLT”
    myRole=‟Sender‟
    partnerRole=‟Recorder‟
    />
   <parnterLink name=‟orderProduct‟
    partnerLinkType=‟lns:orderProductLT‟
    myRole=‟Purchaser‟
    partnerRole=‟Supplier‟
    />
   ...
</partnerLinks>
 <variables>
   <variable name =”PO”
    messageType=”lns:POMessage”/>
   ...
</variables>
```

Related partnerLinks, variables and correlation sets.

```
<correlationSets>
  <correlationSet name=”cor_CRM_1”
     properties=”cor:InstanceID”/>
    ...
</correlationSets>
 <flow name=”compositeLocalWf”>
  <links>
    <link name=
    ”raiseOrder-to-placeOrder”/>
    <link name=
    ”placeOrder-to-invoiceCustomer”/>
    ...
  </links>

  <sequence>
```

The content of the composite local workflow process.

```
 <receive name=”raiseOrder”
   partnerLink="customerOrder"
   invokeType=”local”>
   </receive>
```

The customer’s order request starts the product ordering process.

```
    <flow>
    <invoke name=”customerTranRecordTask”
  partnerLink=”recordOrder”
   invokeType=”local”>
     <source linkName=”toCRMProcess”/>
       <correlations>
        <correlation set=”cor_CRM_1”
           initiate=”yes”
           pattern=”out”
           cardinality=”single”/>
  </correlations>
</invoke>
```

Invoke the “customer transaction record” task of the CRM process.

```
    <invoke name=”PlaceOrderWithManuTask”
     partnerLink=”orderProduct”
       invokeType=”foreign”/>
   </flow>
    ...
  </sequence>
```

Invoke the “place order with manufacturer” task of the product ordering process.

```
  <sequence name=”CRMProcess”>
   <pick createInstance=”yes”>
    <onMessage partnerLink=”customerOrder”
                 invokeType=”local”>
      <target linkName=”toCRMProcess”>
      ...
     </onMessage>
   </pick>
 </sequence>
</flow>
```

The content of the CRM process

```
<perceivableProcesses>
 <perceivableProcess name=”productionProcess”>
   <partnerLinks>
   ...
    </partnerLinks>
   <variables>
```

The content of the perceivable process, i.e. the production process from the manufacturer.

```
   ...
   </variables>
   <flow>
    <sequence>
     <receive partnerLink="orderProduct"
              invokeType="foreign"/>
    ...
    </sequence>
   </flow>
  </perceivableProcess>
</perceivableProcesses>
</process>
```

Figure 7.4 Extended BPEL business process document

As shown in Figure 7.4, the two local workflow processes, i.e. the product ordering process and the CRM management process, merge their behaviours including Web service invocations in the <*process*> element. In addition, the perceivable workflow process, i.e., the manufacturer's production process, lists its behaviours in the <*perceivableProcess*> element.

7.2 Business Process Management System Architecture

7.2.1 Architecture overview

AMR Research [96] has enumerated the following basic functionalities that are expected to be embodied in a business process management system:

1. Process Modelling
2. Collaborative Development
3. Process Documentation
4. Application Integration
5. Process Automation
6. B2B Collaboration
7. End-User Deployment

These aspects cover the lifecycle of business process management, from process modelling to process execution, from front end interface to back end integration. According to these aspects, we present a business process management system architecture as shown in Figure 7.5. This business process management system implements our organisation-oriented view methodology with the benefits from Web service's inherent advantages in distributed computing.

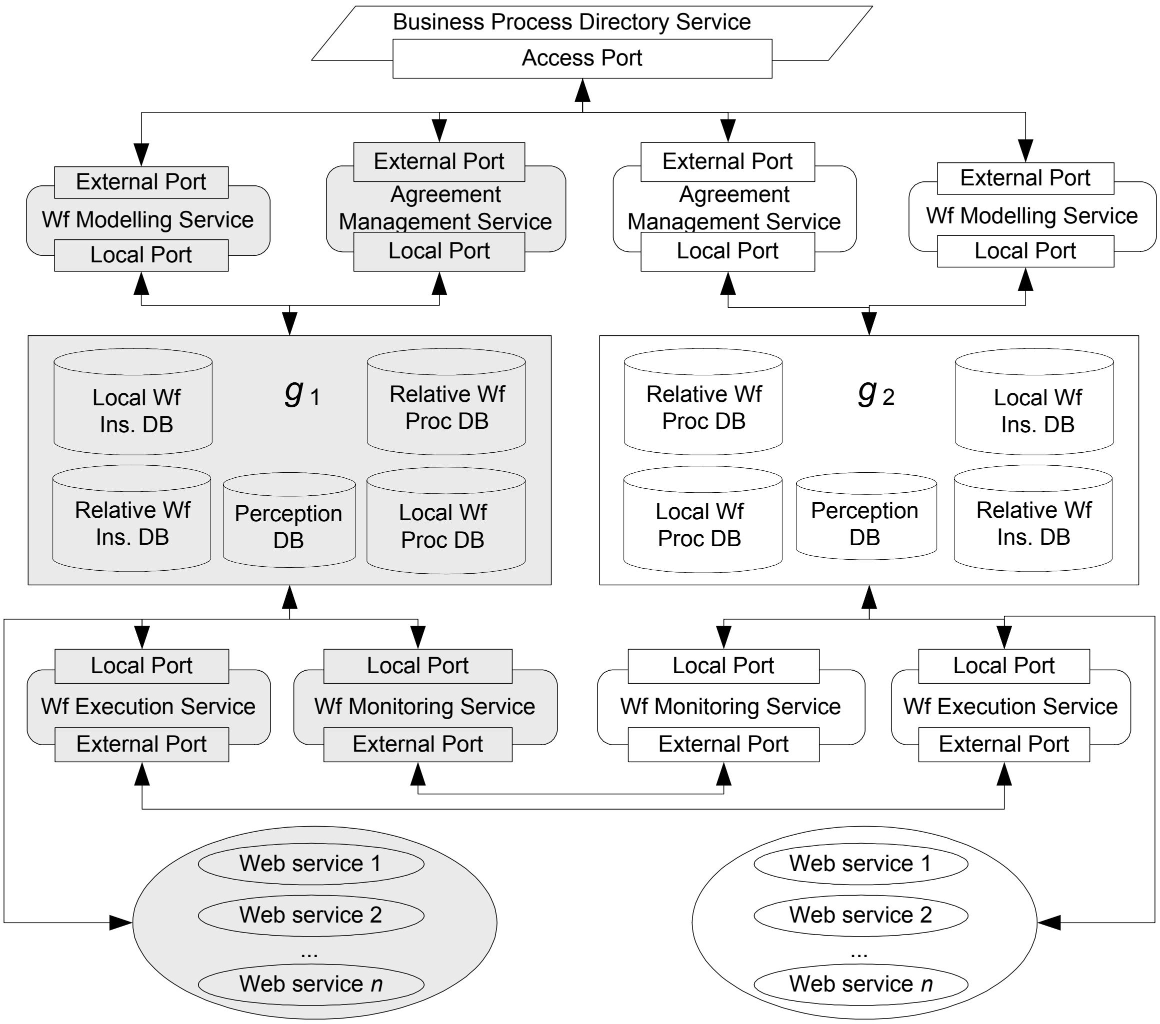

Figure 7.5 Architecture of the proposed business process management system

Figure 7.5 displays two collaborating business process management systems belonging to different organisations, respectively. An independent *Business Process Directory Service* stores all the business processes published by organisations in its *perceivable workflow process database*, to provide a common directory service for all organisations. Organisations can browse these published business processes and combine appropriate ones to create a collaborative business process.

The business process management system of each organisation consists of four administrative services, viz., the *Agreement Management Service*, the *Workflow Modelling Service*, the *Workflow Execution Service* and the *Workflow Monitor Service*. These four administrative services collectively support the functional aspects discussed before:

- *Agreement Management Service* handles the documentation for collaboration preparation and perceivable workflow process generation, and therefore supports Collaborative Development and the initialisation of B2B Collaboration;

- *Workflow Modelling Service* is responsible for the modelling of both local workflow processes and relative workflow processes, and therefore supports Process Modelling and Collaborative Development;

- *Workflow Execution Service* is responsible for the enactment of local workflow processes and relative workflow processes, and therefore supports Application Integration, Process Automation and End-User Deployment;

- *Workflow Monitoring Service* is responsible for collaboration execution monitoring, and therefore partially supports Process Automation and B2B Collaboration.

Each administrative service has both local operation ports and external operation ports, which are accessible to intra-organisational components and databases, and partner organisations' components and databases, respectively. In addition, several databases are deployed for Process Documentation. The Web services shown in big ovals work as independent functional components, which collectively provide business functions under the invocation of *Workflow Execution Services*, in harmony with the underlying collaborative business processes.

The whole lifecycle of a relative workflow process through these four administrative services goes as follows: First, the *Agreement Management Service* wraps a local workflow process into a series of perceivable workflow processes for different partner organisations; Thereafter, the *Workflow Modelling Service* generates relative workflow processes with the perceivable workflow processes from partner organisations; Finally, relative workflow processes will be executed by the *Workflow Execution Services* of the host organisation and the partner organisations. The monitoring over the execution will be handled by the *Workflow Monitor Services* of involved organisations.

7.2.2 Business Process Directory Service

The *Business Process Directory Service*, shown in Figure 7.6, provides a common directory service to all organisations. In Figure 7.6, the arrows in different styles represent different information flows. This service is designed to support relative workflow process modelling. On one side, this directory service can publish the perceivable workflow processes generated by the *Agreement Management Services* of participating organisations. On the other side, the *Workflow Modelling Services* of participating organisations can download the perceivable workflow processes and compose them into proper relative workflow processes.

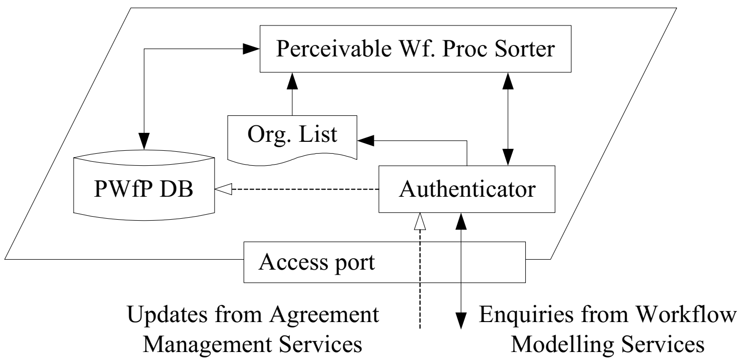

Figure 7.6 Business Process Directory Service

The major components of *Business Process Directory Service* are listed below:

- *Organisation list* records the information of all participating organisations. Technically, this list corresponds to a table, which contains the ID, description of major business, authentication information, etc. of organisations;

- *Authenticator* is in charge of identity authentication for all updates and requests from organisations, according to the *organisation list*;

- *Perceivable workflow process database* (*PWfP DB*) stores the published perceivable workflow processes from organisations' *Agreement Management Services*.

- *Perceivable workflow process sorter* is responsible for sorting out perceivable workflow processes from the *perceivable workflow process database*, according to organisations' requests. This selection process is subject to the visibility constraints defined in perceptions, and target organisations of perceptions. This guarantees that an organisation can only retrieve the perceivable workflow processes, which are available to this organisation.

7.2.3 Agreement Management Service

The function of *Agreement Management Service*, shown in Figure 7.7, is to define perceptions and generate perceivable workflow processes for partner organisations. This service maintains a list of partner organisations, and uses the *perception generator* to create proper perceptions on local workflow processes for these partner organisations. With these perceptions, the *perception locator* and the *perceivable workflow process generator* work together to wrap local workflow processes into perceivable workflow processes for collaboration preparation. Finally, these perceivable workflow processes will be published to the *Business Process Directory Service*.

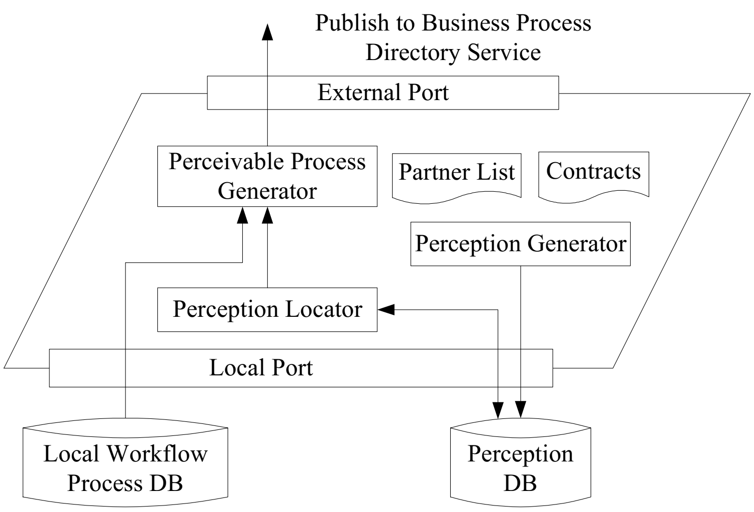

Figure 7.7 Agreement Management Service

The major components of the Agreement Management Service are listed below:

- *Perception generator* extracts the visibility constraints from the manually signed commercial contracts, and then incorporates these visibility constraints to the perceptions defined for different partner organisations. This procedure is detailed in Algorithm 3.1 of Chapter 3.

- *Partner list* is to record the names of all potential and currently involved partner organisations and their authority levels. Technically, the *partner list* corresponds to a table, which contains the information of organisation names, authority levels etc.

- *Perception database* stores all edited perceptions. This database contains several tables to store the information of Perception ID, related organisation, authority level, and visibility constraints (which will be defined using the graphical WS-BPEL modeller).

- *Perception locator* is to seek the *perception database* to find the proper perception defined for a specific organisation. Technically, perception locator is realised as a joint query for the partner list table and the tables in the *perception database*.

- *Perceivable workflow process generator* is responsible for wrapping a local workflow process into a perceivable workflow process. The generator uses the *perception locator* to find a proper perception for a local workflow process and wrap the local workflow process to a perceivable workflow process. Algorithm 3.2 of Chapter 3, details this procedure for composing a perceivable workflow process. The generated perceivable workflow processes will be sent to the *Business Process Directory Service* for publication.

7.2.4 Workflow Modelling Service

The *Workflow Modelling Service*, shown in Figure 7.8, is responsible for local workflow process modelling and relative workflow process assembling. This service uses the *perceivable workflow process locator* and the *local workflow process locator* to access and load perceivable workflow processes and local workflow processes from the *Business Process Directory Service* and the *local workflow process database*, respectively. The *local workflow process editor* and the *relative workflow process assembler* share a common graphical process definition tool to generate local workflow processes and relative workflow processes, respectively.

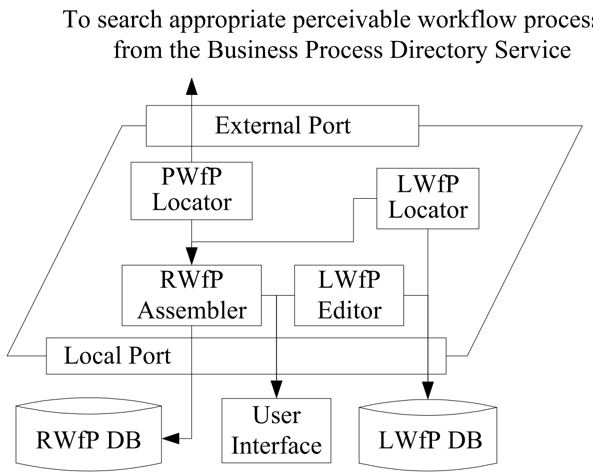

Figure 7.8 Workflow Modelling Service

The major components of the Workflow Modelling Service are listed below:

- *Local workflow process database* (*LWfP DB*) stores the defined local workflow processes. This component corresponds to a table containing the designed local workflow processes in the extended WS-BPEL format.

- *Relative workflow database* (*RWfP DB*) stores the defined relative workflow processes. This component corresponds to a table containing the designed relative workflow processes in format of WS-BPEL documents.

- *Local workflow process locator* (*LWfP Locator*) is to seek the local workflow database for a specific local workflow process. This locator is technically realised as a query to the *local workflow process database*.

- *Perceivable workflow process locator* (*PWfP Locator*) is to seek perceivable workflow processes published from the *business process directory service*. This locator is technically realised as a query to the directory service.

- *Local workflow process editor* (*LWfP Editor*) works as a process definition tool. With this tool, a user can compose a local business process by dragging and clicking within a visual modelling environment. Thereafter, the user may save the visual business process in form of a WS-BPEL document in the *local workflow process database*. Figure 7.9 shows the graphical interface of the *local workflow process editor*, where a business process is represented as grey rectangles and links. Messages are represented as black ovals.

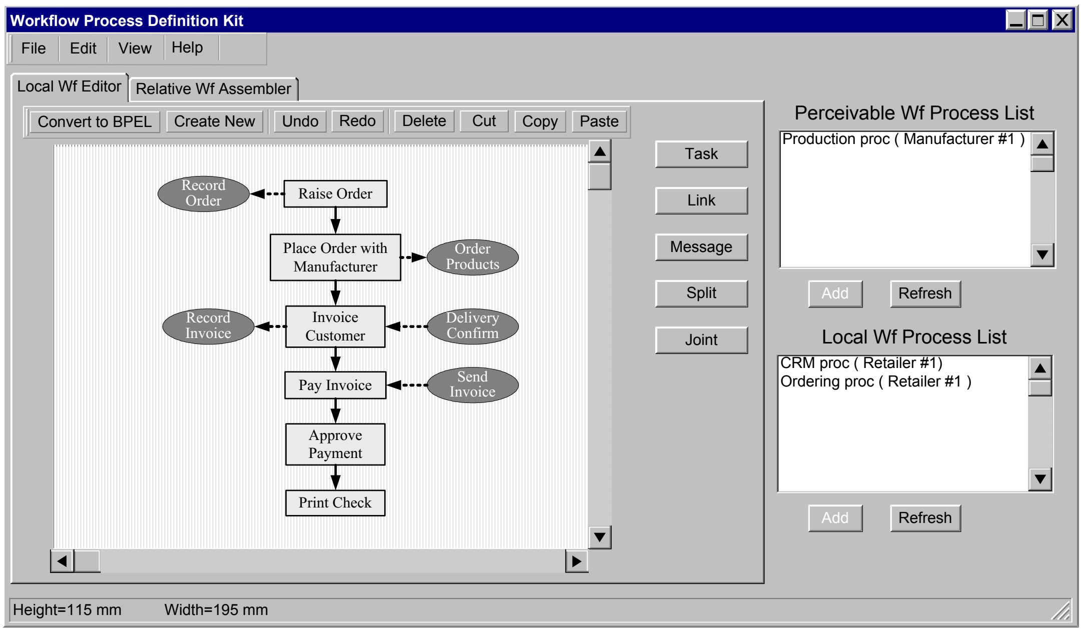

Figure 7.9 Local workflow process editor user interface

- *Relative workflow process assembler* also provides a graphical interface for users to visually assemble a relative workflow process. This relative workflow process can be created by combining proper local workflow processes in the local workflow process database, and proper perceivable workflow processes retrieved via the *perceivable workflow process locator*. Figure 7.10 shows the user interface of the *relative workflow process assembler*. The relative workflow assembler can automatically help the user to match compatible messages between selected workflow processes, according to Algorithm 3.3 of Chapter 3.

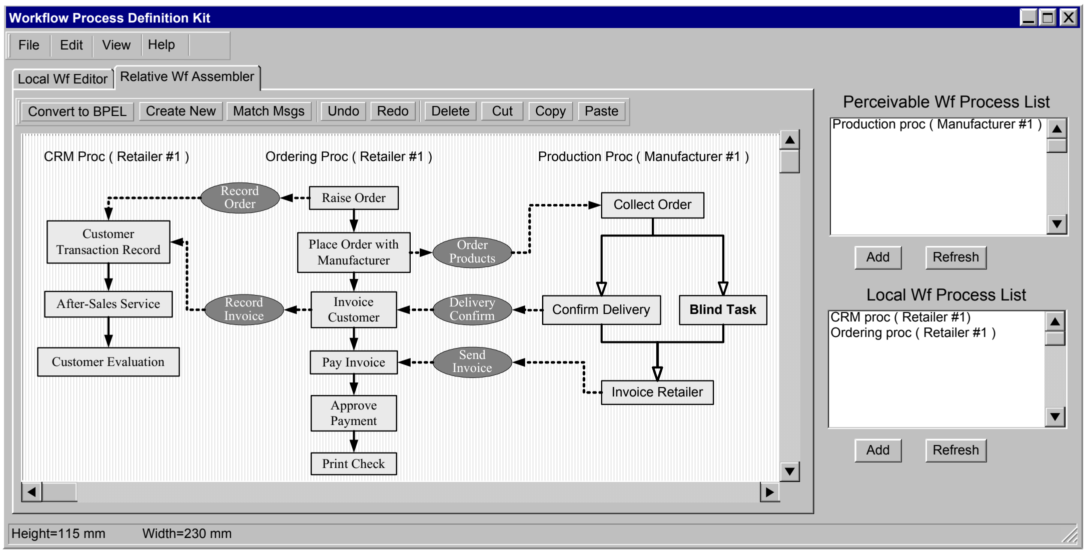

Figure 7.10 Relative workflow assembler user interface

7.2.5 Workflow Execution Service

7.2.5.1 Introduction

The function of *Workflow Execution Service*, shown in Figure 7.11, is to coordinate and enact the execution of business process instances. This service is responsible for the instance level management of local workflow processes and relative workflow processes. As for a local workflow process, this service works as a traditional workflow engine, responsible for creating new workflow instances, navigating workflow instance execution and controlling their interaction with workflow participants and applications. As for a relative workflow processes, the execution of a relative workflow instance depends on the cooperation between the Workflow Execution Services of participating organisations. This is because that a relative workflow process contains both a local part and a foreign part, and the execution of a relative workflow instance may involve interactions across organisational boundaries. Particularly, a *workflow coordinator* is assigned to coordinate the communication and interaction between participating organisations for collaborative business process execution.

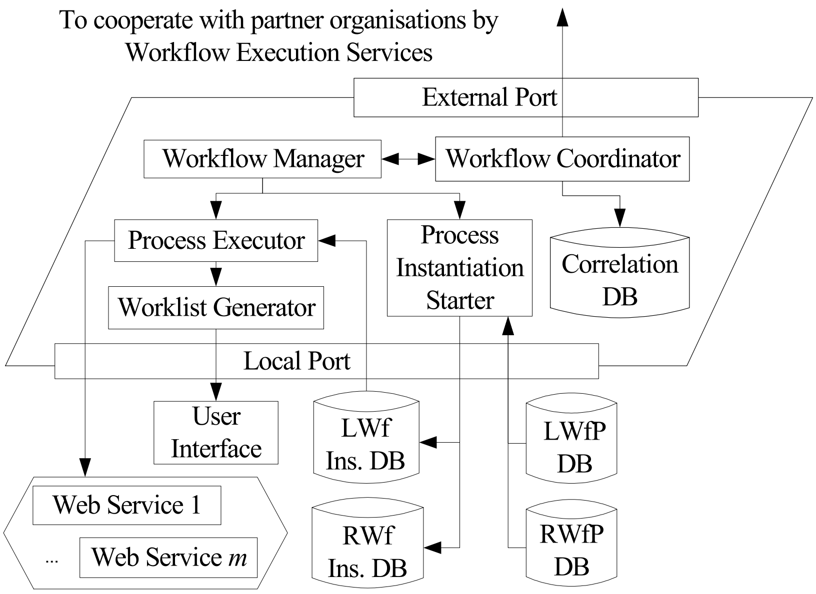

Figure 7.11 Workflow Execution Service

As Figure 7.11 shows, a *workflow manager* is in charge of the business process execution in general. A *process instantiation starter*, a *process executor* and a *worklist generator* together fulfil the functionalities for intra-organisational workflow process execution; while a *workflow coordinator* is particularly assigned for communicating with other *Workflow Execution Services*.

The major components of *Workflow Execution Service* are listed below:

- *Process instantiation starter* creates instances of specified workflow processes, and then sets them up with initial values. After initiated, these instances will be passed to the *process executor*.

- *Process executor* handles the execution of the local part of a relative workflow instance. Technically, it is in charge of the invocations to local Web services.

- *Local workflow instance database* (*LWf Ins. DB*) and *relative workflow instance database* (*RWf Ins. DB*) are used to store the instances of local workflow processes and relative workflow processes, respectively.

- *Workflow manager* is responsible for navigating the execution of workflow instances in general, and these instances include the instances of both local workflow processes and relative workflow processes. The *workflow manager* initiates instances via the *process instantiation starter*, uses the *worklist generator* for human involvement, and dispatches intra-organisational interactions and inter-organisational interactions to the *process executor* and the *workflow coordinator*, respectively.

- *Correlation database* stores the correlation information between collaborative business process instances, including the IDs of local workflow instances, the IDs of correlated workflow instances belonging to partner organisations, the IDs of corresponding relative workflow instances, etc.

- *Workflow coordinator* particularly handles the cooperation with the *Workflow Execution Services* of partner organisations. Technically, *workflow coordinator* takes care of the partner links with “foreign” *invoke* attributes, which stand for cross-organisational interactions. After Each interaction, the *workflow coordinator* needs to update the *correlation database* with newly retrieved correlation information.

- *Worklist generator* is responsible for updating the worklists for participating users. A worklist displays the details of tasks that are assigned to a user. The user can view and complete these tasks through the integrated interface, which is shown in Figure 7.12.

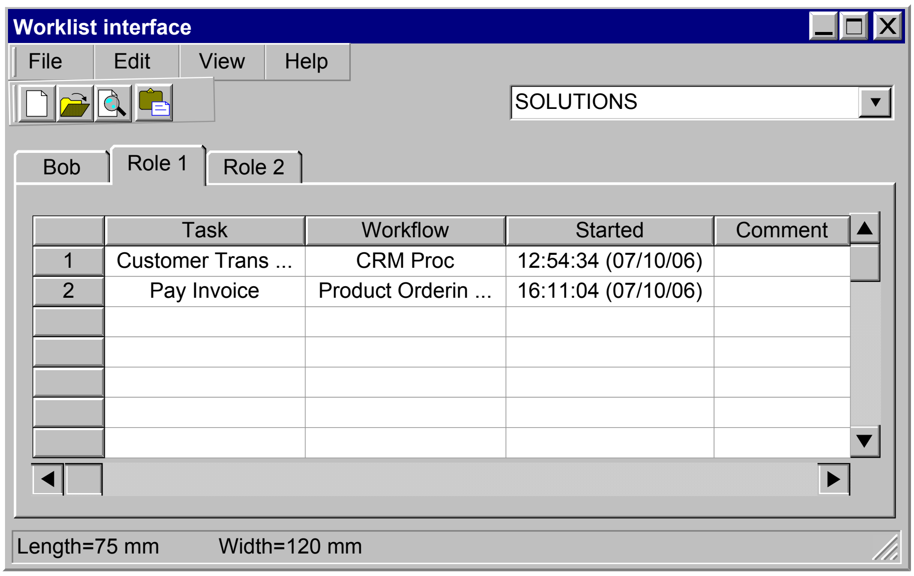

Figure 7.12 Worklist user interface

As the organisation-oriented view methodology models collaborative business processes from the perspective of each individual organisation, different organisations may create different relative workflow processes for the same collaboration. Therefore, the critical mission of Workflow Execution Service is to harmonise the underlying business collaboration with these different relative workflow processes. The instance correlation identifies the coupling relationship between collaborating workflow processes, and therefore the Workflow Execution Services can follow these instances correlations to coordinate the collaboration. In particular, Algorithm 4.3 of Chapter 4 details how we specify instance correspondences in the context of collaborative business processes.

7.2.5.2 Execution interaction

Regarding the example business process discussed in Section 7.1.2, Figure 7.13 describes a part of the interaction between related components of two collaborating Workflow Execution Services. The blocks at the top of Figure 7.13 represent the components of the retailer's Workflow Execution Service and the manufacturer's Workflow Execution Service. The vertical bars denote different processing phases, while the arrows denote the messages between components.

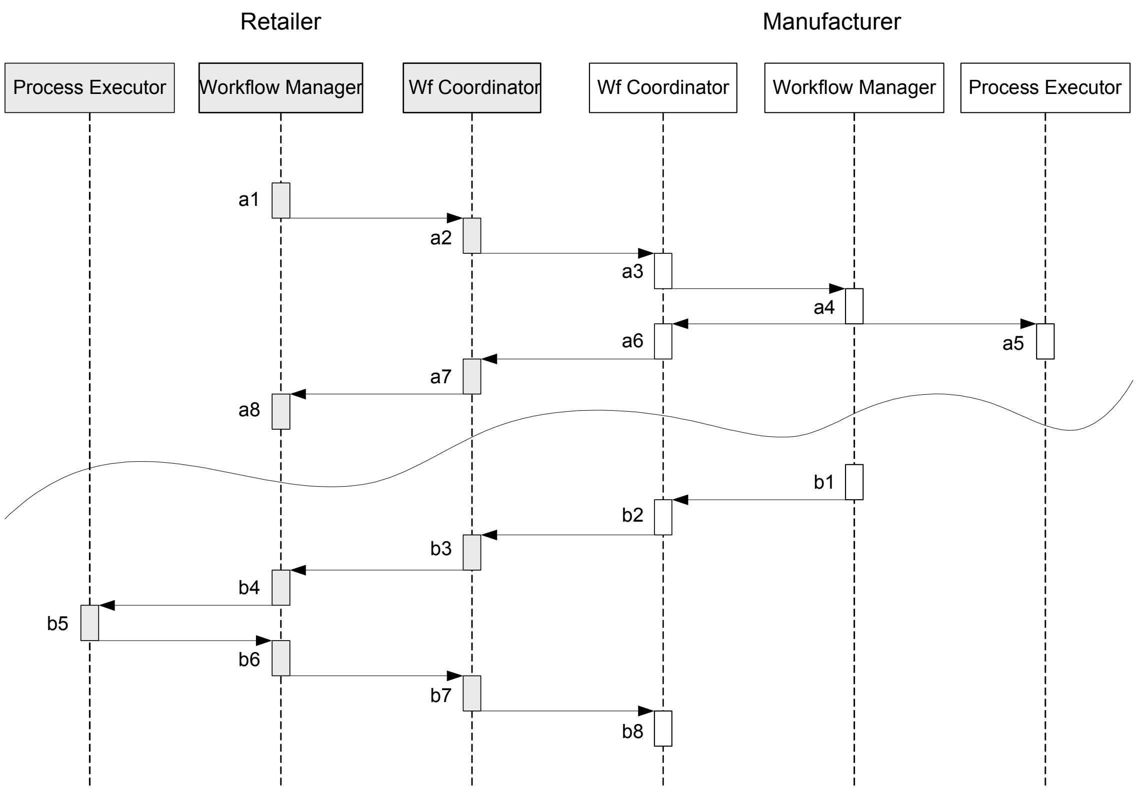

Figure 7.13 Execution interaction

Here, phase "a1" denotes that the *workflow manager* executes the WS-BPEL business process given in Figure 7.4 to the point of the following segment.

```
< invoke name =     "PlaceOrderWithManuTask"
  partnerLink =     "orderProduct"
   invokeType =     "foreign" />
```

According to Figure 7.13, the interaction sequence and each interaction step are detailed as below:

I. In phase "a1", the "invokeType" attribute of this partner link is set "foreign", which means this partner link leads to an interaction across organisational boundary. Therefore, the *workflow manager* passes this partner link and the instance ID of the running WS-BPEL business process to the *workflow coordinator*.

II. In phase "a2", the *workflow coordinator* first checks the *correlation database* for pre-existing correlations, according to the received instance ID. As this is the first interaction between the retailer and the manufacturer, there is no pre-existing correlation yet. Thus, the *workflow coordinator* needs to contact the manufacturer's *workflow coordinator*, and pass this partner link to the latter.

III. In phase "a3", the *workflow coordinator* of the manufacturer receives this partner link, and then it checks the *correlation database* for pre-existing correlations. Finally, it redirects the partner link to the *workflow manager*.

IV. In phase "a4", the manufacturer's *workflow manager* first determines the proper local production workflow instance to handle the requested order, and then notifies the *workflow coordinator* about the selected production workflow instance. At the same time, the *process executor* is notified to execute the partner link by invoking the specified Web service port in phase "a5".

V. In phase "a6", with the notification of the selected production workflow instance, the *workflow coordinator* creates a new correlation between the selected production workflow instance and the requesting product ordering workflow instance. This new correlation is saved in the *correlation database*. After that, the *workflow coordinator* replies to the retailer's *workflow coordinator* with the newly created correlation.

VI. In phase "a7", the retailer's *workflow coordinator* records the correlation retrieved from the manufacturer, and notifies the *workflow manager* that the partner link has been successfully handled. Then, the workflow manager will go ahead with the next task, i.e., "Invoice customer" in phase "a8".

As the collaboration goes on, the manufacturer runs to the "confirm delivery" task, which corresponds to the following segment in the manufacturer's WS-BPEL business process.

```
< invoke name =     "ConfirmDeliveryTask"
  partnerLink =     "confirmDlvry"
   invokeType =     "foreign"/>
```

According to Figure 7.13, the interaction sequence and each interaction step are detailed as below:

I. In phase "b1", the manufacturer's *workflow manager* informs the *workflow coordinator* to handle the "foreign" partner link; and in phase "b2", the *workflow* *coordinator* checks the *correlation database*, and finds the correlated product ordering workflow instance, for example the instance ID of the correlated product ordering workflow process is “RP-01-34”.

II. In phase “b3”, the *workflow coordinator* contacts the counterpart of the retailer, with the partner link and the instance ID of correlated workflow process, i.e., “RP-01-34”.

III. At the site of the retailer, the *workflow coordinator* passes the partner link to the *process executor* via the *workflow manager* in phases “b3” and “b4”, respectively. Thereafter, in phase “b5”, the *process executor* invokes this partner link using local product ordering workflow instance “RP-01-34”.

IV. The *workflow coordinator* checks if any new correlations are generated during this partner link execution according to the retrieved information from the *process executor* and *workflow manager* in phase “b6” and “b7”, respectively.

V. If needed, the *workflow coordinator* will synchronise the new correlations with the manufacturer’s *workflow coordinator* in phase “b8”.

7.2.6 Workflow Monitor Service

The *Workflow Monitor Service*, shown in Figure 7.14, is to handle both intra-organisational and inter-organisational workflow tracking. As discussed in Chapter 5, the inter-organisational workflow tracking starts from a relative workflow instance, and then spreads to the correlated workflow instances beyond neighbouring organisations. Therefore, the inter-organisational workflow tracking follows a propagation process. Particularly, the *monitoring manager* is employed to propagate status enquiries to partner organisations. In addition, this manager also responds to partner organisations’ enquires by reporting the execution status of requested local workflow instances with the help of a *local workflow process locator* and a *local monitor*.

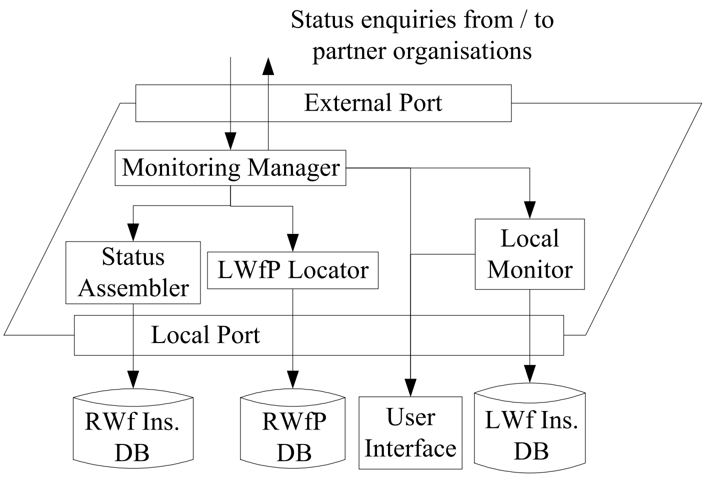

Figure 7.14 Workflow Monitor Service

The major components of Workflow Monitor Service are listed below:

- *Monitoring manager* is in charge of the collaborative business process monitoring in general. When tracking the perceivable workflow processes of a relative workflow process, the manager propagates status enquiries to partner organisations, and asks the *status assembler* to incorporate the retrieved status information to the workflow instances stored in the relative workflow instance database. Algorithm 5.2 of Chapter 5 details this tracking procedure. On the other hand, when receiving the status enquiries from partner organisations, the manager calls the *local workflow process locator* to map the requested perceivable workflow instance to the corresponding local workflow instance, and then asks the *local monitor* to check the execution status of this local workflow instance, and finally returns the status information back.

- *Local workflow process locator* (*LWfP Locator*) searches the local workflow process database to find the corresponding local workflow process specified by the status enquiry from partner organisations. Thereafter, it reports the found local workflow process to the *monitoring manager*.

- *Local monitor* is responsible for monitoring the execution of local workflow instances. The user interface of this local monitor is given in Figure 7.15.

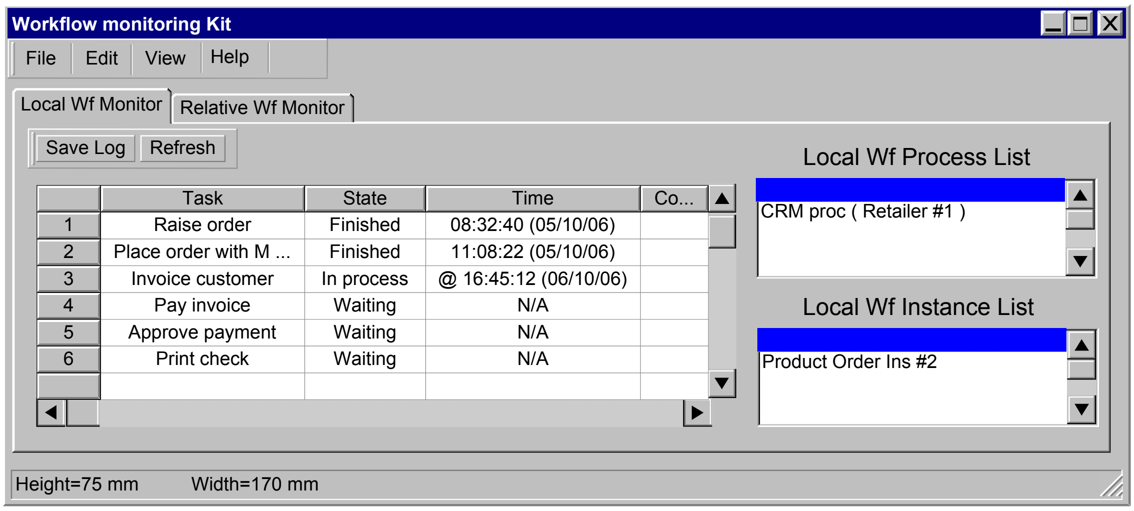

Figure 7.15 Local workflow monitor interface

- *Status assembler* is particularly used to incorporate the retrieved execution status to relative workflow instances. During the propagation, Algorithm 5.1 of Chapter 5 may be used to create tracking structures.

7.3 Summary

This chapter described our work on facilitating the organisation-oriented view methodology with Web service technologies. At modelling level, this work extends the current WS-BPEL standard to characterise the specific components, processes and operations of our relative workflow model. At system level, this work proposes an architecture design of a business process management system comprising four major services, namely, *Agreement Management Service*, *Workflow Modelling Service*, *Workflow Execution Service* and *Workflow Monitor Service*. These four services fully support the life cycle of relative workflow process modelling and execution.

Chapter 8

Review and Discussion

8.1 Thesis Review

This thesis is dedicated to provide a comprehensive solution for collaborative business process management. Theoretically, this solution relies on an organisation-oriented view methodology, which observes a collaborative business process from the perspective of individual organisations. This perspective innovatively emphasises the autonomy and the privacy protection issues in business collaborations. With this perspective, each organisation proactively selects partner organisations, and customises a collaborative business process according to its business objectives and partnerships. We believe this perspective naturally characterises how organisations conduct business collaboration with partner organisations.

Figure 8.1 illustrates the major components of this solution based on the organisation-oriented view methodology. A relative workflow model is proposed to formalise the organisation-oriented view methodology at process level. Unlike public view workflow modelling approaches, the relative workflow model allows the relativity of organisations in terms of perceiving same collaborative business process. In addition, this relative workflow model follows a bottom-up assembling scheme, and therefore can build up a collaborative business process in a “browse-and-pick” mode.

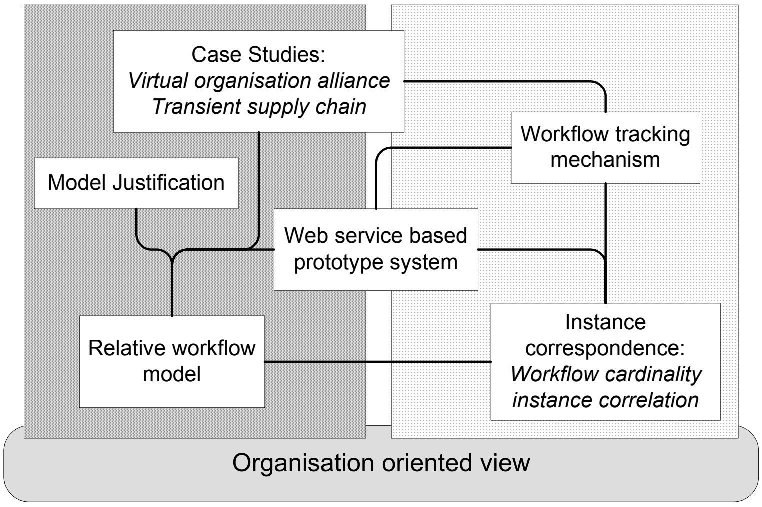

Figure 8.1 Organisation-oriented research framework

On the basis of this relative workflow model, a Petri net based method is proposed to identify instance correspondence in collaborative business processes, in terms of workflow cardinality and instance correlation. Furthermore, a workflow tracking mechanism based on matrices is proposed to enforce workflow tracking across organisational boundaries and even beyond neighbouring organisations. Two related case studies on virtual organisation alliances and transient supply chains further justify the applicability and show advantages of our proposed solution in practical collaboration scenarios. A prototype system design is also given to demonstrate the feasibility of the whole solution, and its deployment in the Web service environment.

Next, this chapter is to review these works in details.

- *Relative workflow model*

At the process level, the relative workflow model defines a series of components for business process modelling in the organisation-oriented view context.

First, a set of visibility constraints are defined to characterise the granularity of organisations' perception on a specific business process. Currently three visibility constraints are used to set a task to be either invisible, contactable or trackable for a partner organisation. With these visibility constraints, an organisation can customise the levels of openness of its own business processes for different partner organisations according to the partnerships and authority levels.

Second, an original intra-organisational business process, which is defined as a local workflow process, will be converted to a perceivable workflow process after applying these visibility constraints. Thus, in the organisation-oriented view context, a collaborative business process contains both local part, i.e., local workflow processes, and perceivable part, i.e., perceivable workflow processes from partner organisations. A typical procedure for building up a collaborative business process therefore follows a "browse-and-pick" mode. On one side, each organisation publishes perceivable workflow processes to a public directory service; on the other side, an organisation may check those published workflow processes that are perceivable from it, then choose appropriate ones and create a collaborative business process by linking the selected perceivable workflow processes with its own local workflow processes. Hence, different organisations may generate different collaborative business processes in the form of relative workflow processes for the same collaboration. This feature reflects the relativity characteristics.

Last, corresponding algorithms are given to detail the procedures for extracting visibility constraints from commercial contracts, converting a local workflow process into a perceivable workflow process according to proper visibility constraints, and assembling local workflow process(es) and perceivable workflow process(es) into a relative workflow process. These algorithms promote the modelling automation up to operational level.

- *Model justification*

At process level, we justify the proposed relative workflow model in terms of information sufficiency and necessity. The proof for information sufficiency is based on the reasoning from relative workflow definition; and the proof for information necessity is based on the analysis of workflow functioning scope.

- *Instance correspondence*

On the basis of these process level work, we shift our focus to instance correspondence in the collaborative business process context. As a relative workflow process always comprises multiple local workflow processes and perceivable workflow processes, complex correspondences may exist between the instances of these involved processes at run time. To tackle this issue, a novel correspondence Petri net model is proposed to characterise instance correspondence in terms of the workflow cardinality at build time and the instance correlation at run time.

At build time, particular cardinality parameters are invented to describe the quantitive relationship between participating business processes. To adapt to the organisation-oriented observation mechanism, we convert the four traditional bilateral relations, viz., single-to-single, single-to-many, many-to-single and many-to-many, to two uni-directional relations, viz., to-single and to-many. Correspondingly, two cardinality parameters are defined to represent these two relations.

At run time, the interactions and the resulted common business data between collaborating business processes structure the correlations between business processes. The underlying correlations combine the related business process instances together into a logical instance of a collaborative business process. This logical instance indicates an actual case of meaningful business collaboration, and this logical instance acts as the virtual object for collaborative business process management. A correlation structure is therefore introduced for business process correlation specification.

Finally, we incorporate the cardinality parameters and the correlation structure into the traditional Petri net model with proper extensions, and provide a solution to specify instance correspondences. Corresponding algorithms are also given to formalise the generation of such a correspondence Petri net, and illustrate how to specify and trace instance correlations on the fly with the generated Petri nets.

- *Workflow tracking*

To exploit the practical application of instance correspondence for collaborative business processes, we forward our research to inter-organisational workflow tracking in particular. In the collaborative business process context, the workflow tracking requires the perception on the whole collaboration, which always goes beyond neighbouring organisations.

On the basis of the proposed relative workflow model, we realise visibility transitivity via “trackable” visibility constraint. Furthermore, a tracking structure is proposed to represent an organisation’s perception over the collaboration. This tracking structure employs matrices to represent the trackable tasks and perceivable inter-process links. Correspondingly, three representation matrices are developed for describing tracking structures. In addition, three matrix operations are defined to formalise the procedures for composing, connecting and extending a tracking structure. Corresponding algorithms detail how an organisation updates a tracking structure and uses it to collect execution status from partner organisations on the fly.

Compared with other workflow tracking solutions, our approach not only enables an organisation to track other organisations for its involved parts of collaborative business processes, but also allows different organisations to track same collaborative business process differently.

- *Case studies*

To further consolidate the feasibility of the organisation-oriented view solution, we conducted two case studies on its application in typical B2B collaboration scenarios.

The first case study focuses on the collaboration among Australian toolmaking firms, which form a virtual organisation alliance for speciality synergy. As the virtual organisation alliance is fully motivated by demand-and-supply relations, the member organisations share a very loose and transient partnership with each other. A virtual organisation alliance may be formed or dismissed dynamically. Our organisation-oriented view solution provides good supports to those typical features, such as trustiness, uni-directional contracting and agile collaboration of a virtual organisation alliance. The visibility control mechanism firmly guarantees the business privacy between collaborating organisations, and prevents potential authority violation during collaboration. The modelling of relative workflow processes follows a "browse-and-pick" assembling mode, and this proactive modelling mode particularly suits the uni-directional contracting of a virtual organisation alliance. In addition, this modelling mode empowers organisations to customise their collaboration processes according to their own business objectives, and therefore organisations can collaborate in an agile manner.

The second case study analyses a dairy production supply chain collaboration. As a supply chain may grow to include the suppliers' suppliers and the customers'

customers, the structure of a supply chain may scale up on the fly. Driven by the market demands, the participating organisations of a supply chain may also change their business processes or collaboration processes to adapt to customers' requirements. Therefore, the partnerships between organisations of a transient supply chain are dynamic rather than static. In the organisation-oriented view context, this scalable structure can be obtained by inserting corresponding perceivable workflow processes, and the dynamic partnerships can be represented by updating perceptions. Besides, the inter-organisational workflow tracking approach discussed in Chapter 5 can assist organisations to track and trace the execution progress of a supply chain. The extendable tracking structure well suites the scalable supply chain.

- *Facilitating system implementation*

To facilitate the organisation-oriented view methodology, we designed system architecture for collaborative business process management. To ease the implementation work for collaborative business processes, we adopted the popular Web service technology, implemented this architecture by a series of Web services, and deployed WS-BPEL language as the default business process definition language.

To realise the organisation-oriented modelling perspective, we extended the current WS-BPEL language with a series of new elements and attributes. First, elements <perceivableProcesses> and <perceivableProcess> are designed to partition a WS-BPEL business process into an executable part and a perceivable part. These two parts correspond to the local workflow processes and perceivable workflow processes belonging to a relative workflow process, respectively. Second, a new attribute "invokeType" is incorporated to WS-BPEL partner links to denote whether a partner link leads to a cross-organisation invocation. This attribute will assist the orchestration between the business process management systems of participating organisations. In addition, standard WS-BPEL <correlation> element is extended with a new attribute "cardinality" for the cardinality parameter. Finally, we give a mapping from relative workflow model components to WS-BPEL and WSDL elements for convenient conversion.

To enforce the B2B collaboration execution, we designed a new business process management system on a Web service platform, which allows each participating organisation to collaborate with each other. This system consists of four administrative services, viz., Agreement Management Service, Workflow Modelling Service, Workflow Execution Service and Workflow Monitoring Service. In addition, a common Business Process Directory Service is used for all organisations to publish their business processes in the form of perceivable workflow processes. First, the Agreement Management Service extracts visibility constraints from commercial contracts and publishes the generated perceivable workflow processes to the Business Process Directory Service; Then, the Workflow Modelling Service assembles relative workflow processes with its local workflow processes and the perceivable workflow processes from partner organisations; Thereafter, the Workflow Execution Services of collaborating organisations collectively handle the execution of relative workflow processes. Last, the Workflow Monitor Service is responsible to track the execution status of running relative workflow process instances.

8.2 Advantages of this Research

The proposed relative workflow model is dedicatedly designed to support *cooperation*, *autonomy* and *openness* in the context of collaborative business processes. This model applies a task level visibility control mechanism to secure business privacy and organisational autonomy. The visibility transitivity enables the extendable perception scope of an organisation. To compare our relative workflow model with other business process models, we summarise the differences between these models in Table 8.1. These items are classified in terms of modelling direction, supports for autonomy, privacy and view, as well as information hiding mechanism and information perception scope.

Table 8.1 Comparison table

	Modell-ing direction	*Autonomy*	*Privacy*	*Design view*	*Information hiding mechanism*	*Information perception scope*
Relative workflow	Bottom-up	Fully concerned; Full control of designing	Task level, classified visibility control	Organisation-oriented view	Customisable visibility constraints	Beyond neighbouring organisations
Workflow view	Bottom-up in general	Implicitly concerned	Discarding private tasks in the view	Restricted view, but not explicitly organisation-oriented	Workflow views with interoperable interfaces	May cross neighbouring organisations
Public-to-Private	Top-down	Not concerned	Black-boxes substitute private tasks or sub-processes	Partial public view	Private processes	May cross neighbouring organisations
CrossFlow	Bottom-up in general	Not concerned, as it focuses on tightly coupled processes		Public view	Proxy-gateways provides primitive integrity and security protection	Cover all partner organisations
WS-BPEL	Bottom-up in general	Not concerned	Wrapped with Web services	Public view	Coarse grained Web services	Perception within neighbouring organisations.

From Table 8.1, we see that the relative workflow model is superior in its fine granularity of visibility control and flexible perception scope. The bottom-up modelling direction and the organisation-oriented observation together support a kind of “browse-and-pick” mode for assembling collaborative business processes. In this mode, each organisation owns the full control of choosing partner organisations and customising collaboration processes. Finally, this organisation-oriented view allows that different organisations can observe the same business process differently, according to the diverse partnership and business objectives.

In summary, our organisation-oriented view solution possesses the following appealing features:

- *Support of high autonomy in collaborations*

As an autonomous entity, each organisation is in charge of defining the collaboration structure and behaviours with its partner organisations. This mechanism enables an organisation to fulfil its own business planning and management without being forced to adapt to the restrictions or irrationalities from a third party designer or a main contractor. Thus, each organisation owns high autonomy in handling its business collaboration.

- *Support of information protection*

The visibility control mechanism prevents private information from disclosure at both task level and process level. A participating organisation is now able to tune the openness level on its internal business processes to partner organisations, according to different partnerships. Therefore, this mechanism guarantees the organisation's privacy protection during collaborations, and it also secures the necessary openness for cooperation at the same time.

- *Support of flexible collaboration*

The "browse-and-pick" modelling mode of relative workflow processes frees organisations from inflexibility of pre-defined collaborative business processes. Organisations can change partner organisations, modify the collaboration behaviours, insert or remove proper business processes to or from the existing collaborative business process, etc. All of these customisations can be done in an ad hoc manner. Thus, our organisation-oriented view solution supports the flexible collaboration.

- *Advanced information hiding mechanism*

The proposed constraint-based visibility control mechanism provides a powerful yet flexible information hiding solution for collaborating organisations. This visibility control mechanism well distinguishes the diverse partnerships and authority levels between collaborating organisations. Other research works, such as workflow view and public-to-private approaches, attempt to support information hiding using partial workflow views and private processes, respectively. However, the partial workflow view and the private process only provide a primitive information hiding mechanism. Neither of them combines the observation with corresponding partner organisations, and therefore fails to reflect the diverse partnerships between collaborating organisations.

- *Scalable perception scope*

Some approaches, like WS-BPEL, model a collaborative business process from the perspective of a pivot organisation. Though it follows the organisation-oriented observation, the generated collaborative business process can only include the directly interacted business processes from partner organisations. This limits the host organisation's perception within its neighbouring organisations. On the contrary, our relative workflow model enables the visibility transitivity via "trackable" visibility constraints. Therefore, an organisation's perception scope can extend beyond neighbouring organisations to cover the business processes of non-neighbouring organisations.

8.3 Tradeoffs of this Research

The migration to organisation-oriented business process management may bring some tradeoffs, which can be potential limitations. Some tradeoffs and deduced limitations are summarised as follows, although they may be outweighed by many advantages offered by our methodology:

(1) In the relative workflow context, different organisations deploy different collaborative business processes for the same collaboration. The inconsistence between these collaborative business processes inevitably results in the complex coordination between participating organisations. In practical application environment, this may require extra functionalities for storage and coordination. In the proposed prototype system architecture, three components, namely the relative workflow process database, perceivable workflow database and local workflow process database, are particularly designed for business process definition storage. More components, such as the workflow coordinator of Workflow Engine Service, perceivable workflow process locator and local workflow process locator of Workflow Modelling Service are designed for the coordination.

(2) The extraction of visibility constraints from commercial contracts assumes that all contracts conform to the format defined in the COSMOS model. This assumption may not stand in most cases, since many contracts are issued in natural languages. Therefore, the automatic conversion algorithm is only applicable in limited situations. Instead, extra human efforts may be required for interpreting the contracts to perceptions.

(3) The Petri net based instance correspondence approach relies a lot on dynamic propagation for run time correlation tracing. Such propagations are likely to increase the communication and computation load of the whole system. The workflow tracking approach uses matrices to represent tracking structures. Yet the matrices may contain quite a lot of zeros, which result in space inefficiency of data storage. This problem can be erased by adopting some special data structures, such as chain table, in our future work.

(4) The extension to the standard WS-BPEL language is only a primitive proposal at current stage, which requires more adoption for wider application in practice. A more comprehensive extension including extra extended elements or attributes for instance correspondence and workflow tracking will be done to enhance WS-BPEL towards our organisation-oriented view methodology. At appropriate time, we may submit the proposal to WS-BPEL standardisation institute for evaluation.

8.4 Summary

In summary, our organisation-oriented view research is dedicated in supporting collaborative business process management with focuses on three fundamental issues, viz., autonomy, openness and cooperation. This research moves current collaborative business process management technology forward to an advanced stage with more pragmatic features for modern collaborations. In addition, this research yields a comprehensive solution for collaborative business process management. The deployment of the organisation-oriented view solution will contribute a lot to empower organisations with a competitive edge.

On the other hand, the advantages of the organisation-oriented view methodology are achieved at the cost of some compromises. Extra attention should be paid to these tradeoffs in future research to counteract and minimise these limitations.

Chapter 9

Conclusions and Future Work

9.1 Summary of the Thesis

In this final and concluding chapter, we summarise the major contributions of the work, and provide an outlook into potential future activities.

The primary focus of the research presented in this thesis, falls in the area of collaborative business process management. The organisation structure of this thesis went as follows:

- Chapter 1 introduced the definition of business processes, and reviewed the history of business process management. Chapter 1 also described the aims of this work, the key issues addressed in this thesis and the structure of this thesis.

- Chapter 2 reviewed the related standards, research and products. The past work has put tremendous efforts on intra-organisational business process management, and already arrived to a relatively mature stage. On the contrary, technologies for inter-organisational business process management are still struggling to meet the same level. Most work done in the area of inter-organisational business process management mainly concentrated on communication protocols, universal business process definition languages, infrastructure systems, but stopped before some non-functional yet crucial issues, such as organisational autonomy, privacy protection, change management etc. Some research, such as the workflow view approach and the public-to-private approach, attempted to tackle the privacy disclosure in business collaboration. However, supports from these approaches are still far from satisfactory.

- Chapter 3 analysed the requirements for inter-organisational business process management, especially in modern collaboration scenarios. Three important problems, viz., low organisational autonomy, coarse granularity of openness, poor flexibility and adaptability, were pointed out by analysing a motivating example. Thereafter, an organisation-oriented workflow model, i.e., the relative workflow model was proposed to tackle the mentioned problems. This model defined a collaborative business process from the perspective of individual organisations, and applied a set of visibility constraints to customise partner organisations' perception on business processes. A series of definitions and corresponding algorithms were given to formalise this model and related procedures.

- Chapter 4 focused on instance correspondence in collaborative business processes. The instance correspondence was characterised in terms of workflow cardinality and workflow correlation. A novel correspondence Petri net model was established with proposed cardinality parameters and correlation structures. With the help of this correspondence Petri net model, an organisation can identify the dynamic instance correlations between collaborating business processes, and trace the execution status of the collaborative business process.

- Chapter 5 furthered the instance correspondence research for conducting inter-organisational workflow tracking. A matrix-based framework, including a series of representation matrices and matrix operations, was presented to illustrate tracking structure representation and generation. Corresponding algorithms were given to describe the dynamic tracking procedure beyond organisational boundaries.

- Chapter 6 studied two cases to demonstrate the practical deployment and virtues of the proposed organisation-oriented view methodology. The first case study exploited the collaboration within a toolmaking virtual organisation alliance. The second case study analysed the collaboration within a dairy production transient supply chain. The two case studies illustrated that our organisation-oriented view methodology is practically applicable, and is beneficial to support modern collaboration applications.

- Chapter 7 presented a business process management system design for the organisation-oriented view methodology, which was implemented for demonstration and proof-of-concept purposes. This system design was based on the Web service platform, and deployed WS-BPEL as default process definition and execution language with special extension for relative workflow features. The system design proved the concepts of organisation-oriented view are technically feasible at implementation level.

- Chapter 8 gave an in-depth discussion of the pros and cons of the proposed methodology, together with a comparison with peer approaches. The advantages of the organisation-oriented view methodology were summarised, and at the same time the tradeoffs were also pointed out.

9.2 Contributions

The significance of this research is that it fundamentally addresses the urgent issues in the area of collaborative business process management, i.e., formal modelling, instance correspondence, inter-organisational workflow tracking and infrastructure system design.

This research investigates a novel framework for collaborative business process management, which can be regarded as a paradigm shift. This framework and corresponding organisation-oriented view methodology exploit the observation perspective in business process modelling and execution, with particular concerns on privacy protection and organisational autonomy. The major outcomes of this research, namely, the relative workflow model, the instance correspondence approach, the inter-organisational workflow tracking approach and the prototype system design, provide a comprehensive solution to current collaborative business process management. Therefore, this research illustrates cutting-edge technologies for modern business collaboration coordination and management. Critical features, i.e., autonomy, openness and cooperation, which are overlooked or neglected by most research, are sufficiently discussed and well supported by our research. This research contributed new and advanced knowledge to the area of business process management. With this knowledge, business process technologies can be deployed to much wider application scenarios with various kinds of organisations. In particular, the major contributions of this thesis are listed below:

- *The identification of existing problems in conventional business process management and their causes.*

  This thesis has illustrated that the existing problems in conventional collaborative business process management, such as limited organisational autonomy, low privacy protection and poor flexibility, are ultimately caused by the public view design scheme. Based on this finding, it has been advocated that this public view over collaborations should be replaced with a relative view. With this relative view, different organisations can observe the same collaboration differently, and therefore this relative view reflects the inherent organisation-oriented nature of collaboration management.

- *The innovative organisation-oriented view perspective for modern collaborative business process modelling and management.*

  The organisation-oriented view perspective emphasises the distinction between participating organisations in authority levels, partnerships etc. This innovative perspective can be considered as a paradigm shift for collaborative business process modelling and management, since the traditional public view neutralises the diversity between participating organisations. This view perspective provides a pragmatic foundation for future research targeting business security and privacy.

- *The proposal of relative workflow model.*

  Relative workflow model establishes a formal foundation for the organisation-oriented view methodology. This model defines a series of terms for business process representation in the organisation-oriented view context. It also specifies the detailed procedure for extracting visibility constraints from contracts, wrapping perceivable workflow processes, and generating relative workflow processes. Therefore, the proposal of the relative workflow model materialises the organisation-oriented view methodology from a concept to an operational model.

- *The approaches on instance correspondence and tracking functionality.*

Past research on instance correspondence or workflow monitoring and tracking stopped at a superficial or conceptual level. On the contrary, this thesis conducted an in-depth investigation on instance level management over business processes, and explicitly formalised two approaches for instance correspondence and inter-organisational workflow tracking, which are founded on a correspondence Petri net model and matrices, respectively. Following the organisation-oriented view perspective, these two approaches explained how to identify instance level correlations, and how to trace the execution status of a collaborative business process using these correlations. This research contributed to the organisation-oriented view methodology with instance level enhancements.

- *The Web service based system design.*

A system has been designed on Web service platform for demonstration and proof-of-concept purposes. The current version has manifested the feasibility of deploying our organisation-oriented view solution to support business collaborations. This system serves a good basis for future extension, evaluation and improvement of the approaches proposed in this thesis.

9.3 Future Work

In the future, further investigation will be carried out to refine the organisation-oriented view solution. Future research includes an extended set of visibility constraints, improvement of the system design, migration of the system design to SAP product line, incorporating instance correspondence to the standard business process definition language, structural and logical validation, and so on.

- *An extended set of visibility constraints.*

Current visibility constraint set, containing “invisible”, “contactable” and “trackable” constraints, is designed to differentiate three visibility levels, and support business process interaction and tracking. New types of visibility constraints are to be created for further complicated functionalities, such as historical data enquiry for order trace back in supply chain scenarios, critical synchronisation points in virtual organisation alliances, etc. Thereby, the visibility control will be improved to an advanced level, and in turn can better adapt to modern collaborations.

- *Improvement of the system design.*

The prototype system architecture discussed in this thesis only pictures a rough diagram about how the component services are specified and how they cooperate to fulfil basic business process management functionalities. Currently, only the four major management functionalities, viz., contracting, modelling, execution and monitoring, are concerned. Yet, some other functionalities, like role management, historical data logging, application programming interfaces and so on, are not touched in this design. More amendments are to be done in the future to refine the system design.

- *Incorporating instance correspondence to BPEL language.*

Our extension to current BPEL language mainly focuses on differentiating the local part and perceivable part of a BPEL business process. The unidirectional cardinality parameters are the only improvement for instance correspondence support. To fully support the instance correspondence mechanism and inter-organisational workflow tracking, instance correlation structures and proper operators for tracking are expected to be incorporated to BPEL specification in our future work.

- *Validation of collaborative business processes.*

The work reported in this thesis does not cover the business process validation in terms of structure or logic. Yet, collaborative business processes are inherently weak in structure and logic, since they are normally created on site without any pre-existing execution experience. Moreover, the visibility control mechanism also complicates the validity of generated collaborative business processes. Thus, it is necessary to simulate and validate the collaborative business processes before applying them in practice. Particular policies are therefore expected for regulating business process validity in the organisation-oriented view context. Corresponding simulation and analysis approaches and tools are also expected in our future work.

Bibliography

[1] WfMC: "Workflow Management Coalition - Terminology & Glossary (WFMC-TC-1011), Feb-1999, 3.0", 1999.

[2] van der Aalst, W.M.P., ter Hofstede, A.H.M., and Weske, M.: "Business Process Management: A Survey", In Proceedings of International Conference on Business Process Management, pp.1-12, 2003.

[3] Smith, H. and Fingar, P.: Business Process Management - The Third Wave: Mehan-Kiffer Press, 2003.

[4] van der Aalst, W.M.P.: "Interorganizational Workflows: An Approach based on Message Sequence Charts and Petri Nets", Systems Analysis - Modelling - Simulation, vol. 34, pp.335-367, 1999.

[5] IBM: "IBM WebSphere Business Integration Handbook", 2005.

[6] SAP: "SAP Business Workflow and WebFlow Documentation".

[7] Oracle: "Oracle BPEL Process Manager Documentation".

[8] ForresterResearch: http://www.forrester.com.

[9] Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., Trickovic, I., Weerawarana, S.: "Business Process Execution Language for Web Services (BPEL4WS) 1.1", 2003.

[10] ebXML: "http://ebxml.org", 2002.

[11] RosettaNet: "http://www.rosettanet.org/", 2001.

[12] Action Technologies, Inc., (http://www.actiontech.com), 1984.

[13] TIBCO Software Inc., (http://www.tibco.com/), 1985.

[14] Digital Equipment Corporation, (http://www.digital.com), 1957.

[15] Hewlett-Packard Company: Process Manager, (http://www.ice.hp.com).

[16] WfMC: The Workflow Management Coalition, (http://www.wfmc.org), 1993.

[17] OMG: The Object Management Group, (http://www.omg.org), 1989.

[18] OMG/BPMI: "Object Management Group / Business Process Management Initiative - Business Process Modeling Notation (BPMN) 1.0", 2004.

[19] OMG: "Final Report of the Workflow Management Facility revision Task Force 1.2. Technical Report dtc/99-08-04", 1999.

[20] OASIS: The Organization for the Advancement of Structured Information Standards (OASIS), (http://www.oasis-open.org), 1993.

[21] BPMI: The Business Process Management Initiative, (http://www.bpmi.org).

[22] W3C: The World Wide Web Consortium, (http://www.w3c.org), 1989.

[23] OBI: The Open Buying on The Internet (OBI) Consortium, (http://www.openbuy.org/).

[24] Dumas, M., van der Aalst, W.M.P., and Hofstede, A.H.M.t.: Process-Aware Information Systems: Wiley Interscience, 2005.

[25] WfMC: "Workflow Management Coalition - The Workflow Reference Model (WFMC-TC-1003), 19-Jan-95, 1.1", 1995.

[26] WfMC: "Workflow Management Coalition Reference Model Interface 4 Workflow Interoperability: The key to E-Commerce and to process scalability".

[27] Hoare, C.: Communicating Sequential Processes: Prentice-Hall, 1985.

[28] Milner, R.: Communication and Concurrency: Prentice-Hall, 1989.

[29] Standardisation, I.O.f.: "LOTOS: A Formal Description Technique Based on the Temporal Ordering of Observational Behaviour", 1989.

[30] van der Aalst, W.M.P.: "Formalization and Verification of Event-Driven Process Chains", Information & Software Technology, vol. 41, pp.639-650, 1999.

[31] van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., and Barros, A.P.: "Workflow Patterns", Distributed and Parallel Databases, vol. 14, pp.5-51, 2003.

[32] van der Aalst, W.M.P. and ter Hofstede, A.H.M.: "YAWL: Yet Another Workflow Language", Information Systems, vol. 30, pp.245-275, 2005.

[33] WfMC: "Workflow Management Coalition - Workflow Process Definition Interface -- XML Process Definition Language (XPDL) (WFMC-TC-1025), October-2005, 1.14", 2005.

[34] van der Aalst, W.M.P. and Weske, M.: "The P2P Approach to Interorganizational Workflows", In Proceedings of International Conference on Advanced Information Systems Engineering, pp.140-156, 2001.

[35] Chiu, D.K.W., Cheung, S.C., Till, S., Karlapalem, K., Li, Q., and Kafeza, E.: "Workflow View Driven Cross-Organizational Interoperability in a Web Service Environment", Information Technology and Management, vol. 5, pp.221-250, 2004.

[36] Liu, D.-R. and Shen, M.: "Modeling Workflows with a Process-View Approach", In Proceedings of International Conference on Database Systems for Advanced Applications, pp.260-267, 2001.

[37] Schulz, K.A. and Orlowska, M.E.: "Facilitating Cross-organisational Workflows with a Workflow View Approach", Data & Knowledge Engineering, vol. 51, pp.109-147, 2004.

[38] Zhao, J.L.: "Workflow Management in the Age of e-Business (Tutorial)", In Proceedings of International Conference on System Sciences, 2002.

[39] Perrin, O. and Godart, C.: "A Model to Support Collaborative Wrk in VIrtual Enterprises", Data & Knowledge Engineering, vol. 50, pp.63-86, 2004.

[40] Arkin, A.: "Business Process Modeling Language (BPML)", 2001.

[41] Thatte, S.: "XLANG - Web Services for Business Process Design", 2001.

[42] Alonso, G., Casati, F., Kuno, H., and Machiraju, V.: Web Services: Concepts, Architectures and Applications: Springer Verlag, 2004.

[43] van der Aalst, W.M.P.: "Process-Oriented Architectures for Electronic Commerce and Interorganizational Workflow", Information System, vol. 24, pp.639-671, 1999.

[44] Muth, P., Wodtke, D., Weissenfels, J., Angelika Kotz Dittrich, G.W.: "From Centralized Workflow Specication to Distributed Workflow Execution", Journal of Intelligent Information Systems, vol. 10, pp.159-184, 1998.

[45] WfMC: "Workflow Management Coalition - Wf-XML 2.0", 2005.

[46] Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S.: "Web Services Description Language (WSDL) 1.1", 2001.

[47] Leymann, F.: "Web Services Flow Language (WSFL) 1.0", 2001.

[48] Arkin, A., Askary, S., Fordin, S., Jekeli, W., Kawaguchi, K., Orchard, D., Pogliani, S., Riemer, K., Struble, S., Takacsi-Nagy, P., Trickovic, I., and Zimek, S.: "Web Service Choreography Interface (WSCI) 1.0", 2002.

[49] Barros, A.P., Dumas, M., and ter Hofstede, A.H.M.: "Service Interaction Patterns", In Proceedings of Proceedings of the 3rd International Conference on Business Process Management (BPM 2005), Nancy, France, pp.302-318, 2005.

[50] OASIS: "OASIS Web Services Security (WSS) Standards", 2006.

[51] IBM, BEA_Systems, Microsoft, SAP_AG, Sonic_Software, and VeriSign: "WS-Policy Specification", 2006.

[52] Griffel, F., Boger, M., Weinreich, H., Lamersdorf, W., Merz, M.: "Electronic Contracting with COSMOS - How to Establish, Negotiate and Execute Electronic Contracts on the Internet", In Proceedings of International Enterprise Distributed Object Computing Workshop, pp.46-55, 1998.

[53] Verharen, E., Papazoglou, M.: "Introduction to Contracting in Distributed Transactional Workflow", In Proceedings of Annual Hawaii International Conference on System Science, 1998.

[54] Grefen, P., Aberer, K., Hoffer, Y., Ludwig, H.: "CrossFlow: Cross-Organizational Workflow Management for Service Outsourcing in Dynamic Virtual Enterprises", IEEE Data Engineering Bulletin, vol. 24, pp.52-57, 2001.

[55] Alonso, G., Fiedler, U., Hagen, C., Lazcano, A., Schuldt, H., and Weiler, N.: "WISE: Business to Business e-Commerce", In Proceedings of International Workshop on Research Issues on Data Engineering: Information Technology for Virtual Enterprises, 1999.

[56] Lazcano, A., Alonso, G., Schuldt, H., Schuler, C.: "The WISE Approach to Electronic Commerce", International Journal of Computer Systems Science & Engineering, Special Issue on Flexible Workflow Technology Driving the Networked Economy, vol. 15, 2000.

[57] Heuvel, W.-J.v.d. and Weigand, H.: "Cross-Organizational Workflow Integration Using Contracts", In Proceedings of Business Object Component Workshop of the ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications, 2000.

[58] BEA: "WebLogic Integration 7.0 Documentation", 2005.

[59] Hur, W., Bae, H., and Kang, S.-H.: "Customizable Workflow Monitoring", Concurrent Engineering-Research and Applications, vol. 11, pp.313-325, 2003.

[60] Savarimuthu, B.T.R., Purvis, M., and Fleurke, M.K.: "Monitoring and Controlling of a Multi-Agent based Workflow System", In Proceedings of Australasian Computer Science Week Frontiers, pp.127-132, 2004.

[61] Wang, M. and Wang, H.: "Intelligent Agent Supported Flexible Workflow Monitoring System", In Proceedings of 14th International Conference on Advanced Information Systems Engineering, pp.787-791, 2002.

[62] Guabtni, A. and Charoy, F.: "Multiple Instantiation in a Dynamic Workflow Environment", In Proceedings of 16th International Conference on Advanced Information Systems Engineering (CAiSE 2004), Riga, Latvia, pp.175-188, 2004.

[63] Zhou, J., Shi, M., and Ye, X.: "On Pattern-based Modeling for Multiple Instances of Activities in Workflows", In Proceedings of International Workshop on Grid and Cooperative Computing, Hainan, China, pp.723-736, 2002.

[64] Dumas, M. and ter Hofstede, A.H.M.: "UML Activity Diagrams as a Workflow Specification Language", In Proceedings of 4th International Conference on the Unified Modeling Language, Modeling Languages, Concepts, and Tools, Toronto, Canada, pp.76-90, 2001.

[65] Anderson, M. and Allen, R.: "Workflow Interoperability - Enabling E-Commerce (WfMC White Paper)", 1999.

[66] Merz, M., Griffel, F., Tu, M.T., Müller-Wilken, S., Weinreich, H., Boger, M., and Lamersdorf, W.: "Supporting Electronic Commerce Transactions with Contracting Services", International Journal of Cooperative Information Systems, vol. 7, pp.249-274, 1998.

[67] van der Aalst, W.M.P. and ter Hofstede, A.H.M.: "Workflow Patterns: On the Expressive Power of (Petri-net-based) Workflow Languages. (Invited Talk)", In Proceedings of Proceedings of the Fourth Workshop on the Practical Use of Coloured Petri Nets and CPN Tools (CPN 2002), Aarhus, Denmark, pp.1-20, 2002.

[68] Chiu, D.K.W., Karlapalem, K., Li, Q., and Kafeza, E.: "Workflow View Based E-Contracts in a Cross-Organizational E-Services Environment", Distributed and Parallel Databases, vol. 12(2-3), pp.193–216, 2002.

[69] Zhao, X. and Liu, C.: "Tracking over Collaborative Business Processes", In Proceedings of International Conference on Business Process Management, Vienna, Austria, pp.33-48, 2006.

[70] Zhao, X., Liu, C., Yang, Y., and Sadiq, W.: "Handling Instance Correspondence in Inter-Organisational Workflows", In Proceedings of the 19th International Conference on Advanced Information Systems Engineering, Trondheim, Norway, pp.51-65, 2007.

[71] Zhao, X., Liu, C., and Yang, Y.: "An Organisational Perspective on Collaborative Business Processes", In Proceedings of the 3rd International Conference on Business Process Management, Nancy, France, pp.17-31, 2005.

[72] Petri, C.A.: Kommunikation mit Automaten, vol. PhD. Bonn: Institut fur instrumentelle Mathematik, 1962.

[73] Reisig, W.: A Primer in Petri Net Design. Berlin: Springer, 1992.

[74] Jensen, K.: Coloured Petri Nets: Basic Concepts, Analysis Methods, and Practical Use, vol. 1. Berlin: Springer, 1996.

[75] van der Aalst, W.M.P. and van Hee, K.M.: "Business Process Redesign: A Petri-net-based Approach", Computers in Industry, vol. 29, pp.15-26, 1996.

[76] WfMC.: "Workflow Management Coalition Audit Data Specification [WfMC 1015]", 1998.

[77] WfMC.: "The Workflow Reference Model, [WfMC 1003]", 1995.

[78] BEA: "Using BEA WebLogic Components", 2000.

[79] IBM.: "BPEL4WS Business Processes with WebSphere Business Integration", 2004.

[80] Osterle, H., Fleisch, E., and Alt, R.: Business Networking - Shaping Collaboration between Enterprises: Springer Verlag, 2001.

[81] Gans, G., Jarke, M., Lakemeyer, G., and Schmitz, D.: "Deliberation in a Modeling and Simulation Environment for Inter-organizational Networks", In Proceedings of Advanced Information Systems Engineering, pp.242-257, 2003.

[82] Zdravkovic, J. and Johannesson, P.: "Cooperation of Processes through Message Level Agreement", In Proceedings of Advanced Information Systems Engineering, pp.564-579, 2004.

[83] Wetzel, I. and Klischewski, R.: "Serviceflow beyond Workflow? IT Support for Managing Inter-Organizational Service Processes", Information Systems, vol. 29, pp.127-145, 2004.

[84] Lazcano, A., Schuldt, H., Alonso, G., and Schek, H.-J.: "WISE: Process based E-Commerce", IEEE Data Engineering Bulletin, vol. 24, pp.46-51, 2001.

[85] Colombo, E., Francalanci, C., and Pernici, B.: "Modeling Coordination and Control in Cross-Organizational Workflows", In Proceedings of DOA/CoopIS/ODBASE, pp.91-106, 2002.

[86] Schulz, K. and Orlowska, M.: "Facilitating Cross-organisational Workflows with a Workflow View Approach", Data & Knowledge Engineering, vol. 51, pp.109-147, 2004.

[87] Quirchmayr, G., Milosevic, Z., Tagg, R., Cole, J.B., and Kulkarni, S.: "Establishment of Virtual Enterprise Contracts", In Proceedings of Proceedings of 13th International Conference Database and Expert Systems Applications, pp.236-248, 2002.

[88] Australian Centre for Toolmaking Innovation, (http://www.austool.com.au).

[89] Gimpel, H., Ludwig, H., Dan, A., and Kearney, R.: "PANDA: Specifying Policies for Automated Negotiations of Service Contracts", In Proceedings of International Conference on Service-Oriented Computing, Trento, Italy, pp.287-302, 2003.

[90] New, S. and Westbrook, R.: Understanding Supply Chains - Concepts, Critiques, and Futures: Oxford University Press, 2004.

[91] SCC: The Supply-Chain Council, (http://www.supply-chain.org), 1996.

[92] SCC: The Supply-Chain Operations Reference-Model (SCOR), (http://www.supply-chain.org/cs/root/scor_tools_resources/scor_model/scor_model).

[93] Ayers, J.B.: Handbook of Supply Chain Management, 2nd ed: Boca Raton, 2006.

[94] W3C: "Web Services Description Language (WSDL) 1.1", 2001.

[95] Andrews, T.: "Business Process Execution Language for Web Services (BPEL4WS) 1.1", 2003.

[96] AMR Research Inc., (http://www.amrresearch.com/), 2000.