Towards a more flexible inter-organizational workflow design: the application of local criteria principle

Geng Liang
University of Wollongong

Towards A More Flexible Inter-organizational Workflow Design: The Application of Local Criteria Principle

A thesis submitted in partial fulfillment of the requirements for the award of the degree

Master of Information System (Research)

From

University of Wollongong

By

Geng Liang

(2854004)

Information Systems

School of Economics and Information Systems

2006

Thesis Certification

CERTIFICATION

I, Geng Liang, declare that this thesis, submitted in partial fulfilment of the requirements for the award of the Degree of Master of Information Systems (Research) at the University of Wollongong, is wholly my own work otherwise I have given fully documented references or acknowledgement to the work of others. The document has not been submitted for qualification at any other academic institution.

Geng Liang

March 2006

Acknowledgement

Without the help of many people, I might never have finished this thesis. I am particularly enormously indebted to Dr. Sim Kim Lau and Dr Zhaohao Sun whose patience and guidance helped me in all the time of the research. Thank you also to my colleagues from the commerce research centre for their support and valuable hints. I would also like to thank the department of Information Systems for financially supporting me to attend the conference. Finally, I am grateful to my parents who encourage and support me in many ways unconditionally throughout the duration of the thesis.

List of Publication

Liang, G, Lau, S,K. and Sun, Z. 2005 “Developing Web Services Using Workflow Model: An Inter-organizational Perspective”, in the *Proceedings of The Australian Conference of Information System (ACIS) 2005*, CD-ROM, 29 Nov.-2 Dec., Manly Pacific Hotel, Sydney, Australia.

List of Figures

List of Tables

Table of Contents

Abstract

Process oriented technique such as workflow management system is attracting increasing attentions in the e-commerce environment, especially from an inter-organizational workflow perspective. A soundness concept, which identifies if a workflow can terminate properly, is an important factor that needs to be considered when designing a workflow. In an inter-organizational workflow environment, the soundness issues become more complicated considering the size of the overall workflow and the privacy of the workflow within individual organization. One of the major approaches to tackle this problem is to use local criteria principle. This principle allows the soundness of the entire inter-organizational workflow to be achieved without knowing the detail workflow of every participating organization. This way, the privacy problem is addressed. Moreover, the coordination effort among business partners to ensure soundness can be reduced. However, this principle sometime can become too restrictive when we attempt to ensure the overall workflow is sound while changes are made to the workflow. It is not always easy to reach a balance between achieving soundness and flexibility.

The public-to-private (P2P) approach is one of the important inter-organizational workflow design methods that follow the local criteria principle. The concept of branching bisimulation is considered as the local criterion in the P2P approach. This research investigates approaches to make the P2P approach more flexible by proposing a limited equivalence concept in an inter-organizational workflow context that can function as a local criterion. This research will also prove that this new local criteria concept will make the overall inter-organizational workflow sound.

Chapter 1. Introduction

1.1. Research Background

Process oriented techniques such as Workflow Management System and Web Services Composition are attracting more and more attentions in the e-commerce environment. These techniques will result in more flexible information systems to handle the increasingly complex business processes, especially for systems integration among organizations. However before the potential of process oriented techniques can be fully realized, several theoretical issues need to be addressed. One of these issues is the "soundness" problem (van der Aalst 1996). "Soundness" is a criterion proposed by van der Aalst (1996) to ensure a process can terminate properly. It is an important issue because a mistaken process definition can lead to undesirable situation such as deadlock, which is unaccepted in real life business processes. A simple approach to verify the soundness of workflow is to require all business processes to be declared to all participants. This may be difficult in business environment due to the unwillingness of business partners to expose details of their business processes. In addition, due to the size and complexity of a workflow, a soundness verification algorithm that requires details of each component of a workflow will not provide sufficient support to a complex business process, especially for business processes that cross the boundaries of more than one organizations.

When designing workflow from an inter-organizational perspective, making sure soundness is achieved is an important issue which needs to be considered. It is relatively easy to design an inter-organizational workflow from scratch. However, to ensure existing workflows work together in an inter-organizational workflow and to allow flexible modification while ensuring

soundness is achieved at the same time become relatively complex and this is an important issue to be investigated in this research.

One of the basic principles that have been proposed to address the issue of inter-organizational workflow design and soundness verification is called the "local criteria for global soundness" principle (Kindler et al. 2000). A local criterion refers to a criterion that can be verified locally without the need to use or know detail of each component of a workflow. It is expected that as long as all the sub-workflows are sound and follow the local criteria separately under specified conditions, the whole workflow combined with these sub-workflows are guaranteed to be sound. This way, the local criteria principle makes it possible to achieve global soundness of the entire inter-organizational workflow without exposing all private workflows to every participant (Kindler et al. 2000). Several approaches have been proposed based on this principle in the literature (Kindler et al. 2000, van der Aalst 1999, van der Aalst 2003, van Glabbeek & Stork 2003, Wombacher et al. 2004). To some extent these approaches have solved the problem of inter-organizational workflow design and soundness verification. One of these workflow design approaches is called the public-to-private (P2P) approach (van der Aalst et al. 2002, van der Aalst 2003), and the concept of branching bisimulation (van Glabbeek & Weijland 1996) is considered as the local criterion in the P2P approach (van der Aalst 2003). However, one of the problems of the P2P approach is that the local criterion can sometimes become too restrictive when the direction of message exchanged across organizations is not taken into account.

1.2. Research Aim

This research aims to investigate the role of local criteria principle plays in inter-organizational workflow design and its application in soundness verification. The research also aims to investigate ways to improve the P2P approach to make it more flexible.

1.3. Research Objectives

The objectives of this research are as follows:

1. To investigate the feasibility and appropriateness of using process oriented techniques in system integration and business-to-business (B2B) scenario.
2. To investigate issues of inter-organizational workflow design and soundness verification.
3. To investigate the role of the local criteria principle in the inter-organizational workflow design.
4. To investigate ways to improve the local criterion in the P2P approach so that more flexible inter-organizational workflow is possible.

1.4. Organization of the Thesis

The remainder of this thesis is organized as follows.

Chapter 2 presents literature review. In this chapter, background of distributed information system and process oriented techniques, including workflow management system and web services composition, are introduced. Then concepts of workflow model, soundness, and local criteria for global soundness principle, as well as existing approaches addressing the problem of workflow design will be discussed.

Chapter 3 presents an improved local criterion for the P2P approach so that more flexible inter-organizational workflow design is possible. Van der Aalst (2003)'s concepts of Petri net workflow model, branching bisimulation and public-to-private mechanism for inter-organizational workflow will be used as the basis of the new local criterion. In this chapter, these concepts will be explained first before a new concept called limited equivalence is proposed to design a more flexible inter-organizational workflow. Finally, conclusion and future research direction are presented in the last chapter.

Chapter 2. Literature Review

Process oriented techniques provide an important alternative to integrate existing information systems for the need of today's increasing complex business process. Workflow management system and web services composition are two widely known process oriented techniques.

The concept of workflow model and workflow management has come into being since 1970s, and web service is a new technique that appeared after the year 2000. Recently a number of researches have been conducted to investigate the relationship between workflow management and web services (Leymann, Roller & Schmidt, 2002, van der Aalst et al. 2003, Cardoso et al. 2004, Zhao & Cheng 2005). On one hand, the impact of web services on workflow management has become more and more significant both in academic and industrial sectors, and this has made workflow management a much more important approach in information system development. On the other hand, the knowledge in workflow management domain, especially workflow models, is playing an increasingly important role in the design and development of web services composition. This research focuses on how these process oriented techniques can be used in system integration and B2B scenario; in particular from an inter-organizational workflow perspective.

To investigate the concept of workflow management and Web services, we need to start from the idea of distributed information system, especially from the perspective of application integration, and explore the concept of top-down and bottom-up approaches in information system development. This literature review is organized as follows: In section 2.1, the concept of distributed information system and middleware are discussed. Section 2.2 focuses on the

use of workflow management system (WfMS) in application integration. Then, Web services techniques will be discussed in section 2.3. Issues of inter-organizational workflow design are presented in section 2.4, and research gap is identified in section 2.5.

2.1. Distributed Information Systems and Middleware

Process oriented technique is part of a long-term effort to fully utilise existing information system and to make them work together. To truly understand how it comes into being and what problem it is expected to solve, it is necessary to consider the evolution process of distributed information system and the basic method in information system development.

According to Alonso et al. (2004), at a conceptual level, there are three layers in an information system: presentation, application logic, and resource management. Presentation layer is responsible for presenting information to external entities and allowing the external entities to interact with the system by submitting operations and getting responses from the system. Application logic layer performs the actual operation requested by the client through the presentation layer. Its result is the input of the presentation logic. Resource management layer includes data source used by the application logic layer. Data can be stored in databases, file systems, or in any other format. It is also possible that resource management layer includes data source provided by other information system. All information systems have clients, which are entities that use the services provided by the information system. (Alonso et al. 2004, p4-5). Figure 1 (adopted from Alonso et al. 2004, p4) illustrates an information system and its client.

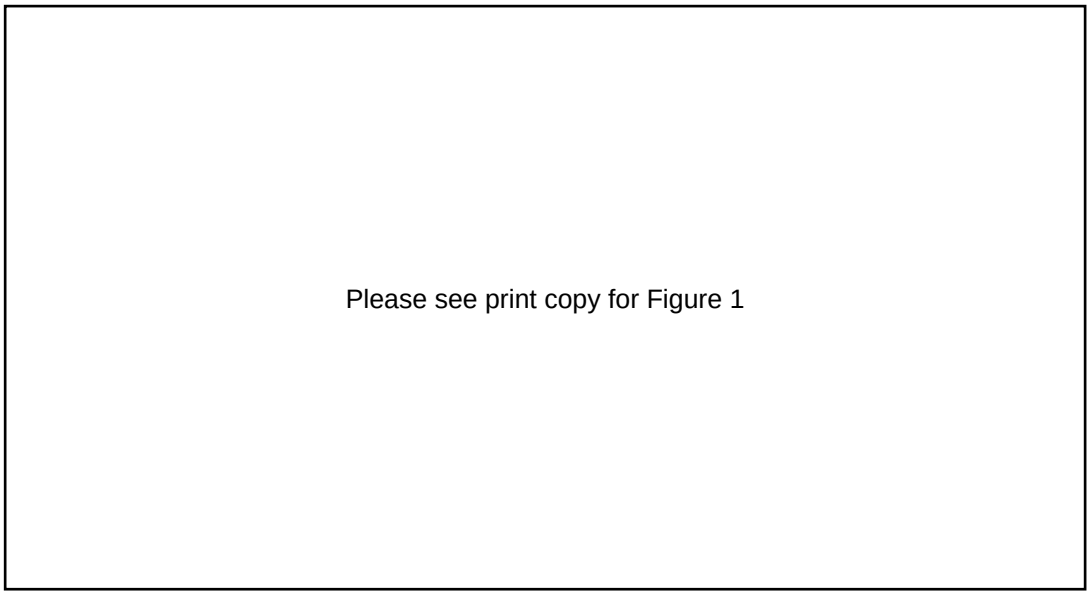

Figure 1. The different layers of an information system (Source: Alonso et al. 2004 p4-5)

There are two basic approaches to design an information system. The first approach is top-down design. Top-down design starts from the upper layer of the system, then turns to the layers supporting the upper layer. Firstly, the functionality of the system from the client's point of view is identified. After that the presentation formats and protocols are defined. Based on the definition of the presentation layer, the application logic of the system is designed. The final step is to design the data structure needed to implement the application logic (Alonso et al. 2004 p6). The top-down design is mainly suitable to be used in situation when an information system is designed entirely from scratch. However in today's business environment, this design method is rarely used because there are many information systems that have already existed and have been used effectively. It will be very inefficient to re-design all the existing applications again in order to develop a new information system. In fact few organizations can afford to do so. That is why we need a second approach to design an

information system: the bottom-up design. This method starts from the functionality needed by the client. However, in the next step, it focuses on identifying existing systems that can be used in the resource management layer of the new system. After that, the application logic and presentation logics are designed (Alonso et al. 2004 p6-9). Most of the current information systems are designed using the bottom-up design and we will focus on bottom-up design in this research. All concepts which we will discuss later in this thesis including workflow management and Web services are based on bottom-up design. They aim to address the problem of application integration, that is: "How to better make use of the existing applications when trying to fulfill new business requirements?"

The evolution of information system architecture can be seen as an investigation procedure to better solve the problem of application integration. In the following we will discuss different architectures of an information system which include 1-tier, 2-tier, 3-tier and how the application integration problem can be addressed.

In the one-tier architecture, presentation, application logic, and resource management layers are merged into a single tier. Such systems are mostly mainframes-based and the clients are dumb terminals. When they need to be integrated with other information system, the only way is screen scraping. In such situation, an external system simulates the behavior of a dumb terminal to interact with the 1-tier system. It is obvious that this procedure is highly ad hoc and inefficient, which makes it difficult to integrate with such systems. Figure 2 (adopted from Alonso et al. 2004, p12) shows a one-tier information system.

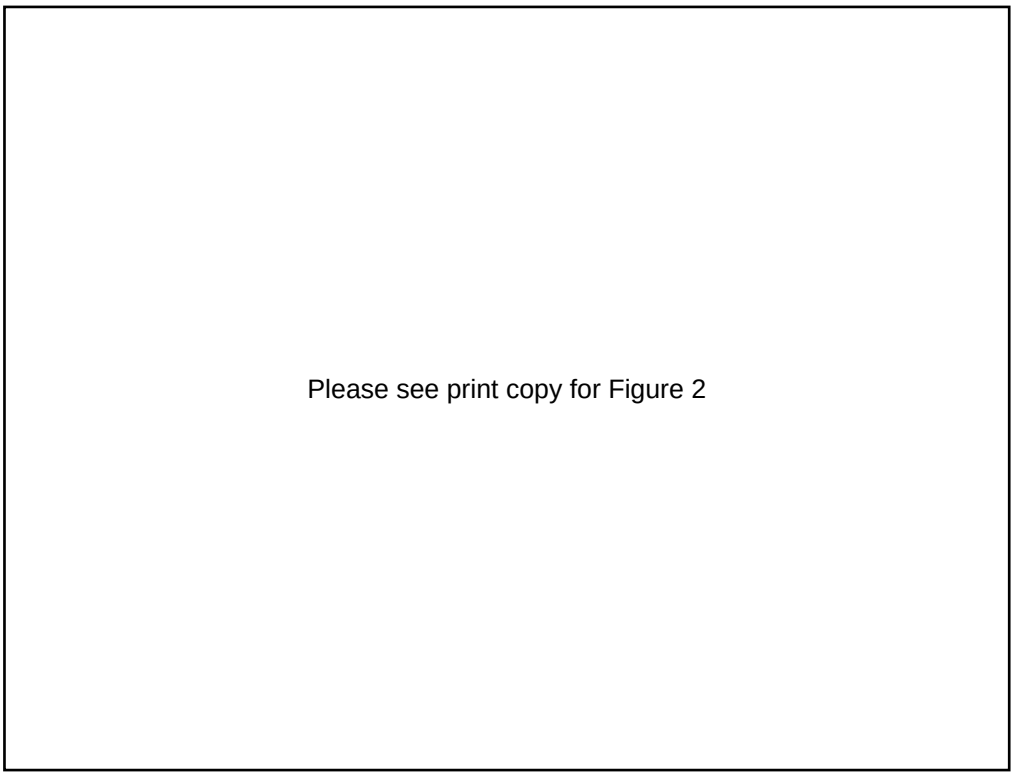

Figure 2. One-tier architecture. (Source: Alonso et al. 2004, p12)

In the two-tier architecture, presentation layer is moved to the client computer. This change leads to a new communication mechanism: remote procedure call (RPC), which enables client and server to interact by means of procedure calls. More importantly, the idea of application program interface (API) comes into being. An API specifies how to invoke a service, the responses that can be expected, and possibly even what effects the invocation will have on the internal state of the server. A server provides a set of APIs and the client can use them to interact with the server, which makes it easier to design and develop the communication mechanism. From the perspective of application integration, two-tier architecture makes it possible to use the client to integrate two information systems. In such a client, an extra

application layer is added. Figure 3 (adopted from Alonso et al. 2004, p15) shows an integration scenario using two-tier architectures.

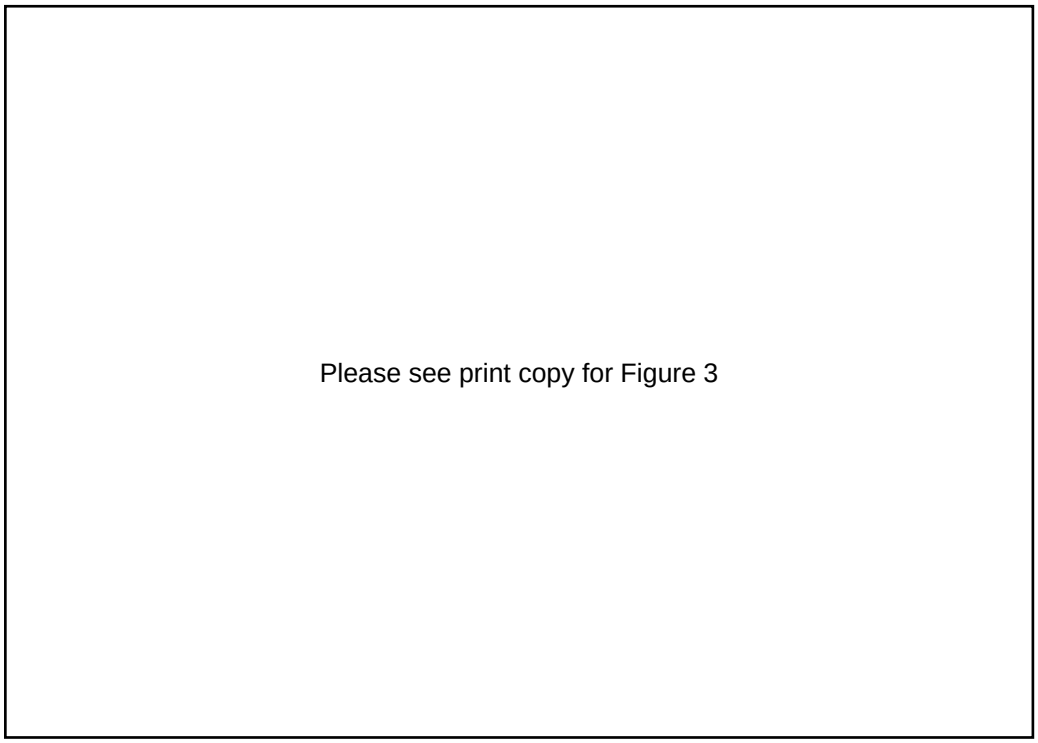

Figure 3. The client is the integration engine in two-tier architectures. (Source: Alonso et al. 2004, p15)

The mechanism of using client as the integration engine in two-tier architecture soon becomes unmanageable as the clients increase in size and complexity. Three-tier system addresses this problem by separating the three layer of an information system and using the application logic layer to integrate different applications. The application logic layer in such an architecture is also known as middleware, as it obviously resides at the middle tier. Middleware techniques such as Transaction processing monitors (TP monitors) and Common Object Request Broker Architecture (CORBA) are used to support the communication across the three tiers in such architecture. Using three-tier architecture, it is possible for the servers to interact with each

other and the clients do not need to be responsible for the integration logic. Figure 4 (adopted from Alonso et al. 2004, p17) shows a three-tier architecture in a system integration scenario.

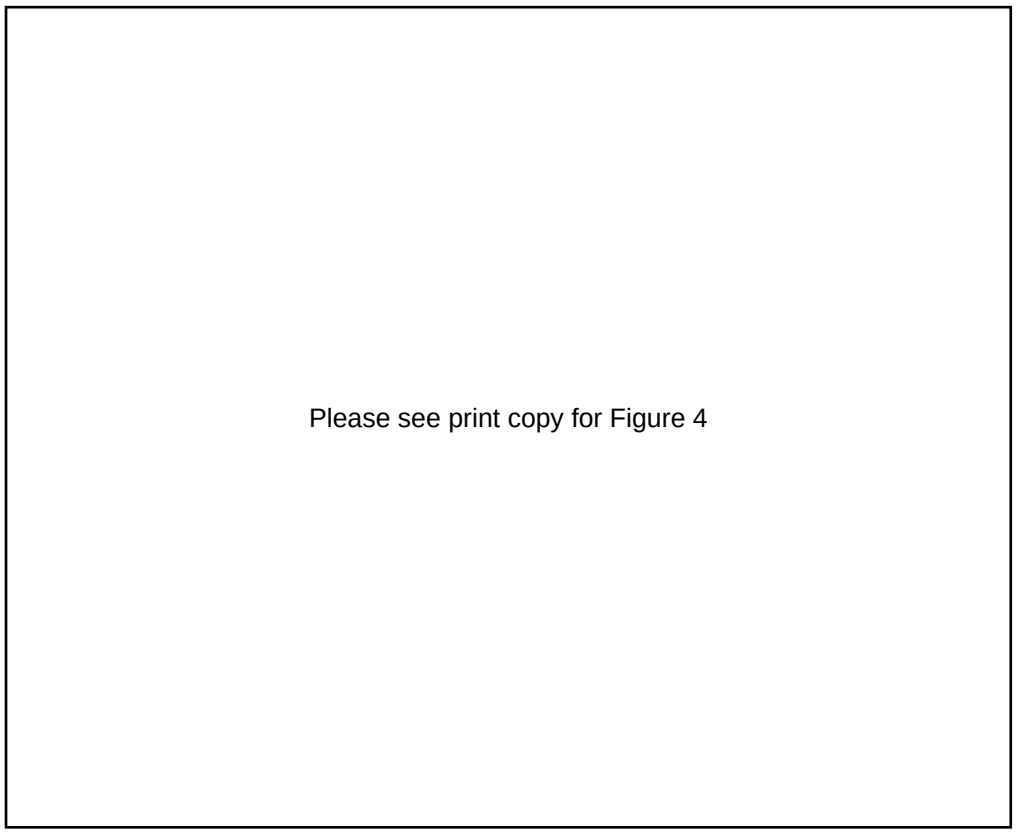

Figure 4. Integration of systems using a three-tier approach (Source: Alonso et al. 2004, p17)

The evolution procedure from one-tier to three-tier architecture makes it easier to integrate different systems into a coherent one. Even this evolution results in performance problems when communication cost across different tiers increase, the gain in flexibility outweighs the performance issues. However, in a more complex business scenario, where there are a number of applications need to be executed in a particular sequence, the three-tier system may not be able to handle such requirements. For this reason, we need process oriented technique, which will be discussed in the following section.

2.2. Workflow Management System

The techniques discussed in the last section, such as RPC, TP monitor, and COBRA, focus on point-to-point communication across different tiers of a system. However, there is no mechanism in these techniques to facilitate the definition and maintenance of the execution sequence of a list of applications, which is necessary when we try to integrate systems in a more complex business process. One perspective of the concept of business process focuses on the sequence of jobs and tasks perform in an organization. Business process is a set of one or more interdependent activities, which collectively achieve a business objective or policy goal (Chiu et al. 2004). The activities involved could be human-oriented activities or computer systems processes. In an organization, people are assigned to different tasks and they are working together to achieve a specific goal. Every member has its own responsibility and they have to work in a certain sequence to make sure the work is satisfactorily performed. Business process reengineering (BPR) is a management trend related to business process (van der Aalst 2002). This trend originated from Taylor's scientific management theory (Taylor 1911). The aim is to find ways to improve efficiency of an organization by conducting research in the routine aspect of work activities in which the goal of the business has been redefined. Every step of a business process is carefully studied to determine if a step can be removed or reorganized based on the goal of the business process.

With the use of computer systems in the organization, computer processes have replaced large number of manual works. People have worked cooperatively with computers to achieve the goals of the organization, and computer-based information systems have played an increasingly important role in the business processes of an organization. To make business

process fully automated by the computer system, one of the possible approaches is the top-down approach in information system. However, this approach which redesigns the whole data structure and information system in the organization from scratch can be expensive, and sometimes can be impossible to do because nowadays a business process is not restricted to a single organization, and the speed of change required by the competitive business environment can hinder this approach. Workflow management system is one of the products which represent the bottom-up approach when we attempt to fully utilise computer system in business processes. Workflow management system is a software system with explicit business process definition. In the 1970s, the rudiments of workflow management systems, which is often known as office information systems, were developed and implemented in the office automation domain. These systems differ from other systems on their explicit workflow models. Although there was great optimism on the applicability of office information systems in the beginning, very few applications have succeeded. Both the application of this technology and research almost stopped for a decade during the eighties. However, the idea of using a process model to define and implement a business process came into being. In the nineties, there was a renewed interest in workflow management system, particularly in the areas of banking and insurance (van der Aalst et al. 2003).

There are two types of workflow management systems: human-oriented and system-oriented workflow management systems (Georgakopoulos et al. 1995). Human-oriented workflow involves humans collaborating in performing tasks and coordinating tasks. System-oriented workflow accesses heterogeneous, autonomous, and/or distributed information systems. While human-oriented workflow implementations control and coordinate human tasks, system-oriented workflow implementation controls and coordinates software tasks (Georgakopoulos et al. 1995). When we discuss workflow management system used in

application integration, we refer to system-oriented workflow management system. Its explicit workflow model makes it much easier to design and implement business process involves in different computer systems. In the next section, we will discuss the concept of workflow model in detail.

2.2.1. Workflow Models

One of the major characteristics of workflow management systems is the separation of function logic and flow logic. Function logic deals with a specific task, such as updating a customer record or calculating order discount, while flow logic deals with combining many functions in some sequences to solve more complex problem such as processing an order (Cardoso et al. 2004). From an application integration perspective, the flow logic of a workflow management system represents the integration logic. In a workflow management system, the flow logic is represented as a workflow model. Usually a workflow model is illustrated in a graphical model representing business processes, however it can also be shown using algebra models. Several components may be included in the workflow model which includes activities, orders, states, participants and messages between participants. In general, different workflow models focus on different components of a workflow.

A workflow model usually doesn't give detail of an activity but mainly focus on flow logic. Each activity in a workflow model represents a functional logic and they are put together in a certain sequence to represent the flow logic. The execution sequence of the activities is the order of a workflow. State is another important component in a workflow model. The state of a workflow identifies which activity/activities is/are executing in the workflow. Data shared by different activities may also be included. Petri net is an example of graphical workflow

model. In a Petri net workflow model, the state of a workflow is shown by the distribution of tokens on the circles.

Participants are often not represented explicitly in a workflow model. However for an inter-organizational workflow, it is necessary to identify different participants involved in the workflow. In workflow model representation such as UML activity diagram (Chiu et al. 2004) and Petri net (van der Aalst 1999, van der Aalst 2000, Hamadi & Benatallah 2003), a vertical line is used to divide the workflows that belong to different organizations. These workflow models can be used in a more detailed workflow representation when every participant (including human and computer system) in the same organization needs to be identified. Sometimes messages between the participants of a workflow are also represented in a workflow model. There are two types of messages: synchronous and asynchronous messages. If a participant must wait for the feedback of the message before doing another task, it is a synchronous message. Otherwise, it is an asynchronous message (Alonso et al. 2004, p. 22). Synchronous and asynchronous messages are illustrated with different symbols in a message sequence chart (van der Aalst 1999).

Figure 5 shows an example of a simple workflow model using the Petri net model representation. The rectangles represent business activities, the arrows show the business rules and the relation between the business activities. The state of the workflow is illustrated by the distribution of tokens on the circles. Participants and messages between participants are not shown in a basic Petri net model.

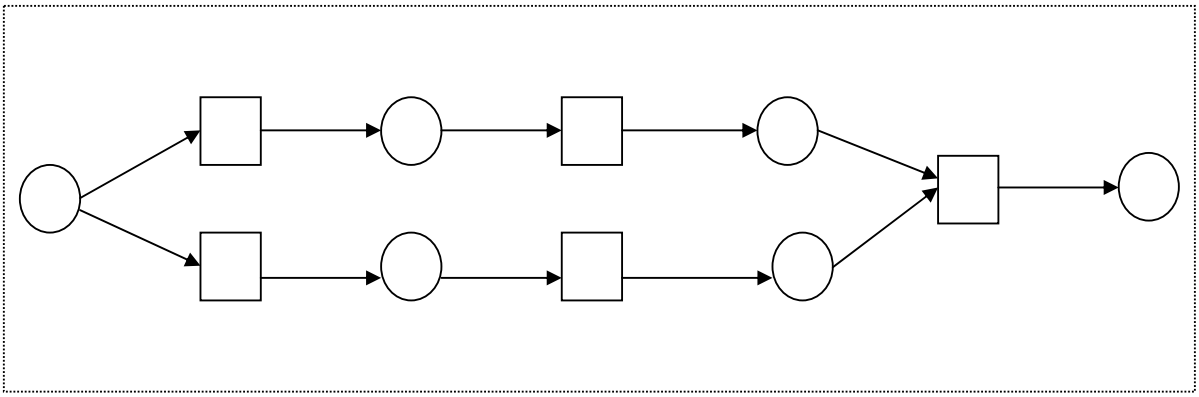

Figure 5: A Simple Workflow Model in Petri Net

The separation of flow logic and function logic makes the implementation of a workflow management system different from other computer systems. A two-level programming model, with programmers implementing the functions logic and non-programmers such as business specialists designing flow logic, is used in the implementation of workflow management system (Leymann & Roller 2002). At the core of workflow management system architecture, an "enactment service" creates workflow instances based on the workflow model and performs every step by offering work to workers if it is a human-oriented workflow management system, or launching various kinds of applications if it is a system-oriented workflow management system (van der Aalst 2002). The applications, which a workflow management system can launch, include legacy systems, databases or other workflow management systems. After the implementation of the workflow management system, the flow logic can still be changed in a flexible way and it is not necessary to implement the function logics for every activity again. This feature of workflow management system makes it much easier for organizations to engage in a BPR project. This is because their business processes are easier to understand and change. This flexibility is very important to an application integration project.

2.2.2. Inter-organizational Workflow

It is possible for one single workflow management system to be used by users in different organizations or to launch applications owned by different organizations. For example, a workflow management system can be used to integrate two or more Enterprise Resource Planning (ERP) systems used by different organizations in supply chain integration across different organizations (Cardoso et al. 2004). To clarify the term inter-organizational workflow, we refer to workflows on two or more workflow management systems.

There are two main issues that make inter-organizational workflow a domain different from workflow within an organization. Firstly, it is difficult to model and implement the interaction between two or more workflow management systems that involved in an inter-organizational workflow. It is very possible that different vendors have developed these workflow management systems separately. In general, there is still no widely supported standard for interoperability of workflow management system (van der Aalst & Kumar 2003). Secondly, it is natural that an organization is unwilling to publish its internal workflow, which makes inter-organizational cooperation more difficult (Chiu et al. 2004, Shen & Liu 2001, Kindler et al. 2000, Gou et al. 2003).

There are two modes of operation for the interaction between two different workflow management systems in an inter-organizational workflow environment: loose integration and tight integration (van der Aalst & Kumar 2003). In the loose integration mode, a main workflow keeps running in one single workflow engine. When inter-organizational integration is needed, one activity of the workflow will access the workflow provided by other organization and wait for the response. Upon receipt of the feedback, the main workflow

continues. In this mode, consistency among their workflows is hard to maintain, but it is possible for the organizations involved to keep their internal workflow secret. Several mechanisms have been proposed to prevent exposing the entire internal workflow when cooperating in an inter-organizational workflow. Examples include process-view mechanism (Shen & Liu 2001) and workflow-view mechanism (Chiu et al. 2004). These mechanisms are useful to expose part of a workflow to other organization for the purpose of developing a workable inter-organization workflow while keeping the proprietary workflow secret at the same time. But a complete algorithm to create the partial "view" from a workflow is still missing.

In the tight integration mode, when inter-organizational integration is needed, the workflow engine will ship the entire file of workflow specification to another workflow engine. When the other engine finishes its work, it will change the state of the file and transfer it back to the previous workflow engine, where the task will resume. In this mode, consistency between two workflows is easy to maintain. The disadvantage of this approach is participants in an inter-organizational workflow have to expose their entire internal workflow to one another.

2.3. Web Services

Web services are one form of distributed information systems. It is one of the most recent standardization efforts which attempt to solve many important problems in the domain of distributed information system, especially for system development using the bottom-up approach. From a function logic perspective, web services technique provides a new universal solution for the problem of how to invoke application on other systems, which is a major issue in system integration. Web service is a new solution for "machine to machine"

communication based on the Internet technology. Furthermore, the appearance of web services composition language makes it possible to integrate a number of web services into a workflow. This addresses the standardization problem of flow logic in workflow management system. The W3C definition states that 'a web service is a software application identified by a URI, whose interfaces and bindings are capable of being defined, described, and discovered as XML artifacts. A Web service supports direct interactions with other software agents using XML-based messages exchanged via Internet-based protocols' (W3C 2002).

To enable two application systems developed on different operating systems with different programming languages to understand and be able to talk to each other, it is necessary to have a standard. One set of standards that make web services technique possible include XML (eXtensible Markup Language), HTTP (Hypertext Transfer Protocol, SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language) and UDDI (Universal Description, Discovery). These standards provide a set of rules that govern how applications make use of common web service technologies so that every system is speaking the same language and makes web service interoperability practical (Monson-Haefel 2004). All of SOAP, WSDL and UDDI are presented in a XML format, and can be described through certain XML schemas. The standards of web service work together to provide the foundation of publish, discovery and access of web services. SOAP and HTTP are used to enable the interaction between two systems, whereas WSDL and UDDI make dynamic discovery of service interfaces possible. WSDL is used to describe web services and UDDI works like a Yellow Page to support the register and searching of web services. These standards were adopted by most major computer software and service companies including Sun Microsystems, Microsoft, IBM, BEA and Software AG. Figure 6 (adopted from IBM. 2001) shows the basic web services architecture.

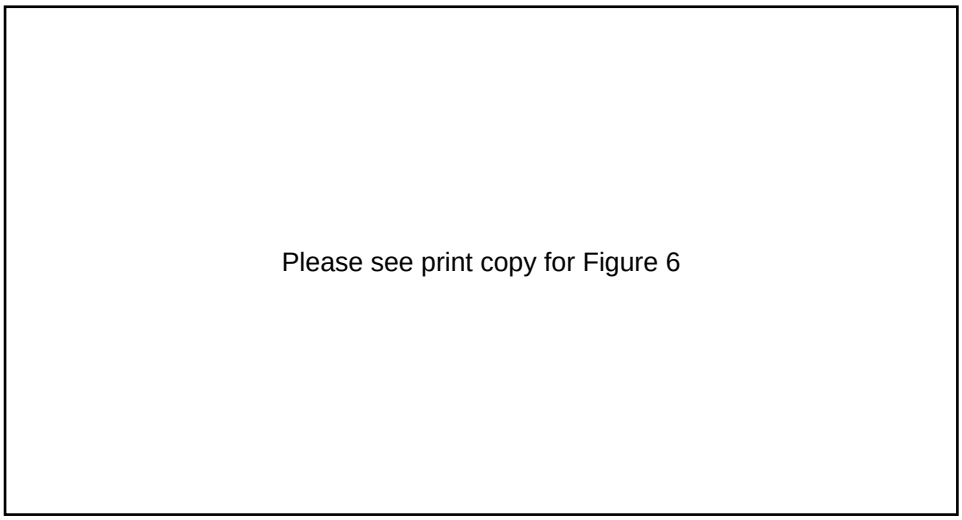

Figure 6: Web Services Architecture (Source: IBM. 2001)

This architecture is the basic architecture found in almost every introduction document to web services architecture. In Figure 6, web service registry, web service provider, and web service requester are three different software systems. Firstly, the web service provider registers its service in a web service registry. Then a service requester finds the service it needs in the web service registry. Finally, the web service requester interacts with the web service provider.

In a web services conceptual model, different standards are used in different layers to make the interoperability of web services possible. Figure 7 (adopted from IBM. 2001) shows the conceptual stack of web services. In general, the upper layers build upon the capabilities provided by the lower layers. In Figure 7, the text on the left represents standard technologies applied at that particular layer of the stack. The bottom layers of the stack, representing the base web services stack, are relatively mature and are more standardized than the layers higher in the stack (IBM, 2001).

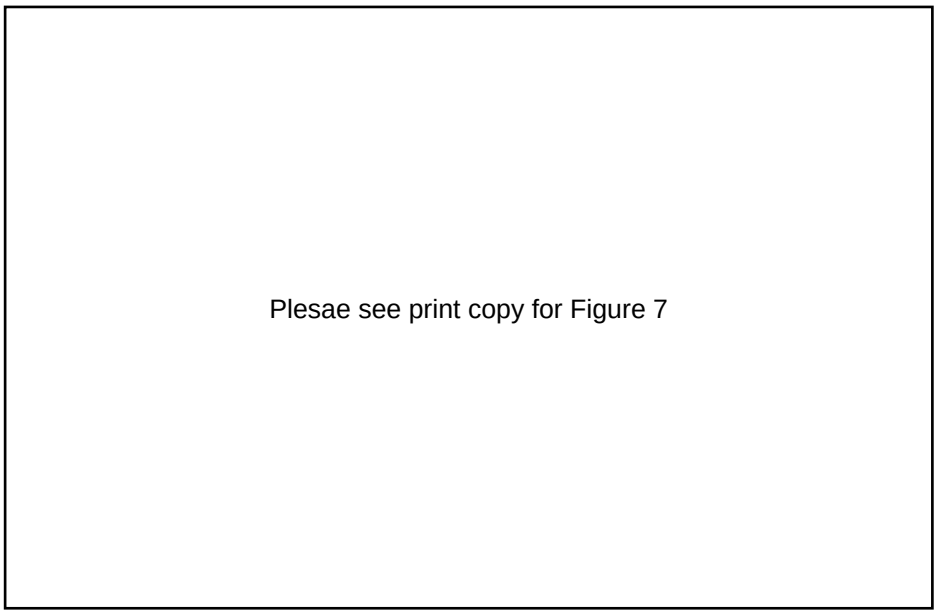

Figure 7: Web Services Conceptual Stack (Source: IBM. 2001)

We can see that the lower four layers are all well defined. They are the basis for a single trigger of web services and support the scenarios including sending a message (one-way), receiving and sending a message (request-response), sending and receiving a message (solicit-response), or receiving a message (notification) (van der Aalst et al. 2003). From a workflow perspective, these four layers are concerned with the function logic of a workflow, however they do not provide the mechanism of sequencing and state reserving. Furthermore one single web service cannot do the work of an entire business process. Very often a business process contains more than one business activities, thus the highest layer of the web service stack, the business process management layer, is required. This layer describes how a number of business activities are organized in a business process to achieve a business objective. These business processes can be modeled as a business protocol often in the form of a workflow model. Although there are some standards and products that have been developed, there is still no one widely accepted standard in this domain. It is worth noting that different names

have been used for the standards of business process management layer in the literatures (zur Muehlen et al. 2004, van der Aalst et al 2003). These include web service flow language, web service orchestration language, web service composition language, and web-enabled workflow language. We adopt the name of "web service composition language" in this thesis.

There are two types of web services composition language based on their design philosophies: REST-oriented and SOAP-oriented. A recent trend is to use these two design philosophies together in the web service composition (zur Muehlen et al. 2004). One of the SOAP-oriented web service composition languages, BPEL4WS (Business Process Execution Language for Web Services), is the most well-known web service composition language. The BPEL4WS specification is built on IBM's WSFL (Web Services Flow Language) and Microsoft's XLANG (Web Services for Business Process Design). It uses both a block-structured language and a graph-oriented model as workflow model. A process in BPEL4WS both provides and/or uses services described in the WSDL. The most significant difference between WSDL and BPEL4WS is that BPEL4WS provides state management function, which corresponds to the state factor in the workflow model (van der Aalst et al. 2003).

2.4. Inter-Organizational Workflow Design

Before we can effectively discuss workflow design, we will first need to have a way to evaluate the design. Soundness and privacy of a workflow implementation are two factors which we will consider here.

Soundness is a basic requirement for a workable workflow based on the "workflow net" model. (van der Aalst 1996). A sound workflow refers to a workflow that can always reach 'proper termination' and contain no dead transition. Informally, a 'dead transition' is a useless activity in a workflow net model. Figure 8 shows an example of unsound workflow.

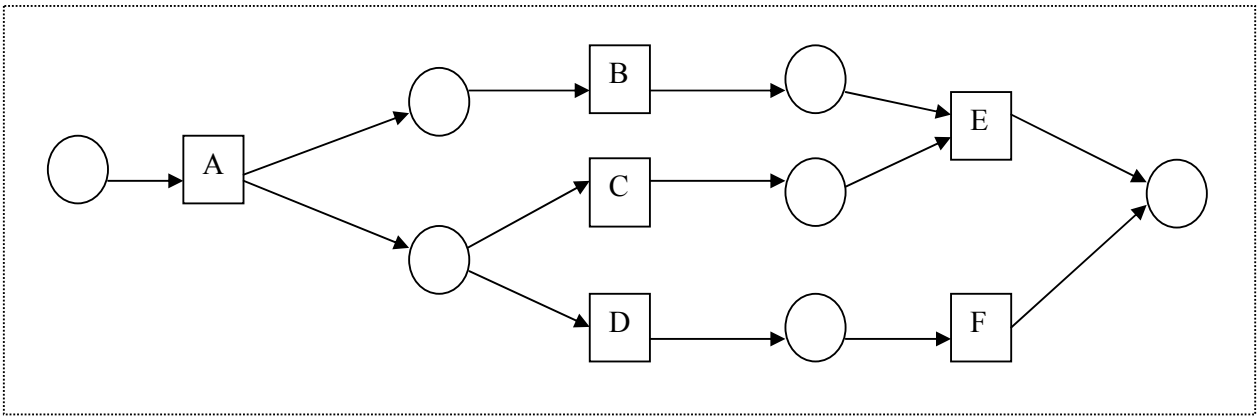

Figure 8: An Unsound Workflow

In Figure 8, activity A is executed first. Then there are two possibilities for the next task to be executed: (B and C) or (B and D). If activities B and C are executed, the whole workflow can terminate successfully after execution of E. However if activities B and D are executed, then a token will be stuck after activity B because activity E cannot be executed without C, and the whole workflow cannot terminate properly. Thus it is meaningful and important to verify the soundness of an existing workflow.

The verification of soundness for both workflows within an organization and in an inter-organizational workflow environment can be done automatically by using analytic techniques of Petri net such as liveness and boundedness verification. (van der Aalst 1996, van der Aalst 1999). However the problem of verifying soundness of an inter-organizational workflow becomes more difficult when participants are not willing to expose or reveal their internal

workflows. One approach proposed to solve this problem is based on scenario specified in terms of sequence diagrams and a slightly revised version of Petri net (Kindler et al. 2000). Gou et al. (2003) propose a similar approach based on UML sequence diagram and Petri net.

When we discuss privacy of internal workflow implementation, most literatures refer to hiding detail of private workflows within an organization (van der Aalst 1999, Kindler et al. 2000, van der Aalst 2003). But in fact, internal workflow implementation not only refers to the private workflow. It also includes third party workflows which the private workflow connects to. Consider a B2B scenario involving three participants: a customer, a distributor, and a producer. For the distributor, its internal workflow implementation includes not only its own private workflow, but also the public workflow between the distributor and the producer. This is not only common sense in real life business world, but also is a well accepted requirement for information system integration, i.e., when you call a program or function, you do not need to know the detail of its implementation, including how it uses other program modules. Instead what is required is the interface of the program module. However, this issue has not been fully addressed in existing literatures of workflow management.

Although approaches of soundness verification already existed, it is sometimes not enough to know that a workflow is not sound. We need to be able to determine which part of the workflow that causes the unsoundness and correct it into a sound workflow. Furthermore, how to design a sound workflow without exposing unnecessary private information has become an increasingly important and practical problem.

As discussed, we focus on inter-organizational workflow design in this research. Similar to the design of other information system, in practice, it is rare to design an inter-organizational

workflow from scratch assuming there is no existing workflow for all business partners. There are already many existing workflow management systems implemented in the business world. Therefore how do we make these existing workflow management systems work together and allow flexible workflow modification within an organization become an increasingly important research issue. One of the key factors to be considered to address this issue is called "local criteria for global soundness" (Kindler et al. 2000). It is expected that when a number of sound sub-workflows are organized together following a specific local criterion, the global soundness of the entire workflow is guaranteed. Such local criteria, if proven to be able to guarantee global soundness, bring several benefits. Firstly, since global soundness comes with local criteria, all business partners no longer need to expose or reveal details of their private workflows for soundness verification. All that need to be done is to check if their private workflows have satisfied the local criteria, which addresses the privacy issue. Secondly, when one or more business partners want to modify their private workflows according to their own business needs, such local criteria can provide a guideline to make sure the modification does not harm the global soundness. Finally, when new business partners need to be brought into an inter-organizational workflow, the local criteria principle makes it possible to keep the entire workflow sound and minimize the coordination with every existing business partners at the same time. This way it makes B2B cooperation more flexible. In the literatures, several approaches (Kindler et al. 2000, van der Aalst 1999, van der Aalst 2003, van Glabbeek & Stork 2003, Wombacher et al. 2004) based on this principle have been proposed. Although not all of them result in all the above benefits, the results reported in the literatures still allow us to understand more about the relation between sub-workflows and the entire workflow and make it easier to design a sound inter-organizational workflow when there are existing sub-workflows.

A natural way to propose a local criterion is to distinguish between public workflows and private workflows first, and then prove that if all the private workflows are consistent with the public workflow in some ways, then the overall workflow is sound. Van der Aalst (1999)'s approach follows this intuition. It is based on a workflow model using Petri net and messages sequence diagram. Firstly, a "workflow net" model using Petri net concept is used to describe the private workflow. Then synchronous and asynchronous messages connecting different organizations are modeled using message sequence diagram. Using such a model, a "consistency" criterion is proposed to decide if a private workflow presented by the Petri net can work with the public workflow presented by the message sequence diagram. Unfortunately, it is very difficult to even define the criteria in a situation when there are more than one message sequence charts involved, which is common in practice. As a result, only situation called "1-consistent", corresponding to situation when only one message sequence chart is involved can be considered, which is too restrictive. Kindler et al. (2000) proposed a similar but more comprehensive approach based on the concept of scenarios, which is similar to the concept of a message sequence chart. This approach is able to deal with situation when more than one scenarios are involved. However, Kindler et al. (2000) did not provide proof for one of the theorems reported to achieve this.

van Glabbeek & Stork (2003) proposed a local criterion with the concept of 'Query Nets', which does not need the notion of a public workflow. This approach is restrictive in another way: the messages exchanged among different private workflows need to be organized as pairs of input and output places. Obviously real life business process does not have such restrictions.

Wombacher et al. (2004)'s approach does not need the notion of a public workflow either. It was inspired by the appearance of Web services technique. Web services architecture provides a mechanism to enable publishing and searching for a single web service. Thus it is useful when an organization wants to outsource part of its workflow. However when two organizations want to integrate their workflows seamlessly, this mechanism of web services is not enough. The flow logic of the workflows involved should also be able to be published and searched. Wombacher et al. (2004) propose a match making mechanism based on a revised version of Finite State Automata (FSA) model to decide whether two workflows are compatible to work together. Using this method, it is easier for two organizations to establish business relations without relying on manual negotiation for the specification of public workflow. Figure 9 shows a procurement match making scenario using the FSA model. The scenario is adopted from Wombacher et al. (2004).

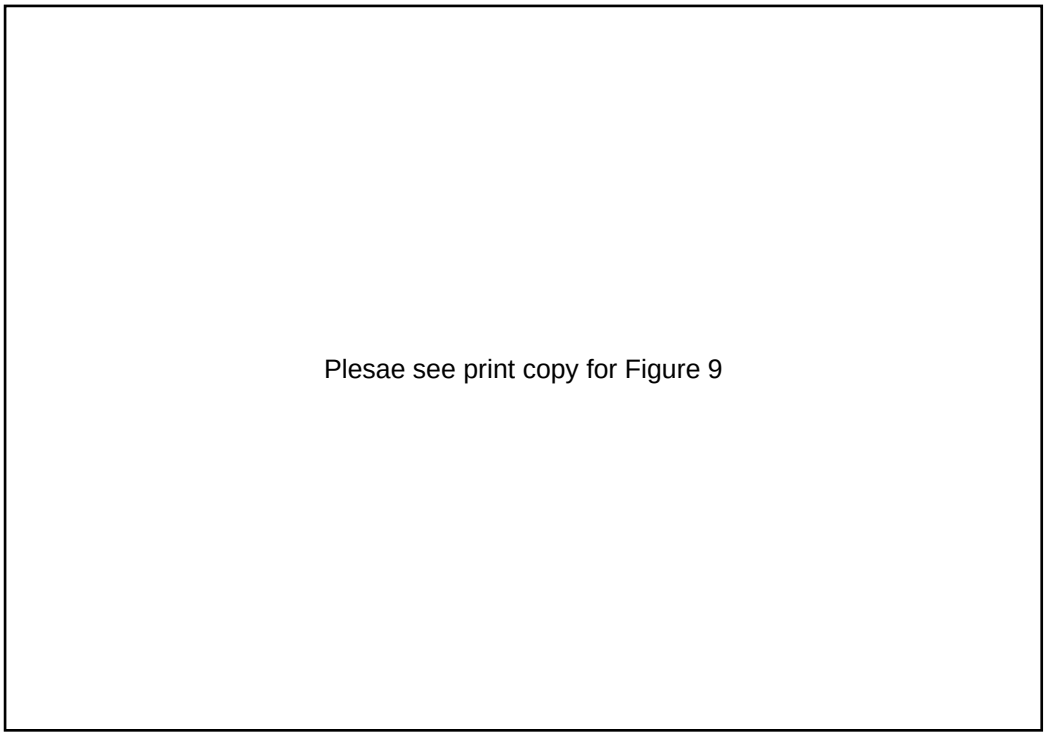

Figure 9: A procurement match making scenario using the FSA model. (a) Vendor Message Sequence (b) unmatched Customer Message Sequence (c) matched Customer Message Sequence (Source: Wombacher et al. 2004)

The FSA model focuses on the messages and states of a workflow. The states are represented with circles and the arrows show the messages. The sending and receiving messages lead to the change of state of the workflows. Figure 9 (a) shows the message sequence supported by a vendor from its customers. First, an order message is received. If the product is not available, a message of "no stock" will be sent back. Otherwise, the vendor will deliver the product and then receive the payment. Two types of payment methods including credit card and invoice payments are supported. There are two errors in the message sequence in Figure 9 (b) if we want to use it as a customer message sequence to interact with the vendor's message sequence showed in Figure 9 (a): the first one is that its payment message is sent before the delivery

message, the second is that is doesn't support the "no stock" message, which is a "mandatory" message (Wombacher et al. 2004). On the other hand, the message sequence in Figure 9 (c) matches with the vendor message sequence although it doesn't support the invoice payment message because the invoice payment message is an "optional" message. Wombacher et al. (2004)'s approach can decide whether two workflows are compatible to work together in an automatic way. However, this match making mechanism does not consider the internal implementation of the public workflow and the situation when more than two participants are involved in an inter-organizational workflow.

This research concentrates on the P2P approach proposed by van der Aalst (2003). The P2P approach contains three steps. Firstly, the public part of the inter-organizational workflow is created. Secondly, the public workflow is partitioned over the organizational entities involved. Finally, for each organizational entity, a private workflow that is a subclass of the relevant part of the public workflow is created using the concept of projection inheritance. According to van der Aalst (2003), the resultant inter-organizational workflow using this approach is guaranteed to be sound. Furthermore, as the public workflow is designed before the private workflow is created, it is not necessary to consider the issue of keeping the internal workflow secret. As long as the private workflow is a subclass of their relevant part of the public workflow under the concept of projection inheritance, the organizations involved in the inter-organizational workflow can design and modify their internal private workflows in any way they want, and it is not necessary to be concerned with the soundness of the entire inter-organizational workflow (van der Aalst 2003). Workflow inheritance is a concept derived from the concept of branching bisimulation. Basically, workflow inheritance refers to a subclass of a workflow under projection inheritance that can simulate any behavior of its super class after hiding all the new methods added in the subclass. The concept of branching

bisimulation and workflow inheritance and the P2P approach will be further discussed in chapter 3. Furthermore, in the P2P approach, branching bisimulation and projection inheritance can be considered as the local criteria.

2.5. Research Gap

From the literatures above, we can see that local criteria play a very important role in most workflow design approaches. Without such a method of local criteria ensuring global soundness, soundness verification would require all participants to reveal their workflow details after any modification is made to any part of the workflow. However, a local criterion sometimes can become too restrictive when we try to protect the soundness of the entire workflow from changes. At the same time it prevents possible workflow modification without affecting the overall soundness of the entire workflow. It is not always easy to achieve a balance between guaranteeing soundness and flexibility. In this research, we attempt to improve the flexibility of the P2P approach (van der Aalst 2003) by proposing a limited equivalence concept as a local criterion in the inter-organizational workflow context. We will also prove that this new local criterion does not affect the soundness of the entire workflow, which is important to be considered based on the local criteria principle.

Chapter 3. Inter-Organizational Workflow Design Using Local Criteria Principle

In this chapter, an improvement for the local criteria in the P2P approach (van der Aalst 2003) will be introduced. This research proposes a new local criteria concept that can make P2P method to be more flexible and at the same time guaranteeing global soundness of the inter-organizational workflow. As the proposed improvement is derived from van der Aalst (2003)'s Petri net workflow model and the workflow inheritance framework, we will describe his framework first. In the first section of this chapter, relevant concepts of van der Aalst (2003)'s labeled Petri net model, WF-net and workflow inheritance concept will be presented. In the second section, the new local criteria concept called the limited equivalence will be presented and we will prove it to follow the principle of local criteria for global soundness, thus ensuring the entire workflow is sound. The third section summarizes this chapter.

3.1. Preliminary Concept on Labeled Petri net, WF-net, and Workflow Inheritance

In this section, we present van der Aalst (2003)'s work on labeled Petri net, WF-net model and workflow inheritance as the basis of our proposed improvement framework to be presented in the second section of this chapter. We will use the same symbols and definitions

as in van der Aalst (2003)'s paper. In this section we will only present those definitions given by van der Aalst (2003) that are relevant to the proposed improvement here. A complete list of definitions can be found in van der Aalst (2003)'s paper.

The following are the basic mathematical notations used. Let U be some universe of identifiers and L be some set of action labels. $L_v = L \setminus \{\tau\}$ is the set of all visible labels with τ represents a silent action, which is action that cannot be observed. Silent action is a concept used in defining branching bisimilarity, which will be discussed later. U^* denotes the set of all sequences over U and ε is an empty sequence (i.e., sequence of length 0). For some relation R over U (i.e., $R \subset U \times U$), R^* is the transitive closure of R and R^{-1} is the inverse of R. A function f from set A to set B is denoted as $f : A \to B$, and $rng(f)$ is the range of f.

van der Aalst (2003)'s work on workflow model and workflow inheritance can be divided into three parts. The first part models the static factors of workflow including tasks, orders, and sometimes participants which involve in an inter-organizational workflow. However, it does not include the state of a workflow. The second part is concerned with the soundness standard of a workflow. The third part focuses on the workflow inheritance concept. The latter two parts concentrate on the state factor of a workflow model, which shows the state evolution of a workflow from beginning to the end.

3.1.1. Modeling The Static Factors of A Workflow

As explained, the research in this chapter is based on van der Aalst (2003)'s work. All definitions given here are based on those in his paper. Firstly, a formal definition of labeled P/T-net is presented.

Definition 1 (Labeled P/T-net). *A labeled Place/Transition net is a tuple* (P,T,M,F,λ) *where:*

1. $P \subseteq U$ *is a finite set of places,*
2. $T \subseteq U$ *is a finite set of transitions such that* $P \cap T = \phi$,
3. $M \subseteq L_v$ *is a finite set of methods such that* $M \cap (P \cup T) = \phi$
4. $F \subseteq (P \times T) \cup (T \times P)$ *is a set of directed arcs, called the flow relation, and*
5. $\lambda: T \to M \cup \{\tau\}$ *is a labeling function.*

In a labeled P/T-net, elements of $P \cup T$ are referred to as nodes. A node $x \in P \cup T$ is called an input node of another node $y \in P \cup T$ if and only if there exists a directed arc from x to y, denoted by xFy. Node x is called an output node of y if and only if there exists a directed arc from y to x. The set of all input nodes of some node x is called preset of x, denoted by $\bullet x$ and the set of all output nodes of some node x is called post set of x, denoted by $x \bullet$.

To recap what has been described in chapter 2, a transition is represented by a square and a place is represented by a circle in a graph model. Figure 10 shows a workflow between a customer and a producer. It can be seen that an 'i' place and an 'o' place represents the start and the end of the workflow respectively. In Figure 10, the "order" place is an output node of the "send order" transition, and an input node of the "receive order" transition. It is also the

only element in the preset of the transition of "receive order". Similarly, it is easy to see that transition "receive order" is the only transition in the post set of "order".

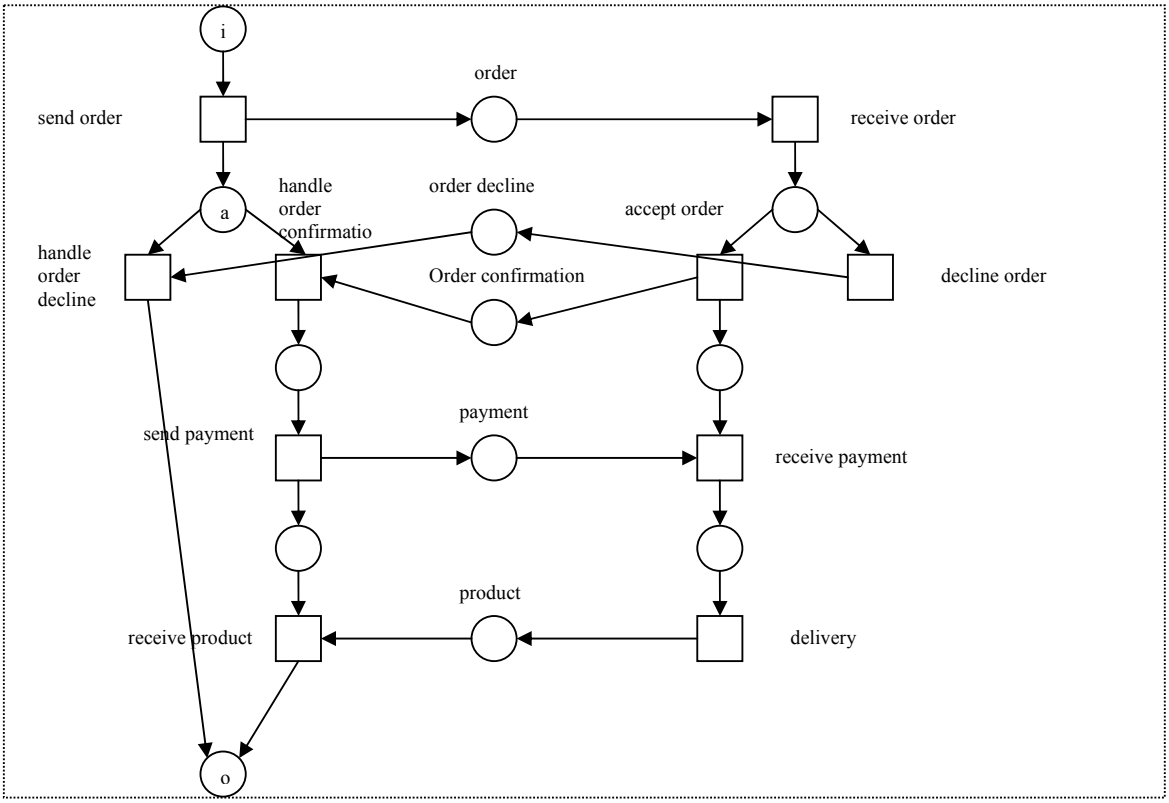

Figure 10. An example of labelled P/T-net.

The following connectedness definition is needed to define the workflow net (WF-net), which is a workflow model based on labeled P/T-net. Generally speaking, a connected labeled P/T-net means there exists a path between its every two nodes.

Definition 2 (Connectedness). *A labeled P/T-net* $N = (P, T, M, F, \lambda)$ *is weakly connected, or simply connected, if and only if, for every two nodes* x *and* y *in* $P \cup T$, $x(F \cup F^{-1})^* y$. *Net* N *is strongly connected if and only if, for every two nodes* x *and* y *in* $P \cup T$, $xF^* y$.

The concept of WF-net is used to model a workflow. Every WF-net has one unique input place and one unique output identifying the start and the end of a workflow separately. Figure 10 is an example of WF-net. The following definition formally defines the WF-net.

Definition 3 (WF-net). *Let* $N = (P, T, M, F, \lambda)$ *be a labeled P/T-net. Net* N *is a workflow net (WF-net) if and only if the following conditions are satisfied:*

1. *instance creation:* P *contains an input (source) place* $i \in U$ *such that* $\bullet i = \phi$,
2. *instance completion:* P *contains an output (sink) place* $o \in U$ *such that* $o\bullet \in \phi$,
3. *connectedness:* $N' = (P, T \cup \{t'\}, M, F \cup \{(o,t'),(t',i)\}, \lambda \cup \{(t',\tau)\})$ *is strongly connected* $(t' \notin T)$,
4. *method use:* $M = rng(\lambda) \setminus \{\tau\}$,
5. *visible start: for any* $t \in T$ *such that* $t \in i\bullet : \lambda(t) \in L_v$, *and*
6. *visible end: for any* $t \in T$ *such that* $t \in o\bullet : \lambda(t) \in L_v$.

The following symbols are used to identify the (unique) input/output place, the set of start/stop transitions and a WF-net without source and sink place. These symbols will be used in the remainder sections of this chapter.

Definition 4 (source, sink, start, stop, strip). *Let* $N = (P, T, M, F, \lambda)$ *be a WF-net.*

1. $Source(N)$ *is the (unique) input place* $i \in P$ *such that* $\bullet i = \phi$,
2. $Sink(N)$ *is the (unique) output place* $o \in P$ *such that* $\bullet o = \phi$,
3. $Start(N) = \{t \in T \mid i \in \bullet t\}$ *is the set of start transitions,*
4. $Stop(N) = \{t \in T \mid i \in t\bullet\}$ *is the set of stop transitions, and*

5. $Strip(N) = (P', T, M, F \cap ((P' \times T) \cup (T \times P')), \lambda)$ *with* $P' = P \setminus \{source(N), \sin k(N)\}$ *is the WF-net without source and sink place.*

Referring to Figure 10 again, transition "send order" is the only start transition, and stop transitions include transitions "receive product" and "handle order decline".

To model an inter-organizational workflow, a more complicated architecture including inter-organizational workflow net (IOWF-net) and flattened IOWF-net is needed. In an IOWF-net, firstly, every private workflow is modeled as a single WF-net. Then a number of channels are used to model the connections among the private workflows. Figures 11, 12 and 13 show an example of IOWF-net. Figure 11 shows the channels between a customer and a producer. The subflows of the customer and the producer are illustrated in figures 12 and 13 respectively.

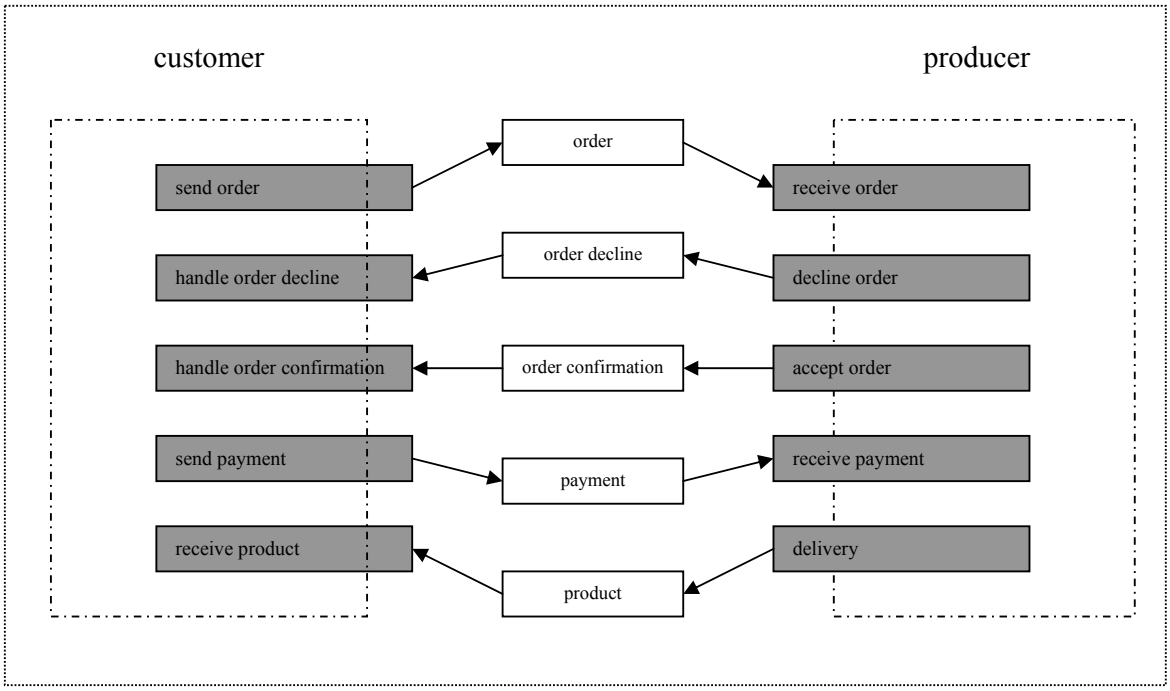

Figure 11. Channels between a customer and a producer.

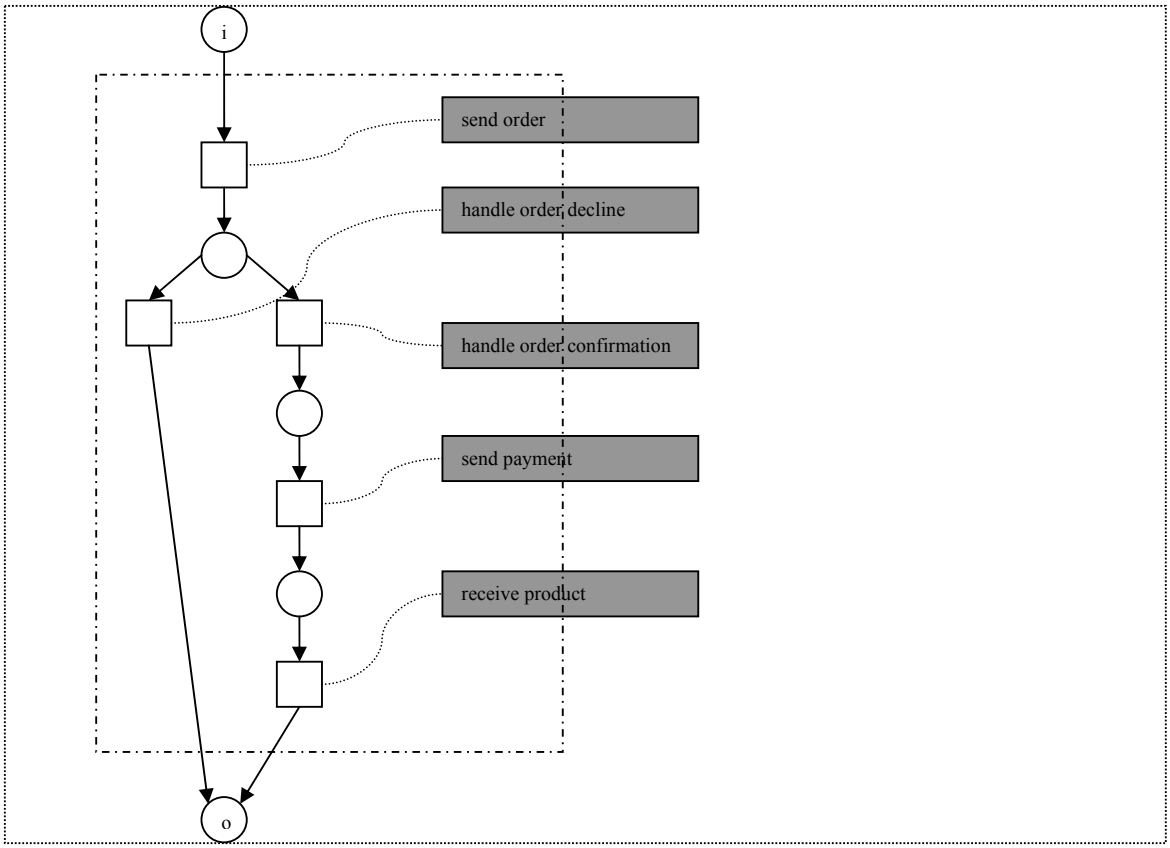

Figure 12. Sub-workflow of customer.

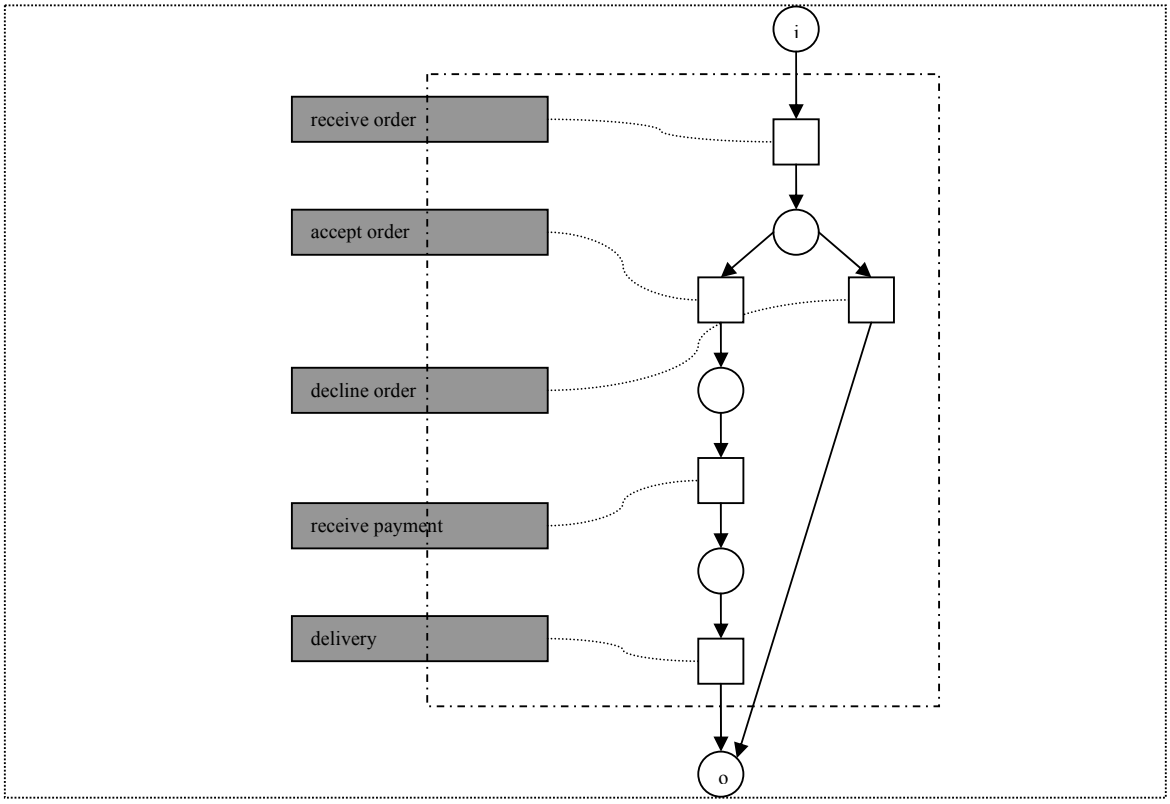

Figure 13. Sub-workflow of producer

Definition 5 (IOWF-net). *An inter-organizational workflow net (IOWF-net) is a tuple* $(C, n, N_1, ..., N_{n-1}, G)$ *where:*

1. $C \subseteq U$ *is a finite set of channels,*
2. N_0, N_1, ..., N_{n-1} *are* n *labelled P/T-nets such that:*
   a. $\forall k : 0 \le k < n : N_k = (P_k, T_k, M_k, F_k, \lambda_k)$
   b. $\forall k,l : 0 \le k < l < n : (P_k \cup T_k \cup M_k) \cap (P_l \cup T_l \cup M_l) = \phi$, *and*
   c. $\forall k : 0 \le k < n : (P_k \cup T_k \cup M_k) \cap C = \phi$,
3. $M = (\cup k : 0 \le k < n : M_k)$ *is the union of methods, and*
4. $G \subseteq (C \times M) \cup (M \times C)$ *is a set of directed arcs, called the channel flow relation.*

From the above definition, it is easy to see that only observable transitions have connections with other WF-nets.

The concept of a flattened IOWF-net is needed because an IOWF-net should be able to transfer into a general WF-net to make it possible to apply general workflow evaluation criteria such as soundness in inter-organizational workflows. Basically, a flat function transfers an IOWF-net into a general WF-net by adding a place for each channel and removing redundant source and sink places. To illustrate, Figure 10 is the corresponding flattened IOWF-net of the IOWF-net in Figure 11. Formally, flattened IOWF-net is defined as follows.

Definition 6 (*flat*(Q)***)***. *Let* $Q = (C, n, N_1, ..., N_{n-1}, G)$ *be an IOWF-net,* $N = (P, T, M, F, \lambda)$ *is the such that:*

1. $P = C \,\Upsilon\, (\cup k : 0 \le k < n : P_k)$,
2. $P_i = \{Source(N_k) \mid 0 \le k < n\}$,
3. $P_o = \{Sink(N_k) \mid 0 \le k < n)$,
4. $T = (\cup k : 0 \le k < n : T_k)$,
5. $M = (\cup k : 0 \le k < n : M_k)$,
6. $\lambda = (\cup k : 0 \le k < n : \lambda_k)$, *and*
7. $F = (\cup k : 0 \le k < n : F_k) \,\Upsilon\, \{(p,t) \in P \times T \mid (p, \lambda(t)) \in G\} \,\Upsilon\, \{(t,p) \in T \times P \mid (\lambda(t), p) \in G\}$.

Let $N' = (P', T, M, F', \lambda)$ *be the labeled P/T-net obtained by removing all places* $X = \{p \in P_i \mid {}^{N}\bullet(p\,{}^{N}\bullet) \neq \{p\}\} \,\Upsilon\, (p \in P_o \mid ({}^{N}\bullet\, p){}^{N}\bullet \neq \{p\}\}$, *i.e.,* $P' = P \setminus X$ *and* $F' = F \,\mathrm{I}\, ((P' \times T) \,\Upsilon\, (T \times P'))$. $flat(Q) = N'$ *is the flattened IOWF-net.*

3.1.2. Soundness of A Workflow

The definitions in the last section dealt with the static factors of a workflow model. To show the evolution of a workflow from instance creation to completion, the concept of state is important. In a P/T-net model, the state is represented by the distribution of tokens over the places. A workflow model that includes state is called the marked, labeled P/T net and it is defined as follows.

Definition 7 (Marked, labeled P/T-net). *A marked, labeled P/T-net is a pair* (N, s), *where* (P, T, M, F, λ) *is a labeled P/T-net and s is a bag over* P *denoting the marking (also called state) of the net. The set of all marked, labeled P/T-nets is denoted as* N.

Here, for some bag X over alphabet A and $a \in A$, $X(a)$ denotes the number of occurrences of a in X. The set of all bags over alphabet A is denoted as $\beta(A)$. For example, $[a^3, b^2, c^2]$ denotes the bag with three elements a, two elements b, and two elements c.

Before soundness can be defined, the following definitions in Table 1 are needed. Most of these definitions described how one marked and labeled P/T-net can evolve into another by firing one or one sequence of transitions. Generally speaking, before firing a transition, a transition needs to be enabled. A transition is enabled when each of its input places contains a token. Firing a transition eliminates one token on each of its input places and adds one token on each of its output places.

Please see print copy for Table 1

Table 1. Definitions needed to define Soundness. (Source: van der Aalst, 2003, p10-13)

Refering to Figure 10 again, when the "i" place contains one token, the "send order" transition is enabled. After firing of this transition, there is no token on place i, but one token on each of its output places including "order" and "a". Soundness is formally defined as follow.

Definition 8 (soundness). *A WF-net with* $Source(N) = i$ *and* $Sink(N) = o$ *is said to be weakly sound if and only if the following conditions are satisfied:*

1. *safeness:* $(N, [i])$ *is safe,*
2. *proper completion: for any reachable marking* $s \in |N, [i]\rangle$, $o \in s$ *implies* $s = [o]$,
3. *completion option: for any reachable marking* $s \in |N, [i]\rangle$, $[o] \in |N, s\rangle$.

N *is said to be strongly sound, or simply sound, if and only if, in addition there are no dead transitions, i.e.,* $(N, [i])$ *contains no dead transitions. The set of all (strongly) sound WF-net is denoted by* ω.

Definition 8 given above is the formal definition of soundness which is informally discussed in chapter 2. Soundness is the standard used in this research to evaluate a workflow. An unsound workflow is unacceptable in real life application. In this research, we only consider the concept of weakly soundness. The set of all sound WF-nets is denoted W. The following definition gives a function to obtain all observable methods from a marked and labelled P/T-net.

Definition 9 (Alphabet operator α). *Let (N,s) be a marked, labeled P/T-net in N, with $N=(P,T,M,F,\lambda)$. $\alpha:\mathrm{N}\to P(L_v)$ is a function such that $\alpha(N,s)=\{\lambda(t)\mid t\in T\wedge\lambda(t)\neq\tau\}$ and t is not dead.*

When a sub-workflow (also known as subworkflow or subflow) is used in an inter-organizational workflow, it is expected that it should behave as it is executed alone. However, it is possible that when a subflow is activated, there are still tokens on its places. Note that this scenario is not possible if a subflow is executed alone. In an inter-organizational workflow environment, scenarios such as this makes a workflow more difficult to analyse, sometimes even results in undesirable characteristics such as deadlock. To prevent the happening of such scenarios, an activation safeness concept is needed. Basically, an activated-safe subflow in a marked and labelled P/T-net is activated only when there is no token on any of its places. This concept of activation safeness is formally defined as follow.

Definition 10 (Activation safeness). *Let (N,s) be a marked, labeled P/T-net in N, where $N=(P,T,M,F,\lambda)$. A subset of places $P'\subseteq P$ is activation safe in (N,s) if and only if for any reachable state $s'\in[N,s\rangle$, any transition $t\in{\bullet}P'\backslash P'{\bullet}$, and any place $p\in P':(N,s')[t\rangle$ implies $s'(p)=0$.*

From the definition above, a set of places P' is activation safe if all transitions producing tokens for P' but not consuming tokens from P' are not enabled as long as there are tokens in P'. Using Figure 10 as an example, when P' is the set of places belong to the producer's sub-workflow in Figure 13, the transition producing tokens for P' but not consuming tokens from P' is the transition "receive order". As the set of places P' is activated only when transition such as "receive order" is enabled, thus it is guaranteed that when a sub-workflow

$N'=(P',T',M',F',\lambda')$ is involved in an inter-organizational workflow represented by an IOWF-net Q, then for any states s of Q, there is a corresponding state s' on the places of P' such that $s' \in |N',[i]\rangle$. That is, when a sub-workflow is used by an inter-organizational workflow, any possible state represented by the distribution of the tokens over its places is also reachable when the sub-workflow is executed alone.

Using the concept of activation safeness, the soundness concept for IOWF-net can be formally defined. Such a sound IOWF-net does not just respond to a general sound WF-net, in fact, every subflow of it needs to be sound and be activated-safe too.

Definition 11 (IOWF-net Soundness). *Let* $Q=(C,n,N_1,...,N_{n-1},G)$ *be an IOWF-net and let* $N=(P,T,M,F,\lambda)$ *be the corresponding flattened net without dead transitions, i.e.,* $N=flat(Q)$. *Q is sound if and only if*

1. $(\forall k: 0 \le k < n: N_k \in \omega)$, *i.e., all subflows are sound,*
2. $N \in W$, *i.e., the flattened IOWF-net is a sound WF-net, and*
3. $\forall k: 0 \le k < n: P_k \setminus \{source(N_k), \sin k(N_k)\}$ *is activation safe in* $(N,[i])$, *i.e., there no multiple activation.*

3.1.3. Branching Bisimilarity and Workflow Inheritance

This section discusses the concepts of branching bisimilarity and workflow inheritance (van der Aalst 2003), which are important concepts in workflow analysis and will be further discussed in the next section when the proposed new local criterion is presented.

Table 2 shows three definitions which are used to define branching bisimilarity. Branching bisimilarity refers to a relation between two marked labelled P/T-net. Generally speaking, if two marked and labelled P/T-nets are branching bisimilar, it means that anyone of them can simulate every step of the observable behaviour of the other after executing zero or more silent actions. Several revised versions of branching bisimilarity can be found in van Glabbeek & Weijland (1996).

Please see print copy for Table 2

Table 2. Definition needed to define branching bisimilarity. (Source: van der Aalst, 2003, p 15-16)

According to Definition A in Table 2, for a marked and labeled Petri net $p \in \mathrm{N}$, $\downarrow p$ means it is in a state when only one token is on the unique output place. Definition B is used to demonstrate that a marked labeled P/T-net can evolve into another marked, labeled P/T-net by executing a sequence of zero or more silent actions. For example, when one sees $p \Rightarrow p'$, one can say that a marked labeled P/T-net p can evolve into another marked labeled P/T-net p' by executing a sequence of zero or more silent actions. Definition C is used to demonstrate a marked labeled P/T-net can evolve into another marked labeled P/T-net with only one token

on the output place by executing a sequence of zero or more silent actions. For example, $\Downarrow p$ means that marked labeled P/T-net p can reach the termination state after a firing sequence containing zero or more silent actions.

Let p and p' be two marked, labeled P/T nets, $p\left|(a)\right\rangle p'$ means $(a=\tau \wedge p=p') \vee (p\left|a\right\rangle p'$. Therefore, for a none τ action, $p\left|(a)\right\rangle p'$ is the same as $p\left|a\right\rangle p'$. But for a silent action, $p\left|(a)\right\rangle p'$ means that zero or one silent action is executed. This symbol will be used in the definition of branching bisimilarity to be given below.

Definition 12 (Branching bisimilarity). *A binary relation* $R \in \mathrm{N} \times \mathrm{N}$ *is called a branching bisimulation if and only if, for any* $p, p', q, q' \in \mathrm{N}$ *and* $a \in L$,

1. $pRq \wedge p\left|a\right\rangle p'$ □

$$\exists q', q'' : q', q'' \in \mathrm{N} : q \Rightarrow q'' \wedge q''\left|(a)\right\rangle q' \wedge pRq'' \wedge p'Rq'),$$

2. $pRq \wedge q\left|a\right\rangle q'$ □

$$\exists p', p'' : p', p'' \in \mathrm{N} : p \Rightarrow p'' \wedge p''\left|(a)\right\rangle p' \wedge p''Rq \wedge p'Rq'), \text{ and}$$

3. pRq □

$$(\downarrow p \text{ □ } \Downarrow q \text{ □} \downarrow q \text{ □} \Downarrow p).$$

Two marked, labeled P/T-nets are called branching bisimilar, denoted by $p \sim_b q$, *if and only if there exists a branching bisimulation* R *such that* pRq.

Figure 14 shows a scenario of branchinig bisimilarity and how it works in the inter-organizational workflow. The marked WF-nets in figures 14 and 12 are said to be branching bisimilar. It can be seen that several silent transitions represented by τ are added in Figure 14

comparing with Figure 12. The marked WF-net in Figure 14 can still simulate any behaviour of the marked WF-net in Figure 12. For instance, after firing transition “send order”, it is possible that a sequence of silent transitions is fired. However, after any firing sequence of silent transitions, the marked WF-net in Figure 14 can still choose to fire transition “handle order decline” or “handle order confirmation” after firing a sequence of silent transitions, i.e., it can still execute any possible step that the marked WF-net in Figure 12 executes after firing the transition “send order”. As the customer is not able to know if the producer will confirm or decline the order after an order is sent, it is necessary for its workflow to be able to handle both responses even after firing a number of silent transitions. This is the major reason why branching bisimilarity is used in inter-organizational workflow. It can be seen that after replacing the customer workflow in Figure 12 with the workflow in Figure 14, which is branching bisimilar to that of Figure 12, the IOWF-net in Figure 11 is still sound. Such result is generalized and proven to be correct in van der Aalst (2003)’s paper.

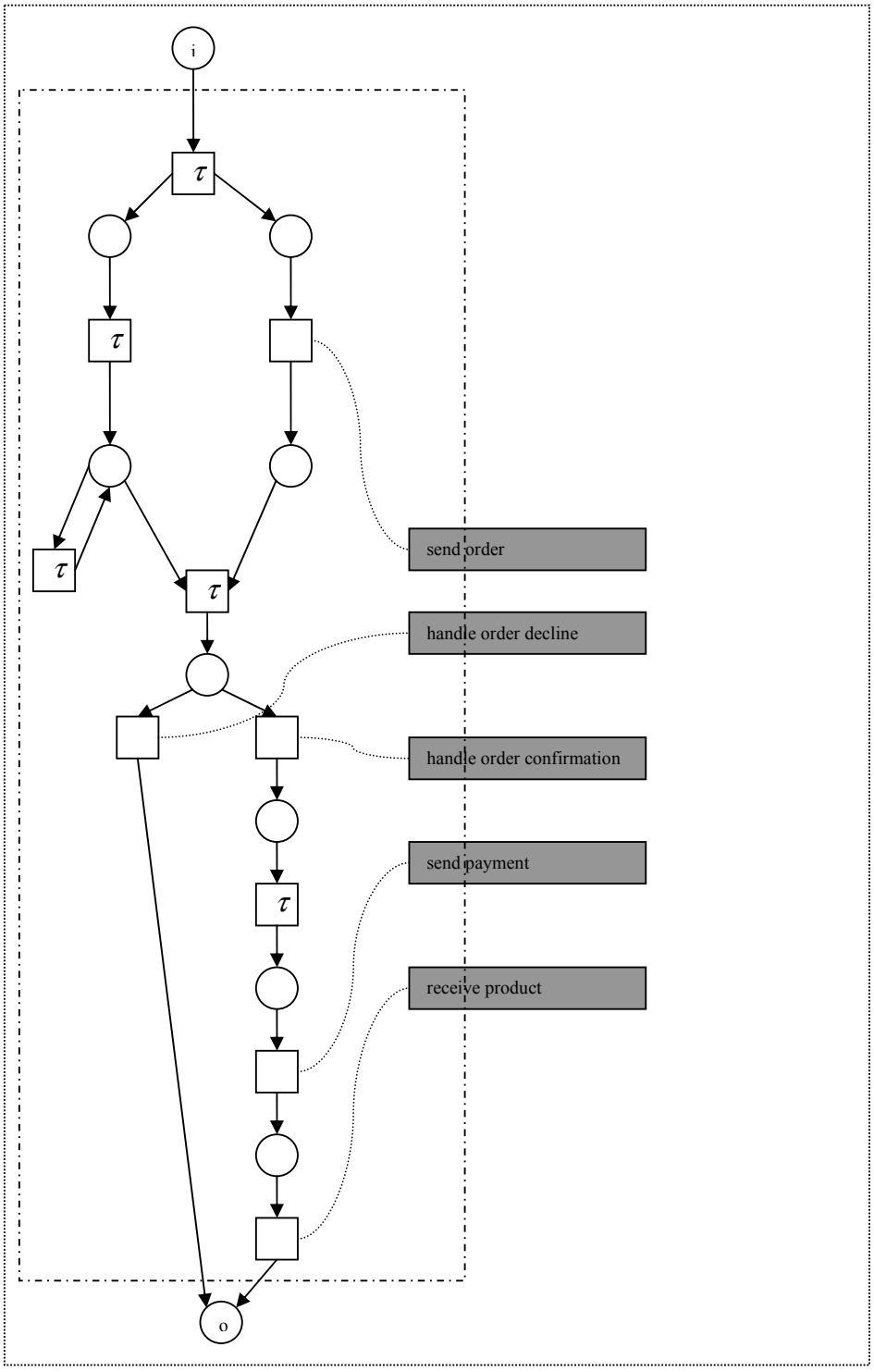

Figure 14. An alterative customer sub-workflow.

Note that the conditions in the above definition require that after executing any number of silent actions, i.e., $q \Rightarrow q''$, as long as there is no observable transition is fired, we can still get

a marked, labelled P/T-nets q'' which are branching bisimilar with the original state of the other marked, labelled P/T-net, i.e., pRq'', and at the same time pRq. In the next section, we will show that under some specific circumstances, this requirement can become too restrictive in the inter-organizational workflow. Therefore, a revised version of branching bisimilarity with weaker requirement is proposed in the next section.

The branching bisimilarity concept can be used to defined a behavioural equivalence relation between WF-nets. According to van der Aalst (2003), two WF-nets are behavioural equivalent if they can simulate each other through the evolution process from instance creation to completion.

Definition 13 (Behavioral equivalence of WF-net). *For any two WF-nets* N_0 *and* N_1 *in* ω, $N_0 \cong N_1$ *if and only if* $(N_0,[i]) \sim_b (N_1,[i])$.

Before the concept of workflow inheritance can be given, we first need to define an abstraction operator to hide the new transitions added in the subclass workflow. Basically, what the abstraction operator does is to change a set of transitions into silent transitions.

Definition 14 (Abstraction). *Let* $N = (P,T,M,F,\lambda_0)$ *be a labeled P/T-net. For any* $I \subseteq L_v$, *the abstraction operator* τ_I *is a function that renames all transition labels in* I *to the silent action* τ. *Formally,* $\tau_1(N) = (P,T,M,F,\lambda_1)$ *such that, for any* $t \in T$, $\lambda_0(t) \in I$ *implies* $\lambda_1(t) = \tau$ *and* $\lambda_0(t) \notin I$ *implies* $\lambda_1(t) = \lambda_0(t)$.

According to van der Aalst (2003), there are four notions of workflow inheritance including projection inheritance, protocol inheritance, life-cycle inheritance and protocol/projection heritance. However, only projection inheritance is used in the P2P approach. Basically, a workflow is the subclass of another workflow if after changing one set of its transitions into silent transitions, these two workflows are branching bisimilar.

Definition 15 (Inheritance). *For any two (weakly) sound WF-nets* N_0 *and* N_1 *in* ω, N_1 *is a subclass of* N_0 *under projection inheritance, denoted by* $N_1 \leq_{pj} N_0$, *if and only if there is an* $I \subseteq L_v$ *such that* $(\tau_I N_1, [i]) \sim_b (N_0, [i])$.

In the next section, a revised version of WF-net equivalence is proposed in this research as the new local criteria of the P2P approach.

3.2. An Improved Local Criteria for More Flexible Inter-organizational Workflow

This research proposes a new local criteria concept for the P2P approach. Example will be provided to show that this new local criterion allows more flexible workflow modification. Furthermore, two theorems will be given to prove that this local criterion will result in a globally sound inter-organizational workflow. To formulate this new local criterion, several new definitions are needed. Firstly the concept of WF-net weakly equivalence will be defined below. This concept uses a revised version of weakly bisimilarity on WF-nets. The concept of weakly bisimulation is proposed by Milner (1980).

Definition 16 (Weakly equivalence of WF-nets). *For any two WF-nets* $N_1 = (P_1, T_1, M_1, F_1, \lambda_1)$ *and* $N_2 = (P_2, T_2, M_2, F_2, \lambda_2)$, N_2 *is called to be weak equivalent to* N_1, *denoted* $N_2 \cong_W N_1$, *if and only if there exists two relation* $R_1 \in (N_2, s') \times (N_1, s), R_2 \in (N_1, s) \times (N_2, s')$ *such that for any* $p = (N_1, s_1)$, $p'' = (N_1, s_1'')$, $p' = (N_1, s_1')$, $q = (N_2, s_2)$, $q'' = (N_2, s_2'')$ *and* $q' = (N_2, s_2')$,

1. $(N_1, [i]) R_2 (N_2, [i])$ *and* $(N_2, [i]) R_1 (N_1, [i])$,
2. $qR_1 p \wedge p|a\rangle p' \wedge a \in L_v \;\square\; \exists q', q'' : q \Rightarrow q'' \wedge q''|a\rangle q' \wedge q' R_1 p'$,
3. $qR_1 p \wedge p|a\rangle p' \wedge a = \tau \;\square\; \exists q' : q \Rightarrow q' \wedge q' R_1 p'$,
4. *For any* $q \in |N_2, [i]\rangle$, $\exists p$ *such that* $qR_1 p$ *or* $\exists p', q', \sigma \in T^* : q|\sigma\rangle q' \wedge q' R_1 p'$,
5. $pR_2 q \wedge q|a\rangle q' \wedge a \in L_v \;\square\; \exists p'', p' : p \Rightarrow p'' \wedge p''|a\rangle p' \wedge p' R_2 q'$,
6. $pR_2 q \wedge q|a\rangle q' \wedge a = \tau \;\square\; \exists p' : p \Rightarrow p' \wedge p' R_2 q'$.

The concept of weakly equivalence is concerned with an old WF-net N_1 and a new WF-net N_2. Two relations R_1 and R_2 between the states of the old and new WF-net are involved in the definition. Requirement 1 states that there exists relations R_1 and R_2 when N_1 and N_2 are both in the state that only one token is on the 'i' place. Requirements 2, 3 and 4 deal with the first relation R_1, which demonstrates the ability of the new WF-net to simulate the behavior of the old WF-net. Requirement 2 and 3 state that before executing any number of silent transitions, the new workflow can reach a corresponding state of the old workflow regardless of whether an observable or silent transition is fired in the old workflow. Requirement 4 requires that, for an arbitrary state of the new WF-net, either there is a corresponding state of the old WF-net which it can simulate, or it can reach a state that can simulate any observable behavior of the old workflow by executing a sequence of transitions.

Since $(N_2,[i])R_1(N_1,[i])$ and the new WF-net N_2 can simulate any observable behavior of the old WF-net N_1, for any state of N_1, there is a corresponding state in N_2 under the relation R_1. But for arbitrary state of N_2, it is not necessary that there exists a corresponding state in N_1. Requirements 5 and 6 deal with the second relation R_2, which demonstrates the ability of the old WF-net to simulate the behavior of the new WF-net. Requirement 5 and 6 state that after firing a transition (silent or observable transition) in the new workflow, there is always a state in the old workflow corresponding to the new state of the new workflow. Since $(N_1,[i])R_2(N_2,[i])$ and the old WF-net N_1 can simulate any observable behavior of the new WF-net N_2, for any state of N_2, there is a corresponding state in N_1 under the relation R_2. But for arbitrary state of N_1, it is not necessary that there exist a corresponding state in N_2.

Comparing with branching bisimulation and behavioral equivalence, weakly equivalence is a more flexible equivalence relation. For instance, for a state of the new WF-net q such that qR_1p, it can evolve into another state q'' after executing a sequence of silent action. The concept of weak equivalence allows the new state q'' does not simulate all the possible behaviors of the corresponding marked, labeled P/T-net p, which is not permitted in branching bisimulation.

Figures 15 and 16 show two alternative workflows for the customer and producer scenario presented in figures 10 to 13. The workflow in Figure 12 and Figure 15 are weakly equivalent, and the workflow in Figure 13 and Figure 16 are also weakly equivalent. Both these two pairs of WF-nets are not branching bisimilar because after the exection of the 'decline report' transition, workflows in figures 15 and 16 can no longer simulate all possible behaviors of those in figures 12 and 13.

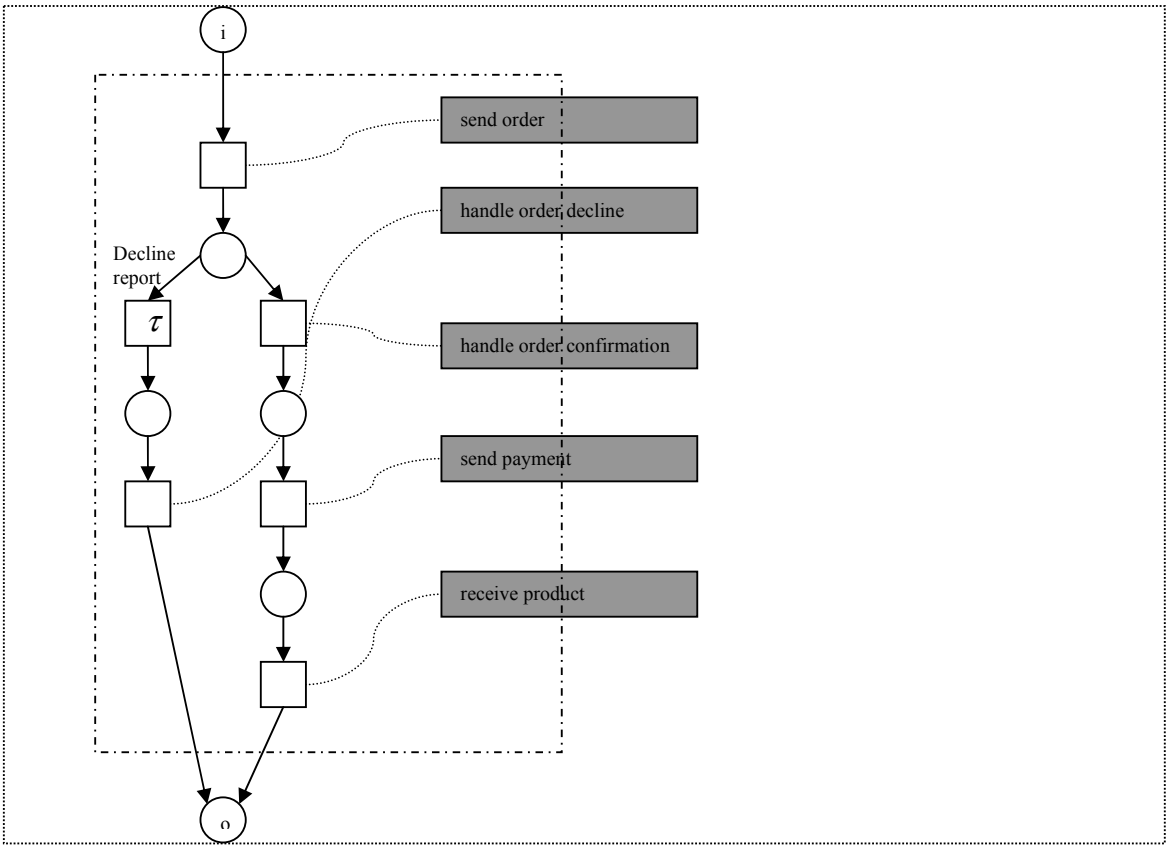

Figure 15. An alternative customer workflow

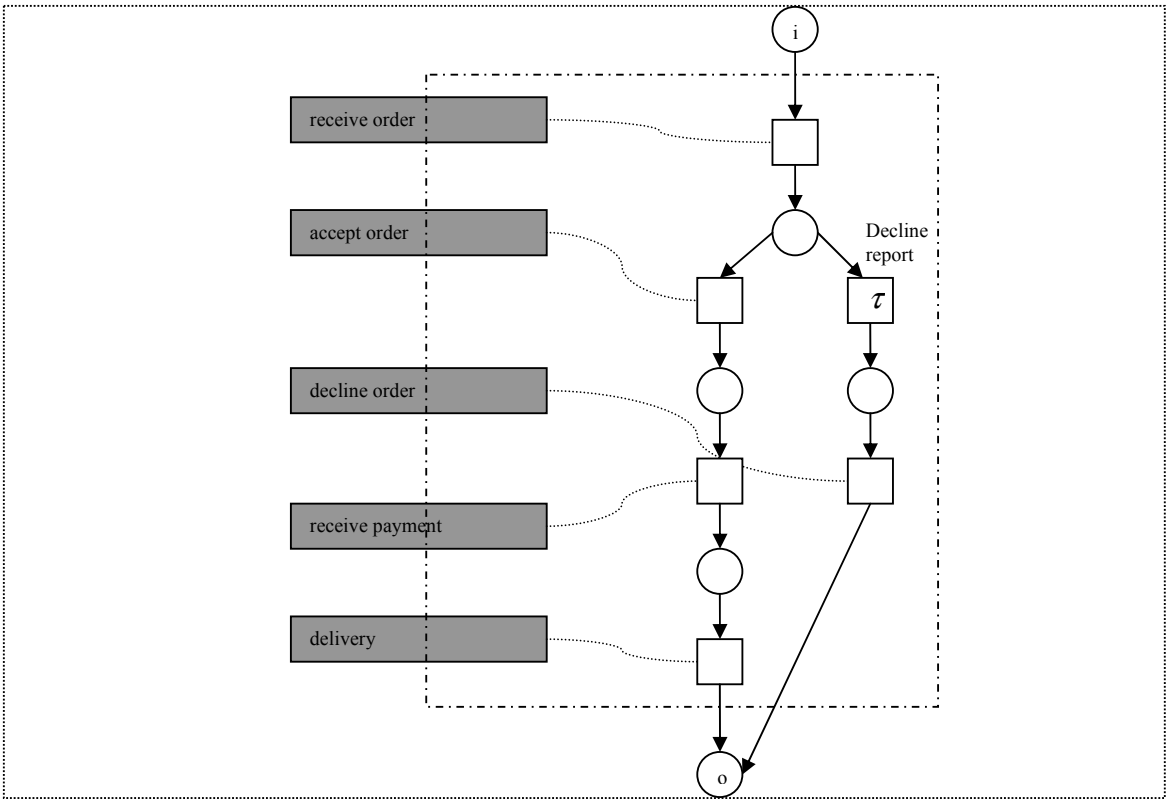

Figure 16. An alternative producer workflow

From Figure 16 and Figure 13, it can be seen that after the execution of the "decline report" transition in Figure 16, the workflow can no longer simulate all possible behaviors after executing the "receive order" transition in Figure 13. However, comparing the same two states in these two figures, we can find that the workflow in Figure 13 can still simulate all possible behaviors after firing the "decline report" transition. This is the reason why two different relations are needed in Definition 16.

Next we will discuss the feasibility to use the WF-net weakly equivalence concept as a local criterion in the inter-organizational workflow by using examples in figures 15 and 16 to illustrate. Recalled that in the previous sections, the P2P approach states that if for each organizational entity, a private workflow is designed such that it is branching bisimilar to the

relevant part of the public workflow after changing the new transitions to be silent, the resultant inter-organizational workflow is guaranteed to be sound. Part of the reasons why behavioral equivalent and branching bisimilarity are used in the P2P approach is to prevent the situation shown in Figure 15. Let us compare it with Figure 12, it can be seen that in Figure 15, a new "decline report" transition is added in the workflow of the customer, which makes the customer workflow in Figure 15 not branching bisimilar to the customer workflow in Figure 12 after changing the new transitions to be silent. Such modification also makes the whole workflow combined with the sub-workflows in figures 15 and 13 unsound. In Figure 15, the "decline report" transition can be fired before the distributor sends a response message back. In such a state, the workflow of the customer is not able to handle all possible responses the producer sends back. If an "order confirmation" message is received after the transition "decline report" is fired, the entire workflow is in deadlock. Therefore, WF-net weakly equivalence concept is not able to function as a local criterion in the P2P approach.

However, a seemingly similar modification in the producer workflow does not affect the soundness of the entire workflow, as demonstrated in Figure 16. It can be seen that similar to the workflow of Figure 15, a new "decline report" transition is added to the producer workflow in Figure 16 comparing with the workflow in Figure 13. It makes the workflow in Figure 16 not branching bisimilar to the workflow in Figure 13. However, such a modification does not make the overall workflow combined with the sub-workflows in Figure 12 and Figure 16 unsound. This is different to the situation shown in figures 15 and 13. It seems that the local criterion of branching bisimilarity and behavioral equivalence is too restrictive under this circumstance. If one takes a closer look at both figures 15 and 16, a subtle difference can be found. In Figure 15, the "decline report" is added before a transition receives message, whereas in Figure 16, the "decline report" is added before a transition sends message. This

difference indicates that the direction of the message exchanged among the sub-workflows is an important factor that needs to be considered when we attempt to propose a more flexible local criterion. The following definitions 17, 18 and 19 formulate the new local criterion on the base of the concept of WF-net weakly equivalence. Definition 17 provides a function to obtain the set of all transitions in a sequence of transitions. Definition 18 gives a function to obtain the set of transitions that receives message from other sub-workflows. These two definitions will be used in Definition 19 when the new local criterion is formally defined. As the direction of the message needs to be taken into account, the definitions 18 and 19 are formulated in an inter-organizational workflow context using the IOWF-net concept described in the last section.

Definition 17 ($con(\sigma)$ ***)****. For any* $\sigma \in T^*$ *such that* $\sigma = t_1,...,t_n$, $con(\sigma) = \{t_1,...,t_n\}$.

For example, for $\sigma = t_1, t_2, t_3$, $con(\sigma) = \{t_1, t_2, t_3\}$.

Definition 18 (incoming transitions). *Let* $Q = (C, n, N_1,..., N_i,..., N_{n-1}, G)$ *be an IOWF-net such that* $N_1,..., N_{n-1}$ *are* $n-1$ *labeled P/T nets.* $\forall k : 0 \le k < n : N_k = (P_k, T_k, M_k, F_k, \lambda_k)$. *For* $i : 0 \le k < n : N_i = (P_i, T_i, M_i, F_i, \lambda_i)$, $inc(N_i) = \{t \mid t \in T_i \wedge \exists c \in C : (c, \lambda_i(t)) \in G\}$, *is called the set of incoming transitions of* N_i, *denoted* $inc(N_i)$.

To illustrate the concept of incoming transitions, consider the IOWF-net combined with the sub-workflows in figures 12 and 16. The set of incoming transitions of the producer workflow includes the transitions ‘receive order’ and ‘receive payment. The following definition formally defines the concept of limited equivalence proposed in this research.

Definition 19 (limited equivalence of WF-nets in IOWF net). *Let* $Q=(C,n,N_1,...,N_i,...,N_{n-1},G)$ *and* $Q'=(C,n,N_1,...,N_i',...,N_{n-1},G)$ *be two IOWF-net such that* $N_1,...,N_i,...,N_{n-1}$ *and* N_i' *are* $n+1$ *WF-nets.* $\forall k:0\le k<n:N_k=(P_k,T_k,M_k,F_k,\lambda_k),N_i'=(P_i',T_i',M_i',F_i',\lambda_i')$. N_i' *is called to be weak equivalent to* N_i *in* Q, *denoted* $N_i'\cong_Q N_i$, *if and only if there exists two relation* $R_1'\in(N_i',s')\times(N_i,s),R_2'\in(N_i,s)\times(N_i',s')$ *such that for any* $p=(N_1,s_1)$, $p''=(N_1,s_1'')$, $p'=(N_1,s_1')$, $q=(N_2,s_2)$, $q''=(N_2,s_2'')$, $q'=(N_2,s_2')$ *and* $a\in L$,

1. $(N_i,[i])R_2'(N_i',[i])$ *and* $(N_i',[i])R_1'(N_i,[i])$,
2. $qR_1'p\wedge p|a\rangle p'\wedge a\in L_v \;\square\; \exists q',q'':q\Rightarrow q''\wedge q''|a\rangle q'\wedge q'R_1'p'$,
3. $qR_1'p\wedge p|a\rangle p'\wedge a=\tau \;\square\; \exists q:q\Rightarrow q'\wedge q'R_1'p'$,
4. *For any* $q\in|N_i',[i]\rangle$, $\exists p$ *such that* qR_1p *or* $\exists p',q',\sigma\in T^*$ *such that* $\forall t\in con(\sigma),t\notin inc(N_i')\wedge q|\sigma\rangle q'\wedge q'R_1'p'$,
5. $pR_2'q\wedge q|a\rangle q'\wedge a\in L_v \;\square\; \exists p'',p':p\Rightarrow p''\wedge p''|a\rangle p'\wedge p'R_2'q'$,
6. $pR_2'q\wedge q|a\rangle q'\wedge a=\tau \;\square\; \exists p':p\Rightarrow p'\wedge p'R_2'q'$.

Comparing with Definition 16, it can be seen that the concept of limited equivalence is defined in an IOWF-net context. Moreover, for relation R_1, only the incoming transitions are allowed to be used by the new WF-net to reach a state that can simulate all possible behaviors of the old WF-net. It is obvious that limited equivalence is a criterion that can be verified locally without the need to know other sub-workflows in the inter-organizational workflow.

Consider the IOWF-net combined with the sub-workflows in figures 12 and 13, the producer workflow shown in Figure 16 is limited equivalent with the workflow shown in Figure 13. However, the customer workflow shown in Figure 15 is not limited equivalent with the workflow shown in Figure 12. This is because after the execution of the transition 'decline report', the workflow in Figure 15 is in a state where there is no corresponding state in the old customer workflow shown in Figure 12 under the relation R_2. Moreover, it needs to fire the transition 'handle order decline', which is an incoming transition, to reach a state that there is a corresponding state in the old customer workflow under the relation R_2. Obviously, it violates requirement 4 of Definition 19.

To apply the new local criterion in the inter-organizational workflow design, we need to prove that after changing one or more sub-workflows of the overall workflow into one that is of limited equivalent, the overall workflow will remain sound. The following two theorems have been proposed in this research to show that this new local criterion does not affect the soundness of the overall workflow. Theorem 1 states that under certain conditions, one sub-workflow can be replaced by a workflow which is limited equivalent with the original one without affecting the soundness of the overall workflow. Theorem 2 shows that after changing all sub-workflows into the limited equivalent ones, the entire inter-organizational remain sound. The proof of Theorem 2 is based on the result of Theorem 1.

Theorem 1. *Let* $Q_1 = (C, 2, N_a, N_b, G)$ *and* $Q_2 = (C, 2, N_a, N_c, G)$ *be two IOWF-net such that*

1. $N_a = (P_a, T_a, M_a, F_a, \lambda_a)$, $N_b = (P_b, T_b, M_b, F_b, \lambda_b)$, $N_c = (P_c, T_c, M_c, F_c, \lambda_c)$ *are three labeled P/T nets,*
2. N_b *and* N_c *are two sound WF-nets,*
3. $N_1 = flat(Q_1) = (P_1, T_1, M_1, F_1, \lambda_1)$, $N_2 = flat(Q_2) = (P_2, T_2, M_2, F_2, \lambda_2)$,

4. N_1 *is a sound WF-net,*

5. *In* N_1, *for any* $t \in Start(N_1) \,\mathbf{Y}\, Stop(N_1) : \lambda_1(t) \neq \tau$, *i.e., the start and stop transitions in* N_1 *are observable,*

6. *In* N_2, *for any* $t \in Start(N_2) \,\mathbf{Y}\, Stop(N_2) : \lambda_2(t) \neq \tau$, *i.e., the start and stop transitions in* N_2 *are observable,*

7. *In* N_1, *for any* $t_1, t_2 \in T_1$ *such that* $\lambda_1(t_1) \neq \tau \wedge \lambda_1(t_2) \neq \tau$, $\lambda_1(t_1) \neq \lambda_1(t_2)$, *i.e., for every two observable transition in* N_1, *they correspond to different methods,*

8. *In* N_2, *for any* $t_1, t_2 \in T_2$ *such that* $\lambda_2(t_1) \neq \tau \wedge \lambda_2(t_2) \neq \tau$, $\lambda_2(t_1) \neq \lambda_2(t_2)$, *i.e., for every two observable transition in* N_2, *they correspond to different methods,*

9. $\forall t \in Start(N_b), \lambda_b(t) \neq \tau$, $\forall t \in Start(N_c), \lambda_c(t) \neq \tau$, *and* $\{\lambda_b(t) \mid t \in Start(N_b)\} = \{\lambda_c(t) \mid t \in Start(N_c)\}$, *i.e., all start transitions in* N_b *and* N_c *are observable, and the set of start transitions in* N_b *and* N_c *coincide,*

10. $\forall t \in Stop(N_b), \lambda_b(t) \neq \tau$, $\forall t \in Stop(N_c), \lambda_c(t) \neq \tau$, *and* $\{\lambda_b(t) \mid t \in Stop(N_b)\} = \{\lambda_c(t) \mid t \in Stop(N_c)\}$, *i.e., all stop transitions in* N_b *and* N_c *are observable, and the set of stop transitions in* N_b *and* N_c *coincide,*

11. P_B *is activation safe in* $(N_1, [i])$,

12. *for* $I = \alpha(N_C) \setminus \alpha(N_B) = \alpha(N_2) \setminus \alpha(N_1), \tau_I(N_C) \cong_{Q1} N_B$,

Then

1. *for* $I = \alpha(N_C) \setminus \alpha(N_B) = \alpha(N_2) \setminus \alpha(N_1)$, $\tau_I(N_2) \cong_W N_1$ *and* P_C *is activation safe in* $(N_2, [i])$.

2. N_2 *is a sound WF-net.*

Proof.

1. To prove for $I = \alpha(N_C) \setminus \alpha(N_B) = \alpha(N_2) \setminus \alpha(N_1), N_1 \cong_W \tau_I(N_2)$, we need to prove there exists two relations $R_1 \in (N_2, s') \times (N_1, s), R_2 \in (N_1, s) \times (N_2, s')$ consistent with the requirements in definition 16. Since for $I = \alpha(N_C) \setminus \alpha(N_B) = \alpha(N_2) \setminus \alpha(N_1)$, $\tau_I(N_C) \cong_{Q1} N_B$, there exists two relations $R_1' \in (N_C, s') \times (N_B, s), R_2' \in (N_B, s) \times (N_C', s')$ satisfying the requirements in Definition 19.

We prove the existence of R_2 firstly. R_2 is defined as follows: $R_2 = \{((N_1, s), \tau_I(N_2, s')) \mid s \in |N_1, [i]\rangle \wedge s = s_A + s_B \wedge s' = s_A + s_C \wedge s_A \in \beta(P_1 \setminus P_B) \wedge s_B \in \beta(P_B) \wedge s_C \in \beta(P_C) \wedge s_B \in |N_B, [i]\rangle \wedge s_C \in |N_C, [i]\rangle \wedge ((s_B = 0 \wedge s_C = 0) \vee (N_B, s_B) R_2' (\tau_I N_C, s_C))\}$. It is easy to see that $(N_1, [i]) R_2 \tau_I(N_2, [i])$. It remains to be shown that

- $(N_1, s_1) R_2 (\tau_I N_2, s_2) \wedge (N_2, s_2) |\lambda(t)\rangle (N_2, s_2') \wedge \lambda(t) \in L_v$ □ $\exists (N_1, s_1''), (N_1, s_1') : (N_1, s_1) \Rightarrow (N_1, s_1'') \wedge (N_1, s_1'') |\lambda(t)\rangle (N_1, s_1') \wedge (N_1, s_1') R_2 \tau_I (N_2, s_2'))$
- $(N_1, s_1) R_2 \tau_I (N_2, s_2) \wedge (N_2, s_2) |\lambda(t)\rangle (N_2, s_2') \wedge \lambda(t) = \tau$ □ $\exists (N_1, s_1') : (N_1, s_1) \Rightarrow (N_1, s_1') \wedge (N_1, s_1') R_2 (N_2, s_2')$.

Let $s_1 = s_A + s_B$, $s_2 = s_A + s_C$, $s_1' = s_A' + s_B'$, $s_2' = s_A' + s_C'$, $s_1'' = s_A'' + s_B''$ and $s_2'' = s_A'' + s_C''$. The following four situations are considered.

Assume $t \in T_A$, since the transitions in T_A are only affected by places in $P_1 \setminus P_B = P_2 \setminus P_C$, and s_1 and s_2 share the same state s_A on $P_1 \setminus P_B = P_2 \setminus P_C$, t is also enabled in (N_1, s_1). Because firing a transition in T_A only affects places in $P_1 \setminus P_B = P_2 \setminus P_C$, firing t from

(N_1, s_1) lead to a same state s_1' on $P_1 \setminus P_B$ like s_2' on $P_2 \setminus P_C$. Therefore, for $(N_1, s_1) R_2 (\tau_I N_2, s_2) \wedge (N_2, s_2) \left| \lambda(t) \right\rangle (N_2, s_2') \wedge \lambda(t) \in L_v$, there exists $(N_1, s_1'') = (N_1, s_1)$ and (N_1, s_1') such that $(N_1, s_1) \Rightarrow (N_1, s_1'') \wedge (N_1, s_1'') \left| \lambda(t) \right\rangle (N_1, s_1') \wedge (N_1, s_1') R_2 \tau_I (N_2, s_2')$. For $(N_1, s_1) R_2 \tau_I (N_2, s_2) \wedge (N_2, s_2) \left| \lambda(t) \right\rangle (N_2, s_2') \wedge \lambda(t) = \tau$, there exists $(N_1, s_1') : (N_1, s_1) \left| \lambda(t) \right\rangle (N_1, s_1')$ such that $(N_1, s_1) \Rightarrow (N_1, s_1') \wedge (N_1, s_1') R_2 \tau_I (N_2, s_2')$.

Assume $t \in Start(N_C)$. From requirement 9 of the theorem, $t \in L_v$. Since transitions in $Start(N_B)$ and $Start(N_C)$ is only affected by places in $P_1 \setminus P_B = P_2 \setminus P_C$, and s_1 and s_2 share the same state on $P_1 \setminus P_B = P_2 \setminus P_C$, t is also enabled in (N_1, s_1). Because P_B is active safe in $(N_1, [i])$, when t in $Start(N_B)$ is enabled, $s_B = 0$. According to the definition of R_2, when $s_B = 0$, $s_C = 0$. Moreover, the effect of firing t from (N_1, s_1) and (N_2, s_2) result in (N_1, s_1') and (N_2, s_2') such that $s_1' = s_A' + s_B' \wedge s_2' = s_A' + s_C'$ and $(N_B, s_B') R_2' (\tau_I N_C, s_C')$. Therefore, for $(N_1, s_1) R_2 (\tau_I N_2, s_2) \wedge (N_2, s_2) \left| \lambda(t) \right\rangle (N_2, s_2') \wedge \lambda(t) \in L_v$, there exists $(N_1, s_1'') = (N_1, s_1)$ and (N_1, s_1') such that $(N_1, s_1) \Rightarrow (N_1, s_1'') \wedge (N_1, s_1'') \left| \lambda(t) \right\rangle (N_1, s_1') \wedge (N_1, s_1') R_2 \tau_I (N_2, s_2')$.

Assume $t \in Stop(N_C)$. From requirement 10 of the theorem, $t \in L_v$. And obviously $s_C \neq 0$, therefore, $(N_B, s_B) R_2' \tau_I (N_C, s_C)$ and t can be fired after execution of zero or more silent transitions from (N_1, s_1). Moreover, the effect of firing t in (N_1, s_1) and (N_2, s_2) is identical on places of $P_1 \setminus P_B = P_2 \setminus P_C$. Since N_B and N_C are both sound WF-net, for the places of P_B and P_C, after firing t, $(s_B' = 0 \wedge s_C' = 0)$. Therefore, for $(N_1, s_1) R_2 (\tau_I N_2, s_2) \wedge$

$(N_2, s_2)|\lambda(t)\rangle(N_2, s_2') \wedge \lambda(t) \in L_v$, there exists (N_1, s_1'') and (N_1, s_1') such that $(N_1, s_1) \Rightarrow (N_1, s_1'') \wedge (N_1, s_1'')|\lambda(t)\rangle(N_1, s_1') \wedge (N_1, s_1')R_2\tau_I(N_2, s_2')$.

Assume $t \in T_C \setminus (Start(N_C) \cup Stop(N_C))$. Since $s_C \neq 0$, therefore $(N_B, s_B)R_2'(\tau_I N_C, s_C)$. If $\lambda(t) = \tau$, for $(N_1, s_1)R_2\tau_I(N_2, s_2) \wedge (N_2, s_2)|\lambda(t)\rangle(N_2, s_2') \wedge \lambda(t) = \tau$, there exists (N_1, s_1') $= (N_1, s_1)$ such that $(N_1, s_1) \Rightarrow (N_1, s_1') \wedge (N_1, s_1')R_2(N_2, s_2')$. If $\lambda(t) \in L_v$, since (N_B, s_B') $R_2'\tau_I(N_C, s_C')$, t can be fired after execution of zero or more silent transitions from (N_1, s_1). Moreover, the effect of firing t in (N_1, s_1) and (N_1, s_1) is identical on places of $P_1 \setminus P_B = P_2 \setminus P_C$. Therefore, for $(N_1, s_1)R_2(\tau_I N_2, s_2) \wedge (N_2, s_2)|\lambda(t)\rangle(N_2, s_2') \wedge \lambda(t) \in L_v$, there exist (N_1, s_1'') and (N_1, s_1') such that $(N_1, s_1) \Rightarrow (N_1, s_1'') \wedge (N_1, s_1'')|\lambda(t)\rangle(N_1, s_1') \wedge$ $(N_1, s_1')R_2\tau_I(N_2, s_2')$.

From the above discussion of the four situations, it can be shown that R_2 satisfies the requirements 5 and 6 in Definition 16.

Before the existence of R_1 is proved, we first prove that P_C is active safe in $(N_2, [i])$. For any s_2 such that $(N_2, s_2) \in |N_2, [i]\rangle \wedge (N_2, s_2)|t\rangle \wedge t \in Start(N_C)$, since $(N_1, [i])R_2$ $\tau_I(N_2, [i])$, there exists s_1 such that $(N_1, s_1)R_2\tau_I(N_2, s_2)$. Let $s_1 = s_A + s_B$, $s_2 = s_A + s_C$. Since transitions in $Start(N_C)$ is only affected by places in $P_1 \setminus P_B = P_2 \setminus P_C$, and s_1 and s_2 share the same state on $P_1 \setminus P_B = P_2 \setminus P_C$, t is also enabled in (N_1, s_1). Because P_B is active safe in $(N_1, [i])$, $s_B = 0$. According to the definition of R_2, when $s_B = 0$, $s_C = 0$. Therefore, P_C is active safe in $(N_2, [i])$.

To prove $I = \alpha(N_C) \setminus \alpha(N_B) = \alpha(N_2) \setminus \alpha(N_1)$, $\tau_I(N_2) \cong_W N_1$, the existence of R_1 also need to be proven. As P_C is active safe in $(N_2,[i])$, R_1 can be defined as follows: $R_1 = \{(\tau_I(N_2,s),(N_1,s')) \mid s \in |N_2,[i]\rangle \wedge s = s_A + s_C \wedge s' = s_A + s_B \wedge s_A \in \beta(P_1 \setminus P_B) \wedge s_B \in \beta(P_B) \wedge s_C \in \beta(P_C) \wedge s_B \in |N_B,[i]\rangle \wedge s_C \in |N_C,[i]\rangle \wedge ((s_B = 0 \wedge s_C = 0) \vee (\tau_I N_C, s_C) R_1' (N_B, s_B))$. It is obvious that $\tau_I(N_2,[i]) R_1 (N_1,[i])$. It remains to be shown that

- $(N_2,s_2)R_1(N_1,s_1) \wedge (N_1,s_1)|\lambda(t)\rangle(N_1,s_1') \wedge \lambda(t) \in L_v \Rightarrow \exists (N_2,s_2'),(N_2,s_2'') : (N_2,s_2) \Rightarrow (N_2,s_2'') \wedge (N_2,s_2'')|\lambda(t)\rangle(N_2,s_2') \wedge (N_2,s_2')R_1(N_1,s_1'))$.
- $(N_2,s_2)R_1(N_1,s_1) \wedge (N_1,s_1)|\lambda(t)\rangle(N_1,s_1') \wedge \lambda(t) = \tau \Rightarrow \exists (N_2,s_2') : (N_2,s_2) \Rightarrow (N_2,s_2') \wedge (N_2,s_2')R_1(N_1,s_1')$.
- For any $(N_2,s_2) \in |N_2,[i]\rangle$, $\exists (N_1,s_1) : (N_2,s_2)R_1(N_1,s_1)$ or $\exists \sigma \in T^*, (N_2,s_2'),(N_1,s_1')$ such that $(N_2,s_2)|\sigma\rangle(N_2,s_2') \wedge (N_2,s_2')R_1(N_1,s_1')$.

Let $s_1 = s_A + s_B$, $s_2 = s_A + s_C$, $s_1' = s_A' + s_B'$, $s_2' = s_A' + s_C'$, $s_1'' = s_A'' + s_B''$ and $s_2'' = s_A'' + s_C''$. The proof of the first two requirements is similar to the proof of existence of R_2. The following four situations are considered.

Assume $t \in T_A$. Since the transitions in T_A are only affected by places in $P_1 \setminus P_B = P_2 \setminus P_C$, and s_1 and s_2 share the same state on $P_1 \setminus P_B = P_2 \setminus P_C$, t is also enabled in (N_2,s_2). Because firing a transition in T_A only affects places in $P_1 \setminus P_B = P_2 \setminus P_C$, firing t from (N_2,s_2) lead to a same state s_2' on $P_2 \setminus P_C$ like s_1' on $P_2 \setminus P_C$. Therefore, for

$\tau_I(N_2,s_2)R_1(N_1,s_1)\wedge(N_1,s_1)\big|\lambda(t)\big\rangle(N_1,s_1')\wedge\lambda(t)\in L_v$, there exists $(N_2,s_2'')=(N_2,s_2)$ and (N_2,s_2') such that $(N_2,s_2)\Rightarrow(N_2,s_2'')\wedge(N_2,s_2'')\big|\lambda(t)\big\rangle(N_2,s_2')\wedge\tau_I(N_2,s_2')R_2$ (N_1,s_1'). For $\tau_I(N_2,s_2)R_1(N_1,s_1)\wedge(N_1,s_1)\big|\lambda(t)\big\rangle(N_1,s_1')\wedge\lambda(t)=\tau$, there exists (N_2,s_2'): $(N_2,s_2)\big|\lambda(t)\big\rangle(N_2,s_2')$ such that $(N_2,s_2)\Rightarrow(N_2,s_2')\wedge\tau_I(N_2,s_2')R_1(N_1,s_1')$.

Assume $t\in Start(N_B)$. From requirement 9 of the theorem, $t\in L_v$. Since transitions in $Start(N_B)$ is only affected by places in $P_1\setminus P_B=P_2\setminus P_C$, and s_1 and s_2 share the same state on $P_1\setminus P_B=P_2\setminus P_C$, t is also enabled in (N_2,s_2). Because P_B is active safe in $(N_1,[i])$, $s_1=0$. According to the definition of R_1, when $s_B=0$, $s_C=0$. Moreover, the effect of firing t from (N_1,s_1) and (N_2,s_2) result in (N_1,s_1') and (N_2,s_2') such that $s_1'=s_A'+s_B'\wedge s_2'=s_A'+s_C'$ and $(\tau_I N_C,s_C')R_1'(N_B,s_B')$, which satisfies the definition of R_2. Therefore, for $\tau_I(N_2,s_2)R_1(N_1,s_1)\wedge(N_1,s_1)\big|\lambda(t)\big\rangle(N_1,s_1')\wedge\lambda(t)\in L_v$, there exists $(N_2,s_2'')=(N_2,s_2)$ and (N_2,s_2') such that $(N_2,s_2)\Rightarrow(N_2,s_2'')\wedge(N_2,s_2'')\big|\lambda(t)\big\rangle(N_2,s_2')$ $\wedge\tau_I(N_2,s_2')R_1(N_1,s_1')$.

Assume $t\in Stop(N_B)$. From requirement 10 of the theorem, $t\in L_v$. And obviously $s_B\neq 0$, therefore, $\tau_I(N_C,s_C)R_1'(N_B,s_B)$, and t can be fired after executing a sequence of silent transitions from (N_2,s_2). Moreover, the effect of firing t in (N_1,s_1) and (N_2,s_2) is identical on places of $P_1\setminus P_B=P_2\setminus P_C$. Since N_B and N_C are both sound WF-net, for the places of P_B and P_C, after firing t, $(s_B'=0\wedge s_C'=0)$. Therefore, for $\tau_I(N_2,s_2)R_1(N_1,s_1)\wedge(N_1,s_1)\big|\lambda(t)\big\rangle(N_1,s_1')\wedge\lambda(t)\in L_v$, there exists (N_2,s_2'') and (N_2,s_2') such that $(N_2,s_2)\Rightarrow(N_2,s_2'')\wedge(N_2,s_2'')\big|\lambda(t)\big\rangle(N_2,s_2')\ \wedge\tau_I(N_2,s_2')R_1(N_1,s_1')$.

Assume $t \in T_B \setminus (Start(N_B) \Upsilon Stop(N_B))$. Since $s_B \neq 0$, $(\tau_I N_C, s_C')R_1'(N_B, s_B')$. If $\lambda(t) = \tau$, for $\tau_I(N_2, s_2)R_1(N_1, s_1) \wedge (N_1, s_1)|\lambda(t)\rangle(N_1, s_1') \wedge \lambda(t) = \tau$, there exists $(N_2, s_2') = (N_2, s_2)$ such that $(N_2, s_2) \Rightarrow (N_2, s_2') \wedge \tau_I(N_2, s_2')R_1(N_1, s_1')$. If $\lambda(t) \in L_v$, since $(\tau_I N_C, s_C')R_1'$ (N_B, s_B'), t can be fired after execution of zero or more silent transitions from (N_2, s_2). Moreover, the effect of firing t in (N_1, s_1) and (N_2, s_2) is identical on places of $P_1 \setminus P_B = P_2 \setminus P_C$. Therefore, for $\tau_I(N_2, s_2)R_1(N_1, s_1) \wedge (N_1, s_1)|\lambda(t)\rangle(N_1, s_1') \wedge \lambda(t) \in L_v$, there exists (N_2, s_2'') and (N_2, s_2') such that $(N_2, s_2) \Rightarrow (N_2, s_2'') \wedge (N_2, s_2'')|\lambda(t)\rangle$ $(N_2, s_2') \wedge \tau_I(N_2, s_2')R_1(N_1, s_1')$.

Now we consider the third requirement for R_1. For any $(N_2, s_2) \in |N_2, [i]\rangle$, Let $s_2 = s_A + s_C$. If $s_C = 0$ or there exists $(N_B, s_B) \in |N_B, [i]\rangle$ such that $\tau_I(N_C, s_C)R_1'(N_B, s_B)$, according to the R_1 defined above, there exists (N_1, s_1) such that $\tau_I(N_2, s_2)R_1(N_1, s_1)$. If $s_C \neq 0$ and there is no $(N_B, s_B) \in |N_B, [i]\rangle$ such that $\tau_I(N_C, s_C)R_1'(N_B, s_B)$, since for $I = \alpha(N_C) \setminus \alpha(N_B) = \alpha(N_2) \setminus \alpha(N_1)$, $\tau_I(N_C) \cong_Q N_B$, for (N_C, s_C), there exists $\sigma \in T^*, s_C', s_B'$ such that $\forall t \in con(\sigma), t \notin inc(N_C) \wedge (N_C, s_C)|\sigma\rangle(N_C, s_C') \wedge (N_C, s_C')$ $R_1(N_B, s_B')$. Because $\forall t \in con(\sigma), t \notin inc(N_i')$, the same sequence of transitions can be executed from (N_2, s_2) such that $(N_2, s_2)|\sigma\rangle(N_2, s_2')$, $s_2' = s_A' + s_C'$ and $(N_C, s_C')R_1(N_B, s_B')$. Since $(N_2, s_2') \in |N_2, [i]\rangle$, according to the R_1 defined above, $(N_2, s_2')R_1(N_1, s_1')$. Therefore, the third requirement for R_1 is verified. Till now, all the three requirements for R_1 is verified. From the above discussion of the four situations, it can be shown that the R_1 defined satisfies the requirements 2, 3 and 4 in definition 16.

Since the two relation $R_1 \in (N_2, s') \times (N_1, s), R_2 \in (N_1, s) \times (N_2, s')$ defined satisfy all the requirements in Definition 16, for $I = \alpha(N_C) \setminus \alpha(N_B) = \alpha(N_2) \setminus \alpha(N_1), N_1 \cong_W \tau_I(N_2)$.

2. To prove N_2 is a sound, the three requirements of soundness need to be verified.

Safeness: $(flat(Q_2), [i])$ is safe. For any marked, labeled P/T net $(N_2, s_2) \in \left| N_2, [i] \right\rangle$, according to the result one, there exists a (N_1, s_1) such that $(N_1, s_1) R_2 (N_2, s_2)$. According to the R_2 defined, $(N_1, s_1) \in \left| N_1, [i] \right\rangle \wedge s = s_A + s_B \wedge s' = s_A + s_C \wedge s_A \in \beta(P_1 \setminus P_B) \wedge s_B \in \beta(P_B) \wedge s_C \in \beta(P_C) \wedge s_B \in \left| N_B, [i] \right\rangle \wedge s_C \in \left| N_C, [i] \right\rangle \wedge ((s_B = 0 \wedge s_C = 0) \vee (N_B, s_B) R_2' \tau_I(N_C, s_C))$. For the places $p \in P_2 \setminus P_C$, as N_1 is a sound WF-net, $s_A(p) = s_1(p) \leq 1$. For the places $p \in P_C$, since P_C is active safe in $(N_2, [i])$ and N_C is a sound WF-net, $s_C(p) \leq 1$.

Proper completion: for any reachable marking $s \in \left| flat(Q_2), [i] \right\rangle$, $o \in s$ implies $s = [o]$. For $(N_2, s_2) \in \left| N_2, [i] \right\rangle : o \in s_2$, according to the result one, there exists a (N_1, s_1) such that $(N_1, s_1) R_2 (N_2, s_2)$. According to the R_2 defined, $(N_1, s_1) \in \left| N_1, [i] \right\rangle \wedge s = s_A + s_B \wedge s' = s_A + s_C \wedge s_A \in \beta(P_1 \setminus P_B) \wedge s_B \in \beta(P_B) \wedge s_C \in \beta(P_C) \wedge s_B \in \left| N_B, [i] \right\rangle \wedge s_C \in \left| N_C, [i] \right\rangle \wedge ((s_B = 0 \wedge s_C = 0) \vee (N_B, s_B) R_2' \tau_I(N_C, s_C))$. For the places $p \in P_2 \setminus P_C$, Since N_1 is a sound WF-net, the other place except o are empty. For the places $p \in P_C$, according to the R_2 defined, if $S_B = 0$, then $S_C = 0$.

completion option: for any reachable marking $s \in |N,[i]\rangle$, $[o] \in |N,s\rangle$. According to result one, for any $(N_2,s_2) \in |N_2,[i]\rangle$, $\exists (N_1,s_1) : (N_2,s_2) R_1 (N_1,s_1)$ or $\exists \sigma \in T^*, (N_2,s_2'), (N_1,s_1')$ such that $(N_2,s_2) |\sigma\rangle (N_2,s_2') \wedge (N_2,s_2') R_1 (N_1,s_1')$. Because N_1 is sound, from (N_1,s_1') or (N_1,s_1'), there is a $\sigma' \in T^*$ such that $(N_1,s_1) |\sigma'\rangle (N_1,[o])$ or $(N_1,s_1') |\sigma'\rangle (N_1,[o])$. From (N_2,s_2') an equivalent path can be executed to fire a stop transition and reach $(N_2,[o])$.

□

Based on the results in Theorem 1, we will now prove that for an IOWF-net containing a number of WF-nets, the entire workflow remains to be sound after replacing each sub-workflow with a limited equivalent workflow.

Theorem 2. *Let* $Q = (C,n,N_1,...,N_{n-1},G)$ *and* $Q' = (C,n,N_1',...,N_{n-1}',G)$ *be two IOWF-net such that*

1. Q *is a sound IOWF-net,*
2. $\forall k : 0 \le k < n$, $N_k = (P_k,T_k,M_k,F_k,\lambda_k)$ *and* $N_k' = (P_k',T_k',M_k',F_k',\lambda_k')$ *are sound WF-nets*□
3. $N_1 = flat(Q) = (P_1,T_1,M_1,F_1,\lambda_1)$, $N_2 = flat(Q') = (P_2,T_2,M_2,F_2,\lambda_2)$,
4. N_1 *is a sound WF-net,*
5. *In* N_1, *for any* $t \in T_1 : \lambda_1(t) \ne \tau$, *i.e., all transitions in* N_1 *are observable,*
6. *In* N_2, *for any* $t \in Start(N_2) \,\mathrm{Y}\, Stop(N_2) : \lambda_2(t) \ne \tau$, *i.e., the start and stop transitions in* N_2 *are observable,*
7. *In* N_1, *for any* $t_1,t_2 \in T_1 : \lambda_1(t_1) \ne \lambda_1(t_2)$, *i.e., for every two transition in* N_1, *they correspond to different methods,*

8. *In* N_2*, for any* $t_1, t_2 : \lambda_2(t_1) \neq \tau \wedge \lambda_2(t_2) \neq \tau, \lambda_2(t_1) \neq \lambda_2(t_2)$*, i.e., for every two observable transition in* N_2*, they correspond to different methods,*

9. $\forall k : 0 \leq k < n$, $\forall t \in Start(N_k), \lambda_k(t) \neq \tau, \forall t \in Start(N_k'), \lambda_k'(t) \neq \tau$, *and* $\{\lambda_k(t) \mid t \in Start(N_k)\} = \{\lambda_k'(t) \mid t \in Start(N_k')\}$, *i.e., all start transitions in* N_k *and* N_k' *are observable, and the set of start transitions in* N_k *and* N_k' *coincide,*

10. $\forall k : 0 \leq k < n$, $\forall t \in Stop(N_k), \lambda_k(t) \neq \tau, \forall t \in Stop(N_k'), \lambda_k'(t) \neq \tau$, *and* $\{\lambda_k(t) \mid t \in Stop(N_k)\} = \{\lambda_k'(t) \mid t \in Stop(N_k')\}$, *i.e., all stop transitions in* N_k *and* N_k' *are observable, and the set of stop transitions in* N_k *and* N_k' *are observable,*

11. $\forall k : 0 \leq k < n$, P_k *is activation safe in* $(N_1, [i])$,

12. $\forall k : 0 \leq k < n$, *for* $I = \alpha(N_k') \setminus \alpha(N_k)$, $N_k \cong_{Q1} \tau_I(N_k')$,

Then

1. $\forall k : 0 \leq k < n$, P_k' *is activation safe in* $(N_2, [i])$.

2. N_2 *is a sound WF-net.*

Proof.

$\forall k : 0 \leq k < n$, let $Q^k = (C, n, N_1', ..N_k', N_{K+1}, ..., N_{n-1}, G)$, i.e., Q^k is the IOWF-net with the first *k* sub-workflows replaced by their limited equivalent workflows. Induction is used to prove that $\forall k : 0 \leq k < n$, $P_1', ..P_k', P_{K+1}, ..., P_{n-1}$ are all activation safe in $(flat(Q^k), [i])$ and $flat(Q^k)$ is sound.

Base case. Assume that $k = 0$. Obviously $Q^0 = Q_1$, and therefore, $\forall k : 0 \leq k < n$, P_k is activation safe in $(flat(Q^0), [i])$ and $flat(Q^0)$ is sound.

Inductive step. Assume that $k \geq 1$. The induction hypothesis states that $P_1', ..P_{k-1}', P_K, ..., P_{n-1}$ are all activation safe in $(flat(Q^{k-1}), [i])$ and $flat(Q^{k-1})$ is sound. We need to prove that $flat(Q^k)$ is sound. Let $Q_A = (C, n-1, N_1', ..N_{k-1}', N_{K+1}, ..., N_{n-1}, G')$ such that $G' = G \setminus (C, M_k) \Upsilon (C, M_k)$. To fit the requirement of Theorem 1, let $Q_1 = Q^{k-1}, Q_2 = Q^k$, $N_A = flat(Q_A)$, $N_B = N_{k-1}$ and $N_C = N_k$. It is easy to see that they satisfy all the requirements stated in Theorem 1. Therefore, from Theorem 1, $flat(Q^k)$ is sound. Moreover, since for $I = \alpha(N_C) \setminus \alpha(N_B) = \alpha(N_2) \setminus \alpha(N_1)$, $\tau_I(N_2) \cong_W N_1$, considering all the start and stop transitions in the sub-workflow are observable and the firing sequence in $flat(Q^k)$ after abstraction is the same as $flat(Q^{k-1})$, $P_1', ..P_k', P_{K+1}, ..., P_{n-1}$ are all activation safe in $(flat(Q^k), [i])$.

Since $flat(Q') = flat(Q^n)$, $\forall k : 0 \leq k < n$, P_k' is activation safe in $(N_2, [i])$ and N_2 is a sound WF-net. □

3.3. Summary

In this chapter, a new local criteria concept called limited equivalence has been proposed for the P2P approach. Theorem 1 and 2 show that this new local criteria concept does not affect the overall soundness of the entire inter-organizational workflow. The inter-organizational workflow which is made up of the sub-workflows in figures 12 and 16 shows that, comparing with the original local criterion of branching bisimulation, the new local criterion of limited equivalence allows more flexible workflow modification within an organization. As this new

local criterion takes the direction of messages exchanged among different sub-workflows into account, using this local criterion, a participant of an inter-organizational workflow will need to know the direction of arcs connecting the sub-workflow before making changes. This information was not required in the original P2P approach proposed by van der Aalst (2003). As the information is not concerned with the workflow details within one organization, we still consider the new local criterion is an improved one.

Chapter 4. Conclusion and Future Research

This chapter concludes the thesis. In Section 1, the research results will be discussed. Section 2 presents research contribution and future research will be outlined in Section 3.

4.1. Research Results

This section is organized as follows: In section 4.1, the major research results is summarized. Section 2.2 focuses on the limitation of the research outcome.

4.1.1 Summary

This research has investigated the concept of local criteria as an important concept to help achieve the overall soundness and privacy of sub-workflow in an inter-organizational workflow model. The local criteria principle makes it possible to achieve the overall soundness of the entire inter-organizational workflow without exposing workflow details within the organization. However, as explained in Chapter 3, it is possible that a local criterion can become too restrictive under certain circumstances when we attempt to ensure global soundness is achieved. This research has proposed a new local criteria concept called limited equivalence for the P2P approach, and has shown that this new concept allows more flexible workflow modification. Proofs have also been presented to show this new concept does not affect the overall soundness of the entire inter-organizational workflow.

4.1.2 Limitation of Research Outcome

Most of the existing local criterion, including the new one proposed in this research, only supports the situation when the number of the participants involved in an inter-organizational workflow is fixed. However, in a real-life business environment, workflow participants change from time to time. For example, consider an inter-organizational workflow with two participants, a producer and a customer. When a third participant, for example a part producer needs to be brought into the inter-organizational workflow, not only the private workflow, but

also the public workflow may need to change. The local criterion proposed by this research may not be able to support such scenario. Thus, a more robust local criterion needs to be investigated.

4.2. Research Contribution

This is an improved concept over the WF-net equivalence by taking the direction of message exchanged among sub-workflows into account. By doing this, the new local criterion is able to support more flexible workflow change within an organization. As demonstrated in Figure 16 and explained in chapter three, some previously forbidden changes of private workflow now become possible. In an application situation, a computer program should be developed according to a local criterion to check if a private workflow change is permitted.

4.3. Future Research

For future research, we will develop a computer program to allow the limited equivalence relation between WF-nets to be verified automatically. Such a program is useful when a participant of an inter-organizational workflow wants to ensure the modified sub-workflow does not make the entire workflow unsound.

Furthermore, a comprehensive approach for flexible inter-organizational workflow design is still missing. Many possible scenarios in real life business processes such as business process outsourcing are still difficult to implement using the process oriented techniques. Particularly,

situations including flexible change of workflow participants are not fully supported by existing mechanisms. On the other hand, Web services composition technique provides an alterative way for workflow implementation on the Internet. As suggested by Wombacher et al. (2004), the existing service publishing and finding mechanism can be improved to make implementation of inter-organizational workflow easier. However, before this can be realised, the soundness problem still needs to be considered. In the future research, issues on process publishing and finding mechanism will be further investigated.

References

Alonso, G. Casati, F. Kuno, H. & Machiraju, V. 2004. *Web Services: Concepts, Architectures and Applications*. Springer-Verlag Berlin Heidelberg 2004.

Cardoso, J. Bostrom, R. P. & Sheth, A. 2004. 'Workflow Management Systems and ERP systems: Differences, Commonalities, and Applications', *Information Technology and Management*, 5 (3-4), pp. 319-338

Chiu, D. K. W. Cheung, S. C. Till, S. Karlapalem, K. L. Li, Q. & Kafeza, E. 2004. 'Workflow View Driven Cross-Organizational Interoperability in a Web Service Environment', *Information Technology and Management,* 5(3-4), pp. 221-250

Georgakopoulos, D. Hornick, M & Sheth, A. 1995. 'An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure', *Distributed and Parallel Database*, 3, pp. 119-153

Gou, H. M. Huang, B. Q. Liu, W. H. & Li, X. 2003. 'A framework for virtual enterprise operation management', *Computer in Industry,* 50 (3), pp. 333-352

Hamadi, R. & Benatallah, B. 2003. "A Petri Net-based Model for Web Service Composition", *Fourteenth Australasian Database Conference (ADC2003), Adelaide, Australia. Conferences in Research and Practice in Information Technology*, Vol. 17.

IBM. 2001. *Web Services Conceptual Architecture (WSCA 1.0)*, http://www-306.ibm.com/software/solutions/webservices/pdf/WSCA.pdf, Accessed 08 July 2005

Kindler, E. Martens, A. & Reisig, W. 2000. 'Inter-Operability of Workflow Applications: Local Criteria for Global soundness', In: van der Aalst, W.M.P.; Desel, J.; Oberweis, A.: *Lecture Notes in Computer Science, Vol. 1806: Business Process Managements-models, techniques and empirical studies*, pages 235-253. Springer-Verlag.

Leymann, F. & Roller, D. 2002. 'Using flows in information integration', *IBM System Journal*, 41(4), pp. 732-742

Leymann, F. Roller, D. & Schmidt, M. T. 2002. 'Web Services and business process management', *IBM Systems Journal,* 41(2), pp. 198-211

Milner, R. 1980. 'A calculus of communication system', *Lecture Notes in Computer Science*, vol. 92. Springer-Verlag, New York.

Monson-Haefel, R. 2004. *J2EE Web Services*. Boston : Addison-Wesley

Shen, M. & Liu, D. R. 2001. 'Coordinating Interorganizational Workflows Based on Process-Views' *Lecture Notes In Computer Science*; *Proceedings of the 12th International Conference on Database and Expert Systems Applications* 2113 pp. 274-283

Taylor, F. W. 1911. *The Principles of Scientific Management*, http://melbecon.unimelb.edu.au/het/taylor/sciman.htm, Accessed 08 July 2005

van der Aalst, W. M. P. 1996. 'Structural Characterizations of Sound Workflow Nets', *Computing Science Reports 96/23 Eindhoven University of Technology*

van der Aalst. W.M.P. 1998. 'The Application of Petri Nets to Workflow Management.' *The Journal of Circuits, Systems and Computers*, 8(1). pp. 21-66

van der Aalst, W. M. P. 1999. 'Interorganizational workflows: An approach based on message sequence charts and Petri nets', *Systems Analysis-Modelling-Simulation*, 34(3), pp. 335-367

van der Aalst. W.M.P. 2000. 'Loosely Coupled Interorganizational Workflows: Modeling and Analyzing Workflows Crossing Organizational Boundaries.' *Information and Management*, 37(2): pp. 67-75

van der Aalst, W. M. P. 2002. 'Making Work Flow: On the Application of Petri Nets to Business Process Management', In: J. Esparza, C. Lakos (Eds.): *Lecture Notes in Computer Science, Vol. 2360: 23rd International Conference on Applications and Theory of Petri Nets, Adelaide, Australia, June 24-30, 2002*, pages 1-22. Springer Verlag

van der Aalst, W. M. P. van Hee, L. M. and van der Toorn, R. A. 2002. 'Component-based software architectures: A frameworkl based on inheritance of behavior", *Science of computer Programming* 42(2-3) pp. 129-171

van der Aalst, W. M. P. 2003. 'Inheritance of Inter-organizational Workflows: How to Agree to Disagree Without Loosing Control?', *Information Technology and Management* 4 (4), pp. 345-389

van der Aalst, W. M. P. Dumas, M. & ter Hofstede, A. H. M. 2003. 'Web Service Composition languages: Old Wine in New Bottles?', In G. Chroust and C. Hofer, editors, *Proceeding of the 29th EUROMICRO Conference: New Waves in System Architecture*, IEEE Computer Society, Los Alamitos, CA. pp. 298-305.

van der Aalst, W. M. P. & Kumar, A 2003. 'XML-Based Schema Definition for Support of Interorganizational Workflow', *Information Systems Research*, 14(1), pp. 23-46

van Glabbeek, R. J. & Stork, D. G. 2003. 'Query Nets: Interacting Workflow Modules that Ensure Global Termination', *Proceedings Business Process Management 2003, Eindhoven, The Netherlands, LNCS, Springer,*

van Glabbeek, R. J. & Weijland, W.P. 1996. 'Branching Time and Abstraction in Bisimulation Semantics', *Journal of the ACM*, 43(3), pp. 555-600

W3C. Web Services Architecture Requirements, Oct. 2002. http://www.w3.org/TR/wsa-reqs, Accessed 08 July 2005

Wombacher, A. Fankhauser, P. Mahleko, B. & Neuhold, E. 2004. 'Matchmaking for Business Processes Based on Choreographies', *International Journal of Web Services Research*, 1(4), pp. 14-32

Zhao J. L. & Cheng H. K. 2005. 'Web services and process management: a union of convenience or a new area of research?', *Decision Support System*, 40 (1), pp. 1-8

zur Muehlen, M. Nickerson, J. V. & Swenson, K. D. 2004. "Developing Web Services Choreography Standards-The Case of REST vs. SOAP", *Decision Support System*, 40 (1), pp. 9-29