

December 2005

Developing Web Services Using Workflow Model: An Inter-organizational Perspective

Sim Lau

University of Wollongong

Geng Liang

University of Wollongong

Zhaohao Sun

University of Wollongong

Follow this and additional works at: <http://aisel.aisnet.org/acis2005>

Recommended Citation

Lau, Sim; Liang, Geng; and Sun, Zhaohao, "Developing Web Services Using Workflow Model: An Inter-organizational Perspective" (2005). *ACIS 2005 Proceedings*. 32.
<http://aisel.aisnet.org/acis2005/32>

This material is brought to you by the Australasian (ACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ACIS 2005 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Developing Web Services Using Workflow Model: An Inter-organizational Perspective

Geng Liang
School of Economics & Information Systems
University of Wollongong
Email: gl850@uow.edu.au

Dr. Sim Kim Lau
School of Economics & Information Systems
University of Wollongong
Email: simlau@uow.edu.au

Dr. Zhaohao Sun
School of Economics & Information Systems
University of Wollongong
Email: zsun@uow.edu.au

Abstract

This paper discusses how a workflow model can be used in the design and development of web services composition. We particularly investigate the development of web services composition in an inter-organizational workflow environment. We discuss respectively how to design an inter-organizational workflow from scratch when there is no existing internal workflow, and how to make existing internal workflows work together in an inter-organizational workflow environment.

Keyword

Web Services, workflow management, Petri nets

1 Introduction

The concept of workflow model and workflow management has come into being since 1970s, and web service is a new technique that appeared after 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, Dumas and ter Hofstede 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. This paper discusses how workflow model can be applied in the design and development of web services.

The paper is organized as follows: In section 2, the concept of workflow management and workflow model will be discussed. Section 3 focuses on web services and web services composition. In section 4, we discuss the design issues in web services composition from an inter-organizational workflow perspective. Finally we conclude the paper and outline future work.

2 Workflow Management

Generally, there are two basic approaches in information system development: data oriented and process oriented approaches. Recently there is a significant shift from data oriented to process oriented approach (van der Aalst, 2002). Workflow management system can be viewed as a product of shift to process oriented approach in information system.

The concept of business process focuses on 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 were assigned to do different works by 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 (1911)'s scientific management. To find ways to improve the efficiency of an organization, research is conducted in the routine aspect of work activities in which the goal of the business was redefined. Every step of a business process is carefully studied to determine if the step could 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 play a more and more important role in the business processes of an organization. However, the first attempt of using computer in business processes is not workflow management system. On the contrary, it is the database management system. The seventies and eighties were dominated by data-driven approaches in information systems. At that time, "the focus of information technology was on storing and retrieving information and as a result data modeling was the starting point for building an information system. The modeling of business process was often neglected and processes had to adapt to information technology" (van der Aalst 2002).

Even in the 1970s, the years when data-driven approach dominates information systems development, the rudiment of workflow management system, which is often known as office information system were developed and implemented in the office automation domain. These systems differ from the database management system on their explicit workflow models. Although there was great optimism about the applicability of office information systems, very few applications succeeded. Both the application of this technology and research almost stopped for a decade during the eighties. In the nineties, there was a renewed interest in workflow management system, particularly in the areas of banking and insurance (van der Aalst, Dumas and ter Hofstede 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 access heterogeneous, autonomous, and/or distributed information systems. While human-oriented workflow implementations often control and coordinate human tasks, system-oriented workflow implementation often control and coordinate software tasks (Georgakopoulos et al. 1995).

Data oriented and process oriented information systems are suitable for different types of business processes. According to Cardoso et al. (2004) there is three types of business processes: 1) business process involving human, 2) business process involving systems and applications, and 3) transactional business process. Generally speaking, human-oriented workflow management system is suitable for the first type of business process, system-oriented workflow management system is suitable for the second type, and data oriented information systems such as enterprise resource planning (ERP) systems are suitable for the third type of business process. We will focus on system-oriented workflow management system in this paper.

2.1 Workflow Models and Their Implementations

One of the major characteristics of workflow management system 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). In a workflow management system, the flow logic is represented in 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 workflow model including 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. States 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.

It is often that no participant is 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), a vertical line is used to divide the workflows 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 1 shows an example of a simple workflow model using 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.

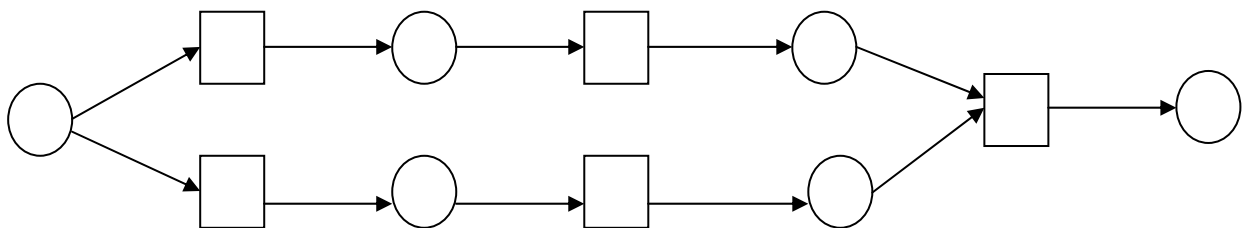


Figure 1: A Simple Workflow Model in Petri Net

The separation of flow logic and function logic makes the implementation of a workflow management system different compare with other computer systems. A two-level programming model, with programmers implementing the functions logic and nonprogrammers such as business specialists designing flow logic, is used in the implementation of workflow management system (Leymann & Roller 2002). As the core of workflow management system architecture, an “enactment service” creates workflow instances based on the workflow model and perform 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 that a workflow management system can launch include legacy systems, databases as well as 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 more easier for organizations to engage in a BPR project. This is because their business processes are easier to understand and change. On the other hand, data oriented systems such as the ERP cannot provide this kind of flexibility. Although they still considered themselves to have a workflow model, their workflow models are implicit and are often embedded in the applications and parameter tables. Thus this type of workflow model is difficult for the programmers and business specialists to understand and change. Recently there is a move for the data oriented systems to replace the flow logic embedded in the applications and parameter tables with a workflow management system to make these systems more “workflow driven” and to make it more flexible (Cardoso et al. 2004).

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 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 develop these workflow management systems. In general, there is still no widely supported standard for interoperability of workflow management system (van der Aalst and 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 and 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, the 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 whole 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 the clear 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.

3 Web Services

The trend from data oriented to process oriented is becoming more significant when web services come into being. Web services technique provides a new universal solution for the problem of how to invoke application on other system, which is a major issue in system-oriented workflow management system. Furthermore, the appearance of web services composition language makes it possible to integrate a number of web services into a workflow. Web service is a new solution for “machine to machine” communication based on the Internet technology. 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 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 provides 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 2 is adopted from IBM. (2001). It illustrates the basic web services architecture:

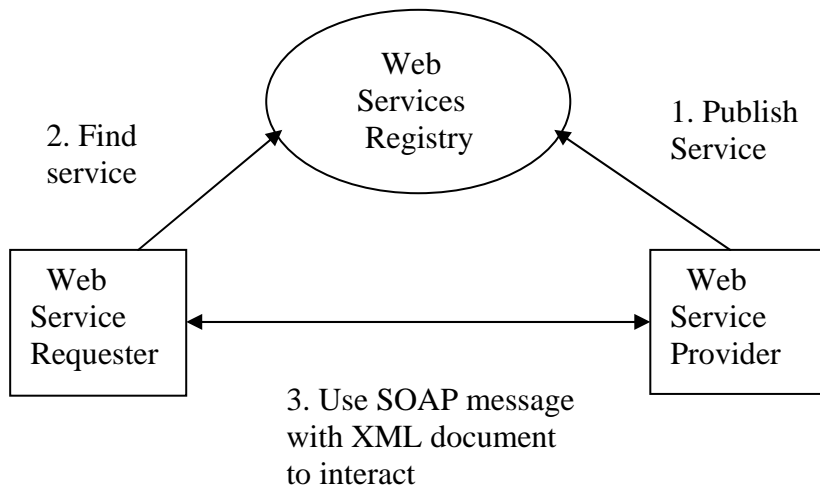


Figure 2: Web Services Architecture

This architecture is the basic architecture found in almost every introduction document to web services architecture. In Figure 2, 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 3 is adopted from IBM. (2001). It shows the conceptual stack of web services. In general, the upper layers build upon the capabilities provided by the lower layers. In Figure 3, the text on the left represents standard technologies that apply at that particular layer of the stack. The bottom layers of the stack, representing the base web services stack, are relatively mature and more standardized than the layers higher in the stack (IBM. 2001).

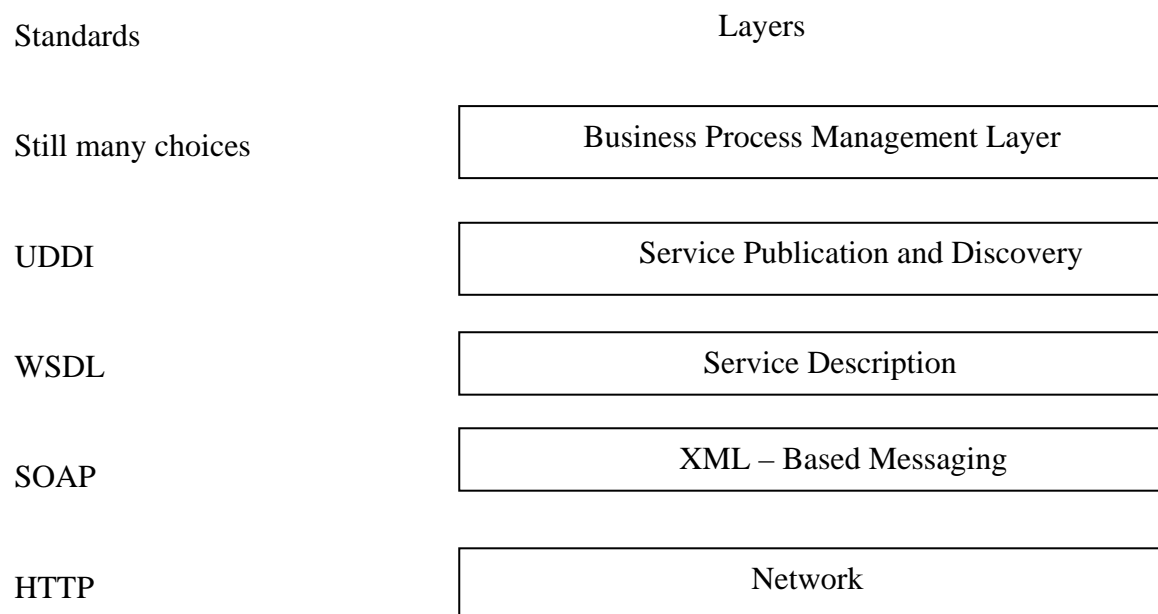


Figure 3: Web Services Conceptual Stack

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, Dumas and ter Hofstede 2003). From a workflow perspective, they are all about function logic of a workflow, but do not provide the mechanism of sequencing and state reserving. However one single web service cannot do the work of an entire business process. It is often that a business process contains more than one business activities. That is why 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. Thus 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 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, Dumas and ter Hofstede 2003). These include web service flow language, web service orchestration language, web service composition language and web-enabled workflow language. However we adopt the name of “web service composition language” in this paper.

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 WSDL. The most significant difference between WSDL and BPEL4WS is that BPEL4WS provides the state management function, which corresponds to the state factor in the workflow model (van der Aalst, Dumas and ter Hofstede 2003).

4 Design Issues For Web Services Composition In An Inter-organizational Workflow Environment

Two research problems have been identified in the design for web services composition in an inter-organizational workflow environment. The first issue is concerned with the following question: “When there is no existing internal workflow, how do we design an inter-organizational workflow from scratch?” and the second question is “When there are two or more internal workflows already existed in the organizations, how do we make these internal workflows work together in an inter-organizational workflow?” It is no doubt that these two problems are interrelated.

To solve these two problems, a workflow model for inter-organizational workflow is needed. One of the useful inter-organizational workflow models is based on Petri net and messages exchanged between internal workflows (van der Aalst 1999). Firstly, a “workflow net” model using Petri net concept is used for internal workflow. Then synchronous and asynchronous messages connecting different organizations are also modeled using message sequence diagram. Using such a model, some analytic techniques on Petri net such as liveness and boundedness verification can be used in workflow analysis.

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 4 shows an example of unsound workflow.

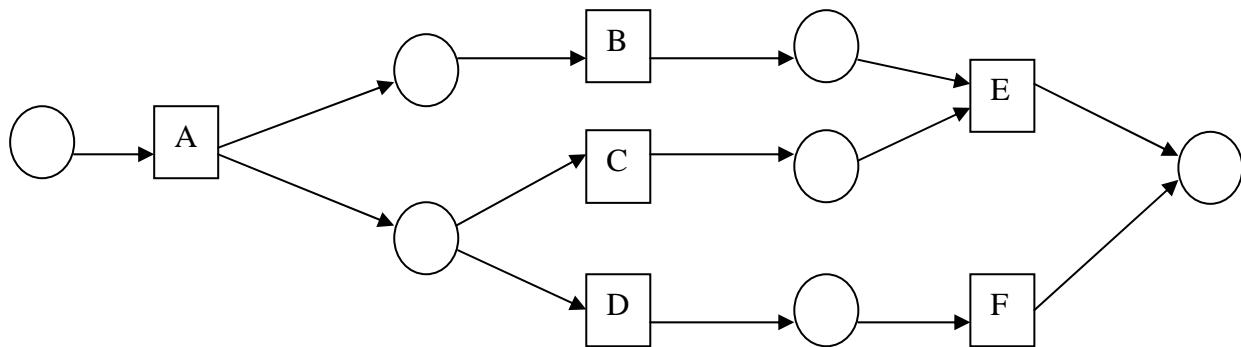


Figure 4: An Unsound Workflow

In Figure 4, 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 (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 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.

Although these approaches can be used to decide whether a workflow is sound, 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. Further more, how to design a sound inter-organizational workflow from scratch becomes a more practical problem.

A private-to-public (P2P) method is proposed to design an inter-organizational workflow from scratch (van der Aalst 2003). The 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 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 using the concept of inheritance (van der Aalst 2003), the organizations involved in the inter-organizational workflow can design and modify their internal private workflows in any way they want, and there is no need to worry about the soundness of the whole inter-organizational workflow.

In practice, it may be rare to design an inter-organizational workflow from scratch. There are already many workflow management system implemented in the business world. Therefore how do we make these existing workflow management systems work together become an important research issue. Public-to-private approach cannot be used to solve this problem because the private workflows are already there but the public workflow is still missing.

Web services architecture provides a mechanism to enable the 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. The match making mechanism proposed by Wombacher et al. (2004) is helpful here. The mechanism uses a revised version of Finite State Automata (FSA) model to decide whether two workflows are compatible to work together. Using this method, it will be easier for two organizations to establish business relations without relying on manually negotiation for the specification of public workflow. However, this match making mechanism does not consider the internal

implementation of the public workflow. Figure 5 shows a procurement match making scenario using the FSA model. The scenario is adopted from Wombacher et al. (2004).

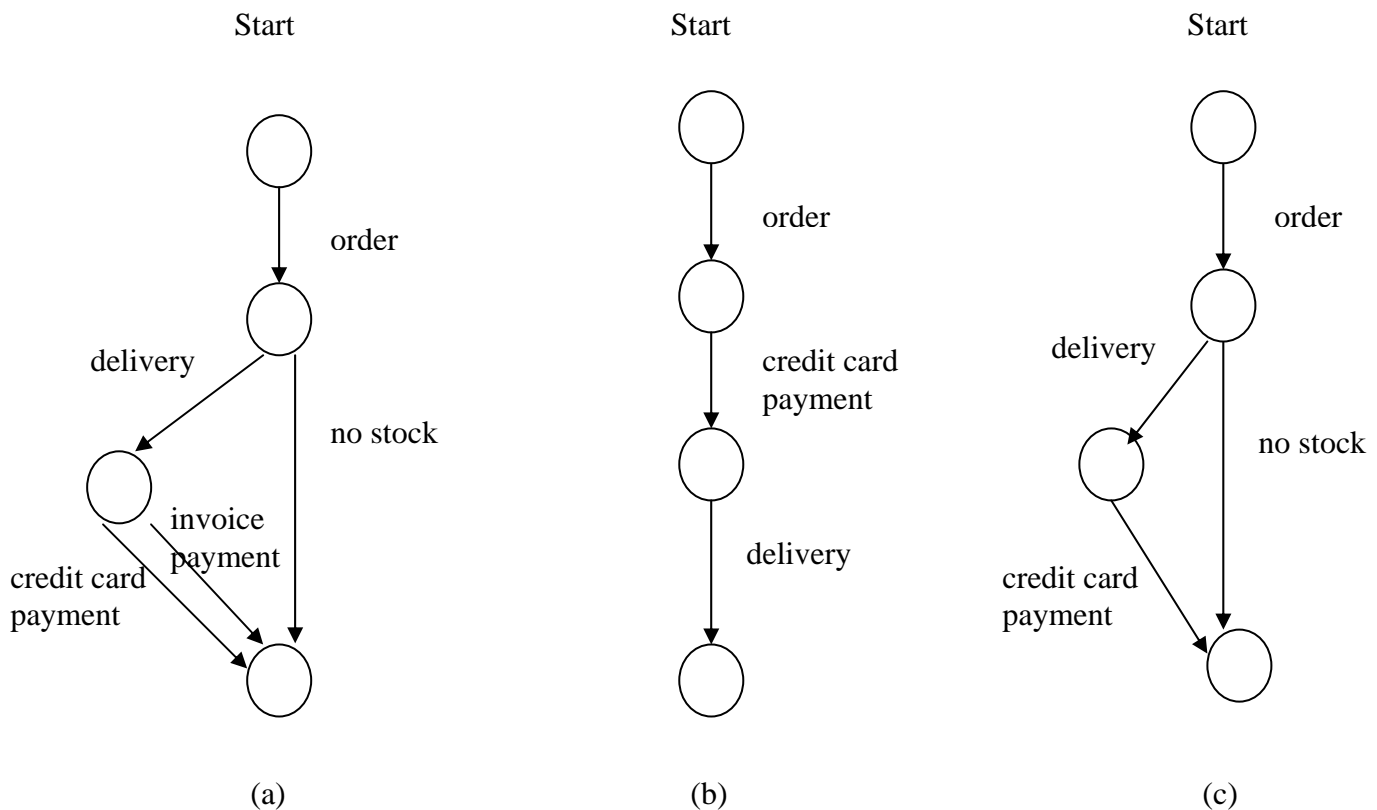


Figure 5: (a) Vendor Message Sequence (b) unmatched Customer Message Sequence (c) matched Customer Message Sequence

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 5 (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 firstly deliver the product, then receive the payment. Two types of payment method including credit card payment and invoice payment are supported. There are two errors in the message sequence in Figure 5 (b) as a customer message sequence interacting with the vendor’s message sequence showed in Figure 5 (a): the first one is that its payment message is sent before the delivery message, the second is that it 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 5 (c) is matched 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 automatically.

5 Conclusion and Future Work

In this paper, design issues in web services composition in an inter-organizational workflow are discussed. We focus on two research problems: 1) how to design an inter-organizational workflow from scratch when there is no existing internal workflow, 2) how to make existing internal workflows work together in an inter-organizational workflow environment. A P2P approach has been proposed to overcome the first problem, and a match making mechanism is thought to be helpful to solve the second problem. The relationship between the public and private workflow will be further investigated in future works.

6 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
- 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
- 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. 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
- 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

7 COPYRIGHT

Geng Liang, Dr. Sim Kim Lau & Dr. Zhaohao Sun © 2005. The authors assign to ACIS and educational and non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive licence to ACIS to publish this document in full in the Conference Papers and Proceedings. Those documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. Any other usage is prohibited without the express permission of the authors.