A Framework to Manage Message Level Authorisation in Service Oriented Collaborative Business Processes

ABSTRACT

Service oriented architecture and underlying Web service technologies facilitate the business collaboration in a diverse manner. This diversification, on the other hand, creates more security challenges than ever in the business world. Modern business often requires collaboration between individual social entities with different security policies defined and enforced.

Business processes are normally developed separately on different platforms across organisations. In most cases, they do not follow the same strategy. Emerging Web service and business process technologies have provided technological support for business collaboration across organisation boundaries. However, security concerns have become one of the main barriers that prevent its widespread adoption.

The importance of security in a computer-based environment has resulted in a large stream of research that focuses on the technical defences associated with protection in providing mathematical theories, cryptographic algorithms, and distributed systems and network security solutions. In other words, the existing work in the security area mainly contribute to providing solutions at the data, network, and computer systems level, and target either for single organisation or simple collaborations. However the challenges of security management in the rich domain of business collaboration constitute a vibrant area of security research, which has so far received only limited attention and has never been addressed to its entirety.

Existing business collaboration methodologies seldom consider security issues which address business integration and legal requirements. The inherited openness and distribution nature of Web services based inter-organisational business processes may result in more security breaches. Current business process related standards do not provide any support for business process

security protection even if the participating organisations already have a working security policy.

The challenge is how security policy is specified, compared, integrated, enforced and managed for collaborative services. The aim of the research is to develop a security management system that covers the entire life-cycle of secure business collaboration from strategy level, security specification from organisational level, system management from design time specification, monitoring and enforcement from run time. In this thesis, we propose a scenario-based requirements analysis approach to make the requirements clear as the first step. The description and explanation of a set of requirements are based on modelling a variety of representational business collaboration scenarios with Petri Nets.

The security policies on how to create and maintain the dependencies, among business partners are also studied. We focused on the consistence and potentially contradict among partners at various levels of collaboration with one another. In order to provide the role-based access control capability in widely accepted de facto standards, WS-BPEL and BPEL4People, we extended these standards by our design time authorisation specification - BPEL4RBAC. BPEL4RBAC extends its ability from both RBAC side and WS-BPEL side. As an extension, BPEL4RBAC is highly compatible with WS-BPEL and BPEL4People standards. This ensures the access control functions can be seamlessly integrated with WS-BPEL.

To cater for run-time authorisation verification, we designed Role-Net to enforce collaboration reliability in terms of authorisation policies. The specification and reliability properties guarantee the correctness of running collaborative business process in a secure way.

DECLARATION

I, Xin Wang, declare that the PhD thesis entitled *A Framework to Manage Message Level Authorisation in Service Oriented Collaborative Business Processes* is no more than 100,000 words in length including quotes and exclusive of tables, figures, appendices, bibliography, references and footnotes. This thesis contains no material that has been submitted previously, in whole or in part, for the award of any other academic or diploma. Except where otherwise indicated, this thesis is my own work.

Signature Date

ACKNOWLEDGEMENT

It is very hard to overstate my appreciation to my supervisor, Professor Yanchun Zhang. Despite the numerous efforts on inspiring me in the path of research, he has demonstrated the way of how to work as a professional scholar. The methods and knowledge he taught me are valuable in the career of research, while I will benefit from the Prof. Zhang's life wisdom behind these representations. I deeply appreciate his significant direction and invaluable comments at all stages of my study. With his enthusiasm and inspiration, I can now step in the gate of the world of scientific research.

I also thank my associate supervisor, Associate Professor Hao Shi, for her exceptional support and encouragement. Her patience and suggestions made many research ideas in this thesis possible.

I would like to thank my colleagues in the Centre for Applied Informatics Research in the School of Engineering and Science. Their kindness and helpful discussion also formed an important part of this thesis. My appreciation also goes to the School of Engineering and Science, and Office for Postgraduate Research. Without their support, the accomplishment of this research work would not have been possible.

A special thank goes to Mr. Haiyang Sun from Department of Computing at Macquarie University, who made research cooperation a pleasure journey.

Finally, I would like to extend my deepest gratitude to my families, who support me all along with my research. The completion of the thesis would not have achieved without their encouragement, support and understanding.

PUBLICATION LIST

1. X. Wang, Y. Zhang and H. Shi, *Scenario-based Petri Net Approach for Collaborative Business Process Modelling*, Proc of 2007 IEEE Asia-Pacific Services Computing Conference (APSCC'07), Tsukuba, Japan, pp 18 – 25.
2. Y. Zhang, H. Shi, X. Wang, J. Zhang, *Collaborative Legal Information Sharing P2P Network*, 2007 IFIP International Conference on Network and Parallel Computing, Dalian, China, pp 41 – 47.
3. X. Wang, Y. Zhang, H. Shi, and J. Yang, *BPEL4RBAC: An Authorisation Specification for WS-BPEL*, Proc of 2008 Web Information Systems Engineering (WISE'08), Auckland, New Zealand, pp. 381 – 395.
4. X. Wang, Y. Zhang and H. Shi, *Access Control for Human Tasks in Service Oriented Architecture*, Proc of 2008 IEEE International Conference on e-Business Engineering (ICEBE'08), Xi'an China, pp 455 – 460.
5. H. Sun, X. Wang, J. Yang, and Y. Zhang, *Authorisation Policy Based Business Collaboration Reliability Verification*, Proc of 2008 International Conference on Service Oriented Computing (ICSOC'08), Sydney, Australia, pp. 579 – 584.
6. H. Sun, J. Yang, X. Wang and Y. Zhang, *A Verification Mechanism for Secured Message Processing in Business Collaboration*, Proc of 2009 Asia-Pacific Web Conference (APWeb'09), Suzhou, China, pp 480 – 491.

TABLE OF CONTENTS

TABLE OF FIGURES

CHAPTER 1

INTRODUCTION

In the past few decades, changes in the economic environment, such as globalisation, mass customisation, new competitive pressure have forced organisations to search for innovations and gain competitive advantages. A key factor to maintain competitiveness is the capability to cooperate with existing partners and potential customers in a standardised way. Therefore, "Managing business processes is a necessity for every organisation [1] ."

Automating business services on demand and adapting business processes (BP) to market changes are main necessities to facilitate collaboration of business processes. Scholars and researchers from management science are presenting new ideas to help corporations [1]. The increasing effectiveness and efficiency of BP Management are attracting companies to gain competitive advantage among their rivals.

Service-orientated methodologies are applied to facilitate business collaboration with partners and customers. This emerging paradigm provides loosely coupled and distributed business services across organisational boundaries [2]. Then, aiming at a same business goal, the business collaboration among multiple organisations is achievable.

On the other hand, however, balancing business collaboration and system security are competing goals [3] as illustrated in Figure 1-1. Business applications contain information with variable levels of sensitivity in nature. However, in business process environment, the business activities are highly unpredictable comparing with single user applications [4]. In contrast, the open access in

business process requires higher level of integrity and confidentiality. Many research works have been done in this challenging area [5] [6] [7] [8].

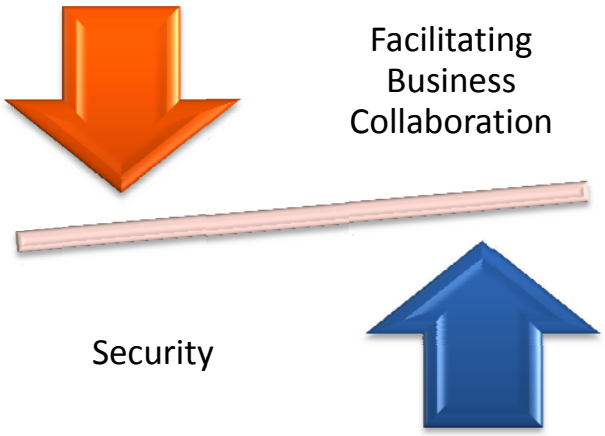

Figure 1-1 Security VS Collaboration

1.1 BACKGROUND

From a historic view, information technology has played as a significant role in each evolvement in business world, especially in recent years. Word processing software enabled office automation. Networks made paperless office and remote office possible. Wireless technology enlightened mobile businesses and Web 2.0 is booming blog and social network at the moment.

In order to facilitate collaborative business processes with partners and customers, the service-orientated methodologies are applied to business process modelling. This emerging paradigm provides loosely coupled and distributed functionalities to deliver flexible business services. In this case, Business processes are composed by services, which work as viable components across organisational boundaries [2]. Then, aiming at a same business goal, the business collaboration among multiple organisations is achievable.

Security of computer-based business systems is, by design, the key element for protecting the confidentiality, integrity, and accessibility of the system and

services. Given the information and service-intense characteristics of our modern economy (e.g., based more on Internet), it should be no surprise to learn that security is a growing concern among most organisations.

It is especially true when organisations try to construct extensive networks of communication links to engage each other in order to deliver their corporate business services. For example, medical centre needs to work with health insurance companies, general practicians and specialists to deliver its build-to-order service to its customers. In this scenario, different parties may have their own security policies with their own implementation and enforcement mechanism. In order for them to work together and not violate each other's security policy, technological support are required to allow the parties involved to ascertain that their security policies and their partners' can be checked, tested, and enforced during the collaboration. All of this requires continuously adjusting and aligning security policies within end-to-end business processes that span diverse organisations.

1.2 MOTIVATING SCENARIO

Here is an example taken from a normal bank loan application process. Bank is in the centre of the whole processes. Customer submits the loan application to the bank. Credit agency and real estate will assess the customer credit level and the property value respectively. Bank will make the final decision. Insurance company will provide the insurance for the property.

Figure 1-2 Loan Application Involved Partners

Bank divides the application and outsources the related parts to professional partners. There are multiple criteria in terms of evaluating the loan risk which is the core part leading to the final decision. At this stage, we assume that each collaborating partner has their own access control mechanism and appropriate IT system.

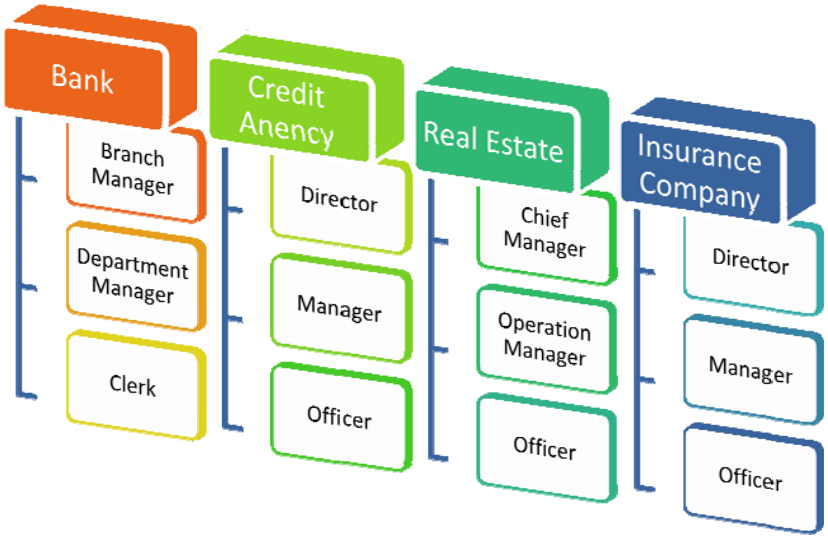

Figure 1-3 Organisation Charts

From the bank's perspective, low risk application is processed by normal **CLERK** while the applications ranked as higher risk level are to be assessed by **DEPARTMENT MANAGER**. The assessment of risk level involves the proposed loan amount, customer credit rating and property value.

Even if we assume that each partner has their own systems. Then how can they communicate and enforce the security policies on track in accordance to business collaboration? In this scenario, before any assessment result send back to bank, the receiver need appropriate permission to open the result. For instance, if the customer credit rating below A, the loan application need to be approved by **DEPARTMENT MANAGER**. While the real estate agent has a similar policy, if the property value is over one million, the assessment report is to be reviewed by the **OPERATION MANAGER**.

These kinds of collaboration policies can be hard-coded into existing systems given the condition that the internal security policies are not constantly updating. It is not realistic, however, in the business world as new partnership is forming and ceasing nearly every day. The bank might outsource the property valuation service to another real estate agent, for example, if the current one couldn't provide the result in certain timeframe or the competitor can provide a better service with lower cost.

1.3 PROBLEM STATEMENT

Business processes can communicate asynchronously for some simple application integration in current Web service environment. However, it has not achieved adequately support for the complex and critical business processes. Research efforts have shown that harmony business and IT alignment can improve business performance [9] [10]. However the harmony has not achieved in BPM area. Many IT-focused approaches towards process modelling failed

because they only concentrated on the selection of software solution rather than on the challenges of how to align business and IT [1].

1.3.1 Collaborative - the missing technical layer

In order to capture the collaborative business process models, the demands of business collaborations development must be captured. The collaboration models design should also be verified according to existing economic conditions, government regulations, industry policies and so on.

The ambiguity of BPM requirements prevents implementing the real power of BPM [11]. BPM is closely related to business strategy and supported by state-of-the-art IT techniques. Consequently, the alignment plays an increasingly significant role in a successful BPM project. This research focuses on the business IT alignment area to provide a clear vision on its requirements. As shown in figure 1-4.

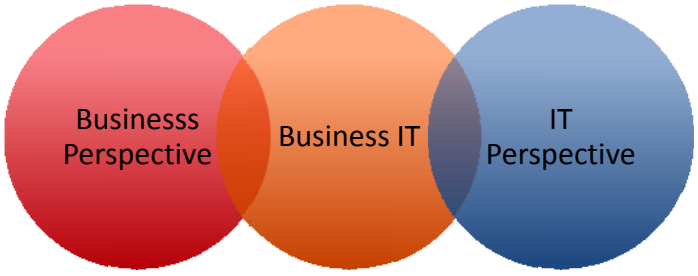

Figure 1-4 Relationship of Business and IT

As mentioned in a variety of modelling standards, business process can be categorised into public process, private process and collaborative process [2]. And the collaborative process is the essential process in business collaboration since partners links their interior processes, public or private, to achieve smooth interaction. The relationship of these processes is shown in figure 1-5.

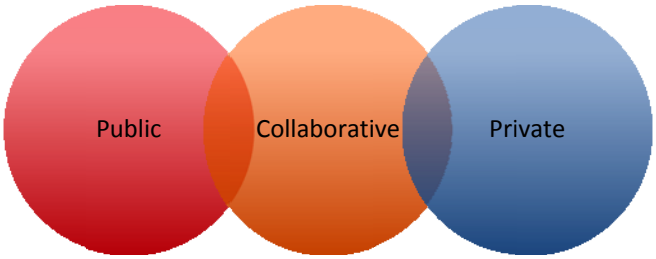

Figure 1-5 Collaborative Process

Business processes are developed separately on different platforms. In most cases, they do not follow the same strategy. Existing BPM methodologies seldom consider security issues which address business integration and legal requirements [22]. Therefore the area of research is of vital importance to software engineering and distributed computing. It plays part in the development of next generation technologies that contribute to a massively distributed computing infrastructure made up of many different Internet resident software services aiming to interoperate over the network to virtually form a single logical system offering on-demand and value-added user services. This research aims to make an impact on fundamental research on security aspects of service oriented collaborations. Furthermore, it aims to develop generic infrastructure that is broadly applicable to several industry sectors and applications such as e-health, e-logistics or e-government.

1.3.2 Security – the missing guardian

Security is listed as No. 1 tech flop by InfoWorld [12] review of IT industry practice of the last 20 years. This thesis addresses the critical security issues in service based business collaboration and provides solutions for the design and integration of secured business services. The security of collaborative business process is crucial and significant for the business success of organisations as

security problems would affect companies and their stakeholders in terms of profit and reputation.

Emerging Web service and business process technologies have provided technological support for business collaboration across organisation boundaries. However, security concerns have become one of the main barriers that prevent its widespread adoption [1].

The importance of security in a computer-based environment has resulted in a large stream of research that focuses on the technical defences associated with protection in providing mathematical theories, cryptographic algorithms, and distributed systems and network security solutions. In other words, the existing work in the security area mainly contribute to providing solutions at the data, network, and computer systems level, and target either for single organisation or simple collaborations (ie. single sign-on). However the challenges of security management in the rich domain of business collaboration constitute a vibrant area of security research, which has so far received only limited attention and has never been addressed to its entirety.

1.4 RESEARCH GOALS AND SCOPE

This research work contributes towards developing and managing secured and extendible e-business applications. This facilitates bridging the gap between business collaboration requirements and Web service methodologies. From the system architecture' perspective, this is an opposite way to the current research activities and standards-oriented approaches that focus mainly on technique based solutions aimed for data, network and system level security.

The overall objective of this research is to provide a holistic approach to role based security management in business collaboration. This research work involves business management, security requirement analysis and access control,

service architecture. Web service standards of W3C and OASIS on service security and business process management are also studied in details. The research scope is depicted in the following diagram.

Figure 1-6 Research Scope

For a holistic approach to security management in business collaboration, this covers the entire management life-cycle from design time specification, policy checking to run time monitoring, enforcement, and negotiation. In order to achieve this objective, we need to deliver the following outcomes:

1. Exploring the requirements of different collaboration patterns which may require different integrated and collaborative security policies;
2. Developing a formal verification model that can be used to verify the compliance of the security policies for different collaboration patterns;
3. Extending WS-BPEL and BPEL4People standards to support secure service based business collaboration at design time;
4. Designing mechanisms for enforcing the compliance of the security policies for different collaboration patterns at conceptual level;
5. Prototyping a secure SOA system to implement business collaboration based access control policies.

Increasingly businesses are using the Internet and the web to deliver tailored, on-demand services to partners and clients. Delivery of these services requires as key elements extensive networks of communication links between business partners and integration of the business processes of the partners. Service Oriented Architecture (SOA) has emerged in the past decade which provides a new way to use, re-use and manage IT systems. In contrast to traditional application-centric view, applications in SOA paradigm are concerned with how to expose services and which services to expose. Service is the core idea in SOA, which is "functionality encapsulated in a form that is readily consumable by other applications and services [2]". SOA considers IT systems as a collection of reusable services rather than a collection of static applications. Consequently, SOA lowers the technology barriers of inter- and intra-organisation business process management.

Currently, access to information is most often approached from a simplistic perspective of specifying what other users of the particular system can do to the information (in terms of access rights). These access rights are specified and enforced by many different technologies, with varying degrees of compatibility. It can be seen from the above that the current business practices involve the propagation of information between organisations. Agreements (and mechanisms) for propagating such information needs to be an accompanying process to understand and enforce the security policies of all involved parties. This requires not just a mechanistic application of the sum of all policies (as such an approach would likely fail with policies being applied out of context) but a process [13] that results in a secure handling of information and accessing services satisfactory to all parties.

There are various access control models addressing different aspects in the access control domain. Role based access control (RBAC) [5] has emerged in 1990s. By associating permissions and roles, RBAC allows the access control

model in the same way that maps naturally to an organisation's structure, and the concept of a role is in correspondence to an organisational position. Several constraints may apply to an RBAC model. For example, Separation of Duty (SoD) is one of the well-known security principles which requires two or more different people to be responsible for the completion of a task or set of related tasks [14]. To protect the interest of organisations, the conflicting roles must not be assigned to the same user in a business process [15]. While in some cases, the same user might be required to perform two different activities. This is considered as a binding of duty constraint (BoD). BoD and SoD are typical security policies. These security policies are embodied in RBAC to specify these access control constraints.

The Task-based Access Control (TBAC) was built on the RBAC, which models access control from task oriented perspective [16]. TBAC approach separates system level activities to support scalable and reusable access control models. Organisation based Access Control (OBAC) [17] model aims to share specific data and functionality with collaboration partners. The specification of the security policy is completely parameterized by the organisation in order to handle simultaneously security policies associated with different participating partners.

The RBAC, TBAC and OBAC methods provide efficient and effective access control capability for current application-centric systems. In the SOA era, most security issues arise from the interaction among applications rather than inside of applications. The application based access control mechanisms, therefore, are no longer suitable for security in service-centric IT systems. There is no comprehensive approach to secure SOA. Therefore traditional access control cannot provide adequate shield for SOA due to its complexity [18].

Research has also been done in the area of security policy specification [19] [20] [21]. Most of these studies focus on how to specify security information at

the message level by extending existing languages or other technical security solutions. There is also research work being carried out on Web Service Security [22] [23] [19], however again these studies focused on the specific communication level rather than specifying the security policy required for business collaboration and integration.

The most notable set of emerging specifications for service security policy are those outlined in the Web Services (WS) roadmap. The roadmap consists of a number of component specifications, the core amongst them are WS-Security [24], WS-Policy [25], and WS-Trust [26]. WS-Security is a specification for securing the whole or part of an XML message using cryptographic technology, and attaching security credentials. WS-Policy is used to describe the security policies in terms of their characteristics and supported features. In fact it is a meta-language which can be used to create various policy languages for different purpose including access control policies. WS-Trust defines a trust model that allows security tokens to be exchanged using mechanisms provided by WS-Security and allows online trust relationships to be established according to the requirements supplied by WS-Policy for the issuance and dissemination of credentials within different trust domains. Security Assertion Markup Language (SAML) [27] on the other hand, is used to exchanging authentication and authorisation data between security domains. SAML has become the definitive standard underlying many web 'Single Sign-On' solutions in the enterprise identity management problem space.

Some very interesting work has been done and outlined in the area of collaborative systems [3] [28]. However these work only focused on the aspects of policy specification and modelling for protecting data and resources. Because these works was not set up in the context of service based business collaboration, the issues of policy consistency and comparability among different organisations were completely overlooked. In [29], a mechanism called Access Path Discovery

was developed to support secured cross domain collaboration. However the work was based on a simple collaboration type, i.e., chain of collaboration. The proposed solution does not work for other types of collaborations, e.g., joined collaboration, outsourced collaboration, collaboration with propagation, etc, which will be studied fully in the project.

The works mentioned above has focused on the low level security issues in terms of protocols or security specification languages. Furthermore, these studies only provide solutions to some aspects of security issues in terms of: security policy specification, access control in distributed environment, and access decision making. What is missing and unclear is what needs to be specified as security policies in the setting of service based diverse business collaborations and how criteria for compatibility and consistency checking can be defined, enforced, and managed. Only when a full understanding of the nature and characteristics of collaborative process itself and its relation to the policies of all involved organisations is achieved, can theoretical models and mechanism be developed. This is exactly what this research is heading to.

1.5 RESEARCH METHODOLOGIES

To the best of our knowledge, there are very few studies reported in literature that systematically and thoroughly address the problem of security issues in service based business collaboration [30]. Current studies with application security approaches have limitations in meeting the challenges in dealing with the complexity of collaborative business although some standardisation have already been achieved in this area [31].

In this research, we will undertake a thorough investigation on the problems of SOA based collaborative business process management in terms of security policy specification, verification and enforcement.

The first research aspect lies in cross-organisational business collaboration requirements analysis within access control context. We start with a thorough requirement analysis on access control for service based business collaboration. We will adopt a scenario-based analysis approach. A scenario is "a brief description of an event that is both process-focused and user-centric" [32]. Scenario-based methods can be deployed to catch and analyse users' behaviours and interactions with the target systems. Different scenarios represent different kinds of situations and focus on different processes and users. The research results from software engineering, requirements engineering have shown that scenario-based analysis approach is an effective method to model users' behaviour and capture system profile [33]. The scenarios will be gathered from a variety of academic and industry projects from various countries and several industry sectors. The requirements on high level business strategy can be transformed into security policies.

A thorough study has been carried out to identify collaboration patterns and their requirements for consistency and comparability checking and policy integration.

The second research aspect is the development of a mechanism that can be used to compare consistency and comparability of different security policies from collaborating services based on collaboration patterns at design time, and to verify and enforce the agreed (integrated/collaborative) security policy at run time.

In role-based access control, users are assigned roles, and roles are associated with permissions or sets of operations. In the service oriented computing environments, users access data or perform tasks via services invocations. Each service is associated with a number of operations on data elements.

Individual service may have its own authorisation requirement. A coordinating service may need to exchange policy and credential information as well as managing the operation details. To deal with these issues, we developed solutions to realise security policy collaborative business process environment. This description includes service security capability and security constraints. Security capability describes the security features of a Web service such as name of the service requestor, a set of credentials, or a set of particular parameters required to invoke the service or role performed by the service requestor. On the other hand, security constraints refer to a set of conditions that a Web service could impose on another Web service in order to cooperate with it. Based on these descriptions, we develop a method to check the security constraints of the individual Web service to determine whether they are compatible to the specified security requirements. We also propose to build an authorisation model for expressing different access control policies and constraints. The model includes collaborative access control rules and Role Dependency Tree (RDT). The RDT is built upon the requirements of collaborative business process in terms of access control.

The third research aspect is the development of a Web service understandable specification that will be used to monitor, detect, and manage the policy of collaborating parties so that the policy alignment and compliance can be maintained. We also investigate how to employ and integrate this access control technique with existing Web services and security standards.

When collaboration opportunity arrives, questions may be raised such as: is it possible for collaboration under the current involving parties' authorisation policy? Whose policy shall be accepted and made it available to the end user? Whether and under what conditions a service is allowed to be forwarded to other parties? Furthermore different types of business collaborations exist in terms of the way collaboration is carried out, which may require different authorisation

control support, and the decision rules to determine consistency and comparability of partners' security policies. In [35] several access control patterns are identified such as service propagation, service composition, service outsourcing, etc.

Role Based Access Control (RBAC) as an access control mechanism has been widely accepted in the business world [23]. In RBAC, users are assigned with roles to process messages or perform tasks [21] [22]. However, in business collaboration environment, role assignment or modification are more complicated and prone to error because different parties and services are involved. For example, due to the peer-based collaboration nature, incorrect role assignment or modification may occur in any parties' services, or messages transferred from one organisation may be processed by unqualified roles in other collaborating business partners. Therefore, verification mechanism for such variation of role authorisation is critical to manage secured message processing in business collaboration.

Consistency in the security policy model refers to the alignment and compliance between partners' policies. Key to the facilitation of consistency is the understanding that whenever a change happens, it must be propagated to the right partners. Therefore mapping rules need to be specified between relevant elements of the involving policies. Only after these mapping rules are properly specified, can changes be detected and change reaction is a matter of re-negotiation and transferring the collaborative security policy from inconsistency to consistency status. To this end, techniques and algorithms will be proposed to identify an access control conflict (or a deviation to a policy) and appropriate actions which need to be taken.

Both computer-based automatic processes and human based manual interactions are taken into consideration. In the current business practice, legacy

business application, Web service based applications, pure human interactions and BPEL4People standardised human activities are involved and mixed. We need to take these users into account when designing the collaborative business process.

Access control patterns to be formally analysed, defined, and rules to be developed to identify the patterns. We define different levels of compatibility and consistency based on the identified access control patterns, and determines the acceptable and negociatable cases.

We exploit a role authorisation model (Role-Net) to provide such verification mechanism by introducing a reliability property named as Role Authorisation Based Dead Marking Freeness and an algebraic verification method [36]. Through this verification, unsecured message processing in terms of authorisation policy conflicts can be detected in business collaboration. We provide a mechanism to dynamically determine the required roles for each service according to authorisation policies.

This security mechanism will ensure that the policies between the two different services are conflict-free when a new authorisation terms is detected, verified and reinforced. Our goal is to compose only those Web services that are compatible with respect to the security requirements.

In order to analyse the system architecture and dynamic properties of the system, we will adopt an effective modelling technique, Petri Nets, to model components for dynamic, complex service analysis and synthesis. The theoretical analysis of Petri Nets has been transformed into the BPEL process, which provides an effective language for implementation.

The fourth research aspect of this study is the prototype design of secured SOA system within the proposed framework. The practical issues in the real applications will also be taken into consideration. Such security enhanced

prototype will not only provide information protection in computer science research scenarios, but also benefit business and finance field in how to secure their IT investment. In this thesis, we implement a prototype for access control capability of the proposed framework. The prototype is built on .NET framework with the latest version of WS-BPEL support.

1.6 CONTRIBUTION OF THE THESIS

The business world is changing every day, new legal requirements, changes in strategy, reorganisation of partnership, etc. Organisations need to adapt themselves to these changes accordingly. Efficient, effective and dynamic collaboration among business partners will be an advantage over competitors. The reorganisation of requirements is a first step to achieve this goal.

As the first step, we review a variety of representational scenarios and modelled these scenarios iteratively by Petri Nets. Based on these scenarios, we proposed the description and explanation of a set of requirements for collaborative business process modelling. These requirements are illustrated from both collaboration levels, namely strategic, organisational, transactional, operational levels and abstraction levels, namely ontological, conceptual, functional levels. The identified requirements provide a comprehensive understanding for practitioners in this area. Successful satisfaction of these requirements can lead to harmony business-IT alignment in business process modelling.

Secondly, we discuss the security rules in business and discussed their general and advanced characteristics. Currently, access to information is most often approached from a simplistic perspective of specifying what other users of the particular system can do to the information (in terms of access rights). These access rights are specified and enforced by many different technologies, with varying degrees of compatibility. It can be seen from the above that the current

business practices involve the propagation of information between organisations. Agreements (and mechanisms) for propagating such information needs to be an accompanying process to understand and enforce the security policies of all involved parties.

Resulting from above discussion, we gain the view that a security rule in business collaboration scenario must be understood by business people, which is intended to assert business structure or to control the behaviour of the business processes. It is associated with a precise schema and it is declarative in nature. In the perfect world, it can be easily made communicatable, executable and easily modifiable. Each rule furthermore has several characteristics that help facilitate its management tasks such as status, version, documentation, and so on.

Thirdly, we have proposed our BEPL4RBAC authorisation specification which supports the access control capability in business process environment. The BPEL4RBAC extends the classical RBAC model with organisation and business process elements appended. These two elements are essential for representing access control information in business process scenario. The BPEL4RBAC policy language is also formally defined to describe authorisation information. The access control and authorisation requirements illustrated in BPEL4RBAC model can be mapped into this policy language. All these information are integrated with WS-BPEL seamlessly. The system architecture investigates the feasibility of BPEL4RBAC. With the separation of Access Enforcement Module and Access Decision Module, access decision strategies and security policies can be developed by physically isolated users or organisations. These strategies and policies might be changed very frequently according to the real world need. Thus, BPEL4RBAC system ensures the availability and performance scalability in heavy duty business process environment.

As a WS-BPEL compatible extension, BPEL4RBAC extends its ability from both RBAC side and WS-BPEL side. The greatest advantage of BPEL4RBAC over others is the high compatibility with WS-BPEL standard since BPEL4RBAC policy language is an extension of latest WS-BPEL specification. This ensures the access control functions can be seamlessly integrated into WS-BPEL. The system architecture also provides the adaptability with other security standards to enhance its security level further. Moreover, the extensibility of BPEL4RBAC is not limited to XACML or WS-Policy based standards as long as they can be adapted in accordance with WS-BPEL.

In the next step, we improve the access control policy with broader security policies by taking consideration of human activities. The RBAC model is extended with service element which is essential in Web service processable business functions. On top of the extended RBAC model, we extend from WS-BPEL side to accommodate access control capability. The proposed adapter can integrate the security constraints into WS-BPEL and BPEL4People seamlessly. Besides the compatibility with these standards, existing legacy IT resources, such as XACML based security policies can also be mapped onto our proposed architecture which provides better aid for SOA migration.

At last but not least, we propose a role authorisation model, Role-Net, for the authorisation verification of business collaboration. Currently, classic RBAC based approaches cannot provide model to simulate role authorisation in business collaboration, nor verification mechanism to enforce collaboration reliability in terms of authorisation policy. In this thesis, we provide a role authorisation model (Role-Net) to verify authorisation policy based business collaboration reliability. A reliability property based on Role-Net is also defined and discussed. The mechanism on how to dynamically determine the required roles for each service can be designed by exploring role based authorisation policies. A policy-based specification integrated with existing policies based on Role-Net.

1.7 ORGANISATION OF THE THESIS

In this chapter, we provide a brief overview of the main topics of Web service based BPM, consequent security concerns in section 1.1. After illustrating a motivating scenario that inspired our work, we sketch several current research challenges in BPM area and contrasted them against the requirements in achieving secure business collaboration. The research scope and methodologies are described on how to tackle these challenges.

In the next chapter, we will fully discuss the related works together with the comparison and contrast of existing methods that address these issues. We investigate if and how current works can help to meet these objectives. The methodologies in major recent works of business process management, service-oriented architecture and role based access control areas are thoroughly discussed.

In chapter 3, a scenario-based requirements analysis approach is proposed. Making the requirements crystal clear is the first step in advancing the research tasks. By reviewing a variety of representational scenarios, we modelled these scenarios iteratively by Petri Nets. Then, we propose the description and explanation of a set of requirements for collaborative business process modelling.

Creating and maintaining and the dependencies, among business partners, remain as challenging tasks. How to ensure the consistence among partners at various levels of collaboration? How to capturing different requirements that might potentially contradict with one another? Chapter 4 answers these questions by introducing the security rules in collaborative business process.

BPEL4RBAC, the specification for enforcing access control in WS-BPEL and BPEL4People, is described in Chapter 5 and 6. In Chapter 7, we provide a

mathematic foundation, Role-Net, to verify the security rules in run-time. The specification and reliability property based on Role-Net is presented.

The architecture of our prototype system is introduced in Chapter 8. The last chapter, Chapter 9, provides a summary regarding the advantages and known issues in this research.

CHAPTER 2

LITERATURE REVIEW

In the previous part we have presented a big picture of this research on what are the identified requirements of business collaboration in security context and how to achieve collaborative business process management in a secure manner.

In this part we present the full literature review that led us to further strengthen the outlined requirements. In this review we will particularly focus on those works that have focused on collaborative business process development and design. The purpose of literature review is to gain insight in what research has been done already; in turn this process enabled us to identify useful ideas, unsolved issues and shortcomings in current methods. To this end we analyse the related literature in context of the research objectives we predetermined in section 1.5.

Accordingly, the remainder of this chapter is structured as follows: in section 2.1 we explore the literature on the context in which business collaborations take place. Next we investigate current proposals to capture this context which describe business collaborations in section 2.2. Subsequently we review the Web service related standards in section 2.3. Following that in section 2.4 we discuss WS-BPEL and BPEL4People in details. Access control methodologies are introduced in section 2.5. We also discuss the important observations to extrapolate Petri Net together with related technologies.

2.1 BUSINESS PROCESS MANAGEMENT

From the business point of view, the phrase *business process* can be traced back to 1960's. Originally, business process referred to a sequence of activities and the industry only focused on how to produce more products. In the

movement of business management, more factors have been appended to business process, such as cost and quality. Business process became one of the main considerations for business administrators and the management of business process turned into a research task [37].

In this research, we adopt the definition that is acceptable from both business and technical fields. “A business process is a set of logically related tasks performed to achieve a well defined business outcome [38].”

Twenty years ago, the idea of Business Process Management (BPM) has been introduced as a weapon to sustain or gain competitive advantages. BP Management aims to improve, optimise and adapt existing business processes through structured approaches in an organisation. BP Management cooperates closely with other methodologies from management science, such as Total Quality Management, Six Sigma, Performance Management etc. Another point from business view is that the management of business process will create value for both organisation and its partners since standardised and reusable processes will lowering the cost among them. [39].

To achieve coordinated and standardised processes, the first step is to define it. This is actually a modelling activity which involves identifying the tasks in business activities and mapping the tasks to operators and users. For many years, efforts in information systems modelling of the organisation behaviour help to realise the process modelling approaches [40].

At this stage, Business Process Modelling comes into being. Business Process Modelling has the same acronym BPM with Business Process Management. From a system engineering perspective, BP Management includes whole lifecycle of business process while BP Modelling is a first step in that procedure. The relationship of business process, BP Management and BP Modelling can be shown as:

Figure 2-1 Relationship of BP, BP Management and BP Modelling

In order to maintain competitiveness, organisations need to cooperate with existing and potential partners smoothly and dynamically to provide services to customers on the fly. Business collaboration has never been as important as today. In the collaborative business environment, however, BP Management and Modelling need come across the organisational boundaries and consequently face new challenges.

For example, the cooperating partners might have established their information system separately many years ago. When they work on a project together, even if the standardised processes have been negotiated successful, their information systems cannot understand each other since they were heterogeneous systems. Thus, more manual operations would be involved and cost more. More partners involved, the case will be more complex to manage.

Therefore, we need a paradigm to support the composition and implementation of cross-organisational collaborative business process [2]. In the past few years, Web service technology based service-oriented methodologies have emerged to solve this kind of problems.

2.2 BUSINESS COLLABORATION

Business collaboration is about cooperation between organisations by linking their business processes and exchanging information in order to achieve some shared goals and benefits. As we discuss the business collaboration in Web

service and BPM context, so in this research we treat collaborative business process management as an interchangeable phrase of business collaboration.

IT systems have been deployed to make business collaboration more efficient by business process automation. In 1990s, such IT-based coordination system was typically facilitated with Enterprise Data Interchange (EDI) for large organisations [41] [42]. EDI and the underlying technology could only fulfil limited requirements and possibilities. The relatively high costs also prevent EDI from wide spreading in the enterprise world [43].

In the middle 1990s, workflow based solutions were introduced and implemented for business management systems. The Workflow Management Coalition (WfMC) [44] founded in 1993 as the first industrial consortium. WfMC aims at promoting framework and interoperability for open architecture workflow management. Workflow based methodologies have almost concentrated on production like systems. In this period, rigid processes have also received most attention on workflow based automation of business processes.

These systems provide a standardised way of defining processes with a set of structured activities that can be subsequently executed and monitored from organisation's perspective. Generally, they have automated instances derived from pre-defined models for actual business enactment. Such approaches are suitable for standardised intra-organisational and inter-organisational workflows where processes are highly stable and well-defined. As we described in the chapter 1, however, Web service provides unlimited potential cooperation opportunities for hidden partners. Purely workflow approaches are ill-equipped in this scenario due to their rigid and centralised characteristics to describe and implement the more unstructured and dynamic oriented business collaboration.

Web services based solutions make interoperability more cost effective and manageable in terms of isolated and heterogeneous IT systems. In addition,

organisations have adopted IT-based solutions to coordinate the interactions between their automated business processes when collaborating with other parties in order to gear up semi-automated, complicated electronic transactions [45].

Organisations are typically part of inter-organisational and intra-organisational structures. The intra-organisational role performs its own internal business processes. While as a partner of cooperation between organisations, inter-organisational role works together with its partners to achieve some shared business benefits. These usually involve some kinds of information exchange with other organisations, these inter-organisational activities, in turn, influence the internal processes of these parties.

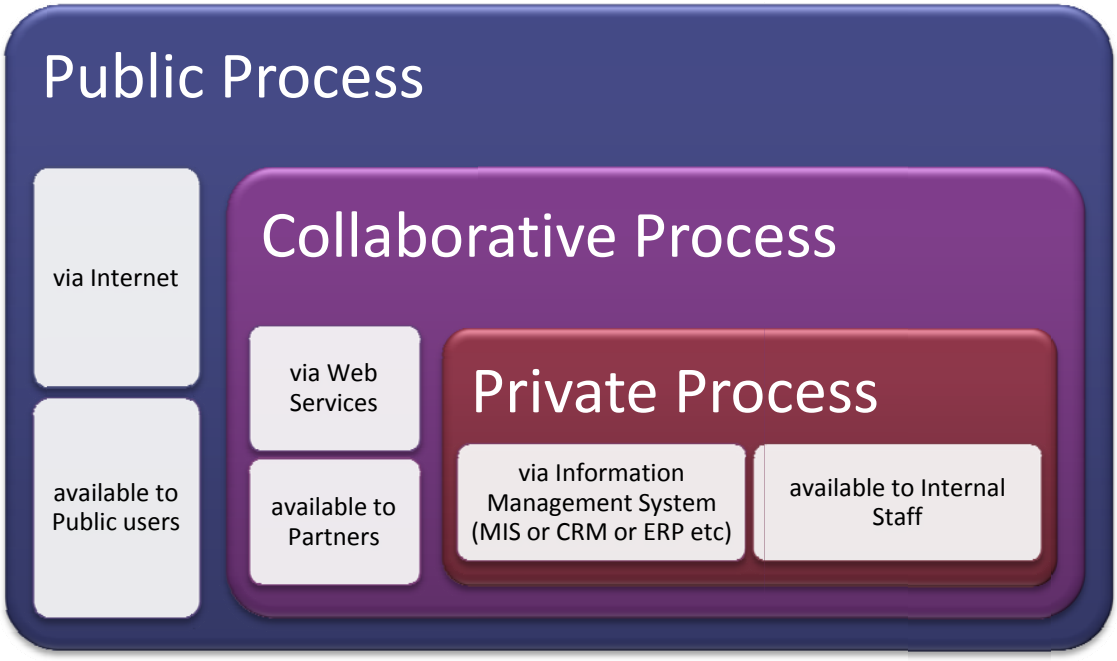

Figure 2-2 Business Process Layers

Business collaborations are often referred to as collaborative processes (or inter-organisational processes), whereas internal business procedures are often called intra-organisational processes (or private processes). The publicly available

business services are released as public processes. The public process and collaborative process are ultimately handled and managed by private process.

Business processes are connecting link between the strategies and operational activities such as information systems, business applications. In nature, business processes are multi-dimensional as organisations must be able to work together both from a business and technical point of view. In addition, collaborative processes are exposed to different business partners in a different way which require certain support from private processes. On the other hand, private processes are also influenced by many factors across the organisation departments such as its available resources and activities [46].

2.3 WEB SERVICES AND SOA

W3C [47] defines "A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards."

The Web is a viable way for cooperating partners to deliver and retrieve services. Simple Web services can offer simple functions such as weather reporting and order status checking. While complex Web services can engage other Web services as components to complete business transactions such as travel planning and real estate broking. Building and consuming interoperable Web services has become the scheme for organisations to deliver services for customers and cooperate with partners [48].

Simple Object Access Protocol - SOAP is a standard for exchanging XML-formatted messages in the implementation of Web services. As an application

layer protocol, SOAP standardises the message transferring among the organisations into a common data format, and defines PRC-style and Document-style as the interaction models for message negotiation and transmission. SOAP is naturally wired with HTTP to receive and send transport protocol packets which allows for easier communication through firewalls [49].

Web Service Description Language - WSDL is an XML format for describing network service as a set of endpoints operation on messages containing either document-oriented or procedure-oriented information. The abstract definitions of messages are separate from their concrete instance. The messages are abstract description of the data to be exchanged. A port is defined by associating a network address with a reusable binding where port types state abstract collection of operations. WSDL is often used together with SOAP to provide Web services on the Internet [50].

Universal Description, Discovery, and Integration - UDDI is a platform-independent, XML based registry for organisations to list their available services of over the Internet. UDDI is designed to be inter-operated with SOAP and WSDL by defining a set of services supporting the description and discovery of (1) businesses, organisations, and other Web services providers, (2) the available Web services, (3) the technical interfaces to access these Web services. A UDDI business registration consists of three components: white pages which define the address and other key information of service identifiers, yellow pages which classify the information according to the industrial taxonomies, and green pages that describe the service including the technical specifications of Web service and pointers to the file and URL based discovery mechanism [51]. The relationship of the SOAP, WSDL and UDDI is depicted in the following diagram.

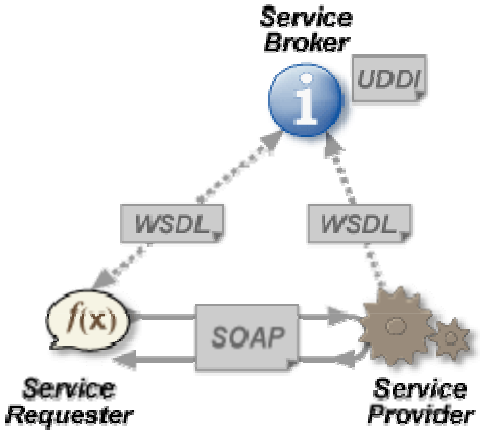

Figure 2-3 Web Services Architecture[1]

Through these three standardised components, the Web service infrastructure can implement the platform-independent interactions in loosely-coupled environment. SOAP encapsulates the message transferred among the organisations with the specific binding which is independent to the transport level protocols. WSDL provides standardised interface which hides the implementation and make heteronymous system communicate with each other. The service provider registers their services in the service registry which stores the necessary information of the service in the UDDI repository. As soon as the service requester makes a query in the service registry, related service information will be returned to the service requester. At this stage, the service requester will then be able to interact with service provider for further negotiation.

Business process can be represented by Web services. And on the other hand, Web service techniques can be used to implement business process. This relationship of Web services and business process management has been illustrated in [52]. The development of Web Service Composition methodologies provides technical foundation for BPM, especially for cross-organisational

[1] Figure Source: http://en.wikipedia.org/wiki/File:Webservices.png

processes. Varieties of service composition schemes and process modelling techniques have been developed [48].

Increasingly businesses are using the internet and the web to deliver tailored, on-demand services to partners and clients. Delivery of these services requires as key elements extensive networks of communication links between business partners and integration of the business processes of the partners. Service Oriented Architecture (SOA) has emerged in the past decade which provides a new way to use, re-use and manage IT systems.

In contrast to traditional application-centric view, applications in SOA paradigm are concerned with how to expose services and which services to expose. Service is the core idea in SOA, which is "functionality encapsulated in a form that is readily consumable by other applications and services [2]". SOA considers IT systems as a collection of reusable services rather than a collection of static applications. Consequently, SOA lowers the technology barriers of inter- and intra-organisation business process management.

2.4 WS-BPEL AND BPEL4PEOLPLE

Today's organisations are facing higher rate of changing market conditions, new competitive threats, new customer requirements ant etc. All of these situations are driving the need for a quick respond IT infrastructure in supporting new business models and requirements. In the real world, however, most organisations are equipped semi-automated business processes which are composed from complex electronic transactions.

Service-orientated methodologies, associated with XML related technologies and standards, are applied to facilitate business collaboration with partners and customers. This emerging paradigm provides loosely coupled and distributed business services across organisational boundaries [1]. Compared with traditional

business applications, Web services aggregate isolated business functionalities in a standardised way that helps to achieve a significant reduction in development cost and easier deployment for participating business partners [7] [8]. Business process based collaboration is constructed by combining Web services through one of the process specification languages. WSBPEL is one of these languages that provide the full set of syntax and notations for Web service based business processes.

2.4.1 WS-BPEL

WS-BPEL 2.0 [53] fills this requirement gap and covers the ideas of two rivals, WSFL [54] and XLANG [55], developed by IBM and Microsoft respectively from 2001. WS-BPEL, initially named BPEL4WS, is built on top of several Web services and XML standards, including SOAP [49], WSDL [50], UDDI [51], XML Schema [56] and XPath [57].

For the specification of different sections, WS-BPEL relies on Web Service Description Language (WSDL) [50], which mainly defines the functional characteristics of Web services interfaces. The non-functional part is addressed in a generic policy language WS-Policy [25]. WS-Policy also based on several standards to depict security requirements in Web services environments, including WS-SecurityPolicy [58] and WS-SecureConversation [59].

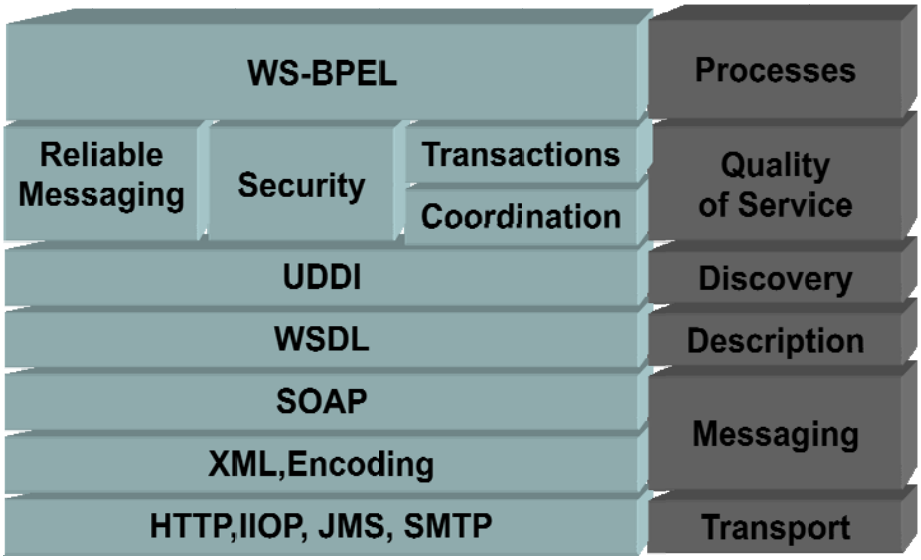

Figure 2-4 WS-BPEL Building Blocks[2]

Normally, Web service interactions can be described in abstract business processes or executable processes. Abstract processes depict business interactions by explicitly specifying the message exchanging behaviour of involving partners. There is a separation from the collaborative and private parts of the business process. This separation allows organisation to keep their internal business activities' implantations secret. Abstract processes serve as a descriptive role which may have more than one use cases in the business interactions. Executable business processes model actual behaviour for partners without separating external aspects of the processes from internal activities. This difference between abstract and executable business processes is expressed solely in the availability of different sets for data handling [53].

2.4.2 BPEL4People

However, the common scenario of a business process depends on a person to fulfil a certain human task as a part of process activity. This important issue has not been covered by WS-BPEL which means human involved process activities cannot be specified using WS-BPEL [53]. Motivated by this shortcoming, two

[2] Figure source: http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html

specifications have been released in June 2007 that address Web service domain and cover the integration of human tasks: WS-BPEL Extension for People (BPEL4People) [60] and Web Services Human Task (WS-HumanTask) [61].

2.4.3 Web Service Security

Web services are building block for business processes. The existing security standards for Web services should be also taken into consideration when providing security features for WS-BPEL. A variety of security standards have been proposed for Web service architecture at different levels.

WS-Security [24] is the foundation for building secure Web services. It aims to realise message-level security for exchanging SOAP messages. Based on WS-Security, WS-Policy [25] provides a general purpose model and corresponding syntax for expressing Web services policies. The WS-Policy is constructed by a set of messaging-related assertions. The assertions can be defined in a set of security policy assertions related to supporting the WS-Security specification, such as WS-SecurityPolicy [58] from OASIS, WS-PolicyAssertions [62] from IBM and WS-PolicyConstraints [63] from SUN. and In addition, WS-PolicyAttachment [64] is introduced to describe how to attach these policies to Web services [15].

Besides WS-Policy based architecture, there are some other XML-based languages that can be used to express Web services policies, such as SAML [27] and XACML [65]. With these languages we can specify access control rules that protect Web services from unauthorized access and ensure integrity and confidentiality of exchanged messages [15].

2.5 ACCESS CONTROL

Currently, access to information is most often approached from a simplistic perspective of specifying what other users of the particular system can do to the information (in terms of access rights). These access rights are specified and

enforced by many different technologies, with varying degrees of compatibility. It can be seen from the above that the current business practices involve the propagation of information between organisations. Agreements (and mechanisms) for propagating such information needs to be an accompanying process to understand and enforce the security policies of all involved parties. This requires not just a mechanistic application of the sum of all policies (as such an approach would likely fail with policies being applied out of context) but a process [13] that results in a secure handling of information and accessing services satisfactory to all parties.

There are various access control models addressing different aspects in the access control domain. Role based access control (RBAC) [5] has emerged in 1990s. By associating permissions and roles, RBAC allows the access control model in the same way that maps naturally to an organisation's structure, and the concept of a role is in correspondence to an organisational position. Several constraints may apply to an RBAC model. For example, Separation of Duty (SoD) is one of the well-known security principles which requires two or more different people to be responsible for the completion of a task or set of related tasks [14]. To protect the interest of organisations, the conflicting roles must not be assigned to the same user in a business process [15]. While in some cases, the same user might be required to perform two different activities. This is considered as a binding of duty constraint (BoD). BoD and SoD are typical security policies. These security policies are embodied in RBAC to specify these access control constraints.

The Task-based Access Control (TBAC) was built on the RBAC, which models access control from task oriented perspective [16]. TBAC approach separates system level activities to support scalable and reusable access control models. Organisation based Access Control (OBAC) [17] model aims to share specific data and functionality with collaboration partners. The specification of the security

policy is completely parameterized by the organisation in order to handle simultaneously security policies associated with different participating partners.

The RBAC, TBAC and OBAC methods provide efficient and effective access control capability for current application-centric systems. In the SOA era, most security issues arise from the interaction among applications rather than inside of applications. The application based access control mechanisms, therefore, are no longer suitable for security in service-centric IT systems. There is no comprehensive approach to secure SOA. Therefore traditional access control cannot provide adequate shield for SOA due to its complexity [18].

2.5.1 Role Based Access Control

Access control mechanisms aim at protecting information and resources at different levels of granularity by configuring and enforcing access policies [66]. In RBAC, the security policy does not directly grants permissions to users but assigned to appropriate roles on the basis of specific policy [17]. Consequently, the assignment of users to roles is separated from the assignment of permissions to roles [67].

Several constraints may apply to an RBAC model. For example, Separation of Duty (SoD) is one of the well-known security principles. By partitioning related tasks and privileges, SoD reduces the possibility of fraud or errors [15]. To protect the interest of organisations, the conflicting roles must not be assigned to the same user in a business process [15]. While in some cases, the same user is required to perform two different activities. This is considered as a binding of duty constraint. The security policy is embodied in RBAC to specify these access control constraints.

Although the concept of role has existed for a long time in systems security, the work presented by Sandu in [5] has prompted a renewed interest in this approach. This greatly simplifies security management [6]. RBAC model is now

adopted in many commercial products to different degrees since access control is an important requirement of information systems. RBAC was found to be the most attractive solution for providing security characteristics in inter-organisational business systems [68]. Moreover, it would be much easier for organisations to enhance security protection from existing RBAC based policies.

2.6 VALIDITY OF BUSINESS COLLABORATION

Organisations rely on their business processes to embody their existence. It is therefore a vital task that these processes are modelled and carried out in a manner conform to requirements. Due to the complexity of business collaborations, however, verification of these characteristics poses organisations with an extra challenge. Moreover, the dynamic business collaboration environment further complicates organisations not only verifying their own business processes, but also to ensure that their business partners conform the same way. In summary, this requires mechanisms for the formal verification for collaborative business processes.

Process algebras and calculi are another group of model checking based validation approaches which provide a tool for the high-level description of concurrent interactions and communications a series of independent processes. There are many languages and dialects in this area, such as Communicating Sequential Processes (CSP) [69], Calculus of Communicating Systems (CCS) [70] and Language of Temporal Ordering Specification (LOTOS) [71].

2.6.1 Petri Nets

Petri Net is a net theory introduced by Dr. Petri in 1962 for distributed system modelling [72]. A Petri Net graphically represents the structure of a distributed system as a directed graph where nodes can be distinguished in places and transitions. Places may contain a number of tokens. Transitions represent the

move from one place to another, where places and transitions are connected via directed arcs. The advantages of its graphically and mathematically founded modelling formalism with various algorithms for design and analysis make it a good candidate for modelling the collaborative business transactions [73].

Definition 2-1

$A\ Petri\ Net\ is\ a\ tuple\ N = (P, T, F), where$:
$\boldsymbol{P}\ is\ a\ set\ of\ places\ graphically\ represented\ as\ circle.$
$\boldsymbol{T}\ is\ a\ set\ of\ transitions\ graphically\ represented\ as\ dark\ bar.$
$P \cap T = Null$
$F = \{P \times T\} \cup \{T \times P\}$

$is\ the\ flow\ relation\ between\ places\ and\ transitions.$

Marking of a Petri Net is an allocation of tokens to the places of the net formally defined as a function $M: P \longrightarrow R^{|P|}$, where $R^{|P|}$ is $|P| \times 1$ vector. The marking reflects the state of the Petri Net after each firing. In a marking k, if a token in p, then $M_k(P) = 1$, otherwise $M_k(P) = 0$. While M_0 is the initial Marking of Petri Net.

Petri Net graphs can be formally verified for several properties which applied to business process verification. A place represents a process state in business process. Transitions then govern how the process moves from one state to another. The business activities treat taking tokens as input and producing tokens as output in the business process management engines.

Definition 2-2

$A\ Marked\ Petri\ Net\ is\ a\ tuple\ S = (N, M_0)$
$where\ N\ is\ Petri\ Net, M_0\ is\ the\ initial\ marking$

Through initial marking, we observe that the Petri Net can reach a series of markings according to the firing of transactions. A transaction t is enabled under

M written as $M[t>$, if $\cdot t \subseteq M$, where $\cdot t = \{y \in P | (y,x) \in F \cap x \in T\}$. A firing sequence among multiple transactions $t_i (i = 1 \dots n)$ can be written as $[t_1 > M'[t_2 > M''$ …. , where the firing sequence.

Through investigating the marking M, we can deduce the characteristic of Petri Net using several analysis tools, e.g., $Incident\ Matrix$ and $Transitive\ Matrix$. Here we will introduce the two matrixes in detail.

Definition 2-3

For the Petri Net N with n transactions and m places, the incident matrix $A = [a_{ij}]$ is an $m \times n$ matrix [74] and its typical entry is given by

$$a_{ij}^{+} = a_{ij}^{+} - a_{ij}^{-}$$

where $a_{ij}^{+} = \begin{cases} 1 & (x,y) \quad In\ F \\ 0 & (x,y)\ Not\ In\ F \end{cases}$ $\qquad a_{ij}^{+} = \begin{cases} 1 & (y,x) \quad In\ F \\ 0 & (y,x)\ Not\ In\ F \end{cases}$

given that $x \in T\ and\ y \in P$

Definition 2-4

A labelled place transitive matrix [75]

$L_{BP} = A^{-}\ Diag(t_1, t_2, \dots., t_n)(A^{+})^{T}$

where $A^{-} = \left[a_{ij}^{-}\right] and\ (A^{+})^{T} = [a_{ij}^{+}]^{T}$ (T represents transpose matrix),

t_i (I = 1,2,……,n) is

$$|t_i| = \begin{cases} 1 & fire\ t_i \\ 0 & not\ fire\ t_i \end{cases}$$

Also we use L_{BP}^{*} to extend the original transitive matrix in which t in L_{BP} is replaced by t/d in L_{BP}^{*} if a transition t apprers d times in the same column of L_{BP}.

Coloured Petri Net is an extension of Petri Net in which tokens are assigned with values. In business transaction, various types of messages can be transferred within or across organisations. Therefore, the message types can be represented as colored tokens in CPN. Another extension of Petri Net is Hierarchical Petri Net (HPN) in which different levels of abstraction and refinement can be specified. In this thesis, we call it net refinement or refinement when a transition or place can be represented as one or more HPNs.

2.6.2 Petri Net with Verification

In this section, we will illustrate the general reliability properties for Petri net associated with the verification approaches on them.

Properties

Reachability: the possibility of reaching a given state through the firing sequence.

$$\forall\, M_k \in M, \exists\, M_{k-1}, M_{k-1} \hookrightarrow_{\alpha} M_k$$

where M is the set of state (markings), $\hookrightarrow$ is the performance of transition firing, α is the firing sequence.

Boundness: the maximal the minimal numbers of token in one place at given state.

$$\forall_p \in P\,, minium \leq |p|_{Token} \leq maximum$$

Safeness: only on token in each place at given state.

$$\forall_p \in P\,, |p|_{Token} = 1$$

Dead Marking Freeness: All markings having enalbed transitions.

$$\forall M_k \in \mathbf{M}, \exists t \in T \;\; M \rightarrow t$$

where → is the performance of transition enabling.

Soundness: this properties is only suitable for the flow-typed Petri Net model, which has one input place i and one output place o linked by the movement of token. When the token reaches the output place from input place, there is no other tokens left in the net.

For every reachable state M, there exists a firing sequence leading from state M to state o

$$\forall M, i \ \hookrightarrow_{\alpha} \ M \implies M \hookrightarrow_{\alpha} o$$

State o is the only state reachable from state i with at least one token in place o

$$\forall M, i \ \hookrightarrow_{\alpha} \ M \cap M \geq o \implies M = o$$

There is no dead transition

$$\forall t \in T, \exists M_{k-1}, M_k \in M \quad i \hookrightarrow_{\alpha} M_{k-1} \hookrightarrow_{t} M_k$$

The Petri net provides various approaches on verifying reliability by desired reliability properties mentioned above, such as using state equation to evaluate the Reachability, and using transitive matrix to detect the Dead Marking.

M_k is reachable from M_0 in a marked Petri Net $S = (N, M_0), if \ B_f \nabla M = 0$, where $\nabla M = M_k - M_0$ and B_f is a given $m \cdot r \times n$ matrix in a Petri Net with n transitions and m places. r is the rank of incident matrix A of the marked Petri Net S. [74]

2.7 RELATIVE METHODOLOGIES

Research has been done in the area of role authorisation in business collaboration. Petri Net is a widely used technique for role authorisation modelling. We shall look into some of the representative work in these areas.

There are many research works [21] [15] focus on enhancing security features for business process management systems. Some of these works addressed on access control ability to current WS-BPEL.

Bertino, Crampton and Paci [21] developed RBAC-WS-BPEL and Business Process Constraints Language (BPCL) languages to address this issue. The RBAC-WS-BPEL is an adapted version of RBAC model with business process element introduced within. The authorisation specification is composed from authorisation schema, represented by XACML, and authorisation constraints, represented by BPCL which is an XML based language. However, XACML does not directly support the notation of roles, and hence it lacks some essential features in RBAC such as separation of duty and role hierarchy. BPCL tries to counter-balance the authorisation constraints in an XML-formatted language. But the integration of BPCL with WS-BPEL and XACML is a challenging task at this stage.

Liu and Chen [15] developed another extended RBAC model, WS-RBAC. Three new elements are introduced into the original RBAC model, namely enterprise, business process and Web services. The authorisation constraints are described in WS-Policy [25] and WS-PolicyAttachment [64]. However, these two standards are designed for message level Web services security. It is difficult to express some access control constraints on this layer such as role hierarchy and permissions. WS-RBAC enhances the ability of RBAC model in the business process environment. But it also increases the complexity of performing authorisation constraints in this architecture.

Knorr in [76] has proposed a role based access control method through Petri Net workflows. Role authorisation rights were granted according to the state of the workflow. The access control matrices were also deployed at this stage to define the role authorisation policies. However, the role authorisation issues causing unreliability of workflow were only detected at design time. The role

authorisation's conflicts and errors causing unreliable business collaboration at runtime still cannot be detected.

Although plenty of existing models and approaches have been presented which focus on managing reliable business collaboration in terms of role authorisation, they are still insufficient in: (1) describing role authorisation in business collaboration with regard to the organisation's peer nature; (2) detecting role authorisation errors and verifying business collaboration reliability in terms of role authorisation.

Business collaboration can become unreliable in terms of authorisation policy conflicts, for example, when (1) incorrect role assignment or modification occurs when the required role is inconsistent with the role assignment for a message in a service within one organisation, or (2) messages transferred from one organisation are accessed by unqualified roles in other collaborating business partners.

Current approaches cannot provide model to simulate role authorisation in business collaboration, nor verification mechanism to enforce collaboration reliability in terms of authorisation policy. In this research, we fill in the gap by providing a role authorisation model (Role-Net) to verify authorisation policy based business collaboration reliability. A reliability property based on Role-Net is also defined and discussed. Currently we are working on providing a mechanism on how to dynamically determine the required roles for each service by exploring role based authorisation policies. A policy-based specification will be developed based on Role-Net as well.

2.8 SUMMARY

This chapter has introduced many related works from which we can draw in our effort to develop an approach for the secure and dynamic collaborative

business process management. It is clear that business collaboration is urgently demanded from the business world. The support, however, is limited in the traditional workflow and process based approaches. The solutions to achieve this in an efficient and effective way are yet to be discovered. At the same time, there are several gaps that currently stand in the way of the successful. The contribution of the research presented in this thesis address both security perspective of Web service enabled business process and cross-organisation perspective of collaborative process oriented business collaboration.

The following chapter will begin with the requirement analysis of such solution as a first step.

CHAPTER 3

SCENARIO-BASED REQUIREMENTS ANALYSIS

In this chapter, we start with a thorough requirement analysis on service based business collaboration. We adopt a scenario-based analysis approach. A scenario is "a brief description of an event that is both process-focused and user-centric" [32]. Scenario-based methods can be deployed to catch and analyse users' behaviours and interactions with the target systems. Different scenarios represent different kinds of situations and focus on different processes and users. The research results from software engineering, requirements engineering have shown that scenario-based analysis approach is an effective method to model users' behaviour and capture system profile [33]. The scenarios will be gathered from a variety of academic and industry projects from various countries and several industry sectors. The requirements on high level business strategy can be transformed into security policies.

3.1 BUSINESS SCENARIOS AND RESEARCH APPROACH

The word scenario, defined as "a brief description of an event", has been widely used in many fields. Business institutions consider scenarios as possible outcomes from certain decisions. Scenario planning and thinking are generally used to calculate returns and control business risk. In computer science community, on the other hand, a scenario is "a description technique that is both process-focused and user-centric" [32]. Practitioners in Human-Computer Interaction (HCI) area have deployed scenario-based methods to catch and analyse users' behaviours and interactions with target system. Different scenarios represent different kinds of situations and focus on different processes and users.

Scenario Example 1: Motor Damage Claims Scenario. The CrossFlow project [77] has observed a motor damage claims scenario in insurance industry. This scenario is described from an insurance company's view and other parties including the assessor, approved repairers, broker and road assistant company are also involved in. The normal insurance claim processes among these cooperating parties are introduced from organisational perspective as well as the exceptions to these processes. Existing information systems deployed in these parties are also described. All process-related forms are provided as data entities.

Scenario Example 2: B2B Insurance Partner Platform Scenario. Another example is insurance partner platform scenario in INTEROP project [78]. This scenario is also related to insurance industry, though, with a focus on the insurance platform and interoperability among potential partners. A B2B business model is proposed at the beginning of this scenario. Customer, sales partner, insurance company and sub service provider are main concerns in this business model. Then, the software architecture, based on distributed environment, is designed for this business model. Finally, the interoperability analysis identifies business processes on strategic, business, implementation and execution levels.

In collaborative business process modelling, the process and the user are our main concerns. The research results from software engineering, requirements engineering have shown that scenario-based analysis approach is an effective method to model users' behaviour and capture system profile [33].

We conducted a requirement analysis in business and IT alignment context. The scenarios were gathered from a variety of academic and industry projects which include various countries and several industry sectors. The selected scenarios were relevant from the perspectives of cross-organisational system users and real world industrial users.

3.2 BP MODELLING RELATED ACADEMIC AND INDUSTRY PROJECTS

We chose 33 scenarios from 6 projects in recent 10 years to conduct our requirements analysis. All of these projects have a focus on collaborative business process. ATHENA Project [79] aims to "enabling enterprises to seamlessly interoperate with others." In ATHENA, the collaborative business process modelling techniques followed both enterprise modelling and technical perspective. ATHENA focused on new and emerging enterprise models including virtual enterprises, fourth party logistics and efficient consumer response scenarios.

CrossFlow project [77] aims to "enable business processes to cross organisational boundaries and provide essential support for the virtual enterprise." CrossFlow considered contract as the foundation of dynamic virtual enterprise collaboration. A contract is a set of fully specified services in CrossFlow. Accordingly, a framework was designed which focused on contract requirements, establishment, enactment, modelling, language and matchmaking. The logistics and insurance scenarios were carefully described.

ECOLEAD project [80] intends to provide "a comprehensive holistic approach" which can materialize "networked collaborative business ecosystems". Based on some foundation theories and supported by IT infrastructures, ECOLEAD targeted at Virtual Organisation (VO) Breeding Environment, Dynamic VO and Professional Virtual Communities. Different topologies were applied to organisations' classification and a variety of scenarios were described in ECOLEAD project.

GLOBEMEN project [81] defined "a reference architecture for virtual manufacturing enterprises". With the involvement of international organisations from Australia, EU, Japan and Switzerland, GLOBEMEN cooperated on wider cultural environments. Main processes of manufacturing, which included sales, services, inter-enterprise management and engineering, were examined. The 11

scenarios described in GLOBEMEN were all manufacturing and engineering organisations.

INTEROP project [78] focused on the "domain of interoperability for enterprise applications and software". A variety of recent projects, standards were examined in INTEROP, as well as some off-the-shelf commercial products and technologies. Platforms and frameworks were the main concern in business process collaboration area. Health-care processes and insurance partner platform scenarios were carefully studied from these perspectives.

SPIDER-WIN project [82] researched on interoperability issues specifically for SMEs which need simple and efficient collaboration with "with low-level local software requirements" [82]. The message format, expressed in XML, has been devised to achieve this goal. Many SMEs scenarios from Italy, Spain and Poland were examined.

3.3 REQUIREMENT DESCRIPTION

As we introduced above, requirements analysis is an essential step from both business and technology perspectives, and as well as the alignment. This section will concentrate on collaborative process in business and IT alignment segment. Our focused area is shown in following table:

Table 3-1 Focused Area

	Business Perspective	Business IT Alignment	IT Perspective
Public Process			
Collaborative Process			
Private Process			

By reviewing these scenarios, we summarises them into two dimensions: collaboration and abstraction. Collaboration levels represent the enterprise architecture in cross-organisational business collaboration environment from high level decision-making to low level office workflows. There are four collaboration levels identified, that is based on the requirements analysis and partly inspired by [83] [84] [85].

Table 3-2 Collaboration levels

Requirements	Description
Strategic	Business goals of collaborative organisations. With the understanding of dynamic industry structures and organisations positioning
Organisational	Business context of collaborative organisations. With the determination of cross-organisational procedures and coordination mechanisms
Transactional	Business transactions among collaborative organisations. With the perception of seamless information frameworks
Operational	Business activities within transactions. With the observation of running activities and dynamic react to exceptions

In the motor damage claims scenario, the description of cooperating parties provides organisational procedures. And the information systems and process-related forms can be analysed from transactional and operational levels. In another case, the insurance partner platform is described from strategic and organisational levels. Abstraction levels describe the function generalization in the collaborative business process context from high level ontology notions to

low level function partitions. Three abstraction levels have been measured in the following table. The categorisation is derived from scenarios we examined and partly elicited by [86] [87] [85].

Table 3-3 Abstraction levels

Requirements	Description
Ontological	Metamodel of sharing semantics. With the proposition of underlying semantics and syntax
Conceptual	Agreements on concepts and notations. With the description of structures, composition and organisation properties
Functional	Implementation for business functions. With the definition of interactive methods and functions

3.4 PETRI NETS BASED MODELLING

In this section, Petri Nets modelling method is described with two examples from the scenarios introduced in section 3.3. Then, we will model the requirements identified from all scenarios into Petri Nets in ontological, conceptual and functional levels.

3.4.1 Petri Nets Modelling Methodology

Petri Nets were invented by Carl Adam Petri in 1962. Over the decades, Petri Nets were widely applied in the area of system analysis and design, especially for discrete and distributed systems. The scenarios introduced in section 3.2 is analysed and modelled by Petri Nets in this section.

According to [88] and [89], a channel-agency net is defined for these scenarios:

Definition 3-1

A Petri Net is a triple $N = (S, T, F)$ *iff S* and *T* and are disjoint sets.

S is a finite, nonempty set of so-called *channels.*

T is a finite, nonempty set of so-called *agencies.*

$F \subseteq (S \times T) \cup (T \times S)$ is a binary relation, the *flow relation* of *N*.

Channels, represented as circles (○). Agencies, represented as boxes (□). In Petri Net, each channel represents a passive system component. Each agency represents an active system component.

3.4.2 Scenario Modelling Example 1: Motor Damage Claims Scenario

As described in Section 3.3, the motor damage claims scenario can be modelled from functional level as following:

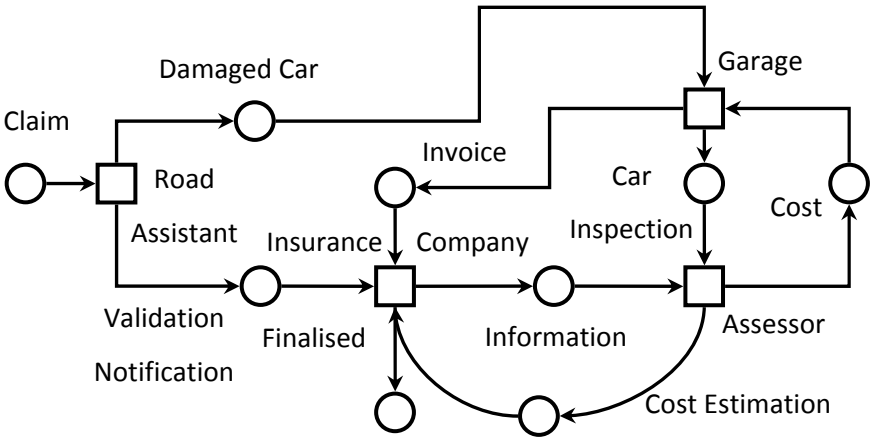

Figure 3-1 Motor damage claim Petri Net: Functional Level

3.4.3 Scenario Modelling Example 2: B2B Insurance Partner Platform Scenario

Another scenario B2B insurance partner platform, although explaining a similar process on insurance industry, it focuses on how to create business value

on a B2B business model basis. This scenario can be modelled as two Petri Nets in according to its description. The first Petri Nets is about business processes among cooperating partners and the insurance platform. This scenario can be shown in the following figure:

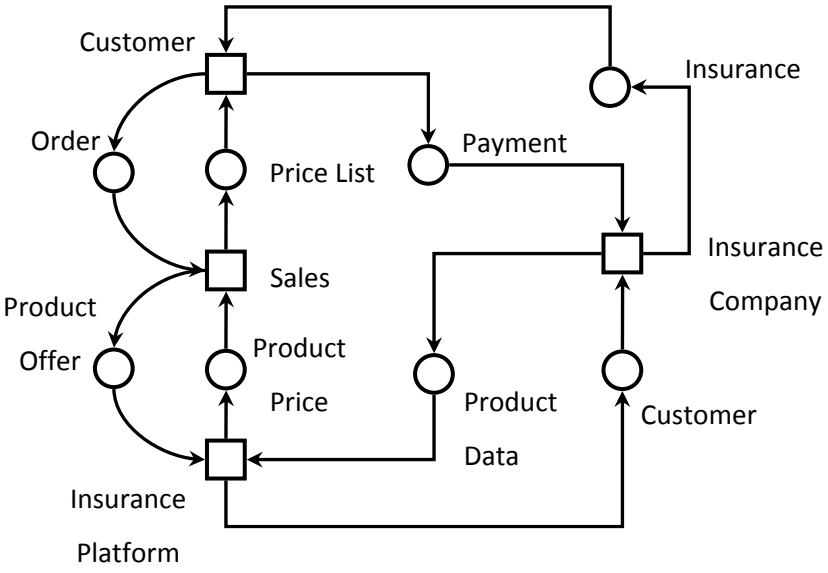

Figure 3-2 B2B insurance partner platform Petri Net: Conceptual Level

Another Petri Nets model on ontological level is based on its business strategies.

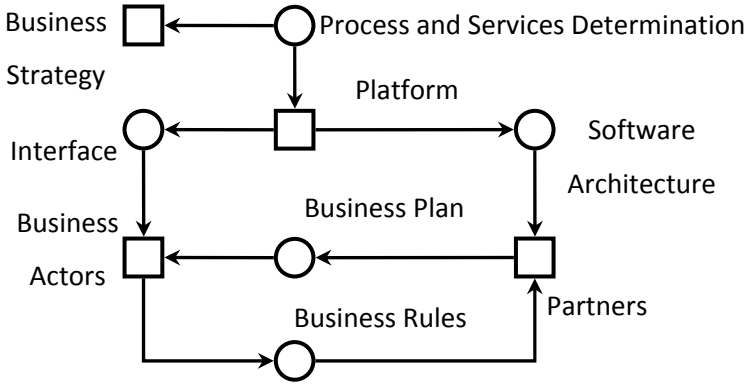

Figure 3-3 B2B insurance partner platform Petri Net: Ontological Level

3.4.4 Scenario Based Analysis

By analysing focused area of the 33 scenarios introduced in Section 3.2 from collaboration levels: strategic, organisational, transactional and operational. These scenarios can be categorised in the following table:

Table 3-4 Projects by collaboration levels

Scenario Name	Projects	Focused Collaboration Level			
		Strategic	Organisational	Transactional	Operational
CPFR	ATHENA	●		●	
Virtual Network	ATHENA	●	●		
Virtual Enterprise	ATHENA	●	●		
eProcurement	ATHENA	●		●	
Efficient Consumer Response (ECR)	ATHENA		●		●
Fourth Party Logistics (4PL)	ATHENA		●	●	
Logistics	CrossFlow		●	●	●
Insurance	CrossFlow		●	●	●
Electricity market	ECOLEAD	●	●		
Industrial districts	ECOLEAD	●		●	
Production network	ECOLEAD			●	●
Soccer team	ECOLEAD		●		●
Power plant	ECOLEAD	●		●	
Squads	ECOLEAD		●	●	
Adhoc networks in emergency management	ECOLEAD		●	●	●
Virtuelle fabrik	ECOLEAD		●	●	

Grid components in a service oriented infrastructure	ECOLEAD	●	●		
Verkko A (Network A)	ECOLEAD	●		●	
Policy Chains	ECOLEAD	●	●		
AGORA	GLOBEMEN	●		●	
C-Project	GLOBEMEN		●	●	
E3S (E-Service Support System)	GLOBEMEN		●		●
KCE	GLOBEMEN		●	●	
GAIA DEE	GLOBEMEN		●		●
GAIA SCM Consulting	GLOBEMEN	●	●		
NeOS Distributed Manufacturing	GLOBEMEN		●	●	
NeOS Maintenance	GLOBEMEN		●	●	
NeOS Renewal	GLOBEMEN			●	●
NeOS Sales Support	GLOBEMEN		●	●	
EPM DE Environment	GLOBEMEN	●		●	
Health-care processes	INTEROP		●	●	●
Insurance Partner Platform	INTEROP	●	●		
SMEs(group)	SPIDER-WIN	●	●	●	●

3.4.5 Modelling Approach

In order to analyse these scenarios by Petri Nets modelling, we define the elements in collaborative business process in more detail:

Definition 3-2

T_i ***i = (1,2,3,4,5,6,7,8,9)*** is finite, nonempty subset of ***T***.

T_1 = Strategy, T_2 = Interoperability; T_3 = Framework; T_4 = Syntax; T_5 = Business Context; T_6 = Agreement; T_7 = Procedure; T_8 = Syntax; T_9 = System;

S_i ***i = (1,2,3,4,5)*** is finite, nonempty subset of ***S***.

S_1 = Architecture, S_2 = Mapping; S_3 = Integration; S_4 = Interface; S_5 = Environment;

where:

$$\forall T_j \ \wedge \ \forall T_k \ , j \ \in \ S_i, k \in S_i \ \Rightarrow \ T_j \cap \ T_k = \ \emptyset$$

$$\forall S_j \ \wedge \ \forall T_k \ , j \ \in \ S_j, k \in T_i \ \Rightarrow \ S_j \cap \ T_k \subseteq F \neq \ \emptyset$$

Firstly, we model every scenarios from all projects listed in Table 4 by Petri Nets. Two examples are shown in Section 4.2 and 4.3. Motor damage claim Petri Nets model provides modelling knowledge on conceptual and functional levels. The B2B Insurance Partner Platform scenario provides modelling knowledge on ontological and conceptual levels.

Then, we merge and abstract from similar Nets interactively. The following is the Petri Nets (channel-agency net) that are derived from above scenarios. According to section 3.3, the scenarios were abstracted from ontological level, conceptual level and functional level and shown in Figure 3-4, 3-5 and 3-6.

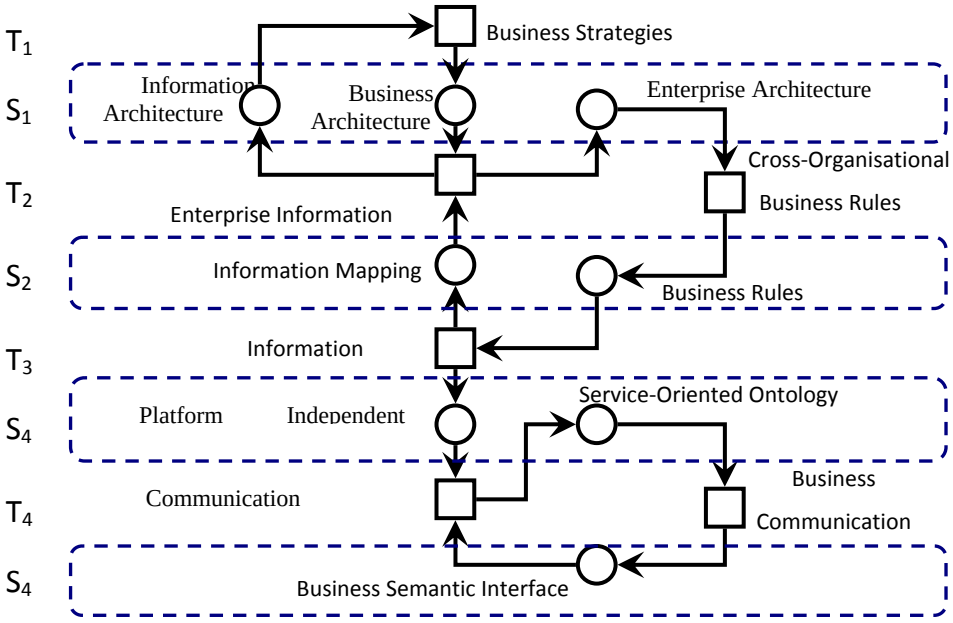

Figure 3-4 Ontological Level Petri Nets

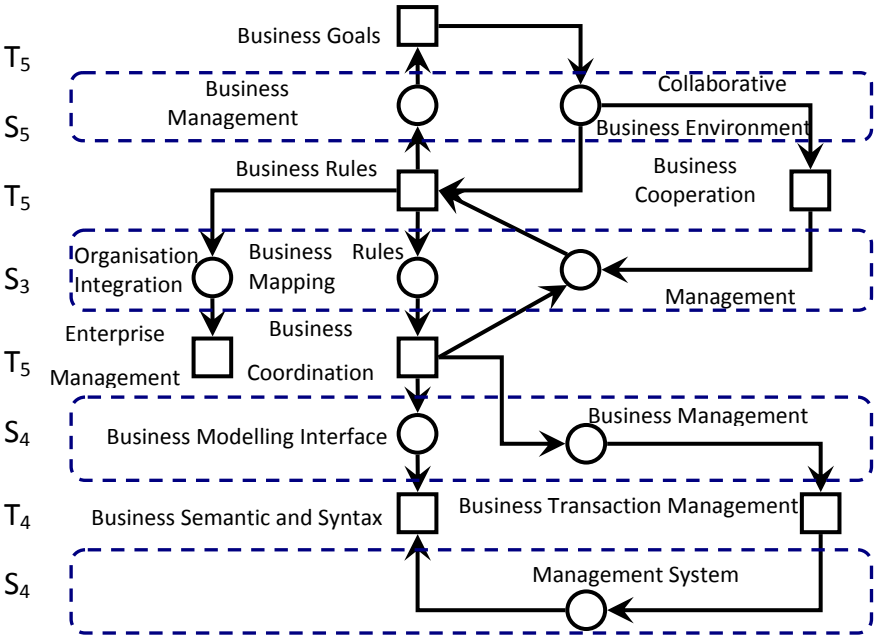

Figure 3-5 Conceptual Level Petri Nets

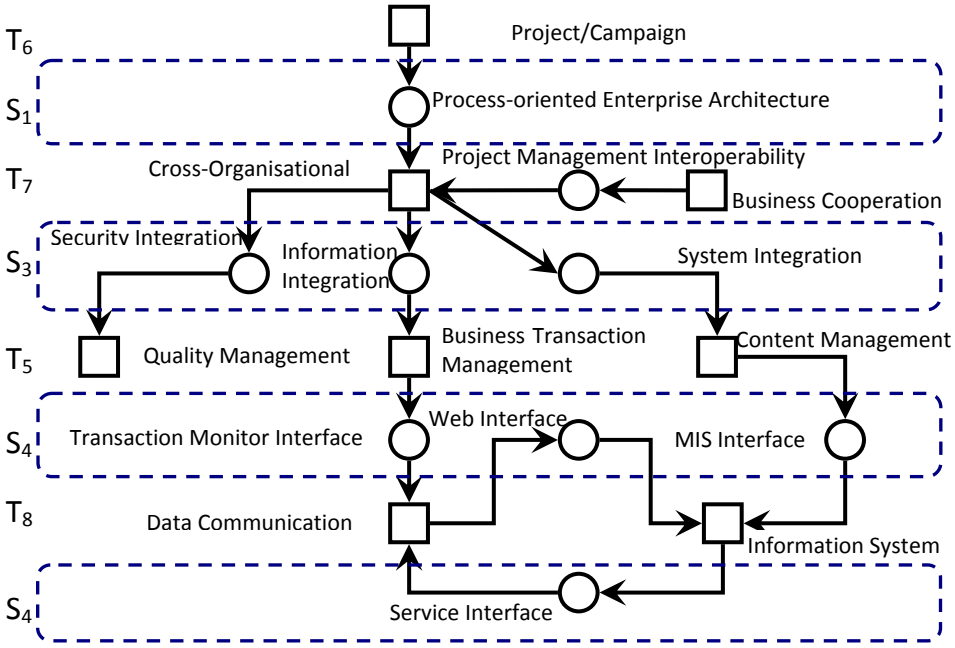

Figure 3-6 Functional Level Petri Net

3.4.6 Requirements for Collaborative Business Process

By analysing and modelling the scenarios, requirements can be elaborated from both collaboration and abstraction levels. We can now summarise the requirements for collaborative business process in the following table.

Table 3-5 Requirements for Collaborative Business Process

	Strategic	Organisational	Transactional	Operational
Ontological	To negotiate common cooperation strategies.	To maintain enterprise architecture coherence.	To sustain information framework consistence.	To provide a common communication infrastructure.
Conceptual	To determine business goals.	To obtain business rules.	To decide coordination mechanisms.	To define standard semantic and syntax.

Functional	To achieve projects' agreements.	To define cross-organisational procedures.	To implement reliable and secure business transactions.	To monitor and react to information transfer.

3.5 CASE STUDY: COLLABORATIVE LEGAL INFORMATION SHARING ON P2P NETWORK

Peer-to-Peer (P2P) information sharing attracts much attention from both legal and IT communities. P2P architectures have the potential to accelerate communication processes and reduce collaboration costs. A prototype, named vuCRN has been developed for researchers to facilitate document sharing legally. However, some legal issues are preventing P2P to realise its real power. In this case study, the design processes of our P2P-based collaborative legal information sharing system are introduced. First P2P information sharing architecture and legal issues are described. Then legal concerns and the next stage of the prototype vuCRN are analysed from ontological level, conceptual level and functional level respectively.

3.5.1 Introduction

In recent years, expanding use of the Internet and network technologies are inspiring changes on how individuals share their resources. A Peer-to-Peer (P2P) paradigm has emerged [90]. In P2P architecture, each participant is also known as a peer that acts as both client and content provider. Increasingly, the resources could be made available to other users by being published from a user's machine [91].

In P2P environment, users can manage resources across heterogeneous platforms. According to the distributed nature of P2P computing, a P2P-based

resource management system can also provide higher resource availability and scalability [92].

Existing P2P networks allow users to upload and download files freely. Moreover, current P2P networks have not addressed licensing or document management issues. Consequently, many P2P networks are 'polluted' with unauthentic and illegal files [93]. At this point, how to use the information legally is a problem obsessing the power of P2P.

This research tries to addresses these shortcomings and provides a way for researchers to share and manage their resources legally. A collaborative research network prototype has been developed to support research across multiple disciplines, industries and sectors by providing a reliable mechanism for an open exchange of information [93].

In order to solve the legal issues in a professional way, this research involves collaboration between legal and IT research professionals. The legal team designs copyright licences to set permissions and privileges to document. The collaborative research network will include both copyright licensing information and digital rights management ("DRM") information over the P2P network.

IT team develops a P2P network prototype that enables effective control over users through existing user authentication. The system determines who can upload or download or modify the contents. A range of "machine or network readable" licenses are attached to files that are uploaded to the network whilst simultaneously facilitating free public access to material on the network [93]. The proposed collaborative research network is shown in Figure 3-7:

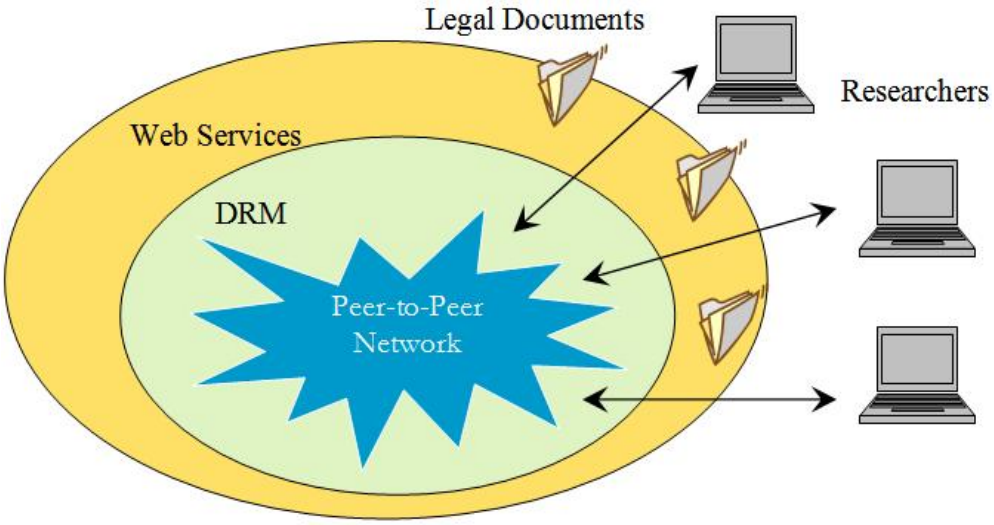

Figure 3-7 Collaborative research network

3.5.2 P2P Information Sharing

P2P is often described as "collaborative networking" technology. Each peer may store data relevant to that peer and potentially useful to other peers in the network. Currently, P2P applications have been successful for special cases such as exchanging music files [94].

3.5.3 Technical Architecture

In P2P computing environment, computer resources and services are sharing through direct communication between systems. Server capabilities are enabled on computers that traditionally acted as clients. Information resources in P2P networks can be navigated through numerous peers which are waiting to be queried for these resources. When a peer decides that data hosted on another peer is useful, it visits directly this peer in order to obtain that data. Now traditional clients can share processing power, bandwidth, and storage. Each functional unit in the network behaves similarly [94]. "The essential characteristic of a P2P network is that any machine in the network is logically capable of both providing and consuming information [95]."

Compared with client-server model, P2P-based services have several advantages. From a resource perspective, users can manage resources residing in heterogeneous platforms. This architecture extends computing ability to function and scale in the presence of a large population of nodes and networks. Moreover, Instead of being handled by a single company, institution or person, administration and maintenance responsibility for the operation are also distributed among the users. A central server is no longer needed and the overhead of its administration is also economised. Consequently, A P2P-based resource management model can provide higher resource availability and scalability due to the distributed nature of P2P computing.

Furthermore, each peer can select one of the available service providers based on service levels and conditions. Surely, there can be multiple peers for the same resources with different service levels or conditions. Moreover, P2P-based service provides higher utilisation of Internet service resources [91]. Finally, P2P architectures have the potential to accelerate communication processes and reduce collaboration costs through the ad hoc administration of working groups [95].

3.5.4 Legal Issues

Rodriguez [96] has identified 10 legal and commercial issues that prevent P2P from realising its real power.

In traditional client-server architectures, users only need to trust a small set of servers deployed by the content provider in most cases. In P2P, however, any computer is a potential server and building trust relationships becomes a hard task. Although some networks may require registration for identification, generally P2P networks do not require users to authenticate before logging on to the network. There is no access control on file sharing and shared files are available to everyone [93]. The open and anonymous nature of these networks

results in a complete lack of accountability for contents uploaded onto the network. This limitation opens the door to abuses of these networks by malicious peers [95]. Consequently, P2P networks need a mechanism to determine which nodes may be malicious and may introduce corrupted content. Moreover, P2P networks need to identify corrupted information and be robust to attackers [96].

Privacy is another concern. Since users connect to other computers to download resources, they potentially expose their important personal information, such as IP address, geographical location and viewing preferences. These users might be annoyed by marketing campaigns, spam, or even security attacks. At this point, mechanisms must be launched to ensure that user's privacy and security [96].

Last but not the least, existing P2P networks should also make an effort to foster the distribution of legal content. In this regard, it is very important for P2P distribution to publicise and encourage legal uses. P2P based applications should help protect copyrights over P2P networks [96].

3.5.5 Legal Information Sharing

In order to share research resources efficiently, we have to take existing research repositories into account. As soon as the legal issues solved, these variety of resources will be "activated" into our research network. Moreover, metadata also plays a significant role in licence implementation and rights management.

Research Network Context

Nearly every university already has a large educational and research resource repository distributed within the university boundary. These resources are under control of the single entities or individuals. All approaches for the distribution of educational media based on central repositories, however, have failed so far [94].

Furthermore, setting up and maintaining central servers are costly. On the other hand, distributing educational materials might not directly benefit the sponsoring university. While the researchers from other institutions who need to access these contents are limited to retrieve them.

In order to facilitate the exchange of research resources efficiently and effectively, approaches based on metadata enhanced P2P networks are necessary. In a P2P-based collaborative research scenario, each research institution acts not only as content provider but also content consumer. Content provider will not lose their control over their resources since any material will be licensed before uploaded to P2P network. The content provider will also be benefit from being able to access to a whole resource networks [94].

DRM, Licence and Metadata

Digital Rights Management (DRM) is an important facilitating mechanism for protecting copyrights. DRM, compared with copyright, refers to "technological tools and mechanisms that monitor content use and protect against unauthorized uses or distributions" [97]. Then, DRM can shield intellectual property by helping content owners enforce usage restrictions and affirm property rights on their copyrighted materials.

In our collaborative research network, researchers have to understand their legal responsibilities and liability [98]. We deploy the idea of open content licences which are generic, payment-free, licence agreements that are attached to material in which copyright owners grant users of the material very broad rights, such as the right to copy, distribute and in some cases, modify the material. Users can accept these agreements when they download the material or exercise any of the rights granted in the licence agreement and are not required to contact the copyright owner. One of the benefits of open content licences is that they make material easily accessible for use by others. The licence, determined

by the nature of documents, defines the access permissions of a document. Figure 3-8 and Table 3-6 show the restriction imposed on the document.

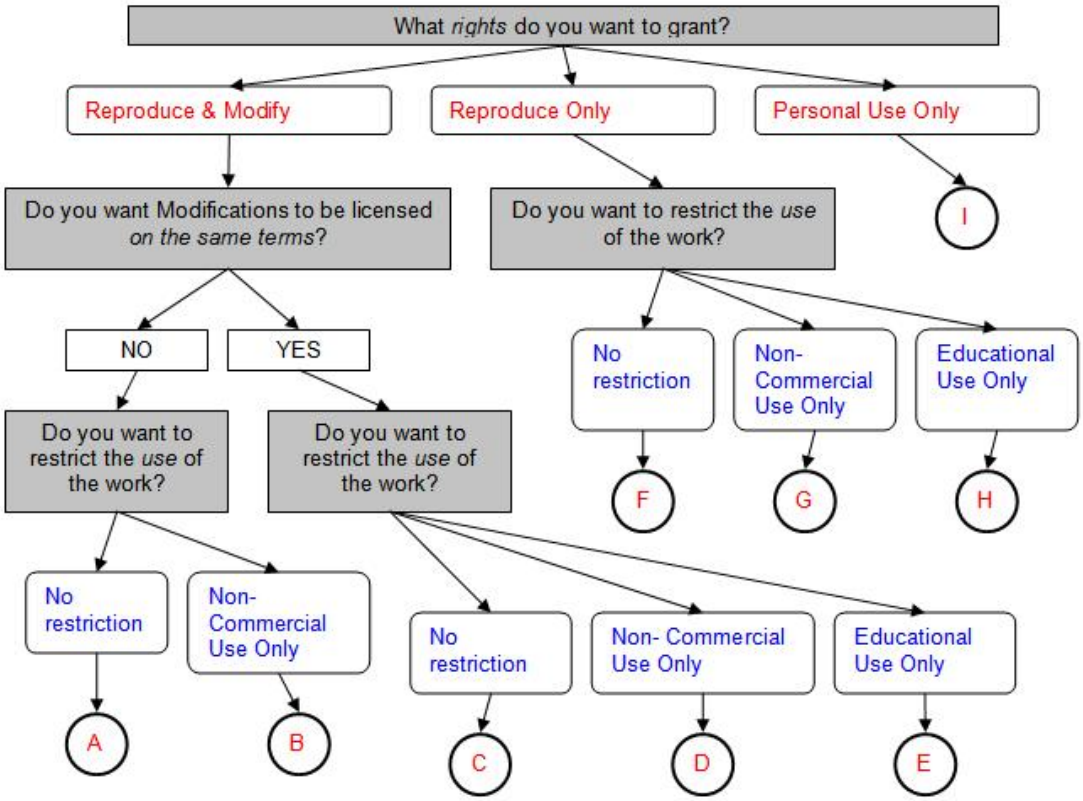

Figure 3-8 Licence Flowchart

In order to facilitate interoperability and reusability of research resources, the architecture should support DRM and document licence in P2P environment. P2P based document metadata will be used in such systems. Metadata embodies the licence of each uploaded document.

Table 3-6 Restriction and Licence Types

	Restriction on Type of Use		
Licence Types	No restriction	Non-commercial	Educational
Reproduce & Modify	A	B	
Reproduce &	C	D	E

Modify on Same Terms			
Reproduce Only	F	G	H
Personal Use Only		I	

The licensing process is actually attaching appropriate licence information into a document's metadata. Our P2P network can then understand the licensed documents. Consequently, only permitted user group can download, reproduce or modify the documents.

Web Services

Web services provide a way for applications communication over the Internet. Since the Internet contains heterogeneous applications and platforms, Web services help to solve the interoperability of those various applications, platforms and frameworks. The Web services architecture is based on the interactions between three primary components: the service provider, service registry, and service requestor. These components interact with each other using publish, find and bind operations. The service provider provides access to Web services and publishes the service description in a service registry. The service requestor finds the service description in a service registry and uses that information to bind to the services. Service discovery defines a process for locating service providers and retrieving service descriptions [91].

From a P2P perspective, Web services are more intended to promote interoperability and extensibility among various applications, platforms and frameworks in terms of modularising external applications [94]. Web services provide the universal information architecture to solve the integration problem. While on the other hand, P2P provides the distributed network architecture that directly make the resources available on the Internet. The interaction among platforms may potentially combine advantages of them and will show the way to

a distributed computing environment [99]. In our collaborative legal information sharing infrastructure, each peer will be required to offer a number of basic services and may offer additional advanced services.

3.5.6 Prototype Design

A P2P based collaborative research network, named vuCRN, has been implemented. In the flowing section, system requirements and design consideration are illustrated from ontological level, conceptual level and functional level.

Ontological Level Requirements Analysis

"An ontology is an explicit specification of an abstract, simplified view of a world we desire to represent" [100] Ontologies are used to capture knowledge about some domain of interest. It describes the concepts in the domain and also the relationships that hold between those concepts.

In this research, ontology based metadata is deployed to facilitate domain knowledge access between IT and legal team. Moreover, ontology properties define relationships of system requirements. The licence ontology can be mapped from licence flowchart. Figure 3 is an example illustrated by licence C, D and E. License ontology describes usage limitations of different licenses while restriction ontology defines the purposes of these licenses.

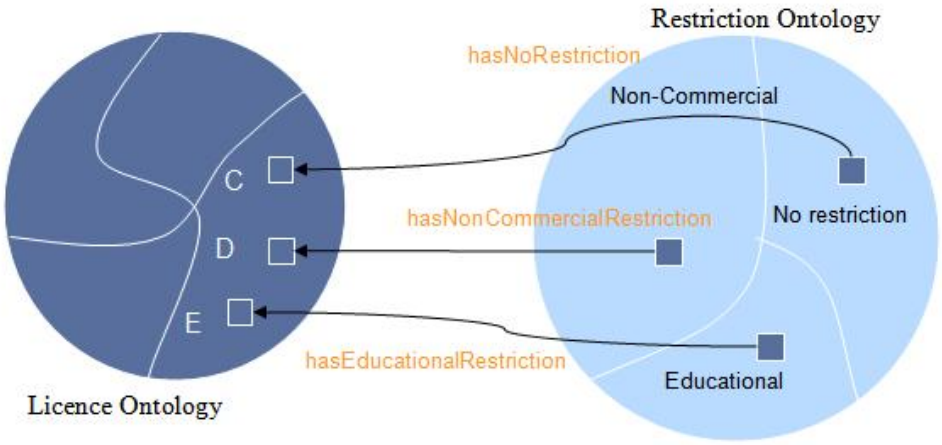

Figure 3-9 License Ontology

Left represents various licenses in license ontology. Right is restriction ontology that can be mapped to license ontology. The properties link the licence ontology and restriction ontology. And the following figure, which is part of Figure 3-8, shows the original restriction type and licence type for license C, D and E.

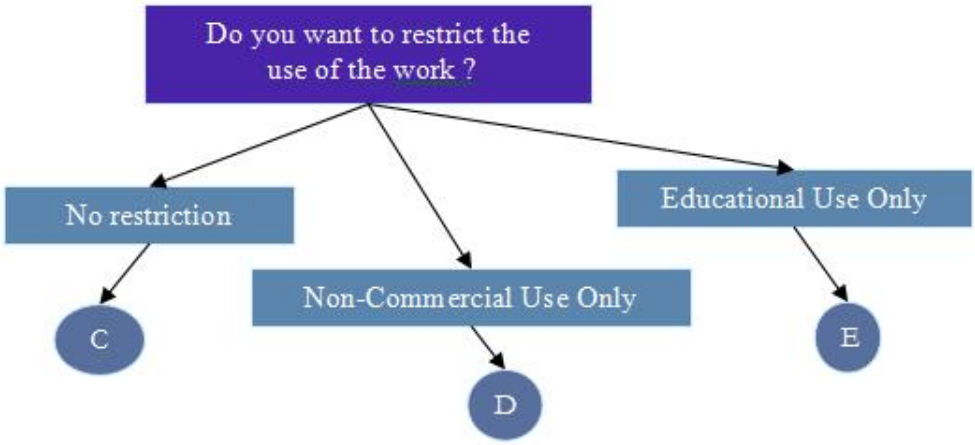

Figure 3-10 Restriction Mapping On License Ontology

The most recent development in standard ontology languages is OWL from the World Wide Web Consortium (W3C) [101]. It is based on a different logical model which makes it possible for concepts to be defined as well as described. Complex

concepts can therefore be built up in definitions out of simpler concepts. The following is part of OWL code that indicates above licence ontology.

```
<owl:Class rdf:ID="Licence_C">
     <rdfs:subClassOf
          rdf:resource="#Reproduce_and_Modify_on_Same_Terms"/>
     <rdfs:subClassOf>
     <owl:Restriction>
               <owl:onProperty rdf:resource="#hasNoRestriction"/>
               <owl:allValuesFrom rdf:resource="#Licence_C"/>
          </owl:Restriction>
     </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Licence_D">
     <rdfs:subClassOf
          rdf:resource="#Reproduce_and_Modify_on_Same_Terms"/>
     <rdfs:subClassOf>
          <owl:Restriction>
               <owl:onProperty
                    rdf:resource="#hasNonCommercialRestriction"/>
               <owl:allValuesFrom rdf:resource="#Licence_D"/>
          </owl:Restriction>
     </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Licence_E">
     <rdfs:subClassOf
          rdf:resource="#Reproduce_and_Modify_on_Same_Terms"/>
     <rdfs:subClassOf>
          <owl:Restriction>
               <owl:onProperty
```

```
                    rdf:resource="#hasEducationalRestriction"/>
                <owl:allValuesFrom rdf:resource="#Licence_E"/>
            </owl:Restriction>
        </rdfs:subClassOf>
</owl:Class>
```

P2P based document metadata will be used to facilitate interoperability and reusability of research resources. Metadata embodies the licence of each uploaded document. The licensing process is actually attaching appropriate licence information into a document's metadata. Our P2P network can then understand the licensed documents. Consequently, only permitted user group can download, reproduce or modify the documents. Petri Net [89] method is deployed to infrastructure modelling. In Petri Net, each circle is a channel that represents a passive system while each box is an agency that represents an active system component. The arrow indicates the information flow. Figure 3-11 shows Petri Net model for ontological level.

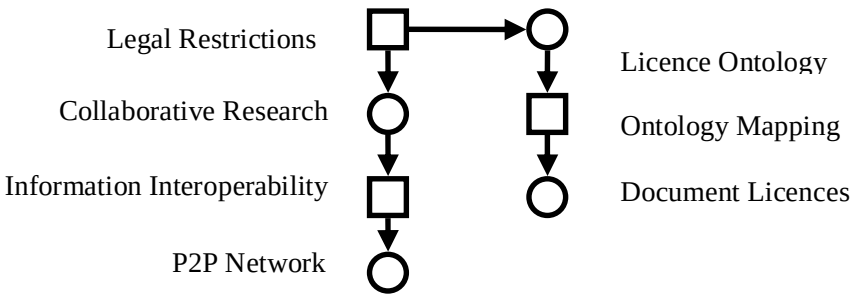

Figure 3-11 Ontology Level Petri Net

Conceptual Level Requirements Analysis

Ontology based metadata facilities the access to domain knowledge between IT and legal team. Legal professionals design copyright licences to set permissions and privileges to document. Metadata embodies licence to original document.

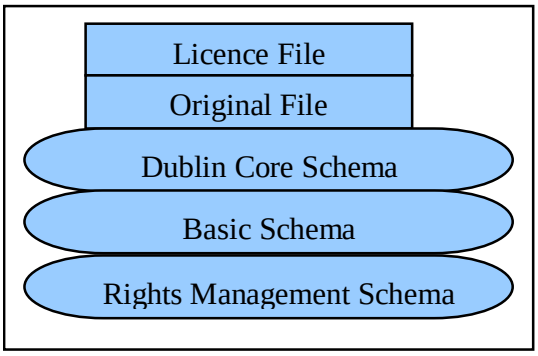

Figure 3-12 Final Document Content Structure

The original document is licensed by "document licence services" and then uploaded onto P2P network. At this stage, P2P services are responsible for document transfer among peers.

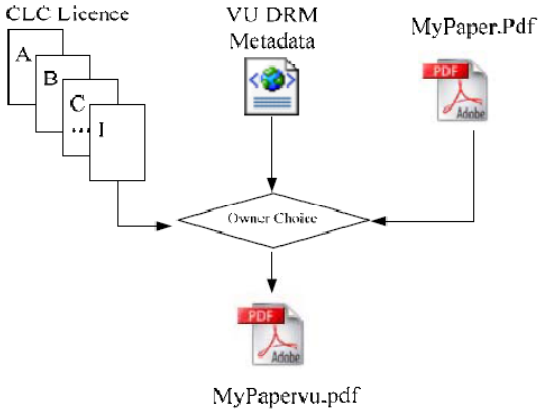

Figure 3-13 Document Flowchart

When a user downloads the licensed document, the "document licence services" again decides which permission the user has according to his or her user group. After that, the modifiable document might be modified by the permitted user and enter another circle in network. Figure 3-14 shows the Petri Net Model for conceptual level.

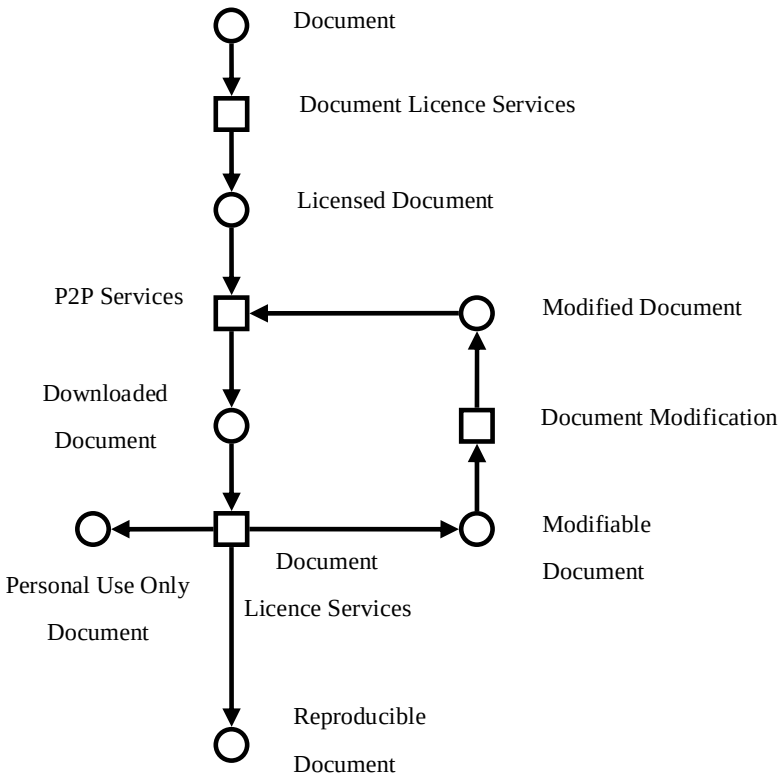

Figure 3-14 Conceptual Level Petri Net

Functional Level Requirements Analysis

The licence is used to enforce legal restrictions. The original document is licensed by "vuCRN document licence services" and then uploaded onto JXTA P2P network. At this stage, JXTA services are responsible for document transfer among peers. When a user downloads the licensed document, the "document licence services" again decides which permission the user has according to his or her user group. After that, the modifiable document might be modified by the permitted user and enter another circle in network. As a result, the requirements gathered from above models can be described in Figure 3-15.

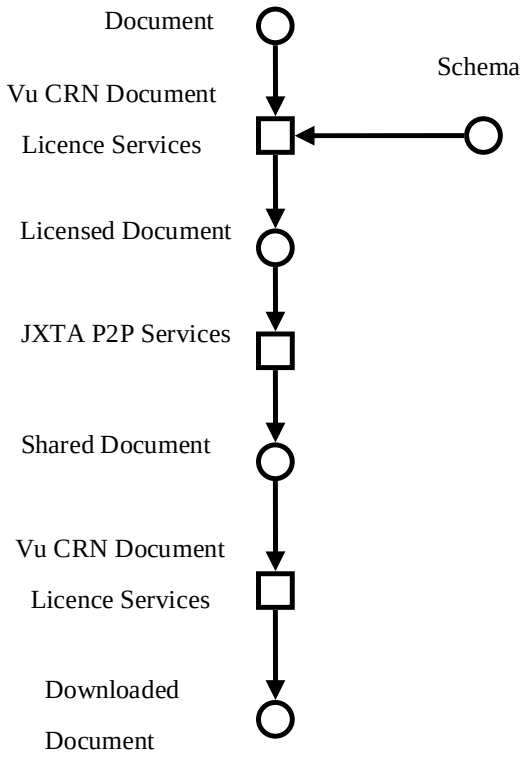

Figure 3-15 Functional Level Petri Net

Once the licence is selected by user, the licensed document is shared on our P2P network. Other peer can download this document according to their privilege.

Figure 3-16 Licence Selection Panel

Our prototype system is based on JXTA [102] which is an Open Source decentralised P2P platform created by Sun Microsystems in 2001. It allows any device connected to a network to exchange messages. In essence, JXTA provides a set of XML based protocols to cover typical P2P functionality. Its goal is to develop basic building blocks and services to enable innovative applications for peer groups. Figure 3-16 and 3-17 shows the Licence Selection Panel and Peer Group.

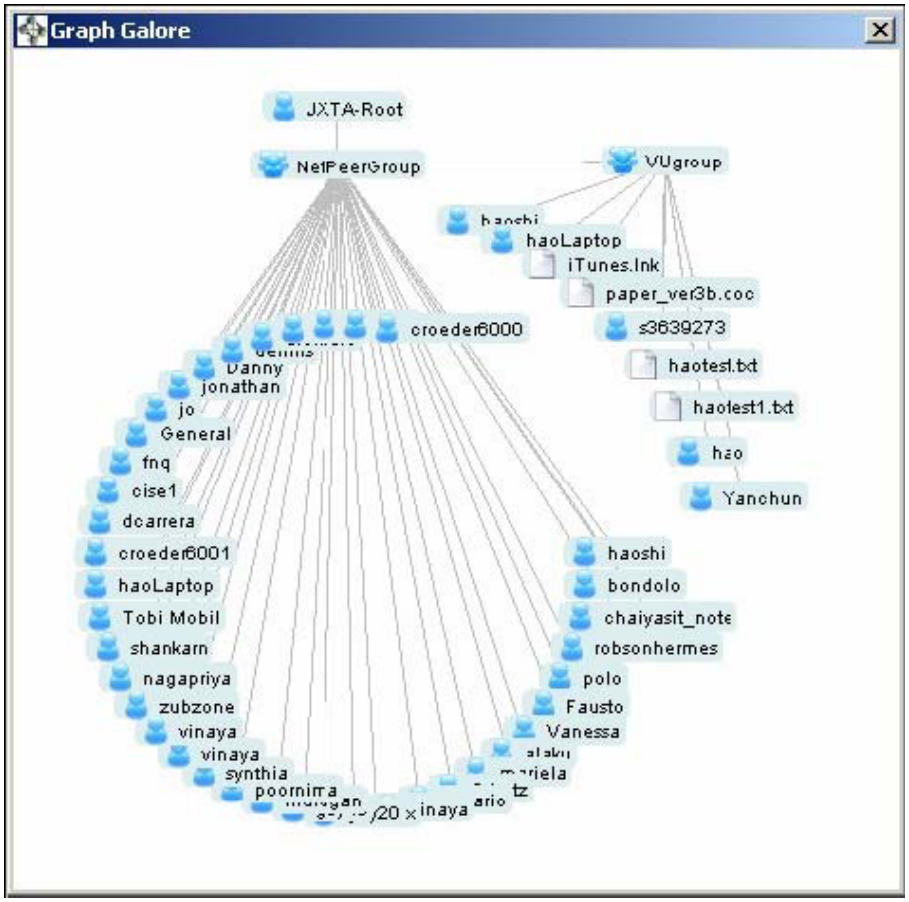

Figure 3-17 Peer Group

The following code illustrates part of the schema file which embodies related document licence.

```
<rdf:Description rdf:about =
‘“xmlns:xmpRiggts=”http://ns.adobe.com/xap/1.0/rights/’>
```

```
      <xmpRights:Marked>True</ xmpRights:marked>
      <xmpRights:Owner>
            <rdf:bag>
                  <rdf:li>s3732166</rdf:li>
            </rdf:bag>
      </xmpRights:Owner>
      <xmpRights:UsageTerms>
            <rdf:Alt>
                  <rdf:li xml:lang='x-default'>
                        Microsoft Word –Document
                  </rdf:li>
                  <rdf:li>xml:lang='en-us'>License B</rdf:li>
            <rdf:Alt>
      </xmpRights:UsageTerms>
</rdf:Description>
```

Requirements

The primary goal of the proposed infrastructure is to provide a P2P network for collaboration between academics and researchers. By applying Petri Net modelling approach, the requirements can be derived from above models. These requirements can be described in Table 3-7.

Table 3-7 Restriction and Licence Types

	Strategic	Organisational
Ontological	Determine legal restrictions level	Ontology mapping between legal and IT terms
Conceptual	Design document licences	Design DRM rules.
Functional	Design P2P Network Architecture	Develop P2P Web Services for cross-platform interoperability

	Transactional	Operational
Ontological	Sustain information framework consistence.	Design P2P based communication infrastructure.
Conceptual	Design licence coordination mechanisms.	Define standard metadata semantic and syntax.
Functional	Attach licence to each document.	Upload and download licensed document in P2P network

Petri Net modelling approach provides a clear pathway for gathering requirements and presents a comprehensible roadmap to achieve P2P collaborative research network particularly in terms of collaborative information sharing.

3.5.7 Conclusion and future work

The primary goal of proposed network is to provide a P2P network for collaboration between academics and researchers. File sharing is only the beginning for this P2P network. The success of JXTA based prototype for file sharing indicates a viable way to develop a collaborative research network for academics and researchers to facilitate legal document sharing. Consequently, information sharing will be able to overcome legal barriers and become an important source for creative works. Eventually, it will help to expand such collaborative research network to facilitate collaborative research across various disciplines.

3.6 SUMMARY

The business world is changing every day, new legal requirements, changes in strategy, reorganisation of partnership, etc. Organisations need to adapt themselves to these changes accordingly. Efficient, effective and dynamic

collaboration among business partners will be an advantage over competitors. The reorganisation of requirements is a first step to achieve this goal.

In this chapter, by reviewing a variety of representational scenarios, we modelled these scenarios iteratively by Petri Nets. Then, we proposed the description and explanation of a set of requirements for collaborative business process modelling. These requirements are illustrated from both collaboration levels, namely strategic, organisational, transactional, operational levels and abstraction levels, namely ontological, conceptual, functional levels. This chapter provides a more comprehensive understanding for practitioners in this area. Successful satisfaction of these requirements can lead to harmony business-IT alignment in business process modelling.

With these requirements from business side, we are going to discuss the rules in typical collaborative business environment in next chapter. These rules embody and transform the requirements to business rules which in turn reinforce the fulfilment of business necessities.

CHAPTER 4

RULES IN COLLABORATIVE BUSINESS PROCESSES

Organisations are facing more challenges of capturing their internal processes as well as the collaborative processes, especially in the domain of information security. Even the scenario based requirements are beneficial to the business process modelling in a modularised manner. Creating and maintaining and the dependencies, among business partners, remain as challenging tasks. How to ensure the consistence among partners at various levels of collaboration? How to capturing different requirements that might potentially contradict with one another?

These questions bring us to the second step in our research roadmap. We discuss on business policies and rules which are introduced to handle these issues in business collaboration. Different types of business rules are reviewed and summarised. The algorithms are introduced to solve conflicts in rule conflict and generate collaborative rules.

4.1 RULE BASIC

As defined in Cambridge Learner's Dictionary , a rule is considered to be 'an accepted principle or instruction that states the way things are or should be done, and tells you what you are allowed or are not allowed to do'. From the business point of view, rules construct a fundamental mechanism in the decision making and reasoning process. In any properly organised business process, related rules work as guide to all the staff to make their decision. Organisations are human constructed, so rules govern human behaviour as well as organisational behaviour. Organisations like Bank and Real Estate Agency rely heavily on their policies and rules to govern the way in which they carry out their business (like

customer service policies, work order processing, business policies, contracts, operating manuals and so on).

From the system point of view, rules are typically expressed as if-then statements. The **if** part of a rule is called **antecedent** which comprises the conditions in certain context. The **then** part of a rule is referred to as its **consequent** which constitutes the conclusions and decisions. In other word, it defines what will be true as a consequence given the fact that the rule's conditions are true.

A rule can have multiple conditions which are connected by 'and' and 'or'. An example of an everyday rule in **Bank** is that 'if the risk of an application is ranked as high, then only **DEPARTMENT MANAGER** can assess and approve this application'.

In this chapter we present a rule based approach to facilitate dynamic business collaboration development and management. Rules are basically statements that define what to do or not to do. Organisations use rules to guide and control their activities in order to conduct business in a desired manner. Rules have been implemented in one form or another in organisations since they first came into existence. Participating partners need to coordinate their internal rules into both agreed collaborative rules before any business activities. How to identify the collaborative rules and how to resolve conflicts between rules are major concerns in this chapter.

In the remainder of this section we discuss these rules. We first discuss the brief history of business rule in section 4.2. Next, in section 4.3 and 4.4 we look into the characteristics of rules in the specific context of business collaboration. Subsequently, we introduce and investigate the collaborative business rule algorithms in section 4.5 and 4.6.

4.2 RULE HISTORY

Rules can be found all the way through literature in business collaboration context. In the focus of such rules has been mostly on so-called 'business rules'. Many definitions of these business rules exist. Whereas in [103], they are perceived as 'natural language sentences that describe data requirements to the business users.' Ross defines [104] a business rule as 'a statement that indicates a discrete, operational practice or policy in running a business without reference to any particular implementation technology'. [105] proposed that business rules are 'statements that define or constrain some aspect of the business, which is intended to assert business structure or to control or influence the behaviour of the business. Alternatively, more recently the work in Object Modeling Group defines [106] them as 'rules that govern the way a business operates, where rules are defined as declarations of policy or conditions that must be satisfied.'

What these definitions have in common is that they have a limited observation in terms of collaborative business process context. They treat business rules solely as some forms of constraints for the preservation and management of data. The rules in that we are interested in is extended to collaboration areas. We refer to this broader set of rules as business rules in this research to emphasis the fact that these rules actually govern the collaboration stakeholders, and are pervasive throughout the whole business collaboration.

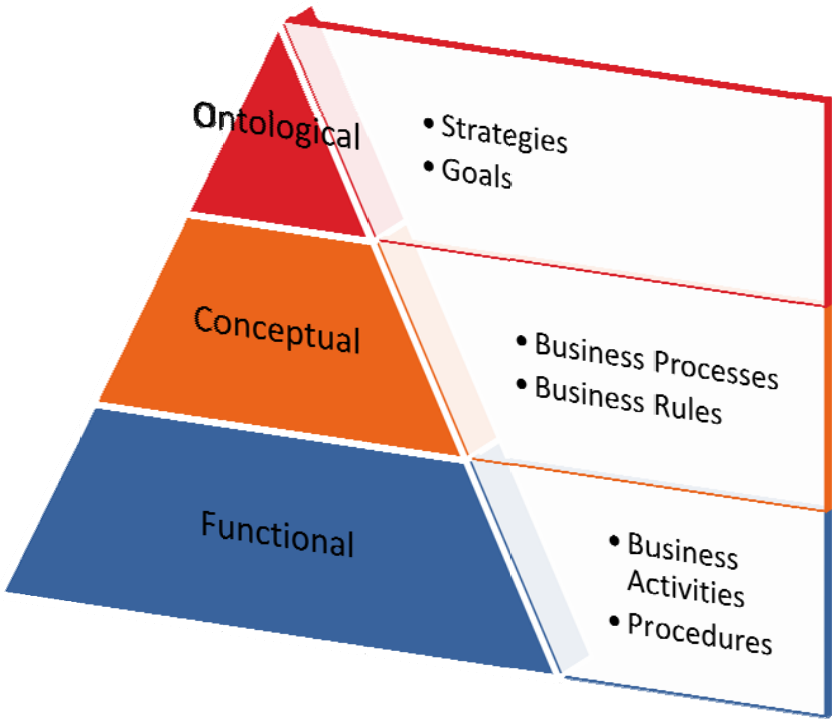

Figure 4-1 Positioning of Business Rules in Organisations

At the top of the pyramid, we find the overall strategies and goals of organisation. These usually consist of high level abstract descriptions of the organisation's objectives. For example, being an innovative and learning organisation. To achieve these objectives they illustrate more formally in business rules and processes in which describe how the business of the organisation should be conducted in order to attain the overall goals. These kinds of processes are often written in handbooks, for example a sales training manual may provides a description of how a home loan application is to be processed.

Business rules, at this stage, provide concrete statements that enforce restrictions on the business processes. The enforcement depends on implementations on the conceptual and functional levels. There are two main types of rules on these levels: human involved rules and automatic enforced rules. Human involved rules are considered as a 'soft' part of the organisation. In contrast, automatic enforced rules are encoded in the form of applications,

databases, networks and other forms of information technologies. When placed in the context of business collaboration, the diagram in Figure 4-1 might become even more complicated. For example, part of the organisation may not be automated by business applications. The partner may have several Web services 'talking' to this part. How to enforce the security rules to collaborate with each have implementation level rules? We will discuss more about this topic in Chapter 7. For the moment, we concentrate on collaborative business rules.

4.3 REQUIREMENTS ON COLLABORATIVE BUSINESS RULES

Business rules are pervasive throughout organisations as we stated earlier. By making these rules explicit organisations can effectively incorporate changes ranging from strategic to functional level. The collaborative rules need to be determined and finetuned by both partners. Sometime, this involves adjustment of their internal rules which are aligned with private processes. Consequently, the specification of collaborative rules can be easily changed by defining and modifying the rules in their existing processes. Organisations are able to make appropriate changes to their current underlying business.

Normally, in a collaborative business environment, an organisation not only requires the correct role assignment to access messages for its own services, but also the right role to access the messages it passes to its collaborating partners. But business collaboration is peer based and automated with Web services. Authorisation policies defined for individual organisations normally cannot be seen by others. Therefore, in order to guarantee that the messages transferred among organisations can be accessed by the qualified roles in business collaboration, each organisation need to send its collaborating partners the required role information together with messages to be accessed at collaborators' services. Based on this assumption, a message transferred and processed between services will be associated with two types of roles: one is the

access role of the current service; the other is the required role for the next service. The authorisation policies in various organisations can then be coordinated to enforce authorisation control in business collaboration.

4.3.1 STEP Principles

There are several concerns in rule based organisations. For example, many organisations are currently facing is that they have business logic embedded in business processes. In this case, they often take substantial time to change. In the dynamic business environment, the life span of business models has been significantly shortened, and as such it is critical for organisations to be able to adapt to changes on-the-fly. von Halle [107] have identified four fundamental principles, so-called STEP principles when adopting rule based approaches.

Separation

Rules should be developed and managed separately. Rule management system is introduced to support the authoring, deployment and management of rules. In this way, rules become more accessible and available for reuse. Furthermore, separation helps rule consistency checking which assists organisations to determine whether rules lead to conflicting situations throughout the business activities.

Traceability

Business people can see where a rule comes from and how to apply the rule in different situations. From the management point of view, traceability significantly enhances the capacity of organisations to explain and reason about their motivations. Explicitly separated business rules also increases their traceability. Traceability also enables organisations to assess the impact of rule changes on the business. For example, due to the financial crisis, banks may

tighten their lending standards. Related business rules can then be traced and modified accordingly.

Externalisation

Externalisation enables organisations to negotiate the rules in business collaboration with other parties. Organisations are aware of the existing rules and how these rules govern the business. By design, the rules should be separate from internal processes and external processes. This also assists organisations to trace and assess the impact from business collaborations.

Position rules for change

Separation and externalisation enable organisations to position their business rules in a structured and organised way. Organisations are then able to manage their strategies, business activities in a dynamic manner that changes can be easily implemented when taking strategies and processes into consideration. In this way, organisations can gain extensive control over their business collaborations.

Even with the above assumed setting, business collaboration could still be unreliable in terms of security policy conflicts among cross organisational processes. For example, incorrect rule assignment or modification can occur when the required role is inconsistent with the rule assignment for a message in a service within one organisation. Messages transferred from one organisation are accessed by unqualified roles in other collaborating business partners. We refer to authorisation rule based business collaboration reliability as that the desired messages transferred within or across organisations are accessed and modified through services by qualified roles according to authorisation rules.

4.3.2 Adaptability

Besides the STEP principles, we identified two properties in collaborative business rules. The first identified property is adaptability which requires the ability to handle changes in both design-time and run-time business collaboration.

In design-time, a fault message in response to a request and undefined interface are examples of common exceptions. Organisations can work together to figure out how to handle such exceptions by specifying suitable derivation rules in their design schemas and related processes. If this kind of exceptions occurs, the business partner will then behave in accordance with the specified rules.

To illustrate, for the event that Bank B requires credit rating less than level A to be handled by **DEPARTMENT MANAGER**. While in the credit agency C, only if the customer rating is or less than B, the result will be sent back to higher level staff. In this case, the related rules will then marked as 'exceptional'. The business people will then involve in solving this issue to get both agreed rule. This new rule will be added to both Bank B and credit agency C's rule repository. It will also be activated in the next round of B&C's business activity.

By nature, run-time collaboration behaviour is inevitably unpredictable. The actual rule processing is conducted at run-time. Organisations can influence the design by changing their rules' definition until the real business activities start. The underlying business collaborations can always be extended on an on-demand basis without having full knowledge at design time. The rules should able to accommodate reasonable changes in an identical manner and deduce how to behave that are applicable in the specific circumstances. The major concern in run-time counterpart is how to identify the rule conflict. Once the conflict positioned, it will follow the same procedure to solve the issue as in design time

rule. We will return to this matter in Chapter 7 where we have developed a Petri Net based methodology to verify run-time business rules.

4.3.3 Dynamicity

Dynamicity defines the ability to modify existing business rules at internal, collaborative and public perspectives. In a business collaboration scenario, organisations aim to deliver flexibility of collaborating partners. Rules are categorised into internal rules, public rules and collaborative rules. An example for public rule is 'a loan applicant must be older than 18'. A collaborative role may define 'customer credit rating below C will not be applicable to any loan'. An internal rule may define 'loan amount over one million must be approved by **BRANCH MANAGER**'.

The dynamicity property requires these types of rules to be consistent throughout business processes. Organisations can define different rules in their design schemas to handle different situations. The changes in the dynamism category are expected to balance between inter-organisation and intra-organisation processes. Business collaborations are usually too complex to identify and define all possible behaviours at collaborative level as exhibit interactions not fully predictable at design time.

In section 4.6 we introduce the rule negotiation algorithms to handle modified rules on existing designs can be accessed and communicated with affected business activities.

Normally, in the rule based approach, new derivation rules are defined and applied while individual business collaborations are already running. As such, if an unplanned situation occurs, organisations can simply add new rules or modify existing rules to cater the new scenarios.

In case the new rules affect already carried out in the well-established business collaboration, then this is resolved in the same manner as just described for internal processes and public processes.

4.4 REVIEW OF MOTIVATING EXAMPLE

Let us take an example of Bank Loan Application which involves two collaborating parties: Bank and Credit Rating Agency. The loan application process begins with receiving an application from a Customer. The Bank will then require Credit Rating Agency to check the Customer's credit level. The risk assessment will be executed at the Bank side when the result for the credit check returns. The eligible applications are approved while the unacceptable high risk applications are rejected.

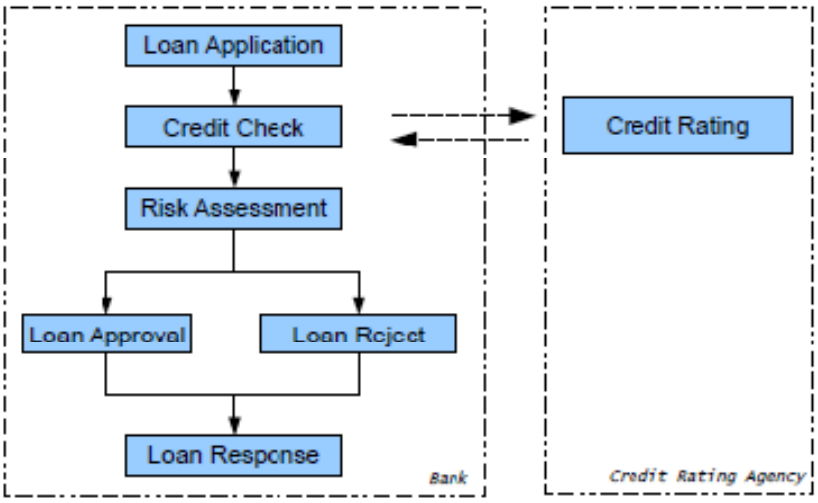

Figure 4-2 Home Loan Application

We present the role based authorisation policies of Bank which are separated into two categories according to the scope of message transfer:

Intra-organisation message transfer: If a message is transferred between services within the same organisation, a role must be assigned to process the

message at the service according to authorisation policies. For example, Risk Administrator in Bank is assigned to deal with huge-amount loan application at Risk Assessment service.

Inter-organisation message transfer: Since all involving organisations in business collaboration are peers with equal rights, their internal authorisation policies cannot be totally revealed to each other. However, individual organisations need to send their collaborators the message being processed as well as the role requirement to enforce that the message can be accessed by qualified role of the collaborators. For instance, the Bank may require **CREDIT CHECK MANAGER** to access the credit check requirement message at Credit Check service of Credit Rating Agency to protect client's privacy.

The role authorisation in business collaboration environment is not only determined by the two types of role authorisation policies, but also restricted by three types of role authorisation constraints according to role-to-role relationship: Role Hierarchy, Role Dependency and Role Conflict:

- The Role Hierarchy is used to map the organisational structure with role hierarchy. For example, on the Bank side, **BANK MANAGER** (Director Level Role) is on top of the role hierarchy. **BANK MANAGER** governs **LOAN MANAGER** and **RISK ADMINISTRATOR** (Manager Level Roles). **LOAN MANAGER** governs **LOAN OFFICER** (General Officer Level Role) while **RISK ADMINISTRATOR** governs **RISK OFFICER** (General Officer Level Role).
- A Role Dependency defines the dependency relationship among different roles. For instance, if the message is accessed by **LOAN MANAGER** in Loan Application service in Bank due to protecting client's privacy, then Bank must require **CREDIT CHECK MANAGER** to deal with the loan at Credit Check service in Credit Rating Agency.

- Role Conflict is intended to determine the conflict constraints of roles authorisation. For example, at Bank side, if the message has been processed by an upper level role in role hierarchy, then no lower level role can be assigned the permission to access the message at following service.

4.5 RULE ASSERTION

As even this simple example already shows a change in one part of business collaboration can have cascading affect on the whole collaboration. Organisations require the mechanisms to give them the ability to cope with collaborative rules in an easy and effective manner.

Definition 4-1 Business Rule

$\mathcal{R}[\![A]\!]$ is the set of rules in organisation A. ${}^{i}_{L}\mathcal{R}[\![A]\!]$ is the internal rule set of organisation A; ${}^{j}_{C}\mathcal{R}[\![A]\!]$ is the collaborative rule set of organisation A; ${}^{k}_{P}\mathcal{R}[\![A]\!]$ is the public rule set of organisation A. Where ${}^{i}_{L}\mathcal{R}[\![A]\!] \cup {}^{j}_{C}\mathcal{R}[\![A]\!] \cup {}^{k}_{P}\mathcal{R}[\![A]\!] = \mathcal{R}[\![A]\!]$.

For example:

${}^{1}_{L}\mathcal{R}[\![Bank]\!]$: Loan amount over $100K, send application to **MANAGER** for approval.

${}^{2}_{L}\mathcal{R}[\![Bank]\!]$: Customer credit level below A, send to application to **MANAGER** for approval.

${}^{1}_{C}\mathcal{R}[\![Bank]\!]$: Customer credit rating below B, send result back to superordinates' level role.

A rule will typically consist of one or more rule assertions. The assertion describes a set of logically related conditions that they govern and constrain business collaboration in a coherent and consistent manner. Assertions enable organisations to cope with different business scenarios when cooperating with different partners.

Rule assertions allow organisations to depict several options for the same situation in order to providing collaboration interface with potential partners. But the alternatives must be mutually exclusive. In other word, there is one and only one alternative rule applicable under give circumstance. Business rules must be understandable by business people who are involved in the communication of business processes with business people from other organisations.

4.6 RULE NEGOTIATION ALGORITHM

In this section, we introduce the rule negotiation algorithm which is mainly focus on solving conflicts for collaborative processes at design-time. We designed a Rule Dependency Tree (RDT) to map the structure of each RBAC compatible business rule in participating organisations. The negotiation algorithm will then utilise the RDTs to compare rule requirements and available rules.

Rule Dependency Tree (RDT)

Rule Dependency Tree is a binary tree. The root node stores the conditions of the rule. The subordinate nodes store the organisation role hierarchy information of the involved processes and rules. The left child stores the subordinate role while the right child stores the permission to this level.

The left children map the organisation structure from the top level to bottom. While the right children store the decisions of the given conditions.

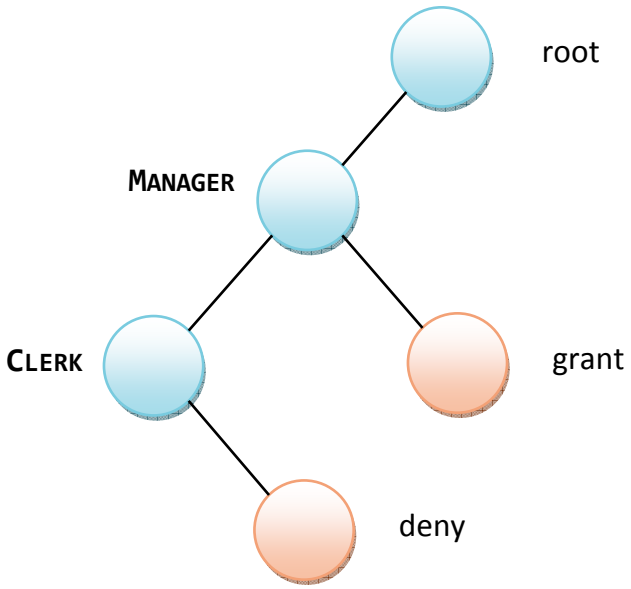

Figure 4-3 Rule Dependency Tree

We create new rule instead of changing service. The service is impossible to be modified every time when dealing with new partners. But it is more realistic to change rule to accommodate each partner's requirements.

```
PROCEDURE CheckConflict (LocalService, CollaborativeService, Message)
FOR EACH involved local rule rule_L_(i)
     requiredRole := GetRequiredRole (LocalService, Message, rule_L_(i))
     FOR EACH involved collaborative rule rule_C_(i)
          proposedRole := GetProposedRole (CollaborativeService, Message,
                          rule_C_(i))
          currentNode := root.left
          WHILE currentNode.Name <> roleName
               IF currentNode.left IS NOT NULL
               THEN currentNode := currentNode.left
               ELSE RETURN NULL
               ENDIF
          ENDWHILE
          IF check(requiredRole.name) <> check(proposedRole.name)
```

```
            THEN ConflictTable.Add(requiredRole, proposedRole)
            ENDIF
      ENDFOR
ENDFOR
RETURN ConflictTable
```

The collaboration parties take their own rules in the policy assertion that involved in the business process. We assume the assertion sets are true for each participant. This is one of the business collaboration assumptions as intra-organisational assertions guide the process and activities within the individual organisation.

Then the related rules proceed to negotiation procedure. The required role from local organisation and proposed role from partner are evaluated. If no conflict exists, a collaborative rule is then edited and stored in the rule repository. If there is a conflict, the two organisations will go into a negotiation procedure.

At this stage, in this negotiation procedure, business people and business analyst are involved in order to meet the requirements from both technical perspective and business point of view. The RDT is then re-built after both agreed collaborative rule as the internal rule(s) might be changed at local organisation or partner's end.

4.7 SUMMARY

Thus far in this section we introduced the security rules in business and discussed their general and advanced characteristics. Currently, access to information is most often approached from a simplistic perspective of specifying what other users of the particular system can do to the information (in terms of access rights). These access rights are specified and enforced by many different

technologies, with varying degrees of compatibility. It can be seen from the above that the current business practices involve the propagation of information between organisations. Agreements (and mechanisms) for propagating such information needs to be an accompanying process to understand and enforce the security policies of all involved parties.

Resulting from above discussion, we gain the view that a security rule in business collaboration scenario must be understood by business people, which is intended to assert business structure or to control the behaviour of the business processes. It is associated with a precise schema and it is declarative in nature. In the perfect world, it can be easily made communicatable, executable and easily modifiable. Each rule furthermore has several characteristics that help facilitate its management tasks such as status, version, documentation, and so on.

This requires not just a mechanistic application of the sum of all policies (as such an approach would likely fail with policies being applied out of context) but a process [13] that results in a secure handling of information and accessing services satisfactory to all parties. We are going to discuss the access control specification and verification in the next three chapters.

CHAPTER 5

AUTHORISATION SPECIFICATION - BPEL4RBAC

In the first two chapters, we discussed that business process management is designed to make business activities and trade easier and more cost effective. The increasing business integration and legal requirements raise the need for secure business processes. The openness and distribution nature of inter-organisational business processes may result in more security breaches. As a widely accepted business process standard, WS-BPEL does not provide any support for business process security protection even if the participating organisations already have a working security policy. To address this problem, we have developed an authorisation specification BPEL4RBAC for WS-BPEL in this chapter. Through BPEL4RBAC access control model, with an extension for WS-BPEL, called BPEL4RBAC policy language, the secure WS-BPEL is then achievable. The former introduces the access control capability into business process environment while the later is used to represent the authorisation information in WS-BPEL.

To address these problems, this chapter aims to provide a theoretical foundation for realising effective access control in business process management systems that can adequately meet the distinctive security challenges in Web services environment. We introduce BPEL4RBAC, an authorisation specification, to provide access control and authorisation constraints ability to existing WS-BPEL standard.

The remainder of this chapter is organised as follows: Section 5.1 and 5.2 describes the application scenarios of business process and access control. The access control requirements are illustrated with a running example in Section 5.3.

Section 5.4 introduces the BPEL4RBAC model and policy language in detail. The last section, Section 5.5, summarises the contribution and discusses future works.

Balancing business collaboration and system security are competing goals [3]. Business applications contain information with variable levels of sensitivity in nature. However, in business process environment, the business activities are highly unpredictable comparing with single user applications [4]. In contrast, the open access in business process requires higher level of integrity and confidentiality. Many research works have been done in this challenging area [108] [6] [30] [8].

5.1 BUSINESS PROCESS AND ACCESS CONTROL

In this section, we illustrate WS-BPEL and RBAC model separately by running examples. For easy understanding, we use a common scenario: bank loan application. The processes in this scenario are quite straightforward. First, a customer applies for bank loan. Then the bank conducts credit check according to this applicant and the risk level is also assessed comprehensively. Finally the loan application is approved to the eligible applicant while unacceptable high risk applications are rejected. In this scenario, the business processes are formalised by original WS-BPEL 2.0 code. The access control requirements and constraints are described in plain language.

5.2 BUSINESS PROCESS IN BPEL

WS-BPEL is designed to describe business processes in a structured way. The business logic is expressed as a group of activities which are performed by invoking Web services. The `<process>` element is on the top level of WS-BPEL specification. The attributes of `<process>` specifies the process name and related namespace.

From the system point of view, <partnerLink> element indicates the external Web services to be invoked from this process. The <variable> element defines the data variables involved in this process. Some basic control structures are also deployed to describe business logic. The <sequence> element contains sequentially executed activities while the <flow> element specifies concurrently performed activities. These elements may contain one or more basic elements such as <receive> and <reply> which define the message flows in a process.

In bank loan application scenario, we can divide the whole process into six activities:

1. The customer applies for bank loan (apBL)
2. Loan officer conducts credit check based on customer's application (ccAP)
3. Low risk application is automatically approved (rkBL)
4. High risk application is re-assessed by loan admin or bank manager (reAP)
5. Some high risk applications are rejected while others are approved based on reassessment result. (dcBL)
6. The application result is sent to customer by loan officer. (rtBL)

The flowchart of above process is shown in following figure:

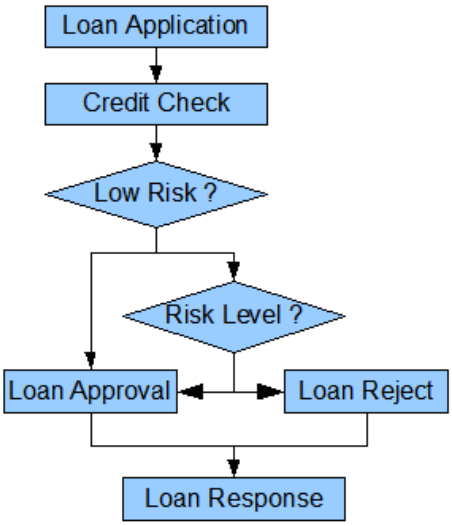

Figure 5-5-1 Bank Loan Application Flowchart

Based on this process, we can work out the WS-BPEL code for bank loan application.

```
<!-- WSBPEL syntax for loan process -->
<process name = "loanApprovalProcess"
     targetNamespace = "http://myloan.com/loanprocessing"
     xmlns="http://docs.oasis-open.org/wsbpel/2.0/process/executable"
     xmlns:lns = "http://myloan.com/loanprocessing/wsdl/">
<!-- Define the partners involved in -->
<partnerLinks>
     <!-- Customer application submission-->
     <partnerLink name="customer"
          partnerLinkType="lns:loanPartnerLinkType"
          myRole="loanService" />
     <!-- Agency providing customer credit rating-->
     <partnerLink name="creditCheck"
          partnerLinkType="lns:creditCheckLinkType"
          partnerRole="creditAgency" />
     <!-- Approver makes decision -->
```

```
<partnerLink name="approver"
     partnerLinkType="lns:loanApprovalLinkType"
     partnerRole="approver" />
</partnerLinks>
<!-- Variables -->
<variables>
     <variable name="loanRequest"
          messageType="lns:loanRequestMessage" />
     <variable name="creditRequest"
          messageType="lns:creditInformationMessage" />
     <variable name="loanDecision"
          messageType="lns:loanDecisionMessage" />
</variables>
<!-- receive loan request from customer -->
<sequence>
     <receive partnerLink = "customer" portType =
          "lns:loanPartnerLinkType"
          operation="request" variable="loanRequest"
          createInstance="yes">
     </receive>
     <assign>
          <copy>
               <from partnerLink="customer"/>
               <to variable="loanRequest"/>
          </copy>
     </assign>
     <flow>
          <links>
               <link name = "receive-to-assess" />
               <link name = "assess-to-approval" />
```

```
            <link name = "assess-to-approver" />
            <link name = "assess-to-response" />
            <link name = "approver-to-approval" />
            <link name = "approver-to-reply" />
            <link name = "approval-to-reply" />
</links>
<!-- creditCheck, high risk go to approver -->
<invoke partnerLink = "creditCheck"
     portType = "lns:creditCheckLinkType"
     operation="checkCredit" inputVariable="loanRequest"
     outputVariable="creditRequest">
<targets>
     <target linkName="receive-to-assess" />
</targets>
<sources>
     <source linkName="assess-to-approval">
          <transitionCondition> $loanRisk.level='low'
          </transitionCondition>
     </source>
     <source linkName="assess-to-approver">
          <transitionCondition>
               $loanRisk.level!='low'
          </transitionCondition>
     </source>
</sources>
</invoke>
<!--Approver makes decision-->
<invoke partnerLink = "approver"
     portType = "lns:loanApprovalLinkType"
     operation="approve"
```

```
                inputVariable="creditRequest"
                outputVariable="loanDecision">
                <targets>
                     <target linkName="approver-to-approval"/>
                </targets>
                <sources>
                     <source linkName="approval-to-reply" />
                </sources>
           </invoke>
           <!--Reply to costomer-->
                <reply partnerLink="customer"
                     portType="lns:loanPartnerLinkType"
                     operation="response"
                     variable="loanDecision">
                <targets>
                     <target linkName="approver-to-reply" />
                     <target linkName="approval-to-reply" />
                </targets>
                </reply>
           </flow>
     </sequence>
</process>
```

The <partnerLinks> indicates the participators in the bank loan application process: the customer who applies a bank loan, the credit agency who conducts the credit check and the approver who makes a decision.

In <sequence> element, the bank loan application process is conducted in the following order: the <receive> element designates that the loan request is received from a customer, the credit check is invoked by <invoke> element, the

risk level is described by <transitionCondition> element, the approver makes a final decision for the loan application be another <invoke> element, finally the loan response is sent back to the customer by <reply> element.

5.3 ACCESS CONTROL WITH RBAC

Although RBAC have been implemented by varieties of applications and can be represented in many ways, we choose to express our example in plain English to provide a universal understanding in this example. We also strictly adhere to the original RBAC to avoid any specific considerations and problems in particular systems.

In the bank loan application process, we can set some practical permissions and constraints according to RBAC model. Since the role is the core element in RBAC, we first describe the role hierarchy of a bank in the following figure:

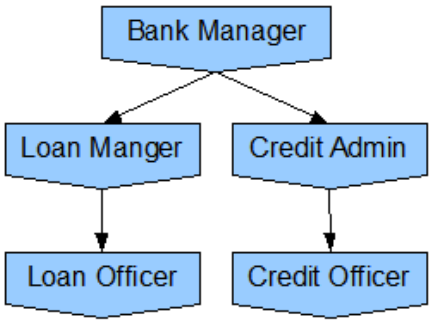

Figure 5-2 Role Hierarchy in a Bank

As we discussed above, the main idea in RBAC considers simple constraints that can be effectively checked and implemented. We can describe the access control requirements and constraints as follows:

1) Role assignment and permissions:

a) **BANK MANAGER** is on top of the role hierarchy. **BANK MANAGER** governs **LOAN MANAGER** and **CREDIT ADMIN**. **LOAN MANGER** governs **LOAN OFFICER** while **CREDIT ADMIN** governs **CREDIT OFFICER**.
b) **LOAN MANAGER** role and **CREDIT ADMIN** role can only be assigned to department managers.
c) Only **LOAN OFFICER** is permitted to handle loan application.
d) Only **CREDIT OFFICER** is permitted to conduct credit check.
e) High risk loan applications must proceed to **LOAN MANAGER**.
f) **LOAN OFFICER** provides final result to the applicant.

2) Mutually exclusive roles:
   a) The **LOAN OFFICER**, who receives loan application from customer A, must be the **LOAN OFFICER** provides loan response to customer A.
   b) **LOAN OFFICER** role and **CREDIT OFFICER** role must be assigned to different staff.
   c) **LOAN OFFICER** role and **LOAN MANAGER** role must be assigned to different staff.
3) Cardinality:
   a) There must be at least one staff assigned as **LOAN OFFICER**.
   b) There must be one staff and only one assigned as **BANK MANAGER**.
4) Prerequisite roles:
   a) A staff member is assigned to **CREDIT OFFICER** only when a new loan application received.
   b) A manager is assigned to **LOAN MANAGER** only when loan application is rated as high risk.

5.4 BPEL4RBAC MODEL AND POLICY LANGUAGE

We extend from original RBAC model to provide access control and authorisation constraints ability to existing WS-BPEL. This extended RBAC model

is called BPEL4RBAC model in our proposed architecture. Moreover, in order to provide WS-BPEL compatible access control policy specification, we facilitate the WS-BPEL extension mechanism to build up our BPEL4RBAC policy language.

Firstly, we provide formal definition of BPEL4RBAC model. Then, the bank loan application process is represented in this way as a running example. The BPEL4RBAC policy language is introduced in next subsection.

5.4.1 BPEL4RBAC Model

We extend existing RBAC model with considerations of business process.

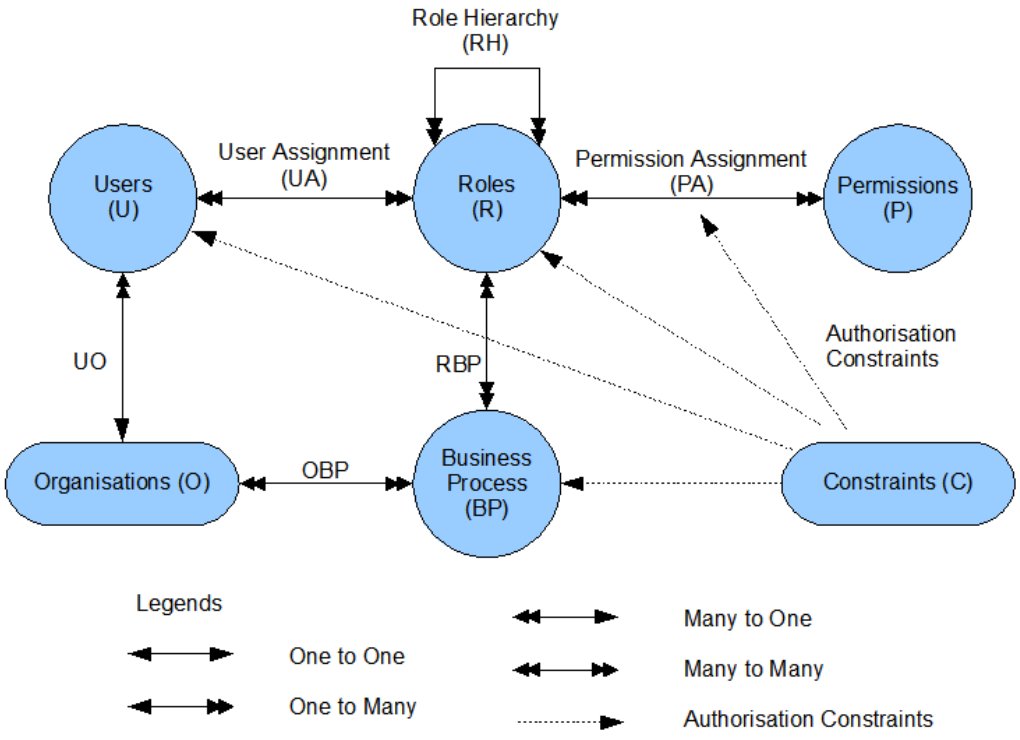

Figure 5-3 BPEL4RBAC model

In BPEL4RBAC, a `user` is human being belongs to an organisation. A `role` is a named job function within the business process context that regards the authority and responsibility. A `permission` is an approval of actions granted to specific roles. A `constraint` regulates the relations between different elements.

In order to provide access control capability to WS-BPEL, we add two elements to original RBAC model, namely organisation and business process.

An organisation is a group of users with structure of roles and responsibilities functioning to participating business processes. A business process is "a set of logically related tasks performed to achieve a well defined business outcome"[7].

User assignment (UA) and permission assignment (PA) are both many-to-many relationship since a user can be assigned to many roles and a role can have one or more users. Role hierarchy (RH) maps the nature structure of an organisation. User organisation (UO) relationship indicates which user belongs to which organisation. Organisation business process (OBP) relationship specifies which business process is developed or consumed by which organisation. Role business process (RBP) relationship describes which role is involved in which business process.

Definition 5-5-1 Basic Elements:

$U = \{u_1, u_2, \dots \dots u_i\}$, set of users;

$R = \{r_1, r_2, \dots \dots r_j\}$, set of roles;

$P = \{p_1, p_2, \dots \dots p_k\}$, set of permissions;

$UA \subseteq U \times R$, a many-to-many user to role assignment relation;

$PA \subseteq P \times R$, a many-to-many permission to role assignment relation;

$RH \subseteq R \times R$, a partial order on R, represents the role hierarchy.

Definition 5-5-2 BPEL4RBAC Business Process

$BP = \{bp_1, bp_2, \dots \dots bp_m\}$, set of business processes;

$BA = \{ba_1, ba_2, \dots \dots ba_n\}$, set of business activities;

$BPBA \subseteq BP \times BA$, a many-to-many business process to business activities assignment relation where $bpba = \{(ba, bp) \in BPBA | ba \in BA, bp \in BP\}$;

$bp: BP \longrightarrow 2^{BA}$, a function mapping each business process (bp) to the set of business activities;

$RBP \subseteq R \times BP$, a many to many role to business process, where $role: BP \longrightarrow 2^{R}$ is a function mapping each business process to the set of roles, where $role(r_i) = \{r | (u_i, r_j) \in UA\}$;

Definition 5-5-3 BPEL4RBAC Organisation

$O = \{o_1, o_2, \ldots \ldots o_i\}$, set of organisations;

$UO \subseteq U \times O$, a many to one user to organisation assignment relation, where $user: U \longrightarrow O$ is a function mapping user (u_i) to organisation (o_l), where $uo = \{(u, o) \in UO | u \in U, o \in O\}$;

$OBP \subseteq O \times BP$, a many to many organisation to business process assignment relation, where $o: BP \longrightarrow 2^{O}$ is a function mapping organisation (o_l) to business process (bp_m), where $obp = \{(o, bp) \in OBP | o \in O, bp \in BP\}$;

5.4.2 Access Control Schema in Bank Loan Process

With above definition, we can describe the access control schema as follows:

Role ID	Role Name
R1	Bank Manager
R2	Loan Manager
R3	Credit Admin
R4	Credit Officer
R5	Loan Officer

Table 5-1 Roles

RH ID	Hierarchy
RH1	Bank Manager > Loan Manager
RH2	Bank Manager > Credit Admin
RH3	Loan Manager > Loan Officer
RH4	Credit Admin > Credit Officer

Table 5-2 Role Hierarchy

UA ID	Role	User
UA1	Loan Manager	Only to department manager
UA2	Credit Manager	Only to department manager

Table 5-3 User Assignment

Permission ID	Role	Activity	BPEL Activity ID
P1	Loan Officer	handle loan application	apBL
P2	Credit Officer	conduct credit check	ccAP
P3	Loan Manager	handle high risk loan application	reAP
P4	Loan Officer	provide final result to applicant	rtBL

Table 5-4 Permissions

Constraint ID	Object(Activity)	Consequent Object (Subsequent Activity)	Condition	Related BPEL Activity ID
C1	Loan Officer (Handles loan application	Loan Officer (Provides loan response	Same User	apBL, rtBL

	from customer A)	to customer A)		
C2	Loan Officer	Credit Officer	Different User	apBL, ccAP
C3	Loan Officer	Loan Manager	Different User	rkBL, reAP
C4	Loan Officer		At least one user assigned to this role	apBL
C5	Bank Manager		Only one user assigned to this role	
C6	Credit Officer (A staff assigned to this role)		Only when a new loan application received	rkBL
C7	Loan Manager (A staff assigned to this role)		Only when loan application is rated as high risk	reAP

Table 5-5 Constraints

5.4.3 BPEL4RBAC Policy Language

As an extension to WS-BPEL, BPEL4RBAC policy language is layered on top of WS-BPEL. Its features can be aggregated with WS-BPEL features during the business processes. The extension introduces a set of elements to provide role based access control capability.

The root element in BPEL4RBAC is `<policy>`. The basic elements in BPEL4RBAC language are `<user>`, `<role>`, `<permission>`, `<organisation>`, `<business process>` and `<constraints>`. In order to differentiate BPEL code and BPEL4RBAC extension, we use "`b4r`" prefix to indicate BPEL4RBAC namespace and "`bpel`" prefix to designate BPEL code. The overall syntax is shown as follows:

```
<bpel:process name = "NCName"
     ...
     xmlns:b4r = "http://myloan.example.com/bpel4rbac"
     ...>
     <bpel:extensions
          namespace = "http://myloan.example.com/bpel4rbac"
          mustUnderstand = "yes">
     </bpel:extensions>
     <bpel:extensionActivity>
          <b4r:policy>
               <b4r:roles>
                    <b4r:role ID = "NCName">role</b4r:role>+
               </b4r:roles>
               <b4r:roleHierarchys>
                    <b4r:roleHierarchy ID = "NCName">+
                         <b4r:seniorRole>senior role</b4r:seniorRole>
                         <b4r:juniorRole>junior role</b4r:juniorRole>
                    </b4r:roleHierarchy>
               </b4r:roleHierarchys>
               <b4r:userAssignments>
                    <b4r:userAssignment ID = "NCName">+
                         <b4r:role>role</b4r:role>
                         <b4r:assignment>
                              user to role assignment
                         </b4r:assignment>
                    </b4r:userAssignment>
               </b4r:userAssignments>
               <b4r:permissions>
                    <b4r:permission ID = "NCName">+
                         <b4r:role>role</b4r:role>
```

```
                <b4r:permittedActivity>
                    permitted activity for this role
                </b4r:permittedActivity>
            <b4r:bpelActivity>
                related BPEL activity
            </b4r:bpelActivity>
            </b4r:permission>
        </b4r:permissions>
        <b4r:constraints>
            <b4r:constraint ID = "NCName">+
                <b4r:object>object name</b4r:object>
                <b4r:activity>
                    business process activity</b4r:activity>?
                <br4:bpelActivity>
                    related BPEL activity</br4:bpelActivity>?
                <b4r:consequentObject>
                    object name</b4r:consequentObject>?
                <b4r:subsequentActivity>?
                    business process activity
                <b4r:subsequentActivity>
                <br4:bpelActivity>
                    related BPEL activity
                </br4:bpelActivity>?
                <b4r:constraintAssertion>
                    condition
                </b4r:constraintAssertion>
            </b4r:constraint>
        </b4r:constraints>
        <b4r:businessProcesses>
            <b4r:businessProcess ID ="NCName">+
```

```
                        <b4r:activity>
                             business process activity
                        </b4r:activity>
                        <br4:bpelActivity>
                             related BPEL activity
                        </b4r:bpelActivity>
                        <bpel:partnerLink name = "NCName"/>
                    </b4r:businessProcess>
               </b4r:businessProcesses>
          </b4r:policy>
     </bpel:extensionActivity>
</bpel:process>
```

With the extensibility of WS-BPEL, the `<bpel:extensions>` element imports the BPEL4RBAC extension to BPEL code. The `<b4r:policy>` is the root element of proposed extension. All other elements and activities in BPEL4RBAC are enclosed.

The `<b4r:roles>`, `<b4r:roleHierarchys>`, `<b4r:userAssignments>`, `<b4r:permissions>`, `<b4r:constraints>` and `<b4r:businessProcesses>` indicate group of roles, role hierarchy, user assignment, permissions, constraints and business processes respectively.

The `<b4r:role>` element is used to define a role in an organisation while the `<b4r:roleHierarchys>` element specifies the role hierarchy. The roles definition and role hierarchy illustrated in Table 6-1 and 6-2 can be encoded as:

```
     <b4r:roles>
          <b4r:role ID = "R1">Bank Manager</role>
          ...
          <b4r:role ID = "R5">Loan Officer</role>
     </b4r:roles>
```

```
<b4r:roleHierarchys>
     <b4r:roleHierarchy ID = "RH1">
          <b4r:seniorRole>Bank Manager</b4r:seniorRole>
          <b4r:juniorRole>Loan Manager</b4r:juniorRole>
     </b4r:roleHierarchy>
     ...
</b4r:roleHierarchys>
```

The <b4r:userAssignment> element defines user assignment in a business process while <b4r:permission> element describes the permission imposed on a specific role or activity. The user assignment and permissions illustrated in Table 6-3 and Table 6-4 can be encoded as:

```
<b4r:userAssignments>
     <b4r:userAssignment ID = "UA1">
          <b4r:role>Loan Manager</b4r:role>
          <b4r:assignment>
               Only to department managers
          </b4r:assignment>
     </b4r:userAssignment>
     ...
</b4r:userAssignments>

<b4r:permissions>
     <b4r:permission ID = "P1">
          <b4r:role>Loan Officer</b4r:role>
          <b4r:permittedActivity>
               handle loan application
          </b4r:permittedActivity>
```

```
            <b4r:bpelActivity>apBL</b4r:bpelActivity>
        </b4r:permission>
        ...
    </b4r:permissions>
```

The <b4r:constraint> element is used to define access constraint in business process. The <b4r:object> element and <b4r:consequentObject> element designate the role or user whom the constraint applies on. The <b4r:activity> element and <b4r:subsequentActivity> element indicate which activity or process is restrained by this constraint. The <b4r:bpelActivity> element describes the association between the activity or process in WS-BPEL with the activity in BPEL4RBAC. The <b4r:constraintAssertion> is the assertion that describes the constraint condition. The <b4r:object> and <b4r:constraintAssertion> elements are compulsory while other elements are optional according to the constraint. The following code illustrates the constraints in Table 5-5:

```
    <b4r:constraints>
        <b4r:constraint ID = "C1">
            <b4r:object>Loan Officer</b4r:object>
            <b4r:activity>
                Handles loan application from customer A
            </b4r:activity>
            <br4:bpelActivity>apBL</b4r:bpelActivity>
            <b4r:consequentObject>Loan Officer</b4r:consequentObject>
            <b4r:subsequentActivity>
                Provides loan response to customer A
            </b4r:subsequentActivity>
```

```
            <br4:bpelActivity>rtBL</b4r:bpelActivity>
            <b4r:constraintAssertion>Same User</b4r:constraintAssertion>
        </b4r:constraint>
        ...
        <b4r:constraint ID = "C7">
            <b4r:object>Loan Manager</b4r:object>
            <b4r:activity>
                A staff assigned to this role
            <b4r:activity>
            <br4:bpelActivity>reAP</b4r:bpelActivity>
            <b4r:constraintAssertion>
                Only when loan application is rated as high risk
            </b4r:constraintAssertion>
        </b4r:constraint>
    </b4r:constraints>
```

The `<b4r:businessProcess>` element associates the activity in WS-BPEL and BPEL4RBAC by `<b4r:activity>`, `<br4:bpelActivity>` and `<bpel:partnerLink>` elements correspondingly. The following is an illustrating example code:

```
    <b4r:businessProcesses>
        <b4r:businessProcess ID ="BP1">
            <b4r:activity>Handle loan application</b4r:activity>
            <br4:bpelActivity>apBL</b4r:bpelActivity>
            <bpel:partnerLink name = "customer"/>
        </b4r:businessProcess>
        ...
        <b4r:businessProcess ID ="BP6">
            <b4r:activity>
```

```
                Send application result to customer
            </b4r:activity>
            <br4:bpelActivity>rtBL</b4r:bpelActivity>
            <bpel:partnerLink name = "customer"/>
        </b4r:businessProcess>
    </b4r:businessProcesses>
```

5.5 SUMMARY

BPEL4RBAC extends its ability from both RBAC side and WS-BPEL side. The greatest advantage of BPEL4RBAC over others is the high compatibility with WS-BPEL standard since BPEL4RBAC policy language is an extension of latest WS-BPEL specification. This ensures the access control functions can be seamlessly integrated into WS-BPEL. The system architecture also provides the adaptability with other security standards to enhance its security level further. Moreover, BPEL4RBAC can be extended to other standards apart from XACML or WS-Policy based standards as long as they can be adapted in accordance with WS-BPEL.

In this chapter, we have proposed our BEPL4RBAC authorisation specification which supports the access control capability in business process environment. The BPEL4RBAC extends the classical RBAC model with organisation and business process elements appended. These two elements are essential for representing access control information in business process scenario. The BPEL4RBAC policy language is also formally defined to describe authorisation information. The access control and authorisation requirements illustrated in BPEL4RBAC model can be mapped into this policy language. All this information is integrated with WS-BPEL seamlessly. The system architecture investigates the feasibility of BPEL4RBAC. With the separation of Access Enforcement Module and Access Decision Module, access decision strategies and security policies can be

developed by physically isolated users or organisations. These strategies and policies might be changed very frequently according to the real world need. Thus, BPEL4RBAC system ensures the availability and performance scalability in heavy duty business process environment.

CHAPTER 6

ACCESS CONTROL FOR HUMAN TASKS

Business process management is designed to make business activities and trade easier and more cost effective. WS-BPEL and BPEL4People extension together coordinate the Web services and human activities within business process. However, the increasing business integration and legal requirements raise the need for secure business processes. The openness and distribution nature of inter-organisational business processes may result in more security breaches. Existing standards does not provide any support for business process security protection even if the participating organisations already have a working security policy. To address this problem, we extend traditional RBAC model to access control capability into business process environment. And an extension for WS-BPEL is also developed to represent the authorisation information in a formal manner.

6.1 BUSINESS PROCESS AND ACCESS CONTROL

In this section, we illustrate WS-BPEL, BPEL4People and access control model separately by running examples. For easy understanding, we use simplified bank loan application scenario. First, a customer applies for bank loan. Then the bank conducts credit check and property value assessment by 3rd party partners. The risk assessment service will handle the application and make a decision according to its risk level. Finally the application is approved to the eligible applicant while unacceptable high risk applications are rejected by the human approver. The UML activity diagram for this loan application process is shown in Figure 6-1.

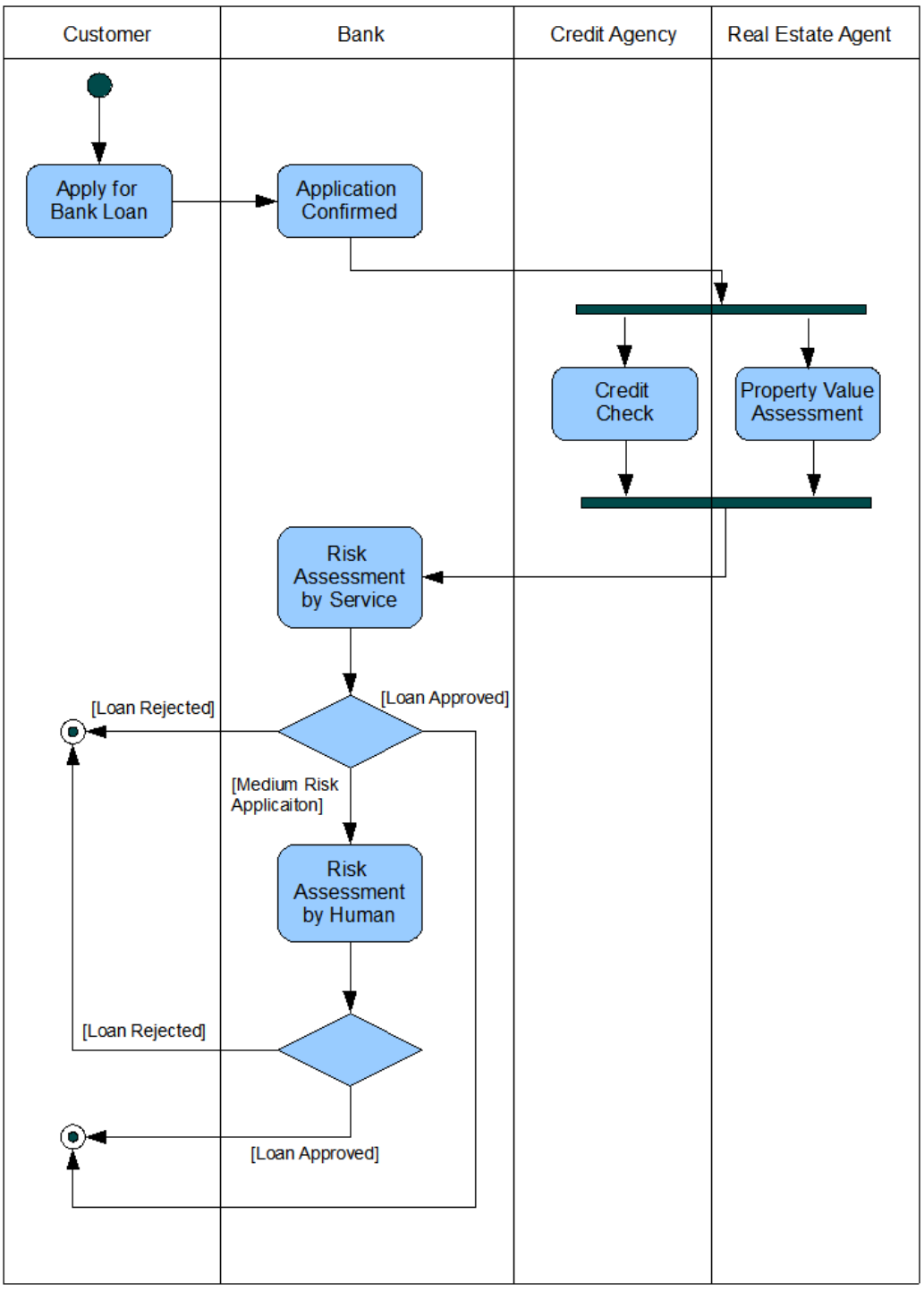

Figure 6-1 Home Loan Application Activity Diagram

In this scenario, the business process can be formalised by WS-BPEL 2.0 code with BPEL4People extension which is shown as follows.

```
<!-- WSBPEL syntax for loan process -->
<process name = "loanApprovalProcess"
     targetNamespace = "http://myloan.example.com/loanprocessing"
     xmlns = "http://docs.oasis-open.org/wsbpel/2.0/process/executable"
     xmlns:lns = "http://myloan.example.com/loanprocessing/wsdl/">
     <!-- Define the partners involved in -->
     <partnerLinks>
          <!-- Customer application submission-->
          <partnerLink name="customer"
               partnerLinkType="lns:loanPartnerLinkType"
               myRole="loanService" />
          <!-- Risk assessment web service-->
          <partnerLink name="riskAssess"
               partnerLinkType="lns:riskAssessLinkType"
               partnerRole="riskAssessment" />
          <!-- Human approver makes decision -->
          <partnerLink name="approver"
               partnerLinkType="lns:loanApprovalLinkType"
               partnerRole="approver" />
     </partnerLinks>
     <!-- Variables -->
     <variables>
          <variable name="loanRequest"
               messageType="lns:loanRequestMessage" />
          <variable name="riskAssess"
               messageType="lns:riskAssessmentMessage" />
          <variable name="loanDecision"
               messageType="lns:loanDecisionMessage" />
     </variables>
     <!-- receive loan request from customer -->
```

```
<sequence>
     <receive partnerLink = "customer"
          portType = "lns:loanPartnerLinkType"
          operation="request" variable="loanRequest"
          createInstance="yes">
     </receive>
     <assign>
          <copy>
               <from partnerLink="customer"/>
               <to variable="loanRequest"/>
          </copy>
     </assign>
     <flow>
          <links>
               <link name = "receive-to-assess" />
               <link name = "assess-to-approval" />
               <link name = "assess-to-approver" />
               <link name = "assess-to-response" />
               <link name = "approver-to-approval" />
               <link name = "approver-to-reply" />
               <link name = "approval-to-reply" />
          </links>
          <!-- Risk assessment by web service -->
          <invoke partnerLink = "creditCheck"
               portType = "lns:riskAssessLinkType"
               operation="riskAssess"
               inputVariable="loanRequest"
               outputVariable="creditRequest">
               <targets>
                    <target linkName="receive-to-assess" />
```

```
            </targets>
            <sources>
                <source linkName="assess-to-approval">
                    <transitionCondition>
                        $loanRisk.level='low'
                    </transitionCondition>
                </source>
                <source linkName="assess-to-approver">
                    <transitionCondition>
                        $loanRisk.level='medium'
                    </transitionCondition>
                </source>
                <source linkName="assess-to-response">
                    <transitionCondition>
                        $loanRisk.level='high'
                    </transitionCondition>
                </source>
            </sources>
        </invoke>
        <!--Human task activities-->
        <htd:tasks>
            <htd:task name="assessLoanApplication">
                <htd:documentation xml:lang="en-UK">
                    The task is used to assess loan risk.
                </htd:documentation>
                <htd:peopleAssignment>
                    <htd:potentialOwners>
                        <htd:from
                            logicalPeolpleGroup="loanManagers">
                        </htd:from>
```

```
                    </htd:potentialOwners>
                </htd:peopleAssignment>
            </htd:task>
        </htd:tasks>
        <!--Human approver makes decision-->
        <extensionActivity>
            <b4p:peopleActivity
                name="loanRiskAssessment"
                inputVariable="loanApplication"
                outputVariable="applicationResult">
                <b4p:localTask
                    reference="tns:assessLoanApplication">
            </b4p:peopleActivity>
        </extensionActivity>
        <!--Reply to costomer-->
        <reply partnerLink="customer"
            portType="lns:loanPartnerLinkType"
            operation="response"  variable="loanDecision">
            <targets>
                <target linkName="approver-to-reply" />
                <target linkName="approval-to-reply" />
            </targets>
        </reply>
    </flow>
  </sequence>
</process>
```

WS-BPEL is designed to describe business processes in a structured way. The business logic is expressed as a group of activities which are performed by invoking Web services. The `<process>` element is on the top level of WS-BPEL

specification. The attributes of <process> specifies the process name and related namespace.

From the system point of view, <partnerLink> element indicates the external Web services to be invoked from this process. The <variable> element defines the data variables involved in this process. Some basic control structures are also deployed to describe business logic. The <sequence> element contains sequentially executed activities while the <flow> element specifies concurrently performed activities. These elements may contain one or more basic elements such as <receive> and <reply> which define the message flows in a process.

The <partnerLinks> indicates the participators in the bank loan application process: the customer who applies a bank loan, the credit agency who conducts the credit check and the approver who makes a decision.

In <sequence> element, the bank loan application process is conducted in the following order: the <receive> element designates that the loan request is received from a customer, the credit check is invoked by <invoke> element, the risk level is described by <transitionCondition> element, the approver makes a final decision for the loan application be another <invoke> element, finally the loan response is sent back to the customer by <reply> element.

In BPEL4People extension, <htd:task> element defines a task within the people activity. The <htd:potentialOwners> element designate the people in charge of this activity from pre-defined user groups. The <b4p:peopleActivity> element refers to the task to be performed within the business process. In the example, a loan manager from the manager group is assigned to handle the loan application.

6.2 ACCESS CONTROL IN SOA

Although the concept of role has existed for a long time in systems security, the work presented by Sandhu in [5] has prompted a renewed interest in this approach. This greatly simplifies security management [6]. RBAC model is now adopted in many commercial products to different degrees since access control is an important requirement of information systems. RBAC was found to be the most attractive solution for providing security characteristics in inter-organisational business systems [68]. Moreover, it would be much easier for organisations to enhance security protection from existing RBAC based policies.

6.2.1 Traditional RBAC Model

In RBAC, a `user` is human being belongs to an organisation. A `role` is a named job function within the business process context that regards the authority and responsibility. A `permission` is an approval of actions granted to specific roles. A `constraint` regulates the relations between different elements.

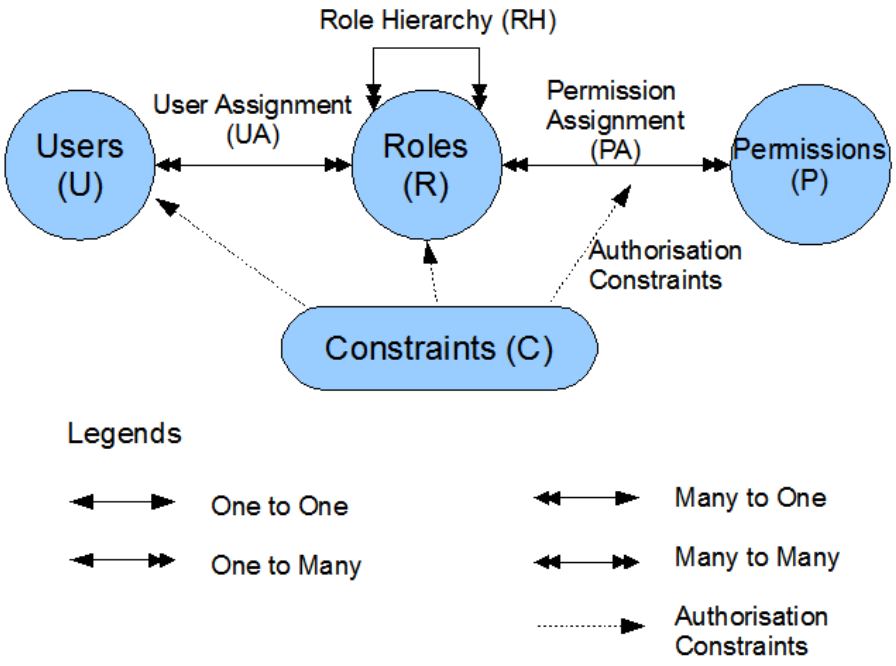

Figure 6-2 Traditional RBAC Model

6.2.2 Extended RBAC Model

In order to provide access control capability to WS-BPEL and BPEL4People, we add a `service` element to original RBAC model, which indicates the Web service deployed within the enterprise system. We also divide `roles` into `human role` and `computer role`. The human role indicates the tasks to be performed by human users, while the computer role indicates the tasks to be performed by Web services.

`User assignment` (UA) and `service assignment` (SA) are both many-to-many relationship since a user or service can be assigned to many roles and a role can have one or more users. `Role hierarchy` (RH) maps the nature structure of an organisation.

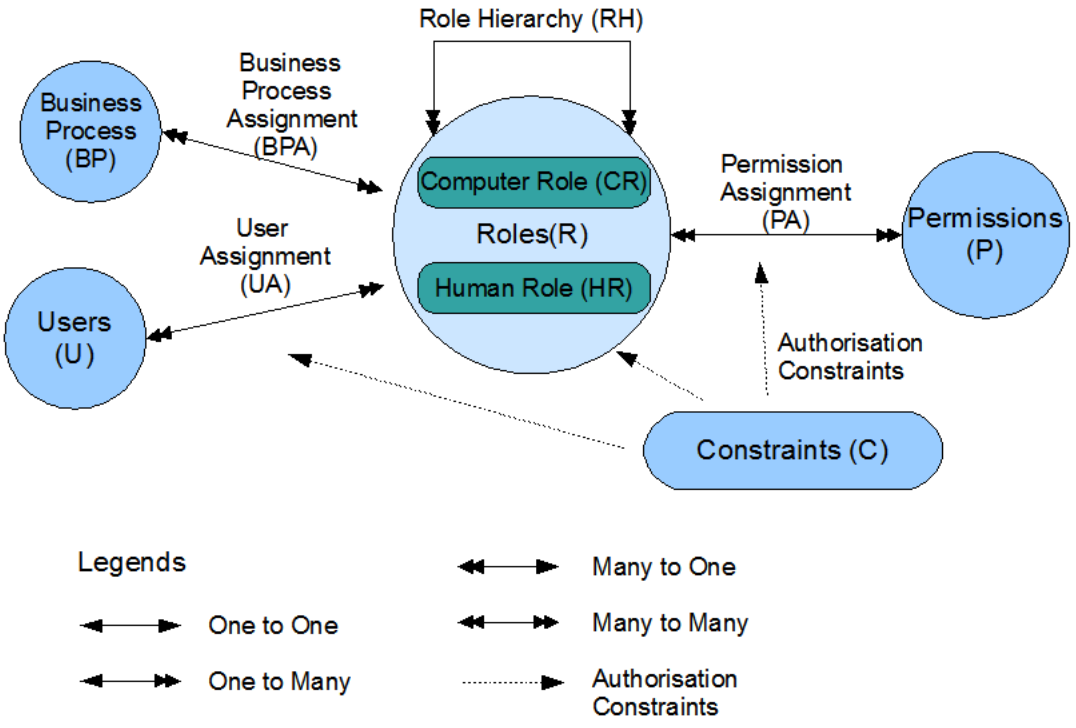

Figure 6-3 Extended RBAC Model

Role hierarchy maps the nature structure of organisations. The function of a role can be implemented by a human user or a Web service. So we don't split

roles into two different elements. From the organisational point of view, there is no difference between these two kinds of roles. The difference only resides in information system level. In most SOA scenarios, the organisation may replace the human functions by services step by step. Our proposed model extension addresses the SOA upgrade in this kind of progressive manner. We can define the extended model as follows:

Definition 6-6-1 Basic Elements:

$U = \{u_1, u_2, \ldots \ldots u_i\}$, set of users;

$R = \{r_1, r_2, \ldots \ldots r_j\}$, set of roles;

$P = \{p_1, p_2, \ldots \ldots p_k\}$, set of permissions;

$PA \subseteq P \times R$, a many-to-many permission to role assignment relation;

$RH \subseteq R \times R$, a partial order on R, represents the role hierarchy.

Definition 6-6-2 RBAC Extension

$S = \{s_1, s_2, \cdots\cdots s_l\}$, set of Web services;

$HR = \{hr_1, hr_2, \cdots\cdots hr_m\}$, set of human roles;

$CR = \{cr_1, cr_2, \cdots\cdots cr_n\}$, set of computer roles;

HR and CR are subset of R, where $CR \cup HR = R$ and $CR \cap HR = \phi$

$UA \subseteq U \times HR$, a many-to-many user to human role assignment relation;

$SA \subseteq S \times CR$, a many-to-many service to computer role assignment relation;

6.3 BPEL EXTENSION

In order to provide access control capability to business process engine, we design an adapter to accommodate the access control constraints. Figure 6-4 illustrates the architecture of the proposed adapter.

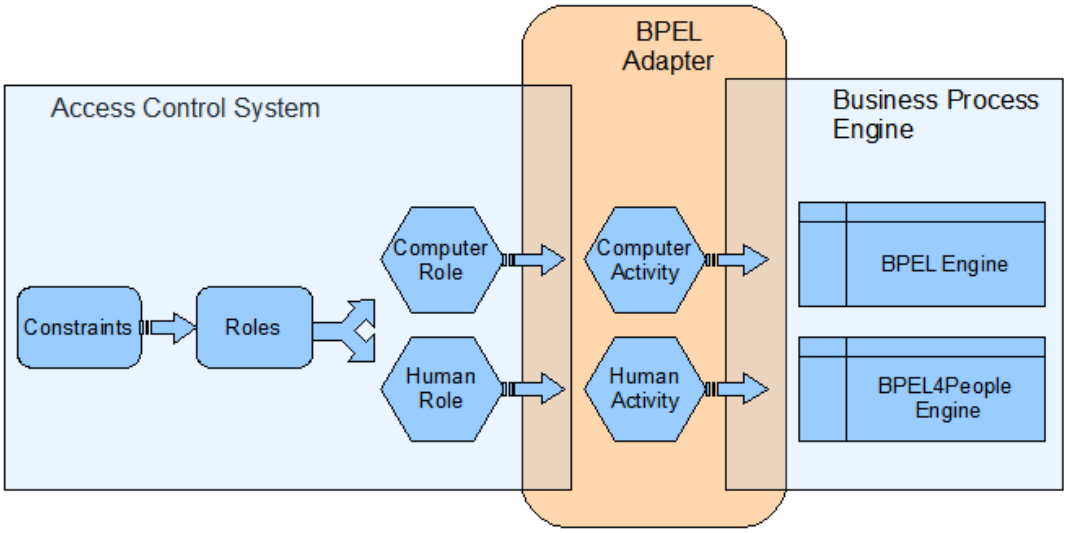

Figure 6-4 System Architecture for BPEL Extension

The Web services can be represented in WS-BPEL while the human tasks are described in BPEL4People. In order to integrate RBAC into business process, we need to extend both WS-BPEL and BPEL4People specification to map these requirements. The extension is layered on top of WS-BPEL and BPEL4People. Its features can be aggregated with Web service and human activity features during the business processes. The extension introduces a set of elements to provide role based access control capability.

In order to differentiate BPEL code and BPEL4People extension, we use “bpac” prefix to indicate namespace for our proposed extension. The overall syntax is shown as follows:

```
<bpac:constraints>
     <bpac:constraint>+
```

```
          <bpac:role>
               <bpac:humanRole>Role Name</bpac:humanRole>
| <bpac:computerRole>Role Name</bpac:computerRole>
          </bpac:role>
          <bpac:permission>
               <bpac:object reference=”NCNAME”>Name</bpac:object>
               <bpac:action>Name</bpac:action>
          </bpac:permission>
     </bpac:constraint>
</bpac:constraints>
```

The root element for the access control extension is `<bpac:constraints>` which includes one or more <bpac:constraint>. Each <bpac:constraint> defines a <bpac:role> and the <bpac:permission> for this role. As we discussed above, the role can be a human role or a computer role which are represented as <bpac:humanRole> and <bpac:computerRole> respectively. The <bpac:object> describes which task that the role can participate in. The reference attribute indicates the human task for human role or port type for computer role. The <bpac:action> element defines which action that the role can perform. The access control constraint for loan manager and risk assessment service can then be coded as follows:

```
<bpac:constraints>
     <bpac:constraint>
          <bpac:role>
               <bpac:humanRole>loan manager</bpac:humanRole>
          </bpac:role>
          <bpac:permission>
               <bpac:object reference="lns:riskAssessLinkType">
                    home loan application
```

```
                </bpac:object>
                <bpac:action>make decision</bpac:action>
           </bpac:permission>
     </bpac:constraint>
<bpac:constraint>
           <bpac:role>
                <bpac:computerRole>
                     risk assessment service
                </bpac:computerRole>
           </bpac:role>
           <bpac:permission>
                <bpac:object reference="tns:assessLoanApplication">
                     home loan application
                </bpac:object>
                <bpac:action>assess risk</bpac:action>
           </bpac:permission>
     </bpac:constraint>
</bpac:constraints>
```

6.4 ACCESS CONTROL CONSTRAINTS

In some real world scenarios, organisation may already have established their security policies in their information systems. For example, XACML (eXtensible Access Control Markup Language) [65] is a standard language developed by OASIS for describing access control policies. In the home loan application process, we can define the loan manager human role as follows:

```
<PolicySet xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
     PolicySetId="PPS:manager:role"
     PolicyCombiningAlgId="&policy-combine;permit-overrides">
```

```
<!-- Permissions for the loan manager role -->
<Policy PolicyId="Permissions:specifically:for:the:loan:manager:role"
     RuleCombiningAlgId="&rule-combine;permit-overrides">
     <!-- Permission to make decision for loan application -->
     <Rule RuleId="Permission:to:approve:a:loan:application"
          Effect="Permit">
          <Target>
               <Resources>
                    <Resource>
                    <ResourceMatch
                         MatchId="&function;string-equal">
                         <AttributeValue
                              DataType="&xml;string">
                                   home loan application
                              </AttributeValue>
                         <ResourceAttributeDesignator
                              AttributeId="&resource;resource-id"
                              DataType="&xml;string"/>
                    </ResourceMatch>
                    </Resource>
               </Resources>
               <Actions>
                    <Action>
                         <ActionMatch
                              MatchId="&function;string-equal">
                              <AttributeValue
                                   DataType="&xml;string">
                                        make decision
                              </AttributeValue>
                              <ActionAttributeDesignator
```

```
                                        AttributeId="&action;action-id"
                                        DataType="&xml;string"/>
                                </ActionMatch>
                            </Action>
                        </Actions>
                    </Target>
                </Rule>
            </Policy>
</PolicySet>
```

BPEL and BPEL4People handle the separation of human activity and Web service. Access control language can only focus on how to describe the constraints and policies for each activity. Thus, any security policy language can be integrated into our proposed access control extension. For XML based language, we can adopt XSLT to transform them into required format.

6.5 HUMAN AND WEB SERVICE PATTERNS

In the current business practice, legacy business application, Web service based applications, pure human interactions and BPEL4People standardised human activities are involved and mixed. We have differentiated some patterns to apply RBAC into traditional applications, WS-BPEL and BPEL4People based applications.

In the world before service-oriented architecture, only business applications, and human users are involved. The human user will be able to access certain application on the conditions that the user is authorised to the permitted role and the role has the required permission to access this application. The following figure is a simple scenario in traditional RBAC enabled business application architecture.

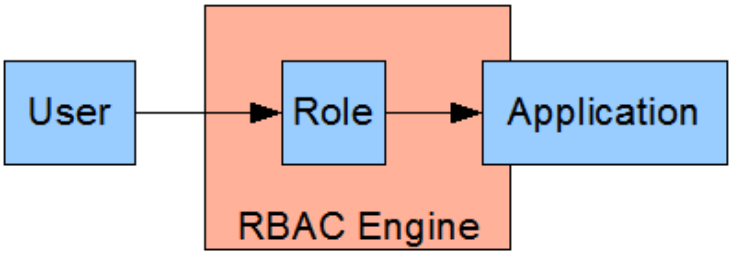

Figure 6-5 Traditional RBAC enabled business application

From the system point of view, business applications are basic forms of business collaboration, which means applications talk to each other, such as EDI introduced in Chapter 2.

Even in the age of service-oriented computing, the business rules and business logic might have been already programmed in above legacy system applications for many years. To cope with more and more types of applications, the Web services provide interface and interoperable method in a standardised way to communicate with other applications from heterogeneous systems.

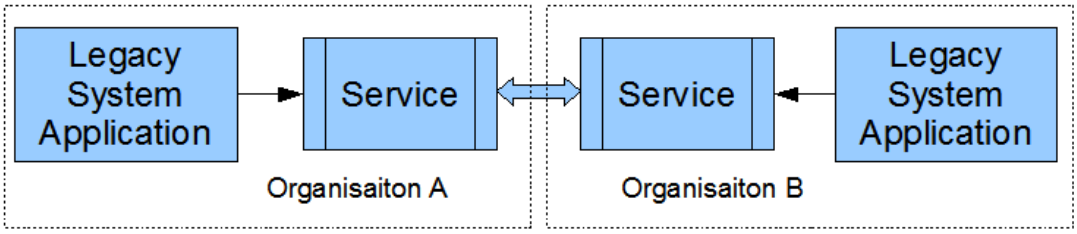

Figure 6-6 Web Service based business collaboration

In the perfect world, all manual work should be replaced by Web services and exposed to all potential partners. However, human activity is inevitable in the near future in the collaborative business environment. Even for the organisations who are adopting SOA, the process might take several years. In the meantime,

the IT system would appear as hybrid forms mixing computer services and human activities.

In order to discover different patterns of automatic and human services, we first revisit the example discussed above from the partnership point of view.

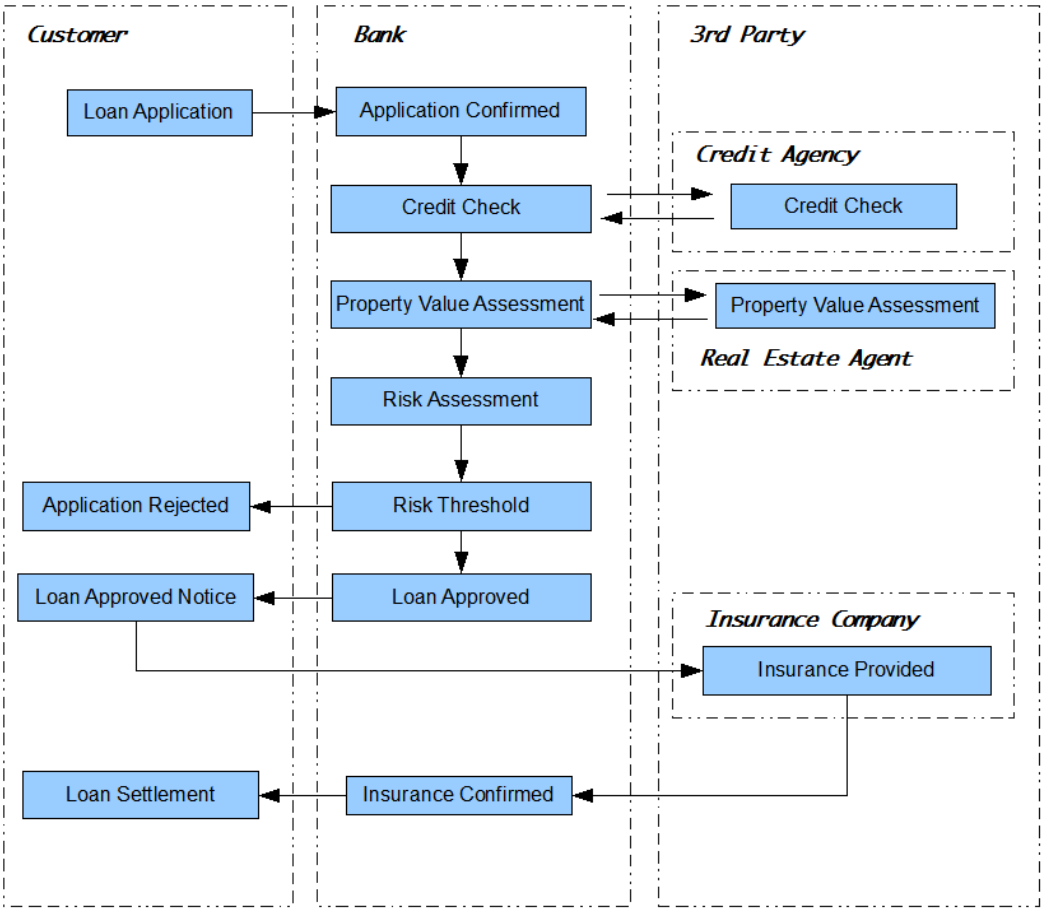

Figure 6-7 Home Loan Scenario Partners

Hybrid Model

In hybrid model, both computer services and human activities will need to communicate with business partners.

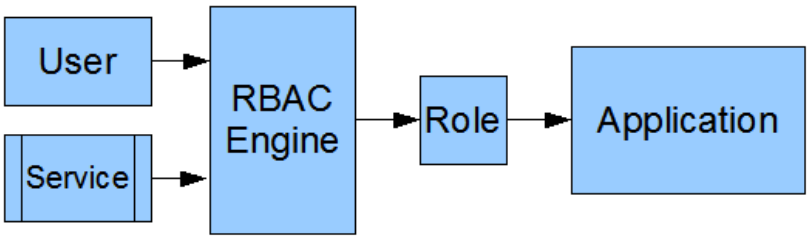

Figure 6-8 Hybrid Model

A service is treated as a user. In other word, service is a special user group. A service will perform the function that used to be performed by human user. This kind of service can be categorised as computer user.

So the service in this pattern will gain the same permission as their human counterpart. They will need to send their request to the RBAC engine to get the required role to communicate with application.

Role Model

In the role model, a service is able to access the required role. There is no need to be verified by RBAC engine for certain tasks performed by computer services. Human users still need to go through the RBAC engine to attain a role to contact with business partners.

This model facilitates the integration with legacy system users and Web services. But the service part needs to be carefully linked to roles.

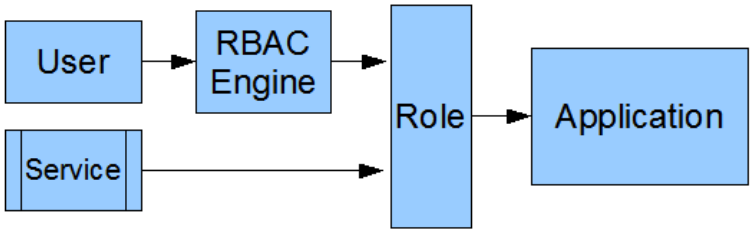

Figure 6-9 Role Model

An example role model is showing below.

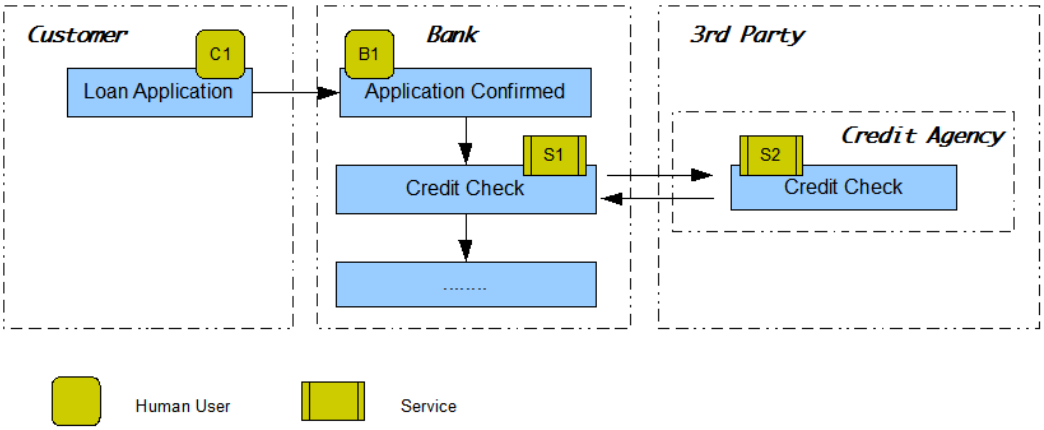

Figure 6-10 Role Model Example

There is no separation of duty requirement between B1 and S1 since S1 is service. So S1 will have the required role to contact credit agency to get customer's credit record.

Service Model

In the service model, computer service is the leading role in the collaborative process. Service is designed to communicate with other applications.

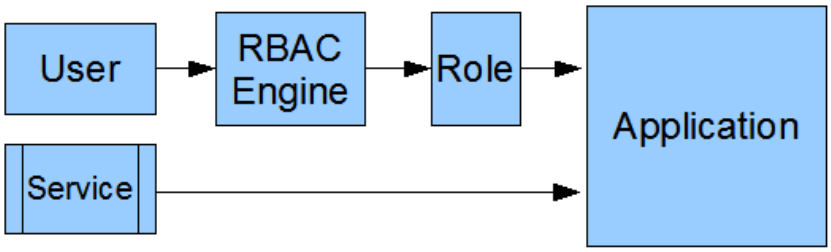

Figure 6-11 Service Model

Example:

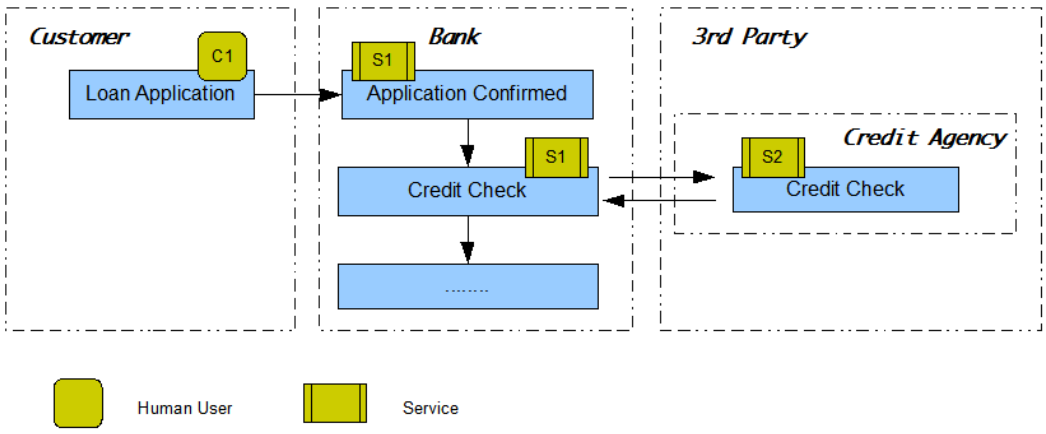

Figure 6-12 Service Model Example

Service S1 will handle the customer's application instead of human user. For example, the customer lodges the application online from the Bank's portal. S1 will handle the application and directly contact S2 from credit agency.

6.6 SUMMARY

A human activity involved access control architecture is presented in this work to address this missing security issue for both Web service and human activity in service-oriented environment. The RBAC model is extended with service element which is essential in Web service processable business functions. On top of the

extended RBAC model, we extend from WS-BPEL side to accommodate access control capability. The proposed adapter can integrate the security constraints into WS-BPEL and BPEL4People seamlessly. Besides the compatibility with these standards, existing legacy IT resources, such as XACML based security policies can also be mapped onto our proposed architecture which provides better aid for SOA migration.

In the next step, we will improve the access control policy with broader security policies by taking consideration of more complicated organisation behaviours. In particular, an organisation may define different roles and different permissions to the same process when cooperating with different partners. The patterns of human activity involvement and business rules will be investigated in this phase. By attaining this task, we can approach a blueprint of a framework for secure business process management system.

CHAPTER 7 AUTHORISATION VERIFICATION - ROLE-NET

Collaborative business can become unreliable in terms of authorisation policy conflicts; for example, when (1) incorrect role assignment or modification occurs in a service within one organisation or (2) messages transferred from one organisation are accessed by unqualified roles in other collaborating business partners. Therefore reliability verification based on access policies is critical for business collaboration. The role authorisation model, Role-Net, is developed to specify and manage role authorisation in business collaboration based on Hierarchical Colored Petri Nets (HCPNs). Moreover, a property named Role Authorisation Based Dead Marking Freeness is defined based on Role-Net to verify business collaboration reliability according to partners' authorisation policies.

Business collaboration is about coordinating the flow of information among organisations and linking their business processes into a cohesive whole. Emerging Web service and business process technologies have provided technological support for business collaboration across organisation boundaries [7]. However security concerns have become one of the main barriers that prevent widespread adoption of this new technology [109]. Models and methods are required to develop and manage secured business collaboration.

7.1 ROLE-BASED AUTHORISATION IN BUSINESS COLLABORATION

We can observe from the above motivating scenario that collaboration authorisation control and enforcement is governed by the authorisation policies of collaborating organisations. Role Based Access Control (RBAC) [110] [5] is a popular security paradigm where users are assigned with roles in order to gain certain permissions to access messages or perform tasks. Hence, RBAC is

normally used to define authorisation policy for managing tasks in an organisation [66].

Any authorisation policy conflicts within or across organisation can lead to unreliable business collaboration as follows:

Incorrect role assignment or modification in a service within one organisation: As we discussed before, any message shall be associated with a required role before it can be processed in a service, and an actual role assigned to process the message in the service. If this 'required role' is not consistent with the role assignment for this coming message in the service, we can conclude that the role assignment is incorrect and authorisation policy has conflict. For instance, when the Value Assessment service is added in the collaboration after the Credit Check service in Bank to evaluate Customer's collateral, Value auditor is the exclusive role assigned to access message in this new service. However, if value audit manager is the required role for accessing this message (note, this requirement may come from previous service that processed this message), then the qualified role does not exist in the service, and the business process will be suspended due to authorisation policy conflict. Let us look at another example in relation to role modification. The risk administrator and risk officer are two possible roles permitted to access a message in the Risk Assessment service depending on the amount of loan applications transferred from credit check service. The risk officer can process small amount loan applications while the risk administrator can deal with both huge and small amount loans. However, when one huge amount loan application is being executed in the collaboration, the permission of the risk administrator may be modified at runtime, e.g., the risk administrator's permission to access the Risk Assessment service is removed. Therefore, no qualified role who has right permission to handle the huge amount loan application which requires risk administrator to evaluate the risk.

Unexpected role access in collaborating business partner: Business partners are peers with their own authorisation policies that are agnostic to each other. Therefore, without central control, it is difficult to guarantee that the message is accessed by the expected roles in business partners' service. For example, due to privacy reason for corporate clients, message sent from Credit Check Requirement service at Bank side may require to be accessed by Credit Check Manager at the Credit Check service in the Credit Rating Agency. However, if the message is accessed and modified by an unexpected role in Credit Rating Agency, e.g. a general credit check officer, then Bank may not accept the credit check result.

In summary, in business collaboration environment, role assignments in different services can be incorrect and role's permissions in existing services can be modified. Verification on such improvisational variation of role authorisation is thereby critical to manage business collaboration reliability. Furthermore, individual organisations cannot control their business partners' authorisation policies. Guaranteeing the message accessed and processed by the qualified roles in other collaborating business partners is challenging.

Therefore, model and techniques are required to verify business collaboration reliability in terms of authorisation policies. Petri Net provides a set of verification mechanisms, and its graphically and mathematically founded modelling formalism with various algorithms for design and analysis [73] makes it a good candidate for modelling access control in business collaboration. In this chapter, we propose a Role-Net model which is developed based on Hierarchical Colored Petri Nets (HCPNs). Role-Net provides a verification mechanism to detect the authorisation policies conflicts within or across organisations. A property named Role Authorisation Based Dead Marking Freeness is also defined based on Role-Net to verify business collaboration reliability.

The rest of paper is organized as follows. In section 7.2, we introduce the specification of Role-Net including net structure and execution policy. The reliability property based on Role-Net is presented in section 7.3, followed by introducing the conflict detection methodology of Role-Net.

7.2 CONCEPTUAL RBAC MODEL FOR COLLABORATION RELIABILITY

In this section, a novel Collaborative RBAC conceptual model (C-RBAC) is introduced firstly. C-RBAC mainly focuses on how role based access control is implemented under the environment of loosely-coupled business collaboration. We also provide a formal syntax of role-to-role authorisation constraints in C-BAC to restrict the generation of the required role within and across organisations.

7.2.1 Specification of Conceptual RBAC Model

A family of conceptual models is constructed to understand the various dimensions of RBAC in [5]:

$RBAC_0$ ：A basic RBAC model conceptual model including basic elements, such as User, Role and Permission.

$RBAC_1$: Based on $RBAC_0$, with Role Hierarchy definition added.

$RBAC_2$: Constraints on different elements in the conceptual model $RBAC_0$ are embedded. For example, a user can not be assigned to conflicting roles at design time, which is known as Static Separation of Duties. A user can be assigned to different roles but cannot be active simultaneously at run-time, which is known as Dynamic Separation of Duties. Cardinality constraints are another addition in $RBAC_2$. It defines the maximum number of users can be assigned to one role.

$RBAC_3$: Group $RBAC_1$ and $RBAC_2$ together.

In Fig. 2, we introduce the $RBAC_3$ in the family of RBAC conceptual models which includes features of all other RBAC conceptual models. In Fig. 2, User element is used to represent a human being who belongs to an organisation. A Role element describes a named job function within the business process context that regards the authority and responsibility. There exists a hierarchical architecture for the member of Role element. Permission is an approval of actions granted to specific roles (also known as privilege). A Constraint regulates the relations between different elements.

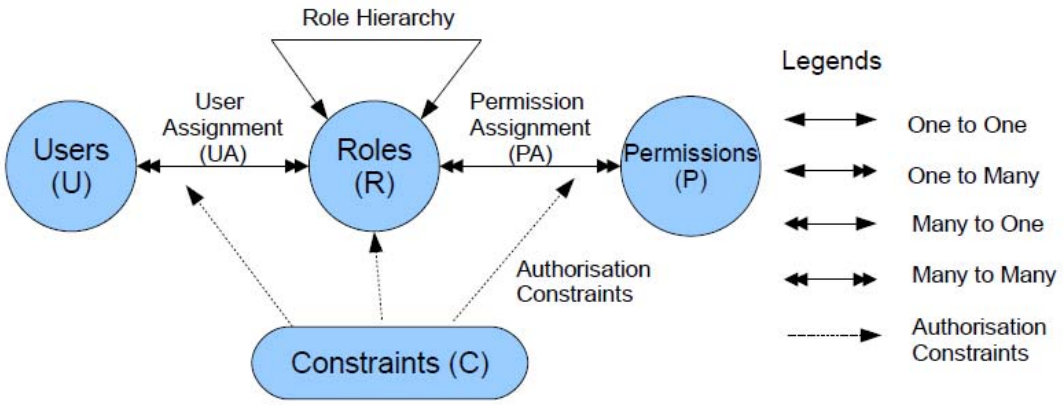

Figure 7-1 Traditional RBAC Model

Many research work have been done in RBAC area. But most of them focused on the enhancement on constraints and permissions. In the client/server and middleware computing environment, the separation of users and roles are suffice [21]. However, in a business collaboration environment, a specific role is assigned to each local service within one organisation to a process message or is required to be allocated to deal with message at service in collaborators. Therefore, in order to address the above issues, we extend the traditional RBAC conceptual model to Collaborative RBAC.

The C-RBAC conceptual model includes sets of several basic elements:

For local organisation: Local Permission (LP), Local Role (LR), Message (M), Local Service (LS).

For collaborators: Collaborative Permission (CP), Collaborative Role (CR), Message (M), and Collaborative Service (CS).

In Figure 5-2, a Local Role (LR) describes a named job function in local organisation regarding the authority and responsibility; (It can be an access role to current service or a required role by next service.) While Local Permission (LP) is an approval of actions granted to specific Local Roles (LR) (also known as privilege) within local organisation. A Message (M) is an object to be accessed and modified by Local Service (LS) which is assigned as specific Local Role (LR). For example, *risk assessment service* in **Bank** can be assigned as **RISK ADMINISTRATOR** role to access huge amount loan application; While it can also be identified as risk officer role to handle small amount loan application. A Collaborative Role (CR) in collaborators is represented as a required role only demanded by local organisation to deal with the Message (M) transferred across organisational boundary. Collaborative Service (CS) can access the Message (M) by assigning Collaborative Role (CR). For instance, the **CREDIT CHECK MANAGER** role in **Credit Rating Agency** is required to be assigned to *Credit Check Service* to deal with Credit Check Requirement message by **Bank** to protect client's privacy.

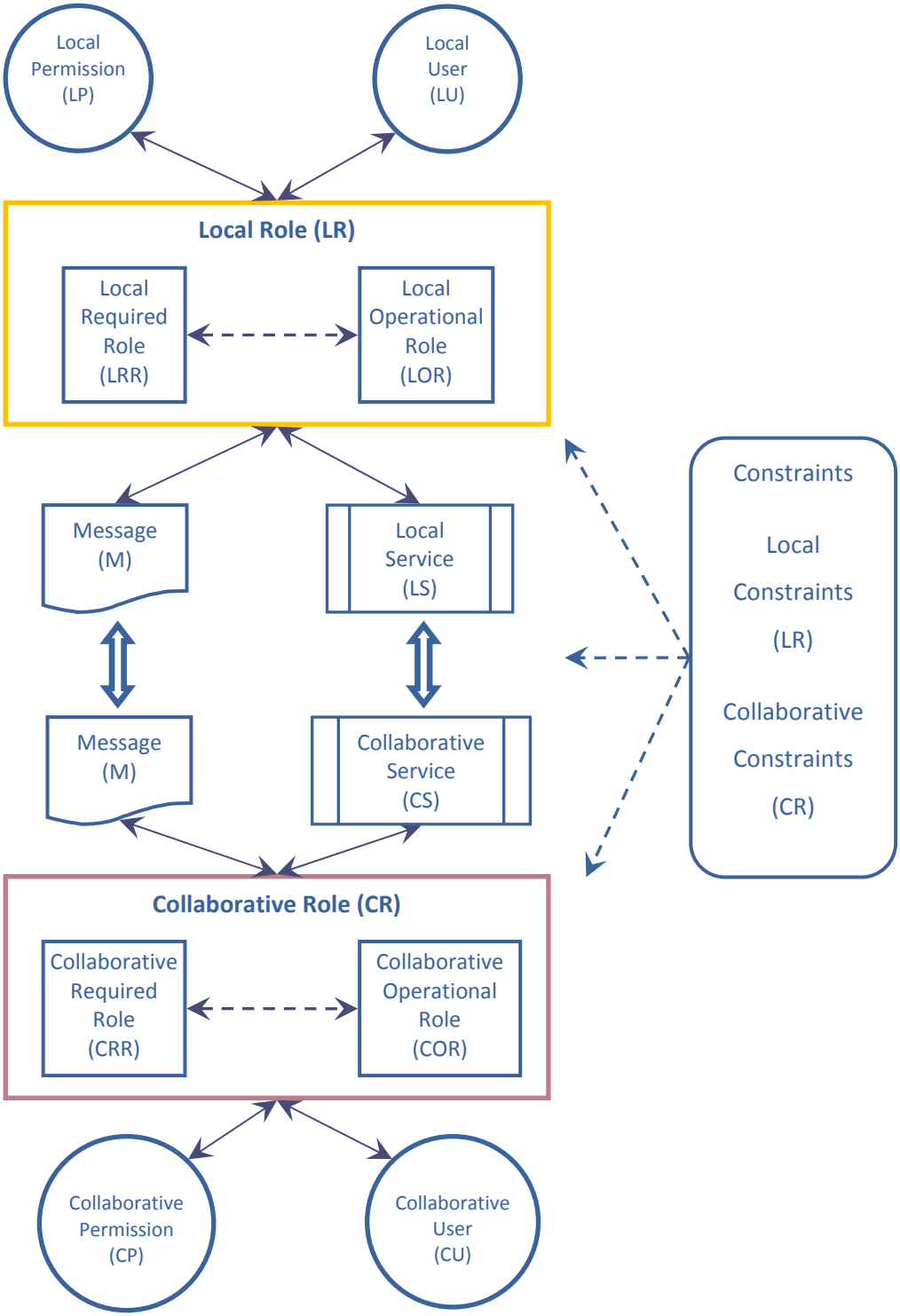

Figure 7-2 Collaborative RBAC Conceptual Model

The concept of the role relation is central to be collaborative RBAC model. There are three Local Role (LR) relations and three Collaborative Role (CR) relations where all of them are many-to-many relations:

For local organisation:

$$Local\ Permission \Leftrightarrow Local\ Role\left(\frac{PLA}{RALP}\right), Message$$
$$\Leftrightarrow Local\ Role\left(\frac{MLA}{LRMA}\right), and\ Service \Leftrightarrow Local\ Role\left(\frac{SLA}{LRAS}\right)$$

For collaborator:

$$Collaborative\ Permission \Leftrightarrow Local\ Role\left(\frac{PCA}{RACP}\right), Message$$
$$\Leftrightarrow Collaborative\ Role\left(\frac{MCA}{CRMA}\right), and\ Collaborative\ Service$$
$$\Leftrightarrow Local\ Role\left(\frac{SLA}{LRAS}\right)$$

Element relations from local organisation point of view are governed by Local Constraints (LC). In this section, we only focus on describing the three types of role-to-role authorisation constraints on the generation of the required role within or across organisations. Here we summarize the formal definitions of relevant basic elements and role relations used in C-RBAC conceptual model. Based on such formal definition, the syntax of role-to-role authorisation constraints will be introduced.

Definition 7-7-1 Basic Elements:

$\boldsymbol{R}$ is the set of roles which has two subsets $\boldsymbol{LR}$ and $\boldsymbol{CR}$,

$\boldsymbol{LR} \subset \boldsymbol{R}, \boldsymbol{CR} \subset \boldsymbol{R}, \boldsymbol{LR} \cap \boldsymbol{CR} = \emptyset$, where $\boldsymbol{LR}$ represents role needed by service in local organisation,and $\boldsymbol{CR}$ indicates the role required by the local

organisation to access message from the collaborator's service. $\forall r_i^l \in \boldsymbol{LR}$, $\forall r_i^c \in \boldsymbol{CR}$;

$\boldsymbol{P}$ is the set of roles which has two subsets $\boldsymbol{LP}$ and $\boldsymbol{CP}$,

$\boldsymbol{LP} \subset \boldsymbol{P}, \boldsymbol{CP} \subset \boldsymbol{P}, \boldsymbol{LP} \cap \boldsymbol{CP} = \emptyset$, where $\boldsymbol{LP}$ represents role needed by service in local organisation,and $\boldsymbol{CP}$ indicates the role required by local organisation to access message at service in collaborators. $\forall p_i^l \in \boldsymbol{LP}$, $\forall p_i^c \in \boldsymbol{CP}$;

$\boldsymbol{S}$ is the set of roles which has two subsets $\boldsymbol{LS}$ and $\boldsymbol{CS}$,

$\boldsymbol{LS} \subset \boldsymbol{S}, \boldsymbol{CS} \subset \boldsymbol{S}, \boldsymbol{LS} \cap \boldsymbol{CS} = \emptyset$, where $\boldsymbol{LS}$ represents role needed by service in local organisation,and $\boldsymbol{CS}$ indicates the role required by local organisation to access message at service in collaborators. $\forall s_i^l \in \boldsymbol{LS}$, $\forall s_i^c \in \boldsymbol{CS}$;

$\boldsymbol{M}$ is the set of Message, $\forall m_i \in \boldsymbol{M}$.

Definition 7-7-2 Basic Assignment:

$\boldsymbol{PLA}$ is the set of local permission to local role assignment, and $\boldsymbol{PCA}$ is the set of collaboration permission to collaboration role assignment.

$\exists r_j^l \in \boldsymbol{LR}, \exists p_k^l \in \boldsymbol{LP}, \boldsymbol{PLA}\left(p_k^l\right) \rightarrow r_j^l$;

$\exists r_j^c \in \boldsymbol{CR}, \exists p_k^c \in \boldsymbol{M}, \boldsymbol{MCA}\left(p_k^c\right) \rightarrow r_j^c$;

$\boldsymbol{RALP}$ is the set of local role to local permission, and $\boldsymbol{RACP}$ represents the set of collaborative role to collaborative permission assignment.

$\exists r_j^l \in \boldsymbol{LR}, \exists p_k^l \in \boldsymbol{LP}, \boldsymbol{RALP}\left(r_j^l\right) \rightarrow p_k^l$;

$\exists r_j^c \in \boldsymbol{CR}, \exists p_k^c \in \boldsymbol{M}, \boldsymbol{RACP}\left(r_j^c\right) \rightarrow p_k^c$;

$\boldsymbol{MLA}$ is the set of message to local role assignment, and $\boldsymbol{MCA}$ is the set of message to collaborative role assignment.

$\exists r_j^l \in \boldsymbol{LR}, \exists m_k \in \boldsymbol{M}, \boldsymbol{MLA}\left(m_k\right) \rightarrow r_j^l$;

$\exists r_j^c \in \boldsymbol{CR},\ \exists m_k \in \boldsymbol{M}, \boldsymbol{MCA}\ (m_k) \rightarrow r_j^c$;

$\boldsymbol{LRAM}$ is the set of local role to message assignment, and $\boldsymbol{CRAM}$ is the set of collaborative role to message assignment.

$\exists r_j^l \in \boldsymbol{LR},\ \exists m_k \in \boldsymbol{M}, \boldsymbol{LRAM}\ (r_j^l) \rightarrow m_k$;

$\exists r_j^c \in \boldsymbol{CR},\ \exists m_k \in \boldsymbol{M}, \boldsymbol{CRAM}\ (r_j^c) \rightarrow m_k$;

$\boldsymbol{SLA}$ is the set of local service to local role assignment, and $\boldsymbol{SCA}$ is the set of collaborative service to collaborative role assignment.

$\exists r_j^l \in \boldsymbol{LR},\ \exists s_q^l \in \boldsymbol{LS}, \boldsymbol{SLA}\ (s_q^l) \rightarrow r_j^l$;

$\exists r_j^c \in \boldsymbol{CR},\ \exists s_q^c \in \boldsymbol{CS}, \boldsymbol{SCA}\ (s_q^c) \rightarrow r_j^c$;

$\boldsymbol{LRAS}$ is the set of local role to local service assignment, and $\boldsymbol{CRAS}$ is the set of collaborative role to collaborative service assignment.

$\exists r_j^l \in \boldsymbol{LR},\ \exists s_q^l \in \boldsymbol{LS}, \boldsymbol{LRAS}\ (r_j^l) \rightarrow s_q^l$;

$\exists r_j^c \in \boldsymbol{CR},\ \exists s_q^c \in \boldsymbol{CS}, \boldsymbol{CRAS}\ (r_j^c) \rightarrow s_q^c$;

7.2.2 Role-to-Role Authorisation Constraints in C-RBAC

Constraint imposes restrictions and access configuration into access control, and become an important component in RBAC2. Traditionally, constraints are managed in a centralized manner. The security policies which implement the constraints are defined within the single organisation. Consequently, they are isolated from cooperating partners' counterpart. In the context of service computing, however, we need to extend the original constraints in business collaboration environment for two reasons. First, the constraints need to be extended to accommodate decentralized architecture which is the foundation requirement in business collaboration. Second, we need to impose new mechanisms onto local services and collaborative services to reflex the

improvement in the service element. Here we present the three types of role-to-role authorisation constraints.

For Local Organisation in Business Collaboration Role hierarchy is used to map the organisational structure. The senior role is automatically authorized with the privileges from its junior roles. For example, in the bank loan scenario, the Bank Manager (Director Level Role) is on top of the role hierarchy. The Bank Manager governs the Loan Manager and the Risk Administrator (Manager Level Roles). Therefore, the Bank Manager is automatically assigned the permissions of the Loan Manager and the Risk Administrator. However, if the permissions of Bank Manager do not contain all of the permissions of the Loan Manager and the Risk Administrator, the Role Hierarchy is violated. Here we present the syntax of Role Hierarchy.

Rule 7–1 Role Hierarchy

$$\exists\, r_j^l, r_{j-1}^l \in \mathbf{LR}, \mathbb{S}(r_j^l) \longrightarrow r_{j-1}^l \; where \; r_{j-1}^l \; is \; the \; senior \; role \; of \; r_j^l \;;$$

$$\exists\, \mathrm{m_k} \in \boldsymbol{M}, \mathrm{s_q^l} \in \boldsymbol{LS}, \forall\, \mathrm{r_{j-1}^l} \in \mathbb{S}(\mathrm{r_j^l}) \Leftrightarrow \left(\boldsymbol{RALP}(\mathrm{r_{j-1}^l}) \supseteq \boldsymbol{RALP}(\mathrm{r_j^l})\right) \cap \left(\boldsymbol{LRAM}(\mathrm{r_{j-1}^l}) \supseteq \boldsymbol{LRAM}(\mathrm{r_j^l})\right) \cap \left(\boldsymbol{LRAS}(\mathrm{r_{j-1}^l}) \supseteq \boldsymbol{LRAS}(\mathrm{r_j^l})\right)$$

Role dependency defines the dependency relationship among different roles. For instance, the role who receives loan application from the Customer at Bank side must be the role who provides loan response to the Customer. Hence, the rule is violated if relevant required dependency roles are absent. We define the Role Dependency as follows:

Rule 7–2 Role Dependency

$$\exists s_a^l, s_b^l \in \boldsymbol{LS}, \exists m_x, m_y \in \boldsymbol{M}, \exists r_i^l, r_j^l \in \boldsymbol{LR}, \exists p_k^l, p_u^l \in \boldsymbol{LP},$$

$$i \neq j, x \neq y, and \; a \neq b$$

$$r_i^l \in \left(\boldsymbol{MLA}(m_x) \cap \boldsymbol{SLA}(s_a^l) \cap \boldsymbol{PLA}(p_k^l)\right) \Rightarrow$$

$$r_i^l \in \left(\boldsymbol{MLA}(m_y) \cap \boldsymbol{SLA}(s_b^l) \cap \boldsymbol{PLA}(p_u^l)\right)$$

Role conflict, on the contrary to Role Dependency, explicitly presents the conflict of roles-to-role authorisation constraints. For example, at Bank side, no lower level role can be assigned the permission to access message if the message has been processed by upper level role, e.g., role in general officer level (Loan Officer) cannot process loan application which has been dealt by a manage level role (Loan Manager).

Rule 7–3 Role Conflict

$$\exists s_a^l, s_b^l \in \boldsymbol{LS}, \exists m_x, m_y \in \boldsymbol{M}, \exists r_i^l, r_j^l \in \boldsymbol{LR}, \exists p_k^l, p_u^l \in \boldsymbol{LP},$$

$$i \neq j\ , x \neq y, and\ a \neq b$$

$$r_i^l \in \left(\boldsymbol{MLA}(m_x) \cap \boldsymbol{SLA}(s_a^l) \cap \boldsymbol{PLA}(p_k^l)\right) \Rightarrow$$

$$\dashv r_i^l \in \left(\boldsymbol{MLA}(m_y) \cap \boldsymbol{SLA}(s_b^l) \cap \boldsymbol{PLA}(p_u^l)\right)$$

For Collaborators in Business Collaboration since business collaboration is peer based and services are autonomous, authorisation policies defined for individual organisations normally cannot be seen by others. Therefore, in order to guarantee that the messages transferred among organisations can be accessed by the qualified roles in business collaboration, each organisation need to send its collaborators the required role information together with messages to be accessed at collaborators' service. Based on such assumption, we present the formal syntax of role-to-role authorisation constraints for collaborators.

The Role hierarchy is used to map the organisational structure with the role hierarchy. Hence, Role hierarchy in collaborators cannot be seen by the local organisation due to protecting internal organisational information.

Role dependency defines the dependency relationship among different roles. For instance, due to privacy of client, if loan manager in Bank is assigned to access loan case at loan application service, then credit check manager in Credit Rating Agency is required to deal with the loan at Credit Check service. We define the Role Dependency as follows:

Rule 7–4 Role Dependency

$$\exists s_a^l \in \boldsymbol{LS}, s_b^c \in \boldsymbol{CS}, \exists m_x, m_y \in \boldsymbol{M}, \exists r_i^l \in \boldsymbol{LR}, r_j^c \in \boldsymbol{CR}, \exists p_k^l \in \boldsymbol{LP}, , p_u^c \in \boldsymbol{CP}$$

$$i \neq j\ , x \neq y, and\ a \neq b$$

$$r_i^l \in \left(\boldsymbol{MLA}(m_x) \cap \boldsymbol{SLA}(s_a^l) \cap \boldsymbol{PLA}(p_k^l)\right) \Rightarrow$$

$$r_i^l \in \left(\boldsymbol{MCA}(m_y) \cap \boldsymbol{SCA}(s_b^c) \cap \boldsymbol{PCA}(p_u^c)\right)$$

Role conflict constraints for collaborators explicitly define the conflict relationship of Local Role (LR) and collaborative role (CR) in business collaboration. For example, in order to guarantee the privacy of client, if loan manager in **Bank** is assigned to access loan case at loan application service, then credit check officer in **Credit Rating Agency** must not deal with the loan at Credit Check service which is required by Bank. We define the Role Conflict as follows:

Rule 7–5 Role Conflict

$$\exists s_a^l \in \boldsymbol{LS}, s_b^c \in \boldsymbol{CS}, \exists m_x, m_y \in \boldsymbol{M}, \exists r_i^l \in \boldsymbol{LR}, r_j^c \in \boldsymbol{CR}, \exists p_k^l \in \boldsymbol{LP}, , p_u^c \in \boldsymbol{CP}$$

$$i \neq j\ , x \neq y, and\ a \neq b$$

$$r_i^l \in \left(\boldsymbol{MLA}(m_x) \cap \boldsymbol{SLA}(s_a^l) \cap \boldsymbol{PLA}(p_k^l)\right) \Rightarrow$$

$$\dashv r_i^l \in \left(\boldsymbol{MCA}(m_y) \cap \boldsymbol{SCA}(s_b^c) \cap \boldsymbol{PCA}(p_u^c)\right)$$

7.3 SPECIFICATIONS OF THE ROLE AUTHORISATION MODEL ROLE-NET

Role-Net is a role-authorisation oriented, Petri-net based model to simulate business collaboration for each participating organisation. It provides a theoretical infrastructure to manage and verify business collaboration reliability in terms of authorisation policies. Role-Net is designed by following the principles:

- A Role-Net is separated into two layers, which correspond to the local organisation and its collaborators respectively. A Refinement Function is used to link them. Refinement Function is originated from Hierarchical Petri Net and used to connect each layer in the stratified Petri-net based model.
- A Role is modelled as a RO-Token in Role-Net. Its movement among consecutive transitions thereby models the role assignment at specific services, which consequently generates a role flow. However, before a *Place*, the RO-Token is called **Operational Role** which represents the role who accesses and modifies the message at previous service; while after a *place* the RO-Token represents **Required Role** which is used to describe the roles required in the next service. (The detailed Petri net terminology such as *Token*, *Place* and *Transition* will be explained in section 3.1).
- There are two types of tokens moved within Role-Net: AO-Token and RO-Token, each of which correspond to application message and role. Role-Net separates the role flow from message flow to realize the authorisation control instead of hard coding the role authorisation specification within the process. These two types of tokens are dynamically combined and moved together during the execution of Role-Net. The correlation of the two types of tokens can guarantee that the desired message can only be accessed by the specific roles at the designated service.

7.3.1 Structure of Role-Net

Place and *transition* are two key components in a Petri Net. They are linked by the *flow relations*. If the places before the transition have accumulated enough tokens, then the transition will be enabled. If the places after the transition are available, then the transition will be fired to pass the tokens to the next place. In a Role-Net, we use places to model the state of business collaboration and provide function to transfer Role-Token from representing Operational role to indicating Required role according to the organisational authorisation policies. Transitions are used to model services and implement corresponding functions. For example, the first transition on the **Bank** side in Figure 7-3 represents the Loan application service. The place before the service is used to model the state of the service and select the required roles to access message in this service. The *flow relations* in a Role-Net represent the execution order of services in business collaboration.

A Role-Net, as mentioned above, is separated into two layers to model the inter-organisational role authorisation in business collaboration. The upper layer is used to model the business process in local organisation and the lower layer is intended to simulate the projection of local organisation's view on its collaborators' process. The details of these two layers are as follows:

- The upper layer of a Role-Net is used to describe the role-based authorisation policy within local organisation only. For example, the upper layer in **Bank's** Role-Net in Figure 7-3 merely models the authorisation policy of Loan Application process on the **Bank** side.
- The lower layer of one organisation's Role-Net models the authorisation policy of the services of the collaborators' with which the organisation is interacting. In other words, if the service in local organisation requires business interaction with its collaborators, the local organisation's projection on collaborators' Role-Net will be modelled as the lower layer

of local organisation's Role-Net. For instance, the part of Role-Net in **Credit Rating Agency** of Figure 7-3 is in the lower layer of **Bank's** Role-Net.

In Figure 7-3, we give an example of Role-Net of **Bank** which is separated into two layers. The upper layer models loan application process at **Bank** side while lower layer simulates the projection of **Bank** on **Credit Rating Agency**'s Role-Net. They are linked by Refinement Function at the transition which is represented as *Credit Check Requirement Service. a, b, c, d, e* in the figure are AO-Tokens which indicates the messages transferred in the business collaboration. r_1 … r_5 and r_0^{ε} … r_3^{ε} including R" 1 are RO-Tokens while r_1 … r_5 represent operational role for each service and the rest for required role. (The execution policies of Role-Net, i.e., how the AO-Token and RO-Token move in the Role-Net, will be presented in the following section.) We present the formal definition of Role-Net's two layers based on Hierarchical Colored Petri Nets as follows:

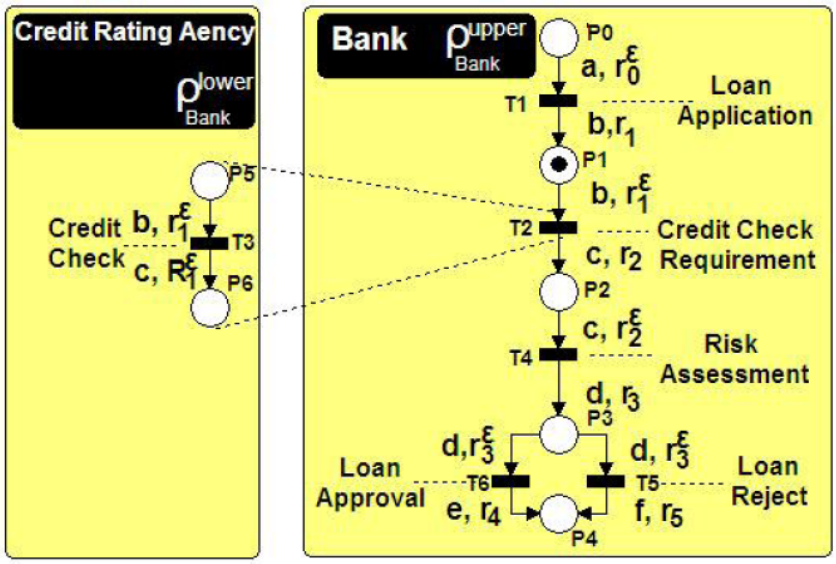

Figure 7-3 Role Net of Bank

Definition 7-7-3

The upper layer of organisation G_i 's Role-Net is a tuple $\rho_{G_i}^{upper}$ = $\left(P_{G_i}^{upper}, T_{G_i}^{upper}, F_{G_i}^{upper}, \Gamma_{G_i}^{upper}, \Delta_{G_i}^{upper}, \Theta_{G_i}^{upper}, \Omega_{G_i}^{upper}\right)$, where

$P_{G_i}^{upper}$ is a set of places in upper layer of G_i's Role-Net which graphically are represented as circle in the above figure.

$T_{G_i}^{upper}$ is a set of transitions graphically represented as dark bar in upper layer of Role-Net in the above figure. $P_{G_i}^{upper} \cap T_{G_i}^{upper} = \emptyset$

$F_{G_i}^{upper} = (p^u \times V \times t^u) \cup (t^u \times V \times p^u)$ is the flow relation between places and transitions, where $p^u \in P_{G_i}^{upper}$, $t^u \in T_{G_i}^{upper}$, and V is the sets of variables $V = \{x, y, \dots\}$ to represent the tokens.

$\Gamma_{G_i}^{upper}(p^u, a, r) \rightarrow Boolean$ is a correlation function to evaluate the relationship of a RO-Token and an AO-Token at specific place, where $a \in$ AO-Token, r ∈ RO-Token, and $p^u \in P_{G_i}^{upper}$. $\Gamma_{G_i}^{upper}$ guarantees that the AO-Token can only be moved with assigned RO-Tokens at specific places.

$\Delta_{G_i}^{upper}(p^u, a, r) \rightarrow r^{\varepsilon}$ is a function to change a RO-Token from representing role (operational role r) that accessed the AO-Token at previous transition to indicating the roles (required roles r^{ε}) which are needed by the next transition, according to role authorisation policies, where $p^u \in P_{G_i}^{upper}$, $a \in$ AO-Token, $r, r^{\varepsilon} \in$ RO-Token.

$\Theta_{G_i}^{upper}(t^u, \varphi, r^{\varepsilon}) \rightarrow Boolean$ is a comparison function, where $t^u \in T_{G_i}^{upper}$, $r^{\varepsilon} \in$ RO-Token, and φ is a threshold variable representing the role element selected from the set γ. γ is the set of roles that are permitted to access and modify the AO-Token in the transition in the upper layer of G_i's Role-Net, named as available role set. The $TRUE$ result of $\Theta_{G_i}^{upper}$ function reflects the existence of qualified roles for specific transition t.

$\Omega_{G_i}^{upper}(t_\beta^u, a) \rightarrow L$ is refinement function on transition t_β^u to connect the Role-Net's lower layer, where $t_\beta^u \in T_{G_i}^{upper}$ represents transition including the link between two layers of Role-Net. $L = \left\{g(x), \{e(x), \rho_{G_i}^{lower}, r(x)\}\right\}, x \in V$. $g(x)$ is a function to evaluate the token that is an input of a transition and decides which $\rho_{G_i}^{lower}$ shall be initiated in other collaborations, $e(x)$ and $r(x)$ are the guard functions of corresponding subnet to evaluate whether or not the $\rho_{G_i}^{lower}$ is available to initiate and exit.

Hence, function $\Delta_{G_i}^{upper}$ can transfer a RO-Token from representing *Operational Role* r to indicating *Required Role* r^{ε} according to authorisation policies. Role

authorisation policy conflicts caused by improvisational variation of role authorisation or unqualified role authorisation in peer collaborators can then be detected if the role that actually accesses and modifies the message at the next service is not in the required role set generated by $\Delta_{G_i}^{upper}$. $\Delta_{G_i}^{upper}$ can be defined by individual organisations according to their authorisation policy rules, which is a topic out of scope of this research.

Definition 7-7-4

The lower layer of organisation G_i's Role-Net is given by the tuple $\rho_{G_i}^{lower}$= $\left(P_{G_i}^{lower}, T_{G_i}^{lower}, F_{G_i}^{lower}, \Gamma_{G_i}^{lower}, \Psi_{G_i}^{lower}\right)$, where

$P_{G_i}^{lower}$ is a set of places in lower layer of G_i's Role-Net which graphically are represented as circle in above figure.

$T_{G_i}^{lower}$ is a set of transitions graphically represented as dark bars in lower layer of Role-Net in above figure. $P_{G_i}^{lower} \cap T_{G_i}^{lower} = \emptyset$

$F_{G_i}^{lower} = (p^u \times V \times t^u) \cup (t^u \times V \times p^u)$ is the flow relation between places and transitions, where $p^u \in P_{G_i}^{lower}$, $t^u \in T_{G_i}^{lower}$, and V is the sets of variables $V = \{x, y, \dots\}$ to represent the tokens.

$\Gamma_{G_i}^{lower}(p^u, a, r) \rightarrow Boolean$ is a correlation function to evaluate the relationship of RO-Token and AO-Token at specific place, where $a \in$ AO-Token, r ∈ RO-Token, and $p^u \in P_{G_i}^{lower}$. $\Gamma_{G_i}^{lower}$ guarantees that the AO-Token can only be moved with assigned RO-Tokens at specific places.

$\Psi_{G_i}^{lower}(t^l, a, r^\varepsilon) \rightarrow (b, R^\varepsilon)$ is a switch function to transfer the value of AO-Token and RO-Token from the input of the transition in lower layer to the output of the transition, where $t^l \in T_{G_i}^{lower}$ $a, b \in$ AO-Token $(a \neq b), r^\varepsilon, R^\varepsilon \in$ RO-Token (r^ε is the required role transferred from Role-Net's upper layer to lower layer. R^ε is the operational role in lower layer and is returned from Role-Net's lower layer to upper layer. When R^ε arrives at the upper layer, it is an input to $\Theta_{G_i}^{upper}$ to detect the qualified role set). Any modification on AO- Token and RO-Token is unknown to organization G_i since it only observes the behaviour of its

collaborators through this lower layer of the net. Hence, the switch function is only used to transfer the value of AO-Token and RO- Token after they have been modified according to collaborator's authorisation polices. Details of AO-Token and RO-Token's execution are explained in the following section.

7.3.2 Execution of Role-Net

Execution Policy of role-net as mentioned above, there are two types of tokens that are operated within a role-net: the Application-Oriented Token (AO-Token) and the Role-Oriented Token (RO-Token) whose movements correspond to the *message flow* and *role flow*. The *message flow* describes the information transferred within and among organisations to facilitate the business collaboration, e.g., customer credit level is a message transferred between the *Credit Check Require* service in **Bank** and *Credit Check service* in **Credit Rating Agency**. The *role flow* is used to describe the sequence of role authorisation on each service in business collaboration. The AO-Token will move together with the relevant RO-Token to correlate the message flow and role flow. The execution policies of Role-Net are described as follows:

Token at Place

- Each RO-token is correlated to a specific AO-token. The *Correlation function* G_i in upper layer and lower layer of Role-Net will check the correlation of these two types of tokens at each place. If a RO-Token and an AO-Token are received separately, the Place will abandon the token as an unexpected role or message respectively.
- Before *Places* in upper layer, the RO-Token r represents the *Operational role* that has accessed the correlated message at previous *Transition*. After *Places* in upper layer, the RO-Token r^{ε} will represent the *Required role* which will be required by the next *transition*. The Function $\Delta_{G_i}^{upper}$ will deal with the transfer of RO-Token at each *Place* in upper layer.

- The place in lower layer is used to receive the AO-Token and RO-Token from upper layer, and return the two correlated tokens together to upper layer after they are processed by the services of the collaborating partners.

Token at Transition

Transition in upper layer of Organisation G_i's Role-Net

If the transition happens between upper layer and lower layer token movement, the AO-Token and RO-Token r^{ε} representing Required Roles will move together to the lower layer of Role-Net as cross-organisational message transfer. The refinement $\Omega_{G_i}^{upper}$ is used to identify the lower layer of organisation G_i's Role-net $\rho_{G_i}^{lower}$ (the lower layer of local organisation's Role-Net represents the local organisation's view on its collaborator's Role-Net). When the modified AO-Token and RO-Token return from the lower layer, the transition in upper layer then invokes the *Comparison Function* $\Theta_{G_i}^{upper}$ to identify the qualified roles.

$\Theta_{G_i}^{upper}$ function is implemented to detect the qualified roles when Required Role r^{ε} arrives at transition in upper layer with AO-Token (no link between lower layer and upper layer in this transition) or returned RO-Token R^{ε} arrives at the transition with AO-Token from lower layer (link between lower layer and upper layer exists in this transition).

Each transition in upper layer of Organisation G_i's Role-Net has a set of available roles γ which are qualified to access the message in this transition. However, depending on the properties of message and role authorisation policies, all or part of them may not be authorised to process message at runtime. Therefore, a threshold φ is dynamically decided by choosing roles from γ at each transition. (If the transition in the upper layer of the Role-Net has link to the lower layer, then φ is selected from R^{ε} and is input in function $\Theta_{G_i}^{upper}$ to verify

whether the AO-Token is modified by the Required Role r^{ε} in the collaborator's Role-Net).

If r^{ε}'s element is equal to the threshold φ, the role in threshold will be moved to the set ϱ as qualified role to access the message in this transition and the threshold will be degraded for the next role in γ. The comparison will continue until all role elements in Required Roles r^{ε} and (or R^{ε}) have been dealt with.

Finally, if ϱ is not empty, then the role elements in this set will be authorised the permission to access the messages in this service. The RO-Token will thus represent the role that actually accesses the message and is moved with AO-Token together to the next places. If ϱ is empty, then there is no qualified role to deal with this message at this service. The process will be suspended due to the role authorisation runtime error. Therefore, by comparing φ with each role element in Required Roles r^{ε}, the qualified role will be selected.

Transition in lower layer of organisation G_i's Role-Net

$\Psi_{G_i}^{lower}$ in the transition of lower layer of organisation G_i's Role-Net is used to transfer the value of AO-Token and RO-Token from the input of the transition to the output of the transition. However, the switch function $\Psi_{G_i}^{lower}$ cannot identify how the value of AO-Token and RO-Token are changed in the transition. It means that the local organisation G_i is agnostic to its collaborator's internal process, including how to deal with the message AO-Token and which role is assigned to process the message. These modifications on AO-Token and RO-Token are implemented according to collaborator's own authorisation policies, and $\Psi_{G_i}^{lower}$ in the lower layer of local organisation G_i's Role-Net can only identify and exchange the result of modification on tokens.

Algorithm 7-1 describes the Role-Net execution in detail. Note, the threshold φ will be assigned a value of RO-Token returned from the subnet if the message is processed by the service in other organisations. φ will be selected from the

available role set γ for each transition in Role-Net if the message is processed by the service within the local organisation.

As example of Role-Net's Execution In this section, we will present a running example of Bank's Role-Net execution shown in Fig. 4. We only focus on discussing how the RO-Tokens and AO-Tokens are moved from place P_1 to P_3. In Fig. 4, the AO-Token b and RO-Token r_1 just leave the T_1 *Loan application service* as the loan application has been received and will require the *Credit Check Requirement service*. We assume that the AO-Token b is accessed in T_1 by **LOAN MANAGER** according to the authorisation policy that loan application with huge amount should only be processed by a **LOAN MANAGER**.

Algorithm 7-1 Role-Net Execution

Input: $\rho_{G_i}^{upper}, \rho_{G_i}^{lower}, x \in$ AO-Token, $r \in$ RO-Token, where x and r in Initial Place;

Output: $\rho_{G_i}^{upper}, \rho_{G_i}^{lower}, x \in$ AO-Token, $r \in$ RO-Token, where x and r in Final Place or Exception;

```
procedure RoleNet Execution
repeat
    //Check the correlation of two types of token
    if Γ_{G_i}(p_{G_i}^{u/l}, x, r) = TRUE then
        //Change Operational Role r to Requires Role r^ε
        r^ε := Δ_{G_i}^{upper}(p_{G_i}^u, x, r);
        //Move the two tokens to next transition
        Transfer(x, r^ε, p_{G_i}^u•);
        ϱ := ∅;
        //Check if lower layer is included in p_{G_i}^u•
        if p_{G_i}^u• = t_β^u then
            //Identify ρ_{G_i}^{lower}, L= {g(x), {e(x), ρ_{G_i}^{lower}, a(x)}}
```

L:= $\Omega_{G_i}^{upper}(t_\beta^u, x)$

if $g(x)$ = TRUE and $e(x)$ = TRUE then

//Move tokens to collaborator's Role-Net

Transfer(x, r^ε, $\rho_{G_i}^{lower}$)

//Waiting for the token process in other organisation

//Obtain the output of $t_{G_i}^l$ from Collaborator's Role-Net

$\Psi_{G_i}^{lower}(t_{G_i}^l, x, r^\varepsilon) \Rightarrow (b, R^\varepsilon)$;

x:=b;

φ:= R^ε;

//Check the qualification of the rule who accesses

//message at collaborator

if $\Theta_{G_i}^{upper}(t_{G_i}^u, r^\varepsilon, \varphi)$ = TRUE then

ϱ := ϱ + φ;

end if

end if

else

//No lower layer is included in $p_{G_i}^u$•

i := $\|r^\varepsilon\|$;

j := $\|\gamma\|$;

//Find the qualified role by comparing r_i^ε and available

//role γ

while $i \neq 0$ do

while $j \neq 0$ do

if $\Theta_{G_i}^{upper}(t_{G_i}^u, r_i^\varepsilon, \varphi)$ = TRUE then

j := 0;

//Qualified role will be stored in set ϱ

ϱ := ϱ + φ;

else

j := j - 1;

//Find another available role

```
                    φ :=γ_j
                end if
            end while
            //Find another available role
            i := i - 1;
            j := ||γ||;
        end while
    end if
    //When qualified role exists
    if ϱ ≠ 0 then
        //One of qualified role will randomly process AO-Token
        r := Random(ϱ);
        //RO-Token change from Required Role to Operational Role
        x := λ(x,r);
        //Process AO-Token by Operational Role
        //Move AO-Token and RO-Token to next place
        Transfer(x,r,t_Gi^u•);
    end if
until Γ_Gi(p_Gi^u/l, x, r) = FALSE or ϱ = ∅ or (x IN Final Place and r IN
Final Place)
```

The Role-Net algorithm executes until the two types of tokens both arrive at final place, or two types of tokens are not correlated together as exceptions, or no qualified role in ϱ to deal with message in a transition.

Step 1 When r_1 and b arrive at P_1, The *Correlation function* Γ_{Bank}^{upper} will validate the relationship of the two tokens. $\Gamma_{Bank}^{upper}(P_1, r_1, b)$ equals FALSE if only one type of token arrives at P_1. In this case, the individual token will be abandoned as follows:

- AO-Token Only: as unexpected message which was not processed by any role at previous service.
- RO-Token Only: as unexpected role which was not assigned to process any message at previous service.

Step 2 If $\Gamma_{Bank}^{upper}(P_1, r_1, b)$ equals TRUE, then the place will begin to transfer RO-Token from representing *Operational Role* to indicating *Required Role* according to authorisation policies. Since a message communication is required between the two organisations, the required role therefore should be the qualified role that is expected by **Bank** to process message at *Credit Check* service in **Credit Rating Agency**, where r_1^{ε} is identified through the function $\Delta_{Bank}^{upper}(P_1, b, r_1) = r_1^{\varepsilon}$.

Step 3 When the RO-Token r_1^{ε} and the AO-Token b arrive at T_2, these two tokens will be moved to the lower layer of the **Bank** Role-Net which is the **Bank**'s perspective on **Credit Rating Agency**'s Role-Net. The lower layer of **Bank**'s Role-Net can be identified by using *refinement function* $\Omega_{G_i}^{upper}\left(T_{2\,Bank}^{\beta},\ b,\ r_1^{\varepsilon}\right) \rightarrow L$. The AO-Token b and RO-Token r_1^{ε} are modified within lower layer transition T_3 as AO-Token c and RO-Token R_1^{ε}. Switch Function lower $\Psi_{G_i}^{lower}(T_3,\ b, r_1^{\varepsilon}) \rightarrow (c, R^{\varepsilon})$ is used to identify the change of Tokens' value when the message (AO-Token) in **Credit Rating Agency** has been accessed by the specific role. The AO-Token c and the RO-Token R^{ε} representing who actually accessed the message at **Credit Rating Agency** are returned to **Bank** together after the AO-Token is processed at **Credit Rating Agency** by the specific roles. The Threshold φ of T_2 will be assigned the value of RO-Token R^{ε}. The *Comparison Function* $\Theta_{Bank}^{upper}(T_2, r_1^{\varepsilon}, \varphi)$ is used in T_2 to validate whether the operation role R^{ε} in T_3 (from **Credit Rating Agency**) is the qualified role to process the message. If yes, the RO-Token r_1^{ε} will be changed to r_2 and the cross organisational process is executed. The AO-Token c and RO-Token r_2 will then be moved to the next place

P_2. Otherwise, the process is suspended as the role in Credit Rating Agency is not qualified to process the AO-Token b.

Step 4 AO-Token c and RO-Token r_2 is checked by *Correlation Function* at P_2 as same as in step 1. The function $\Delta_{Bank}^{upper}(P_2, c, r_2) = r_2^{\varepsilon}$ is also implemented to transfer RO-Token and generates r_2^{ε} according to the **Bank**'s role authorisation policies. However, the service represented by T_4 is a private service in Bank without requiring the communication with other services in **Bank**'s collaborators. Hence, the r_2^{ε} represents the required roles in Bank to access the messages at next transition.

Step 5 When the RO-Token r_2^{ε} and AO-Token c arrive at T_4, the *Comparison Function* $\Theta_{Bank}^{upper}(T_4, r_2^{\varepsilon}, \varphi)$ will be used to choose the qualified role. The threshold φ of T_4 will be selected from the available roles set γ to be compared with the required roles r_2^{ε}. The qualified role will be stored in set ϱ. If ϱ is not empty after comparison, then the qualified role exists. T_4 will thus process the AO-Token c and change RO-Token from r_2^{ε} to r_3 which represents the role who processed AO-Token in this transition. The new AO-Token d and new RO-Token r_3 will finally move together to the next place P_3.

7.3.3 Implementing Role-to-Role Constraints on intra- and Inter-organisation authorisation policy in Role-Net

Algorithm 7-2 Identifying Required Role within Organisation

Input: $a \in$ AO-Token, $r_i \in$ RO-Token, $p_j^u \in P_{G_i}^{upper}$ where r_i is Operational Role

Output: $r_i^{\varepsilon} \in$ RO-Token, where r_i^{ε} is Required Role;

```
procedure Required Role (r_i^ε)
//Create instances of r^l, where r^l ∈ LR, p_k^l, p_u^l ∈ LP
    set r_j^l, r_o^l, r_a^l, r_b^l, r_c^l = new set(r^l);
    r_j^l := r_i ;
```

```
//Identify dependent role by implementing authorisation policy
  in terms of service, message and permission.
```

$r_o^l \coloneqq SLA(p_j^u \bullet) \cap MLA(a) \cap PLA(p_k^l)$;

```
//Implement Role Dependency constraints
```

$$r_o^l \in \left(MLA(a) \cap SLA(p_j^u \bullet) \cap PLA(p_k^l)\right) \Rightarrow r_a^l \in \left(MLA(a) \cap SLA(p_j^u \bullet) \cap PLA(p_u^l)\right)$$

```
//Implement Role Conflict constraints
```

$$r_o^l \in \left(MLA(a) \cap SLA(p_j^u \bullet) \cap PLA(p_k^l)\right) \Rightarrow \dashv r_b^l \in \left(MLA(a) \cap SLA(p_j^u \bullet) \cap PLA(p_u^l)\right)$$

```
//Select required role according to role authorisation policy
  and constraints
```

$r_c^l \coloneqq r_o^l \cap r_a^l$;

$r_c^l \coloneqq r_c^l - r_b^l$;

```
//Implement the Role Hierarchy constraint
```

$r_c^l \coloneqq \mathbb{S}(r_c^l)$;

```
//Assign required roles to RO-Token
```

$r_i^\varepsilon \coloneqq r_c^l$;

return r_i^ε

In Role-Net, function $\Delta_{G_i}^{upper}$ is used to change the RO-Token from representing the Operational Role to indicating Required Role according to role authorisation policy for accessing message transferred within or cross organisations. The generation of the required role is deducted according to the Role-to-Role authorisation constraints in C-RBAC model. Here below we present how the required role for message transferred within one organisation and across organisation is generated.

For one organisation : When one message is processed by specific service within one organisation, then a set of role as required role will be identified according to the role authorisation policy. Then the two types of role-to-role

authorisation constraints-Role Dependency (Rule. 2) and Role Conflict (Rule. 3) will be implemented on the set of required role. Finally, the Role Hierarchy (Rule. 1) will identify all of required role which can assign specific permission to desired service to access message.

For collaborators : since an organisation cannot identify the role authorisation policy in collaborators, nor see the Role Hierarchy, the organisation can only generate roles that will access the message by collaborative service in collaborators according own Role Dependency (Rule 4) and Role Conflict (Rule 5) constraints. The algorithm for generating the required role for collaborators is presented below:

Algorithm 7-3 Identifying Required Role across Organisation

Input: $a \in$ AO-Token, $r_i \in$ RO-Token, $p_j^u \in P_{G_i}^{upper}$ where r_i is Operational Role

Output: $r_i^{\varepsilon} \in$ RO-Token, where r_i^{ε} is Required Role;

procedure Required Role (r_i^{ε})

//Create instances of r^c, where $r^c \in CR$, $p_k^l \in LP$, $r_j^l \in LR$

set r_p^c, r_q^c, r_o^c = new set(r^c);

r_j^l := r_i ;

//Identify Dependent role

$$r_j^l \in \left(MLA(a) \cap SLA(p_j^u \bullet) \cap PLA(p_k^l)\right) \Rightarrow r_p^c \in MCA(a) \cap SCA\left(\Omega\left(p_j^u \bullet \beta, r_j^l, a\right)\right)$$

//Identify Conflict role

$$r_j^l \in \left(MLA(a) \cap SLA(p_j^u \bullet) \cap PLA(p_k^l)\right) \Rightarrow \dashv r_q^c \in MCA(a) \cap SCA\left(\Omega\left(p_j^u \bullet \beta, r_j^l, a\right)\right)$$

//Generate required role

$r_o^c := r_p^c - r_q^c$;

//Assign required roles

$r_i^{\varepsilon} := r_o^c$;

return r_i^{ε}

7.4 DETECTING AUTHORISATION POLICY CONFLICTS

In this section, we define a reliability property role authorisation based dead marking freeness based on Role-Net to verify authorisation policy based business collaboration reliability. An algebraic approach to detect role based authorisation policy conflicts is also presented in this section.

The labelled transitive matrix L_{BP}^{*} [111] used in Petri-Net expresses the relationship between $\bullet\, t$ and $t\, \bullet$ based on transition t ($\bullet\, t$ is the set of pre-places of a transition t while $t\, \bullet$ represents the set of post-places of a transition t). However, it does not elaborate the role authorisation relationship between $\bullet\, t$ and $t\, \bullet$. We extend the transitive matrix by associating role authorisation impact called *Role-embedded transitive matrix* and use it to verify the *role authorisation based dead marking freeness* property.

We firstly describe the syntax of two types of matrix algebraic operator $\diamond$ and Δ which are used to generate *Role-embedded transitive matrix* and verify the *role authorisation based dead marking freeness* property. The grammar of definition follows BNF-like notation:

$$M ::= M_1 \diamond M_2 | M_1 \Delta M_2$$

Where:

$M_1 \diamond M_2$: Given an $n \times m$ matrix M_1, and $m \times n$ matrix M_2, and an $n \times n$ matrix M_3 where $M_1 = [c_{ij}]$, and $M_2 = [d_{ji}]$, and $M_3 = [e_{ii}]$ $(i = 1 \dots n, j = 1 \dots m)$. Then

$$M_3 = M_1 \diamond M_2 \Rightarrow [e_{ii}] = \bigcup_{j=1}^{m} ([c_{ij}] \cap [d_{ji}])$$

$M_1 \vartriangle M_2$: Given an $n \times m$ matrix M_1, and $m \times n$ matrix M_2, and an $n \times n$ matrix M_3 where $M_1 = [c_{ij}]$, and $M_2 = [d_{ji}]$, and $M_3 = [e_{ii}]$ $(i = 1 \dots n, j = 1 \dots m)$. Then

$$M_3 = M_1 \vartriangle M_2 \Rightarrow [e_{ii}] = \bigcup_{j=1}^{m} ([c_{ij}] \cap [d_{ji}])$$

The proposed algebraic operators guarantee that each result of an operation on matrix is still a matrix to which we can again apply algebraic operators. Here below we formally define how these two operators can be used.

Definition 5 Role embedded transitive matrix

$$L_{BP}^{R_0} = (A^-)^T \diamond Diag(\gamma_1, \gamma_2, \dots . \gamma_n) \diamond A^+ \Rightarrow$$

$$[c_{jg}] = \bigcup_{k=1}^{n} \left(\bigcup_{i=1}^{n} \left(\left[\left(a_{ij}^- \right)^T \right] \cap [b_{ik}] \cap \left[a_{kg}^+ \right] \right) \right)$$

where

γ_h $(h = 1,2, \dots n)$ is the available role set of each transition. $Diag(\gamma_1, \gamma_2, \dots . \gamma_n) = [b_{ik}]$ is $n \times n$ matrix $(i, k = 1,2, \dots . n)$;

$A^- = \left[A_{ij}^-\right]$ and $A^+ = \left[A_{kg}^+\right]$ are $n \times m$ matrix $(i, k = 1,2, \dots . n, j, g = 1,2 \dots m)$. T means transposed matrix. Ξ represents the set indicating all role elements in business collaboration, where n is the number of transitions and m is the number of places.

$$a_{ij}^- = \begin{cases} \Xi\ (x, y) \quad In \quad \rho_{G_i}^{\frac{upper}{lower}} \\ \phi\ (x, y)\ Not\ In\ \rho_{G_i}^{\frac{upper}{lower}} \end{cases}$$

$$a_{ij}^+ = \begin{cases} \Xi\ (y, x) \quad In \quad \rho_{G_i}^{\frac{upper}{lower}} \\ \phi\ (y, x)\ Not\ In\ \rho_{G_i}^{\frac{upper}{lower}} \end{cases}$$

$[c_{jg}]$ is role embedded transitive matrix. $L_{BP}^{R_0} = [c_{jg}]$ is then $m \times m$ matrix.

Marking of Petri-Net based model is an allocation of tokens to the places of the net formally defined as a function $M: P \longrightarrow R^{|P|}$, where $R^{|P|}$ is $|P| \times 1$ vector. The marking reflects the state of the Petri Net after each transition firing. In a marking k, if a token in place p, then $M_k(P) = 1$, otherwise $M_k(P) = 0$. A Petri-Net based model is called *Dead Marking Free* if there do not exist places having no enabled transition. It means that the change of marking from state k to state $k + 1$ is not impeded by the absence of a transition firing during the execution of the model. Hence, we extend the property *Dead Marking Freeness* by embedding role authorisation impact called *Role Authorisation Based Dead Marking Freeness*. This reliability property is used in Role-Net to verify authorisation policy based business collaboration reliability by detecting whether there exist places having no enabled transition (resulting in the absence of a transition firing) caused by the errors of role authorisation at runtime. Here we present the formal definition of this property.

Definition 7-5

A Role-Net is authorisation based dead marking free if

$$\forall\, M_k^{*Ro}, \exists M_k^{*Ro}(w) = S_w^{Ro} \neq \emptyset, w = 1..m$$

where

$M_k^{*Ro} = [S_w^{Ro}]$ $(w = 1..m)$ is named RO-Token transitive marking which is used to detect whether the transition is qualified to facilitate the change of RO-Token marking, e.g. $M_4^{*Ro} = [\emptyset\ \emptyset\ \emptyset\ r_2^{\varepsilon} \cap \gamma_3\ \emptyset]$. RO-Token transitive marking is calculated as :

$$M_k^{*Ro} = M_{k-1}^{Ro} \vartriangle L_{BP}^{Ro} \Rightarrow [S_w^{Ro}] = \bigcap_{i=1}^{m} ([Q_i] \cap [c_{ig}])$$

where

$M_{k-1}^{Ro} = [Q_i]\ (i = 1 \dots m)$ is called RO-Token marking which indicates the state of Role-Net from the movement of RO-Token point of view.

L_{BP}^{Ro} is role-embedded transitive matrix. $L_{BP}^{Ro} = \left[c_{ig}\right]\ (i, g = 1 \dots m)$

Example RO-Token Transitive Marking

Here we present an example on how Role-Token marking and RO-Token transitive marking work together to detect authorisation policy conflicts. Let us assume:

$$K = 4, M_{k-1}^{R_0} = M_3^{R_0} = [\ \emptyset\ \emptyset\ r_2^{\varepsilon}\ \emptyset\ \emptyset\ ]$$

$$L_{BP}^{R_0} = \begin{bmatrix} \emptyset & \gamma_1 & \emptyset & \emptyset & \emptyset \\ \emptyset & \emptyset & \gamma_2 & \emptyset & \emptyset \\ \emptyset & \emptyset & \emptyset & \gamma_3 & \emptyset \\ \emptyset & \emptyset & \emptyset & \emptyset & \gamma_4 \\ \emptyset & \emptyset & \emptyset & \emptyset & \emptyset \end{bmatrix}$$

$$M_4^{*R_0} = M_3^{R_0}\ \triangle\ L_{BP}^{R_0} = [\ \emptyset\ \emptyset\ r_2^{\varepsilon}\ \emptyset\ \emptyset\ ], \qquad then\ \exists w = 4$$

$$S_4^{R_0} = \begin{cases} r_2^{\varepsilon}\ \cap\ r_2^{\varepsilon} & Role\ exists \\ \emptyset & No\ Role\ exists \end{cases}$$

Let us assume that the Bank expects the **RISK ADMINISTRATOR** to process the loan case due to huge loan application amount. $r_2^{\varepsilon} =$ {"**RISK ADMINISTRATOR**"} and $\gamma_3 =$ {"**RISK OFFICER**", "**RISK ADMINISTRATOR**, "**BANK MANAGER**"}. There exists $S_4^{R_0}$ which is not empty. The Role-Net of Bank is role authorisation based dead marking free. However, if the **RISK OFFICER** is the only available role for this service, then $\gamma_3 =$ {"**RISK OFFICER**"} and $S_4^{R_0} = \emptyset$. The runtime role authorisation error will be detected resulting in the suspension of business collaboration execution in this service.

Consequently, we can conclude that business collaboration reliability in terms of role authorisation can be modelled and detected in Role-Net: (1) Incorrect role assignment or modification in a service can then be seen as the inconsistent design on role available set γ to specific service where the qualified role is not initially set up in , or as incorrect runtime modification on role available set γ which leads to $r_i^{\varepsilon} \cap \gamma = \emptyset$; (2) Unexpected role access in collaborating business partner can be detected as the returned RO-Token R_i^{ε} is not equivalent to the expected role in r_i^{ε}.

7.5 FEATURES AND ADVANTAGES OF ROLE-NET

The message flow and role flow represented as AO-Token and RO-Token in Role-Net are correlated to guarantee that the service will deal with the desired message by the expected role. Any unexpected message or unexpected role will be detected and abandoned before the service is invoked (Individual AO-Token or RO-Token move in Role-Net cannot be accepted).

Role-Net provides a theoretical platform to verify authorisation policy based business collaboration reliability.

Within one organisation: If inconsistent role assignment to specific service becomes true (which can be stated as wrong initially setup on available role set γ) or any incorrect modification on role assignment occurs at runtime (which is represented as the wrong change on the available role set γ), then Role-Net can detect that no qualified authorised roles exist to access the message at the service.

Across organisations The Role-Net can model business collaboration from one organisation point of view to cater for the peer-based nature in loosely-coupled collaborative business environment. Through the refinement function in hierarchical Petri-Nets, the Role-Net in one organisation can view the collaborator's Role-Net and detect whether the message is accessed and

modified by the expected roles in its collaborators. The authorisation policy based business collaboration reliability in the distributed computing environment can thus be verified and enforced.

7.6 SUMMARY

Business collaboration can become unreliable in terms of authorisation policy conflicts, for example, when (1) incorrect role assignment or modification occurs when the required role is inconsistent with the role assignment for a message in a service within one organization, or (2) messages transferred from one organization are accessed by unqualified roles in other collaborating business partners. Current approaches cannot provide model to simulate role authorisation in business collaboration, nor verification mechanism to enforce collaboration reliability in terms of authorisation policy. In this chapter, we provide a role authorisation model (Role-Net) to verify authorisation policy based business collaboration reliability. A reliability property based on Role-Net is also defined and discussed. The mechanism on how to dynamically determine the required roles for each service can be designed by exploring role based authorisation policies with integration of existing Role-Net based policies.

CHAPTER 8

PROTOTYPE DESIGN

In accordance to our proposed specification, the BPEL4RBAC-based system architecture is illustrated in detail in this section. Since BPEL4RBAC extends WS-BPEL, its architecture is WS-BPEL enabled and compatible with existing Web services standards. The following figure describes how the elements and authorisation constraints are enabled from system point of view.

8.1 SYSTEM ARCHITECTURE

The prototype system is designed in a modular way. Organisations can deploy required modules to accommodate different collaboration scenarios.

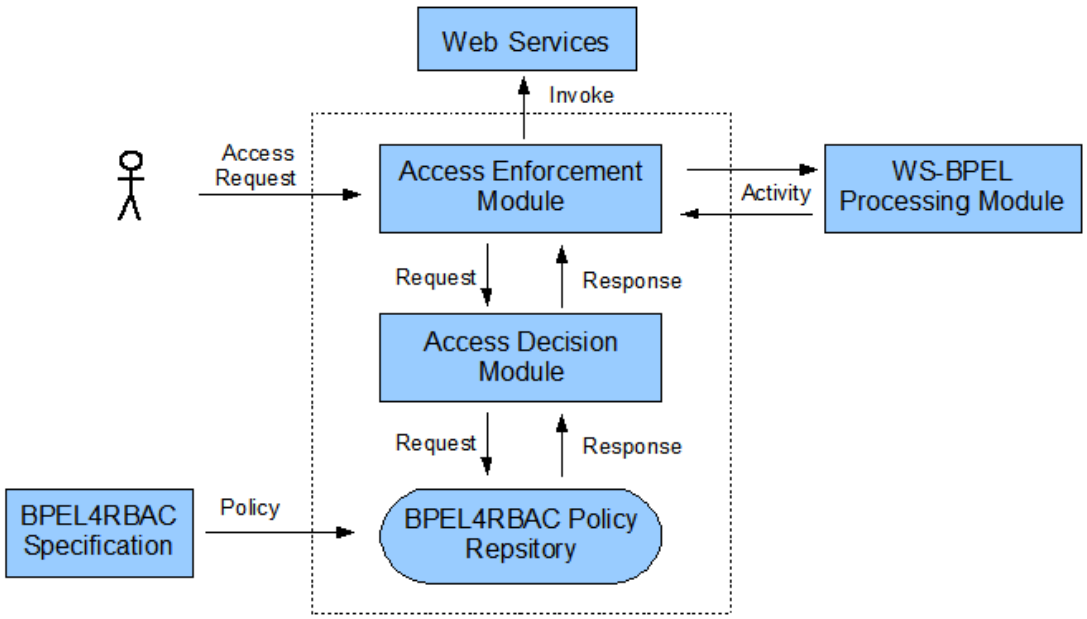

Figure 8-1 System Architecture

The Access Enforcement Module (AEM) is the key component of the entire system. It handles user request, the A forwards the request to Access Decision

Module (ADM). After the response received from ADM, AEM contacts the WS-BPEL module to get the information for requested business activities. Then the corresponding Web services are invoked by AEM.

After receiving the request from AEM, ADM contacts BPEL4RBAC policy repository and perform a check against the user who send this request. The decision is made according to the pre-defined BPEL4RBAC policy which is represented by BPEL4RBAC model.

Based on BPEL4RBAC policies, AEM and ADM work together to ensure the integrity and confidentiality of WS-BPEL represented business processes. Taking the advantage of easy integration with WS-BPEL, BPEL4RBAC can also integrated with message level security standards, such as WS-SecurityPolicy and XACML.

Another benefit comes from the modular design of the system architecture. The BPEL4RBAC policy specification is separated from access decision module. The access decision processing mechanisms for ADM are more flexible with the same security policy. The AEM is separated from ADM as well. Thus the AEM focuses only on coordination with Web services and access control enforcement without touching about any change in access decision strategies and security policies.

Based on the conceptual model, access requests from local role or collaborative role are handled separately in AEM for local or collaborative service respectively. Taking the advantage of the modular design of the system architecture, the authorisation policy is separated from access decision module. The access decision processing mechanisms for ADM are more extensible with the same security policy. The AEM is separated from ADM as well. Thus the AEM focuses only on coordination with Web services and access control enforcement without touching any change in access decision strategies and security policies.

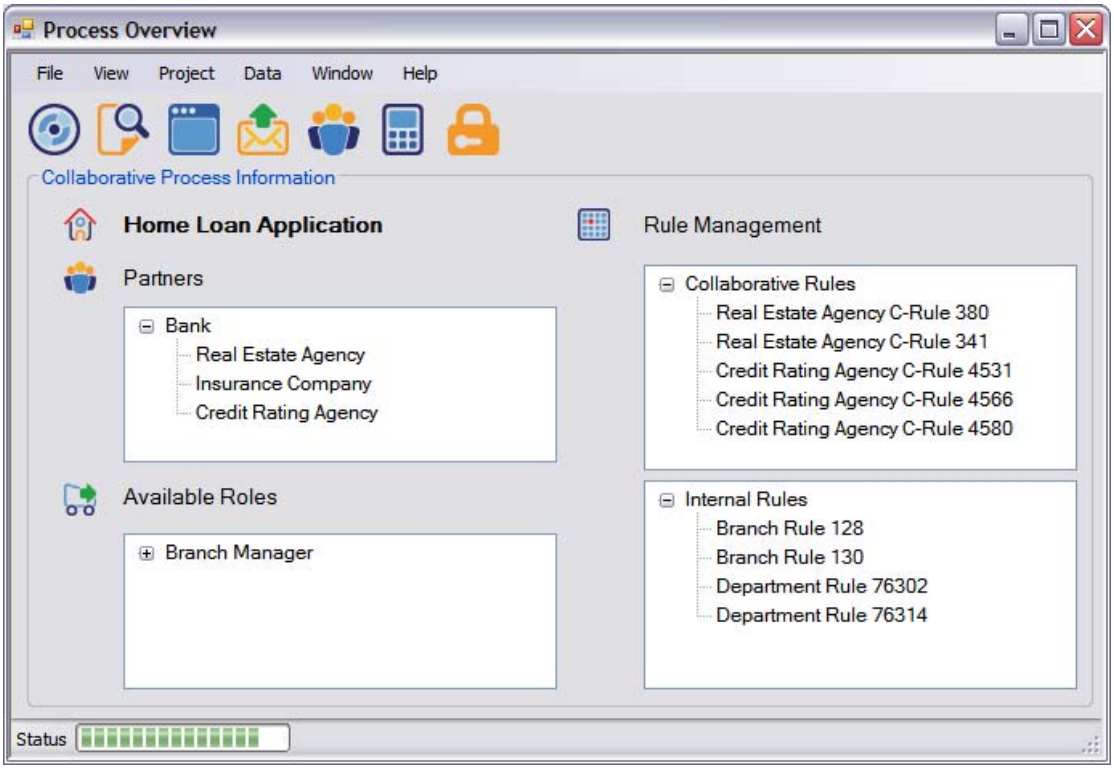

Figure 8-2 Prototype Screenshot

For example, when a customer applies for a home loan, the Home Loan reception service handles the application and categorises the information that the customer submitted. A Clerk is then assigned to handle this new home loan application if the loan amount is less than $1M.

As part of application assessment, the clerk sends the required information to the Real Estate Agency for a property valuation. The information may include the property address which is ranked as a static field in this application instance. In other word, no one can change the proposed address in the whole process.

In this step, the bank's internal Information System is in charge of authenticating the Clerk's identity and makes sure he or she has the permission to contact real estate agency in this case.

After submitting the required information to the Real Estate Agency, the Property Valuation service will process the property valuation application. The

information will go into the real estate agency's internal process which is treated as a black box from the bank's point of view. The assessment result will be release to the collaborative process. But before sending the result to the bank, AEM will check the result against the related rules. For instance, if the property value is over $1M dollars, the result must be approved by department manager in the bank. At this stage, clerk is still the proposed role but the required role is department manager. So a role conflict occurs. The pre-defined rule will prevent the clerk receiving the result. While at the same time, the conflict information will be sending to ADM for a required role.

By searching and checking the available roles in RBAC Rule Repository, ADM will provide a new role, the department manager in this case, to AEM. AEM will consequently check the new role again to enforce system consistency. If the RBAC Rule Repository couldn't find an appropriate role, the participating organizations will proceed to a negotiation process to solve the conflict and design new collaborative rules to handle this kind of situations. The Rule Repository will check the consistency of new rules in the whole collaborative processes. If no further conflicts exist, the new rules will be added in, and otherwise, the affected rules will be sent to the negotiation process aging until there is not conflict exist.

After authorised by AEM, department manager will be assigned as a collaborative role in this process to handle the property assessment result. The department manager will be the only person to approve the loan application in this collaborative process.

In other parallel threads within this process, department manager will be the required role to handle the result from collaborating partners disregarding the current role unless a higher level role is required. For instance, the customer is rated as AAA by the credit rating agency which can be handled by a clerk.

Affected by the property assessment result, however, clerk won't be able to proceed to the next step, and only the department manager can approve the application.

CHAPTER 9

CONCLUSION AND FUTURE WORK

Modern business often requires collaboration between individual social entities with different security policies defined and enforced. The challenge is how these security policies are specified, compared, integrated and managed for collaborative services. None of the published works adequately addresses these issues. The research works introduced in this thesis have developed a security management system that covers the entire life-cycle of security policy management from design time specification to run time monitoring and enforcement. The outcomes in terms of models, algorithms and mechanisms can facilitate the related fields such as secure Web service and business process management.

9.1 SUMMARY OF THIS RESEARCH

Security of computer-based business systems is, by design, the key element for protecting the confidentiality, integrity, and accessibility of the system and services. Given the information and service-intense characteristics of our modern economy, it should be no surprise to learn that security is a growing concern among most organisations. It is especially true when organisations try to construct extensive networks of communication links to engage each other in order to deliver their collaborative business services. For example, a medical centre needs to work with health insurance companies, general practitioners and specialists to deliver its build-to-order service to its customers. In this scenario, different parties may have their own security policies with their own implementation and enforcement mechanism. In order for them to work together and not violate each other's security policy, technological support is

required to allow the parties involved to ascertain that their security policies and their partners' can be checked, tested, and enforced during the collaboration. All of this requires continuously adjusting and aligning security policies within end-to-end business processes that span diverse organisations.

The importance of security in a computer-based environment has resulted in a large stream of research that focuses on the technical defences associated with protection in providing mathematical theory, cryptographic algorithms, and distributed systems and network security solutions. In other words, the existing work in the security area mainly contribute to providing solutions at the data, network, and computer systems level, and target either for single organisation or simple collaborations. However the challenges of security management in the rich domain of business collaboration constitute a vibrant area of security research, which has so far received only limited attention and has never been addressed to its entirety.

The research proposed in this thesis is significant in two aspects. Firstly it addresses the critical security issues in service based business collaboration and provides solutions for the design and integration of secured business services. Security is listed as No. 1 tech flop by InfoWorld [21] review of IT industry practice of the last 20 years. The security of collaborative business process is crucial and significant for the business success of organisations as security problems would affect companies and their stakeholders in terms of profit and reputation.

Secondly, this research work contributes towards developing an extensible framework that is necessary for building and managing secured and extendible e-business applications. This is opposed to the current research activities and standards-oriented approaches that focus mainly on technique based solutions aimed for data, network and system level security.

Business processes are developed separately on different platforms. In most cases, they do not follow the same strategy. Existing BPM methodologies seldom consider security issues which address business integration and legal requirements [22]. Therefore the area of research is of vital importance to software engineering and distributed computing. It plays part in the development of next generation technologies that contribute to a massively distributed computing infrastructure made up of many different Internet resident software services aiming to interoperate over the network to virtually form a single logical system offering on-demand and value-added user services. This research aims to make an impact on fundamental research on security aspects of service oriented collaborations. Further, it aims to develop generic solution that is broadly applicable to several industry sectors and applications such as e-health, e-logistics or e-government.

The outcomes of this research will improve the security protection to service based IT environment. The formal study on security requirement analysis and access control model will contribute to the theoretical advance in service architecture research communities. The proposed specifications will contribute to the practice of service security and business process management related issues.

To the best of our knowledge, there are very few studies reported in literature that systematically and thoroughly address the problem of security issues in service based business collaboration [23]. Current studies with application security approaches have limitations in meeting the challenges in dealing with the complexity of collaborative business although some standardisations have already been achieved in this area [24].

In this research, we undertook a thorough investigation on the problems of SOA security management in terms of RBAC application, WS-BPEL extension, and human activities involvement.

The first innovative aspect of this research lies in the scenario based requirement analysis methodology. This is the first step to clarify the research issue from business world. With the requirements from strategic, organisational, transactional and operational levels, we can enlighten the way to facilitate the business collaboration in a secure way.

The second innovative aspect of this research is the development of authorisation specifications that can be used to specify the security requirements in the service based collaboration systems. The specifications bridge the gap between the business requirements and SOA capabilities in terms of collaboration. Both machine based automatic business processes and human activities are taken into consideration.

The third innovative aspect of this study is the development of a RBAC verification mechanism. The Petri-Net based verification methodology can (1) checked for consistency and comparability at design time; and (2) verified and enforced the agreed (integrated/collaborative) security policy at run time.

9.2 TRADEOFFS OF THIS RESEARCH

The service based business collaboration may bring some tradeoffs in terms of business management, security and WS-BPEL language.

In the security domain, we only focus on RBAC model. A variety of other security models are applied in the enterprise world as we discussed in Chapter 2. We assume that participating organisations are communicating under the RBAC compatible model. If the collaborating partner is using other security model, they

have to enter another round of negotiation to achieve the RBAC compatible collaboration.

In the business level, we assume the communication and negotiation can be done in a perfect manner. In the real world, the unsuccessful business meeting may lead to the breakup of partnership. Trust is another issue in the business world as the reputation is sometime more important than a cheaper price.

At the current stage, the WS-BPEL language is only a proposed solution for service computing. A wider range of adoption will help to promote our specifications in practice. There are more works need to be done to extend elements and attributes in the current version.

9.3 FUTURE WORKS

This research can be extended in many tracks. The other security models should be involved in the business collaboration. A general SOA oriented access control model is more helpful in migrating one model to another. In this way, organisations can communicate in the same language on the negotiation of security level.

The BPEL4RBAC policy language can be further improved by taking consideration of more complicated organisation behaviours. In particular, an organisation may define different roles and different permissions to the same process when cooperating with different partners. Another direction intends to provide connectivity with message level security standards, such as XACML and WS-Policy as we discussed above.

The different patterns of computer service and human activities will be a challenging task for organisations to migrate from legacy system into SOA. As we mentioned previously, even for the WS-BPEL enabled organisations, human

activities are inevitable in some scenarios. We need to find the way of how to balancing these two types of activities in the business collaboration.

Cloud computing is becoming an emerging technology in recent months. It create new way to facilitate collaborative business process management and also introduces new security challenges. The cloud computing will be an attractive research area for both business people and IT researchers.

BIBLIOGRAPHY

[1] J. F. Chang, *Business Process Management System - Strategy and Implementation*. Boca Raton, New York: Auerbach Publications, 2006.

[2] S. Roser and B. Bauer, "A Categorization of Collaborative Business Process Modeling Techniques," Proc of *IEEE International Conference on E-Commerce Technology Workshops*, Washington DC, USA, pp. 43-54, 2005.

[3] W. Tolone, G.-J. Ahn, T. Pai, and S. P. Hong, "Access Control in Collaborative Systems," *ACM Computing Surveys,* vol. 37, pp. 29-41, 2005.

[4] M. T. Siponen and H. Oinas-Kukkonen, "A review of information security issues and respective research contributions," *ACM SIGMIS Database,* vol. 38, pp. 60-80, 2007.

[5] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-Based Access Control Models," *IEEE Computer,* vol. 29, pp. 38-47, 1996.

[6] M. Wu, J. Chen, and Y. Ding, "Study on Role-Based Access Control Model for Web Services and its Application," Proc of *International Conference on Telecommunications and Informatics*, Istanbul, Turkey, 2006.

[7] M. P. Papazoglou and D. Georgakopoulos, "Service-Oriented Computing: Introduction," *Communications of the ACM,* vol. 46, pp. 24-28, 2003.

[8] K. Gottschalk, S. Graham, H. Kreger, and J. Snell, "Introduction to Web Services Architecture," *IBM Systems Journal,* vol. 41, pp. 170-177, 2002.

[9] Y. E. Chan, "Why Haven't We Mastered Alignment? The Importance Of The Informal Organization Structure," *Journal of Strategic Information Systems,* vol. 10, pp. 77-99, 2001.

[10] R. Sabherwal and Y. E. Chan, "Alignment Between Business and IS Strategies: A Study of Prospectors, Analyzers, and Defenders," *Information Systems Research,* vol. 12, pp. 11-33, 2001.

[11] Y. Genovese. (2006). Planning for 2010: Key Issues for Managing Business Processes. [Online]. Available: http://www.gartner.com/DisplayDocument?doc_cd=141119.

[12] N. McAllister. (2008). Tech's all-time top 25 flops. [Online]. Available: http://www.infoworld.com/article/08/01/21/03FE-25-tech-failures_1.html.

[13] P. Chapin, C. Skalka, and X. S. Wang, "Authorisation in Trust Management: Features and Foundations," *ACM Computing Surveys,* vol. 40, 2008.

[14] R. Simon and M. E. Zurko, "Separation of duty in role-based environments," Proc of *10th Computer Security Foundations Workshop (CSFW'97)*, Rockport, MA, USA, pp. 183 - 194, 1997.

[15] P. Liu and Z. Chen, "An Access Control Model for Web Services in Business Process," Proc of *IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 292-298, 2004.

[16] R. K. Thomas and R. S. Sandhu, "Task-based Authorisation Controls (TBAC): A Family of Models for Active and Enterprise-oriented Authorisation Management," Proc of *11th International Conference on Database Security, IFIP WG11*, 1997.

[17] A. A. E. Kalam, R. E. Baida, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte, A. Miege, C. Saurel, and G. Trouessin, "Organization based access control," Proc of *IEEE 4th International Workshop on Policies for Distributed Systems and Networks*, pp. 120-131, 2003.

[18] H. K. Kim, R. Y. Lee, and H. S. Yang, "Frameworks for Secured Business Process Management Systems," Proc of *Fourth International Conference on Software Engineering Research, Management and Applications (SERA '06)*, pp. 57-65, 2006.

[19] S. Indrakanti, V. Varadharajan, and M. Hitchens, "Authorisation Service for Web Services and its Application in a Health Care Domain," *International Journal of Web Services Research,* vol. 2, pp. 94-119, 2005.

[20] J. Wang, "A Web Services Secure Conversation Establishment Protocol Based on Forwarded Trust," Proc of *International Conference on Web Services*, Chicago, Illinois, USA, pp. 569-576, 2006.

[21] E. Bertino, J. Crampton, and F. Paci, "Access Control and Authorisation Constraints for WS-BPEL," Proc of *IEEE International Conference on Web Services (ICWS'06)*, pp. 275-284, 2006.

[22] R. Bhatti, E. Bertino, and A. Ghafoor, "A Trust-Based Context-Aware Access Control Model for Web-Services," *Distributed and Parallel Databases,* vol. 18, pp. 83-105, 2005.

[23] E. Bertino, A. C. Squicciarini, and D. Mevi, "A fine-grained access control model for Web services," Proc of *IEEE International Conference on Services Computing (SCC'04)*, Shanghai China pp. 33-40, 2004.

[24] OASIS. (2006). WS-Security Core Specification 1.1. [Online]. Available: http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf.

[25] W3C. (2006). Web Services Policy 1.2 - Framework (WS-Policy). [Online]. Available: http://www.w3.org/Submission/WS-Policy/.

[26] OASIS. (2007). WS-Trust 1.3. [Online]. Available: http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.html.

[27] OASIS. (2005). Security Assertion Markup Language (SAML) v2.0. [Online]. Available: http://docs.oasis-open.org/security/saml/v2.0/saml-2.0-os.zip.

[28] X. Zhang, M. Nakae, M. J. Covington, and R. S. Sandhu, "Toward a Usage-Based Security Framework for Collaborative Computing Systems," *ACM Transactions on Information and System Security (TISSEC),* vol. 11, 2008.

[29] M. Shehab, K. Bhattacharya, and A. Ghafoor, "Web services discovery in secure collaboration environments," *ACM Transactions on Internet Technology,* vol. 8, 2007.

[30] M. P. Papazoglou and W. Heuvel, "Service oriented computing: state-of-the-art and open research issues," Proc of *International Conference on Service Oriented Computing*, Trento, Italy, 2003.

[31] R. Kanneganti and P. Chodavarapu, *SOA Security*: Manning Publications, 2008.

[32] J. C. S. P. Leite, J. H. Doorn, G. D. S. Hadad, and G. N. Kaplan, "Scenario inspections," *Requirements Engineering,* vol. 10, pp. 1-21, 2005.

[33] J. M. Carroll, *Scenario-based design: envisioning work and technology in system development*: John Wiley & Sons, Inc. New York, NY, USA, 1995.

[34] H. Wang, Y. Zhang, J. Cao, and J. Yang, "Specifying Role-Based Access Constraints with Object Constraint Language," Proc of *6th Advanced Web Technologies and Applications*, Hangzhou, China, pp. 687-696, 2004.

[35] D. D. He and J. Yang, "Access Control: What is required in Business Collaboration," Proc of *Australian Database Conference*, New Zealand, 2009.

[36] H. Sun, X. Wang, J. Yang, and Y. Zhang, "Authorisation Policy Based Business Collaboration Reliability Verification," Proc of *International Conference on Service Oriented Computing (ICSOC'08)*, Sydney, Australia, pp. 579 - 584, 2008.

[37] A. Lindsay, D. Downs, and K. Lunn, "Business Processes - Attempts to Find a Definition," *Information and Software Technology,* vol. 45, pp. 1015-1019, 2003.

[38] M. P. Papazoglou, "Web Services and Business Transactions," *World Wide Web,* vol. 6, pp. 49-91, 2003.

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

[40] P. Green and M. Rosemann, "An Ontological Analysis of Integrated Process Modelling," Proc of *Conference on Advanced Information Systems Engineering (CAiSE)*, Heidelberg, Germany, pp. 225-240, 1999.

[41] R. Bort and G. R. Bielfeldt, *Handbook of EDI*. New York: Warren, Gorham & Lamont, 1994.

[42] V. A. Leyland, *Electronic data interchange: a management view*. New York: Prentice Hall, 1993.

[43] S. Narayanan, A. S. Marucheck, and R. B. Handfield, "Electronic Data Interchange: Research Review and Future Directions," *Decision Sciences, Blackwell Publishing Inc,* vol. 40, pp. 121-163, 2009.

[44] WfMC. (1993). Workflow Management Coalition. [Online]. Available: http://www.wfmc.org/.

[45] H. A. Reijers, R. S. Mans, and v. d. T. R. A., "Improved model management with aggregated business process models," *Data & Knowledge Engineering, Elsevier,* vol. 68, pp. 221-243, 2009.

[46] N. Berente, B. Vandenbosch, and B. Aubert, "Information flows and business process integration," *Business Process Management Journal, Emerald Ltd,* vol. 15, pp. 119-141, 2009.

[47] W3C. (2006). Web Services Architecture Group Note. [Online]. Available: http://www.w3.org/TR/ws-arch/#whatis.

[48] M. P. Papazoglou and J. Dubray, "A Survey of Web Service Technologies," *Technical Report, University of Trento,* vol. DIT-04-058, 2004.

[49] W3C. (2007). Simple Object Access Protocol (SOAP) Specification V1.2. [Online]. Available: http://www.w3.org/TR/soap12-part1/.

[50] W3C. (2007). Web Services Description Language (WSDL) 2.0. [Online]. Available: http://www.w3.org/TR/wsdl20/.

[51] OASIS. (2004). UDDI Version 3.0.2. [Online]. Available: http://uddi.org/pubs/uddi_v3.htm.

[52] F. Leymann, D. Roller, and M. T. Schmidt, "Web services and business process management," *IBM Systems Journal,* vol. 41, pp. 198-211, 2002.

[53] OASIS. (2007). Web Services Business Process Execution Language v2.0. [Online]. Available: http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.pdf.

[54] IBM. (2001). Web Services Flow Language (WSFL). [Online]. Available: http://www.ibm.com/software/solutions/webservices/pdf/WSFL.pdf.

[55] Microsoft. (2001). XLANG: Web Services for Business Process Design. [Online]. Available: http://technet.microsoft.com/en-us/library/aa577463.aspx.

[56] W3C. (2004). XML Schema. [Online]. Available: http://www.w3.org/XML/Schema.

[57] W3C. (1999). XML Path Language (XPath). [Online]. Available: http://www.w3.org/TR/xpath.

[58] OASIS. (2007). WS-SecurityPolicy V1.2. [Online]. Available: http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702/ws-securitypolicy-1.2-spec-os.pdf.

[59] OASIS. (2007). WS-SecureConversation 1.3. [Online]. Available: http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/ws-secureconversation-1.3-os.html.

[60] IBM-SAP, "WS-BPEL Extension for People (BPEL4People)," *A joint white paper,* 2007.

[61] (2007). Web Services Human Task (WS-HumanTask) [Online]. Available: http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel4people/WS-HumanTask_v1.pdf.

[62] IBM. (2002). Web Services Policy Assertions Language. [Online]. Available: http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-polas/ws-polas.pdf.

[63] SUN. (2005). XACML-Based Web Services Policy Constraint Language (WS-PolicyConstraints). [Online]. Available: http://research.sun.com/projects/xacml/IntroToWSPolicyConstraints.pdf.

[64] W3C. (2006). Web Services Policy 1.2 - Attachment (WS-PolicyAttachment). [Online]. Available: http://www.w3.org/Submission/WS-PolicyAttachment/.

[65] OASIS. (2005). eXtensible Access Control Markup Language TC v2.0 (XACML). [Online]. Available: http://docs.oasis-open.org/xacml/2.0/XACML-2.0-OS-NORMATIVE.zip.

[66] H. Wang, J. Cao, and Y. Zhang, "A Flexible Payment Scheme and its Role-Based Access Control," *IEEE Transactions on Knowledge and Data Engineering,* vol. 17, pp. 425-436, 2005.

[67] M. Sloman and E. Lupu, "Security and management policy specification," *IEEE Network,* vol. 16, pp. 10-19, 2002.

[68] C. Yang, "Designing secure e-commerce with role-based access control," *International Journal of Web Engineering and Technology,* vol. 3, pp. 73-95, 2007.

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

[70] R. Milner, *Lectures on a calculus for communicating systems*. London, UK: Springer-Verlag, 1984.

[71] T. Bolognesi and E. Brinksma, "Introduction to the ISO Specification Language LOTOS," *Computer networks and ISDN systems,* vol. 14, pp. 22-59, 1987.

[72] J. L. Peterson, *Petri net theory and the modeling of systems*. Upper Saddle River, NJ, USA: Prentice Hall, 1981.

[73] C. Girault and R. Valk, *Petri nets for systems engineering: a guide to modeling, verification, and applications*. New York, Secaucus, NJ, USA: Springer-Verlag, 2003.

[74] T. Murata, "Petri Nets: Properties, Analysis and Applications," *IEEE,* vol. 77, pp. 541-580, 1989.

[75] Y. Song and J. Lee, "Deadlock Analysis of Petri Nets Using the Transitive Matrix," Proc of *41st SICE Annual Conference*, pp. 689 - 694, 2002.

[76] K. Knorr, "Dynamic access control through Petri net workflows," Proc of *16th Annual Computer Security Applications Conference (ACSAC'00)*, p. 159, 2000.

[77] CrossFlow Project. [Online]. Available: http://www.crossflow.org/.

[78] INTEROP Project. [Online]. Available: http://www.interop-noe.org.

[79] ATHENA Project. [Online]. Available: http://www.athena-ip.org.

[80] ECOLEAD [Online]. Available: http://www.ecolead.org.

[81] GLOBEMEN. [Online]. Available: http://globemen.vtt.fi/.

[82] SPIDER-WIN Project. [Online]. Available: http://www.spider-win.de/spider-win.htm.

[83] J. A. Zachman, "Enterprise Architecture: The Issue of the Century," *Database Programming and Design,* vol. 10, p. 44, 1997.

[84] S. J. Bleistein, A. Aurum, K. Cox, and P. K. Ray, "Strategy-Oriented Alignment in Requirements Engineering: Linking Business Strategy to Requirements of e-Business Systems using the SOARE Approach," *Journal of Research and Practice in Information Technology,* vol. 36, pp. 259-276, 2004.

[85] M. Wang and H. Wang, "From process logic to business logic: A cognitive approach to business process management," *Information & Management,* vol. 43, pp. 179-193, 2006.

[86] R. Weber and Y. Zhang, "An ontological evaluation of Niam's grammar for conceptual schema diagrams," Proc of *Twelfth international conference on Information systems table of contents*, New York, USA, pp. 75-82, 1991.

[87] R. Weber and Y. Zhang, "An analytical evaluation of NIAM's grammar for conceptual schema diagrams," *Infomation Systems Journal,* vol. 6, pp. 147-170, 1996.

[88] W. Reisig, *Petri Nets: An Introduction*: Springer-Verlag New York, Inc. New York, NY, USA, 1985.

[89] W. Reisig, *Primer in Petri Net Design*: Springer-Verlag New York, Inc. Secaucus, NJ, USA, 1992.

[90] S. Androutsellis-Theotokis and D. Spinellis, "A survey of peer-to-peer content distribution technologies," *ACM Computing Surveys (CSUR),* vol. 36, pp. 335 - 371, December 2004 2004.

[91] J. S. Park and J. Hwang, "Role-based access control for collaborative enterprise in peer-to-peer computing environments," Proc of *ACM symposium on Access control models and technologies*, Como, Italy, pp. 93 - 99, 2003.

[92] D. S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, and Z. Xu, "Peer-to-Peer Computing," *HP Laboratories Palo Alto,* 2002.

[93] H. Shi, Y. Zhang, J. Zhang, E. Beal, and N. Moustakas, "Collaborative Peer-to-Peer Service for Information Sharing Using JXTA," Proc of *International Multi-Symposium on Computer and Computational Sciences (IMSCCS'06)*, Hangzhou, Zhejiang, China, pp. 552 - 559, 2006.

[94] W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmér, and T. Risch, "EDUTELLA: a P2P networking infrastructure based on RDF," Proc of *11th international conference on World Wide Web*, Honolulu, Hawaii, USA, pp. 604 - 615, 2002.

[95] D. Schoder and K. Fischbach, "Peer-to-peer prospects," *Communications of the ACM,* vol. 46, pp. 27 - 29, 2003.

[96] P. Rodriguez, S.-M. Tan, and C. Gkantsidis, "On the feasibility of commercial, legal P2P content distribution," *ACM SIGCOMM Computer Communication Review,* vol. 36, pp. 75 - 78, 2006.

[97] M. A. Einhorn and B. Rosenblatt, "Peer-to-Peer Networking and Digital Rights Management," *CATO Institute Policy Analysis,* 2005.

[98] E. Beal, N. Moustakas, and J. Williams, "Peer-to-peer Collaborative Research Project: Draft Licence Agreements," Communication Law Centre 2006.

[99] C. Qu and W. Nejdl, "Interacting the Edutella/JXTA peer-to-peer network with Web services," Proc of *International Symposium on Applications and the Internet*, 2004.

[100] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," *International Journal of Human-Computer Studies,* vol. 43, pp. 907 - 928, 1995.

[101] W3C. (2004). OWL Web Ontology Language. [Online]. Available: http://www.w3.org/TR/owl-features/.

[102] "JXTA Project," 2006. Available: https://jxta.dev.java.net/.

[103] B. von Halle and A. Sandifer, "Desinging by the Rules," *Database Programming and Design,* vol. 4, pp. 11-14, 1991.

[104] R. G. Ross, *The Business Rule Book: Classifying, Defining and Modeling Rules*: Business Rule Solutions, Houston, Tex., 1997.
[105] R. G. Ross, *Principles of the Business Rule Approach*: Addison-Wesley Professional, 2003.
[106] OMG, *Semantics of Business Vocabulary and Business Rules Specification*: Object Modeling Group, 2006.
[107] B. von Halle, *Business Rules Applied*. New York: John Wiley & Sons, 2002.
[108] R. S. Sandhu and P. Samarati, "Access Control: Principle and Practice," *IEEE Communications,* vol. 32, pp. 40-48, 1994.
[109] X. Wang, Y. Zhang, H. Shi, and J. Yang, "BPEL4RBAC: An Authorisation Specification for WS-BPEL," Proc of *Web Information Systems Engineering (WISE'08)*, Auckland, New Zealand, pp. 381 - 395, 2008.
[110] D. Ferraiolo, J. Cugini, and D. R. Kuhn, "Role-based access control (RBAC): Features and motivations," Proc of *11th Annual Computer Security Application Conference*, pp. 11 - 15, 1995.
[111] Y. Song and J. Lee, "Deadlock analysis of Petri nets using the transitive matrix," Proc of *41st SICE Annual Conference*, pp. 689- 694, 2002.