Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 4 / ISSUE 2 / AUGUST 2009 / 88-109
© 2009 Universidad de Talca - Chile

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762009000200008

Authorization Control in Collaborative Healthcare Systems

Daisy Daiqin He[1] and Jian Yang[2]

[1] Macquarie University, Department of Computing, daiqin@ics.mq.edu.au
[2] Macquarie University, Department of Computing, jian@ics.mq.edu.au

Received 15 January 2009; received in revised form 25 June 2009; accepted 16 July 2009

Abstract

Modern healthcare systems require collaborations between individual social entities such as hospitals, medical centers, emergency services and community services. One of the most critical issues in this setting is security and privacy, i.e., who can access what and based on which condition(s). In the healthcare system that crosses different administrative domains, each business unit has its own security policies defined and enforced. Therefore the challenge is how security policies shall be specified, compared and integrated if necessary depending on the nature of the inter-domain collaboration. In this paper, we discuss the challenging access control issues in cross-domain healthcare systems. A framework is provided to support authorization control in such an environment, which takes collaboration semantics into account, as well as individual participant's authorization policies.

Key words: Authorization Control, Access Control, Service Composition, Web Service Collaboration, Web Service Security

1 Introduction

1.1 Security Issues in Business Collaboration

Ankur Laroia from Southern Union Company recorded the following comments: 'People often forget that healthcare is a many-to-many business. You are not just connecting a hospital to a handful of its branch clinics but to an array of internal and external data sources and applications,' notes Leo Sayavedra, an executive at the Sequence Group, an IT consulting company specializing in systems integration. Each healthcare provider, he says, is an information node that sends and receives transactions to entities outside its firewall [20].

In a complex environment like healthcare, countless interactions are carried out among numerous hospitals and institutes in different forms and based on different devices and systems. Technologies are needed to support seamless, secure and dynamic inter-organizational collaborations. Emerging Web Service technologies have provided technological support for collaborations that cross organizational boundaries. However, security concerns become one of the main barriers that prevent widespread adoption of the new technologies. Authorization control in web services, particularly in collaborative environment is an area that has not seen many developments.

Security control in inter-organizational collaboration has different focus from single organization environments. In a single organization, the authorization control policy can be defined in terms of roles and their privileges with the adoption of Role Based Access Control (RBAC) [29]. Given a request to access a resource or perform an operation, the policy is enforced by analyzing the credentials of the requester and the decision is made on whether the requester can perform the requested actions.

Inter-organizational collaborations in distributed environment, like healthcare, has the following characteristics. Firstly, each organization manages its own resources and defines its own authorization policies based on its own interest. Secondly, individual participating organization can join and leave a collaboration at anytime. Thirdly, an organization can play different roles in a collaboration, it can be a service owner, an agent, or a consumer. An organization can also play several roles at a time. Different roles can imply differences on control power of the participant over the collaboration. Fourthly, organizations collaborate with each other in various ways, which require different security control. Due to the nature of the collaboration as just analyzed above, the following issues can happen:

- **Unauthorized Service Propagation:** in inter-organizational collaboration, a service can be accessed by a party who can pass the access rights to other parties. It is important to understand whether and under what conditions the privilege are allowed to be forwarded to other parties. It is also necessary to use authorization policies to control the way in which information or services are propagated between organizations. This is critical with the trend of implementing Electronic Patient Records (EPR) in the developed world. Suppose a patient wants to keep his health records privately except to the attending doctors. If the policy of the doctor's medical center allows other research institutes to access its patient records, then the patient's privacy can not be respected.

- **Authorization policy Inconsistencies:** each organization defines its own authorization policies base on its own interest. Because each organization acts autonomously, inconsistencies between policies of different organizations are common. It is important to understand whether and under what conditions organizations are able to collaborate with the existence of inconsistencies between authorization policies. Suppose, if a radiology institute wants to collaborate with a medical center in accepting bookings from the medical center's online booking system. However, the medical center accepts both Medicare number and OSHC number (Oversea Student Health Cover) as valid credentials to access the online booking system, while the radiology institute does not accept OSHC number as credential to access the service. It is important to understand whether the collaboration can be achieved securely with the existence of this kind of inconsistencies between collaboration participants.

- **Changing participants:** authorization policies of individual organization normally do not address dynamic user groups for collaboration. Each participating organization has its own policies in the collaboration, and participants of the collaboration can change. Authorization policies specified by each individual participant need to be configured together to control the interactions and accesses in the collaboration.

Therefore, before organizations engage in any collaborations, their authorization policies need to be evaluated to determine the possibility of collaboration under the authorization constraints defined by each party. The consistency of authorization policies of different organizations needs to be evaluated before collaboration can be formed. Intuitively, the concept of "authorization policy consistency" means that (for the same service) the authorization policies of different organizations are conflict free, and organizations are able to collaborate in the intended way securely in terms of authorization control policies.

Authorization control issues in a single organization and a single domain have been well studied [17], [35]. A number of works focused on specifications of authorization policy [5], [33], which help us to identify elements and possible constraints in authorization policy. As few extensions to emerging standard, such as SMAL, XACML, WS-Security have been proposed to address the context sensitive aspects [4], [10]. However, authorization controls in collaborative environments have just started to attract the attention of the research community [35], [42], and little attention has been given to consistency study between access control policies of different collaboration parties.

Our research focuses on authorization control issues in inter-organizational collaboration with complex scenarios in the Healthcare domain. We first introduce two important concepts: **Collaboration Pattern** and **Collaboration Policy**

- **Collaboration Pattern** is the way business collaboration is carried out. For example, organizations can collaborate with each other through an agent; organizations can play a role of middleman that passes one organization's service to another.
- **Collaboration Policy** determines whether the prospective partners are able to collaborate securely for the desired collaboration pattern according to their authorization policies. With the presences of policy inconsistencies, **Collaboration Policy** decides if policy inconsistencies are acceptable or negotiable for the intended collaboration. **Collaboration Policy** provides control guidelines to secure inter-organizational collaborations and filter out the participants whose authorization policies are unsuitable for the desired collaboration. **Collaboration Policy** is based on individual party's authorization policy, as well as the collaboration pattern.

In this paper, we present a Policy Driven Authorization Control (PDAC) framework which consists of two major components: **Collaborability Control Center** and **Policy Enforcement Point**. The **Collaborability Control Center** analysis pattern of the requested collaboration and evaluate service authorization policy of the prospective partner. The Policy Enforcement Point control the access to the requested service according to the corresponding authorization policies. This work is an extension of the work presented in [14].

The rest of the paper is organized as follows. In Section 2, we will introduce the research methodology. Related work are discussed in Section 3. An overview of the proposed Policy Driven Authorization Control framework is introduced in Section 4. Major components of the proposed framework are discussed in details in Section 5 and Section 6. In Section 5, we introduce the collaboration pattern analyzer segment of the PDAC framework. Business collaboration patterns are concluded with the examples in Healthcare domain in Section 5.1, collaboration pattern specification and identification methods are provided in Section 5.2 and 5.3. Section 6 introduces the collaborability analyzer in detail. Concluding remarks are given in Section 7.

2 Research Methodology

We adopt design science as our research methodology and adhere to the guidelines suggested in [15]. Design science is a research paradigm for information systems research. We follow the guideline and provide the artifacts for discovered challenges in inter-organizational collaboration environment.

Design as an Artifact: Several research artifacts have been proposed in our research. We presented a PDAC framework to provide the basis for secure inter-organizational collaboration in the dynamic distributed environment. Method has been provided for collaboration pattern identification. Formally specifying collaboration pattern provides the first step towarding business collaboration pattern determination. Moreover, an authorization policy model is proposed to conduct comparison and evaluation between authorization policies of prospective partners. The policy model is encoded in Description Logic and could be proofed by an automatic prover.

Problem Relevance: Electronic commerce is growing rapidly in different industries. Healthcare is such an industry that involves inter-organizational interactions and collaboration. Security concerns become the major barriers for wide adoption of new technologies, and authorization control has become the major challenge for implementation of e-health records and other advanced collaborations. Our research aims to discover challenges in such a context and tackle the identified problems.

Design Evaluation: Scenario based descriptive evaluation is adopted in our research. Sample scenarios are provided in problem identification and solution production. No formal evaluation attempted in the sense of comparison with other artifacts. There simply are no existing artifacts that address exactly the same problem. However, informal argument is used in the discussion of relevant research.

Research Contribution: In our research, different challenges in business collaboration are identified and discussed, such as service propagation, policy inconsistencies. We addressed these challenges with a framework, which guards authorization control in business collaboration.

Research Rigor: Mathematical formalisms are used to describe the specified and constructed artifacts. We use set theory and formal mathematics to specify collaboration patterns. The proposed authorization policy model is encoded in description logic for its expressiveness, and automatic reasoning tool has been used for the automatic proof.

Design as a Search Process: Authorization control in collaborative environment is an area that has not been well studied. Sound solutions are not existing in the knowledge base. Challenges and problems are identified in the paper by performing a thorough analysis of business requirements in healthcare domain.

Communication of Research: Our work provides clear information to both managerial and technical audience. On the one hand, We provide a clear introduction of problems and challenges in the dynamic inter-organizational collaboration in Healthcare industry. Current state of art on relevant E-health studies are briefly summarized in Section 6. On the other hand, authorization policy model is encoded in Description Logic and collaboration patterns are specified with formal mathematical methods.

3 Current State of the Art and Related Work

Our work engages in:

- Discover various challenging authorization control issues in complex inter-organizational collaboration environments, such as health care;
- Propose solution for the authorization control in business collaboration, which takes individual organization's authorization policy into account, as well as collaboration patterns.

We need a framework that could model and specify authorization policies from participating parties so that policies can be compared, verified and integrated if necessary. Our research relates to works across different areas, such as access control, e-health, access policies and service collaboration etc, we discuss these works in following.

3.1 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 organizations.

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 [9] 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) [29] has emerged in 1990s. By associating permissions and roles, RBAC allows the access control model in the same way that maps naturally to an organization's structure, and the concept of a role is in correspondence to an organizational 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 [32]. To protect the interest of organizations, the conflicting roles must not be assigned to the same user in a business process [22]. 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 [36]. TBAC approach separates system level activities to support scalable and reusable access control models. Organization based Access Control (OBAC) [18] model aims to share specific data and functionality with collaboration partners. The specification of the security policy is completely parameterized by the organization 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 can not provide adequate shield for SOA due to its complexity [19].

Research has also been done in the area of security policy specification [1], [16], [40]. 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 [2], [4], [16], 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 [26], WS-Policy [38], and WS-Trust [27]. 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) [25] on the other hand, is used to exchanging authentication and authorization 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 works have been done and outlined in the area of collaborative systems [37], [43]. However these work only focused on the aspects of policy specification and modeling for protecting data and resources. Because these works were not set up in the context of service based business collaboration, the issues of policy consistency and comparability among different organizations were completely overlooked. In [30], 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 discussed in our work. There are few papers on Web service authorization control in the collaborative environment. We are aware of the work presented by [35], which presented a framework for managing authorization policies for Web service compositions. The proposed framework addressed authorization policy conflicts and provided methodology for conflicts detection. [42] proposed an approach to security policy integration and conflict reconciliation, which is relate to our research. The authors presented a similarity-based policy adaptation algorithm to adapt changes in collaborative groups and a negotiation-based protocol for conflict reconciliation but they neglected the fact that different types of collaboration affect the way the collaboration policy is developed as well as the requirements on collaborative partner's authorization policy. An evaluation on collaborative partner's authorization policies have to be carried out before the collaboration is established. Our work is to fill in this gap.

3.2 Privacy and Security Control in E-Health

Advanced technologies and protocols have been instrumental in rethinking and redesigning key elements of health, such as electronic health record. HL7 has been developed as healthcare standards for data and message exchange between different information systems. Web Services technologies promise dynamic collaboration and provide solution to many integration problems. Among these technologies, a lot of attentions have been focused on privacy and security issues relate to health records sharing and accesses. In [11], the author outlines and discusses a number of e-consent issues concerning an individual's access to information held in electronic health record. In [28], consent can be declared by the patient concerns, using a computing facility, the consent is expressed in an e-consent object that enables it to be communicated to other parties. A specific system could use the consent to permit or deny access to the patient record. The e-consent object can be expressed as an assertion. In [41], the authors developed a prototype 'e-Medical Book' for the electronic health record systems, which implementing patients' consent in electronic health records. We regard patient's consent as patient's authorization policy, patient's policy could be used in authorization control and policy evaluation. We consider patient records sharing in the service propagation problem, evaluate other party's authorization policy against the patient's policy for the intended collaboration type before the collaboration is established could protect patient privacy and reduce security breach.

3.3 Policy Comparison in Distributed System

As we mentioned, the collaboration participants' authorization policies should be evaluated before the collaboration is set up to avoid potential conflicts and security breach. Therefore, the authorization policies of individual organizations need to be compared and checked. Policy issues in distributed systems have been actively growing over the years, In [7], the authors proposed an inclusion mappings based policy comparison method, which could be useful for rule-based policies, particularly for recursive rules. Work in [39] addressed security policy reconciliation issues in distributed computing environments, and focus on security provisioning policy. The authors based their reconciliation on structure of the policies, which is similar to our work. It also provides an alternative for policy specification. However, our focus is policy comparison and evaluation rather than reconciliation and we address authorization policies.

In [21], the authors proposed a collaborative resource model, and presented a secure resource coordination mechanism to assist dynamic collaboration within a distributed environment. Before collaboration begins, the involved resources are coordinated. The coordination service compares the requester's policy against the resource provider's policy, which is partly similar to the function of our collaborability analyzer. The requester's policy they proposed only stats how the requesters will redistribute the acquired resources, which may not sufficient for secure collaboration. Different ways of collaboration could also affect the comparison process. These works provided insights on different policy comparison methods, which are valuable for our policy comparison process.

3.4 Apply Description Logic in Access Control

In our proposed framework, description logic has been used for the process of authorization policy evaluation in the collaborability analyzer. Description logics are weak fragments of first-order logic. The compensation for their limited expressively is decidability, making them an option when automated proof is required. Description Logic

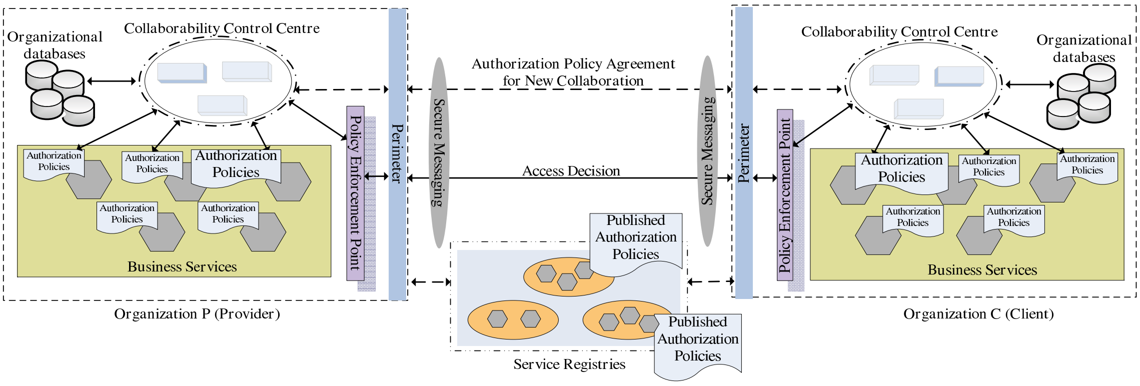

Figure 1: Policy Driven Authorization Control Framework

(DL) has been used in different aspects of access control [8], [23], [31]. In [31], the author proposed an efficient solution to XML access control that use description logic and decidable rules to decide the permissions for an individual at the time they request information. Work presented in [8] demonstrated how to express the RBAC concept in object-oriented systems using description logic and how to use DL to make authorization decisions between the role hierarchy and the object hierarchy.

In summary, most of existing related works only 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 organizations is achieved, theoretical models and mechanism can be developed. This is exactly what this research is heading to and we believe this study is the first step toward achieving an understanding of a secure business collaboration in terms of authorization control.

4 Policy Driven Authorization Control (PDAC) Framework

Authorization control policy is a fundamental part of any authorization control mechanism. Questions such as: how to handle policies that are defined by independent collaboration participants in inter-organizational collaborations. Are organizations, hospitals able to collaborate in certain form securely in terms of authorization policy? To answer these questions, we propose a **Policy Driven Authorization Control** framework to assist authorization control in dynamic collaborative environment.

In service-oriented architecture, services are published in registries or repositories. If a service wants to collaborate with other services, it needs to search the repositories for a service with the required functionality. In our framework, as shown in Figure 1, besides of the functionality and technical requirements, the repository also list relevant authorization policies that are useful for other services to decide if they can fulfill the requirements for binding. Thus, a service searching for a collaboration partner can search the repository for a service with the required functionality and then check if its authorization control policies are compatible.

Authorization control in dynamic collaborative environment has two different aspects. On the one hand, the organization needs to check the requester's credentials and make authorization decision based on its authorization policy. On another hand, the organization needs to evaluate the collaborability of the potential collaboration partners in terms of their authorization policies when a collaboration is required. Depend on the nature of the requested service, both types of authorization controls are required for services that involve inter-organizational collaborations. Therefore, an important step in authorization control in such collaborative environments is to determine: (1) whether the requested service involves any collaboration and if the service is collaboration based?

(2) what type of collaboration it is.

Our proposed framework addresses the issues mentioned above. Two phases of authorization control are required before the access right could be granted to the requester. The first phase checks whether the requested service is collaboration based or not and determines the collaborability of the collaboration partner for the requested service. In the proposed PDAC framework, each organization that provides services in business collaboration has a **Collaborability Control Center (CCC)** and a **Policy Enforcement Point (PEP)** associate with it. The **CCC** handles the first phase of the authorization control. The second phase of the authorization control evaluates whether the requester entitle the requested service base on predefined authorization policies. The second phase of authorization control is realized by PEP. In Figure 1, both the client and the provider organization have a **CCC** and a **PEP**. In dynamic collaborative environments, organization can provide multiple services, and it can be *client* for one service and *provider* for another service.

The **CCC** and **PEP** are critical components for both parties. From the client side, if a service collaboration is required, the organization needs to find a partner that provides required functionality and with compatible authorization policies. From the *provider* side, once a service request is received, the provider organization needs to check if the requested service is based on dynamic inter-organizational collaboration. The provider organization also need to find a partner that provides required functionality and with compatible authorization policies. That is, the provider becomes another client for another collaboration.

As shown in Figure 1, PEP makes access decisions and the messages are sent to client through secure protocols, such as WS-Security [26] and WS-SecureConversation [24]. Authorization policies for established collaborations that configured through the **CCC** at design time are also communicated securely between both organizations.

CCC is the integral part of the PDAC framework, Figure 2 shows the details inside the collaboration control center. As shown in Figure 2, **CCC** consists of three functional segments:

- Collaboration pattern analyzer: analysis the nature of the requested service and identify collaboration pattern of the requested service;
- Collaborability analyzer: evaluate collaboration partner's authorization policy according to the collaboration policy for the identified collaboration pattern;
- Authorization policy configuratator: configure authorization policies for established collaborations.

When a service request is received by an organization, the request is passed to the **CCC** first. The nature of the requested service will be analyzed, and the collaboration pattern analysis will be conducted to determine the collaboration pattern of the requested service. For inter-organizational collaboration based service, the collaborability analysis is required. The analysis result could be collaborable, negotiable and not collaborable. In the case of collaborable, the **CCC** will configure authorization policies for the established collaborations and send to the requester to govern the accesses if a composite service is formed. Agreed policies will be added to the organizational policy databases.

PEP is in charge of the second phase of the authorization control. **PEP** is the place where the requester's credentials are evaluated according to the predefined authorization policies for the requested service. New authorization policies for new collaborations will be added to the existing authorization policy database. The functions of **PEP** include requester's credential validation, credential verification, role assignments, condition enforcement and access decision making. The result of the second phase of authorization control could be: access right granting, reject and negotiation in the case of more credentials are required from the requester. However, our work is focused on the first phase of the authorization control.

It is important to differentiate authorization policy that used to check the requester's credential and make access decisions at the **PEP** with the authorization policies we evaluate in the **CCC**. In **CCC**, we compare authorization policies of the potential partner with the policies of the request receiver for the requested service. These authorization policies are defined within each individual organization for its own interest. The comparisons and analysis between these policies determine the collaborability of the intended collaboration. No credential validations and verifications are carried out in the control center. The authorization policies used in PEP is the

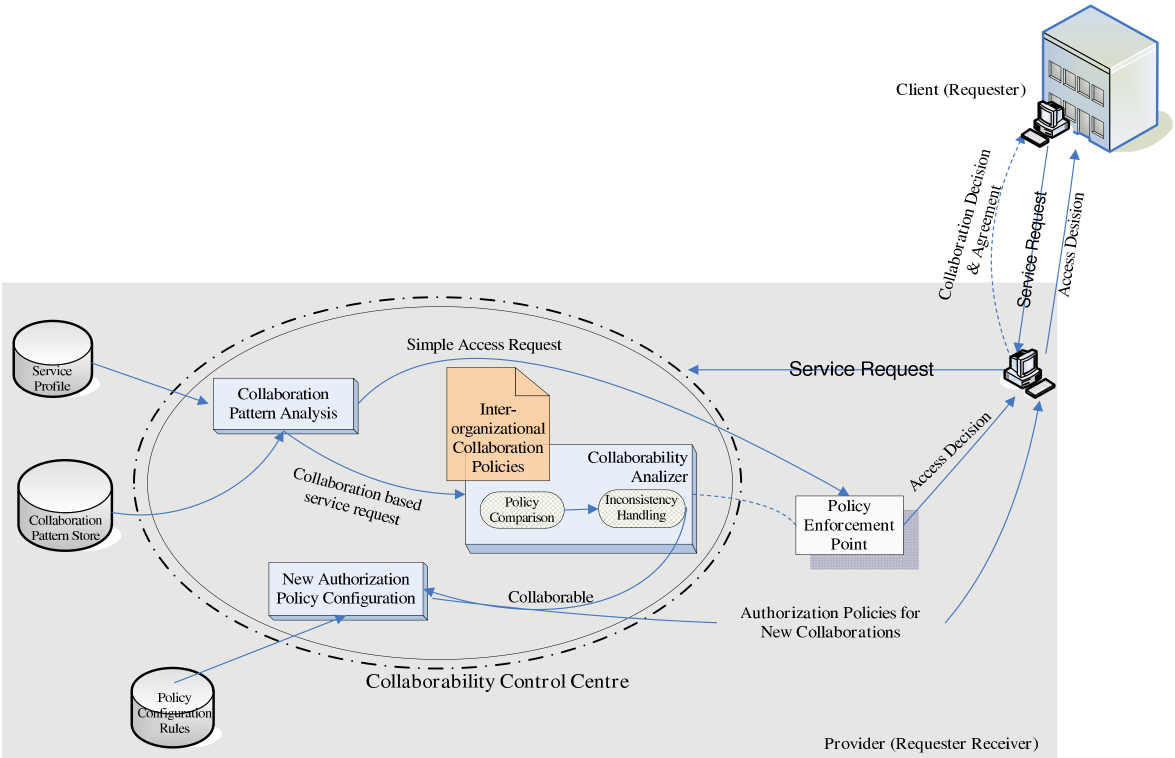
Client (Requester)
Collaboration Decision & Agreement
Service Request
Access Desision
Simple Access Request
Service Request
Service Profile
Collaboration Pattern Analysis
Inter-organizational Collaboration Policies
Collaborability Analizer
Policy Comparison
Inconsistency Handling
Collaboration based service request
Collaboration Pattern Store
Access Decision
Policy Enforcement Point
New Authorization Policy Configuration
Collaborable
Authorization Policies for New Collaborations
Collaborability Control Centre
Policy Configuration Rules
Provider (Requester Receiver)

Figure 2: Policy Driven Authorization Control Framework

collection of predefined policies for each service provided by the request receiver, and these policies are used to govern all the accesses to services provided by the organization, include the established collaborative service.

In the following sections, we shall discuss each functional segments of the **CCC** in detail.

5 Collaboration Pattern Analyzer

The first functional segment in PDAC framework is Collaboration Pattern Analyzer, which analyzes the nature of the service and determine the collaboration pattern of the requested service if collaborations are involved. We first discuss different inter-organizational collaboration patterns.

5.1 Business Collaboration Patterns

Business collaborations consist of complex relationships and interactions among organizations. In the collaborative settings, security considerations should be centered on the relationships between collaborating partners. In this section, we conclude four types of basic collaboration between organizations and provide examples in the Healthcare domain. More complex collaborations in business world can be formed base on different combination of these basic collaboration patterns. In this paper, we will only focus on the basic collaboration patterns. Figure 3 illustrates basic collaboration patterns we have identified. Three types of collaborating roles are involved in collaborations: service owner, service provider and service consumer. It is possible that one organization plays two collaborating roles in a collaboration, e.g. one organization can be both the service provider and the service owner.

Simple Access (SA): SA depicts the most basic scenario in collaborations that involves two organizations. One organization wants to access service that is provided by another organization for self use, and the service does

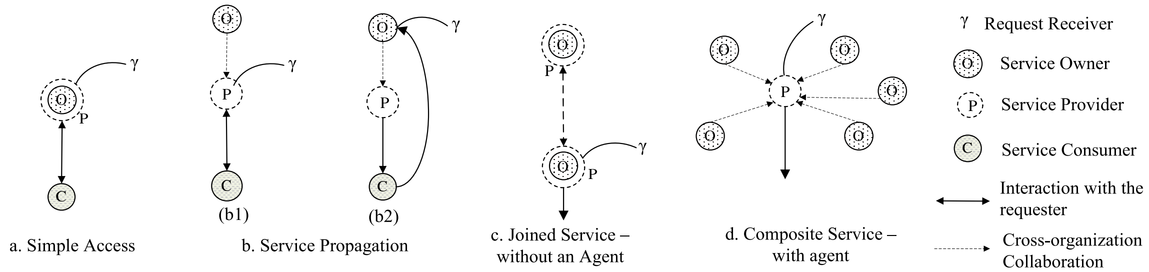

Figure 3: Collaboration Patterns

not involve any collaborations. For example, Staff in a clinic wants to access Online Pharmacy Services provided by a local pharmacy company. Pharmacy company is service provider and service owner in the example. As shown in Figure 3a, **SA** involves one consumer and one service provider – the request receiver, who is also the service owner.

Service Propagation (SP): SP depicts collaboration that involves multiple organizations and *forward* privilege could be passed from one organization to another. This collaboration has privilege propagation. For example, patient is the owner of the patient records who has the ultimate right regards to his own record. Patient can grant access right, i.e., access and forward privileges to his General Practitioner (GP) in a medical center so that the GP can access the patient's health records for diagnosis purpose. If the *forward* privilege is granted to the GP, the GP could forward this access right to a third party based on conditions set by the patient, e.g. staff in hospital emergency room. In this scenario, patient is service owner and GP is the collaborative partner and service propagator who could grant access right to third parties. As shown in Figure 3b, there are different ways that collaboration could be recognized as service propagation.

1. b1: the request receiver is not the service owner, but a provider who provides other party's service;
2. b2: the request receiver is a service owner who does not provide the requested service directly to consumers, another participant is used as service provider.

Apart from **SA** and **SP**, we are also aware of different types of service composition, especially with the emerging Web Services technologies. The Composite Service we discuss here is referring to the service that is based on the integration of multi-service providers. Two different types are identified in service composition:

Joined service without an agent (JS): Two organizations involving in a peer-to-peer collaboration and provide a joined service by integrating their business processes or part of their business processes together to form a new service directly without an agent. For example, heart disease center collaborates with a community service organization to provide a post-treatment care plan to elders. Heart disease center and the community service organization are different business units, but they integrate part of their services together to provide a joined service to patients. As shown in Figure 3c, the requester is a service owner who wants to collaborate with another service owner and service provider – the request receiver.

Composite service with agent (CS): It depicts the scenario that multiple numbers of service providers provide their services through a centralized agent. For example, a Health Insurance Company normally works with different health care providers. The insurance company and each of its health care providers have their own security policies. The insurance company could have several service providers for the same type of health care service and it works as an agent in this situation. However, customers could only access those services through the network of the Health Insurance company. As shown in Figure 3d, the requester is a service owner who wants to collaborate with an agent, who can provide the owner's service to other parties.

CS and **JS** share some common characteristics and present some distinct differences as well. They both involve service composition from multiple organizations. Each participant provides part of the services. However, CS describes the situation that the agent is the center of the collaboration, and all the other participants need to follow rules set up by the agent. Differently, JS describes a peer-to-peer collaboration, in which both participants have equal right in the collaboration, and both parties's rules have to be treated equally.

5.2 Collaboration Pattern Specification

Business collaborations consist of complex relationships and interactions among organizations. In the collaborative settings, security considerations should be centered on the relationships between collaborating partners. The basic collaboration patterns we have identified are focusing on high level business interaction. The real challenges come from the multi-dimensional interactions existing in peer-to-peer based collaboration:

- In the peer setting based collaboration, no organization has total control over all collaborations and interactions, and each organization can only see collaborations that directly relate to itself. Organizations play different roles in a collaboration, they see collaboration pattern from their own perspective, which can be different to other participants in the collaboration. For example, a hospital collaborates with an off-shore institute to provide after-hour interpretation service for radiology images. From the hospital's perspective, this collaboration could be a joined service. However, if the off-shore institute does not perform the interpretation by itself, and it just pass the case to its collaborating radiologists. For this off-shore institute, the collaboration with the hospital is service propagation.
- An organization can play multiple roles in a collaboration. For example, an organization can be the service owner and the service provider at same time in a collaboration. Moreover, if the organization provides several services, it could be the service owner in one service and service provider for another service.

Therefore, it is important for organizations to understand and determine the collaboration pattern from their own perspective according to the roles they play in the collaborations. In addition, a key observation is that, different collaboration patterns result in differences in number of each collaborating role a collaboration has. For example, a joint service can have at least two owners while some simple services have just one service owner.

Our work focuses on business collaboration in the context of Services Computing. Services are inter-organizational services, they could be functions or resources provided by organizations. Organizations can perform a joined service jointly, which is called orchestration in literature; organizations can also collaborate in peer-to-peer setting, which is called choreography in literature. In our studies, we all refer to collaborations. In the following subsection, we define the concepts required by pattern specifications. In Section 5.3, we provide pattern identification method for the requested service.

5.2.1 Definitions

Each participant plays different collaborating roles in a service collaboration. We identify three collaborating roles: service owner, service provider and service consumers:
Service owner: the organization who itself operates the whole service or part of the service.
Service provider: the organization who provides services to other parties while the services are based on inter-organizational collaborations and are completely provided by other organizations. In another words, the service provider does not operate the service at all but provides the service, e.g. car dealer;
Service consumer: the collaboration participant who access the service for self use only.
In order to clearly differentiate roles performed by participants, we made the concept of each collaborating role exclusive to one another. However, the definitions are the intension of the collaborating roles. The extension of each role may not be exclusive to one another. In extension of the above definitions, an organization could play multiple collaborating roles, e.g. a service owner could also act as a service provider and a service consumer could act as a service provider.

Definition 1 Participants and collaborating roles:

- We define participants in a service collaboration as a tuple $\{\mathcal{O}, \mathcal{P}, \mathcal{C}\}$, where $\mathcal{O}$ denotes the set of all service owners, $\mathcal{P}$ denotes the set of all service providers, and $\mathcal{C}$ denotes the set of all service consumers.

For example, $c \in \mathcal{C}$ means the requester is a service consumer. We use $\mathcal{Q}$ to denote the set of all participants so that $\mathcal{O} \subseteq \mathcal{Q}, \mathcal{P} \subseteq \mathcal{Q}, \mathcal{C} \subseteq \mathcal{Q}$.

- We use γ to denote the request receiver, if $\gamma \in \mathcal{P}$, we say the request receiver is the provider of the requested service.
- We use $|\,|$ indicates the size of a role set. For example, if $\mathcal{P} = \{PAC\ Radiologist, CAP\ Radiologist\}$, $|\mathcal{P}| = 2$.
- We use different operators to denote relationships between participant sets. We use $\cap$ to denote 'AND' relationship, $\cup$ to denote 'OR' relationship, $\otimes$ is used to denote 'EXCLUSIVE OR' and $\oplus$ is used to denote the 'OVERLAPPING' relationship

For example, $(\mathcal{O} \oplus \mathcal{P})$ means that service owner can also play the service provider role.

Definition 2 A collaboration pattern is defined as *cop* $\in \{\mathcal{SA}, \mathcal{SP}, \mathcal{JS}, \mathcal{CS}\}$, where $\mathcal{SA}$ refers to Simple Access Pattern; $\mathcal{SP}$ refers to service propagation collaboration pattern; $\mathcal{JS}$ refers to collaboration patterns: Joined Service without Agent and $\mathcal{CS}$ refers to Composite Service with Agent.

Given the concepts and definitions above, we can specify basic collaboration patterns we have concluded. For example, as shown in Figure 1b, there are different ways that collaboration could be recognized as service propagation.

1. The request receiver is not the service owner, but a provider who provides other party's service. We can specify this collaboration pattern as the follows:

$$\gamma \in \mathcal{P};\ |O| = 1;\ |\mathcal{P}| = 1;\ \mathcal{P} \otimes \mathcal{O} \tag{1}$$

2. In another possible **SP**, the request receiver is a service owner who does not provide the requested service directly to consumers, another participant is used as service provider. This could be specified as he follows:

$$\gamma \in \mathcal{O};\ |O| = 1;\ |\mathcal{P}| \geq 1;\ \mathcal{P} \otimes \mathcal{O} \tag{2}$$

5.3 Requested Collaboration Analysis

Another critical issue related to collaboration pattern is how to identify collaboration pattern of the service that is requested by the requester.

Identify collaboration pattern of the requested service is essential for collaborability analyze and configure policies for new collaborations as different collaboration patterns expose different requirements to collaboration partner's authorization policy. It is very important to know information such as: whether the request receiver is the service owner or just a provider who provides other organization's service etc. Only organization itself holds such information about how they organize all the services they claimed. Such information normally can not get from service registries for business confidentiality. However, this information is critical for organizations to handle collaboration requests, establish new collaboration and organize services.

To perform collaboration pattern analysis, the control center in PDAC framework retrieves information about the requested service from a **Service Profile** database, which stores detail information about each service provided by the request receiver. The information include owner of the service, required providers of the service, number of providers required and the collaborating role of the request receiver in the service. The control center maps all the relevant information with predefined collaboration patterns that stores in a database called: **Collaboration Pattern Store**.

We propose a Requested Collaboration Model ($\mathcal{RCM}$) here, which allows us to map all the relevant information with pre-specified collaboration patterns to determine the pattern of the requested collaboration.

Definition 3 Service Profile $\mathcal{SP}ro$ contains information about the requested service that is only known to the organization itself. $\mathcal{SP}ro$ = $(\mathcal{O}_{\mathcal{SP}ro}, \mathcal{P}_{\mathcal{SP}ro}, \gamma)$ where $\mathcal{O}_{\mathcal{SP}ro}$ is service owners in the requested service; $\mathcal{P}_{\mathcal{SP}ro}$ is service providers required in the requested service. For the request receiver γ, $\gamma \in \mathcal{O}$ means the request receiver is the service owner.

Definition 4 A request that is received from a requester is defined as *req* = $(\mathcal{S}, \mathcal{CRE}, \mathcal{OP})$, where $\mathcal{S}$ is the service the requester intends to perform operations on. $\mathcal{CRE}$ is a set of credentials of the requester. $\mathcal{OP}$ is a set of operations that the requester wants to perform on the requested service. For example, *req* = $(Claim\ insurance, (Policy\ number), (lodge))$ states that the requester wants to lodge an insurance claim, and he has a policy number as credential.

Definition 5 We define set of service owners, providers and consumers in the requested collaboration service as $\mathcal{O}_{\mathcal{RCM}}$ and $\mathcal{P}_{\mathcal{RCM}}$. This information come from the existing service profile for the requested service.

Definition 6 Requested Collaboration Model $\mathcal{RCM}$ is a collection of information that is relevant to the requested collaboration. $\mathcal{RCM}$ = $(\gamma, \mathcal{O}_{\mathcal{RCM}}, \mathcal{P}_{\mathcal{RCM}}, |\mathcal{O}_{\mathcal{RCM}}|, |\mathcal{P}_{\mathcal{RCM}}|, \mathcal{REL}(\mathcal{O}_{\mathcal{RCM}}, \mathcal{P}_{\mathcal{RCM}}))$.

- $|\mathcal{O}_{\mathcal{RCM}}|$, $|\mathcal{P}_{\mathcal{RCM}}|$ are the multiplicity of the owner and the provider set of the intended collaboration.
- $\mathcal{REL}(set1, set2)$ is a comparison function that is used to check relationships between two participating role sets in the requested collaboration so that we can map with collaborating role relationships in the pre-specified collaboration patterns. For example, if $\mathcal{O}_{\mathcal{RCM}} \cap \mathcal{P}_{\mathcal{RCM}} = \emptyset$, then $\mathcal{REL}(\mathcal{O}_{\mathcal{RCM}}, \mathcal{P}_{\mathcal{RCM}})$ is 'EXCLUSIVE OR'; if the result is an empty set, then $\mathcal{REL}(\mathcal{O}_{\mathcal{RCM}}, \mathcal{P}_{\mathcal{RCM}})$ is 'OVERLAPPING'.

$\mathcal{RCM}$ contains all the relevant information that is required for pattern analysis. The query starts from the role of γ, multiplicity of the service owner and the service provider, and then collaborating role relationships as the last. Once the query finds matching conditions in the pre-specified collaboration pattern database, the pattern of the requested service can be determined.

6 Collaborability Analyzer

Each organization may define its own set of authorization policies for the specific service within the organization. If these individual policies are in conflict with each other, the collaboration is unlikely to be achieved. Therefore, before collaboration can be established, the prospective partner's authorization policies need to be analyzed and compared. Furthermore, as we mentioned before, there are different patterns of collaboration, each pattern has its own requirements on collaboration partner's authorization policy. These requirements are called **"Collaboration Policy"** in our work. Collaboration Policy determines whether prospective partners are able to collaborate securely for desired collaboration pattern based on their authorization policies. The main function of collaborability analyzer is to analysis and compare collaboration partner's authorization policies with the policies of the request receiver for the requested collaboration pattern and make collaboration decision according to the corresponding collaboration policies. In the following subsection, we introduce collaboration policy in more detail.

6.1 Collaboration Policy

Collaboration Policy based on collaboration patterns and focus on relationships between individual authorization policies of collaboration participants. Therefore, we first introduce an authorization policy model for individual service. Our previous work [12] provided a theoretical authorization policy model which is based on the Core RBAC [29]. We recall the model here briefly to assist discussions on Collaboration Policies.

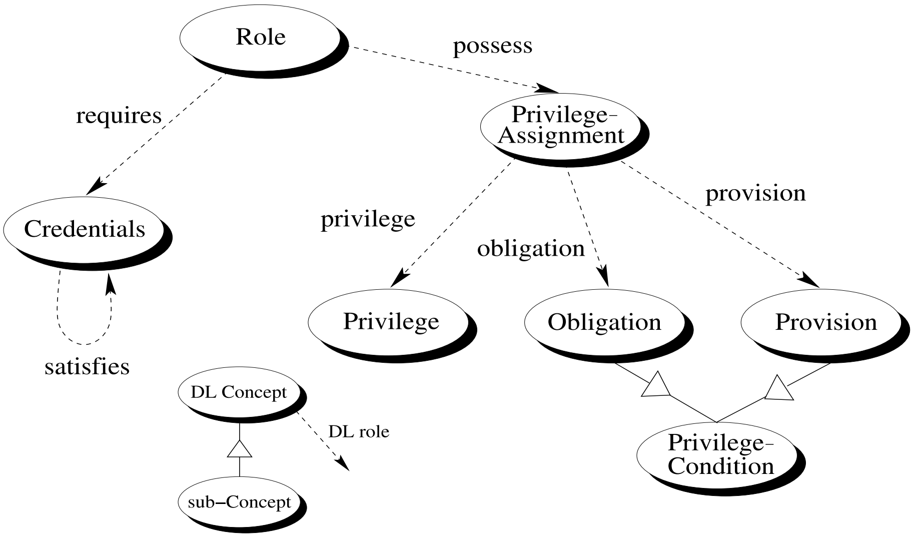

Figure 4: Authorization Policy Model

6.1.1 Authorization Policy Model

Each service has a policy describing the roles that eligible to access the service and the privileges assigned to each role. Figure 4 depicts the policy model. The main elements in the model are roles, privileges, credentials, provisions and obligations.

- Role: A role is a job function within the context of an organization with some associated semantics regarding the authority and responsibility conferred on the user assigned to the role (e.g. ‘Doctor’, ‘Nurse’) [29].
- Credential: Credentials are signed assertions describing attributes of the owner [34]. Examples of credentials are digital certificate, identity number etc. A user needs to disclose certain credentials to be able to carry a ‘role’.
- Privilege: Privilege is an approval to perform an operation on a service or an object and it is assigned to roles. Examples of privileges could be ‘forward’, ‘access’, ‘delete’ for the specific service.
- Privilege Condition: Conditions are special agreements associated with required privileges. When the system makes decisions about authorization, it not only checks credential of requester, but also conditions that need to be met before access could be granted. There are two types of conditions: provision and obligation. Provisions are conditions that need to be satisfied before authorization is granted; obligations are conditions that must be fulfilled after the decision [3], such as writing to a log file. Hence, provisions further constrain the right to access a privilege, such as restricting the time a privilege is access. Not all privileges have conditions. E.g. ‘Forward’ on Patient Records might associate with some additional conditions as its sensitivity and privacy reason, while ‘access’ may not have any conditions associated.

The users of the service are not defined and may be, for example, humans or computer agents. Users are not statically assigned to roles; rather credentials are used to authenticate the holder as authorized to act in a particular role. Each role is assigned a number of privileges, obligation and provision conditions are attached to each assignment of a privilege to a role. The only relations directly supported in a DL are binary roles; hence, the ternary relation between privileges, obligations and provisions in the policy model needs to be represented by a concept [12]. The concept *PrivilegeAssignment* is introduced to represents the grouping and relations *privilege*, *obligation* and *provision* to link a *PrivilegeAssignment* to its associated privilege, obligation and provision. Obligation and Provision are themselves subconcepts of Privilege Condition.

Base on this model, we can analize and specify requirements on prospective partner's authorization policy. Collaboration policies are also be developed upon this model.

6.1.2 Collaboration Policies for Different Collaborations

Collaboration Policy outlines requirements for collaboration partner's authorization policy, the requirements are based on policy comparison between the collaboration participants. Collaboration Policies for each collaboration pattern are discussed in the follows:

1. **Simple Access (SA):**
   For simple access that only involves one service requester and one service owner, the focus is to satisfy related authorization policies of service owner, and the properties of the requester must match requirements of service owner. If the requested service is SA, the request can be sent to PEP for the second phase of authorization control straight away.
2. **Service Propagation (SP):**
   Service Propagation normally requires authorization policies of collaborative partner comply with ones specified by the service owner. Let us take example in the previous section and consider the patient health record as a service. For privacy concern, if the patient does not want doctors to forward his health records to others except to staff in emergency room of specific hospitals. However, under the authorization policy of the medical center, doctors have the right to forward patient health records to other partner institutes and labs as well. In this case, collaboration is unlikely to be successful. The doctors could forward patient records to parties that are not allowed by the patient once they gained access right. This is not acceptable from the patient's view. The propagation nature of **SP** implies a 'forwarding' behavior that is performed by collaborative partner.Unwanted 'forwarding' could happen if the partner has authorization policy that is ***looser*** than the owner's. Therefore, policies of the collaboration partner should comply with or stricter than ones of the service owner.

   There are three typical ways to have one policy looser than the other:

   - One set of policies have more roles to access the same service than the other set of policies;
   - One set of policies have less credentials required for an equivalent role than the other set;
   - For an equivalent role, more privileges are assigned in one set of policies than the other set.

   Based on the above discussion, in **SP**, the Collaboration Policy for **SP** should be:

   - All roles are specified by the collaboration partner (the service propagator) should be included in policy of the service owner for the requested service.
   - For the comparable roles, credentials required by the service owner should all be included in credential requirements of the prospect partner. Thus, service requester who is accepted by the partner will meet requirements of the service owner accordingly.
   - Privileges of the collaboration partner should all be included in privileges specified by the service owner for the comparable roles so that the partner will not perform unwanted actions on the collaborated service from the service owner's point of view.

   We have concluded two types of service composition in the previous section, **CS** and **JS**. Composite service we discuss here is referring to the service that is based on the integration of services that is provided by multiple service providers. The integration nature of service composition also requires certain level of policy integration. Therefore, service composition requires collaboration partners have at least 'integratable' authorization policies. Collaboration Policies for two types of service composition are discussed in the following:
3. **Joined service without an agent (JS)**:
   JS implies that both parties in the joined service collaboration have equal position since both parties are providers of a part of the joined service. None of the parties possesses dominating position in the joined

service. Thus, authorization policies of both parties should comply with each other instead of comply with any one of the parties. However, the nature of the collaboration is 'integration', we can consider it as 'integratable' as long as both parties have some common properties. This particularly necessary for roles specified by different collaboration partner when collaboration participants providing different categories of services. Roles specified in the heart disease clinic (example in section 2.2) could be very different than roles specified by the community service center, but the collaboration could be established base on common roles they have. Thus, the Collaboration Policy for **JS** is:

- For the requested service, some of the roles specified in both parties should be comparable to each other.
- For the comparable roles, credential required in both collaboration participants should be equivalent to each other.
- Same to the credentials, the privileges of the comparable roles in both parties should be equivalent.

4. **Composite service with agent (CS)**:
CS depicts another types of service composition, it also requires 'integratable' authorization policies from collaboration partners. However, different than **JS**, the agent is the center of the collaboration, hence the agent has greater control power over the collaboration and all other participants need to follow rules set up by the agent. Let us take the HCF example we mentioned in the previous section. HCF and each of its health care providers have their own security policies. Authorization policies of the actual service providers have to comply with the agent if they want to become members of HCF and get patients through HCF's network. This is different than **SP** as well, in **SP**, the service owner is in the dominate position. We use "*Owner*" to represent the service owner and "*Provider*" represent the agent in the collaboration. Then the Collaboration Policy for **CS** is:

    - Roles specified by the collaboration partner should have some comparable roles in the policy of the agent.
    - For the comparable roles, credentials required by the agent should all be included in credential requirements of the service owners.
    - Privileges of the collaboration partner should all be included in privileges specified by the agent for the comparable roles.

Collaboration Policy also outlines which possible policy inconsistencies are acceptable for the intended collaborations, and which ones are not. The comparison function in the collaborability analyzer detects inconsistencies between polices from different participants and handle the inconsistencies according to relevant collaboration policy. In the following subsection, we introduce the common authorization policy inconsistencies.

6.2 Authorization Policy Inconsistencies

It is unusual that two organizations have exactly matching authorization policies. In other words, inconsistencies are rifeness. However, in inter-organizational collaboration, we are not looking for partners who have exactly matching policies. Certain forms of inconsistencies are acceptable and some of them are negotiable depend on the collaboration pattern. Authorization Policy inconsistencies between distinct organizations could occur in different ways:

- Different authorization control system. Organizations may using heterogeneous system, e.g., RBAC based system, Matrix based system etc.
- Different specification vocabularies. Organizations may specify their policies by using different words, e.g., one organization may use "Doctor", while another may use "Practitioner".
- Inconsistent authorization policies. Policies specified by different organization may have various conflicts.

Our policy model is based on RBAC as it is the most appropriate approach for Web Services environment and it is easy to migrate from other system. Vocabulary problems have been addressed in [6], XML based approaches have been proposed to provide syntactic tags for policy specification and support interpretation. We are focusing on policy inconsistencies.

Based on the authorization policy model we introduced in the Section 3, let us assume two organizations, A and B, requiring a collaboration, where both organizations have an authorization control policy, P_A for A and P_B for B. Possible inconsistencies between two set of policies could be identified and categorized in the following. Some of the inconsistencies have been encoded in DL in our previous work [12]. However, the model has limitations, it does not cover all the issues we found.

1. **Role Inconsistencies**
   Missing Role (MR): Missing Role Node, represents the roles in P_A that have no comparable role in P_B. If P_A has roles with no equivalent in P_B, then this inconsistency indicates that, for at least those roles, P_A admits privileges that P_B does not.
   Single vs. Group Roles (SGR): another possible inconsistency between P_A and P_B is present when a group of roles in one that all comparable to one single role in another. For example, “Doctor” role in one organization might have several comparable roles in the other organization: physician, psychologist etc.

2. **Credential Inconsistencies**
   There is an inconsistency in credentials when roles judged as comparable have different credential requirements. Such an inconsistency means that equivalent roles in the two organizations have access to similar privileges but with a less stringent authorization requirement in one organization. This inconsistency can be manifest in two ways:
   Absent Credential (AC): Absent Credential between P_A and P_B is present when there is a comparable role between P_A and P_B, but the credentials required for this role in one that have no comparable credentials required in the other.
   Credential Substitution (CT): Credential Substitution are present between P_A and P_B when credential required for a comparable role in one has another acceptable substitution in the other. For example, credit card number is required in one organization while in the other organization, either credit card number **or** bank card number are acceptable.

3. **Privilege Inconsistencies**
   Privilege Inconsistencies are present between P_A and P_B when comparable roles in one have no equivalent privileges in the other and, for comparable roles and equivalent privileges, one party has weaker conditions than the other. The possible cases could be:
   Missing Privilege (MP): represents the privileges allowed in P_A that have no equivalent privileges in P_B for the comparable role.
   Missing Condition (MC): represents the conditions required in P_A that have no comparable conditions in P_B for the comparable role and for the equivalent privilege.
   Weaker Condition (WC): represents the contents of the comparable conditions required in P_A has weaker requirements than conditions required in P_B for the comparable role and for the equivalent privilege.

Above policy inconsistencies could cause potential security breach and conflicts during collaboration. However, not all inconsistencies lead to a ‘rejection’. We classify three instances for inconsistencies here:
Acceptable: it applies to inconsistencies that do not require negotiation. The prospect partner is possible for the desired collaboration even if acceptable inconsistencies exist.
Negotiable: it applies to inconsistencies that can not be accepted immediately but do not lead to immediately rejection. Further negotiation is required to make decision on these inconsistencies.
Not Acceptable: it applies to inconsistencies that lead to immediate rejection. Prospect partners who have these inconsistencies will not be considered for the desired collaboration.

According to the Collaboration Policies, solutions to possible authorization policy inconsistencies are outlined in Table 1. For example, the $\surd$ in the junction of ‘O(**SP**)’and ‘MR’ means that ‘Missing Role’inconsistency is acceptable if it exists in the service owner’s authorization policy when compare to another party’s. All condition related inconsistencies are negotiable for all the patterns because there are many conditions actually include

	Owner(**SP**)	Propagator(**SP**)	Both Parties(**JS**)	Owner(**CS**)	Agent(**CS**)
MR	√	×	△	△	△
SGR	√	×	×	×	√
AC	×	√	×	√	×
CT	×	√	×	√	×
MP	√	×	×	×	√
MC	△	△	△	△	△
WC	△	△	△	△	△

√: Acceptable ×: Not Acceptable △: Negotiable

Table 1: Policy Inconsistency Solutions

many forms of legal documents and agreements. It is infeasible to make decisions without study the detail contents of the conditions thoroughly. Therefore, negotiations are required.

6.3 Authorization Policy Comparison

Policy comparison is based on the authorization policy model we mentioned in Section 5.1.1. The definitions and detail works are presented in [12]. We briefly introduce here for completeness.

We first encode the proposed authorization policy model in $\mathcal{SROIQ}$ and then show how two policies can be evaluated by comparing them in the model. We encode the inconsistency tests as concepts and relations in our authorization policy model. Individual policies expressed using the model can then be compared and tested. Given two set of policies, with the roles and privileges of the two organizations suitably related, a reasoner will prove that the tests are either satisfiable or unsatisfiable. The satisfiable results means the two set of policies have certain forms of inconsistencies as the reasoner detects inconsistencies. The results of the tests can then be analyzed to check whether they satisfy the requirements for the particular collaboration according to the corresponding collaboration policies. Since the tests are part of the general model they are generic, meaning they can be expressed once, proven to encode the required meaning and used to testing any two policies. Because Description Logics are decidable, a reasoner for $\mathcal{SROIQ}$ will always terminate with the proofs, no matter how complex the policies.

Assume two organizations, A and B, requiring a collaboration, where both organizations have an authorization control policy, P_A for A and P_B for B, encoded in the proposed authorization policy model (or defined in some other way and translated into the model). For consistency and collaboration checking, P_A and P_B must first be combined into a single model and then checked. The checking process start from the role assignments of the two set of policies. A comparability relation between the roles in the two organizations is defined next. Then, for each comparable role, the model checks credential requirements for the role of the two set of policies. Privilege assignments checking follows the similar process. For illustration purpose, we recall an example here, which we discussed in more detail in [12]. The following example demonstrates how an authorization control policy can be presented in the DL policy model and how the inconsistencies between the authorization control policies of collaboration partners can be evaluated using a DL reasoner.

Assume a medical center that requires a collaboration with a pathology center. The two will collaborate on the tests of patients, patient records and the test results. The centre will search for a medical service registry for potential pathology collaborators and evaluate their authorization policies to find a suitable partner. The policies of the medical centre and two candidates are shown below.

1. Medical Clinic:
    - Attending doctors have the privilege to access and forward patient information;
    - a provision is attach to the forward privilege: recipients must be a doctor in the chosen pathology institute.

A policy can be defined with individuals classified as follows:
$Role(mc\_doctor)$, $Credentials(mc\_doctor\_id)$, $PrivilegeAssignment(mc\_pa)$, $Privilege(access)$, $Privilege(mc\_forward)$, $Provision(to\_partner\_pathology\_doctor)$ and $Obligation(no\_obligation)$.
With $(mc\_doctor, mc\_doctor\_id) \in requires$, $(mc\_doctor, mc\_pa) \in possess$, $(mc\_pa, mc\_forward) \in privilege$, $(mc\_pa, no\_obligation) \in obligation$, $(mc\_pa, to\_partner\_pathology\_doctor) \in provision$
and a privilege assignment for *access*.

2. Pathology institute X:

   - Attending doctors have privilege forward patient information;
   - a provision attach to forward privilege: recipients must be doctors in X

   The policy definitions are as follows:
   $Role(doctor_X)$, $Credentials(doctor\_and\_path\_id)$, $PrivilegeAssignment(pa_X)$, $Privilege(forward_X)$, $Provision(to\_X\_pathology\_doctor)$ and $Obligation(no\_obligation_X)$.
   With $(doctor_X, doctor\_and\_path\_id) \in requires$, $(doctor_X, pa_X) \in possess$, $(pa_X, forward_X) \in privilege$, $(pa_X, no\_obligation_X) \in obligation$, $(pa_X, to\_X\_pathology\_doctor) \in provision$.

3. Pathology institute Y:

   - Attending doctors have privilege to forward patient information;
   - a provision is attach to forward privilege: recipient must be either a doctor in Y or staff in a collaborating research institute.

   The policy definitions are similar to above, with the important provision
   $Provision(to\_Y\_pathology\_doctor\_or\_research)$.

To find a suitable collaboration partner, the medical center needs evaluate its own policy with pathology institutes X and Y. The policies are tested in the combined model; one test with institute X and one with Y. Assume in both cases that the medical center is policy A and that each pathology institute is policy B. The relationships for $role\_comp_{AB}$ ($role\_comp_{AB}$ reflects the notion that the correspondence between roles of the two set of policies are almost be an equivalence) and others are straightforward, importantly provision $to\_X\_pathology\_doctor$ is more restrictive than $to\_partner\_pathology\_doctor$, hence $(to\_X\_pathology\_doctor, to\_partner\_pathology\_doctor) \in prov\_order$ (if $(x,y) \in prov\_order$ then x is a stronger provision than y), while $to\_Y\_pathology\_doctor\_or\_research$ is less restrictive than the provision in the medical center and so is not related by $prov\_order$.

In the comparison with institute X the reasoner proves the required concepts as unsatisfiable, and thus shows that Pathology institute X is a suitable partner for collaboration. However, in testing with institute Y, the reasoner proves that concept $rcomp\_pa\_nord_{BA}$ is satisfiable (because Pathology institute Y allows extra forwarding privileges), and thus shows that Pathology institute Y is not a suitable collaboration partner. $rcomp\_pa\_nord_{BA}$ satisfiable reflects the notion that for comparable roles, P_A has more stringent constraints than P_B.

6.4 Collaborative Policy Configurator

Each organization can specify its own set of authorization policies to manage the accesses to its resources. However, in business collaboration we need to analize all the relevant authorization policies from the participating organizations and configure authorization policy to govern the accesses for each collaborations that dynamically established. Configuration rules are based on the principles discussed in the **Collaboration Policies**. We analyzed control power of different collaborating parties in the light of different collaboration patterns and defined policy configuration rules accordingly [13]. Collaborative policy configuration is a process of combining and consolidating policies. If the prospect partner is "Collaborable" (the result from the Collaborability Analyzer), the **Collaborative Policy Configurator** can configure authorization policies for the established collaborations, and the configured authorization policies will be sent to PEP to serve the second phase of authorization control.

7 Conclusion

Challenging security and authorization control issues arise in business collaboration. Current technical solutions focus mostly on authorization control in a single organization or a single system, which are not suitable for dealing with the authorization control issues in complex inter-organizational collaboration.

In this paper we proposed **Policy Driven Authorization Control** framework for authorization controls in inter-organizational collaboration environments. Based on the framework, different inter-organizational collaboration patterns can be specified and identified. Prospective collaboration partners can be evaluated based on the corresponding Collaboration Policies, and possible policy inconsistencies can be detected and handled according to the intended collaboration pattern. Several important concepts have been introduced in this work: **Collaboration Pattern** and **Collaboration Policy**. This work is focused on high level business interactions and collaborations. We believe it is important for authorization control in service collaboration setting to take the effects of high level business requirements and individual party's policy into consideration.

References

[1] E. Bertino, J. Cramption, and F. Paci. Access control and authorization constraints for ws-bpel. In IEEE International Conference on Web Services (ICWS), pp. 275–284, Chicago, Illinois, USA, 2006.

[2] E. Bertino, A. C. Squicciarini, and D. Mevi. A fine-grained access control model for web services. In IEEE International Conference on Services Computing, pp. 33–40, Shanghai, China, September 2004.

[3] Claudio Bettini, Sushil Jajodia, X. Sean Wang, and Duminda Wijesekera. Provisions and obligations in policy management and security applications. In 28th International Conference on Very Large Data Bases (VLDB), Hong Kong, 2002.

[4] R. Bhatti, E. Bertino, and A. Ghafoor. A trust-based context-aware access control model for web-services. In IEEE International Conference on Web Services, pp. 184–191, San Diego, CA, June 2004.

[5] R. Bhatti, E. Bertino, A. Ghafoor, and J. Joshi. Xml-based specification for web services document security. IEEE Computer, vol. 37, no. 4, pp. 41–49, 2004.

[6] R. Bhatti, J. Joshi, E. Bertino, and A. Ghafoor. Access control in dynamic xml-based web-services with x-rbac. In International Conference on Web Services, pp. 243–249, Las Vegas, Nevada, USA, June 2003.

[7] P. A. Bonatti and F. Mogavero. Comparing rule-based policies. In Proceedings of 9th IEEE International Workshop on Policies for Distributed Systems and Networks, pp. 11–18, New York, USA, 2008.

[8] Jung-Hwa Chae and Nematollaah Shiri. Description logic framework for access control and security in object-oriented systems. In RSFDGrC, pp. 565–573, 2007.

[9] P. Chapin, C. Skalka, and X. S. Wang. Authorization in trust management: Features and foundations. ACM Computing Surveys, 2008.

[10] Y. Demchenko, L. Gommans, and C. D. Laat. Using saml and xacml for complex resource provisioning in grid based applications. In 8th IEEE International Workshop on Policies for Distributed Systems and Networks, pp. 183–187, Bologna, Italy, June 2007.

[11] H. Grain. E-consent design and implementation issues for health information managers. Health Information Management, vol. 33, no. 3, pp. 84–88, 2004.

[12] D. D. He, M. Compton, K. Taylor, and J. Yang. Access control: What is required in business collaboration? In 2009 Australian Database Conference (ADC 2009), Wellington, Newzealand, January 2009.

[13] D. D. He and J. Yang. A policy driven authorization control framework for business collaboration. In IEEE SCW, pp. 17–24, Salt Lake City, Utah, USA, July 2007.

[14] D. D. He and J. Yang. Security policy specification and integration in business collaboration. In 2007 IEEE International Conference on Services Computing (SCC 2007), pp. 20–27, Salt Lake City, Utah, USA, July 2007.

[15] A. R. Hevner, S. T. March, J. Park, and S. Ram. Design science in information systems research. MIS Quarterly, vol. 28, no. 1, pp. 75–105, 2004.

[16] S. Indrakanti, V. Varadharajan, and M. Hitchens. Authorization service for web services and its application in a health care domain. International Journal of Web Services Research, vol. 2, no. 4, pp. 94–119, 2005.

[17] L. Kagal, M. Paolucci, N. Srinivasan, K. Sycara, and G.Denker. Authorization and privacy for semantic web services. IEEE Intelligent Systems, vol. 19, no. 4, pp. 50–56, 2004.

[18] A. A. E. Kalam, R. E. Baide, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte, A. Miege, C. Saurel, and G. Trouessin. Organization based access control. In IEEE 4th International Workshop on Policies for Distributed Systems and Networks, pp. 120–131, Villa Olmo, Lake Como, Italy, 2003.

[19] H. K. Kim, R. Y. Lee, and H. S. Yang. Frameworks for secured business process management systems. In 4th International Conference on Software Engineering Research, Management and Applications (SERA'06), pp. 57–65, Seattle, Washington, USA, 2006.

[20] A. Laroia. Web services and xml: Leveraging web services to connect the healthcare enterprise. (2002, April) EbizQ. [Online]. Available: http://www.ebizq.net/topics/webservices/features/1546.html.

[21] D. Liu, S. Nepal, D. Moreland, S. Chen, C. Wang, and J. Zic. Secure and conditional resource coordination for successful collaborations. In Proceedings of International Conference on Collaborative Computing, Florida, USA, 2008.

[22] P. Liu and Z. Chen. An access control model for web services in business process. In IEEE/WIC/ACM International Conference on Web Intelligence, pp. 292–298, Beijing, China, 2004.

[23] Saravanan Muthaiyah and Larry Kerschberg. Dynamic integration and semantic security policy ontology mapping for semantic web services (sws). In ICDIM, pp. 116–120, 2006.

[24] OASIS. (2005) Ws-secureconversation 1.3. [Online]. Available: http://www.oasisopen.org/ws-sx/ws-secure conversation/200512/ws-secureconversation-1.3-os.doc.

[25] OASIS. (2006) Security assertion markup language (saml). [Online]. Available: http://saml.xml.org/saml-specific ations.

[26] OASIS. (2006) Ws-security core specification 1.1. [Online]. Available: http://www.oasisopen.org/committees/download.php/16790/wss-v1.1-os-SOAPMessageSecurity.pdf.

[27] OASIS. (2006) Ws-trust 1.3. [Online]. Available: http://docs.oasis-open.org/ws-sx/ws-trust/200512.

[28] Clarke R. A critical element of trust in e-business. In 15th Bled Electronic Commerce Conference, Bled, Slovenia, 2002.

[29] R. S. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman. Role-based access control models. IEEE Computer, vol. 29, no. 2, pp. 38–47, 1996.

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

[31] Brian Shields and Owen Molloy. Using description logic and rules to determine xml access control. In DEXA Workshops, pp. 718–724, 2007.

[32] R. Simon and M. E. Zurko. Separation of duty in role-based environments. In 10th Computer Security Foundations Workshop (CSFW'97), p. 183, Rockport, Massachusetts, USA, 1997.

[33] E. G. Sirer and K. Wang. An access control language for web services. In SACMAT, pp. 23–30, 2002.

[34] H. Skogsrud, B. Benatallah, and F. Casati. Trust-serv: Model-driven lifecycle management of trust negotiation policies for web services. In 13th World Wide Web Conf. (WWW 2004), New York, USA, May 2004.

[35] M. Srivatsa, A. Iyengar, T. Mikalsen, I. Rouvellou, and J. Yin. An access control system for web service compositions. In Proceedings of International Conference on Web Services, Salt Lake City, UT, July 2007. IEEE.

[36] R. K. Thomas and R. S. Sandhu. Task-based authorization controls (tbac): A family of models for active and enterprise-oriented authorization management. In 11th International Conference on Database Security, pp. 166–181, California, USA, 1997.

[37] W. Tolone, G. J. Ahn, T. Pai, and S. P. Hong. Access control in collaborative systems. ACM Computing Surveys, 2005.

[38] W3C. (2006) Web services policy 1.2 - framework(ws-policy). [Online]. Available: http://www.w3.org /Submission /WS-Policy/.

[39] H. Wang, S. Jha, M. Livny, and P. D. McDaniel. Security policy reconciliation in distributed computing environments. In Proceedings of 5th IEEE International Workshop on Policies for Distributed Systems and Networks, pp. 137–147, 2004.

[40] J. Wang. A web services secure conversation establishment protocol based on forwarded trust. In International Conferences on Web Services (ICWS), pp. 569–576, Chicago, Illinois, USA, 2006.

[41] K. T. Win, P. Croll, and J. Cooper. Privacy, confidentiality and consent of electronic health record systems. In HIC 2003 RACGP12CC, pp. 65–71, Australia, 2003.

[42] S. S. Yau and Z. Chen. Security policy integration and conflict reconciliation for collaborations among organizations in ubiquitous computing environments. In Proceedings of the 5th International Conference on Ubiquitous Intelligence and Computing, Oslo, Norway, 2008.

[43] X. Zhang, M. Nakae, M. J. Covinton, and R. S. Sandhu. Toward a usage-based security framework for cllaborative computing systems. ACM Transactions on Information and System Security (TISSEC), 2008.