Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Minimal role mining method for Web service composition

Chao HUANG†, Jian-ling SUN†‡, Xin-yu WANG, Yuan-jie SI

(*Department of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China*)

†E-mail: {hch, sunjl}@zju.edu.cn

Received Apr. 2, 2009; Revision accepted July 29, 2009; Crosschecked Apr. 5, 2010

Abstract: Web service composition is a low cost and efficient way to leverage the existing resource and implementation. In current Web service composition implementations, the issue of how to define the role for a new composite Web service has been little addressed. Adjusting the access control policy for a new composite Web service always causes substantial administration overhead from the security administrator. Furthermore, the distributed nature of Web service based applications makes traditional role mining methods obsolete. In this paper, we analyze the minimal role mining problem for Web service composition, and prove that this problem is NP-complete. We propose a sub-optimal greedy algorithm based on the analysis of necessary role mapping for interoperation across multiple domains. Simulation shows the effectiveness of our algorithm, and compared to the existing methods, our algorithm has significant performance advantages. We also demonstrate the practical application of our method in a real agent based Web service system. The results show that our method could find the minimal role mapping efficiently.

Key words: Web service composition, Role base access control (RBAC), Role mining, Access control policy, Role mapping, Web service security

doi:10.1631/jzus.C0910186 **Document code:** A **CLC number:** TP309

1 Introduction

Web service, which is based on the infrastructure of three major standards—the simple object access protocol (SOAP), the Web service definition language (WSDL), and universal description discovery and integration (UDDI), has been widely adopted by financial enterprises to build up the IT systems (Dustdar and Schreiner, 2005; Eid *et al.*, 2008). However, a single Web service may not satisfy changing system requirements in dynamic systems, such as the multi-agent system (MAS) (Sycara *et al.*, 2003; Talib *et al.*, 2006). This creates a need for automated Web service composition that enables the construction of a powerful, robust service network by integrating a number of collaborated agent-based Web services. Assume that there are three domains, D1, D2, and D3, in the foreign exchange order MAS. Web service S1 hosted in D1 provides the real time rates for the currency pair; service S2 in D2 accepts the foreign exchange order and makes the trade; service S3 in D3 records the trade order and generates the report. In such a case, a service that takes the given currency pair and accomplishes the trade with the latest market rate is not available. However, through Web service composition, S1, S2, and S3 can be composed into a new service CS which accepts the currency pair as input and accomplishes the deal. Although such Web service composition provides a cheap, effective, and efficient means for application integration over existing resources, all the benefits can be obtained only after the access control policy is set up properly.

The role base access control (RBAC) model, proposed by Ferriaiolo and Sandhu in the 1990s, has been used widely as a powerful way to satisfy the access control needs of Web service (Ferraiolo *et al.*, 2001; Essmayr *et al.*, 2004; Carminati *et al.*, 2005; Li and Tripunitara, 2006). RBAC96 is currently the most widely used access control model in enterprises because of its fine grained control over the privilege (Li *et al.*, 2007). Furthermore, using RBAC we can model

‡ Corresponding author

a wide range of access control policies including discretional access control (DAC) and mandatory access control (MAC) (Ferraiolo *et al.*, 2001). RBAC is widely accepted as a best practice and implemented in various systems such as the Microsoft active directory, SELinux, FreeBSD, Solaris, and Oracle DBMS (Park *et al.*, 2001; Ferraiolo *et al.*, 2003). An access control policy is a statement that specifies the rules about how to setup the process for granting or denying authorizations to the users (Schaad *et al.*, 2001).

With the increasing complexity of Web service based applications, more and more distributed Web services need to do the composition (Lécué *et al.*, 2008). The access control policy for the new composite Web service has to be updated accordingly (Ko *et al.*, 2008). To the best of our knowledge, there is little work addressing which kind of role should be defined for the access control policy to access the composite Web service CS. Role mapping has proven to be an effective and efficient way to handle the interoperation. Through setting up role mapping among the domains, the interoperation across the multiple Web services hosted in different domains will be achieved. However, how to find the proper roles for the composite Web service to set up the role mapping is still an untouched area. This is a new form of role engineering. The concept of 'traditional role engineering' given in Coyne (1996) remains in force: "the process of defining a good, complete, and accurate role system is called role engineering". Atluri (2008) pointed out that role engineering is the most important and expensive step when implementing the RBAC model, which is also true for role engineering in Web service composition. The role engineering methodologies fall into two categories: bottom-up and top-down (Goncalves and Poniszewska, 2008). Top-down role engineering is the requirement engineering in nature. By analysis of the system requirement, role information attached to the function process can be delivered. It is necessary to make a detailed analysis of each function process, in which permission information is embedded. Therefore, top-down role engineering is usually taken along with requirement engineering. Neumann and Strembeck (2002) introduced a scenario-driven role engineering method, in which the 'scenario' is regarded as the core of role engineering, permission and role definition can be inferred via scenario evolution, and the directive process framework is also given. In contrast to the top-down method, the bottom-up method is based on the assumption that, there is valuable access control information embedded in the existing user-permission matrix and the information can be partly or totally retrieved using appropriate data mining methods, so the bottom-up role engineering method is also called 'role mining'. Vaidya *et al.* (2007) formally defined the role mining problem (RMP), and discussed several extensions based on RMP. They proved that such problems are NP-complete and the solution can be made with data mining algorithms. Colantonio *et al.* (2008) presented a cost-driven role mining method, pointing out that it lacks role quality measure metrics; thus, they proposed the cost-based measure method, and also introduced the associate rule-based role mining algorithm to find the lowest cost candidate role set. Schlegelmilch and Steffens (2005) gave another role mining algorithm based on cluster analysis, utilizing the role mining tool ORCA. Adjustments to mining rules can be made during the mining process to acquire a better role set. Besides, there are a great number of researchers (Ene *et al.*, 2008; Frank *et al.*, 2008; Molloy *et al.*, 2008) working on the role mining problem. However, none of the above works addresses the role mining problem in Web service composition. To find the roles in setting up role mapping for Web service composition, the bottom-up role engineering method is a more efficient and reasonable manner, since there exists an access control policy hosted in each individual domain.

2 Motivating problem

Since a composite Web service usually needs to access services located in other domains, the inter-domain operation is unavoidable. To guarantee the security of the enterprise system, first we need to set up a proper access control policy to manage the interoperation.

In Fig. 1, local user Bob of Domain 1 wants to access a simple composite Web service {S1, Sa}. It is easy to assign the role R1 to Bob, since Bob is a valid local user in Domain 1. However, for the access

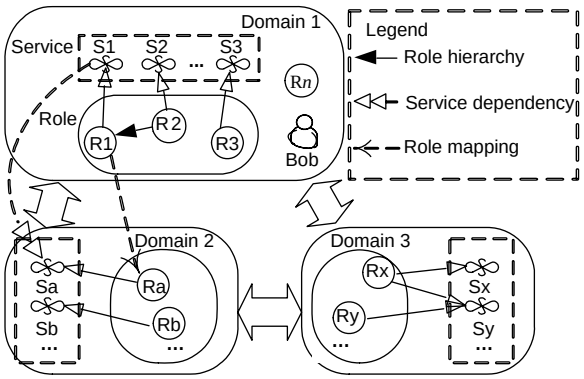

Fig. 1 Motivating problem of role mining for Web service composition

request to service Sa, the access control of the interoperation needs to be set up. Currently, there are two mechanisms for such an interoperation:

1. Add Bob to the valid user set of Domain 2 as Bob′, and assign role Ra to Bob′. When Bob accesses Domain 2, the session will be switched to Bob′ to obtain the permission to access service Sa.

2. Implement the single sign-on across the multiple domains, and assign role Ra to the foreign user Bob. When user Bob requests for the access to service Sa, the single sign-on will maintain the global session of the identification information, and the access controller will check the permission.

As for the first mechanism, it is unreasonable to add the foreign user to local domains, which are separated logically. Another problem is that if there are other users like Tom, Kate, etc. who need to access the composite service {S1, Sa}, Domain 2 has to create 'shadow' users for all those that have to access the composite service. As regards the second mechanism, assigning a local role to the foreign user is also subject to the overhead of management confusion. Besides, both mechanisms are involved in extending the management boundary to a foreign domain, which casts a heavy burden for each local administrator.

For the access control management of Web services based distributed systems, the best practice is to manage the local access control policy locally and to manage the global access control policy concerning foreign domains globally. A role mapping technique can leverage existing access control policies and fill the gap of the access control policy across multi-domain at low cost, which makes it a promising and attractive way to realize the interoperation. Through adding role mapping from role R1 of Domain 2 to role Ra of Domain 2, the users who have been assigned R1 will have the right to act as the one who owns Ra when accessing Domain 2. Such role mapping can be managed globally to alleviate the administration burden, since the access control information is no longer distributed separately at each individual domain. However, all of the benefits can be obtained only after the role mapping is defined and managed properly. In Fig. 1, for the composite Web service {S1, Sa}, both role mappings (R1, Ra) and (R2, Ra) satisfy the requirement for Bob to interoperate with the services located in Domain 2; however, (R1, Ra) is obviously more advantageous, since role mapping (R2, Ra) is against the security principle, least of privilege, i.e., assigning only the necessary permissions to the user and keeping as few user permissions as possible. Compared to (R1, Ra), role mapping (R2, Ra) grants the user an unnecessary permission to access local Web service S2. To the best of our knowledge, there has not been any report of how to mine the minimal role mapping set for a composite Web service.

3 Mining minimal role mapping

How to compose the Web service is out of the scope of this paper. We assume that the Web services have been composed automatically or manually in advance, and that under both situations the Web service composite graph or network is available.

3.1 Primitives

We adopt the NIST standard of the RBAC model. For simplicity, we do not consider any RBAC constraint. In other words, we restrict our study to RBAC1. The RBAC policy is an access control specification, which defines users, roles, permissions, and relations of them. It serves as the fundamental basis for access control implementation. Fig. 2 gives the schematic of the RBAC96 model.

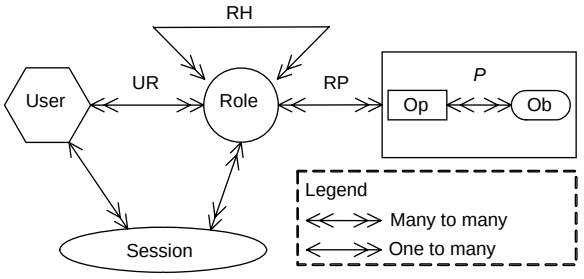

Fig. 2 Schematic of the RBAC96 model
RH: role hierarchy; UR: user-pole; RP: role-permission; Op: operation; Ob: object

Definition 1 (Single domain RBAC policy) A single-domain RBAC policy sp is a 6-tuple (U, R, P, RH, RP, UR):

U, R, P are finite sets of users, roles, and permissions, respectively, wherein P is defined as $P=2^{Op\times Ob}$ (Op and Ob are the sets of operations and objects respectively).

RH (role hierarchy) is a partial order relation defined on role sets, RH$\subseteq R\times R$. If $(r_1, r_2)\in$RH, r_1 is a senior role whereas r_2 is a junior role, and r_1 inherits all the permissions of r_2. The relation can be expressed as $r_1\triangleright r_2$.

RP (role-permission) is a relation defined on the role set and the permission set, RP$\subseteq R\times P$.

UR (user-pole) is a relation defined on the user set and the role set, UR$\subseteq U\times R$.

Definition 2 (Global RBAC policy) The global RBAC policy GP is a 2-tuple (SP, RM):

SP is the set of single-domain RBAC polices as defined in Definition 1.

RM (role mapping) is a binary relation defined on the global role set, RM$\subseteq R\times R$. If $(r_1, r_2)\in$RM, r_1 is a senior role in Domain 1 whereas r_2 is a junior role in Domain 2, and r_1 inherits all the permissions of r_2. The relation can also be expressed as $r_1\triangleright r_2$.

In this paper, for simplicity, we assume that the permissions to access the Web service have been properly defined and the relationship between permissions and Web services is one-to-one. Assuming the Web service set is WS, the following functions can be deduced:

ws_role: $R\rightarrow 2^{WS}$, maps each role to a set of services that are granted to the users who own this role to access.

role_ws: WS$\rightarrow 2^R$, maps each Web service to a set of roles that are assigned the permissions to access the Web service.

3.2 Web service composition

Although there are several modes for Web service composition, the following three are the basic ones. Other more advanced modes can be constructed from these three compositions.

1. Sequential: S1$\odot$S2 represents a composite service that performs service S1 followed by service S2. '$\odot$' is an operator of sequence.

2. Alternative: S1$\oplus$S2 represents a composite service that behaves as either service S1 or service S2. Once one of them executes its first operation the second service is discarded. '$\oplus$' is an alternative operator.

3. Parallel: S1\\S2 represents a composites service that performs both of S1 and S2 in parallel. '\\' is a parallel operator.

The composite Web service could be defined by the following BNF-like notation:

$$CS ::= S \mid S\odot S \mid S\oplus S \mid S \setminus\setminus S,$$

which could also be represented as the Web service composition graph (Fig. 3).

If there are alternative nodes in the Web service composition graph, it can be decomposed as the right part of Fig. 3 to facilitate access control policy analysis. Thus, we will consider the composition graph that contains only sequential and parallel Web service nodes.

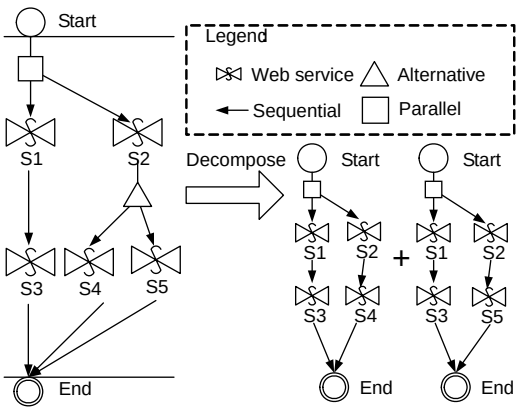

Fig. 3 Web service composition graph

3.3 Problem analysis

In this subsection, we address the problem of role mining for Web service composition. Numerous studies have shown that the automated bottom-up role discovery process provides a number of advantages over the traditional top-down engineering approach. The existing role of the organization can be found in this process by the analysis of existing IT system security architectures. The bottom-up method of role discovery avoids putting in a lot of time and energy to define the role in a top-down process, and it is more accurate because it reflects the practical role of the existing users rather than the system security administrator's opinion. Another important advantage of the bottom-up role in the discovery process is that it can be automated to some extent, using powerful data mining tools and methods.

Take the role mining problem in Fig. 4 as an example. The Web service composition is

$$\text{WSC}=\{\text{S2, S3, S4, S5, S6, S7}\},$$

and there is a role hierarchy (R1, R4)∈RH. R1 is a senior role to R4.

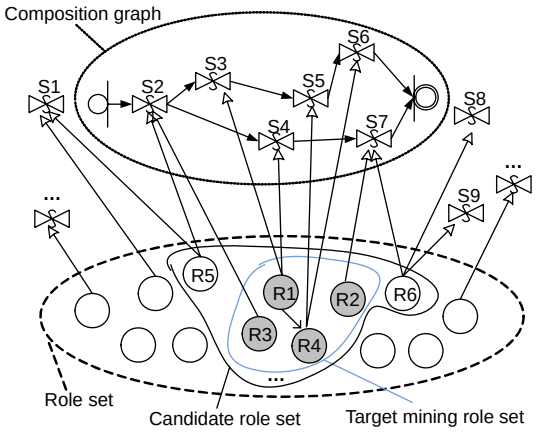

Fig. 4 Role mining problem for Web service composition

From Fig. 4, we could see that the candidate role set to access the composite Web service is

$$\text{CR}=\{\text{R1, R2, R3, R4, R5, R6}\}.$$

Obviously the candidate role set is not the minimal one to access the composite Web service. If the role mapping policy is set up based on the candidate role set, the superfluous administration effort is wasted, and the principle of least privilege is violated. We could find a smaller role set, which is our target mining role set, as follows:

$$\text{TR}=\{\text{R1, R2, R3, R4}\}.$$

Thus, the role mining problem for Web service composition is to find the role set that provides the lowest administration overhead and best complies with the principle of least privilege when setting up role mapping.

Definition 3 (Role mining problem for Web service composition, RMP4WSC) For a Web service composition WSC={ws_1, ws_2, …, ws_n} and a candidate role set CR={r_1, r_2, …, r_n} to access all the services in WSC, the problem is to find the minimal set of roles $R\subseteq$CR, such that

$$\bigcup\nolimits_{r\in R} \text{ws\_role}(r) = \text{WSC}.$$

Given the Web service composition, the candidate role set can be determined via merging the role set of each Web service. The basic role mapping mining problem is concerned with an exact role set to access the composite Web service. However, such an exact role set usually does not exist in the real world. Thus, the security administrator has to tolerate a role set that does not conform to a certain extent. We introduce '∂-consistency' to measure the accuracy of the mining roles.

The reason why the discovered roles fail to give a permission to Web services outside WSC is that it is of great significance for the financial systems to comply with the principle of least privilege, which is also known as the principle of minimal privilege. This principle requires that the subject should be able to access only the resources (objects) that are necessary for its legitimate purpose. According to the business requirement of the financial systems, the access control of the composite Web service should be regulated as strictly as possible, and it should allow the users to access only the Web services that facilitate the fulfillment of their duties. Any additionally granted permission to access the extra Web service may lead to sensitive information leakage. In our real applications, there was once a Web service composition to

generate the audit trail report of the foreign exchange orders. Different traders had different levels to access different order types, for usually one trader has no right to view the trading details of other type orders. However, in the original composition, the roles defined to access the composite service granted the permission not only to generate the report but also to view the detailed audit trail of the orders, which leads to the leakage of sensitive information. Although such leakage is just among the internal traders, the potential risk is serious. No financial systems can stand such a risk. Thus, we emphasize the significance of RMP4WSC, which describes the problem in finding the roles that must cover WSC alone.

Definition 4 (∂-consistency) Given a role set R and a Web service composition WSC, and an integer ∂, ∂-consistency holds between R and WSC, marked as $R\sim_{\partial}$WSC, if and only if

(1) $\mathrm{WSC}\subseteq\bigcup_{r\in R}\mathrm{ws\_role}(r)$, and

(2) $\left|\bigcup_{r\in R}\mathrm{ws\_role}(r)-\mathrm{WSC}\right|\le\partial$.

∂-service consistency is a notation used to bound the degree of difference for a role set when it is assigned to access the Web services. It is also a measurement of the extent to which how the role set complies with the principle of least privilege. The smaller is the value of ∂, the more precise is the role set.

Definition 5 (∂-RMP4WSC problem) Given an integer ∂, WSC=$\{ws_1, ws_2, \ldots, ws_n\}$, and a candidate role set CR=$\{r_1, r_2, \ldots, r_n\}$ to access all the services in WSC, the problem is to find a set of roles $R\subseteq$CR, such that $R\sim_{\partial}$WSC and $|R|$ is minimized.

Although the main purpose is to find the minimal target role set for the role mapping setup, sometimes we need to find the role set that abides by the principle of least privilege most, which is the following problem:

Definition 6 (Maximum consistency RMP4WSC problem, MC-RMP4WSC) Given a Web service composition WSC=$\{ws_1, ws_2, \ldots, ws_n\}$, a positive integer k, and a candidate role set CR=$\{r_1, r_2, \ldots, r_n\}$ to access all the services in WSC, the problem is to find a set of roles CR=$\{r_1, r_2, \ldots, r_n\}$, $|R|=k$, and a positive integer ∂, such that $R\sim_{\partial}$WSC and ∂ is minimized.

Theorem 1 RMP4WSC is NP-complete.

Theorem 2 ∂-RMP4WSC is NP-complete.

Theorem 3 MC-RMP4WSC is NP-complete.

To prove the above three theorems, we reduce them to the classical NP-complete problem, i.e., the set basis problem (Vaidya *et al*., 2007). We follow the steps proposed by Garey and Johnson (1979) to prove that all of the above three problems are NP-complete.

Proof (of Theorem 1) We select the set basis problem as KP and apply the following transformation. Let S denote the Web service set, $c\in C$ the Web service composition WSC, and ws_role(r) ($r\in R$) the element in B. Now the role mining problem to find the role set R is equal to the set basis problem. The mapping is just the simple one-to-one relation; thus, the transformation can be accomplished in polynomial time.

Proof (of Theorem 2) ∂-consistent RMP4WSC is NP-complete. $\partial\ge0$, ∂-consistent RMP4WSC can be categorized into two situations:

1. $\partial=0$. Theorem 2 is an extended version of Theorem 1. If ∂ equals zero, Theorem 2 is reduced to Theorem 1, which has been proved.

2. $\partial>0$. It takes only polynomial time to select the set ts$\in(S-c)$, and $|\mathrm{ts}|=\partial$. We select the set basis problem as KP and apply the following transformation. Let S denote the Web service set, $c\cup$ts the Web service composition WSC, and ws_role(r) ($r\in R$) the element in B. The role mining problem to find the role set R is equal to the set basis problem. The role set R satisfies $R\sim_{\partial}$WSC. The mapping is just the simple one-to-one relation; thus, the transformation can be accomplished in polynomial time.

Proof (of Theorem 3) MC-RMP4WSC is NP-complete. Given the role set R and the Web service composition WSC, it takes only polynomial time to judge whether WSC is contained in $\bigcup_{r\in R}\mathrm{ws\_role}(r)$. It also takes polynomial time to calculate the value of $|\bigcup_{r\in R}\mathrm{ws\_role}(r)-\mathrm{WSC}|$ and to ensure that the value is less than or equal to ∂ under the condition that $|R|$ equals k. We select the set basis problem as KP and apply the following transformation. Let S denote the Web service set, $c\in C$ the Web service composition WSC, and ws_role(r) ($r\in R$) the element in B, and set $\partial=0$. Then the role mining problem to find the role set R is equal to the KP. The mapping is just the simple

one-to-one relation, and thus the transformation can be accomplished in polynomial time.

3.4 Minimal role mining algorithm

Given the candidate role set and the composite Web services, the intuitive way to find the minimal role set to access the composite Web services is to enumerate all the possible combination to form the role se; however, the number of possible role combinations can reach 2^n, wherein n is the candidate role number. Obviously, the quantity of meaningful ones is much less than 2^n and the time taken to enumerate all the possible roles is a big issue. We give our greedy minimal role mining algorithm Greedy-MRM (Algorithm 1). The algorithm consists of three phases:

1. Identification of the initial set of roles. In this phase, we find the entire roles whose granted Web services are totally included in the composite Web service. The roles found in this phase serve as the basis of the minimal role set.

2. Removing redundant roles caused by role hierarchies. The roles found in phase 1 are redundant since there are roles that are junior to others as a result of the role hierarchy. Removing such juniors reduces significantly the size of the roles, especially for the complicated role systems that involve a great number of role hierarchies.

3. Suboptimal role enumeration. In this phase, for each rest role, the granted accessible Web services set is computed, and the one which shares the maximum intersection with the composite Web service is selected.

The main steps of the algorithms are as follows:

Step 1 (line 1): The mining role set MR, the temporary role set TR, and the temporary Web service set TS are initialized.

Step 2 (lines 2-3): Start the loop to find the candidate role set based on the composite Web service.

Step 3 (lines 4-6): Start the loop to add the roles whose granted Web services are completely included in the composite Web service. TR and MR will serve as the basis role set for the mining.

Step 4 (lines 7-11): Start the loop to remove the redundant roles from MR, since if role hierarchies exist among the roles in MR, then only the senior roles will be kept, and the redundant role will be discarded.

Step 5 (lines 12-22): Start the main loop to find the minimal role set. In lines 13-15, the next role will be chosen based on the maximum Web service set intersection. In lines 16-22, the chosen role will be added to the mining role set according to its hierarchy relationship with the existing roles in the set MR.

Step 6 (line 23): The mining role set MR is returned.

Table 1 lists the results after the execution of the Greedy-MRM of the example in Fig. 4.

Algorithm 1 Greedy minimal role mining (Greedy-MRM)

Input: Web service composite graph WSC.
Output: the minimal role set.

```
1  initialize MR←∅; TR←∅; TS←∅;
2  foreach ws∈WSC
3     add role_ws(ws) to CR;
4  foreach r∈CR
5     if ws_role(r)⊆WSC
6        add r to TR and MR;
7  foreach r∈TR
8     foreach r′∈(TR−{r})
9        if r▷r′
10          remove r′ from MR;
11    add ws_role(r) to TS;
12 while (WSC−TS)≠∅ do
13    foreach r∈(CR−TR)
14       find r that maximizes |ws_role(r)∩(WSC−
           TS)|;
15       add r to TR and MR;
16    foreach r′∈MR
17       if r▷r′
18          remove r′ from MR;
19          add r to MR;
20       else if r′▷r
21          remove r from MR;
22    add ws_role(r) to TS;
23 return MR.
```

Table 1 Algorithm results for the example in Fig. 4

Step No.	Variables
1	MR={}, TR={}, TS={}
2	MR={}, TR={}, TS={}, CR={R1, R2, R3, R4, R5, R6}
3	MR={R1, R2, R3, R4}, TR={R1, R2, R3, R4}, TS={}, CR={R1, R2, R3, R4, R5, R6}
4	MR={R1, R2, R3}, TS={S2, S3, S4, S5, S6, S7}, TR={R1, R2, R3, R4}, CR={R1, R2, R3, R4, R5, R6}
5	MR={R1, R2, R3}, TS={S2, S3, S4, S5, S6, S7}, TR={R1, R2, R3, R4}, CR={R1, R2, R3, R4, R5, R6}
6	MR={R1, R2, R3}

4 Simulation

The first purpose of the experimental evaluation is to validate if our algorithm could work in the system that comprises complex role hierarchies. The second purpose is to verify the effect of our approach on reducing the burden of access control management for the security administrator.

We created a test data generator based on the one proposed by Vaidya *et al*. (2008). First a set of roles was created, the number of which was set to a certain maximum number, and the hierarchy relationships were constructed randomly at a certain ratio. Next for each role a random number of permissions were chosen to form the role, and the permissions were associated to a random number of Web services. Finally Web service composition was created via choosing Web services randomly, and the number of composite Web services was set to a certain number.

The experimental data was collected on a machine with a single Intel Core 2 T7500 @ 2.20 GHz, with 1 GB main memory. The operating system was Microsoft XP SP4, and the programming language was Java (JRE version 1.6.0_02). Each data entry of the datasets was obtained as the average time (in ms) over 10 runs.

For each set of experiments, we reported the speed as well as the mining role number. To observe the changes of execution time of different Web service compositions, we define the service-role ratio as follows:

$$\mathrm{SR}=\frac{\text{Composite service number}}{\text{Role number}}.$$

Two kinds of tests were performed on the generated data. One was our greedy approach (G algorithm) to find the sub-optimal role set and the other was the algorithm that exhaustively enumerates all possible role subsets to find the optimal role set (E algorithm). The comparisons were based on two dimensions: execution time and mining results.

Figs. 5a and 5b show the execution time of the G algorithm and the E algorithm respectively, from which we have the following observations:

1. The G algorithm is efficient enough for practical purpose. To the best of our knowledge, most of current Web service compositions are more lightweight than the one tested in our experiment.

2. The execution time increases as the value of SR increases under the condition that the role numbers are the same. This makes sense since for a given role set, the larger is the value of SR, the more complicated is the Web service composition.

3. The G algorithm runs much faster than the E algorithm, since the E algorithm needs to enumerate all the possible role subsets, which takes more loops to obtain the result.

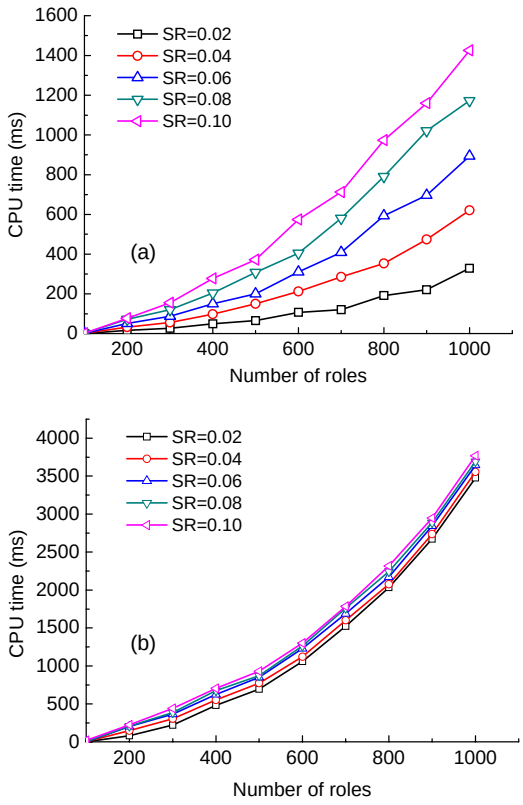

Fig. 5 Execution time of (a) the G algorithm and (b) the E algorithm

Fig. 6 shows the mining results of the G and E algorithms with different numbers of Web services for composition. The E algorithm generated optimal results certainly, since it compares all the possible situations. The results of the G algorithm were also acceptable, only a little different from those generated by the E algorithm. Thus, our approach is qualified to serve as a sub-optimal solution.

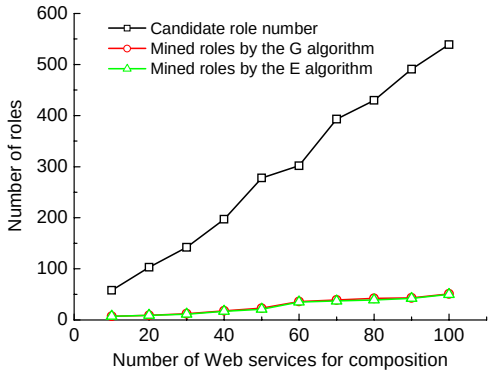

Fig. 6 Mining results of the G and E algorithms

5 Case study

Our approach has been applied in a northern American bank *B*, in which there were a great number of legacy systems. Access control policy management for agent-based Web service composition was a part of the bank's enterprise application integration (EAI), which was to integrate the legacy enterprise application systems as a seamless combination. The solution was based on Web service and agent techniques. There were other traditional EAI solutions. The significant advantage can be obtained via adopting agent and Web service techniques, such as simplicity, open standard, flexibility, and dynamic extensibility.

However, all these benefits could be gained after the access control policy was managed properly. All the local enterprise applications have been upgraded to use the RBAC as the base of the access control policy. It is costly to transfer the whole local access control policy to the central management agent. Thus, role mapping is a sound choice, via keeping the local policy nearly intact and adding the global access control agent to manage the inter-domain access control.

We built up a global access control agent (GACA) based on Java agent development framework (JADE), which is a platform for developing foundations of intelligent physical agents (FIPA) specification in accordance with a multi-agent system. JADE is programmed using Java, which simplifies the implementation of a multi-agent system via utilizing a FIPA standard middleware and a tool supporting debugging and deployment. The JADE platform takes charge of agent communication, agent schedule, agent life cycle management, and other shared resources.

The target of bank's enterprise legacy application integration is a multi-agent system, in which the collaboration among agents is built up on the basis of communication sessions. A session is composed of request, direction, and others. The session of the agents follows a certain state changing rule, and every message will lead the application to a determined state. The major agents are categorized as follows:

AMS (agent management system): An AMS is a built-in agent in JADE, which is used to manage the lifecycle of other agents.

DF (directory facilitator): A DF is used as a yellow page service, by which other agents could register their services and look up others' services.

SB (service broker): An SB is used for the service consumer and service provider to measure the service quality. It also monitors the service running status, and updates the corresponding entry in the DF.

SA (service agent): An SA is used to manage the local Web services for each domain.

SCA (service composition agent): An SCA has the permissions to access all the meta-information of available services. It comprises the services if the composition of current services could meet the new service requirement.

LACA (local access control agent): An LACA takes the responsibility to manage the local access control policies, including local user-role assignment, policy changes, and access evaluation.

GACA (global access control agent): A GACA is responsible for the management of the global access control policy, maintaining the information about the role mappings across multiple domains. GACA contains a component called the 'role miner', which is implemented based on the Greedy-MRM algorithm. A role miner is responsible for finding the minimal role set for a new composite Web service.

Via the collaboration of local agents and global central agents, automated service composition can be achieved, and RMA will take charge of the role mining for each new composite Web service. The process of setting up access control for a new composite Web service (Fig. 7) includes the following steps:

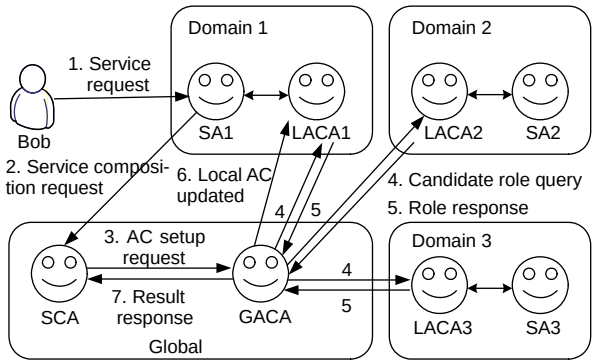

Fig. 7 Process of the access control setup for Web service composition

SA: service agent; AC: access control; SCA: service composition agent; GACA: global access control agent; LACA: local access control agent

Service request: The authenticated user sends a service request to do some operations or access certain resources.

Service composition request: The SA finds that no available Web services satisfy the requirement of the user's request, and then sends the service composition request to the SCA. If no possible composition exists, the response will be sent to SA immediately to indicate a failure of composition; otherwise, a new Web service will be composed and the access control will be set up for the composite Web service.

AC setup request: An access control setup request is sent to the GACA, which contains the role miner.

Candidate role query: The role miner sends the request to each individual LACA to query the candidate roles.

Role response: An LACA checks the request from the role miner against the local access control policy and sends back the response containing the candidate roles in its domain.

Role mining: The role miner finds the minimal role set using the Greedy-RMR algorithm, and after the confirmation of the security administrator, a global role mapping relation will be set up in the GACA.

Local AC update: For the domain from which the composition request is sent out, local AC update information is needed, including the addition of new roles or hierarchies if necessary.

Result response: Finally a notification is sent out to inform the SCA that the access control policy has been set up.

We introduce four practical systems in bank *B* to use our approach (Huang *et al.*, 2009). Note that we change the system names for confidentiality.

1. FXLS, a foreign exchange limit order management system. A foreign exchange system deals with a great number of complicated business rules, which involves many access control roles to guarantee the successful processing of the order.

2. TAS, a trading audit system. Auditing is a critical function for financial systems, since without auditing it is hard to trace the business process, especially when there are some financial frauds. TAD is widely used in bank *B* for a lot of trading systems including foreign exchange and stock trading.

3. GDMS, a global software development management system. In bank *B*, the development teams are distributed worldwide (USA, China, Japan, etc.); GDMS is used for software management throughout the entire development stages.

4. RMS, a resource managing system for employees to manage the resources. The most complicated part of RMS is the reporting module that generates the reports with different sensitive data according to the different user roles.

The numbers of roles and existing Web services are shown in Table 2.

Table 2 Algorithm results of the example

System	Number of roles	Number of Web services
FXLS	≈280	>450
TAS	≈400	>610
GDMS	≈320	>500
RMS	≈150	>230
Total	≈1150	>1790

GACA has been used for more than two years. During this period, the role miner has been requested frequently while a new Web service is composed. Fig. 8 shows the CPU time records of the role miner. The records were selected from the execution log and we selected only those with execution time of larger than 100 ms. The log contained the composition, which may be cancelled by the administrator due to constraint violation; however, this cancellation has no impact on our observation about the performance and effectiveness of our approach in real applications. The

maximum execution time was less than 1.2 s, which is efficient enough for practical enterprise use.

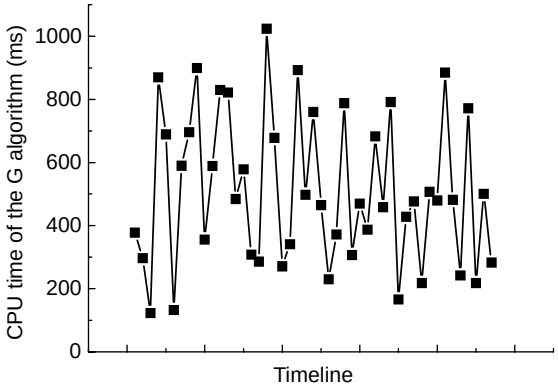

Fig. 8 CPU time of the role miner according to the log

Table 3 shows the application results of our approach. Due to space limitation, we listed only 15 records of the results. In Table 3, |WSC|, CanRN, MRN, CRN, DRN, and ARN are the numbers of Web services, candidate roles, mined roles by applying our approach, roles confirmed by the security manager, roles deleted after review of the security manager, and roles added by the security manager, respectively. AR is the amending ratio,

$$AR=(|ARN|+|DRN|)/|CRN|\times 100\%.$$

And CR is the correct ratio,

$$CR=|CRN|/|MRN|\times 100\%.$$

We used AR and CR to evaluate the effectiveness of the mining algorithm. The lower value of AR and the higher value of CR indicate the better algorithm. Table 3 shows that our approach is acceptable in practical applications.

6 Conclusion

In this paper, we describe the role engineering issue encountered in Web service composition. The role mining problem for Web services has not been addressed yet. Based on implementation experience, we analyze the minimal role mining problem for Web service composition, and give a formal definition. The minimal role mining problem for Web service composition is NP-complete and thus the problem is reduced to a set basis problem. We also propose a greedy algorithm to find the sub-optimal mining result of the roles. We demonstrate the simulation results and introduce a real application context to illustrate the practical usage of our methods. Security aware Web service composition and the way to set the bound to the minimal cover size will be our future work.

Table 3 Application results of our approach

No.	|WSC|	CanRN	MRN	CRN	DRN	ARN	AR (%)	CR (%)
1	18	69	8	8	0	0	0.00	100.00
2	16	58	9	9	0	0	0.00	100.00
3	15	48	10	10	0	0	0.00	100.00
4	31	79	13	12	1	0	8.33	92.31
5	27	74	12	12	0	0	0.00	100.00
6	16	56	9	9	0	0	0.00	100.00
7	23	59	11	10	1	0	10.00	90.91
8	25	61	10	10	0	0	0.00	100.00
9	31	73	14	13	1	0	7.69	92.86
10	18	81	10	10	0	0	0.00	100.00
11	22	66	11	11	0	0	0.00	100.00
12	30	73	13	13	0	0	0.00	100.00
13	30	84	15	14	1	1	14.29	93.33
14	21	70	13	13	0	0	0.00	100.00
15	23	64	12	12	0	0	0.00	100.00

|WSC|, CanRN, MRN, CRN, DRN, and ARN: numbers of Web services, candidate roles, mining roles by applying our approach, roles confirmed by the security manager, roles deleted after review of the security manager, and roles added by the security manager, respectively. AR: the amending ratio, AR=(|ARN|+|DRN|)/|CRN|×100%; CR: the correct ratio, CR=|CRN|/|MRN|×100%

References

Atluri, V., 2008. Panel on Role Engineering. Proc. 13th ACM Symp. on Access Control Models and Technologies, p.61-62. [doi:10.1145/1377836.1377846]

Carminati, B., Ferrari, E., Huang, P.C.K., 2005. Web Service Composition: A Security Perspective. Proc. Int. Workshop on Challenges in Web Information Retrieval and Integration, p.248-253. [doi:10.1109/WIRI.2005.36]

Colantonio, A., di Pietro, R., Ocello, A., 2008. A Cost-Driven Approach to Role Engineering. Proc. ACM Symp. on Applied Computing, p.2129-2136. [doi:10.1145/1363686.1364198]

Coyne, E.J., 1996. Role Engineering. Proc. 1st ACM Workshop on Role-Based Access Control, p.15-16. [doi:10.1145/270152.270159]

Dustdar, S., Schreiner, W., 2005. A survey on Web services composition. *Int. J. Web Grid Serv.*, **1**(1):1-30. [doi:10.1504/IJWGS.2005.007545]

Eid, M., Alamri, A., Saddik, A.E., 2008. A reference model for dynamic Web service composition systems. *Int. J. Web Grid Serv*., **4**(2):149-168. [doi:10.1504/IJWGS.2008.018885]

Ene, A., Horne, W., Milosavljevic, N., Rao, P., Schreiber, R., Tarjan, R.E., 2008. Fast Exact and Heuristic Methods for Role Minimization Problems. Proc. 13th ACM Symp. on Access Control Models and Technologies, p.1-10. [doi:10.1145/1377836.1377838]

Essmayr, W., Probst, S., Weippl, E., 2004. Role-based access controls: status, dissemination, and prospects for generic security mechanisms. *Electron. Comm. Res.*, **4**(1/2):127-156. [doi:10.1023/B:ELEC.0000009285.50078.b2]

Ferraiolo, D.F., Sandhu, R., Gavrila, S., Kuhn, D.R., Chandramouli, R., 2001. Proposed NIST standard for role-based access control. *ACM Trans. Inform. Syst. Secur*., **4**(3):224-274. [doi:10.1145/501978.501980]

Ferraiolo, D.F., Chandramouli, R., Ahn, G., Gavrila, S.I., 2003. The Role Control Center: Features and Case Studies. Proc. 8th ACM Symp. on Access Control Models and Technologies, p.12-20. [doi:10.1145/775412.775415]

Frank, M., Basin, D., Buhmann, J.M., 2008. A Class of Probabilistic Models for Role Engineering. Proc. 15th ACM Conf. on Computer and Communications Security, p.299-310. [doi:10.1145/1455770.1455809]

Garey, M.R., Johnson, D.S., 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman, New York.

Goncalves, G., Poniszewska, M.A., 2008. Role engineering: from design to evolution of security schemes. *J. Syst. Softw.*, **81**(8):1306-1326. [doi:10.1016/j.jss.2007.11.003]

Huang, C., Sun, J., Wang, X., Si, Y., 2009. Selective Regression Test for Access Control System Employing RBAC. Proc. 3rd Int. Conf. and Workshops on Advances in Information Security and Assurance, p.70-79. [doi:10.1007/978-3-642-02617-1_8]

Ko, J.M., Kim, C.O., Kwon, I., 2008. Quality-of-service oriented Web service composition algorithm and planning architecture. *J. Syst. Softw*., **81**(11):2079-2090. [doi:10.1016/j.jss.2008.04.044]

Lécué, F., Delteil, A., Léger, A., 2008. Towards the Composition of Stateful and Independent Semantic Web Services. Proc. ACM Symp. on Applied Computing, p.2279-2285. [doi:10.1145/1363686.1364229]

Li, N., Tripunitara, M.V., 2006. Security analysis in role-based access control. *ACM Trans. Inform. Syst. Secur*., **9**(4):391-420. [doi:10.1145/1187441.1187442]

Li, N., Byun, J., Bertino, E., 2007. A critique of the ANSI standard on role-based access control. *IEEE Secur. Priv. Mag.*, **5**(6):41-49. [doi:10.1109/MSP.2007.158]

Molloy, I., Chen, H., Li, T., Wang, Q., Li, N., Bertino, E., Calo, S., Lobo, J., 2008. Mining Roles with Semantic Meanings. Proc. 13th ACM Symp. on Access Control Models and Technologies, p.21-30. [doi:10.1145/1377836.1377840]

Neumann, G., Strembeck, M., 2002. A Scenario-Driven Role Engineering Process for Functional RBAC Roles. Proc. 7th ACM Symp. on Access Control Models and Technologies, p.33-42. [doi:10.1145/507711.507717]

Park, J.S., Sandhu, R., Ahn, G., 2001. Role-based access control on the Web. *ACM Trans. Inform. Syst. Secur*., **4**(1):37-71. [doi:10.1145/383775.383777]

Schaad, A., Moffett, J., Jacob, J., 2001. The Role-Based Access Control System of a European Bank: A Case Study and Discussion. Proc. 6th ACM Symp. on Access Control Models and Technologies, p.3-9. [doi:10.1145/373256.373257]

Schlegelmilch, J., Steffens, U., 2005. Role Mining with ORCA. Proc. 10th ACM Symp. on Access Control Models and Technologies, p.168-176. [doi:10.1145/1063979.1064008]

Sycara, K., Paolucci, M., Ankolekar, A., Srinivasan, N., 2003. Automated discovery, interaction and composition of semantic Web services. *J. Web Semant.*, **1**(1):27-46. [doi:10.1016/j.websem.2003.07.02]

Talib, M.A., Yang, Z., Ilyas, Q.M., 2006. A framework towards Web services composition modelling and execution. *Int. J. Web Grid Serv*., **2**(1):25-49. [doi:10.1504/IJWGS.2006.008878]

Vaidya, J., Atluri, V., Guo, Q., 2007. The Role Mining Problem: Finding a Minimal Descriptive Set of Roles. Proc. 12th ACM Symp. on Access Control Models and Technologies, p.175-184. [doi:10.1145/1266840.1266870]

Vaidya, J., Atluri, V., Warner, J., Guo, Q., 2008. Role engineering via prioritized subset enumeration. *IEEE Trans. Depend. Secur.*, **99**. [doi:10.1109/TDSC.2008.61]