An Autonomic Service Discovery Mechanism to Support Pervasive Device Accessing Semantic Grid

Tao Guan, Ed Zaluska, David De Roure
School of Electronics and Computer Science
University of Southampton
Southamtpon, UK
{tg04r, ejz, dder}@ecs.soton.ac.uk

Abstract

An important challenge of integrating pervasive devices into Grid environment to enhance pervasive device capabilities is that pervasive devices need to locate, find, select and invoke the appropriate Grid services in an autonomic and flexible way. However, at this stage, both Grid service description and discovery standards are not very sophisticated. Semantic web technology benefits the concept of Grid services on pervasive devices by adding machine-processable explicit knowledge into the interaction between pervasive devices and Grid services. In this paper, we have presented a semantic-based Grid service discovery mechanism to support pervasive device accessing Grid services. In order to protect personal privacy in the pervasive computing environment, a service discovery restricting mechanism is also built to ensure the service can automatically be hidden for unauthorized users.

1. Introduction

During the past several years, there are two significant trends in the area of ubiquitous computing with rapid improvements of the modern computing technology: more pervasive devices are deployed; more integration and computation power is required. Although new generation pervasive devices are gradually improve their absolute capabilities (e.g. mobile phones, PDAs), it is still a challenging objective to create complicated applications on them because pervasive devices are typically resource-constrained, relative to the static counterparts (e.g. desktops, workstation). A feasible solution is that pervasive devices make use of Grid services to access computational resources automatically on demand with appropriate quality of service delivery. Various Grid services enhance the capabilities of pervasive devices to execute complex applications which cannot be run directly on devices themselves. However, in order to realize such a vision, pervasive devices need to locate, find, select and invoke the appropriate Grid services in an autonomic and flexible way. Semantic web technology, providing a very considerable degree of automatic processing, interoperation and integration, will help us to implement an effective Grid service discovery mechanism.

2. Grid Service Description

In our system infrastructure, a mechanism which enables the complete separation between Grid service implementations and Grid service description is adopted. After Grid service providers deploy their services on the Grid platform, they need to publish the service to the information center for being discovered by ubiquitous users. An ontology class is defined with OWL-S [1] language for each Grid services, which specifies the name, the category, and functionality of the Grid service. Because the service implementation and the service description are totally separated, the service provider can publish the same service to different information centers with different service descriptions. This mechanism enables Grid services to be reused under different pervasive computing environments.

3. Restricting Service Discovery

Pervasive users access Grid services with their portable devices, which may expose their personal information. Hence, protecting personal privacy is an essential issue of designing a service discovery mechanism. To solve this problem, we build a service discovery restricting component and deploy it into the service discovery mechanism.

The service discovery restricting component is built based on a style of ontology classes and use the context reasoning method to restrict the service can only be discovered by the right users. We have defined five classes of objects, “User”, “Device”, “Location”, “Event”, and “Restriction”. The “Restriction” class represents all restrictions for a pervasive users. Each “Restriction” class has “permit” and “forbid” properties, describing which kinds of Grid services are permitted to the pervasive user, and which kinds of services are forbidden to the pervasive user. The service provider defines the security level for every service in the service description document, which is published on the information centre. When new pervasive users send the request to information centre to query the desired Grid service, the user personal information decides their secure accessing level during the authorization process, which is shown in their “User” class. The information centre will reason and decide whether Grid services can be exposed by comparing the service’s security level and pervasive users’ accessing level.

4. Information Centre Architecture

The information centre is responsible for discovering Grid services based on user’s security level and initializing the communication between portable devices and the proxy device. Figure 1 shows the architecture of the information centre.

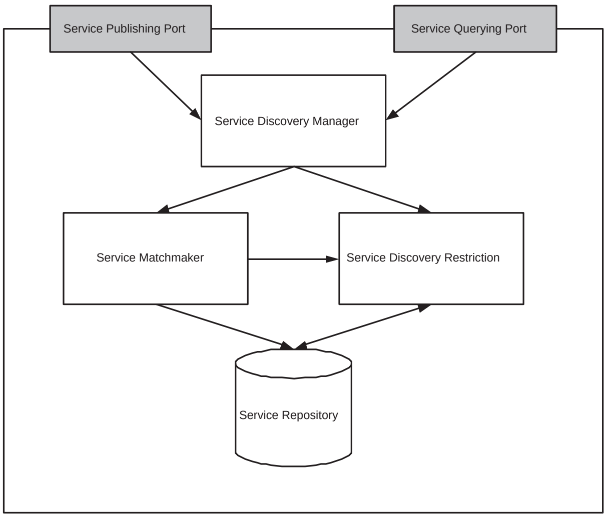

Figure 1. Architecture of Information Centre.

When a service is published through service publishing port, the service discovery manager will forward this service to the service matchmaker component. Usually, the service description contains the semantic information. The service matchmaker component classifies this semantic information, including the service category, the service functionality, the service accessing level, and store them in the service repository.

The information centre also provides a service querying port, which can be used to search a service based on its capabilities. The service query is transmitted to service matchmaker after service discovery manager accepts it. Once the semantic matching based on OWL-S profile information is finished, a list of matching services will be returned to service discovery manager. The service discovery manager then contacts service discovery restricting component, to validate whether all of these services can be accessed by this service requester. After removing the confidential services, a query response which contains the service keys, service names, and service descriptions will be composed. In additional to the basic service information, the matching level for each returned service will also be attached to the response. The service requester will select and invoke the appropriate Grid services based on all of these information.

5. Experimental Results

We have implemented this semantic Grid service discovery mechanism with jUDDI toolkit [2], RACER system [3], MySQL database and AJAX technique. To evaluate our Grid service discovery mechanism, we compare our semantic-extending service registry with traditional UDDI web service registry. We use the system response time as a performance index, and focus on calculating the time required to process a query.

To obtain the time of querying a Grid service, we design an experiment. Both real semantic Grid services and pseudo services are published in the information centre. Altogether, more than two hundred services covering various applications such as information providing and wine selling are deployed on our service registry. We also build a basic UDDI web service registry and deployed a number of web services on it to compare the difference of service query time. The time of querying a service from SSR (Semantic service registry) is a little longer than from the traditional web service registry. This is due to the additional computation efforts required to determine concept subsumption relationships in the Racer system.

References

[1] C.Welty M.Smith and D.McGuinness. Web ontology language guide version 1, 2003.

[2] http://ws.apache.org/juddi/. juddi.

[3] Volker Haarslev and Ralf Moller. Racer: a core inference engine for the semantic web. In *In. Proc. of the 2nd International Workshop on Evaluation of Ontology-based Tools*, pages 27–36, 2003.