A semantic description framework for web services descriptions and matchmaking

Ye Zhang

A SEMANTIC DESCRIPTION FRAMEWORK FOR WEB SERVICES DESCRIPTIONS AND MATCHMAKING

A THESIS SUBMITTED TO DURHAM UNIVERSITY
FOR THE MASTER OF SCIENCE BY RESEARCH DEGREE PROGRAM
IN THE FACULTY OF SCIENCE

2009

By
Ye Zhang
Department of Computer Science
Supervisors: Dr. William Song, Prof. Keith Bennett and Prof. David Budgen

Contents

Abstract

In the M.Sc thesis, the project focuses on the problems of semantic-based representing and retrieving Web services based on the capabilities of services. Service description is critical to application development in Web service environments. There are a number of motivated research developed for representing Web services by different research organisations such as WSDL and UDDI. Both of them are existing standards for Web services. WSDL is designed to provide descriptions of message transport and interface used by each service. UDDI provides a registration structure for businesses and services, and describes businesses and services using their physical attributes in terms of names, addresses, human-understandable business descriptions and service descriptions. Both WSDL and UDDI lack semantic-based description information and a number of essential factors of service capabilities are out of the current description frameworks such as the degrees of service capabilities, relationships between users and services. Moreover, the discovery mechanism provided by UDDI is "exact match" search on the business or service names and descriptions. Actually, service providers and service consumers may have very different background and knowledge, so they do not usually share the same description information for the same item in their minds. It is difficult to locate the proper Web services if users do not express their requirements exactly same with the service provider advertisements. With the consideration of above problems, the development of techniques to semantically represent Web services is necessary for the Web service description and matchmaking.

To address the above problems, we identify several requirements and essential factors that a Web service description framework should have and propose a semantic rich modelling framework to integrate these factors to describe Web services capabilities in unambiguous and computer-understandable forms with ontology. The novel description framework is the Business-Service-User (BSU) framework which provides a semantic based description information for *business, service* and *user*. Another important goal for the BSU framework is to integrate with current Semantic Web markup

languages so that the framework can be easily accessed and understood by the computer. In this thesis, we use the semantic web language OWL to represent the BSU framework and the new semantic description language is called OWL-BSU, which is a computer-interpretable description of the business, service and user. Moreover, a simple and effective matching algorithm is designed to calculate the semantic relationships between service consumers requirements and service descriptions. To make our approach work in the real world, we develop a semantic Web services search engine, which integrates OWL-BSU and the matching algorithm on the top of UDDI registry. The evaluation experimental results have shown that our approach can achieve a great improvement on retrieval performance in terms of recall and precision, comparable to the existing UDDI registry.

Publication: This work has been published and presented in UK e-Science Conference 2004, Ye Zhang and William Song, *Semantic Description and Matchmaking for Grid Services*, Proceedings of UK e-Science All Hands Meeting, ISBN 1-904425-21-6, page, 205-208, September, 2004, Nottingham, UK.

Chapter 1

Introduction

1.1 The Content of Web Services

Recently, Web services [26] have been widely used through Internet and enterprise networks. Web services are developed based on the concept of service-oriented architecture [24] that enables software components including application objects, functions and processes from different systems, to be exposed as services [20]. Currently, several research organisations (e.g., W3C) and computer vendors (i.e., Sun, Microsoft and IBM) provide a number of definitions of Web services depending on their purposes. In the thesis, we use IBM definition of Web services because, to the best of our knowledge, IBM's definition is more concise and cited than other definitions of vast vendors and research organisations. In IBM definition,

"*Web services are a new breed of Web applications. They are self-contained, self-describing, modular applications that can be published, located, and invoked across the Web. Web services perform functions, which can be anything from simple requests to complicated business processes. Once a Web service is deployed, other applications (and other Web services) can discover and invoke the deploy service* [26]".

The architecture of Web service [26] consists of service provider, service broker and service requester which are shown in Figure 1.1. The definitions of the three components of Web services are given in the followings:

- *Service Provider*: a service provider is a network entity in charge of developing and deploying the Web services, which allows to share its services [26].

- *Service Broker*: a service broker is also called service registry (we use the service registry through the thesis), which is the service that returns the location of Web

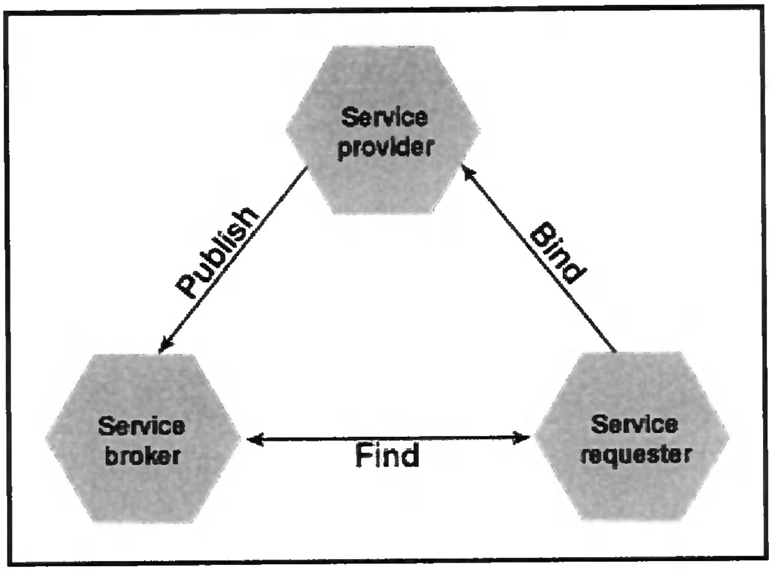


Figure 1.1: Architecture of Web services [26]

services in response to a query from the service requester [26].

- *Service Requester*: a service requester is also called service consumer (we use the service consumer through the thesis), which is responsible for using the shared services. A consumer locates the Web service using the service registry, invokes the required services, and executes it from the service provider [26].

In this architecture, a network entity can play multiple types of roles at the same time [26]. For example, a service provider can be a service consumer and service broker.

1.2 Motivations

The UK e-science program aims to develop support for large-scale sciences through distributed global collaborations [33]. A significant focus of the e-science program is the development of a communication and computational infrastructure to the work of scientists with the Web services [33]. Before using Web services, consumers must locate one or more services that match a given description (user requirements). Two kinds of description methods are considered for service representation at the moment, which are explicit representation and implicit representation [48]. Explicit representation describes the service capability directly and explicitly. Implicit representation, on the other hand, describes the target service using some of the characteristics to form a raw conceptual frame [48]. Actually, in most cases, users do not know the exact representation for the services, or do not know how to structurally form their specification of the services. The users have only some general and vague representations of services they are looking for and they only can provide imprecise description information [18].

Currently, WSDL [14] and UDDI [11] are designed to provide description frameworks for Web services. WSDL is designed to provide descriptions of message transport and interface used by each service [20]. UDDI provides a registration structure for businesses and services, and describes businesses and services using their physical attributes in terms of names, addresses, human-understandable business descriptions and service descriptions [20, 11, 22]. Moreover, the discovery mechanism provided by UDDI is still "exact match" search on the names and descriptions of both business and service through general industrial category [20, 11, 22]. Although the category with taxonomy information contains semantics, it is still a problem for the service consumers to mimic the service providers' understanding of how their services sharing into the categories that they are using. This is because service providers and service consumers may have very different background and knowledge, so they do not usually share the same description information for the same item in their minds. It is difficult to locate the proper Web services if users do not express their requirements exactly same with the service provider advertisements.

We find that current service description frameworks lack semantic-based description information and a number of essential factors of service capabilities are out of the current description frameworks such as the degrees of service capabilities, relationships between users and services. With the consideration of above problems, the development of techniques to semantically represent Web services is necessary for the Web service description and matchmaking.

1.3 Proposed Solutions

To address the problems, an important goal for us is to establish a semantic-based framework where the service descriptions are made and shared between the service providers and consumers. Therefore, we identify several requirements and essential factors that a Web service description framework should have and propose a semantic rich modelling framework to integrate these factors to describe Web services capabilities in unambiguous and computer-understandable forms with ontology. The new description framework is called the Business-Service-User (BSU) framework, which investigates a number of abstract essential attributes and relationships between these attributes in order to represent the capabilities of Web services in a sophisticated way. Another important goal for the BSU framework is to integrate with current Semantic Web markup languages so that it can be easily accessed and understood by the computer. Many different Semantic Web languages are available for supporting semantic-based representation, for example, RDF [9] and OWL [3]. Among these languages, OWL is the most advanced one at the moment. Then, the BSU framework is represented by the OWL language [7] and generates a new description language, OWL-BSU, which is specially for describing businesses, services and users in Web service environments. Moreover, a simple and effective matching algorithm is designed to calculate the semantic relationships between service consumers requirements and service descriptions. To make our approach work in the real world, we develop a semantic Web services search engine, which integrates OWL-BSU and the matching algorithm on the top of UDDI registry. The evaluation experimental results have shown that our approach can achieve a great improvement on retrieval performance in terms of recall and precision, comparable to the existing UDDI registry.

1.4 Aims and Excepted Contributions

The one-year M.Sc project will investigate the technologies developed as part of Web services, Semantic Web, semantic matching algorithm, conceptual modelling to cope with the semantic-based service description and discovery problem in Web service environments. The major four contributions of the project are described in the followings:

- Identifying requirements and essential factors that a Web service description framework should have and proposing a semantic rich modelling framework to describe Web services capabilities with ontology information;

- Integrating the description framework with current Semantic Web markup languages to generate a new semantic description language so that the framework can be understood by computers;

- Developing a matching algorithm which can calculate the relationships between user requirements and provider advertisements by using the new description framework;

- Implementing a prototype of a semantic Web service search engine which integrates the new description framework and matching algorithm together.

1.5 Thesis Structure

The central focus of the thesis is the semantic description framework for Web services. Accordingly, the rest of the thesis is organised as follows:

- **Chapter 2: Literature Survey** we survey a number of existing service description frameworks and service discovery systems, and then discuss the limitations of current approaches;
- **Chapter 3: Semantic Description Framework and Ontology Design** we propose an abstract semantic description framework for Web services, and then show how to generate ontologies based on the new description framework;
- **Chapter 4: Importing OWL-BSU Ontologies into UDDI and Semantic Matching** we will introduce how to map OWL-BSU into the existing UDDI records, which provide a way to record semantic information in the UDDI records. Moreover, a simple and effective matching algorithm will be discussed in this chapter;
- **Chapter 5: Implementations and Evaluations** the system architecture and components of a semantic Web services search engine will be described in detail. Also, evaluation methodologies, metrics and experimental results will be discussed;
- **Chapter 6: Conclusions and Further Work** we will conclude the one-year project and point out some research directions for further work.

Chapter 2

Literature Survey

Before using Web services, consumers must locate one or more services that match their requirements. The service description and discovery mechanisms are critical to Web service application. In Web service environments, a service description framework represents the capabilities of a service. Service discovery consists of the searching and locating the information of services based on the services description in response to the user queries. In this chapter, we start from providing the background knowledge of Web service description frameworks in Section 2.1. Section 2.2 describes a number of service discovery systems. We discuss the limitations of existing approaches in Section 2.3, which give us the inspiration of our research work.

2.1 Service Description Frameworks

In this section, a number of existing service description frameworks have been described including WSDL, UDDI, RDF, DAML-S and OWL-S. We introduce these service description frameworks in detail.

2.1.1 WSDL

Web Services Definition Language (WSDL) [14] is used as the metadata language for representing Web services between service providers and service consumers. It is an XML-based standard and used to describe the two functionality of Web services such as the *attributes* (e.g., service location) and the *access information* (e.g., port type, binding and port) [20]. WSDL employs seven abstract elements to define Web services in terms of *type, message, operation, port type, binding, port* and *services* [14].

The definition of these elements are explained in the literature [14]. In their content, type is a contain of type definitions used by a number of type systems; message is an abstract definition of communicated data; operation is an abstract description of an action which can be supported by Web services; port type is an abstract set of operations which could be supported by some endpoints; binding is a concreted protocol and data format specification used to attach a specific protocol or data format to an message; port is a single endpoint which is defined as a combination of a binding and a network address, operation or endpoint; and service is a collection of related endpoints [14]. WSDL is designed to provide a description framework, which is particularly for the service message transport and interface.

2.1.2 UDDI

Universal Description, Discovery, and Integration (UDDI) [11] is created by an industry initiative for developing a platform-independent and open framework. It is able to provide the capabilities of describing and discovering services by the human-understandable attributes [20]. The description framework in UDDI describes businesses and services using a number of physical attributes in terms of name, address, business description and service description, and also provides an industrial category of business and services. UDDI is one of the core components of Web services, which defines the standard interfaces and registries of Web services. It creates a standard interoperable platform, and enables business companies to retrieve and use Web services in a quick, easy, and dynamic way over the Internet [20]. The mechanisms in UDDI are more like the yellow pages [20, 22].

2.1.3 RDF

The Resource Description Framework (RDF) [9] is originally designed by W3C [12] to enable the exchange, encoding and reuse of structured metadata of resource description. In general, RDF is to be used as a method for modeling information of resource in RDF syntax formats [9]. The syntax of RDF is a simple, but powerful modeling for representing Web services [8], which provides a semantic-based means for publishing both human-readable and machine-processable vocabularies. The description framework of RDF [46] is shown in Figure 2.1, in which resources have properties (attributes or characteristics). When RDF defines a resource, it uses Uniform Resource Identifier (URI) to identify the resource. The properties in the description framework can be identified by property-types, and the corresponding values of property-types [9]. In RDF, property-types describe the relationships of values associated with resources [9, 19]. Moreover, RDF is able to provide a number of structural constraints to express semantics in an unambiguous way [19, 9]. Currently, RDF is widely used in Semantic Web activity for knowledge management applications and it is an alternative approach to represent Web services [19]. For example, in the OMM matchmaker [31], user request descriptions and usage policies are all semantically described by RDF. However, RDF is far more general for representing Web services and it does not provide any specific attributes for Web services.

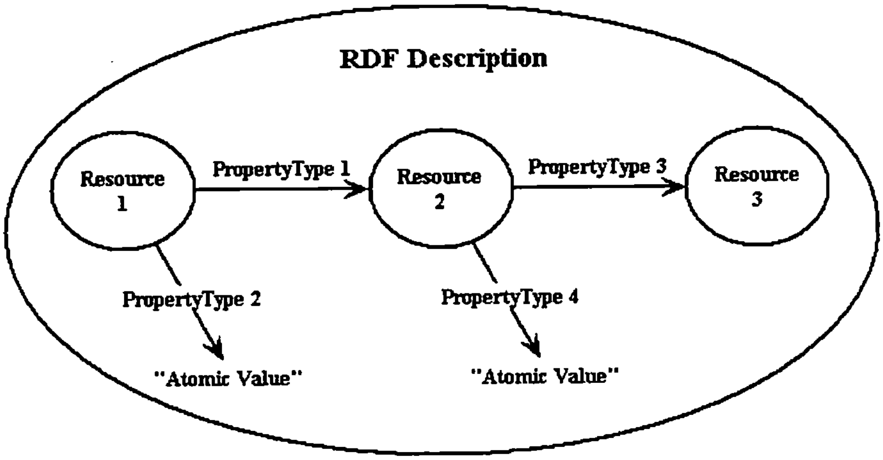


Figure 2.1: RDF Description Framework [46]

2.1.4 DAML-S and OWL-S

Ontology Web Language-Service (OWL-S) [43, 5] is a markup language of Semantic Web for representing services. DAML-S [1] is a former version of OWL-S, which uses the same description framework in Figure 2.2. The difference between DAML-S and OWL-S it that the different Semantic Web languages are used to represent the service description framework, OWL-S is represented by the OWL language and DAML-S is represented by the DAML language. In this Chapter, we describe OWL-S only, because it is the latest version for representing Web services. OWL-S supplies Web service providers unambiguous, computer-understandable forms to describe the properties and capabilities of services [43]. In OWL-S, Web service can be represented by two categories such as simple services and complex services [5]. Simple services are also called atomic services, which are involved in a single Web-accessible computer application, and the complex services are the combination form of the simple services, which are consisted of a number of simple services [6]. OWL-S is expected to implement four major tasks which are *automatic Web service discovery, automatic Web service invocation, automatic Web service composition and interoperation and automatic Web service execution monitoring* [6].

The core class of the OWL-S description framework is *Service* class that provides an organisation point of reference for representing Web services and the *ServiceProfile, ServiceModel*, and *ServiceGrounding* are other classes for representing the service [43, 6]. The service profile explains "what the service does". It provides the information which is required by a service broker to determine whether the service meets service consumer needs [6]. The service model describes "how the service works" and the service grounding specifies how a service consumer can access the service [6].

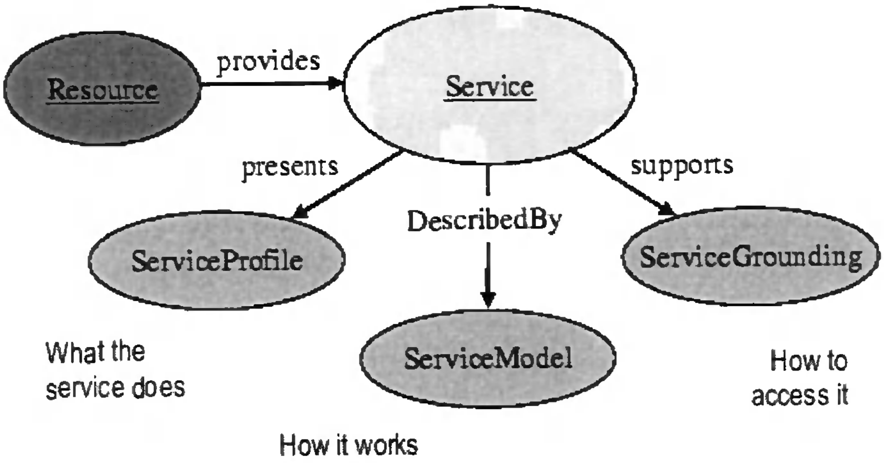


Figure 2.2: OWL-S Description Framework [6]

OWL-S involves automatic locating Web services to satisfy the consumer's requirements. It could also meet the information requirements due to Web service discovery could be specified as computer-understandable semantic markup [43, 6].

2.2 Service Discovery Systems

A service discovery system is a directory-based repository with protocols for network access, which contains a descriptive attribute-based information optimised for search. In this section, we will introduce some existing Web service discovery systems such as UDDI, LARKS matchmaker, DAML+OIL matchmaker and OMM matchmaker.

2.2.1 UDDI

In Section 2.1.2, we had introduced UDDI but focused on the service representation functionality. In this section, we will describe the service discovery mechanism in UDDI. Currently, service consumers are using the UDDI registry to locate services. UDDI provides service registry for service providers and service retrieval for service consumers [11]. Actually, the discovery mechanism provided by UDDI is quite straightforward; the working principle is comparable to the yellow pages repository, which means it provides the "exact match" search by the business and services names and description on a general industrial category of businesses and services [11, 22]. The results would be only obtained, when the service consumer requirements exactly same with the service advertisements in UDDI.

2.2.2 LARKS Matchmaker

Sycara et al. [50] developed RETSINA [49] (Reusable Task Structure-based Intelligent Network Agents) that is a multiagent infrastructure providing service discovery capabilities. In RETSINA, three components are involved including service provider, service requester, and middle agent, which are similar with the components of the Web service architecture [49]. They defined and implemented an agent capability description language called LARKS (Language for Advertisement and Request for Knowledge Sharing) in order to describe the agent or service's capabilities. LARKS employs a local ontology which is to be used by application domain knowledge in advertisements and requirements. The overview of LARKS matchmaking process [50] is shown in Figure 2.3. Three major steps are designed in the service discovery process [50]. First, LARKS matchmaker compares the requirements of service consumers against all advertisements in the database. Second, the matchmaker determines which service capabilities will be the best with the requirements. In the matchmaking process of LARKS matchmaker, every pair of requirement and advertisement needs to go through five different LARKS filters: *context matching, profile comparison, similarity matching, signature matching* and *constraint matching*. Based on the filter types, four matching modes have been implemented in the LARKS matchmaker: *complete* *matching mode, relaxed matching mode, profile matching mode* and *plug-in matching model*. Finally, the matchmaker returns the results with the location information and related capabilities of service [50]. The LARKS matchmaker can provide sophisticated and complicated matching process which employ the techniques from areas of information retrieval, AI and software engineering for computing syntactical and semantic similarity. When using the LARKS matchmaker, the user should clearly understand the five filters and four matching modes, otherwise, the filter and matching modes can be misleading and results cannot satisfy the user's requirements. The service description language in LARKS is *Agent Capability Description Language* which is limited in current Web services description on the lack of Web-related description. Moreover, the LARKS matchmaker has not been experimentally evaluated yet.

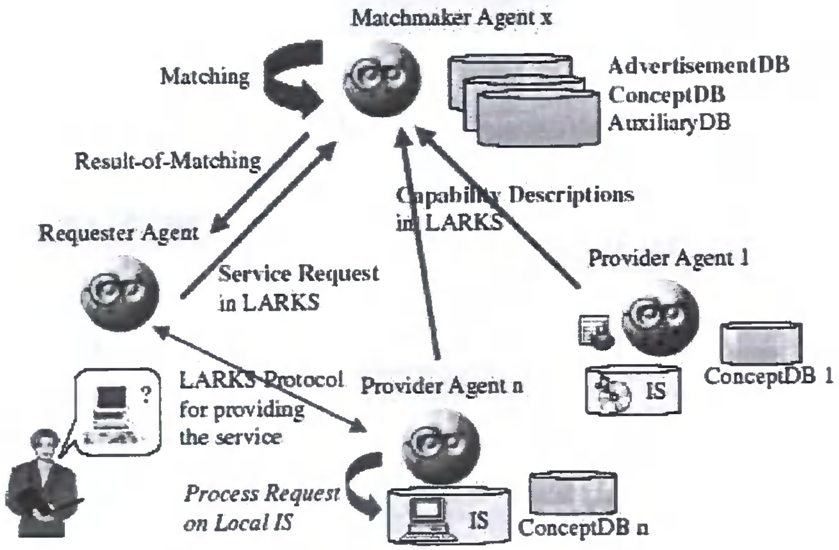


Figure 2.3: LARKS matchmaking [50]

2.2.3 DAML+OIL Matchmaker

Li and Horrocks [40] developed a novel framework for service matchmaking that employs the DAML+OIL logic [34] to match user requirements and service provider advertisements. In the framework, the DAML-S service ontology is used to provide service descriptions. The ontology-based descriptions in the DAML+OIL matchmaker are implemented in a prototype, where agents can provide capabilities for searching and advertising for services that match some semantic description [40]. They employ JADE as the agent platform and the Racer as a DL reasoner [17] to compute the semantic relationships between service advertisements and user requirements. The experimental results have shown that the DAML+OIL matchmaker could cope with large scale e-commerce applications [40]. However, the limitation of DAML+OIL matchmaker is that it cannot deal with the service advertisements and requirements without semantic information because the DL logic does not support semantic information matching [38].

2.2.4 Semantic Service Matchmaker

Kawamura et al. [38, 48] designed and developed a search engine for Web services called Semantic Service Matchmaker in Figure 2.4. It enhances the discovery facilities of UDDI to make use of semantic information. The matchmaking process of the Semantic Service Matchmaker is based on the LARKS algorithm and DAML+OIL logic [47]. In the matchmaker, they provide a description of the WSSP, an extension of WSDL to describe capabilities of Web services. In the service discovery mechanism of semantic service matchmaker, it allows the service consumer to specify how closely match between user requirements and service advertisements [38]. Since the Semantic Service Matchmaker is developed based on the LARKS algorithm, it has the similar limitations of LARKS matchmaker. When using the Semantic Service Matchmaker, the user should clearly understand the five different filters such as *namespace filter, text filter, domain filter, I/O filter* and *constraints filter*, otherwise, the filters can be misleading and results cannot satisfy the user's requirements. The advantage of the Semantic Service Matchmaker, comparable to the LARKS matchmaker, is that the Semantic Service Matchmaker employs a new description language for Web services representations. Moreover, the Semantic Service Matchmaker has not been experimentally evaluated.

2.2.5 OMM Matchmaker

The ontology-based matchmaker (OMM) in Figure 2.5 is developed by Andreas Harth, which is based on an asymmetric description [31]. The system employs RDF-based ontologies to semantically describe requirements and services. The OMM matchmaker provides the ability to describe properties of services and matching preference [31]. Matching between user requirements and service description is done by

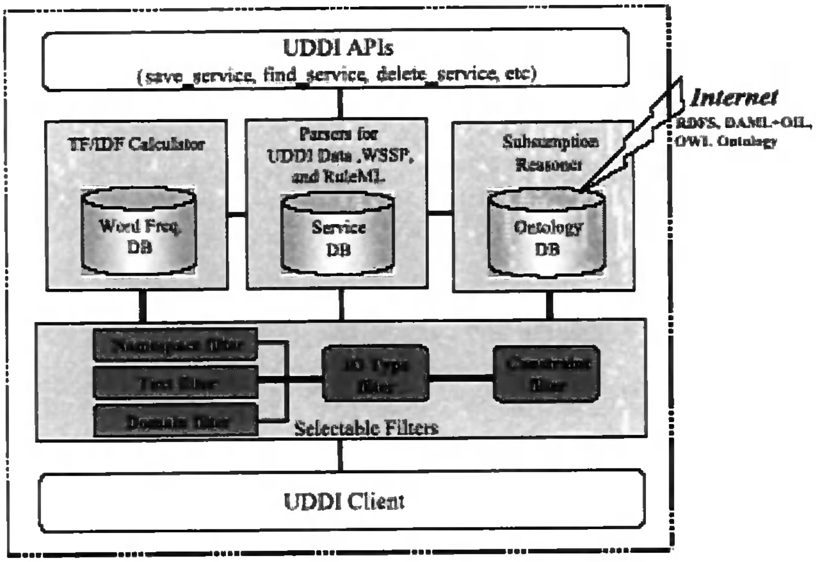


Figure 2.4: Semantic Service Matchmaker [38]

the OMM rules. The OMM rules are described by RDF. The OMM matchmaker also supports two kinds of matching mechanisms including a bi-lateral matching and gang-matching, which allow both resource providers and consumers to specify their matching constraints [31]. The limitations of OMM matchmaker are: 1) the OMM matchmaker does not define any semantic relationships between users requirements and service providers advertisements, which could make it difficult for the consumers to distinguish which services exactly fulfil their requests, or which services provide more capabilities than requested; 2) the OMM matchmaker cannot deal with service descriptions and user requirements without semantic information; 3) the OMM matchmaker has not been experimentally evaluated.

2.3 Discussions

In this chapter, we had surveyed a number of existing Web service description frameworks and discovery systems. Service description is critical to Web service application development and needs to represent the capabilities of Web services. However, current approaches such as WSDL and UDDI are designed to provide descriptions of message transport, interface, physical attributes in terms of name and address with

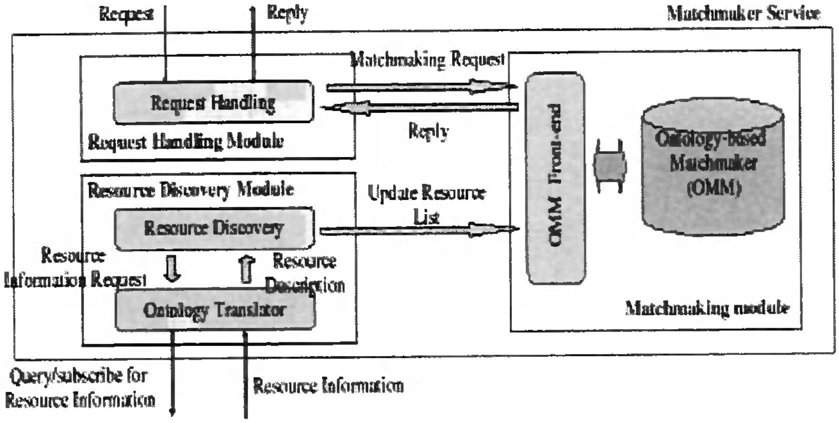


Figure 2.5: OMM matchmaker [31]

human-understandable attributes. Both WSDL and UDDI cannot represent service description information in unambiguous and computer-understandable forms. Due to service providers and service consumers may have very different background and knowledge, computers cannot understand their representations like human being. RDF is widely used in Semantic Web and it could authoritatively represent resources based on their capabilities. However, it is utilised for general resource description; there is no specific ontology and description frameworks for Web services applications. OWL-S and DAML-S provide better solutions for describing Web services with their capabilities, they have special service description framework and are represented by prevailing the Semantic Web languages such as OWL and DAML. OWL-S and DAML-S describe the service capabilities in unambiguous and computer-understandable forms, comparable to WSDL and UDDI. However, a number of essential properties representing Web services are out of the frameworks of OWL-S and DAML-S such as the business class and user class, both of them have significant influence on the service representation and discovery quality. For example, when two consumers want to search a mathematics book through the Web service, if one is a mathematician, who might need a more academic-based book rather then a basic one, though they tend to type the same search terms.

The discovery mechanism provided by UDDI is still exact "keyword-based" search on the names of both business and service through general industrial category. It is a problem that service consumers have to mimic the service providers' understanding of how their services shaped into the categories that they are using because the service providers and service consumers possess different background and knowledge; and do not share the same taxonomy of domain. The semantic service matchmaker can provide service discovery based on the service capabilities with semantic description, which can find the similar services by measuring the semantic distances between advertisements and requirements. The semantic service matchmaker employs OWL-S to represent service, but the weakness of OWL-S limits their service discovery quality. The LARKS matchmaker is a former version of semantic service matchmaker. Horroks has developed a framework for matchmaking based on Semantic Web technology like DAML-S. Unfortunately, a number of essential factors representing Web services are out of the framework of DAML-S and it cannot deal with the information without semantics. The limitations of the OMM matchmaker is that they use RDF to represent services and RDF is too general to describe the capabilities of Web services.

To the end, we found that existing service description frameworks are not enough to represent sophisticated Web services and the functionality of current service discovery systems for locating web services are limited on current description frameworks. Therefore, we need to identify the requirements and essential factors that a Web service description framework should have and propose a semantic rich modelling framework to describe Web services capabilities in unambiguous and computer-understandable forms with ontology. Another important goal for the the new description framework is to integrate with current Semantic Web markup languages so that which can be easily accessed and understood by the computers. Moreover, a new matching algorithm is necessary to calculate the semantic relationships between user requirements and service advertisements with the new description framework. In the next following chapters, we will introduce our novel description framework for Web service description and matchmaking in detail.

Chapter 3

Semantic Description Framework and Ontology Design

In the previous chapter, we have described the existing service description frameworks and discovery systems that were developed by different research organisations and computer vendors. As we discussed, service providers and service consumers may have very different background knowledge. Most of current description frameworks and description languages [11, 14] cannot represent sophisticated business and service capabilities in Web service environments in unambiguous and computer-understandable forms. To address this problem, a few research [9, 43, 5] developed knowledge-based description frameworks for Web services. However, a number of essential factors are out of these work. To address the above problems, a new description framework and knowledge-based metadata are necessary to develop for representing business and service capabilities of Web services in a formal sophisticated ontology-based manner.

Currently, a number of research are focusing on developing formal ontologies for specifying content-specific agreements for a variety of knowledge-based activities [21, 16, 25, 28, 5]. In this thesis, we will develop a description framework and ontology for the Web services only. To build a description framework, it is necessary to go in-depth analysis of business and service conceptions and requirements, and then identify the essential factors that a Web service description framework should have. With these work, we build an abstract business-service-user (BSU) description framework for representing business, service and user in Web service environments. With our BSU framework, we describe our novel work on designing and developing a metadata management technique, which focuses particularly on the business and services sharing in Web service environments. Differing with all existing service description frameworks for Web services, BSU takes into account five new modules including *business module, user module, business-service relationship module, user-service relationship module* and *user-business relationship module.* Our initial work, an early version of BSU framework and ontologies have been published in the UK e-Science Conference 2004 [51]. The work described in this chapter is an extension of that published description framework. This chapter is organised in the followings. Section 3.1 describes the terminologies, requirements and BSU framework. Section 3.2 represents the ontology design of BSU framework and the chapter is concluded in Section 3.3.

3.1 Semantic Description Framework

In this section, we will analyse the concepts of business, service and user in Web service environments and then give the formal definitions of the three components in Section 3.1.1. Section 3.1.2 discussed the requirements and aspects that a Web service description framework should have and Section 3.1.3 represents the BSU framework in detail.

3.1.1 Terminology

In this subsection, we will analyse the concepts of business, service and user in Web service environments and then give the formal definitions of the three components in our framework. Within the existing Web service model, a variety of research organisations give the definitions of service in Web service environments. There is no work about business and user description. Both of them are important factors in Web services. The importance of business has been identified in UDDI registries [20, 11, 22]. In fact, the same service can be available from the different business organisations. In UDDI, when users employ the service, the users need to chose which business to communicate. Moreover, users have significant influence on the service representation and discovery quality. For example, when two consumers want to search a mathematics book through the Web service, if one is a mathematician, who might need a more academic-based book rather then a basic one, though they tend to type the same search terms. In this section, we will analysis the business and user concepts proposed by different researchers in-depth and then give formal definitions of business and user which are particular for the business and user in the BSU framework.

A number of business definitions have been proposed in the last few decades for the different contexts. In reference [45] Shelton defined that "Business objects are abstracts that represent a person, place, thing or concept in the business domain. They package business procedures, policy and controls around business data. Business objects serve as a storage place for business policy and data, ensuring that data is only used in a manner semantically consistent with business intent." In Shelton's definition, the business model is specialised into two subcategories, namely *business entity objects* and *business process objects*, which represent people, places, things and business processes.

In reference [32], the business component approach to distribute system development is introduced. The concept of business is named Business Component and is defined as "the information systems representation, from requirements analysis through deployment and run-time, of an 'autonomous' business concept or process. It consists of all the software artifacts necessary to express, implement and deploy the given autonomous business concepts as an equally autonomous, reusable element of a larger information system."

From the above definitions, several important factors can be derived and given in the followings:

- A *Business* is a person, place, company or thing.
- A *Business* stores abstract services, business policies and business processes.
- A *Business* consists of properties that are needed to represent a given business concept.

The definition should be highly abstract and the implementation details, as well as syntactic and dynamic aspects can safely be ignored for the purpose of defining business in the BSU framework. According to the above three important factors, we integrate them and then give the business definition in the BSU framework.

Definition 3.1 Business

A business in Web services is an object abstraction that consists of properties needed to represent a person, place, thing or process in a given business domain. The business can provide abstract services, business policies and business process in Web service environments.

Moreover, there are a number of service definitions in existing systems. In the followings, we deeply analyse several service definitions from the available literature, and then give a formal definition of service that is used in the BSU framework. Microsoft defines the service is a application logic that is programmatically available [36]. IBM gives a definition of service based on their service model. In their definition, "services are modular applications that perform a specific business task and conform to a particular technical format" [26]. As a leader of World Wild Web, W3C states that service is a software object, "meant to interact with other components, encapsulating certain functionality or a set of functionality. A component has a clearly defined interface and conforms to a prescribed behaviour common to all components within an architecture." [13]

From the above definitions, several important factors can be derived and given in the followings:

- A *Service* is abstract.
- A *Service* is self-contained, self-describing and modular application
- A *Service* can be published, located, and invoked across the Web.
- A *Service* presents a capability to perform tasks

According to the above four important factors, we integrate them and then give the service definition in the BSU framework.

Definition 3.2 Service

Service is an abstract resource that can be a self-contained and self-describing application. Services perform functions to finish the user requirements. It can be published and retrieved through the Web and used by other applications.

Although business and services are two important concepts in Web service environments, users have significant influence on the business and service discovery quality. Unfortunately, the existing Web service frameworks [11, 14, 9, 43, 5] do not provide any description information for the user model. In the BSU framework, we will give a formal definition of user.

Definition 3.3 User

A user in Web services is an object abstraction that consists of properties needed to represent a person, company, or agent in a given business service domain.

3.1.2 Requirements of BSU Framework

In this subsection, we will discuss the requirements of BSU framework. Since a number of essential factors such as service operation, service management, and service resources do not mention in any current description frameworks of Web services, which are important for business and service discovery quality in Web service environments. To address this problem, the BSU framework needs to provide our initial work on representing the following factors for Web service.

- **Object Relationship**. The relationship describes an association between two objects such as *business and service, service and user, user and business*. The relationships in the BSU framework are expected to maintain a set of identifying attributes and ontology information for the related objects.
- **User Description**. The users description will significantly influence the retrieval quality. In fact, different kinds of services are suitable for different users due to users have the different background knowledge. Therefore, users and their requirements are important parts in the BSU framework.
- **Service Resource**. Service resource is an important factor that is required concerning interactions between services. However, existing Web service description frameworks do not provide any representation of service resources. The BSU framework will give a clear definition of service resource in Web service environments.
- **Service State Information** Service consumer and providers are expected to manage, monitor and control the services in the systems. In Web service systems, the service monitoring is determined by the service state information [44]. Some service states are common to all actions, others are specifying to the task. Therefore, in the BSU framework, service state information should be defined and supplied in service execution.

3.1.3 BSU Framework

Service consumers should locate, invoke and use services which provide particular resources. In the previous subsection, we discuss the requirements and essential factors of BSU framework, and we will represent the BSU framework in the followings. Figure 3.1 gives a sketch of the architecture of BSU framework.

- What do *Business* and *Service* require of the *User*, and provide for them? The answer to this question is given in the "**Content**". Thus, the class *SERVICE* presents a *SERVICECONTENT*, and the class *BUSINESS* presents a *BUSINESSCONTENT*.

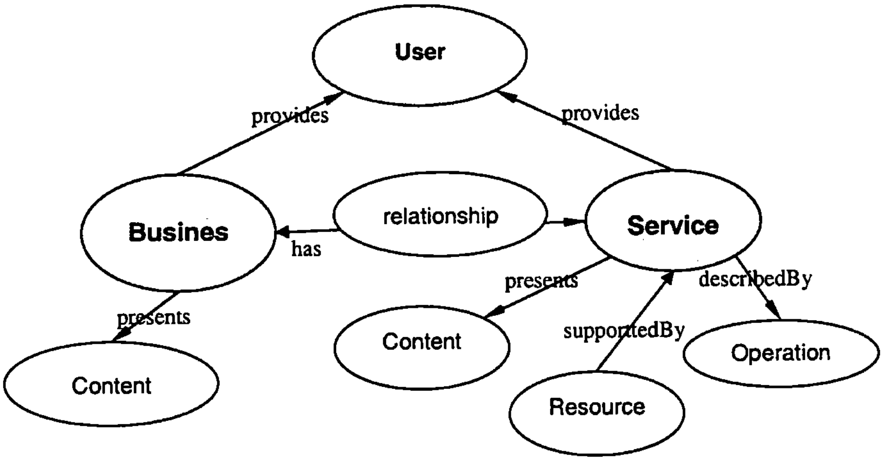


Figure 3.1: the Business-Service-User Framework

- How does *Service* work and how *Service* can be controlled? The answer of this question is shown in the "**Operation**". Thus, the class *SERVICE* is describedBy a *SERVICEOPERATION.*

- What kinds of resources the *Service* provides? The answer for this question is described by the "**Resource**". Thus, the class *SERVICE* is supportedBy a *SERVICERESOURCE.*

- What relationships between *Business* and *Service*? The answer for this question is represented by the "**Relationship**". Thus, the class *SERVICE* and *BUSINESS* have a *RELATIONSHIP.*

In the following sections we will represent the ontology design of the BSU framework including *business content, user, relationship, service content, service operation* and *service resource.*

3.2 Ontology Design

In the previous sections, we gave the formal definitions of Business, Service, User, and present the requirements and essential factors of BSU framework. Moreover, we describe the BSU framework in detail. In this section, we will present our effort to create ontologies of the BSU framework focusing particularly on the problems and needs of business and service discovery in Web service environments. "An ontology is an explicit specification of a conceptualisation, which presents a set of concepts within a domain and the relationships between those concepts [29]". Designing ontology typically consists of three steps [30]. The first step is to group together related resources in order to create an ontology. The second step is to define various meaningful properties for each class of the ontology. The third step is to write the class and properties into the semantic markup languages such as RDF, RDFS and OWL. In the following subsections, we will design ontologies of BSU framework.

3.2.1 Ontologies

Business Ontology

We develop the following hierarchical classification of classes to group the class of the business model in Figure 3.2. The classes presented in Figure 3.2 is explained in the followings. There are four major groups of Business resources that need to be classified, which are are *businessContent, relationship, businessUser* and *service.* The **businessContent** class consists of a set of subclasses, which includes some human-readable information such as *businessName, busienssDescription* and *contactInformation*, and machine-understandable information such as *businessCategory* in Table 3.1 and 3.2. The contents of businessContent class are similar to the business description information in UDDI registry [11], however, the significant difference is that we employ knowledge-based ontology description instead of attributed-based description of UDDI. We have define some **relationships** between *business to business* and *business to service.* Table 3.3 represents the contents of relationship in detail. The **businessUser** class describes a people or an agent that interacts with a business to request a service to performance the task. The detail information of user is described in Table 3.4.

- **businessName** presents the name of business and it is used as an identifier of business.

- **businessDescription** is a human-understandable description of business and summarises what the business offers.

- **contactInformation** provides the contact information of business (shown in Table 3.1).

- **businessCategory** provides the categories of business, which refer to an ontology of business in Table 3.2.

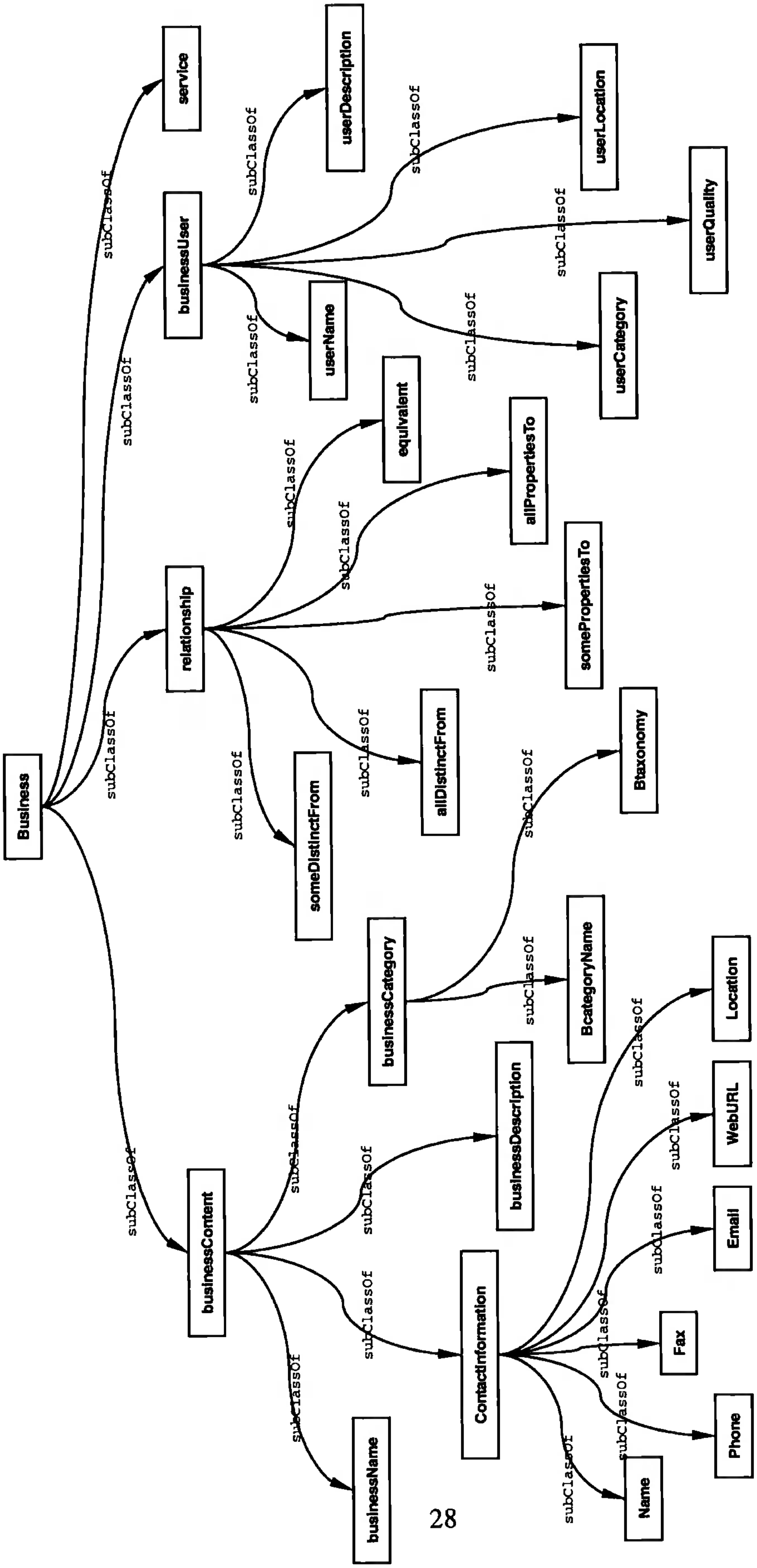


Figure 3.2: Business Ontology

Table 3.1: businessContact Class

Name	Description
Name	The name of business
Phone	A phone number of business
Fax	A fax number of business
Email	An e-mail address of business
Location	A physical address of the business
WebURL	A URL of the business website

Table 3.2: businessCategory Class

Name	Description
BcategoryName	is the name of business category, which can be just a literal, or perhaps the URI of business
Btaxonomy	stores a reference to the taxonomy scheme. It can be the ontology of taxonomy

Service Ontology

There are six major groups of service information in the BSU framework that need to be classified. These groups are *serviceContent, serviceOperation, serviceManagement, serviceResource, serviceUser* and *relationship.* The **serviceContent** class in the BSU framework represents "what the service does", that is, it can give the different types of information needed by service consumers to determine whether the service meets their needs or not. The **serviceContent** class consists of a set of subclasses, which include some human-readable information such as *serviceName, serviceDescription* and machine-understandable information such as *serivceCategory and serviceFunctionality.* The **serviceOperation** class tells "how the service works and how the service is controlled", that is, it describes what happening when the service is carried out. The **serviceManagement** class describes the monitoring behaviours for each specific task. The **serviceResource** defines a family of resource status information performed by services. The **serviceUser** class describes a person or an agent that interacts with a business to request a service to performance the task. The **relationship** describes the association between *service to service* and *service to business.* We develop the following hierarchical classification of classes to group the class of the service model in Figure 3.3. The description of classes will be described in the followings.

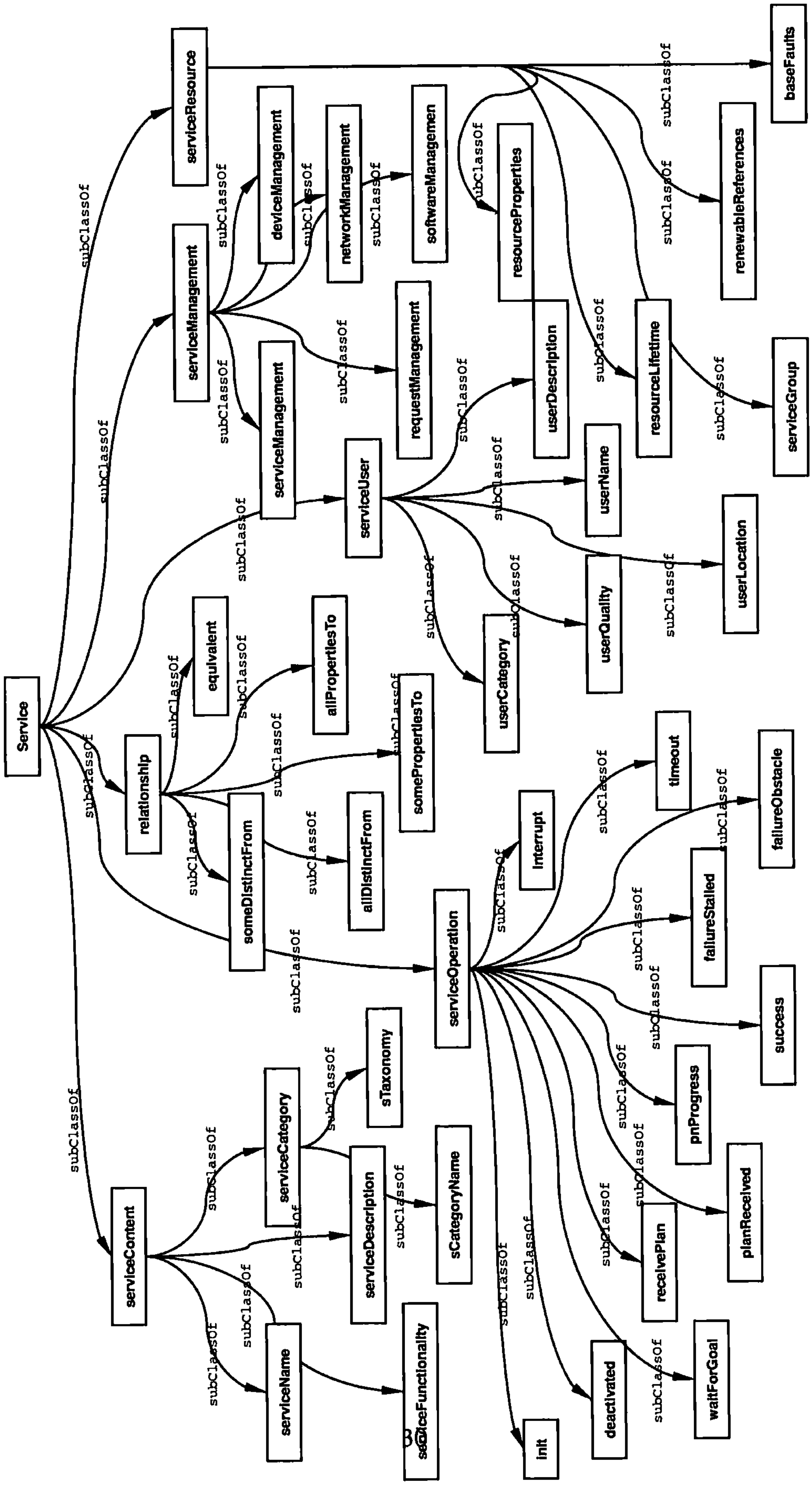


Figure 3.3: Service Ontology

Table 3.3: relationship Class

Name	Description
someDistinctFrom	Two business entities can be described to be someDistinctFrom, which means both business entities have some services and other properties same. *(business to business)*
allDistinctFrom	Two business entities are stated to be allDistinctFrom, which means both business entities do not share any properties in common. *(business to business)*
equivalent	Two business entities can be stated to be equivalent, which means both of them share the same services and properties. *(business to business)*
somePropertiesTo	A business may have a restriction on a property that at least one value for that property is of a service. Therefore, the service has at least one of properties belongs to the business. *(business to service)*
allPropertiesTo	It means that a service inherits all the properties from the business.*(business to service)*

The serviceContent gives the detailed information of what the service does, which consists of a set of sub-classes such as *serviceName, serviceDescription, serivceCategory* and *serviceFunctionality.*

- **serviceName** refers to the name of the service and it can be used as an identifier of the service.
- **serviceDescription** provides a brief description of service. It summarises the capabilities that the service offers.
- **serviceCategory** describes the categories of services in Table 3.5.
- **serviceFunctionality** is the specifications of functionality that the service is providing. The function descriptions of service contain five components including the *input, output, precondition, effect* and *operation* in Table 3.6.

serviceFunctionality specifies what results obtained from the service, including the expected and unexpected results of the service activity. We extend the functionality model in OWL-S [5] to represent the serviceFunctionality in the BSU framework with the additional *operation* sub-class. In BSU, the *serviceFunctionality* includes two aspects: the information transformation (represented by *input, output* and *operation*) and

Table 3.4: businessUser Class

Name	Description
userName	refers to the name of user who is using the business. It can be used as an identifier of user.
userCategory	describes the category ontology of user
userDescription	provides a human-understandable description of user.
userLocation	is a physical address of user.
userQuality	represents the evaluation of the user behaviours in the past service transactions. The user quality heavily affects the the selection of the business.

Table 3.5: serviceCategory Class

Name	Description
sCategoryName	is the name of category, which can be just a literal, or perhaps the URI of the process parameter (a property).
sTaxonomy	stores an ontology of service taxonomy.

the state change produced by the execution of the service (represented by *precondition* and *effect*). The extensively description of precondition, effect, input and output have been discussed in the literature [5]. The new subclass *operation* describes what is happening when the service is working. Here is an example of serviceFunctionality, to complete the sale, a book-selling service requires a credit card number and expiration date as input, but also the precondition that the credit card actually exists and is not overdrawn. The operation is the payment transaction and book reservation. The result of sale is the output of a receipt that confirms the proper execution of the transaction, and as effect the transfer of ownership and the physical transfer of the book from the the warehouse of the seller to the address of the buyer.

serviceOperation: the serviceOperation of the BSU model tells "how the service works and how the service is controlled"; that is, it describes what happening when the service is carried out. Web service systems are dynamic systems, where the properties of services change during the life time of the service. The properties are categorised as factual properties defining the state of the system, and behavioural properties defining the possible operations of the service [42]. At any time, the set of all factual properties defines the current state of the system. An aggregation of operations is termed a process, thus, a process can be divided into operations [42]. However, existing Web service systems do not provide any description information of the service operation.

To address this problem, we employ the system control model [27] to the Web service environments, in which there are eleven subclasses in serviceOperation including *init, deactivated, waitForGoal, ReceivePlan, InProgress, Success, FailureStalled, Failure-Obstacle, TimeOut* and *Interrupt*. Table 3.6 gives the definitions of these subclasses in detail.

Table 3.6: serviceOperation Class

Name	**Description**
init	describes a service that is initialised the process
deactivated	describes a service waiting for activation
waitForGoal	describes a service waiting for target from cognitive controller.
receivePlan	describes a service receiving a plan
planReceived	a plan received for services
pnProgress	service execution in progress
success	service execution terminated successfully
failureStalled	service motors stalled
failureObstacle	service blocked by obstacle
timeout	service execution timeout
interrupt	service execution interrupted

serviceManagement: The nature of Web services offers a number of new challenges for service management compared with the traditional systems due to heterogeneous, dynamics and distributed. Web services management can be seen as a new layer of functionality on the top of a conventional management framework that can map devices and software infrastructure to the services. According to the definition of Web services and UDDI registry, we found that the following components needed to be managed such as services, resources, networks and requests. Therefore, we define abstract ontologies to present the service management classes of Web services in Table 3.7.

serviceResource: one of the main distinguishing characteristics of the Web services is that Web services need resources to support the service (e.g., data, instruments, computers, humans, etc.) [24]. When a resource supported to the service, it must be uniquely identified and, in many cases, have associated metadata to describe it. However, existing Web services description frameworks do not address the requirements

Table 3.7: serviceManagement Class

Name	Description
serviceManagement	involves systems management at the service level. It is the application of service operation policy at the point of receiving a service request.
resourceManagement	involves systems management at the resource level. Resource components are employed in the service process
networkManagement	involves systems management at the network level, such as network protocol.
requestManagement	involves systems management at the user requirement level.

for resource identifications and associated metadata. To address this problem, we integrate the WSRF [37, 10] framework into the BSU framework for representing the service resources. The **serviceResource** defines a family of specifications for accessing resources using Web services, and service resources have their own state information in terms of *resourceProperties, resourceLifetime, baseFaults,* and *serviceGroup* specifications in Table 3.8. The detailed description of WSRF has been described in the reference [10].

Table 3.8: serviceResource Class

Name	Description
resourceLifetime	mechanisms for describing resources, lifetime for user operation
resourceProperties	definitions of resource properties.
renewableReferences	a conventional decoration of a addressing endpoint reference with service policy information
serviceGroup	state a group of resources
baseFaults	a base fault type for service operation when return faults in a message exchange.

relationship describes an association between *service to service* (S2S) and *service to business*(S2B), based on attributes of two entities in Table 3.9. A relationship maintains a set of identifying attributes and ontology information from the related entities. In the BSU framework, services can generate some attributes and ontology information from the other services and businesses when they have some relationships.

serviceUser:is an important part of service, the user description will heavily influence the quality of requirements matching. We define serviceUser ontology for service

Table 3.9: relationship Class

Name	Description
someDistinctFrom	Two services entities may be stated to be someDistinctFrom, which means both services entities have a few same properties *(service to service)*
allDistinctFrom	Two services entities are stated to be allDistinctFrom, which means both service entities do not share any properties in common. *(service to service)*
equivalent	Two service entities can be stated to be equivalent, which means both of them share the same properties and other properties.*(service to service)*
somePropertiesFrom	A service may have a restriction on a property that at least one value for that property is of a business. Therefore, the service has some properties of the business. *(service to business)*
allPropertiesFrom	It means that a service inherits all the properties from a business.*(service to business)*

in Table 3.10.

Table 3.10: serviceUser Class

Name	Description
userName	refers to the name of the user who is using the service. It can be used as an identifier of user.
userCategory	describes the category ontology of user
userDescription	provides a brief description of the user.
userLocation	is a physical address of the user.
userQuality	represents the evaluation of the user behaviours in the past service transactions. The user quality is heavily affect the the selection of the business.

3.2.2 Defining Properties

In the above sections, we have define the ontology classes of businesses, services and users, where user ontology is integrated in the business and service classes. Ontology classes are built to group together similar resources. However, the class hierarchy does not give any information about the classes themselves. To address this problem, we define various meaningful properties for each class of the ontology, which is the second step to design the ontology [30]. In the thesis, we do not provide the full properties of the class of BSU framework due to one year time limitation of MSc course. The full design and development of properties of BSU framework has been left to the future work. Figure 3.4, sketch the properties of part of service that links the BSU framework.

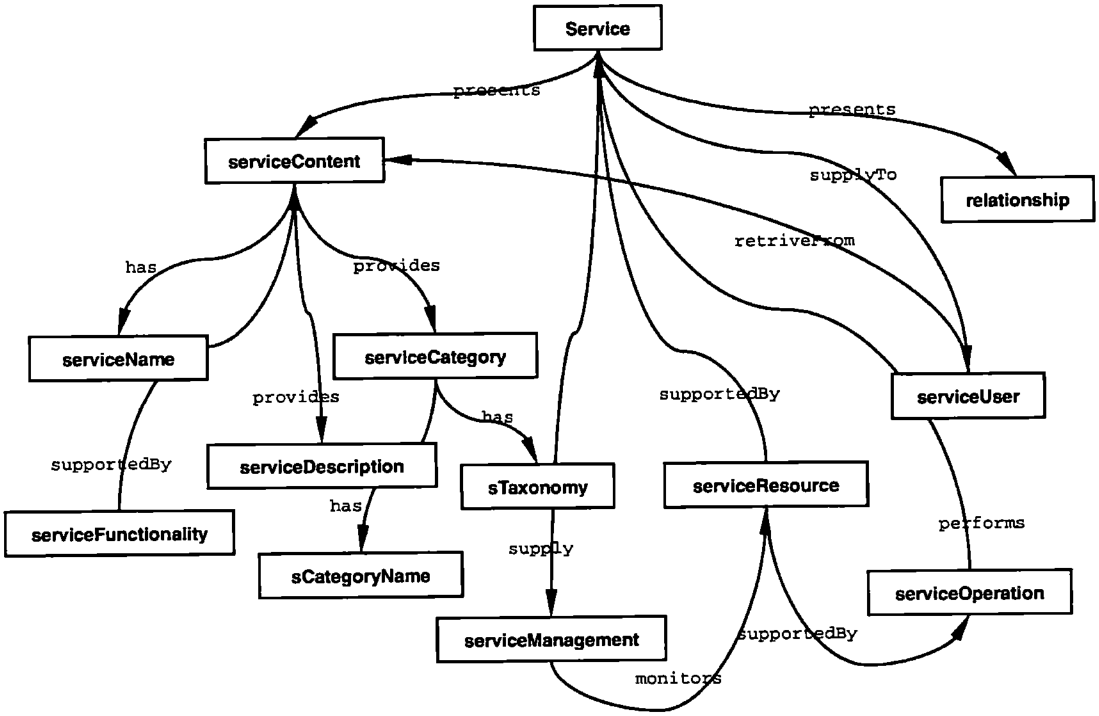


Figure 3.4: Service Properties

3.2.3 OWL-BSU

The third step to design an ontology is to write the classes and properties with the semantic markup languages [30] such as RDF or OWL. The purpose to generate the ontology and metadata instances is that a software agent needs a computer-interpretable description of the business, service and user, and the means by which it is accessed. An important goal for the BSU framework is to integrate with current Semantic Web markup languages so that which can be easily accessed and understood by the computer. Our approach is to use OWL abstract syntax to represent the BSU framework and ontologies. Actually, the BSU framework is a conceptual model that can be described by many different Semantic Markup Languages such as RDF, DAML and OWL. The reason we choose OWL is that OWL is the most advanced semantic markup language at the moment. Through the tight connection with the OWL logic [4, 35] that allows for reasoning on concept taxonomies and the definition of relations between concepts. This subsection describes our novel work of using OWL to represent the BSU framework and the new language schema called OWL-BSU which is specific for the description of business, service and user in Web service environments [51]. In this chapter, we will give two examples of OWL-BSU including user and relationship, which do not appear in any of current Web service description frameworks. The full version of OWL-BSU is presented in the Appendix.

Example of User Name The user name refers to the name of the service that is being offered in Figure 3.5.

```
<owl:DatatypeProperty rdf:ID="userName">
  <rdfs:domain rdf:resource="#User" />
 <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
  </owl:DatatypeProperty>
<owl:Class rdf:about="#User">
  <rdfs:comment>A User can have only one name</rdfs:comment>
<rdfs:subClassOf>
<owl:Restriction>
  <owl:onProperty rdf:resource="#userName" />
 <owl:cardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInte
ger">1</owl:cardinality>
  </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Figure 3.5: Example of User Name

Example of Relationship (business to business) A relationship maintains a set of identifying attributes and ontology information from the related entities. Most relationships simply relate the objects of one entity to those of another by comparing attribute values and ontology information between them. Business to Business has three relationships someDistinctFrom, allDistinctFrom and equivalent in Figure 3.6.

```
<owl:Class rdf:ID="Relationship">
  <rdfs:comment>
      The relationship of business to business
  </rdfs:comment>
  <owl:disjointUnionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#SomeDistinctFrom"/>
    <owl:Class rdf:about="#AllDistinctFrom"/>
    <owl:Class rdf:about="#Equivalent"/>
  </owl:disjointUnionOf>
</owl:Class>
```

Figure 3.6: Example of Relationship

3.3 Summary

In this chapter, we build an abstract BSU framework with in-depth analysis of business and service conception. With our BSU framework, we describe our novel work on designing and developing a metadata management technique, which focuses particularly on the business and services sharing in Web service environments. Differing with all existing service frameworks for Web services, the BSU framework and ontology take into account five new modules including business module, user module, business-service relationship module, user-service relationship module and user-business relationship module. An early version of BSU framework and ontologies have been published in the UK e-Science Conference 2004 [51].

Chapter 4

Importing OWL-BSU Ontologies into UDDI and Semantic Matching

In the previous chapter, we have defined the BSU framework and OWL-BSU which can present the framework in a computer-understandable way. In this chapter, we present our work which will integrate these approaches with the existing UDDI registry so that UDDI can represent a Web service with semantic description information. The reason of integration of OWL-BSU and UDDI is that we take the advantage of the UDDI popularity and OWL-BSU semantic support into a single, uniform architecture. In the followings, we will introduce how to integrate OWL-BSU with the current UDDI registry. However, the current UDDI query mechanism supports simple keyword exact match only, it does not work properly with semantic information (e.g., ontology) which is represented by OWL-BSU. Therefore, a new query mechanism, specifically, a new matchmaking algorithm that supports semantic information match is necessary. To address this problem, we design a simple and effective semantic matching algorithm. This chapter is organised as followings: Section 4.1 discusses how to integrate OWL-BSU into the UDDI registry. In the section 4.2, we will introduce a semantic matching mechanism between service advertisements and user requirements. This chapter will be concluded in Section 4.3.

4.1 Importing OWL-BSU into UDDI

The methodologies of integration and implementation of OWL-S description into the UDDI registry have been discussed in the literature [47]. In the thesis, we will employ their methodologies to import OWL-BSU description into the UDDI registry.

Section 4.1.1 gives a detailed representation of UDDI. Section 4.1.2 discusses the methodologies of importing OWL-BSU into UDDI.

4.1.1 UDDI representations

Before adding OWL-BSU semantic description information into the UDDI registry, it is necessary to give the details of the UDDI registry. The representation of services in UDDI [11] is shown in Figure 4.1. In UDDI, a business is stated as a BusinessEntity object which can store human-understandable information like business name and contact information such as the physical address. [20, 22, 11]. The *BusinessEntity* is associated with one or more *Business Services* which can be provided by the businesses. A Business Service is associated with one or more *Binding Templates* that can point out the service access end point.

The representations of businesses in UDDI support to describe the basic information of services, for example, how they are named, who to contact to gain information, how to invoke them [20, 22, 11]. However, they do not provide any information about the service type and descriptions of service capabilities [47]. The important thing in the UDDI is that UDDI can provide a data structured which is TModel allowing the specification of additional attributes of the entities described in the UDDI. The TModel is an identifier that is a unique key assigned to the TModel in form a UUID, and the TModel Locators specify information that is used within UDDI to control the use of the TModel [11].

4.1.2 Integrating OWL-BSU into UDDI

In the chapter, we import OWL-BSU description into the UDDI registry by using the methodologies of [47]. The mapping of OWL-BSU into UDDI Representations is shown in Figure 4.2 and 4.4. Due to the OWL-BSU and UDDI are both designed for Web services description and discovery, some of attributes of services designed in both models are similar and can be mapped directly from OWL-BSU to the UDDI registry. For example, businessName (OWL-BSU) to Name (UDDI), businessDescription (OWL-BSU) to Description (UDDI). Moreover, the OWL-BSU provides a solution for allowing semantic description of services such as ontologies. Therefore, some specific ontology information in OWL-BSU such as BCategory, relationship are represented by using the TModel mechanisms in UDDI.

In Figure 4.2, the description of the business in the BSU framework is mapped into

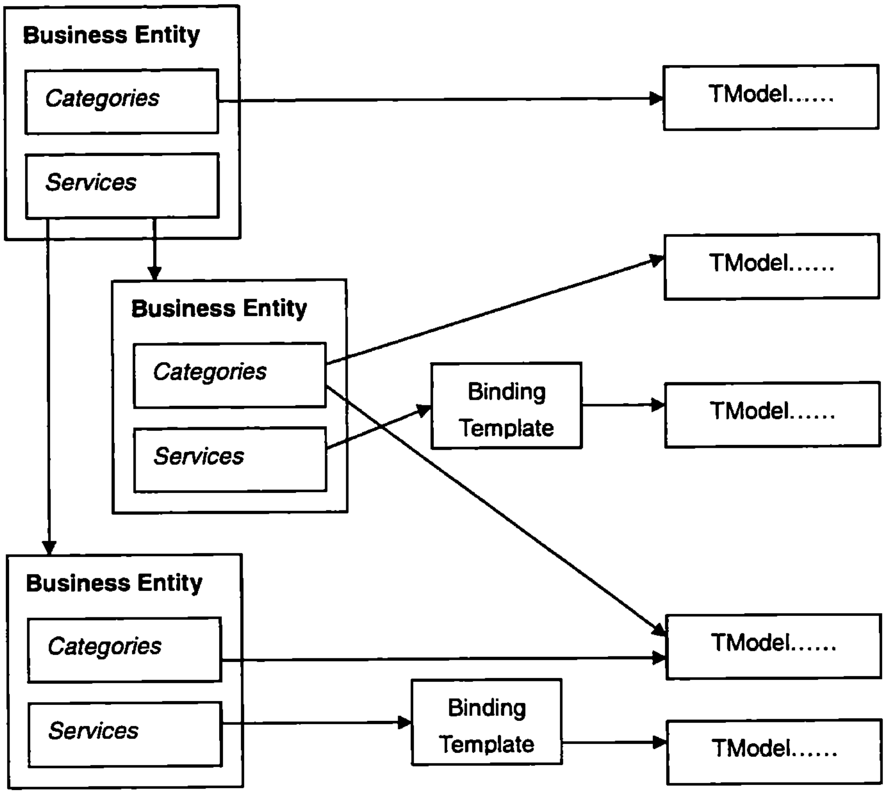


Figure 4.1: UDDI Service Representation [11]

an instance of the UDDI BusinessEntity that is used as a representation of the Business. If a business with the same information is already available in the UDDI registry that business is updated and referenced by the business service description, otherwise a new business instance will be created. The mapping of the other attributes of OWL-BSU requires the specification of a set of UDDI TModels, one for each attribute of the OWL-BSU representation. The BusinessService of UDDI records use these TModels to store the values of OWL-BSU semantic information. For example, one of the TModels, the relationship TModel has a special meaning; it states that the service advertised has an OWL-BSU service representation, and its value is the URI of the OWL-BSU service. After mapping, OWL-BSU can be completely embedded in UDDI.

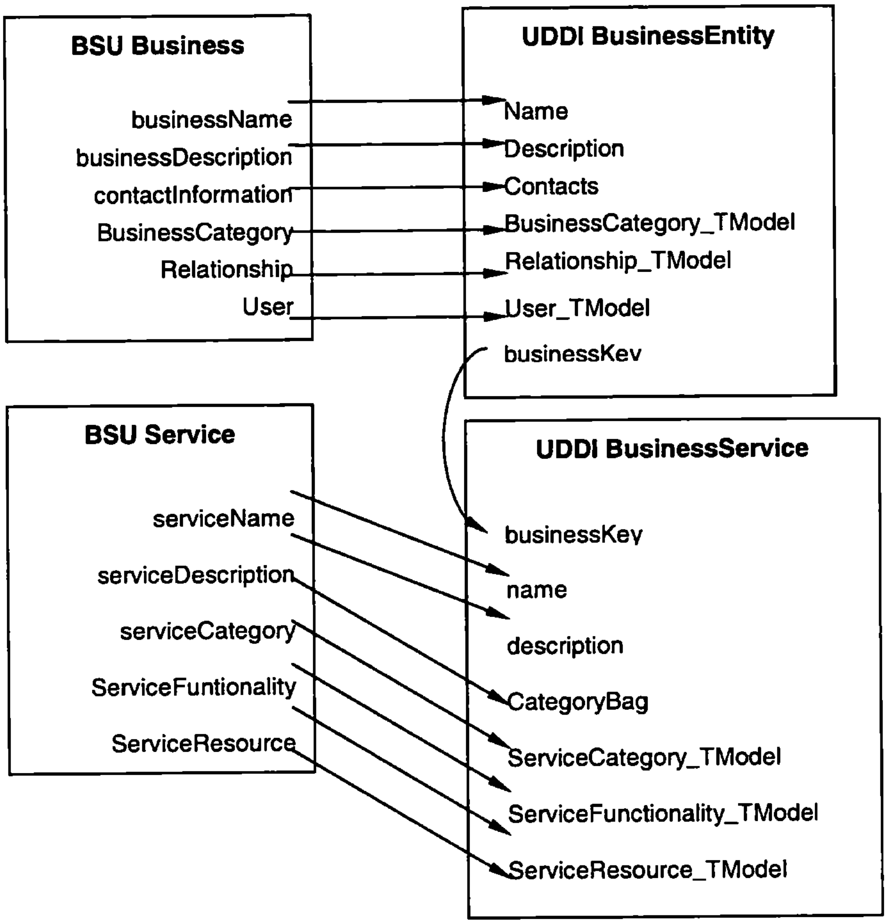


Figure 4.2: mapping from OWL-BSU to the UDDI representation

4.2 Semantic Matching

Although, the OWL-BSU description information has been imported into the UDDI registry, without an advanced matching algorithm the query mechanism in UDDI cannot calculate the relationships in ontology. Therefore, a new matching algorithm that support semantic ontology information is necessary. Currently, there are a few semantic matching algorithms, such as literature [50, 39, 41, 38] that can support semantic matching with ontologies. These algorithms can be used in our OWL-BSU ontology straightforwardly. However, according to the literature [23], the performance of current semantic matching algorithms is poor because of the complexity of algorithms. To address this problem, we propose a simple and effective semantic matching algorithm to calculate the semantic relationships between ontologies [51]. In the semantic matching algorithm, we define three semantic relationships between users requirements and providers advertisements, which are *exact match, full match,* and *part match* [51]. These semantic relationships are similar to the exact match, plug-in match and relaxed match in [38]. The major difference between their relationships and ours is that in their approach, the relaxed match contains the plug-in and exact matches, and the plug-in match includes exact match; in our approach, exact match, full match and part match do not contain each other and they are equal in the relation level. The interpretation of the semantic algorithm is shown in the followings.

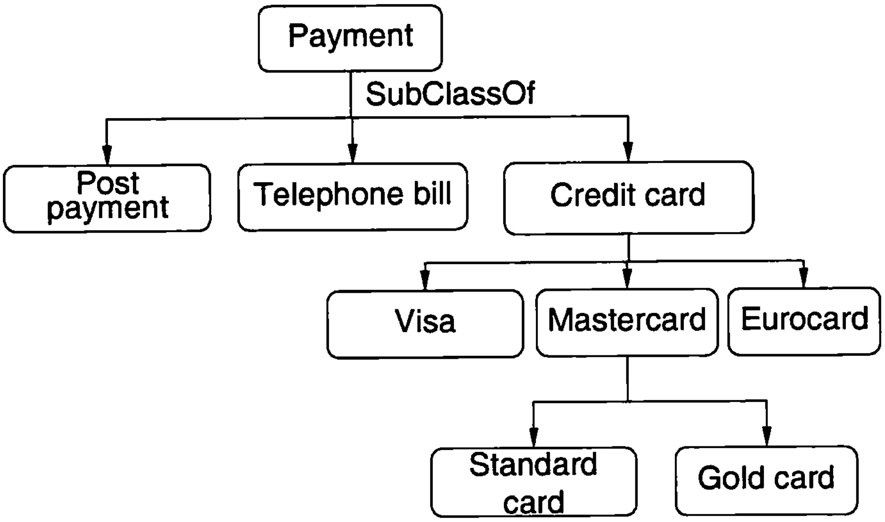


Figure 4.3: payment ontology tree

- **Exact match**: (*request.input = advertisement.output*) It is the highest degree of the semantic matching, when users requirements and providers advertisements are equivalent. For example, in Figure 4.3, the service provider registers the service with a MasterCard in the ontology of the payment. If a user requests a MasterCard service, both the advertisement from the service provider and users requirement are same, in this case, we call this situation exact match. This is the best case when a user is looking for the specific services and he/she exactly knows what he/she wants and can represent the requirement clearly [51].

- **Full match**: (*request.input < advertisement.output*) It is a situation that a service provides more capabilities than users request. An example of full match is the match between a request of Visa card payment and an advertisement of credit card payment because the Visa card payment is a sub-service of credit card payment in the payment ontology tree, shown in Figure 4.3. The registry will return the advertisements that are containing not only Visa card payment services but also the services that are registered as credit card payments. Some of the returned services can provide more capabilities than requested [51].

- **Part match**: ($request.input > advertisement.output$) It is a situation that the service can only fulfil a part of user's request. For example, if a user requests a service which can support "Credit card", and the only service available is one service that can support "MasterCard". Since "Credit card" is a super set of "MasterCard", which is shown in Figure 4.3, and "Credit card" contains two more kinds of cards except MasterCard. Therefore, the service registered as MasterCard cannot meet all of the user's request, it only can provide a part of capability of user's request [51].

Algorithm 1 semantic matching algorithm

```
domatch(request.input)
for all advertisements in Repository do
  if $request.input = advertisement.output$ then
    return exact match
  else if $request.input < advertisement.output$ then
    return full match
  else if $request.input > advertisement.output$ then
    return part match
  end if
end for
```

4.3 Summary

In this chapter, we have presented our initial work on integrating OWL-BSU into existing UDDI registry so that UDDI can represent a Web service with semantic description information. The reason of integration of OWL-BSU and UDDI is that we take advantage of the UDDI popularity and OWL-BSU semantic support into a single, uniform architecture. However, the current UDDI query mechanism supports simple keyword exact match only, it does not work properly with semantic information (e.g., ontology) which is represented by OWL-BSU. Therefore, we develop a simple and effective semantic matching algorithm to address this problem. In the next chapter, we will integrate these approaches together to implement a prototype which is a Semantic Web Service Search Engine and evaluate the retrieval performance of the search engine.

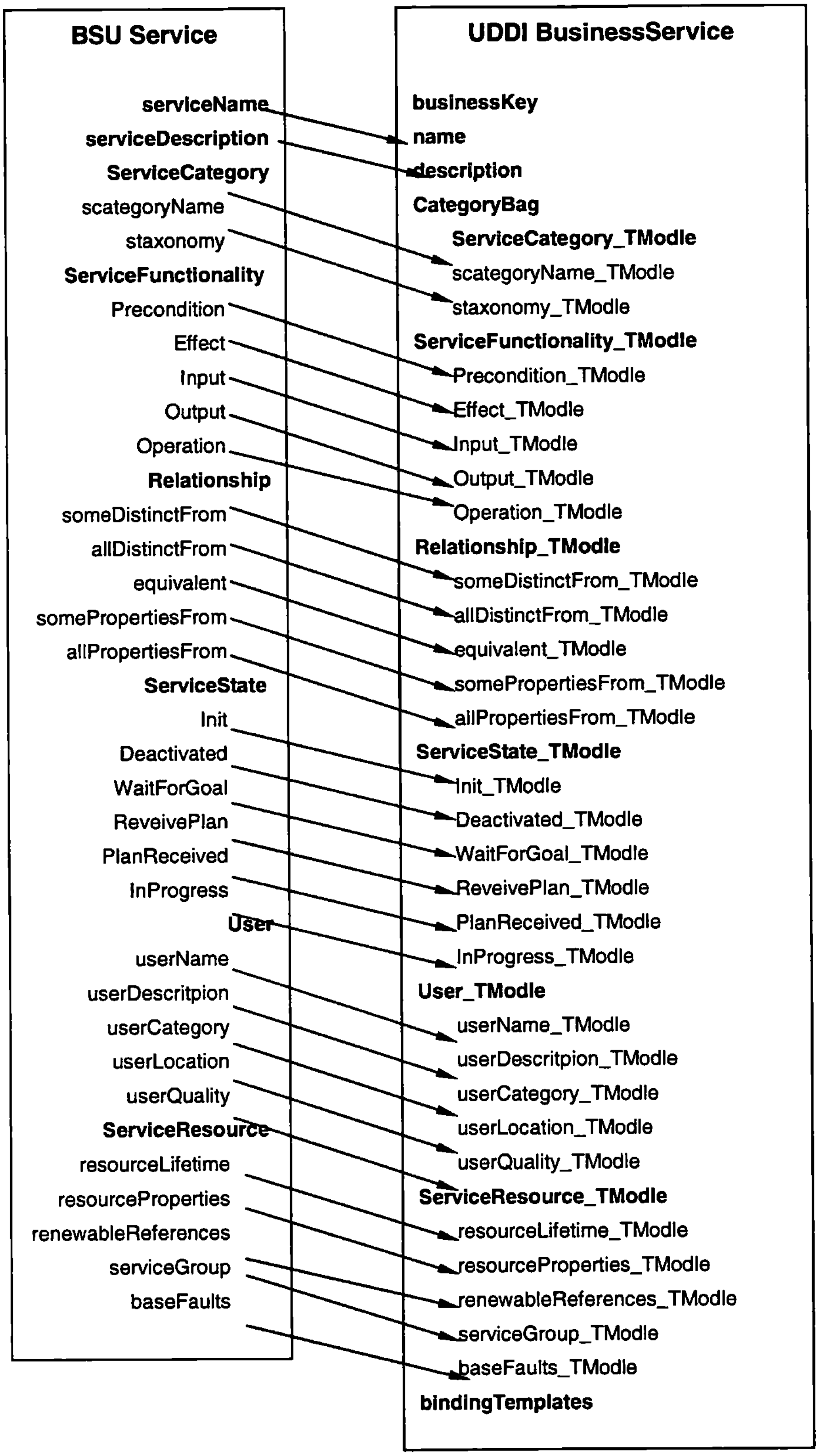


Figure 4.4: mapping from OWL-BSU to the UDDI representation

Chapter 5

Implementation and Evaluation

In this chapter, we will integrate our approaches which include the OWL-BSU and the semantic matching algorithm together to implement a Semantic Web services (SWS) search engine, and then evaluate the retrieval performance of the SWS search engine. This chapter is organised as followings. Section 5.1 represents the system architecture and working progress of SWS search engine. Section 5.2 describes the implementation details of the SWS search engine. The system will be evaluated in Section 5.3, also, the evaluation methodologies, metrics and the experimental results will be described and discussed. This chapter will be concluded in Section 5.4.

5.1 Semantic Web Services Search Engine

In this section, we will describe the architecture of SWS search engine, the business and service registration progress and the service discovery in principle.

5.1.1 Architecture

As we discussed in the previous chapter, OWL-BSU has been integrated with UDDI. The SWS search engine is a bridge between the service provider or service requester and the UDDI registry. The architecture of the SWS search engine is shown in Figure 5.1, which is similar to the work of Semantic Service Matchmaker [48, 38]. The significant difference is that in their architecture OWL-S is employed and our system architecture implements with OWL-BSU. The SWS search engine can be smoothly incorporated with any UDDI registry and empower the registries with OWL-BSU semantic description and processing. The SWS search engine is fitted into the triangle structure of Web services. In order to integrate the UDDI registry, the SWS search engine API will be designed the same as the UDDI API to facilitate the users with a seamless connection. Another benefit of integrating OWL-BSU with UDDI is that the SWS search engine can compile the current Web service standards and technologies, such as SOAP, and WSDL.

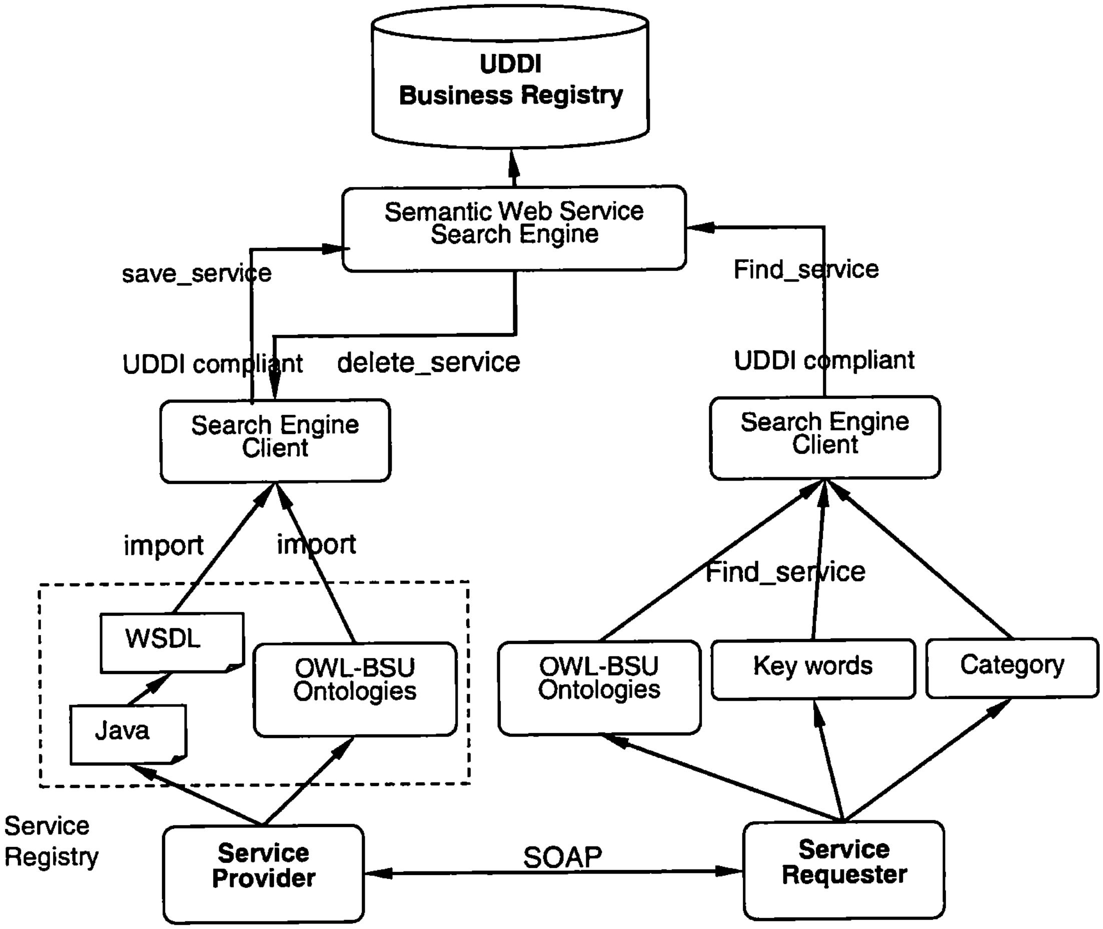


Figure 5.1: Architecture of the SWS search engine

5.1.2 Business and Service Registrations

Users need to register their services with the semantic information to the SWS search engine. In this subsection, we will show how the process of registration works. Figure 5.2 illustrates a Business Registration structure, where general information (some information without semantic description) is registered normally including *business name, business description, CategoryBag*, and *contact information*. The semantic information of Business registration will be different in the way that providers need to choose which pacific category their business belonging to.

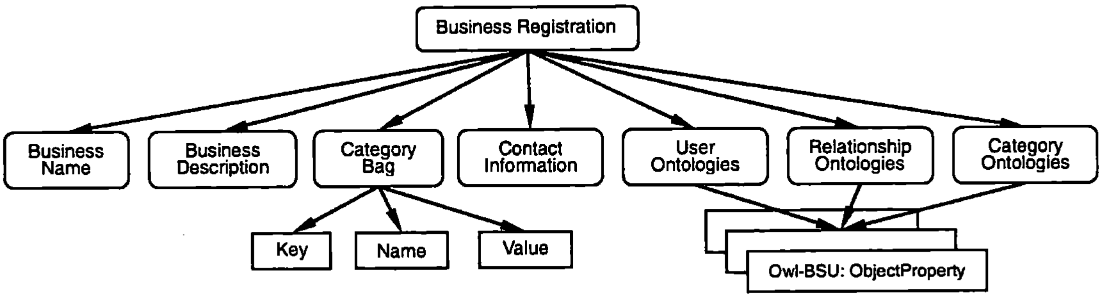


Figure 5.2: Business Registration Structure

After registering a business, the service provider should register a service of the business. A semantic description framework for service registration is illustrated in Figure 5.3, which is similar to the Business Registration. In the service registration progress, some attributes such as *service name, service description* and *CategoryBag* are registered without semantic information. Other information including *category ontologies, resource ontologies, relationship ontologies* and *user ontologies* in the OWL-BSU service description contain a set of functions that the service is to perform. These information will be registered with the ontology category. For example, If a user wants to register their service as a credit card service, it needs to choose the payment ontology and select the credit card category. The service registration process is shown in Figure 5.4. In the whole registration processing, the user also needs to register User ontologies, Resource ontologies and Relationship ontologies based on the BSU framework. After that, the SWS search engine extracts the semantic annotation and store it into the ontology database.

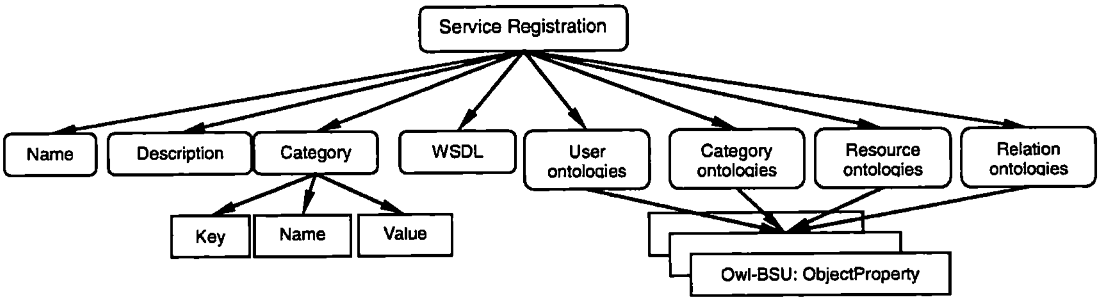


Figure 5.3: Service Registration Structure

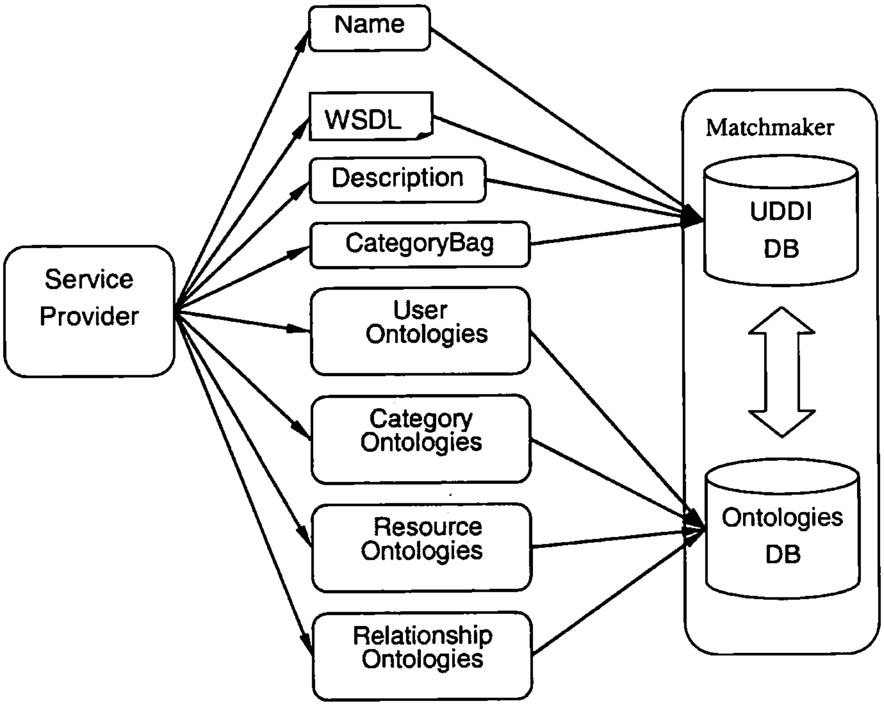


Figure 5.4: Service Registration Process

5.1.3 Discovery

The discovery process of the SWS search engine is shown in Figure 5.5. During the service discovery, a client sends the request to the SWS search engine. The search request should consist of the normal UDDI search fields such as human-readable inforamtion such as service name and CategoryBag, and the semantic search items such as user ontologies, categories ontologies and so on. Also, users need to choose which semantic relationships they want to use. When receiving a request, the semantic matching algorithm of SWS search engine for the ontologies matching engine will use the data from the ontologies database to start the matching process. The matching engine will search all the possible services that are semantically similar to the service requested in the SWS search engine and UDDI database. Having obtained the matching results, the SWS search engine sends back the result lists to the client.

5.2 Implementation of the SWS Search Engine

Due to the time limitations of one-year master programme, we cannot implement all the functions and ontologies of BSU framework in the prototype of the SWS search

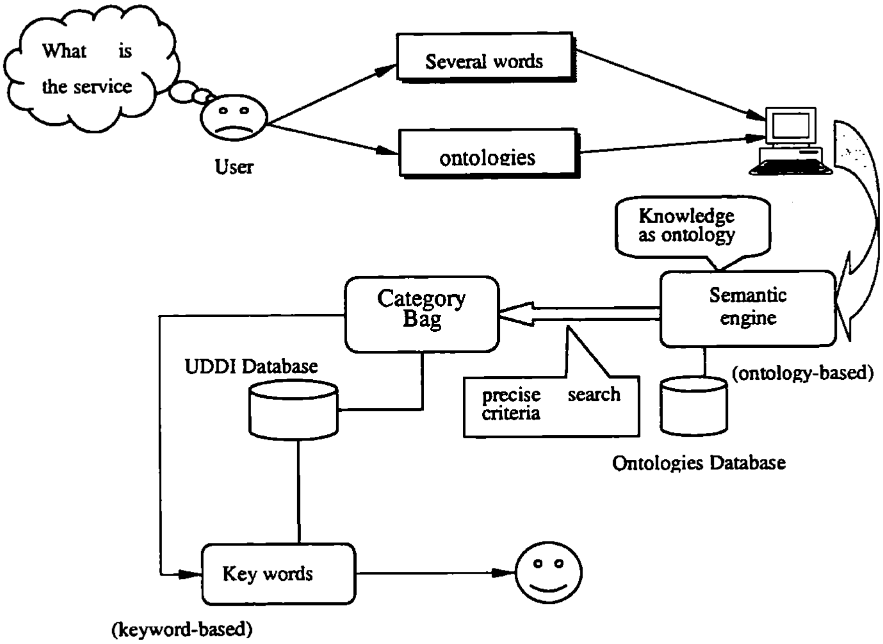
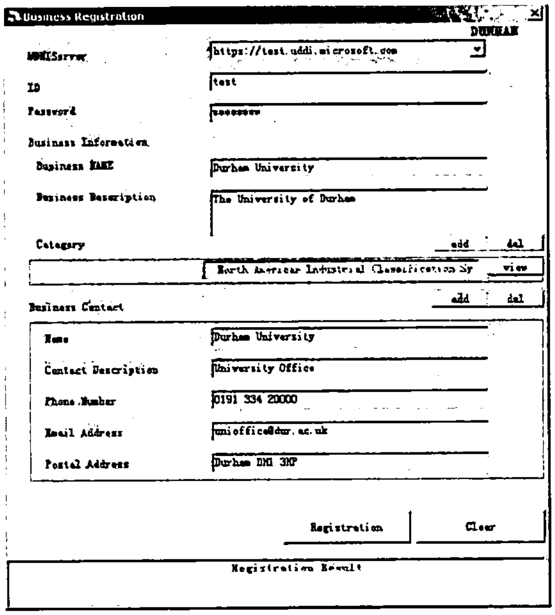


Figure 5.5: The process of service discovery

engine. In fact, we only build three ontologies in the SWS search engine in terms of payment ontology, student ontology and electronic journal ontology. Moreover, we simulate the registry and discovery functionality of UDDI registry in the SWS search engine. The SWS search engine is implemented by Visual Basic 6.0 and MySQL. In the followings, we will represent some snapshots of the SWS search engine user interface, which consists of *Business Registration, Service Registration* and *Service Discovery*.

5.2.1 Business Registration

The client interface of the SWS search engine for the creation of the Business is shown in Figure 5.6. To create a business, users should import *Business Name, Business Description, Category*, and *Business Contact information*. Business name stores the name of the company. CategoryBag stores data categorised on the industry type and product type. After having entered all the business information, push the "Registration" button in order to execute the registration. In the first version of SWS search engine, we do not implement BSU ontologies. The only ontology description implemented so far is in the service registration.



UDDIServer	https://test.uddi.microsoft.com
ID	test
Password	********
Business Information	
Business NAME	Durham University
Business Description	The University of Durham
Category	North American Industrial Classification Sy
Business Contact	
Name	Durham University
Contact Description	University Office
Phone Number	0191 334 20000
Email Address	unioffice@dur.ac.uk
Postal Address	Durham DH1 3HP

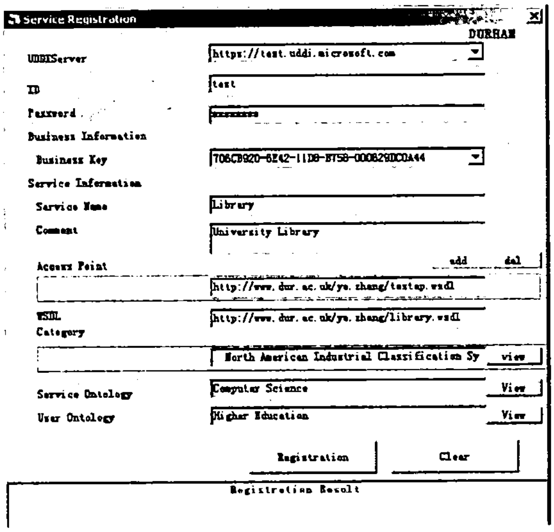
Figure 5.6: Business Registration

5.2.2 Service Registration

After creating a Business, the service provider should register services that belongs to the Business. The service registration interface of the SWS search engine is shown in Figure 5.7. To create a service, the service provider should import the WSDL file in the SWS search engine, and import the ontology annotation using the Ontology Viewer (shown in Figure 5.8), which parses ontology files written by OWL-BSU, then show them as a graphical tree. The service provider should specify any ontology class which is annotated to each parameter by click a node in the window. Once the ontology information of service has been choosen, and then the service provider should press on the "Registration" button to finish the service registration process.

5.2.3 Service Discovery

Figure 5.9 specifies the search interface of SWS search engine. When service requester are looking for services. First, service requester should import the service name and select a CategoryBag. Second, the service requester may select the ontology information through the ontology viewer in Figure 5.8. If the service provider does not select the ontology information, the SWS search engine will retrieve the services using simple keyword match only, which makes the search engine exactly same with



UDDIServer	https://test.uddi.microsoft.com
ID	test
Password	********
Business Information	
Business Key	706CB920-6E42-11D8-B758-000629DC0A44
Service Information	
Service Name	Library
Comment	University Library
Access Point	http://www.dur.ac.uk/ye.zhang/testap.wsdl
WSDL	http://www.dur.ac.uk/ye.zhang/library.wsdl
Category	North American Industrial Classification Sy
Service Ontology	Computer Science
User Ontology	Higher Education

Figure 5.7: Business Service Registration

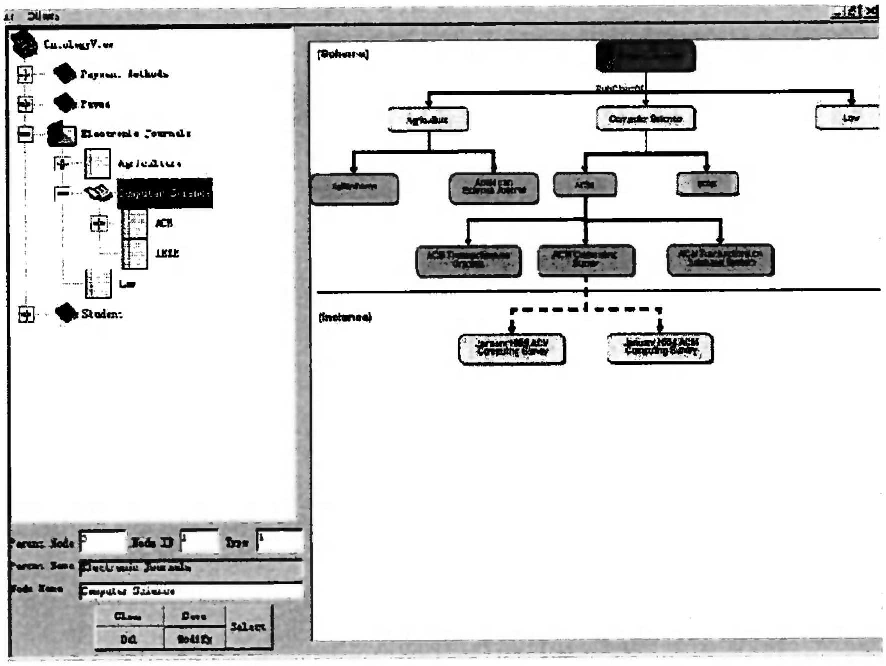
the UDDI registry.

5.3 Evaluations

After the implementation of the SWS search engine, an evaluation of the system is usually carried out. The type of evaluation to be considered heavily depends on the objectives of the system. In the thesis, the objective of the experiments is to evaluate the retrieval performance of our approaches including the BSU framework, semantic matching algorithm and the SWS search engine. To evaluate the system, we compare the retrieval performance of the UDDI registry with the SWS search engine. In the followings, we will describe the dateset, evaluation methodologies and the experimental results.

5.3.1 Dataset

Since, there has been no standard data for evaluating the retrieval performance of semantic description and matching of Web services, we simulate thirty Web services manually. These Web services are registered into the SWS search engine and the UDDI registry. First, the thirty Web services are described by using three BSU ontologies in the SWS search engine such as payment ontology, student ontology and electronic



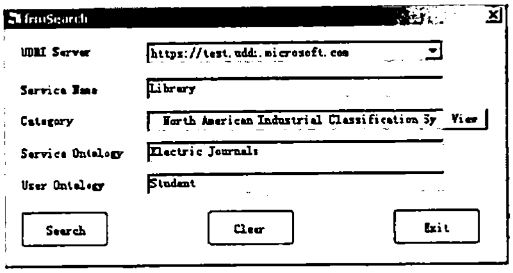
Parent Node 3 Node ID 1 Type 1
Parent Name Electronic Journals
Node Name Computer Science

Figure 5.8: Ontology Viewer

journal ontology. Ten Web services are belonging to one ontology respectively. In another word, ten Web services register with payment ontology, ten Web services register with student ontology and ten Web services register with electronic journal ontology. Second, the same thirty Web services are registered without semantic information into the UDDI registry that is simulated by the SWS search engine. Moreover, the SWS search engine and UDDI registry need some queries terms for evaluation experiments. The queries used in the experiments are the keywords which are extracted from the description information of thirty Web services so that the results will be guaranteed and under the control. Totally, we generated ten query terms for the experiments such as "master", "ACM Transactions", "ieee", "agbioform", "master card", "visa", "phone bill", "student" and "msc".

5.3.2 Evaluation Methodology

The most common measures of retrieval system are precision of the results returned by different systems [18]. In fact, since the service requester's query request is inherently vague, in the most cases, the retrieved Web services are not exact answers [18]. Therefore, both the UDDI registry and the SWS search engine require the



UDDI Server: https://test.uddi.microsoft.com

Service Name: Library

Category: North American Industrial Classification Sy

Service Ontology: Electric Journals

User Ontology: Student

Figure 5.9: Business Service search

evaluation of how precise the answer set is. Retrieval performance evaluations have been extensively studied in the Information Retrieval communities. We use two common retrieval metrics to evaluate the precision of both approaches, which are *Recall*, and *Precision* [18].

- **Recall** is the fraction of the relevant Web services which has been retrieved

$$Recall = \frac{|r|}{|A|}$$

- **Precision** is the relevant fraction of the retrieved Web services

$$Precision = \frac{|r|}{|R|}$$

R is the set of Web services stored in the SWS search engine or the UDDI registry, in the experiments, we set this a fix value 30 due to the total number of Web services is 30.

A is the set of relevant Web services, in the experiments, we define that relevant services include the service *exact match*, *full match* and *part match* the users requirements.

r the intersection of R and A

$|*|$ size of the set

5.3.3 Experimental Results

A set of experiments were conducted to study the retrieval performance of the SWS search engine and the UDDI registry. The first experiment is an initial investigation of recall using our BSU framework and semantic matching algorithm comparing with the UDDI registry. It will show how well the relevant results are returned by different approaches. The second experiment studied the effect of precision by using our approaches and the UDDI registry.

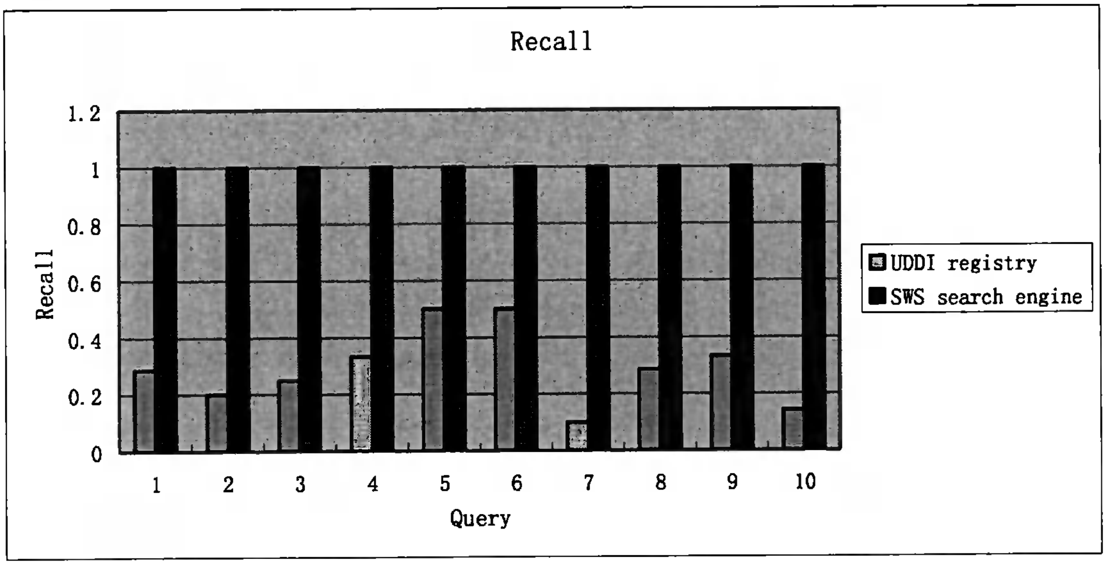


Figure 5.10: Recall

Figure 5.10 shows the experimental results of recall of the SWS search engine and the UDDI registry by using 10 query terms. From the figure, we can see the best case is in Query 7, the recall returned from the SWS search engine is 10 times greater than the recall obtained from the UDDI registry. Query 7 is "student". In the simulated experimental data, 1 service is registered as "student" and 9 services are registered as "undergraduate", "master", "PhD", and other sub-classes of "student". These services can fulfil the part of request, which can be retrieved as part match by the SWS search engine but cannot be retrieved by UDDI. Also, we can see the worst case is that recall returned from the SWS search engine is 2 times greater than the recall obtained from the UDDI registry. For example, Query 5 is "visa". In the simulated experimental data, there are 2 services are registered as "visa", and 2 services are registered as "Credit card", which can be retrieved as full matching by the SWS search engine but cannot be retrieved by UDDI. The average of recall of the SWS search engine is 3.3 times greater than the UDDI returned.

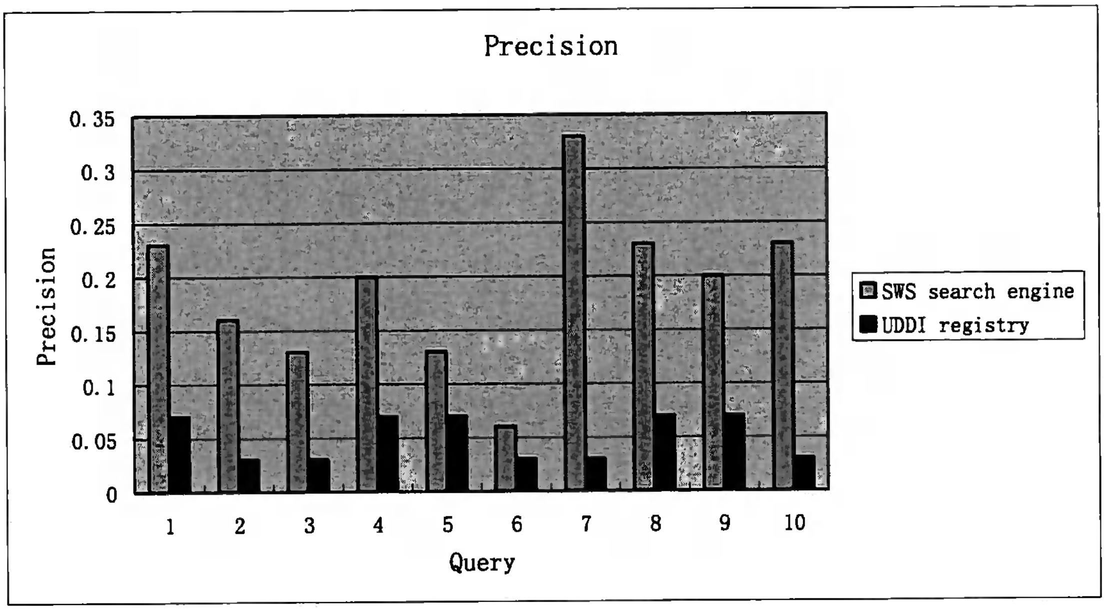


Figure 5.11: Precision

Figure 5.11 shows the experimental results of precision of the SWS search engine and the UDDI registry by using 10 query terms. As we can see, the precision of the UDDI registry is much worse than the results achieved by the SWS search engine. The average precision has been improved 3.8 times by using the SWS search engine. The best performance comes from Query 7 and the worst one is from Query 5, the reasons are same to the previous experiments. However, there are some limitations in our experiments: first, the scale of our experiments is very small, in the experiments we only randomly built 30 Web services and 10 query terms; the small-scale experiments will influence the accuracy of the results and may prefer the SWS search engine. Second, we do not take the system performance under consideration such as response time. Therefore, according to current experimental results, our conclusion is that our approaches such as the BSU framework, semantic matching algorithm and SWS search engine can significantly improve the retrieval performance of Web service discovery on two aspects in terms of recall and precision.

5.4 Summary

In this chapter, we integrated our approaches which include the OWL-BSU and semantic matching algorithm together to implement a Semantic Web services search engine, and evaluate the SWS search engine with recall and precision. We describe the architecture of the SWS search engine, business and service registration, service discovery, implementation of the SWS search engine in details. Moreover, we present the evaluation methodologies, dataset, metrics and the experimental results. The conclusion is that our approaches such as the BSU framework, semantic matching algorithm and SWS search engine can significantly improve the retrieval performance of Web service discovery on two aspects in terms of recall and precision.

Chapter 6

Conclusions and Further Work

In this chapter, we will summarise the research work in this thesis, go through the content of the thesis and discuss our major contributions and significance. Moreover, the further research directions will be described.

6.1 Summary of Thesis

The work in this thesis provides the new semantic description framework for business and service representation and matchmaking in Web service environments. The new description framework for business and service is called the business-service-user (BSU) framework. We deeply analyse the business and service conceptions and then gave the formal definitions of business, service and user in our framework. Also, we identify the requirements and aspects that a Web service description framework should integrate. The BSU framework is designed and developed for a metadata management technique, which focuses particularly on the business and services sharing in Web service environments. Differing with all existing service frameworks for Web services, the BSU framework takes into account five new modules including *business module, user module, business-service relationship module, user-service relationship module* and *user-business relationship module.* After defining the abstract components of BSU framework, we present our effort to create ontologies based on the BSU framework, which is using Semantic Web technologies, focusing particularly on the problems and needs of business and service discovery in Web service environments. Designing ontology typically consists of three steps. The first step is to group together related resources in order to create an ontology. The second step is to define various meaningful properties for each class of the ontology. The third step is to write the class and properties into the semantic markup languages such as OWL and the new work is named OWL-BSU, which is a machine-understandable language.

With the above work, we present our work which will integrate these approaches with existing UDDI registry so that UDDI can represent a Web service with semantic description information. The reason of integration of OWL-BSU and UDDI is that we take advantage of the UDDI popularity and OWL-BSU semantic support into a single, uniform architecture. We introduce how to integrate OWL-BSU with the current UDDI registry. However, the current UDDI query mechanism supports simple keyword exact match only, it does not work properly with semantic information (e.g., ontology) which is represented by OWL-BSU. Therefore, a new query mechanism, specifically, a new matchmaking algorithm that supports semantic information match is necessary. To address this problem, we design a simple and effective semantic matching algorithm. The new matchmaking algorithm defines three semantic relationships for matching which are *exact match, full match*, and *part match*.

Then, we integrated our approaches which include the OWL-BSU and semantic matching algorithm together to implement a Semantic Web service search engine. The Semantic Web service search engine is a prototype of search engine for business and service discovery with semantic based OWL-BSU description in Web service environments. We discussed the architecture of the SWS search engine, business and service registration, service discovery, implementation of the SWS search engine. We evaluate the retrieval performance of SWS search engine. The evaluation methodologies, dataset, metrics and experimental results are discussed in Chapter 5. According to the experimental results, we found that our approaches can improve the retrieval performance of We service discovery on two aspects in terms of recall and precision.

6.2 Contributions

There are four contributions in the thesis. The first and major contribution of the thesis is the BSU framework, which are more sophisticated than any existing description frameworks in Web services. Current standards such as WSDL and UDDI are designed to provide descriptions of message transport, interface, physical attributes in terms of name, address, business description and service description. Both WSDL and

UDDI cannot represent service description information in unambiguous and computer-understandable forms. OWL-S and DAML-S provide much better solutions for describing Web services, they have special service description framework and represented by prevailing Semantic Web language such as OWL and DAML. Unfortunately, a number of essential properties representing Web services are out of the frameworks of OWL-S and DAML-S such as the business class and user class, both of them have significant influence on the service discovery quality. Compared with OWL-S and DAML-S, BSU framework and OWL-BSU take into account five new modules including business module, user module, business-service relationship module, user-service relationship module and user-business relationship module.

The second and major contribution of the thesis is OWL-BSU. We represent these new models by the Semantic Web language to make them computer-understandable, which is OWL-BSU. The description information of OWL-BSU is far more beyond to the current OWL-S and DAML-S. The full version of OWL-BSU is shown in the Appendix.

The third contribution of this thesis is that we proposed a simple and effective semantic matching algorithm. Currently, there are a few semantic matching algorithms, such as literature [50, 39, 41] that can support semantic matching with ontologies. These algorithms can be used in our OWL-BSU ontology. However, according to the literature [23], the performance of current semantic matching algorithms is poor because of the complexity of algorithms. To address the problem, we propose a simple and effective semantic matching algorithm to calculate the semantic relationships between ontologies [51].

The fourth contribution of the thesis is an engineering contribution which is the Semantic Web Service Search Engine. The SWS search engine is implemented by the OWL-BSU and semantic matching algorithm, which is evaluated by recall and precision.

6.3 Further Research

In this thesis, we addressed the problems of semantic-based representing and discovery Web services based on the capabilities that services provide. Our work still have some limitations. First, the experiments scale are small, do not take consideration of query response time. Therefore, other respects of evaluation experiments will be considered in the further. Second, there are still a lot of Business and Services characteristics out of our description framework and ontologies. Thisis an enabling solutionthat opens the door tothe semantic description that will ultimately provide the semantic functionality to the user.In the future, we will find out more Business and Service characteristics and integrate them into our description framework. Third, the semantic matching algorithm needs to be evaluated and compared with current semantic matching algorithms. Fourth, because of the time limitation, some aspects in the BSU framework are highly abstract and need to be detailed in the future.

References

[1] DAML-S 0.9. In *http://www.daml.org/services/daml-s/0.9/ (accessed in March 2007).*

[2] Object Management Group. In *http://www.omg.org/, (accessed in March, 2007).*

[3] OWL Features. In *http://www.w3.org/TR/owl-features/ (accessed in March,2007).*

[4] OWL logic. In *http://www.w3.org/TR/owl-guide/, (accessed in March, 2007).*

[5] OWL-S 1.0. In *http://www.daml.org/services/owl-s/1.0/ (accessed in March, 2007).*

[6] OWL-S 1.1b. In *http://www.daml.org/services/owl-s/1.1B/owl-s/owl-s.html, (accessed in March, 2007).*

[7] OWL Semantics. In *http://www.w3.org/TR/owl-semantics/ (accessed in March, 2007).*

[8] RDF syntax grammer. In *http://www.w3.org/TR/rdf-syntax-grammar/ (accessed in March 2007).*

[9] Resource Descripiton Framework (RDF). In *http://www.w3.org/RDF/ (accessed in March, 2007).*

[10] The WS-Resource Framework. In *http://www.globus.org/wsrf/ (accessed in March, 2007).*

[11] UDDI. In *http://www.uddi.org/ (accessed in March, 2007).*

[12] W3C. In *http://www.w3.org/ (accessed in March, 2007).*

[13] Web Services. In *http://www.w3.org/2002/ws/, (accessed in March, 2007).*

[14] WSDL. In *http://www.w3.org/TR/wsdl (accessed in March, 2007).*

[15] OMG business objects architecture interoperability specification. *Technical report, OMG Business Objects Domain Task Force*, 1997.

[16] L. Allen and N. Lebrer, DARPA/Rome laboratory planning and scheduling initiative knowledge representation specification language Version 2.0.1, *Language specification and manual*, ISX Corporation, 1992.

[17] F. Baader, D. Calvanese, D. McGuinness, D. Nardi and P. Patel-Schneider, *The description logic handbook: theory, implementation, and applications*, Cambridge University Press, 2003.

[18] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, Addison-Wesley Longman Publishing Co., 1999.

[19] K. Candan, H. Liu, and R. Suvarna, Resource Description Framework: metadata and its applications, *SIGKDD Explorations Newsletter*, Volume 3(1), pages 6–19, 2001.

[20] F. Curbera, M. Duftler, R. Khalaf, W. Nagy, N. Mukhi and S. Weerawarana, Unraveling the web services web: An introduction to SOAP, WSDL, and UDDI, *IEEE in Internet Computing*, Volume 6(2), pages 86–93, 2002.

[21] M. Cutkosky, R. Engelmore, R. Fikes, M. Genesereth, T. Gruber, W. Mark, J. Tenenbaum and J. Weber. Pact: An experiment in integrating concurrent engineering systems, *IEEE Computer*, Volume 26(1), pages 28–37, 1993.

[22] J. Dai and R. Steele, UDDI access control, In Proceedings of *the International Conference on Information Technology and Applications*, pages 778–783, 2005.

[23] S. Das, E. Chong, G. Eadon and J. Srinivasan, Supporting ontology-based semantic matching in RDBMS, In Proceedings of *the Thirtieth international conference of VLDB*, pages 1054–1065, 2004.

[24] P. Fankhauser, Beyond Web services - conceptual modelling for service oriented architectures, In *Conceptual Modeling*, Volume 3288, pages 865–866, 2004.

[25] R. Fikes, M. Cutekosky, T. Gruber and J. Baalen. Knowledge sharing technology project overview. *Technical Report KSL 91-71*, Knowledge Systems Laboratory, Standford University, 1991.

[26] K. Gottschalk, S. Graham, H. Kreger and J. Snell. Introduction to Web services architecture. In *IBM Systems Journal*, Volume 41, pages 170–177, 2002.

[27] A. Gromann, A. Henschel and M. Thielscher, A robot control system integrating reactive control, reasoning, and execution monitoring, In Proceedings of the *First International Workshop on Knowledge Representation and Approximate Reasoning*, 2003.

[28] R. Gruber, Ontolingua: A mechanism to support portable ontologies, *Technical Report KSL 91-66*, Knowledge Systems Laboratory, Standford University, 1992.

[29] N. Guarino, Formal ontology and information systems, In Proceedings of the *First International Conference on Formal Ontologies in Information Systems*, pages 3–15, 1998.

[30] N. Guarino, R. Poli and T. Gruber, Toward principles for the design of ontologies used for knowledge sharing, In Proceedings of the *International Workshop on Formal Ontology in Conceptual Analysis and Knowledge Representation*, 1993.

[31] A. Harth, S. Decker, Y. He, H. Tangmunarunkit and C. Kesselman, A Semantic matchmaker service on the Grid, In Proceedings of the *WWW (Alternate Track Papers & Posters)*, pages 326–327, 2004.

[32] P. Herzum and O. Sims, The business component approach, In Proceedings of the *the OOPSLA Workshop on Business Object Design and Implementation*, 1998.

[33] T. Hey and A. Trefethen, The UK e-science core programme and the Grid, In *Future Generation Computing Systems*, Volume 18, pages 1017–1031, 2002.

[34] I. Horrocks. DAML+OIL: A reason-able Web ontology language, In *Web Services, E-Business, and the Semantic Web*, Volume 2512, pages 174, Springer, 2002.

[35] I. Horrocks, OWL: A description logic based ontology language, In Principles and Practice of Constraint Programming, Volume 3709, pages 5-8, Springer, 2005.

[36] R. Howard, ASP+ Web services. In Proceedings of the *Wrox Professional Web Developer Conference*, 2000.

[37] M. Humphrey, G. Wasson, M. Morgan and N. Beekwilder, An early evaluation of WSRF and WS-notification via WSRF.net, In Proceedings of the *Fifth IEEE/ACM International Workshop on Grid Computing*, pages 172–181, 2004.

[38] T. Kawamura, J. Blasio, T. Hasegawa, M. Paolucci and K. Sycara, A preliminary report of a public experiment of a semantic service matchmaker combined with a UDDI business registry, In Proceedings of the *First International Conference on Service Oriented Computing*, 2003.

[39] J. Lee and K. Lee, XML schema matching based on incremental ontology update, In Proceedings of the *International conference on web information systems engineering*, pages 608–618, 2004.

[40] L. Li and I. Horrocks, A software framework for matchmaking based on Semantic Web technology, *International Journal of Electronic Commerce*, pages 331 - 339, 2003

[41] C. Lin, Z. Wu, S. Deng and L. Kuang, Automatic service matching and service discovery based on ontology, In *Grid and Cooperative Computing*, Volume 3252, pages 99–106, 2004.

[42] T. Männistö, H. Peltonen, T. Soininen, and R. Sulonen, Multiple abstraction levels in modelling product structures, *Data and Knowledge Engineering*, Volume 36(1), pages 55–78, 2001.

[43] D. Martin, M. Paolucci, S. McIlraith, M. Burstein, D. McDermott, D. McGuinness, B. Parsia, T. Payne, M. Sabou, M. Solanki, N. Srinivasan and K. Sycara, Bringing semantics to Web services: The OWL-S approach. In Proceedings of the *First International Workshop on Semantic Web Services and Web Process Composition*, pages 26–42, 2004.

[44] W. Robinson, Monitoring Web service requirements, In Proceedings of the *11th IEEE International Requirements Engineering Conference*, pages 65–74, 2003.

[45] R. Shelton, Enterprise Reuse, *Distributed Computing Monitor*, Volume 10(3), pages 3–21, 1998.

[46] G. Sivaraman, Composite capability/preference profile - a RDF based profiler for mobile devices, In *http://www.tml.tkk.fi/Opinnot/Tik-111.590/2000/Papers/Rdf.html*.

[47] N. Srinivasan, M. Paolucci and K. Sycara. Adding OWL-S to UDDI, implementation and throughput,. In Proceedings of the *First International Workshop on Semantic Web Services and Web Process Composition*, pages 6–9, 2006.

[48] K. Sycara, M. Paolucci, A. Ankolekar and N. Srinivasan, Automated discovery, interaction and composition of semantic web services, *Journal of Web Semantic.*, Volume 1(1), pages 27–46, 2003.

[49] K. Sycara, M. Paolucci, M. van Velsen and J. Giampapa, The RETSINA MAS infrastructure, *Technical Report CMU-RI-TR-01-05*, Robotics Institute, Carnegie Mellon University, 2001.

[50] K. Sycara, S. Wido, M. Klusch and J. Lu, LARKS: Dynamic matchmaking among heterogeneous software agents in cyberspace, *Autonomous agents and multi-agent systems*, 2002.

[51] Y. Zhang and W. Song, Semantic description and matchmaking for Grid services, In *the Proceedings of the UK e-Science All Hands Meeting*, pages 205–208, 2004.

Appendix A

Upper Ontology for Business, Service and User

Upper Ontology for Business—Business.owl

Source: http://www.dur.ac.uk/ye.zhang/ owl-bsu/business/Business.owl

```
  <?xml version="1.0" ?>
      - <!--

  -->
- <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Business.owl#">
- <owl:Ontology rdf:about="">
   <owl:versionInfo>$Id: Business.owl,v 0.5 2004/06/16 01:56:34 martin Exp $</owl:versionInfo>
   <rdfs:comment>Top level of OWL ontology for services. Part of the OWL-BSU ontology; see
    http://www.dur.ac.uk/ye.zhang/owl-bsu/business/.</rdfs:comment>
   </owl:Ontology>
         - <!--

  -->
 - <!--

  -->
- <owl:Class rdf:ID="Business">
   <rdfs:label>Business</rdfs:label>
   <rdfs:comment>See comments above</rdfs:comment>
   </owl:Class>
 - <!--

  -->
- <owl:Class rdf:ID="Content">
   <rdfs:label>BusinessContent</rdfs:label>
   <rdfs:comment>See comments above</rdfs:comment>
   </owl:Class>
 - <!--

  -->
- <owl:Class rdf:ID="User">
   <rdfs:label>BusinessUser</rdfs:label>
   <rdfs:comment>See comments above</rdfs:comment>
   </owl:Class>
 - <!--
```

```
      -->
- <owl:Class rdf:ID="Relationship">
   <rdfs:label>BusinessRelationship</rdfs:label>
   <rdfs:comment>See comments above</rdfs:comment>
      </owl:Class>
   - <!--

      -->
- <owl:ObjectProperty rdf:ID="presents">
   <rdfs:comment>There are no cardinality restrictions on this property.</rdfs:comment>
   <rdfs:domain
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Business.owl#Business"
      />
   <rdfs:range
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Businessl#BusinessConte
      nt" />
   <owl:inverseOf
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Business#presentedBy"
      />
      </owl:ObjectProperty>
- <owl:ObjectProperty rdf:ID="presentedBy">
   <rdfs:comment>There are no cardinality restrictions on this property.</rdfs:comment>
   <rdfs:domain
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Business.owl#BusinessC
      ontent" />
   <rdfs:range
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Business.owl#Business"
      />
   <owl:inverseOf
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Business.owl#presents"
      />
      </owl:ObjectProperty>
- <owl:ObjectProperty rdf:ID="isPresentedBy">
   <rdfs:comment>deprecated form</rdfs:comment>
   <owl:equivalentProperty rdf:resource="#presentedBy" />
      </owl:ObjectProperty>
   - <!--

      -->
- <owl:ObjectProperty rdf:ID="provides">
   <rdfs:comment>There are no cardinality restrictions on this property.</rdfs:comment>
   <rdfs:domain
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Business#Business" />
   <rdfs:range
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Business#BusinessUser"
      />
   <owl:inverseOf
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Business#providedBy" />
      </owl:ObjectProperty>
- <owl:ObjectProperty rdf:ID="providedBy">
   <rdfs:comment>There are no cardinality restrictions on this property.</rdfs:comment>
   <rdfs:domain
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Business#BusinessUser"
      />
   <rdfs:range
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Business#Business" />
   <owl:inverseOf
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Business#provides" />
      </owl:ObjectProperty>
- <owl:ObjectProperty rdf:ID="isProvidedBy">
   <rdfs:comment>deprecated form</rdfs:comment>
   <owl:equivalentProperty rdf:resource="#providedBy" />
      </owl:ObjectProperty>
   - <!--

      -->
- <owl:ObjectProperty rdf:ID="supporttedBy">
   <rdfs:comment>There are no cardinality restrictions on this property.</rdfs:comment>
```

```
  <rdfs:domain
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Business#Business" />
  <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Business#BusinessResou
     rce" />
  <owl:inverseOf
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Business#supports" />
    </owl:ObjectProperty>
- <owl:ObjectProperty rdf:ID="supports">
  <rdfs:comment>There are no cardinality restrictions on this property.</rdfs:comment>
  <rdfs:domain
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Business#BusinessResou
     rce" />
  <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Business#Business" />
  <owl:inverseOf
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Business#presents" />
    </owl:ObjectProperty>
- <owl:ObjectProperty rdf:ID="isSupporttedBy">
  <rdfs:comment>deprecated form</rdfs:comment>
  <owl:equivalentProperty rdf:resource="#supporttedBy" />
    </owl:ObjectProperty>
   - <!--

    -->
- <owl:ObjectProperty rdf:ID="has">
  <rdfs:comment>There are no cardinality restrictions on this property.</rdfs:comment>
  <rdfs:domain
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Business#Business" />
  <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Business#BusinessRelati
     onship" />
  <owl:inverseOf
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Business#belongsTo" />
    </owl:ObjectProperty>
- <owl:ObjectProperty rdf:ID="belongsTo">
  <rdfs:comment>There are no cardinality restrictions on this property.</rdfs:comment>
  <rdfs:domain
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Business#BusinessRelati
     onship" />
  <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Business#Business" />
  <owl:inverseOf rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Business#has"
     />
    </owl:ObjectProperty>
- <owl:ObjectProperty rdf:ID="isBelongsTo">
  <rdfs:comment>deprecated form</rdfs:comment>
  <owl:equivalentProperty rdf:resource="#belongsTo" />
    </owl:ObjectProperty>
    </rdf:RDF>
```

Upper Ontology for Business—Content.owl

Source: http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Content.owl

```
 <?xml version="1.0" ?>
     - <!--

                    -->
- <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:business="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Business.owl#"
     xmlns:user="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/User.owl#"
```

```
    xmlns:relationship="http://www.dur.ac.uk/ye.zhang/ owl-bsu/business/Relationship.owl#"
    xmlns="http://www.dur.ac.uk/ye.zhang/owl-bsu/ service/Content.owl#">
- <owl:Ontology rdf:about="">
  <owl:versionInfo>$Id: Content.owl,v 0.5 2004/06/15 13:32:14 $</owl:versionInfo>
  <rdfs:comment>OWL ontology for Advertisements (i.e. Contents). This file belongs to the
    OWL-BSU 0.5 beta Release. Initial version created by Ye Zhang (ye.zhang@dur.ac.uk) and
    William Song (w.w.song@dur.ac.uk).</rdfs:comment>
  <rdfs:seeAlso rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service" />
  <rdfs:seeAlso rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business" />
          - <!--  -->
  <owl:imports rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Business.owl" />
  <owl:imports
    rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Relationship.owl" />
  <owl:imports rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/User.owl" />
    </owl:Ontology>
  - <!--                                                  -->
  - <!--                                                  -->
  - <!--                                                  -->
  - <!--                                                  -->
  - <!--                                                  -->
        - <!--



            -->
- <owl:Class rdf:ID="Content">
  <rdfs:label>Content</rdfs:label>
  <rdfs:subClassOf
    rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Business.owl#BusinessC
    ontent" />
  <rdfs:comment>Definition of Business Content</rdfs:comment>
    </owl:Class>
  - <!--                                                  -->
  - <!--                                                  -->
  - <!--                                                  -->
  - <!--                                                  -->
  - <!--                                                  -->
        - <!--

            -->
- <owl:DatatypeProperty rdf:ID="businessName">
  <rdfs:domain rdf:resource="#Content" />
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
    </owl:DatatypeProperty>
  - <!--

    -->
- <owl:Class rdf:about="#Content">
  <rdfs:comment>A content can have only one name</rdfs:comment>
- <rdfs:subClassOf>
- <owl:Restriction owl:cardinality="1">
  <owl:onProperty rdf:resource="#businessName" />
    </owl:Restriction>
    </rdfs:subClassOf>
    </owl:Class>
  - <!--                                                  -->
  - <!--                                                  -->
  - <!--                                                  -->
  - <!--                                                  -->
  - <!--                                                  -->
        - <!--



            -->
- <owl:DatatypeProperty rdf:ID="businessDescription">
  <rdfs:domain rdf:resource="#Content" />
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
    </owl:DatatypeProperty>
  - <!--

    -->
- <owl:Class rdf:about="#Content">
  <rdfs:comment>A content can have only one service description</rdfs:comment>
- <rdfs:subClassOf>
- <owl:Restriction owl:cardinality="1">
  <owl:onProperty rdf:resource="#businessDescription" />
```

```
      </owl:Restriction>
      </rdfs:subClassOf>
      </owl:Class>
  - <!--                                                      -->
  - <!--                                                      -->
  - <!--                                                      -->
  - <!--                                                      -->
  - <!--                                                      -->
            - <!--




              -->
- <owl:ObjectProperty rdf:ID="contactInformation">
  <rdfs:domain rdf:resource="#Content" />
      </owl:ObjectProperty>
  - <!--                                                      -->
  - <!--                                                      -->
  - <!--                                                      -->
  - <!--                                                      -->
  - <!--                                                      -->
            - <!--










                -->
- <rdf:Property rdf:ID="businessCategory">
  <rdfs:domain rdf:resource="#Content" />
  <rdfs:range rdf:resource="#BusinessCategory" />
      </rdf:Property>
            - <!--




      -->
- <owl:DatatypeProperty rdf:ID="BcategoryName">
  <rdfs:domain rdf:resource="#BusinessCategory" />
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
      </owl:DatatypeProperty>
  - <!--

      -->
- <owl:Class rdf:about="#BusinessCategory">
  <rdfs:comment>a BusinessCategory is restricted to refer to only onename</rdfs:comment>
- <rdfs:subClassOf>
- <owl:Restriction owl:cardinality="1">
  <owl:onProperty rdf:resource="#BcategoryName" />
      </owl:Restriction>
      </rdfs:subClassOf>
      </owl:Class>
            - <!--




      -->
- <owl:DatatypeProperty rdf:ID="Btaxonomy">
  <rdfs:domain rdf:resource="#BusinessCategory" />
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
      </owl:DatatypeProperty>
            - <!--



      -->
- <owl:Class rdf:about="#BerviceCategory">
```

```
  <rdfs:comment>a BusinessCategory is restricted to refer to only one Btaxonomy</rdfs:comment>
- <rdfs:subClassOf>
- <owl:Restriction owl:cardinality="1">
  <owl:onProperty rdf:resource="#Btaxonomy" />
    </owl:Restriction>
    </rdfs:subClassOf>
    </owl:Class>
  - <!--                                                                  -->
  - <!--                                                                  -->
  - <!--                                                                  -->
  - <!--                                                                  -->
  - <!--                                                                  -->
              - <!--



    -->
- <rdf:Property rdf:ID="Relationship">
  <rdfs:domain rdf:resource="#Relationship" />
  <rdfs:range rdf:resource="#Relationship" />
    </rdf:Property>
- <owl:DatatypeProperty rdf:ID="B2BRelationship">
  <rdfs:domain rdf:resource="#Relationship" />
  <rdfs:range
    rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Relationship.owl#B2BRel
    ationship" />
    </owl:DatatypeProperty>
- <owl:DatatypeProperty rdf:ID="B2SRelationship">
  <rdfs:domain rdf:resource="#Relationship" />
  <rdfs:range
    rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Relationship.owl#B2SRel
    ationship" />
    </owl:DatatypeProperty>
  - <!--

    -->
  <owl:Class rdf:ID="B2BRelationship" />
              - <!--



    -->
- <owl:DatatypeProperty rdf:ID="SomeDistinctFrom">
  <rdfs:domain rdf:resource="#B2BRelationship" />
  <rdfs:range
    rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Relationship.owl#SomeDi
    stinctFrom" />
    </owl:DatatypeProperty>
              - <!--



    -->
- <owl:DatatypeProperty rdf:ID="AllDistinctFrom">
  <rdfs:domain rdf:resource="#B2BRelationship" />
  <rdfs:range
    rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Relationship.owl#AllDisti
    nctFrom" />
    </owl:DatatypeProperty>
              - <!--



    -->
- <owl:DatatypeProperty rdf:ID="Equivalent">
  <rdfs:domain rdf:resource="#B2BRelationship" />
  <rdfs:range
    rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Relationship.owl#Equival
    ent" />
    </owl:DatatypeProperty>
  - <!--

    -->
  <owl:Class rdf:ID="B2SRelationship" />
              - <!--



    -->
- <owl:DatatypeProperty rdf:ID="SomePropertiesTo">
  <rdfs:domain rdf:resource="#B2SRelationship" />
  <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Relationship.owl#SomeP
     ropertiesFrom" />
    </owl:DatatypeProperty>
  - <!--

    -->
- <owl:DatatypeProperty rdf:ID="AllPropertiesTo">
  <rdfs:domain rdf:resource="#B2SRelationship" />
  <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Relationship.owl#AllProp
     ertiesFrom" />
    </owl:DatatypeProperty>
  - <!--                                                              -->
  - <!--                                                              -->
  - <!--                                                              -->
  - <!--                                                              -->
  - <!--                                                              -->
              - <!--




    -->
              - <!--






    -->
- <owl:DatatypeProperty rdf:ID="userName">
  <rdfs:domain rdf:resource="#User" />
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
    </owl:DatatypeProperty>
              - <!--


    -->
- <owl:DatatypeProperty rdf:ID="userDescription">
  <rdfs:domain rdf:resource="#User" />
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
    </owl:DatatypeProperty>
              - <!--




    -->
- <rdf:Property rdf:ID="userCategory">
  <rdfs:domain rdf:resource="#User" />
  <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/User.owl#UserCategory"
     />
    </rdf:Property>
              - <!--





    -->
- <owl:DatatypeProperty rdf:ID="Location">
  <rdfs:domain rdf:resource="#User" />
  <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/User.owl#Location" />
    </owl:DatatypeProperty>
          - <!--



    -->
- <owl:DatatypeProperty rdf:ID="userQuality">
   <rdfs:domain rdf:resource="#User" />
   <rdfs:range
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/User.owl#userQuality" />
    </owl:DatatypeProperty>
    </rdf:RDF>
```

Upper Ontology for Business—User.owl

Source: http://www.dur.ac.uk/ye.zhang/owl-bsu/business/ User.owl

```
    <?xml version="1.0" ?>
         - <!--



          -->
- <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
      xmlns:owl="http://www.w3.org/2002/07/owl#"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
      xmlns:business="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/ Service.owl#"
      xmlns:content="http://www.dur.ac.uk/ye.zhang/owl-bsu/ service/ Content.owl#"
      xmlns:Relationship="http://www.dur.ac.uk/ye.zhang/ owl-bsu/service/Relationship.owl#"
      xmlns="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/User.owl#">
- <owl:Ontology rdf:about="">
   <owl:versionInfo>$Id: User.owl,v 0.5 2004/06/15 13:32:14 $</owl:versionInfo>
   <rdfs:comment>OWL ontology for Advertisements (i.e. Resource). This file belongs to the
      OWL-BSU 0.5 beta Release. Initial version created by Ye Zhang (ye.zhang@dur.ac.uk) and
      William Song (w.w.song@dur.ac.uk).</rdfs:comment>
   <rdfs:seeAlso rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service" />
   <rdfs:seeAlso rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business" />
          - <!--   -->
   <owl:imports rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Business.owl" />
   <owl:imports rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Content.owl" />
   <owl:imports
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Relationship.owl" />
    </owl:Ontology>
  - <!--                                                    -->
  - <!--                                                    -->
  - <!--                                                    -->
  - <!--                                                    -->
  - <!--                                                    -->
          - <!--



    -->
          - <!--



    -->
- <owl:DatatypeProperty rdf:ID="userName">
   <rdfs:domain rdf:resource="#User" />
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
    </owl:DatatypeProperty>
          - <!--
    -->
- <owl:DatatypeProperty rdf:ID="userDescription">
   <rdfs:domain rdf:resource="#User" />
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
   </owl:DatatypeProperty>
            - <!--



    -->
- <rdf:Property rdf:ID="userCategory">
   <rdfs:domain rdf:resource="#User" />
   <rdfs:range
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/User.owl#UserCategory"
      />
   </rdf:Property>
            - <!--



    -->
- <owl:DatatypeProperty rdf:ID="Location">
   <rdfs:domain rdf:resource="#User" />
   <rdfs:range
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/User.owl#Location" />
   </owl:DatatypeProperty>
            - <!--



    -->
- <owl:DatatypeProperty rdf:ID="userQuality">
   <rdfs:domain rdf:resource="#User" />
   <rdfs:range
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/User.owl#userQuality" />
   </owl:DatatypeProperty>
   </rdf:RDF>
```

Upper Ontology for Business—Relationship.owl

Source:
http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Relationship.owl

```
  <?xml version="1.0" ?>
        - <!--



    -->
- <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
      xmlns:owl="http://www.w3.org/2002/07/owl#"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
      xmlns:business="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Business.owl#"
      xmlns:content="http://www.dur.ac.uk/ye.zhang/owl-bsu/ service/ Content.owl#"
      xmlns:user="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/User.owl#"
      xmlns="http:// www.dur.ac.uk/ ye.zhang/owl-bsu/ service/Relationship.owl#">
- <owl:Ontology rdf:about="">
   <owl:versionInfo>$Id: Relationship.owl,v 0.5 2004/06/15 13:32:14 $</owl:versionInfo>
   <rdfs:comment>OWL ontology for Advertisements (i.e. Resource). This file belongs to the
      OWL-BSU 0.5 beta Release. Initial version created by Ye Zhang (ye.zhang@dur.ac.uk)and
      William Song (w.w.song@dur.ac.uk).</rdfs:comment>
   <rdfs:seeAlso rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service" />
   <rdfs:seeAlso rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business" />
```

```
              - <!--


   -->
  <owl:imports rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Business.owl" />
  <owl:imports rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Content.owl" />
  <owl:imports rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/User.owl" />
    </owl:Ontology>
 - <!--                                                     -->
 - <!--                                                     -->
 - <!--                                                     -->
 - <!--                                                     -->
 - <!--                                                     -->
            - <!--







   -->
- <rdf:Property rdf:ID="Relationship">
  <rdfs:domain rdf:resource="#Relationship" />
  <rdfs:range rdf:resource="#Relationship" />
    </rdf:Property>
- <owl:DatatypeProperty rdf:ID="B2BRelationship">
  <rdfs:domain rdf:resource="#Relationship" />
  <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Relationship.owl#B2BRel
     ationship" />
    </owl:DatatypeProperty>
- <owl:DatatypeProperty rdf:ID="B2SRelationship">
  <rdfs:domain rdf:resource="#Relationship" />
  <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Relationship.owl#B2SRel
     ationship" />
    </owl:DatatypeProperty>
 - <!--

   -->
  <owl:Class rdf:ID="B2BRelationship" />
            - <!--




   -->
- <owl:DatatypeProperty rdf:ID="SomeDistinctFrom">
  <rdfs:domain rdf:resource="#B2BRelationship" />
  <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Relationship.owl#SomeDi
     stinctFrom" />
    </owl:DatatypeProperty>
            - <!--




   -->
- <owl:DatatypeProperty rdf:ID="AllDistinctFrom">
  <rdfs:domain rdf:resource="#B2BRelationship" />
  <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Relationship.owl#AllDisti
     nctFrom" />
    </owl:DatatypeProperty>
            - <!--



   -->
- <owl:DatatypeProperty rdf:ID="Equivalent">
  <rdfs:domain rdf:resource="#B2BRelationship" />
  <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Relationship.owl#Equival
     ent" />
    </owl:DatatypeProperty>
  - <!--                                    -->
  <owl:Class rdf:ID="B2SRelationship" />
            - <!--



    -->
- <owl:DatatypeProperty rdf:ID="SomePropertiesTo">
  <rdfs:domain rdf:resource="#B2SRelationship" />
  <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Relationship.owl#SomeP
     ropertiesFrom" />
    </owl:DatatypeProperty>
  - <!--                                                       -->
- <owl:DatatypeProperty rdf:ID="AllPropertiesTo">
  <rdfs:domain rdf:resource="#B2SRelationship" />
  <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business/Relationship.owl#AllProp
     ertiesFrom" />
    </owl:DatatypeProperty>
    </rdf:RDF>
```

Upper Ontology for Service—Service.owl

Source: http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl

```
    <?xml version="1.0" ?>
          - <!--





    -->
- <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl#">
- <owl:Ontology rdf:about="">
  <owl:versionInfo>$Id: Service.owl,v 0.5 2004/06/15 23:56:34 martin Exp $</owl:versionInfo>
  <rdfs:comment>Top level of OWL ontology for services. Part of the OWL-BSU ontology; see
     http://www.dur.ac.uk/ye.zhang/owl-bsu/service/.</rdfs:comment>
    </owl:Ontology>
            - <!--














    -->
  - <!--      -->
- <owl:Class rdf:ID="Service">
  <rdfs:label>Service</rdfs:label>
  <rdfs:comment>See comments above</rdfs:comment>
    </owl:Class>
  - <!--

    -->
- <owl:Class rdf:ID="Content">
  <rdfs:label>ServiceContent</rdfs:label>
  <rdfs:comment>See comments above</rdfs:comment>
    </owl:Class>
  - <!--

    -->
- <owl:Class rdf:ID="Operation">
   <rdfs:label>ServiceOperation</rdfs:label>
   <rdfs:comment>See comments above</rdfs:comment>
    </owl:Class>
  - <!--

    -->
- <owl:Class rdf:ID="Resource">
   <rdfs:label>ServiceResource</rdfs:label>
   <rdfs:comment>See comments above</rdfs:comment>
    </owl:Class>
  - <!--

    -->
- <owl:Class rdf:ID="User">
   <rdfs:label>ServiceUser</rdfs:label>
   <rdfs:comment>See comments above</rdfs:comment>
    </owl:Class>
  - <!--

    -->
- <owl:Class rdf:ID="Relationship">
   <rdfs:label>ServiceRelationship</rdfs:label>
   <rdfs:comment>See comments above</rdfs:comment>
    </owl:Class>
  - <!--

    -->
- <owl:ObjectProperty rdf:ID="presents">
   <rdfs:comment>There are no cardinality restrictions on this property.</rdfs:comment>
   <rdfs:domain
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl#Service" />
   <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl#ServiceConten
     t" />
   <owl:inverseOf
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl#presentedBy"
     />
    </owl:ObjectProperty>
- <owl:ObjectProperty rdf:ID="presentedBy">
   <rdfs:comment>There are no cardinality restrictions on this property.</rdfs:comment>
   <rdfs:domain
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl#ServiceConten
     t" />
   <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl#Service" />
   <owl:inverseOf
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl#presents" />
    </owl:ObjectProperty>
- <owl:ObjectProperty rdf:ID="isPresentedBy">
   <rdfs:comment>deprecated form</rdfs:comment>
   <owl:equivalentProperty rdf:resource="#presentedBy" />
    </owl:ObjectProperty>
  - <!--

    -->
- <owl:ObjectProperty rdf:ID="describedBy">
   <rdfs:domain
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl#Service" />
   <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl#ServiceOperat
     ion" />
   <owl:inverseOf
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl#describes" />
    </owl:ObjectProperty>
- <owl:ObjectProperty rdf:ID="isDescribedBy">
   <rdfs:comment>deprecated form</rdfs:comment>
   <owl:equivalentProperty rdf:resource="#describedBy" />
    </owl:ObjectProperty>
- <owl:Class rdf:about="#Service">
   <rdfs:comment>A service has 0 or 1 operations. (But note that a service with 0 operations does
     not provide automated online access; it exists only for discovery purposes; that is, it exists so
     as to provide a Content.)</rdfs:comment>
- <rdfs:subClassOf>
- <owl:Restriction owl:maxCardinality="1">
  <owl:onProperty rdf:resource="#describedBy" />
    </owl:Restriction>
    </rdfs:subClassOf>
    </owl:Class>
- <owl:ObjectProperty rdf:ID="describes">
  <rdfs:comment>There are no cardinality restrictions on this property. That is, the same service
      operation can be used by many different services.</rdfs:comment>
  <rdfs:domain
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl#ServiceOperat
      ion" />
  <rdfs:range
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl#Service" />
  <owl:inverseOf
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl#describedBy"
      />
    </owl:ObjectProperty>
  - <!--

    -->
- <owl:ObjectProperty rdf:ID="provides">
  <rdfs:comment>There are no cardinality restrictions on this property.</rdfs:comment>
  <rdfs:domain
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl#Service" />
  <rdfs:range
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl#ServiceUser"
      />
  <owl:inverseOf
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl#providedBy"
      />
    </owl:ObjectProperty>
- <owl:ObjectProperty rdf:ID="providedBy">
  <rdfs:comment>There are no cardinality restrictions on this property.</rdfs:comment>
  <rdfs:domain
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl#ServiceUser"
      />
  <rdfs:range
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl#Service" />
  <owl:inverseOf
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl#provides" />
    </owl:ObjectProperty>
- <owl:ObjectProperty rdf:ID="isProvidedBy">
  <rdfs:comment>deprecated form</rdfs:comment>
  <owl:equivalentProperty rdf:resource="#providedBy" />
    </owl:ObjectProperty>
  - <!--

    -->
- <owl:ObjectProperty rdf:ID="supporttedBy">
  <rdfs:comment>There are no cardinality restrictions on this property.</rdfs:comment>
  <rdfs:domain
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl#Service" />
  <rdfs:range
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl#ServiceResour
      ce" />
  <owl:inverseOf
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl#supports" />
    </owl:ObjectProperty>
- <owl:ObjectProperty rdf:ID="supports">
  <rdfs:comment>There are no cardinality restrictions on this property.</rdfs:comment>
  <rdfs:domain
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl#ServiceResour
      ce" />
  <rdfs:range
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl#Service" />
  <owl:inverseOf
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl#presents" />
    </owl:ObjectProperty>
- <owl:ObjectProperty rdf:ID="isSupporttedBy">
  <rdfs:comment>deprecated form</rdfs:comment>
  <owl:equivalentProperty rdf:resource="#supporttedBy" />
    </owl:ObjectProperty>
  - <!--

    -->
- <owl:ObjectProperty rdf:ID="has">
  <rdfs:comment>There are no cardinality restrictions on this property.</rdfs:comment>
  <rdfs:domain
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl#Service" />
  <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl#ServiceRelatio
     nship" />
  <owl:inverseOf
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/ Service.owl#belongsTo" />
    </owl:ObjectProperty>
- <owl:ObjectProperty rdf:ID="belongsTo">
  <rdfs:comment>There are no cardinality restrictions on this property.</rdfs:comment>
  <rdfs:domain
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl#ServiceRelatio
     nship" />
  <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl#Service" />
  <owl:inverseOf
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl#has" />
    </owl:ObjectProperty>
- <owl:ObjectProperty rdf:ID="isBelongsTo">
  <rdfs:comment>deprecated form</rdfs:comment>
  <owl:equivalentProperty rdf:resource="#belongsTo" />
    </owl:ObjectProperty>
    </rdf:RDF>
```

Upper Ontology for Service—Content.owl

Source: http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Content.owl

```
<?xml version="1.0" ?>
      - <!--




  -->
- <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:service="http://www.dur.ac.uk/ye.zhang/owl-bsu/ service/ Service.owl#"
     xmlns:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Resource.owl#"
     xmlns:operation="http://www.dur.ac.uk/ ye.zhang/owl-bsu/service/Operation.owl#"
     xmlns:user="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/User.owl#"
     xmlns:relationship="http://www.dur.ac.uk/ye.zhang/ owl-bsu/ service/Relationship.owl#"
     xmlns="http://www.dur.ac.uk/ye.zhang/owl-bsu/ service/Content.owl#">
- <owl:Ontology rdf:about="">
   <owl:versionInfo>$Id: Content.owl,v 0.5 2004/06/15 13:32:14 $</owl:versionInfo>
   <rdfs:comment>OWL ontology for Advertisements (i.e. Contents). This file belongs to the
     OWL-BSU 0.5 beta Release. Initial version created by Ye Zhang (ye.zhang@dur.ac.uk)and
     William Song (w.w.song@dur.ac.uk).</rdfs:comment>
   <rdfs:seeAlso rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service" />
   <rdfs:seeAlso rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business" />
                 - <!-- -->
   <owl:imports rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/ service/ Service.owl" />
   <owl:imports rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/ service/ Operation.owl" />
   <owl:imports rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Resource.owl" />
   <owl:imports rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/ service/ Relationship.owl"
     />
   <owl:imports rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/ User.owl" />
     </owl:Ontology>
   - <!--                                                -->
   - <!--                                                -->
   - <!--                                                -->
   - <!--                                                -->
   - <!--                                                -->
     - <!--
```

```
    -->
- <owl:Class rdf:ID="Content">
   <rdfs:label>Content</rdfs:label>
   <rdfs:subClassOf
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Service.owl#ServiceConten
     t" />
   <rdfs:comment>Definition of Service Content</rdfs:comment>
     </owl:Class>
  - <!--                                                                -->
  - <!--                                                                -->
  - <!--                                                                -->
  - <!--                                                                -->
  - <!--                                                                -->
              - <!--



                -->
- <owl:DatatypeProperty rdf:ID="serviceName">
   <rdfs:domain rdf:resource="#Content" />
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
     </owl:DatatypeProperty>
  - <!--

     -->
- <owl:Class rdf:about="#Content">
   <rdfs:comment>A content can have only one name</rdfs:comment>
- <rdfs:subClassOf>
- <owl:Restriction owl:cardinality="1">
   <owl:onProperty rdf:resource="#serviceName" />
     </owl:Restriction>
     </rdfs:subClassOf>
     </owl:Class>
  - <!--                                                                -->
  - <!--                                                                -->
  - <!--                                                                -->
  - <!--                                                                -->
  - <!--                                                                -->
              - <!--





     -->
- <owl:DatatypeProperty rdf:ID="serviceDescription">
   <rdfs:domain rdf:resource="#Content" />
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
     </owl:DatatypeProperty>
  - <!--

     -->
- <owl:Class rdf:about="#Content">
   <rdfs:comment>A content can have only one service description</rdfs:comment>
- <rdfs:subClassOf>
- <owl:Restriction owl:cardinality="1">
   <owl:onProperty rdf:resource="#serviceDescription" />
     </owl:Restriction>
     </rdfs:subClassOf>
     </owl:Class>
  - <!--                                                                -->
  - <!--                                                                -->
  - <!--                                                                -->
  - <!--                                                                -->
  - <!--                                                                -->
              - <!--














                -->
```

```
- <rdf:Property rdf:ID="serviceCategory">
  <rdfs:domain rdf:resource="#Content" />
  <rdfs:range rdf:resource="#ServiceCategory" />
    </rdf:Property>
          - <!--



    -->
- <owl:DatatypeProperty rdf:ID="ScategoryName">
  <rdfs:domain rdf:resource="#ServiceCategory" />
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
    </owl:DatatypeProperty>
  - <!--

    -->
- <owl:Class rdf:about="#ServiceCategory">
  <rdfs:comment>a ServiceCategory is restricted to refer to only onename</rdfs:comment>
- <rdfs:subClassOf>
- <owl:Restriction owl:cardinality="1">
  <owl:onProperty rdf:resource="#ScategoryName" />
    </owl:Restriction>
    </rdfs:subClassOf>
    </owl:Class>
          - <!--




    -->
- <owl:DatatypeProperty rdf:ID="Staxonomy">
  <rdfs:domain rdf:resource="#ServiceCategory" />
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
    </owl:DatatypeProperty>
          - <!--





    -->
- <owl:Class rdf:about="#ServiceCategory">
  <rdfs:comment>a ServiceCategory is restricted to refer to only one Staxonomy</rdfs:comment>
- <rdfs:subClassOf>
- <owl:Restriction owl:cardinality="1">
  <owl:onProperty rdf:resource="#Staxonomy" />
    </owl:Restriction>
    </rdfs:subClassOf>
    </owl:Class>
  - <!--                                                    -->
  - <!--                                                    -->
  - <!--                                                    -->
  - <!--                                                    -->
  - <!--                                                    -->
          - <!--






    -->
- <owl:ObjectProperty rdf:ID="serviceFuntionality">
  <rdfs:domain rdf:resource="#Profile" />
  <rdfs:range
    rdf:resource="http://dur.ac.uk/ye.zhang/ owl-bsu/service/ Operation.owl#Funtionality" />
    </owl:ObjectProperty>
- <owl:ObjectProperty rdf:ID="Input">
  <rdfs:subPropertyOf rdf:resource="#serviceFuntionality" />
  <rdfs:range rdf:resource="http://dur.ac.uk/ye.zhang/ owl-bsu/service/ Operation.owl#Input" />
    </owl:ObjectProperty>
- <owl:ObjectProperty rdf:ID="Output">
  <rdfs:subPropertyOf rdf:resource="#serviceFuntionality" />
  <rdfs:range rdf:resource="http://dur.ac.uk/ye.zhang/ owl-bsu/service/ Operation.owl#Output"
    />
    </owl:ObjectProperty>
- <owl:ObjectProperty rdf:ID="Precondition">
  <rdfs:domain rdf:resource="#serviceFuntionality" />
  <rdfs:range
      rdf:resource="http://dur.ac.uk/ye.zhang/ owl-bsu/service/ Operation.owl#Precondition" />
    </owl:ObjectProperty>
- <owl:ObjectProperty rdf:ID="Effect">
  <rdfs:domain rdf:resource="#serviceFuntionality" />
  <rdfs:range rdf:resource="http://dur.ac.uk/ye.zhang/ owl-bsu/service/ Operation.owl#Effect" />
    </owl:ObjectProperty>
- <owl:ObjectProperty rdf:ID="Operation">
  <rdfs:domain rdf:resource="#serviceFuntionality" />
  <rdfs:range
      rdf:resource="http://dur.ac.uk/ye.zhang/ owl-bsu/service/ Operation.owl#operation" />
    </owl:ObjectProperty>
  - <!--                                                                  -->
  - <!--                                                                  -->
  - <!--                                                                  -->
  - <!--                                                                  -->
  - <!--                                                                  -->
            - <!--




    -->
- <rdf:Property rdf:ID="serviceState">
  <rdfs:domain rdf:resource="#Content" />
  <rdfs:range rdf:resource="#ServiceState" />
    </rdf:Property>
- <owl:DatatypeProperty rdf:ID="Init">
  <rdfs:domain rdf:resource="#ServiceState" />
  <rdfs:range
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/ Operation.owl#Init" />
    </owl:DatatypeProperty>
  - <!--

    -->
- <owl:Class rdf:about="#ServiceState">
  <rdfs:comment>a ServiceState is restricted to refer to only Init</rdfs:comment>
- <rdfs:subClassOf>
- <owl:Restriction owl:cardinality="1">
  <owl:onProperty rdf:resource="#Init" />
    </owl:Restriction>
    </rdfs:subClassOf>
    </owl:Class>
- <owl:DatatypeProperty rdf:ID="Deactivated">
  <rdfs:domain rdf:resource="#ServiceState" />
  <rdfs:range
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/ Operation.owl#Deactivate
    d" />
    </owl:DatatypeProperty>
  - <!--

    -->
- <owl:Class rdf:about="#ServiceState">
  <rdfs:comment>a ServiceState is restricted to refer to only Deactivated</rdfs:comment>
- <rdfs:subClassOf>
- <owl:Restriction owl:cardinality="1">
  <owl:onProperty rdf:resource="#Deactivated" />
    </owl:Restriction>
    </rdfs:subClassOf>
    </owl:Class>
- <owl:DatatypeProperty rdf:ID="WaitForGoal">
  <rdfs:domain rdf:resource="#ServiceState" />
  <rdfs:range
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/ Operation.owl#WaitForGoa
    l" />
    </owl:DatatypeProperty>
  - <!--

    -->
- <owl:Class rdf:about="#ServiceState">
  <rdfs:comment>a ServiceState is restricted to refer to only WaitForGoal</rdfs:comment>
- <rdfs:subClassOf>
- <owl:Restriction owl:cardinality="1">
  <owl:onProperty rdf:resource="#WaitForGoal" />
    </owl:Restriction>
    </rdfs:subClassOf>
```

```
     </owl:Class>
- <owl:DatatypeProperty rdf:ID="ReceivePlan">
   <rdfs:domain rdf:resource="#ServiceState" />
   <rdfs:range
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/ Operation.owl#ReceivePla
      n" />
     </owl:DatatypeProperty>
  - <!--

     -->
- <owl:Class rdf:about="#ServiceState">
   <rdfs:comment>a ServiceState is restricted to refer to only ReceivedPlan</rdfs:comment>
- <rdfs:subClassOf>
- <owl:Restriction owl:cardinality="1">
   <owl:onProperty rdf:resource="#ReceivePlan" />
     </owl:Restriction>
     </rdfs:subClassOf>
     </owl:Class>
- <owl:DatatypeProperty rdf:ID="PlanReceived">
   <rdfs:domain rdf:resource="#ServiceState" />
   <rdfs:range
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/ Operation.owl#PlanReceiv
      ed" />
     </owl:DatatypeProperty>
  - <!--

     -->
- <owl:Class rdf:about="#ServiceState">
   <rdfs:comment>a ServiceState is restricted to refer to only PlanReceived</rdfs:comment>
- <rdfs:subClassOf>
- <owl:Restriction owl:cardinality="1">
   <owl:onProperty rdf:resource="#PlanReceived" />
     </owl:Restriction>
     </rdfs:subClassOf>
     </owl:Class>
- <owl:DatatypeProperty rdf:ID="InProgress">
   <rdfs:domain rdf:resource="#ServiceState" />
   <rdfs:range
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/ Operation.owl#InProgress
      " />
     </owl:DatatypeProperty>
  - <!--

     -->
- <owl:Class rdf:about="#ServiceState">
   <rdfs:comment>a ServiceState is restricted to refer to only InProgress</rdfs:comment>
- <rdfs:subClassOf>
- <owl:Restriction owl:cardinality="1">
   <owl:onProperty rdf:resource="#InProgress" />
     </owl:Restriction>
     </rdfs:subClassOf>
     </owl:Class>
  - <!--                                                   -->
  - <!--                                                   -->
  - <!--                                                   -->
  - <!--                                                   -->
  - <!--                                                   -->
           - <!--







     -->
- <rdf:Property rdf:ID="Relationship">
   <rdfs:domain rdf:resource="#Relationship" />
   <rdfs:range rdf:resource="#Relationship" />
     </rdf:Property>
- <owl:DatatypeProperty rdf:ID="S2SRelationship">
   <rdfs:domain rdf:resource="#Relationship" />
   <rdfs:range
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Relationship.owl#S2SRelat
      ionship" />
     </owl:DatatypeProperty>
```

```
- <owl:DatatypeProperty rdf:ID="S2BRelationship">
    <rdfs:domain rdf:resource="#Relationship" />
    <rdfs:range
        rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Relationship.owl#S2BRelat
        ionship" />
      </owl:DatatypeProperty>
   - <!--

      -->
    <owl:Class rdf:ID="S2SRelationship" />
              - <!--



      -->
- <owl:DatatypeProperty rdf:ID="SomeDistinctFrom">
    <rdfs:domain rdf:resource="#S2SRelationship" />
    <rdfs:range
        rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Relationship.owl#SomeDis
        tinctFrom" />
      </owl:DatatypeProperty>
              - <!--


      -->
- <owl:DatatypeProperty rdf:ID="AllDistinctFrom">
    <rdfs:domain rdf:resource="#S2SRelationship" />
    <rdfs:range
        rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Relationship.owl#AllDistin
        ctFrom" />
      </owl:DatatypeProperty>
              - <!--


      -->
- <owl:DatatypeProperty rdf:ID="Equivalent">
    <rdfs:domain rdf:resource="#S2SRelationship" />
    <rdfs:range
        rdf:resource="http:// www.dur.ac.uk/ ye.zhang/owl-bsu/ service/Relationship.owl#Equivale
        nt" />
      </owl:DatatypeProperty>
   - <!--

      -->
    <owl:Class rdf:ID="S2BRelationship" />
              - <!--



      -->
- <owl:DatatypeProperty rdf:ID="SomePropertiesFrom">
    <rdfs:domain rdf:resource="#S2SRelationship" />
    <rdfs:range
        rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Relationship.owl#SomePro
        pertiesFrom" />
      </owl:DatatypeProperty>
   - <!--

      -->
- <owl:DatatypeProperty rdf:ID="AllPropertiesFrom">
    <rdfs:domain rdf:resource="#S2SRelationship" />
    <rdfs:range
        rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Relationship.owl#AllPrope
        rtiesFrom" />
      </owl:DatatypeProperty>
   - <!--                                                                  -->
   - <!--                                                                  -->
   - <!--                                                                  -->
   - <!--                                                                  -->
   - <!--                                                                  -->
              - <!--




      -->
              - <!--
    -->
- <owl:DatatypeProperty rdf:ID="userName">
  <rdfs:domain rdf:resource="#User" />
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
    </owl:DatatypeProperty>
            - <!--


    -->
- <owl:DatatypeProperty rdf:ID="userDescription">
  <rdfs:domain rdf:resource="#User" />
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
    </owl:DatatypeProperty>
            - <!--


    -->
- <rdf:Property rdf:ID="userCategory">
  <rdfs:domain rdf:resource="#User" />
  <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/User.owl#UserCategory" />
    </rdf:Property>
            - <!--


    -->
- <owl:DatatypeProperty rdf:ID="Location">
  <rdfs:domain rdf:resource="#User" />
  <rdfs:range rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/User.owl#Location"
     />
    </owl:DatatypeProperty>
            - <!--


    -->
- <owl:DatatypeProperty rdf:ID="userQuality">
  <rdfs:domain rdf:resource="#User" />
  <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/ service/ User.owl#userQuality" />
    </owl:DatatypeProperty>
  - <!--                                                        -->
  - <!--                                                        -->
  - <!--                                                        -->
  - <!--                                                        -->
  - <!--                                                        -->
            - <!--


    -->
- <rdf:Property rdf:ID="serviceResource">
  <rdfs:domain rdf:resource="#Resource" />
  <rdfs:range rdf:resource="#ServiceResource" />
    </rdf:Property>
            - <!--


    -->
- <owl:DatatypeProperty rdf:ID="ResourceProperties">
  <rdfs:domain rdf:resource="#Resource" />
  <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Resource.owl#ResourcePr
     operties" />
    </owl:DatatypeProperty>
            - <!--
    -->
- <owl:DatatypeProperty rdf:ID="RenewableReferences">
  <rdfs:domain rdf:resource="#Resource" />
  <rdfs:range
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Resource.owl#RenewableR
      eferences" />
    </owl:DatatypeProperty>
            - <!--



    -->
- <owl:DatatypeProperty rdf:ID="ResourceLifetime">
  <rdfs:domain rdf:resource="#Resource" />
  <rdfs:range
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Resource.owl#ResourceLif
      etime" />
    </owl:DatatypeProperty>
  - <!--

    -->
- <owl:DatatypeProperty rdf:ID="ServiceGroup">
  <rdfs:domain rdf:resource="#Resource" />
  <rdfs:range
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Resource.owl#ServiceGrou
      p" />
    </owl:DatatypeProperty>
            - <!--


    -->
- <owl:DatatypeProperty rdf:ID="BaseFaults">
  <rdfs:domain rdf:resource="#Resource" />
  <rdfs:range
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Resource.owl#BaseFaults"
      />
    </owl:DatatypeProperty>
    </rdf:RDF>
```

Upper Ontology for Service—Resource.owl

Source: http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Resource.owl

```
    <?xml version="1.0" ?>
        - <!--



            -->
- <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
      xmlns:owl="http://www.w3.org/2002/07/owl#"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
      xmlns:service="http://www.dur.ac.uk/ye.zhang/owl-bsu/ service/ Service.owl#"
      xmlns:content="http://www.dur.ac.uk/ye.zhang/owl-bsu/ service/ Content.owl#"
      xmlns:operation="http://www.dur.ac.uk/ ye.zhang/owl-bsu/service/Operation.owl#"
      xmlns:user="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/User.owl#"
      xmlns:relationship="http://www.dur.ac.uk/ye.zhang/ owl-bsu/ service/Relationship.owl#"
      xmlns="http://www.dur.ac.uk/ye.zhang/owl-bsu/ service/Resource.owl#">
- <owl:Ontology rdf:about="">
  <owl:versionInfo>$Id: Resource.owl,v 0.5 2004/06/15 13:32:14 $</owl:versionInfo>
  <rdfs:comment>OWL ontology for Advertisements (i.e. Resource). This file belongs to the
      OWL-BSU 0.5 beta Release. Initial version created by Ye Zhang (ye.zhang@dur.ac.uk)and
      William Song (w.w.song@dur.ac.uk).</rdfs:comment>
  <rdfs:seeAlso rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service" />
  <rdfs:seeAlso rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business" />
```

```
              - <!--
   -->
 <owl:imports rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/ service/ Service.owl" />
 <owl:imports rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/ service/ Operation.owl" />
 <owl:imports rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Content.owl" />
 <owl:imports rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/ service/ Relationship.owl"
    />
 <owl:imports rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/ User.owl" />
   </owl:Ontology>
 - <!--                                                   -->
 - <!--                                                   -->
 - <!--                                                   -->
 - <!--                                                   -->
 - <!--                                                   -->
            - <!--


   -->
            - <!--



   -->
- <rdf:Property rdf:ID="serviceResource">
  <rdfs:domain rdf:resource="#Resource" />
  <rdfs:range rdf:resource="#ServiceResource" />
   </rdf:Property>
            - <!--

   -->
- <owl:DatatypeProperty rdf:ID="ResourceProperties">
  <rdfs:domain rdf:resource="#Resource" />
  <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Resource.owl#ResourcePr
     operties" />
   </owl:DatatypeProperty>
            - <!--


   -->
- <owl:DatatypeProperty rdf:ID="RenewableReferences">
  <rdfs:domain rdf:resource="#Resource" />
  <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Resource.owl#RenewableR
     eferences" />
   </owl:DatatypeProperty>
            - <!--



   -->
- <owl:DatatypeProperty rdf:ID="ResourceLifetime">
  <rdfs:domain rdf:resource="#Resource" />
  <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Resource.owl#ResourceLif
     etime" />
   </owl:DatatypeProperty>
 - <!--
   -->
- <owl:DatatypeProperty rdf:ID="ServiceGroup">
  <rdfs:domain rdf:resource="#Resource" />
  <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Resource.owl#ServiceGrou
     p" />
   </owl:DatatypeProperty>
            - <!--

   -->
- <owl:DatatypeProperty rdf:ID="BaseFaults">
  <rdfs:domain rdf:resource="#Resource" />
```

```
<rdfs:range
   rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Resource.owl#BaseFaults"
   />
  </owl:DatatypeProperty>
  </rdf:RDF>
```

Upper Ontology for Service—User.owl

Source: http://www.dur.ac.uk/ye.zhang/owl-bsu/service/User.owl

```
  <?xml version="1.0" ?>
      - <!--



  -->
- <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:service="http://www.dur.ac.uk/ye.zhang/owl-bsu/ service/ Service.owl#"
    xmlns:content="http://www.dur.ac.uk/ye.zhang/owl-bsu/ service/ Content.owl#"
    xmlns:operation="http://www.dur.ac.uk/ ye.zhang/owl-bsu/service/Operation.owl#"
    xmlns:Relationship="http://www.dur.ac.uk/ye.zhang/ owl-bsu/service/Relationship.owl#"
    xmlns:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Resource.owl#"
    xmlns="http:// www.dur.ac.uk/ ye.zhang/owl-bsu/ service/Relationship.owl#">
- <owl:Ontology rdf:about="">
  <owl:versionInfo>$Id: User.owl,v 0.5 2004/06/15 13:32:14 $</owl:versionInfo>
  <rdfs:comment>OWL ontology for Advertisements (i.e. Resource). This file belongs to the
    OWL-BSU 0.5 beta Release. Initial version created by Ye Zhang (ye.zhang@dur.ac.uk)and
    William Song (w.w.song@dur.ac.uk).</rdfs:comment>
  <rdfs:seeAlso rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service" />
  <rdfs:seeAlso rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business" />
          - <!--
  -->
  <owl:imports rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/ service/ Service.owl" />
  <owl:imports rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/ service/ Operation.owl" />
  <owl:imports rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Content.owl" />
  <owl:imports rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Resource.owl" />
  <owl:imports rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/ service/ Relationship.owl"
    />
  </owl:Ontology>
 - <!--                                              -->
 - <!--                                              -->
 - <!--                                              -->
 - <!--                                              -->
 - <!--                                              -->
        - <!--



  -->
        - <!--


              -->
- <owl:DatatypeProperty rdf:ID="userName">
  <rdfs:domain rdf:resource="#User" />
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
  </owl:DatatypeProperty>
        - <!--
  -->
- <owl:DatatypeProperty rdf:ID="userDescription">
  <rdfs:domain rdf:resource="#User" />
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
  </owl:DatatypeProperty>
```

```
            - <!--




   -->
- <rdf:Property rdf:ID="userCategory">
   <rdfs:domain rdf:resource="#User" />
   <rdfs:range
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/User.owl#UserCategory" />
    </rdf:Property>
            - <!--




   -->
- <owl:DatatypeProperty rdf:ID="Location">
   <rdfs:domain rdf:resource="#User" />
   <rdfs:range rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/User.owl#Location"
      />
    </owl:DatatypeProperty>
            - <!--




   -->
- <owl:DatatypeProperty rdf:ID="userQuality">
   <rdfs:domain rdf:resource="#User" />
   <rdfs:range
      rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/ service/ User.owl# userQuality" />
    </owl:DatatypePropoerty>
    </rdf:RDF>
```

Upper Ontology for Service—Relationship.owl

Source: http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Relationship.owl

```
  <?xml version="1.0" ?>
        - <!--




   -->
- <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
      xmlns:owl="http://www.w3.org/2002/07/owl#"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
      xmlns:service="http://www.dur.ac.uk/ye.zhang/owl-bsu/ service/ Service.owl#"
      xmlns:content="http://www.dur.ac.uk/ye.zhang/owl-bsu/ service/ Content.owl#"
      xmlns:operation="http://www.dur.ac.uk/ ye.zhang/owl-bsu/service/Operation.owl#"
      xmlns:user="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/User.owl#"
      xmlns:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Resource.owl#"
      xmlns="http:// www.dur.ac.uk/ ye.zhang/owl-bsu/ service/Relationship.owl#">
- <owl:Ontology rdf:about="">
   <owl:versionInfo>$Id: Relationship.owl,v 0.5 2004/06/15 13:32:14 $</owl:versionInfo>
   <rdfs:comment>OWL ontology for Advertisements (i.e. Resource). This file belongs to the
      OWL-BSU 0.5 beta Release. Initial version created by Ye Zhang (ye.zhang@dur.ac.uk) and
      William Song (w.w.song@dur.ac.uk).</rdfs:comment>
   <rdfs:seeAlso rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service" />
   <rdfs:seeAlso rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/business" />
              - <!--
    -->
   <owl:imports rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/ service/ Service.owl" />
   <owl:imports rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/ service/ Operation.owl" />
   <owl:imports rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Content.owl" />
   <owl:imports rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Resource.owl" />
   <owl:imports rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/ User.owl" />
```

```
    </owl:Ontology>
  - <!--                                                        -->
  - <!--                                                        -->
  - <!--                                                        -->
  - <!--                                                        -->
  - <!--                                                        -->
              - <!--

    -->
- <rdf:Property rdf:ID="Relationship">
   <rdfs:domain rdf:resource="#Relationship" />
   <rdfs:range rdf:resource="#Relationship" />
    </rdf:Property>
- <owl:DatatypeProperty rdf:ID="S2SRelationship">
   <rdfs:domain rdf:resource="#Relationship" />
   <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Relationship.owl#S2SRelat
     ionship" />
    </owl:DatatypeProperty>
- <owl:DatatypeProperty rdf:ID="S2BRelationship">
   <rdfs:domain rdf:resource="#Relationship" />
   <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Relationship.owl#S2BRelat
     ionship" />
    </owl:DatatypeProperty>
  - <!--

    -->
   <owl:Class rdf:ID="S2SRelationship" />
              - <!--

    -->
- <owl:DatatypeProperty rdf:ID="SomeDistinctFrom">
   <rdfs:domain rdf:resource="#S2SRelationship" />
   <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Relationship.owl#SomeDis
     tinctFrom" />
    </owl:DatatypeProperty>
              - <!--

    -->
- <owl:DatatypeProperty rdf:ID="AllDistinctFrom">
   <rdfs:domain rdf:resource="#S2SRelationship" />
   <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Relationship.owl#AllDistin
     ctFrom" />
    </owl:DatatypeProperty>
              - <!--

    -->
- <owl:DatatypeProperty rdf:ID="Equivalent">
   <rdfs:domain rdf:resource="#S2SRelationship" />
   <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Relationship.owl#Equivale
     nt" />
    </owl:DatatypeProperty>
  - <!--

    -->
   <owl:Class rdf:ID="S2BRelationship" />
              - <!--

    -->
- <owl:DatatypeProperty rdf:ID="SomePropertiesFrom">
```

```
  <rdfs:domain rdf:resource="#S2BRelationship" />
  <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Relationship.owl#SomePro
     pertiesFrom" />
    </owl:DatatypeProperty>
 - <!--

    -->
- <owl:DatatypeProperty rdf:ID="AllPropertiesFrom">
  <rdfs:domain rdf:resource="#S2BRelationship" />
  <rdfs:range
     rdf:resource="http://www.dur.ac.uk/ye.zhang/owl-bsu/service/Relationship.owl#AllPrope
     rtiesFrom" />
    </owl:DatatypeProperty>
    </rdf:RDF>
```

Appendix B

The SWS Matchmaker source code written by VB 6.0

Ontology View

```
Option Explicit
Private sqlstr As String
Private XNode As Node
Private strParent As String
Private Function Init_Tree()
    trvSJZD.Nodes.Clear
    trvSJZD.Style = tvwTreelinesPlusMinusPictureText
    trvSJZD.LabelEdit = tvwManual
    trvSJZD.ImageList = imgTree
    txtNodeName.Tag = ""

    Set XNode = trvSJZD.Nodes.Add(, tvwFirst, "root", "OntologyView", "Root", "Root")
    On Error GoTo AdoError
        Set adoRs = Nothing
        sqlstr = "select dmlb,dmsb,dmsb1,dmsb2,dmsb3,dmmc,scope from test_dmzd"
        Set adoRs = AdoCon.Execute(sqlstr, adCmdText)
        Set adoRs.ActiveConnection = Nothing
        If adoRs.RecordCount < 0 Then
            Init_Tree = 1
        Else
            While Not adoRs.EOF
                If adoRs("dmsb") = -1 Then
                    Set XNode = trvSJZD.Nodes.Add("root", tvwChild, "L" & adoRs("dmlb"), _
                                                Trim(adoRs("dmmc")), "ZD1")
                    XNode.ExpandedImage = "ZD2"

                    XNode.Tag = "dmsb=-1 and dmlb=1 and dmsb1=-1 and dmsb2=-1 and dmsb3=-1"
                ElseIf adoRs("dmsb") >= 0 And adoRs("dmsb1") = -1 And adoRs("dmsb2") = -1 And adoRs("dmsb3") = -1
Then '
                    Set XNode = trvSJZD.Nodes.Add("L" & adoRs("dmlb"), tvwChild, _
                        "I" & adoRs("dmsb") & "L" & adoRs("dmlb"), Trim(adoRs("dmmc")), 4, 5)

                    XNode.Tag = "dmlb=" & adoRs("dmlb") & " and dmsb=" & adoRs("dmsb") & " and dmsb1=-1 and
dmsb2=-1 and dmsb3=-1"
                ElseIf adoRs("dmsb") >= 0 And adoRs("dmsb1") >= 0 And adoRs("dmsb2") = -1 And adoRs("dmsb3") = -1
Then '
                    Set XNode = trvSJZD.Nodes.Add("I" & adoRs("dmsb") & "L" & adoRs("dmlb"), tvwChild, _
                        "P" & adoRs("dmsb1") & "I" & adoRs("dmsb") & "L" & adoRs("dmlb"), Trim(adoRs("dmmc")),
4, 5)

                    XNode.Tag = "dmlb=" & adoRs("dmlb") & " and dmsb=" & adoRs("dmsb") & " and dmsb1=" &
adoRs("dmsb1") & " and dmsb2=-1 and dmsb3=-1"
                ElseIf adoRs("dmsb") >= 0 And adoRs("dmsb1") >= 0 And adoRs("dmsb2") >= 0 And adoRs("dmsb3") = -1
Then '
                    Set XNode = trvSJZD.Nodes.Add("P" & adoRs("dmsb1") & "I" & adoRs("dmsb") & "L" &
adoRs("dmlb"), tvwChild, _
                        "Q" & adoRs("dmsb2") & "P" & adoRs("dmsb1") & "I" & adoRs("dmsb") & "L" &
adoRs("dmlb"), Trim(adoRs("dmmc")), 4, 5)

                    XNode.Tag = "dmlb=" & adoRs("dmlb") & " and dmsb=" & adoRs("dmsb") & " and dmsb1=" &
adoRs("dmsb1") & " and dmsb2=" & adoRs("dmsb2") & " and dmsb3=-1"
                ElseIf adoRs("dmsb") >= 0 And adoRs("dmsb1") >= 0 And adoRs("dmsb2") >= 0 And adoRs("dmsb3") >= 0
Then '
                    Set XNode = trvSJZD.Nodes.Add("Q" & adoRs("dmsb2") & "P" & adoRs("dmsb1") & "I" &
adoRs("dmsb") & "L" & adoRs("dmlb"), tvwChild, _
                        "R" & adoRs("dmsb3") & "Q" & adoRs("dmsb2") & "P" & adoRs("dmsb1") & "I" &
adoRs("dmsb") & "L" & adoRs("dmlb"), Trim(adoRs("dmmc")), 4, 5)

                    XNode.Tag = "dmlb=" & adoRs("dmlb") & " and dmsb=" & adoRs("dmsb") & " and dmsb1=" &
adoRs("dmsb1") & " and dmsb2=" & adoRs("dmsb2") & " and dmsb3=" & adoRs("dmsb3")
                End If
                XNode.Sorted = True
                adoRs.MoveNext
            Wend
            Init_Tree = 0
        End If

'       lblObject.Caption = trvSJZD.Nodes("root").Text
'       lblObjectNum.Caption = trvSJZD.Nodes("root").Children

        Set adoRs = Nothing
        Exit Function
AdoError:
    Set adoRs = Nothing
    Init_Tree = 9

End Function

Private Sub cmdAdd_Click()
```

```
    txtParNode.Text = "":           txtNode.Enabled = True
    txtType.Text = "":          txtNodeName.Text = ""
    cmdAdd.Tag = "Add": cmdSave.Enabled = True
    txtNode.SetFocus

End Sub

Private Sub cmdDel_Click()
    cmdAdd.Tag = "Del"
    Call cmdSave_Click
End Sub

Private Sub cmdModify_Click()
    cmdAdd.Tag = "Modify"
    Call cmdSave_Click
End Sub

Private Sub cmdSave_Click()
On Error GoTo AdoError
    Dim dmlb As Integer
    Dim dmsb As Integer
    Dim dmsb1 As Integer
    Dim dmsb2 As Integer
    Dim dmsb3 As Integer
    If txtNodeName.Tag <> "U" Then
        If Len(txtNodeName.Tag) = 2 Then
            dmlb = Mid(txtNodeName.Tag, InStr(txtNodeName.Tag, "L") + 1, 1)
            dmsb = -1
            dmsb1 = -1
            dmsb2 = -1
            dmsb3 = -1
        ElseIf Len(txtNodeName.Tag) = 4 Then
            dmlb = Mid(txtNodeName.Tag, InStr(txtNodeName.Tag, "L") + 1, 1)
            dmsb = Mid(txtNodeName.Tag, InStr(txtNodeName.Tag, "I") + 1, 1)
            dmsb1 = -1
            dmsb2 = -1
            dmsb3 = -1
        ElseIf Len(txtNodeName.Tag) = 6 Then
            dmlb = Mid(txtNodeName.Tag, InStr(txtNodeName.Tag, "L") + 1, 1)
            dmsb = Mid(txtNodeName.Tag, InStr(txtNodeName.Tag, "I") + 1, 1)
            dmsb1 = Mid(txtNodeName.Tag, InStr(txtNodeName.Tag, "P") + 1, 1)
            dmsb2 = -1
            dmsb3 = -1
        ElseIf Len(txtNodeName.Tag) = 8 Then
            dmlb = Mid(txtNodeName.Tag, InStr(txtNodeName.Tag, "L") + 1, 1)
            dmsb = Mid(txtNodeName.Tag, InStr(txtNodeName.Tag, "I") + 1, 1)
            dmsb1 = Mid(txtNodeName.Tag, InStr(txtNodeName.Tag, "P") + 1, 1)
            dmsb2 = Mid(txtNodeName.Tag, InStr(txtNodeName.Tag, "Q") + 1, 1)
            dmsb3 = -1
        ElseIf Len(txtNodeName.Tag) = 10 Then
            dmlb = Mid(txtNodeName.Tag, InStr(txtNodeName.Tag, "L") + 1, 1)
            dmsb = Mid(txtNodeName.Tag, InStr(txtNodeName.Tag, "I") + 1, 1)
            dmsb1 = Mid(txtNodeName.Tag, InStr(txtNodeName.Tag, "P") + 1, 1)
            dmsb2 = Mid(txtNodeName.Tag, InStr(txtNodeName.Tag, "Q") + 1, 1)
            dmsb3 = Mid(txtNodeName.Tag, InStr(txtNodeName.Tag, "R") + 1, 1)
        End If

    End If

    If txtNode.Tag <> "" Then
        If Left(trvSJZD.Nodes(CInt(txtNode.Tag)).Key, 1) = "L" Or InStr(trvSJZD.Nodes(CInt(txtNode.Tag)).Key, "I") > 0 Or
InStr(trvSJZD.Nodes(CInt(txtNode.Tag)).Key,  "P")  >  0  Or  InStr(trvSJZD.Nodes(CInt(txtNode.Tag)).Key,  "Q")  >  0  Or
InStr(trvSJZD.Nodes(CInt(txtNode.Tag)).Key, "R") > 0 Then '
'           If cInt(txtNode.Text) Then
        Else
                MsgBox "Not valid", vbExclamation, Me.Caption
                If txtNode.Enabled Then
                    txtNode.SetFocus
                    SendKeys "{Home}+{End}"
                End If
                Exit Sub
'           End If
        End If
    End If
    txtNodeName.Text = Trim(txtNodeName.Text)
    If txtNodeName.Text = "" Then
        txtNodeName.SetFocus
        Exit Sub
    End If
    Screen.MousePointer = vbHourglass
    Set AdoCmd = Nothing
    Set AdoCmd = New ADODB.Command
    Set AdoCmd.ActiveConnection = AdoCon
    With AdoCmd
            .CommandText = "sp_test_dmzd_access"
            .CommandType = adCmdStoredProc
    End With
    With AdoCmd.Parameters
        .Append AdoCmd.CreateParameter("RETVAL", adInteger, adParamReturnValue)
        .Append AdoCmd.CreateParameter("access", adTinyInt, adParamInput)
        .Append AdoCmd.CreateParameter("dmlb", adTinyInt, adParamInput)
        .Append AdoCmd.CreateParameter("dmsb", adSmallInt, adParamInput)
        .Append AdoCmd.CreateParameter("dmsb1", adSmallInt, adParamInput)
        .Append AdoCmd.CreateParameter("dmsb2", adSmallInt, adParamInput)
        .Append AdoCmd.CreateParameter("dmsb3", adSmallInt, adParamInput)
        .Append AdoCmd.CreateParameter("srdm", adVarChar, adParamInput, 50)
        .Append AdoCmd.CreateParameter("dmmc", adVarChar, adParamInput, 50)
        .Append AdoCmd.CreateParameter("scope", adTinyInt, adParamInput)
    End With
```

```
With AdoCmd
    If txtNode.Tag = "" Then
        MsgBox "Please Select the Correct Node!", vbInformation + vbOKOnly, Me.Caption
        Set AdoCmd = Nothing
        Exit Sub
    Else
        If cmdAdd.Tag = "Add" Then
            If txtNodeName.Tag <> "U" Then
                .Parameters("access") = 1
                If dmsb = -1 Then
                    .Parameters("dmlb") = CInt(dmlb)
                    .Parameters("dmsb") = CInt(txtNode.Text)
                    .Parameters("dmsb1") = -1
                    .Parameters("dmsb2") = -1
                    .Parameters("dmsb3") = -1
                ElseIf dmsb >= 0 And dmsb1 = -1 And dmsb2 = -1 And dmsb3 = -1 Then '
                    .Parameters("dmlb") = CInt(dmlb)
                    .Parameters("dmsb") = CInt(dmsb)
                    .Parameters("dmsb1") = CInt(txtNode.Text)
                    .Parameters("dmsb2") = -1
                    .Parameters("dmsb3") = -1
                ElseIf dmsb >= 0 And dmsb1 >= 0 And dmsb2 = -1 And dmsb3 = -1 Then '
                    .Parameters("dmlb") = CInt(dmlb)
                    .Parameters("dmsb") = CInt(dmsb)
                    .Parameters("dmsb1") = CInt(dmsb1)
                    .Parameters("dmsb2") = CInt(txtNode.Text)
                    .Parameters("dmsb3") = -1
                ElseIf dmsb >= 0 And dmsb1 >= 0 And dmsb2 >= 0 And dmsb3 = -1 Then '
                    .Parameters("dmlb") = CInt(dmlb)
                    .Parameters("dmsb") = CInt(dmsb)
                    .Parameters("dmsb1") = CInt(dmsb1)
                    .Parameters("dmsb2") = CInt(dmsb2)
                .Parameters("dmsb3") = CInt(txtNode.Text)
                ElseIf dmsb >= 0 And dmsb1 >= 0 And dmsb2 >= 0 And dmsb3 >= 0 Then '
                      MsgBox "Pleae Select the Correct Node!", vbInformation + vbOKOnly, Me.Caption
                    Set AdoCmd = Nothing
                    Exit Sub
                End If

            Else
                MsgBox "Pleae Select the Correct Node!", vbInformation + vbOKOnly, Me.Caption
                Set AdoCmd = Nothing
                Exit Sub
            End If

        ElseIf cmdAdd.Tag = "Modify" Then
            .Parameters("access") = 3
            .Parameters("dmlb") = CInt(dmlb)
            .Parameters("dmsb") = CInt(dmsb)
            .Parameters("dmsb1") = CInt(dmsb1)
            .Parameters("dmsb2") = CInt(dmsb2)
            .Parameters("dmsb3") = CInt(dmsb3)
        ElseIf cmdAdd.Tag = "Del" Then
            .Parameters("access") = 2
            .Parameters("dmlb") = CInt(dmlb)
            .Parameters("dmsb") = CInt(dmsb)
            .Parameters("dmsb1") = CInt(dmsb1)
            .Parameters("dmsb2") = CInt(dmsb2)
            .Parameters("dmsb3") = CInt(dmsb3)
        End If
        If cmdAdd.Tag = "Add" Then
            .Parameters("srdm") = Trim(txtParNode.Tag)
        ElseIf cmdAdd.Tag = "Modify" Then
            .Parameters("srdm") = Trim(lblParName.Caption)
        Else
            .Parameters("srdm") = Trim(lblParName.Caption)

        End If
            .Parameters("dmmc") = Trim(txtNodeName.Text)
            If txtType.Text <> "" Then
                .Parameters("scope") = CInt(txtType.Text)
            Else
                .Parameters("scope") = 0
            End If

            .Execute
        End If
    End With
  If AdoCmd("RETVAL") = -9 Then
    Screen.MousePointer = vbDefault
    MsgBox "Not valid", vbExclamation, Me.Caption
    txtNode.SetFocus
    SendKeys "{Home}+{End}"
  ElseIf AdoCmd("RETVAL") = -8 Then
    MsgBox "Not valid", vbExclamation, Me.Caption
  ElseIf AdoCmd("RETVAL") <> 0 Then
    Screen.MousePointer = vbDefault
    If cmdAdd.Tag = "Add" Then
        MsgBox "Input Wrong", vbCritical, Me.Caption
        txtNode.SetFocus
        SendKeys "{Home}+{End}"
    Else
        MsgBox "Wrong", vbCritical, Me.Caption
        txtNodeName.SetFocus
        SendKeys "{Home}+{End}"
    End If
  Else
    Screen.MousePointer = vbDefault
    If cmdAdd.Tag = "Add" Then
```

```
            MsgBox "Done", vbInformation, Me.Caption
'           Set XNode = trvSJZD.Nodes.Add("L" & txtZdlb.Text, tvwChild, _
'                   "I" & Trim(txtDmsb.Text) & "L" & txtZdlb.Text, Trim(txtNodeName.Text), 4, 5)
'           XNode.Tag = "dmlb=" & txtZdlb.Text & " and dmsb=" & Trim(txtDmsb.Text)
            Call Init_Tree
        ElseIf cmdAdd.Tag = "Modify" Then
            MsgBox "Done    ", vbInformation, Me.Caption
            trvSJZD.Nodes(CInt(txtNode.Tag)).Text = Trim(txtNodeName.Text)
        ElseIf cmdAdd.Tag = "Del" Then
            MsgBox "Done    ", vbInformation, Me.Caption
            trvSJZD.Nodes(CInt(txtNode.Tag)).Text = Trim(txtNodeName.Text)
            Call Init_Tree
        End If
    End If

    Set AdoCmd = Nothing
    Screen.MousePointer = vbDefault
    Exit Sub
AdoError:
    Set AdoCmd = Nothing
    Screen.MousePointer = vbDefault

    pErrHandle Me.Caption
End Sub

Private Sub Command1_Click()
    If txtNodeName.Tag <> "U" Then
        If gFlag = 1 Then
            frmSerReg.txtSerOnto.Text = trvSJZD.Nodes(CInt(txtNode.Tag)).Text
            frmSerReg.txtSerOnto.Tag = txtNodeName.Tag
        ElseIf gFlag = 2 Then '
            frmSerReg.txtUserOnto.Text = trvSJZD.Nodes(CInt(txtNode.Tag)).Text
            frmSerReg.txtUserOnto.Tag = txtNodeName.Tag.
        End If
        Unload Me
    Else
        MsgBox "Please Select the correct Nodes!", vbInformation + vbOKOnly, Me.Caption
        Exit Sub
    End If
End Sub

Private Sub Form_Load()
    Call Init_Tree
    txtNode.Text = ""
    txtParNode.Text = "":
    txtType.Text = ""
    txtNodeName.Text = ""
    lblParName.Caption = ""
End Sub

Private Sub trvSJZD_NodeClick(ByVal Node As MSComctlLib.Node)
     cmdAdd.Enabled = False:        cmdAdd.Tag = ""
     txtParNode.Enabled = False:        cmdSave.Enabled = False
On Error GoTo AdoError
    If Node.Key = "root" Then
        txtNodeName.Text = Node.Text
        txtNode.Enabled = True: txtParNode.Enabled = False
'        txtParNode.Text = adoRs("dmsb")
        txtNode.Text = ""
        txtType.Text = ""
        txtNodeName.Text = ""
        lblParName.Caption = ""
        Node.Tag = "U"
        txtNodeName.Tag = "U"
    Else
        sqlstr = "select dmlb,dmsb,dmsb1,dmsb2,dmsb3,dmmc,scope from test_dmzd where " & Node.Tag
        Set adoRs = Nothing
        Set adoRs = AdoCon.Execute(sqlstr, adCmdText)
        Set adoRs.ActiveConnection = Nothing
        If adoRs.RecordCount >= 1 Then
            If Left(Node.Key, 1) = "L" Then
                    txtParNode.Text = adoRs("dmlb")
                    txtNode.Text = adoRs("dmsb")
                    If adoRs("scope") = "" Or IsNull(adoRs("scope")) Then
                    Else
                        txtType.Text = Trim(adoRs("scope"))
                    End If
                    txtNodeName.Text = Trim(adoRs("dmmc"))
                    lblParName.Caption = Node.Parent.Text

                    cmdSave.Enabled = True: cmdAdd.Enabled = True: cmdDel.Enabled = True: cmdAdd.Tag = "Add"
                    txtNode.Enabled = True: txtParNode.Enabled = False: txtNodeName.Enabled = True
                    txtNodeName.Tag = Node.Key: txtParNode.Tag = Trim(adoRs("dmmc"))
            ElseIf Left(Node.Key, 1) = "I" Then
                    txtParNode.Text = adoRs("dmlb")
                    txtNode.Text = adoRs("dmsb")
                    If adoRs("scope") = "" Or IsNull(adoRs("scope")) Then
                    Else
                        txtType.Text = Trim(adoRs("scope"))
                    End If
                    txtNodeName.Text = Trim(adoRs("dmmc"))
                    lblParName.Caption = Node.Parent.Text

                    cmdSave.Enabled = True: cmdAdd.Enabled = True: cmdDel.Enabled = True: cmdAdd.Tag = "Add"
                    txtNode.Enabled = True: txtParNode.Enabled = False: txtNodeName.Enabled = True
                    txtNodeName.Tag = Node.Key: txtParNode.Tag = Trim(adoRs("dmmc"))
            ElseIf Left(Node.Key, 1) = "P" Then
                    txtParNode.Text = adoRs("dmsb")
                    txtNode.Text = adoRs("dmsb1")
                    If adoRs("scope") = "" Or IsNull(adoRs("scope")) Then
                    Else
                        txtType.Text = Trim(adoRs("scope"))
                    End If
                    txtNodeName.Text = Trim(adoRs("dmmc"))
                    lblParName.Caption = Node.Parent.Text

                    cmdSave.Enabled = True: cmdAdd.Enabled = True: cmdDel.Enabled = True: cmdAdd.Tag = "Add"
                    txtNode.Enabled = True: txtParNode.Enabled = False: txtNodeName.Enabled = True
                    txtNodeName.Tag = Node.Key: txtParNode.Tag = Trim(adoRs("dmmc"))
            ElseIf Left(Node.Key, 1) = "Q" Then
                    txtParNode.Text = adoRs("dmsb1")
                    txtNode.Text = adoRs("dmsb2")
                    If adoRs("scope") = "" Or IsNull(adoRs("scope")) Then
                    Else
                        txtType.Text = Trim(adoRs("scope"))
                    End If
                    txtNodeName.Text = Trim(adoRs("dmmc"))
                    lblParName.Caption = Node.Parent.Text

                    cmdSave.Enabled = True: cmdAdd.Enabled = True: cmdDel.Enabled = True: cmdAdd.Tag = "Add"
                    txtNode.Enabled = True: txtParNode.Enabled = False: txtNodeName.Enabled = True
                    txtNodeName.Tag = Node.Key: txtParNode.Tag = Trim(adoRs("dmmc"))
            ElseIf Left(Node.Key, 1) = "R" Then
                    txtParNode.Text = adoRs("dmsb2")
                    txtNode.Text = adoRs("dmsb3")
                    If adoRs("scope") = "" Or IsNull(adoRs("scope")) Then
                    Else
                        txtType.Text = Trim(adoRs("scope"))
                    End If
                    txtNodeName.Text = Trim(adoRs("dmmc"))
                    lblParName.Caption = Node.Parent.Text

                    cmdSave.Enabled = True: cmdAdd.Enabled = False: cmdDel.Enabled = True
                      txtParNode.Enabled = False
                    txtNodeName.Tag = Node.Key: txtParNode.Tag = Trim(adoRs("dmmc"))
            End If
            txtNode.Tag = Node.Index
        Else
            trvSJZD.Nodes.Remove (Node.Index)
        End If
        If Mid(Node.Key, InStr(Node.Key, "L") + 1, 1) = 1 Then
            Set pic1.Picture = LoadPicture("payment.jpg")
        ElseIf Mid(Node.Key, InStr(Node.Key, "L") + 1, 1) = 2 Then
            Set pic1.Picture = LoadPicture("student.jpg")
        ElseIf Mid(Node.Key, InStr(Node.Key, "L") + 1, 1) = 3 Then
            Set pic1.Picture = LoadPicture("e-journal.jpg")
        End If
    End If
    Set adoRs = Nothing
    Exit Sub
AdoError:
        Set adoRs = Nothing
          pErrHandle Me.Caption
End Sub
```

Business Registration

```
Private Sub Form_Load()
    Call Clear
End Sub

Private Sub txtBusName_KeyPress(KeyAscii As Integer)
    If KeyAscii = vbKeyReturn Then
        KeyAscii = 0
        txtBusDes.SetFocus
    End If
End Sub
Private Sub txtCon_KeyPress(KeyAscii As Integer)
    If KeyAscii = vbKeyReturn Then
        KeyAscii = 0
        txtPhone.SetFocus
    End If
End Sub
Private Sub txtEmail_KeyPress(KeyAscii As Integer)
    If KeyAscii = vbKeyReturn Then
        KeyAscii = 0
        txtPostal.SetFocus
    End If
End Sub

Private Sub txtID_KeyPress(KeyAscii As Integer)
    If KeyAscii = vbKeyReturn Then
        KeyAscii = 0
        txtPassW.SetFocus
    End If
End Sub
Private Sub txtName_KeyPress(KeyAscii As Integer)
      If KeyAscii = vbKeyReturn Then
        KeyAscii = 0
        txtCon.SetFocus
    End If
End Sub

Private Sub txtPassW_KeyPress(KeyAscii As Integer)
    If KeyAscii = vbKeyReturn Then
```

```
        KeyAscii = 0
        txtBusName.SetFocus
    End If
End Sub
Private Sub txtPhone_KeyPress(KeyAscii As Integer)
    If KeyAscii = vbKeyReturn Then
        KeyAscii = 0
        txtEmail.SetFocus
    End If
End Sub
Private Sub txtPostal_KeyPress(KeyAscii As Integer)
    If KeyAscii = vbKeyReturn Then
        KeyAscii = 0
        cmdReg.SetFocus
    End If
End Sub
```

Service Registration

```
Option Explicit
Private strBusId As String
Private Sub Clear()
    Dim Str As String
    With cmbUddi
        .Clear
        .AddItem "Test UDDI Server"
        .ItemData(.NewIndex) = 0
        .AddItem "https://uddi.ibm.com/registry.html"
        .ItemData(.NewIndex) = 1
        .AddItem "https://test.uddi.microsoft.com"
        .ItemData(.NewIndex) = 2
        .ListIndex = 0
    End With
    Str = "select business_key,business_ID   from Business_information where business_key in(select max(business_key) from
Business_information)"
    Set adoRs = Nothing
    Set adoRs = AdoCon.Execute(Str, adCmdText)
    Set adoRs.ActiveConnection = Nothing
    If adoRs.RecordCount >= 1 Then
        With cmbBusKey
             .Clear
            .Tag = adoRs("business_key")
            .AddItem adoRs("business_ID")
            .ListIndex = 0
        End With
    End If
    Set adoRs = Nothing
    txtID.Text = ""
    txtPassW.Text = ""
    txtSerName.Text = ""
    txtCom.Text = ""
    txtAccP.Text = ""
    txtWSDL.Text = ""
    txtCate.Text = "": txtSerOnto.Text = "": txtUserOnto.Text = ""
End Sub
Private Sub cmbBusKey_KeyPress(KeyAscii As Integer)
    If KeyAscii = vbKeyReturn Then
        KeyAscii = 0
        txtSerName.SetFocus
    End If
End Sub
Private Sub cmbUddi_GotFocus()
    SendKeys "%+{Down}"
End Sub
Private Sub cmbUddi_KeyPress(KeyAscii As Integer)
    If KeyAscii = vbKeyReturn Then
        KeyAscii = 0
        txtID.SetFocus
    End If
End Sub

Private Sub cmdClear_Click()
    Call Clear
End Sub

Private Sub cmdReg_Click()
    Dim Str
    Set adoRs = Nothing
    Set adoRs = New ADODB.Recordset


    Str = "select top 1 business_ID from Bus_Id where business_tag=0"
    Set adoRs = AdoCon.Execute(Str, adCmdText)
    Set adoRs.ActiveConnection = Nothing
    strBusId = adoRs(0)
    Set adoRs = Nothing

    Str = "update Bus_Id set business_tag=1 where  business_ID in(select top 1 business_ID from Bus_Id where
business_tag=0)"
    AdoCon.Execute (Str)

    Set adoRs = New ADODB.Recordset
    adoRs.Open "select * from Service_information", AdoCon, adOpenDynamic, adLockOptimistic
    adoRs.AddNew
    adoRs("service_key") = Trim(strBusId)
    adoRs("business_key") = cmbBusKey.Tag
    adoRs("service_name") = Trim(txtSerName.Text)
    adoRs("service_comment") = Trim(txtCom.Text)
    adoRs("service_category") = Trim(txtCate.Text)
    adoRs("service_WSDL") = Trim(txtWSDL.Text)
    adoRs("service_ontology") = Trim(txtSerOnto.Text)
    adoRs("user_ontology") = Trim(txtUserOnto.Text)
    adoRs("service_uddiServer") = cmbUddi.ListIndex
    adoRs.Update
    Set adoRs = Nothing
    txtResult.Text = strBusId
    frmRegBusSuccess.lblKey = strBusId
    frmRegBusSuccess.lblBusConName = Trim(txtSerName.Text)
    frmRegBusSuccess.lblBusDes = Trim(txtCom.Text)
    frmRegBusSuccess.lblBusName = Trim(txtSerName.Text)
    frmRegBusSuccess.lblURL = "http://test.uddi.com/ubr/uddiget?businessKey=" & Trim(strBusId)
    frmRegBusSuccess.Show
End Sub
Private Sub Command3_Click()
    gBusSerFlag = 1
    frmCategoryView.Show
End Sub

Private Sub Command4_Click()
    gFlag = 1
    frmOntoView.Show
End Sub

Private Sub Command5_Click()
    gFlag = 2
    frmOntoView.Show
End Sub

Private Sub Form_Load()
    With cmbUddi
        Call Clear

End Sub

Private Sub txtAccP_KeyPress(KeyAscii As Integer)
    If KeyAscii = vbKeyReturn Then
        KeyAscii = 0
        txtWSDL.SetFocus
    End If
End Sub
Private Sub txtCate_KeyPress(KeyAscii As Integer)
    If KeyAscii = vbKeyReturn Then
        KeyAscii = 0
        txtSerOnto.SetFocus
    End If
End Sub

Private Sub txtCom_KeyPress(KeyAscii As Integer)
    If KeyAscii = vbKeyReturn Then
        KeyAscii = 0
        txtAccP.SetFocus
    End If
End Sub

Private Sub txtID_KeyPress(KeyAscii As Integer)
    If KeyAscii = vbKeyReturn Then
        KeyAscii = 0
        txtPassW.SetFocus
    End If
End Sub
Private Sub txtPassW_KeyPress(KeyAscii As Integer)
    If KeyAscii = vbKeyReturn Then
        KeyAscii = 0
        cmbBusKey.SetFocus
    End If
End Sub
Private Sub txtSerName_KeyPress(KeyAscii As Integer)
    If KeyAscii = vbKeyReturn Then
        KeyAscii = 0
        txtCom.SetFocus
    End If
End Sub
Private Sub txtSerOnto_KeyPress(KeyAscii As Integer)
    If KeyAscii = vbKeyReturn Then
        KeyAscii = 0
        txtUserOnto.SetFocus
    End If
End Sub
Private Sub txtUserOnto_KeyPress(KeyAscii As Integer)
    If KeyAscii = vbKeyReturn Then
        KeyAscii = 0
        cmdReg.SetFocus
    End If
End Sub

Private Sub txtWSDL_KeyPress(KeyAscii As Integer)
    If KeyAscii = vbKeyReturn Then
        KeyAscii = 0
        txtCate.SetFocus
    End If
End Sub
```

Search

```
Option Explicit
'Test UDDI Server.
'
'https://uddi.ibm.com/registry.html
'
'https://test.uddi.microsoft.com
Dim SerOntoId As String
Dim UserOntoId As String
Dim nowSerNode() As String
Dim nowDesNode() As String
Dim nowBusNode() As String
Dim oldNumber As Integer
Dim gFlag As Integer
Dim gUS As Integer
Dim gCFlag As Integer
Dim i As Integer
Private sqlstr As String
Private Sub Clear()
    With cmbUddiServer
        .Clear
        .AddItem "Test UDDI Server"
        .ItemData(.NewIndex) = 0
        .AddItem "https://uddi.ibm.com/registry.html"
        .ItemData(.NewIndex) = 1
        .AddItem "https://test.uddi.microsoft.com"
        .ItemData(.NewIndex) = 2
        .ListIndex = 0
    End With
End Sub
Private Sub GetChildNodes()

        oldNumber = 0
        If gCFlag = 1 Then
            If gUS = 0 Then
                sqlstr = "select dmmc from test_dmzd where dmlb in(select dmlb from test_dmzd where dmmc='" &
txtSerOn.Text & "')"
            ElseIf gUS = 1 Then
                sqlstr = "select dmmc from test_dmzd where dmlb in(select dmlb from test_dmzd where dmmc='" &
txtUserOnto.Text & "')"
            End If
        ElseIf gCFlag = 2 Then '
            If gUS = 0 Then
                sqlstr = "select dmmc from test_dmzd where  dmlb in(select dmlb from test_dmzd where dmmc='" &
txtSerOn.Text & "')   and dmsb in (select dmsb from test_dmzd where dmmc='" & txtSerOn.Text & "')"
            ElseIf gUS = 1 Then
                sqlstr = "select dmmc from test_dmzd where  dmlb in(select dmlb from test_dmzd where dmmc='" &
txtUserOnto.Text & "')   and dmsb in (select dmsb from test_dmzd where dmmc='" & txtUserOnto.Text & "')"
            End If
        ElseIf gCFlag = 3 Then
            If gUS = 0 Then
                sqlstr = "select dmmc from test_dmzd where  dmlb in(select dmlb from test_dmzd where dmmc='" &
txtSerOn.Text & "')   and dmsb in (select dmsb from test_dmzd where dmmc='" & txtSerOn.Text & "') and dmsb1 in (select
dmsb1 from test_dmzd where dmmc='" & txtSerOn.Text & "')"
            ElseIf gUS = 1 Then
                sqlstr = "select dmmc from test_dmzd where  dmlb in(select dmlb from test_dmzd where dmmc='" &
txtUserOnto.Text & "')   and dmsb in (select dmsb from test_dmzd where dmmc='" & txtUserOnto.Text & "') and dmsb1 in
(select dmsb1 from test_dmzd where dmmc='" & txtUserOnto.Text & "')"
            End If
        ElseIf gCFlag = 4 Then
            If gUS = 0 Then
                sqlstr = "select dmmc from test_dmzd where  dmlb in(select dmlb from test_dmzd where dmmc='" &
txtSerOn.Text & "')   and dmsb in (select dmsb from test_dmzd where dmmc='" & txtSerOn.Text & "') and dmsb1 in (select
dmsb1 from test_dmzd where dmmc='" & txtSerOn.Text & "') and dmsb2 in (select dmsb2 from test_dmzd where dmmc='" &
txtSerOn.Text & "')"
            ElseIf gUS = 1 Then
                sqlstr = "select dmmc from test_dmzd where  dmlb in(select dmlb from test_dmzd where dmmc='" &
txtUserOnto.Text & "')   and dmsb in (select dmsb from test_dmzd where dmmc='" & txtUserOnto.Text & "') and dmsb1 in
(select dmsb1 from test_dmzd where dmmc='" & txtUserOnto.Text & "') and dmsb2 in (select dmsb2 from test_dmzd where
dmmc='" & txtUserOnto.Text & "')"
            End If
        ElseIf gCFlag = 5 Then
            If gUS = 0 Then
                sqlstr = "select dmmc from test_dmzd where  dmlb in(select dmlb from test_dmzd where dmmc='" &
txtSerOn.Text & "')   and dmsb in (select dmsb from test_dmzd where dmmc='" & txtSerOn.Text & "') and dmsb1 in (select
dmsb1 from test_dmzd where dmmc='" & txtSerOn.Text & "') and dmsb2 in (select dmsb2 from test_dmzd where dmmc='" &
txtSerOn.Text & "') and dmsb3 in (select dmsb3 from test_dmzd where dmmc='" & txtSerOn.Text & "')"
            ElseIf gUS = 1 Then
                sqlstr = "select dmmc from test_dmzd where  dmlb in(select dmlb from test_dmzd where dmmc='" &
txtUserOnto.Text & "')   and dmsb in (select dmsb from test_dmzd where dmmc='" & txtUserOnto.Text & "') and dmsb1 in
(select dmsb1 from test_dmzd where dmmc='" & txtUserOnto.Text & "') and dmsb2 in (select dmsb2 from test_dmzd where
dmmc='" & txtUserOnto.Text & "') and dmsb3 in (select dmsb3 from test_dmzd where dmmc='" & txtUserOnto.Text & "')"
            End If
        End If
        Set adoRs = Nothing
        Set adoRs = AdoCon.Execute(sqlstr, adCmdText)
        Set adoRs.ActiveConnection = Nothing
        While Not adoRs.EOF
            If gFlag = 0 Then
                sqlstr                                    =                                    "select
distinct(a.business_key),b.service_key,a.business_name,b.service_comment,b.service_key,b.service_name ,b.service_ontology,b.
user_ontology from Business_information a,Service_information b ,test_dmzd c where     b.business_key=a.business_key "
                sqlstr = sqlstr & " and b.service_ontology ='" & adoRs("dmmc") & "' and b.user_ontology='" &
Trim(txtUserOnto.Text) & "'"
            ElseIf gFlag = 1 Then
```

```
                    sqlstr                                        =                                        "select
distinct(a.business_key),b.service_key,a.business_name,b.service_comment,b.service_key,b.service_name  ,b.service_ontology,b.
user_ontology from Business_information a,Service_information b ,test_dmzd c where     b.business_key=a.business_key "
                    sqlstr = sqlstr & " and b.service_ontology ='" & adoRs("dmmc") & "' "
                ElseIf gFlag = 2 Then
                    sqlstr                                        =                                        "select
distinct(a.business_key),b.service_key,a.business_name,b.service_comment,b.service_key,b.service_name  ,b.service_ontology,b.
user_ontology from Business_information a,Service_information b ,test_dmzd c where     b.business_key=a.business_key "
                    sqlstr = sqlstr & " and b.user_ontology='" & Trim(txtUserOnto.Text) & "'"
                ElseIf gFlag = 3 Then
                    sqlstr                                        =                                        "select
distinct(a.business_key),b.service_key,a.business_name,b.service_comment,b.service_key,b.service_name  ,b.service_ontology,b.
user_ontology from Business_information a,Service_information b ,test_dmzd c where     b.business_key=a.business_key "

                End If
                Set adoRs1 = AdoCon.Execute(sqlstr, adCmdText)
                Set adoRs1.ActiveConnection = Nothing
                If adoRs1.RecordCount >= 1 Then

                    ReDim Preserve nowSerNode(oldNumber + adoRs1.RecordCount)
                    ReDim Preserve nowDesNode(oldNumber + adoRs1.RecordCount)
                    ReDim Preserve nowBusNode(oldNumber + adoRs1.RecordCount)
                    For i = oldNumber + 1 To oldNumber + adoRs1.RecordCount
                        nowSerNode(i) = adoRs1("service_name")
                        nowDesNode(i) = adoRs1("service_comment")
                        nowBusNode(i) = adoRs1("business_name")
                        adoRs1.MoveNext
                    Next i
                    oldNumber = oldNumber + adoRs1.RecordCount
                End If
                Set adoRs1 = Nothing
                adoRs.MoveNext
            Wend
            Set adoRs = Nothing
            Set adoRs1 = Nothing
End Sub

Private Sub cmdView_Click()
    gBusSerFlag = 2
    frmCategoryView.Show
End Sub

Private Sub Form_Load()
    Call Clear
End Sub

Private Sub Picture1_Click()

End Sub

Private Sub txtSerName_GotFocus()
    SendKeys "{Home}+{End}"
End Sub

Private Sub txtSerName_KeyPress(KeyAscii As Integer)
    If KeyAscii = vbKeyReturn Then
        KeyAscii = 0
        txtCate.SetFocus
    End If
End Sub

Private Sub txtSerOn_GotFocus()
    SendKeys "{Home}+{End}"
End Sub
Private Sub txtUserOnto_GotFocus()
    SendKeys "{Home}+{End}"
End Sub
Private Sub xpClear_Click()
    txtSerOn.Text = "": txtUserOnto.Text = "": txtSerName.Text = "": txtCate.Text = ""
    Call Clear
End Sub
Private Sub xpExit_Click()
    Unload Me
End Sub
Private Sub xpSearch_Click()
    Dim intX As Integer
    Dim i As Integer
    Dim J As Integer

    If txtSerOn.Text <> "" And txtUserOnto.Text <> "" Then
        gFlag = 0
        gUS = 0
        Set adoRs = Nothing
        sqlstr = "select service_ontoid,user_ontoid from Service_information where service_ontology='" & Trim(txtSerOn.Text)
& "' and user_ontology='" & Trim(txtUserOnto.Text) & "'"
        Set adoRs = AdoCon.Execute(sqlstr, adCmdText)
        Set adoRs.ActiveConnection = Nothing
        If adoRs.RecordCount >= 1 Then
            SerOntoId = Trim(adoRs("service_ontoid"))
            UserOntoId = Trim(adoRs("user_ontoid"))
        Else
            MsgBox "Please put the corret words!", vbInformation + vbOKOnly, Me.Caption
            Set adoRs = Nothing
            Exit Sub
        End If
        Set adoRs = Nothing
        If Len(SerOntoId) = 2 Then
            gCFlag = 1
           Call GetChildNodes
        ElseIf Len(SerOntoId) = 4 Then
                    gCFlag = 2
            Call GetChildNodes
                        Set adoRs = Nothing
            sqlstr = "select dmmc from test_dmzd where dmmc<>'" & Trim(txtSerOn.Text) & "' and   dmlb in(select dmlb
from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and dmsb=-1 and dmsb1=-1 and dmsb2=-1 and dmsb3=-1 "
            Set adoRs = AdoCon.Execute(sqlstr, adCmdText)
            Set adoRs.ActiveConnection = Nothing
            While Not adoRs.EOF
                sqlstr                                          =                                          "select
distinct(a.business_key),b.service_key,a.business_name,b.service_comment,b.service_key,b.service_name  ,b.service_ontology,b.
user_ontology from Business_information a,Service_information b ,test_dmzd c where     b.business_key=a.business_key "
                sqlstr = sqlstr & "  and b.service_ontology ='" & adoRs("dmmc") & "' and b.user_ontology='" &
Trim(txtUserOnto.Text) & "'"
                Set adoRs1 = AdoCon.Execute(sqlstr, adCmdText)
                Set adoRs1.ActiveConnection = Nothing
                If adoRs1.RecordCount >= 1 Then

                    ReDim Preserve nowSerNode(oldNumber + adoRs1.RecordCount)
                    ReDim Preserve nowDesNode(oldNumber + adoRs1.RecordCount)
                    ReDim Preserve nowBusNode(oldNumber + adoRs1.RecordCount)
                    For i = oldNumber + 1 To oldNumber + adoRs1.RecordCount
                        nowSerNode(i) = adoRs1("service_name")
                        nowDesNode(i) = adoRs1("service_comment")
                        nowBusNode(i) = adoRs1("business_name")
                        adoRs1.MoveNext
                    Next i
                    oldNumber = oldNumber + adoRs1.RecordCount
                End If
                Set adoRs1 = Nothing
                adoRs.MoveNext
            Wend
        ElseIf Len(SerOntoId) = 6 Then
                        gCFlag = 3
            Call GetChildNodes
                        Set adoRs = Nothing
            sqlstr = "select dmmc from test_dmzd where dmlb = (select dmlb from test_dmzd where dmmc='" &
Trim(txtSerOn.Text) & "') and dmsb=(select dmlb from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and dmsb1=-1
or"
            sqlstr = sqlstr & "  dmlb=(select dmlb from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and
dmsb=-1 and dmsb1=-1 and dmsb2=-1 and dmsb3=-1"
            Set adoRs = AdoCon.Execute(sqlstr, adCmdText)
            Set adoRs.ActiveConnection = Nothing
            While Not adoRs.EOF
                sqlstr                                          =                                          "select
distinct(a.business_key),b.service_key,a.business_name,b.service_comment,b.service_key,b.service_name  ,b.service_ontology,b.
user_ontology from Business_information a,Service_information b ,test_dmzd c where     b.business_key=a.business_key "
                sqlstr = sqlstr & "  and b.service_ontology ='" & adoRs("dmmc") & "' and b.user_ontology='" &
Trim(txtUserOnto.Text) & "'"
                Set adoRs1 = AdoCon.Execute(sqlstr, adCmdText)
                Set adoRs1.ActiveConnection = Nothing
                If adoRs1.RecordCount >= 1 Then

                    ReDim Preserve nowSerNode(oldNumber + adoRs1.RecordCount)
                    ReDim Preserve nowDesNode(oldNumber + adoRs1.RecordCount)
                    ReDim Preserve nowBusNode(oldNumber + adoRs1.RecordCount)
                    For i = oldNumber + 1 To oldNumber + adoRs1.RecordCount
                        nowSerNode(i) = adoRs1("service_name")
                        nowDesNode(i) = adoRs1("service_comment")
                        nowBusNode(i) = adoRs1("business_name")
                        adoRs1.MoveNext
                    Next i
                    oldNumber = oldNumber + adoRs1.RecordCount
                End If
                Set adoRs1 = Nothing
                adoRs.MoveNext
            Wend

        ElseIf Len(SerOntoId) = 8 Then
                    gCFlag = 4
            Call GetChildNodes
                        sqlstr = "select * from test_dmzd where dmlb=(select dmlb from test_dmzd where dmmc='" &
Trim(txtSerOn.Text) & "') and dmsb=(select dmsb from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and
dmsb1=(select dmsb1 from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and dmsb2=-1 and dmsb3=-1"
            sqlstr = sqlstr & " or dmlb=(select dmlb from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and
dmsb=-1 and dmsb1=-1 and dmsb2=-1 and dmsb3=-1 or dmlb=(select dmlb from test_dmzd where dmmc='" &
Trim(Trim(txtSerOn.Text)) & "') and dmsb=(select dmsb from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and
dmsb1=-1"
            Set adoRs = AdoCon.Execute(sqlstr, adCmdText)
            Set adoRs.ActiveConnection = Nothing
            While Not adoRs.EOF
                sqlstr                                          =                                          "select
distinct(a.business_key),b.service_key,a.business_name,b.service_comment,b.service_key,b.service_name  ,b.service_ontology,b.
user_ontology from Business_information a,Service_information b ,test_dmzd c where     b.business_key=a.business_key "
                sqlstr = sqlstr & "  and b.service_ontology ='" & adoRs("dmmc") & "' and b.user_ontology='" &
Trim(txtUserOnto.Text) & "'"
                Set adoRs1 = AdoCon.Execute(sqlstr, adCmdText)
                Set adoRs1.ActiveConnection = Nothing
                If adoRs1.RecordCount >= 1 Then

                    ReDim Preserve nowSerNode(oldNumber + adoRs1.RecordCount)
                    ReDim Preserve nowDesNode(oldNumber + adoRs1.RecordCount)
                    ReDim Preserve nowBusNode(oldNumber + adoRs1.RecordCount)
                    For i = oldNumber + 1 To oldNumber + adoRs1.RecordCount
                        nowSerNode(i) = adoRs1("service_name")
                        nowDesNode(i) = adoRs1("service_comment")
                        nowBusNode(i) = adoRs1("business_name")
                        adoRs1.MoveNext
                    Next i
```

```
          oldNumber = oldNumber + adoRs1.RecordCount
        End If
        Set adoRs1 = Nothing
        adoRs.MoveNext
      Wend
    ElseIf Len(SerOntoId) = 10 Then
           gCFlag = 5
      Call GetChildNodes
          Set adoRs = Nothing
      sqlstr = "select * from test_dmzd where (dmlb=(select dmlb from test_dmzd where dmmc='" &
Trim(txtSerOn.Text) & "') and dmsb=(select dmsb from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and dmsb1=-1 "
      sqlstr = sqlstr & " or dmlb=(select dmlb from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and
dmsb=-1 and dmsb1=-1 and dmsb2=-1 and dmsb3=-1"
      sqlstr = sqlstr & " or dmlb=(select dmlb from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and
dmsb=(select dmsb from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and dmsb1=(select dmsb1 from test_dmzd
where dmmc='" & Trim(txtSerOn.Text) & "') "
      sqlstr = sqlstr & " and dmsb2=(select dmsb2 from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "')  and
dmsb3=-1 or dmlb=(select dmlb from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and dmsb=(select dmsb from
test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') "
      sqlstr = sqlstr & " and dmsb1=(select dmsb1 from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and
dmsb2=-1 and dmsb3=-1 ) or dmmc='" & Trim(txtSerOn.Text) & "'"
      Set adoRs = AdoCon.Execute(sqlstr, adCmdText)
      Set adoRs.ActiveConnection = Nothing
      While Not adoRs.EOF
        sqlstr = "select
distinct(a.business_key),b.service_key,a.business_name,b.service_comment,b.service_key,b.service_name ,b.service_ontology,b.
user_ontology from Business_information a,Service_information b ,test_dmzd c where   b.business_key=a.business_key "
        sqlstr = sqlstr & " and b.service_ontology ='" & adoRs("dmmc") & "' and b.user_ontology='" &
Trim(txtUserOnto.Text) & "'"
        Set adoRs1 = AdoCon.Execute(sqlstr, adCmdText)
        Set adoRs1.ActiveConnection = Nothing
        If adoRs1.RecordCount >= 1 Then

          ReDim Preserve nowSerNode(oldNumber + adoRs1.RecordCount)
          ReDim Preserve nowDesNode(oldNumber + adoRs1.RecordCount)
          ReDim Preserve nowBusNode(oldNumber + adoRs1.RecordCount)
          For i = oldNumber + 1 To oldNumber + adoRs1.RecordCount
            nowSerNode(i) = adoRs1("service_name")
            nowDesNode(i) = adoRs1("service_comment")
            nowBusNode(i) = adoRs1("business_name")
            adoRs1.MoveNext
          Next i
          oldNumber = oldNumber + adoRs1.RecordCount
        End If
        Set adoRs1 = Nothing
        adoRs.MoveNext
      Wend
   End If
  ElseIf txtSerOn.Text <> "" And txtUserOnto.Text = "" Then
    gFlag = 1
    gUS = 0
    sqlstr = "select service_ontoid ,user_ontoid,service_ontology,user_ontology from Service_information where
service_ontology='" & Trim(txtSerOn.Text) & "' "
    Set adoRs = AdoCon.Execute(sqlstr, adCmdText)
    Set adoRs.ActiveConnection = Nothing
    If adoRs.RecordCount >= 1 Then
      SerOntoId = Trim(adoRs("service_ontoid"))
      UserOntoId = Trim(adoRs("user_ontoid"))
    Else
      Set adoRs = Nothing
      sqlstr = "select * from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "'"
      Set adoRs = AdoCon.Execute(sqlstr, adCmdText)
      Set adoRs.ActiveConnection = Nothing
      If adoRs.RecordCount >= 1 Then
        If adoRs("dmsb") >= 0 And adoRs("dmsb1") >= 0 And adoRs("dmsb2") >= 0 And adoRs("dmsb3") >= 0
Then   '
          gCFlag = 5
          Call GetChildNodes
                    Set adoRs = Nothing
          sqlstr = "select * from test_dmzd where (dmlb=(select dmlb from test_dmzd where dmmc='" &
Trim(txtSerOn.Text) & "') and dmsb=(select dmsb from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and dmsb1=-1 "
          sqlstr = sqlstr & " or dmlb=(select dmlb from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "')
and dmsb=-1 and dmsb1=-1 and dmsb2=-1 and dmsb3=-1"
          sqlstr = sqlstr & " or dmlb=(select dmlb from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "')
and dmsb=(select dmsb from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and dmsb1=(select dmsb1 from test_dmzd
where dmmc='" & Trim(txtSerOn.Text) & "') "
          sqlstr = sqlstr & " and dmsb2=(select dmsb2 from test_dmzd where dmmc='" & Trim(txtSerOn.Text) &
"')  and dmsb3=-1 or dmlb=(select dmlb from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and dmsb=(select dmsb
from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') "
          sqlstr = sqlstr & " and dmsb1=(select dmsb1 from test_dmzd where dmmc='" & Trim(txtSerOn.Text) &
"') and dmsb2=-1 and dmsb3=-1 ) or dmmc='" & Trim(txtSerOn.Text) & "'"
          Set adoRs = AdoCon.Execute(sqlstr, adCmdText)
          Set adoRs.ActiveConnection = Nothing
          While Not adoRs.EOF
            sqlstr = "select
distinct(a.business_key),b.service_key,a.business_name,b.service_comment,b.service_key,b.service_name ,b.service_ontology,b.
user_ontology from Business_information a,Service_information b ,test_dmzd c where   b.business_key=a.business_key "
            sqlstr = sqlstr & " and b.service_ontology ='" & adoRs("dmmc") & "' "
            Set adoRs1 = AdoCon.Execute(sqlstr, adCmdText)
            Set adoRs1.ActiveConnection = Nothing
            If adoRs1.RecordCount >= 1 Then

              ReDim Preserve nowSerNode(oldNumber + adoRs1.RecordCount)
              ReDim Preserve nowDesNode(oldNumber + adoRs1.RecordCount)
              ReDim Preserve nowBusNode(oldNumber + adoRs1.RecordCount)
              For i = oldNumber + 1 To oldNumber + adoRs1.RecordCount
                nowSerNode(i) = adoRs1("service_name")
                nowDesNode(i) = adoRs1("service_comment")
                nowBusNode(i) = adoRs1("business_name")
```

```
                               adoRs1.MoveNext
                           Next i
                           oldNumber = oldNumber + adoRs1.RecordCount
                       End If
                       Set adoRs1 = Nothing
                       adoRs.MoveNext
                   Wend
               ElseIf adoRs("dmsb") >= 0 And adoRs("dmsb1") >= 0 And adoRs("dmsb2") >= 0 And adoRs("dmsb3") < 0
Then    '              gCFlag = 4
                   Call GetChildNodes
                                          Set adoRs = Nothing
                   sqlstr = "select * from test_dmzd where dmlb=(select dmlb from test_dmzd where dmmc='" &
Trim(txtSerOn.Text) & "') and dmsb=(select dmsb from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and
dmsb1=(select dmsb1 from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and dmsb2=-1 and dmsb3=-1"
                   sqlstr = sqlstr & " or dmlb=(select dmlb from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "')
and dmsb=-1 and dmsb1=-1 and dmsb2=-1 and dmsb3=-1 or dmlb=(select dmlb from test_dmzd where dmmc='" &
Trim(Trim(txtSerOn.Text)) & "') and dmsb=(select dmsb from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and
dmsb1=-1"
                   Set adoRs = AdoCon.Execute(sqlstr, adCmdText)
                   Set adoRs.ActiveConnection = Nothing
                   While Not adoRs.EOF
                       sqlstr                                        =                                  "select
distinct(a.business_key),b.service_key,a.business_name,b.service_comment,b.service_key,b.service_name ,b.service_ontology,b.
user_ontology from Business_information a,Service_information b ,test_dmzd c where    b.business_key=a.business_key "
                       sqlstr = sqlstr & " and b.service_ontology ='" & adoRs("dmmc") & "' "
                       Set adoRs1 = AdoCon.Execute(sqlstr, adCmdText)
                       Set adoRs1.ActiveConnection = Nothing
                       If adoRs1.RecordCount >= 1 Then

                           ReDim Preserve nowSerNode(oldNumber + adoRs1.RecordCount)
                           ReDim Preserve nowDesNode(oldNumber + adoRs1.RecordCount)
                           ReDim Preserve nowBusNode(oldNumber + adoRs1.RecordCount)
                           For i = oldNumber + 1 To oldNumber + adoRs1.RecordCount
                               nowSerNode(i) = adoRs1("service_name")
                               nowDesNode(i) = adoRs1("service_comment")
                               nowBusNode(i) = adoRs1("business_name")
                               adoRs1.MoveNext
                           Next i
                           oldNumber = oldNumber + adoRs1.RecordCount
                       End If
                       Set adoRs1 = Nothing
                       adoRs.MoveNext
                   Wend
               ElseIf adoRs("dmsb") >= 0 And adoRs("dmsb1") >= 0 And adoRs("dmsb2") < 0 And adoRs("dmsb3") < 0
Then '
                   gCFlag = 3
                   Call GetChildNodes
                                           Set adoRs = Nothing
                   sqlstr = "select dmmc from test_dmzd where dmlb = (select dmlb from test_dmzd where dmmc='" &
Trim(txtSerOn.Text) & "') and dmsb=(select dmlb from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and dmsb1=-1
or"
                   sqlstr = sqlstr & " dmlb=(select dmlb from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and
dmsb=-1 and dmsb1=-1 and dmsb2=-1 and dmsb3=-1"
                   Set adoRs = AdoCon.Execute(sqlstr, adCmdText)
                   Set adoRs.ActiveConnection = Nothing
                   While Not adoRs.EOF
                       sqlstr                                        =                                  "select
distinct(a.business_key),b.service_key,a.business_name,b.service_comment,b.service_key,b.service_name ,b.service_ontology,b.
user_ontology from Business_information a,Service_information b ,test_dmzd c where    b.business_key=a.business_key "
                       sqlstr = sqlstr & " and b.service_ontology ='" & adoRs("dmmc") & "' "
                       Set adoRs1 = AdoCon.Execute(sqlstr, adCmdText)
                       Set adoRs1.ActiveConnection = Nothing
                       If adoRs1.RecordCount >= 1 Then

                           ReDim Preserve nowSerNode(oldNumber + adoRs1.RecordCount)
                           ReDim Preserve nowDesNode(oldNumber + adoRs1.RecordCount)
                           ReDim Preserve nowBusNode(oldNumber + adoRs1.RecordCount)
                           For i = oldNumber + 1 To oldNumber + adoRs1.RecordCount
                               nowSerNode(i) = adoRs1("service_name")
                               nowDesNode(i) = adoRs1("service_comment")
                               nowBusNode(i) = adoRs1("business_name")
                               adoRs1.MoveNext
                           Next i
                           oldNumber = oldNumber + adoRs1.RecordCount
                       End If
                       Set adoRs1 = Nothing
                       adoRs.MoveNext
                   Wend
               ElseIf adoRs("dmsb") >= 0 And adoRs("dmsb1") < 0 And adoRs("dmsb2") < 0 And adoRs("dmsb3") < 0
Then
                   gCFlag = 2
                   Call GetChildNodes
                                            Set adoRs = Nothing
                   sqlstr = "select dmmc from test_dmzd where dmmc<>'" & Trim(txtSerOn.Text) & "' and dmlb
in(select dmlb from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and dmsb=-1 and dmsb1=-1 and dmsb2=-1 and
dmsb3=-1 "
                   Set adoRs = AdoCon.Execute(sqlstr, adCmdText)
                   Set adoRs.ActiveConnection = Nothing
                   While Not adoRs.EOF
                       sqlstr                                        =                                  "select
distinct(a.business_key),b.service_key,a.business_name,b.service_comment,b.service_key,b.service_name ,b.service_ontology,b.
user_ontology from Business_information a,Service_information b ,test_dmzd c where    b.business_key=a.business_key "
                       sqlstr = sqlstr & " and b.service_ontology ='" & adoRs("dmmc") & "' "
                       Set adoRs1 = AdoCon.Execute(sqlstr, adCmdText)
                       Set adoRs1.ActiveConnection = Nothing
                       If adoRs1.RecordCount >= 1 Then

                           ReDim Preserve nowSerNode(oldNumber + adoRs1.RecordCount)
                           ReDim Preserve nowDesNode(oldNumber + adoRs1.RecordCount)
```

```
                                ReDim Preserve nowBusNode(oldNumber + adoRs1.RecordCount)
                                For i = oldNumber + 1 To oldNumber + adoRs1.RecordCount
                                    nowSerNode(i) = adoRs1("service_name")
                                    nowDesNode(i) = adoRs1("service_comment")
                                    nowBusNode(i) = adoRs1("business_name")
                                    adoRs1.MoveNext
                                Next i
                                oldNumber = oldNumber + adoRs1.RecordCount
                            End If
                            Set adoRs1 = Nothing
                            adoRs.MoveNext
                        Wend
                    ElseIf adoRs("dmsb") < 0 And adoRs("dmsb1") < 0 And adoRs("dmsb2") < 0 And adoRs("dmsb3") < 0 And
adoRs("dmlb") >= 0 Then '
                        gCFlag = 1
                        Call GetChildNodes
                    End If
                End If
            End If
            Set adoRs = Nothing
            If Len(SerOntoId) = 2 Then
                        gCFlag = 1
               Call GetChildNodes
            ElseIf Len(SerOntoId) = 4 Then
                gCFlag = 2
                Call GetChildNodes
                      Set adoRs = Nothing
                sqlstr = "select dmmc from test_dmzd where dmmc<>'" & Trim(txtSerOn.Text) & "' and   dmlb in(select dmlb
from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and dmsb=-1 and dmsb1=-1 and dmsb2=-1 and dmsb3=-1 "
                Set adoRs = AdoCon.Execute(sqlstr, adCmdText)
                Set adoRs.ActiveConnection = Nothing
                While Not adoRs.EOF
                    sqlstr                                                    =                                              "select
distinct(a.business_key),b.service_key,a.business_name,b.service_comment,b.service_key,b.service_name  ,b.service_ontology,b.
user_ontology from Business_information a,Service_information b ,test_dmzd c where    b.business_key=a.business_key "
                    sqlstr = sqlstr & " and b.service_ontology ='" & adoRs("dmmc") & "' "
                    Set adoRs1 = AdoCon.Execute(sqlstr, adCmdText)
                    Set adoRs1.ActiveConnection = Nothing
                    If adoRs1.RecordCount >= 1 Then

                        ReDim Preserve nowSerNode(oldNumber + adoRs1.RecordCount)
                        ReDim Preserve nowDesNode(oldNumber + adoRs1.RecordCount)
                        ReDim Preserve nowBusNode(oldNumber + adoRs1.RecordCount)
                        For i = oldNumber + 1 To oldNumber + adoRs1.RecordCount
                            nowSerNode(i) = adoRs1("service_name")
                            nowDesNode(i) = adoRs1("service_comment")
                            nowBusNode(i) = adoRs1("business_name")
                            adoRs1.MoveNext
                        Next i
                        oldNumber = oldNumber + adoRs1.RecordCount
                    End If
                    Set adoRs1 = Nothing
                    adoRs.MoveNext
                Wend
            ElseIf Len(SerOntoId) = 6 Then
                gCFlag = 3
                Call GetChildNodes

                Set adoRs = Nothing
                sqlstr = "select dmmc from test_dmzd where dmlb = (select dmlb from test_dmzd where dmmc='" &
Trim(txtSerOn.Text) & "') and dmsb=(select dmlb from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and dmsb1=-1
or"
                sqlstr = sqlstr & " dmlb=(select dmlb from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and
dmsb=-1 and dmsb1=-1 and dmsb2=-1 and dmsb3=-1"
                Set adoRs = AdoCon.Execute(sqlstr, adCmdText)
                Set adoRs.ActiveConnection = Nothing
                While Not adoRs.EOF
                    sqlstr                                                    =                                              "select
distinct(a.business_key),b.service_key,a.business_name,b.service_comment,b.service_key,b.service_name  ,b.service_ontology,b.
user_ontology from Business_information a,Service_information b ,test_dmzd c where    b.business_key=a.business_key "
                    sqlstr = sqlstr & " and b.service_ontology ='" & adoRs("dmmc") & "' "
                    Set adoRs1 = AdoCon.Execute(sqlstr, adCmdText)
                    Set adoRs1.ActiveConnection = Nothing
                    If adoRs1.RecordCount >= 1 Then

                        ReDim Preserve nowSerNode(oldNumber + adoRs1.RecordCount)
                        ReDim Preserve nowDesNode(oldNumber + adoRs1.RecordCount)
                        ReDim Preserve nowBusNode(oldNumber + adoRs1.RecordCount)
                        For i = oldNumber + 1 To oldNumber + adoRs1.RecordCount
                            nowSerNode(i) = adoRs1("service_name")
                            nowDesNode(i) = adoRs1("service_comment")
                            nowBusNode(i) = adoRs1("business_name")
                            adoRs1.MoveNext
                        Next i
                        oldNumber = oldNumber + adoRs1.RecordCount
                    End If
                    Set adoRs1 = Nothing
                    adoRs.MoveNext
                Wend

            ElseIf Len(SerOntoId) = 8 Then

                gCFlag = 4
                Call GetChildNodes

                       Set adoRs = Nothing
                sqlstr = "select * from test_dmzd where dmlb=(select dmlb from test_dmzd where dmmc='" &
Trim(txtSerOn.Text) & "') and dmsb=(select dmsb from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and
dmsb1=(select dmsb1 from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and dmsb2=-1 and dmsb3=-1"
```

```
                sqlstr = sqlstr & " or dmlb=(select dmlb from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and
dmsb=-1 and dmsb1=-1 and dmsb2=-1 and dmsb3=-1 or dmlb=(select dmlb from test_dmzd where dmmc='" &
Trim(Trim(txtSerOn.Text)) & "') and dmsb=(select dmsb from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and
dmsb1=-1"
                Set adoRs = AdoCon.Execute(sqlstr, adCmdText)
                Set adoRs.ActiveConnection = Nothing
                While Not adoRs.EOF
                    sqlstr                                                  =                                                  "select
distinct(a.business_key),b.service_key,a.business_name,b.service_comment,b.service_key,b.service_name ,b.service_ontology,b.
user_ontology from Business_information a,Service_information b ,test_dmzd c where    b.business_key=a.business_key "
                    sqlstr = sqlstr & " and b.service_ontology ='" & adoRs("dmmc") & "' "
                    Set adoRs1 = AdoCon.Execute(sqlstr, adCmdText)
                    Set adoRs1.ActiveConnection = Nothing
                    If adoRs1.RecordCount >= 1 Then

                        ReDim Preserve nowSerNode(oldNumber + adoRs1.RecordCount)
                        ReDim Preserve nowDesNode(oldNumber + adoRs1.RecordCount)
                        ReDim Preserve nowBusNode(oldNumber + adoRs1.RecordCount)
                        For i = oldNumber + 1 To oldNumber + adoRs1.RecordCount
                            nowSerNode(i) = adoRs1("service_name")
                            nowDesNode(i) = adoRs1("service_comment")
                            nowBusNode(i) = adoRs1("business_name")
                            adoRs1.MoveNext
                        Next i
                        oldNumber = oldNumber + adoRs1.RecordCount
                    End If
                    Set adoRs1 = Nothing
                    adoRs.MoveNext
                Wend
            ElseIf Len(SerOntoId) = 10 Then

                gCFlag = 5
                Call GetChildNodes

                Set adoRs = Nothing
                sqlstr = "select * from test_dmzd where (dmlb=(select dmlb from test_dmzd where dmmc='" &
Trim(txtSerOn.Text) & "') and dmsb=(select dmsb from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and dmsb1=-1 "
                sqlstr = sqlstr & " or dmlb=(select dmlb from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and
dmsb=-1 and dmsb1=-1 and dmsb2=-1 and dmsb3=-1"
                sqlstr = sqlstr & " or dmlb=(select dmlb from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and
dmsb=(select dmsb from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and dmsb1=(select dmsb1 from test_dmzd
where dmmc='" & Trim(txtSerOn.Text) & "') "
                sqlstr = sqlstr & " and dmsb2=(select dmsb2 from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "')  and
dmsb3=-1 or dmlb=(select dmlb from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and dmsb=(select dmsb from
test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') "
                sqlstr = sqlstr & " and dmsb1=(select dmsb1 from test_dmzd where dmmc='" & Trim(txtSerOn.Text) & "') and
dmsb2=-1 and dmsb3=-1 ) or dmmc='" & Trim(txtSerOn.Text) & "'"
                Set adoRs = AdoCon.Execute(sqlstr, adCmdText)
                Set adoRs.ActiveConnection = Nothing
                While Not adoRs.EOF
                    sqlstr                                                  =                                                  "select
distinct(a.business_key),b.service_key,a.business_name,b.service_comment,b.service_key,b.service_name ,b.service_ontology,b.
user_ontology from Business_information a,Service_information b ,test_dmzd c where    b.business_key=a.business_key "
                    sqlstr = sqlstr & " and b.service_ontology ='" & adoRs("dmmc") & "' "
                    Set adoRs1 = AdoCon.Execute(sqlstr, adCmdText)
                    Set adoRs1.ActiveConnection = Nothing
                    If adoRs1.RecordCount >= 1 Then

                        ReDim Preserve nowSerNode(oldNumber + adoRs1.RecordCount)
                        ReDim Preserve nowDesNode(oldNumber + adoRs1.RecordCount)
                        ReDim Preserve nowBusNode(oldNumber + adoRs1.RecordCount)
                        For i = oldNumber + 1 To oldNumber + adoRs1.RecordCount
                            nowSerNode(i) = adoRs1("service_name")
                            nowDesNode(i) = adoRs1("service_comment")
                            nowBusNode(i) = adoRs1("business_name")
                            adoRs1.MoveNext
                        Next i
                        oldNumber = oldNumber + adoRs1.RecordCount
                    End If
                    Set adoRs1 = Nothing
                    adoRs.MoveNext
                Wend
            End If

        ElseIf txtSerOn.Text = "" And txtUserOnto.Text <> "" Then
            gFlag = 2
            gUS = 1
            sqlstr = "select service_ontoid ,user_ontoid,service_ontology,user_ontology from Service_information where
user_ontology='" & Trim(txtUserOnto.Text) & "'"
            Set adoRs = AdoCon.Execute(sqlstr, adCmdText)
            Set adoRs.ActiveConnection = Nothing
            If adoRs.RecordCount >= 1 Then
                SerOntoId = Trim(adoRs("service_ontoid"))
                UserOntoId = Trim(adoRs("user_ontoid"))
            Else
                Set adoRs = Nothing
                sqlstr = "select * from test_dmzd where dmmc='" & Trim(txtUserOnto.Text) & "'"
                Set adoRs = AdoCon.Execute(sqlstr, adCmdText)
                Set adoRs.ActiveConnection = Nothing
                If adoRs.RecordCount >= 1 Then
                    If adoRs("dmsb") >= 0 And adoRs("dmsb1") >= 0 And adoRs("dmsb2") >= 0 And adoRs("dmsb3") >= 0
Then
                        gCFlag = 5
                        Call GetChildNodes
                                        Set adoRs = Nothing
                        sqlstr = "select * from test_dmzd where (dmlb=(select dmlb from test_dmzd where dmmc='" &
Trim(txtUserOnto.Text) & "') and dmsb=(select dmsb from test_dmzd where dmmc='" & Trim(txtUserOnto.Text) & "') and
dmsb1=-1 "
                        sqlstr = sqlstr & " or dmlb=(select dmlb from test_dmzd where dmmc='" & Trim(txtUserOnto.Text) & "')
and dmsb=-1 and dmsb1=-1 and dmsb2=-1 and dmsb3=-1"
                          sqlstr = sqlstr & " or dmlb=(select dmlb from test_dmzd where dmmc='" & Trim(txtUserOnto.Text) & "')
and dmsb=(select dmsb from test_dmzd where dmmc='" & Trim(txtUserOnto.Text) & "') and dmsb1=(select dmsb1 from
test_dmzd where dmmc='" & Trim(txtUserOnto.Text) & "') "
                          sqlstr = sqlstr & " and dmsb2=(select dmsb2 from test_dmzd where dmmc='" & Trim(txtUserOnto.Text)
& "')  and dmsb3=-1 or dmlb=(select dmlb from test_dmzd where dmmc='" & Trim(txtUserOnto.Text) & "') and dmsb=(select
dmsb from test_dmzd where dmmc='" & Trim(txtUserOnto.Text) & "') "
                          sqlstr = sqlstr & " and dmsb1=(select dmsb1 from test_dmzd where dmmc='" & Trim(txtUserOnto.Text)
& "') and dmsb2=-1 and dmsb3=-1 ) or dmmc='" & Trim(txtUserOnto.Text) & "'"
                          Set adoRs = AdoCon.Execute(sqlstr, adCmdText)
                          Set adoRs.ActiveConnection = Nothing
                          While Not adoRs.EOF
                              sqlstr                                    =                                        "select
distinct(a.business_key),b.service_key,a.business_name,b.service_comment,b.service_key,b.service_name ,b.service_ontology,b.
user_ontology from Business_information a,Service_information b ,test_dmzd c where    b.business_key=a.business_key "
                              sqlstr = sqlstr & " and b.user_ontology='" & Trim(txtUserOnto.Text) & "'"
                              Set adoRs1 = AdoCon.Execute(sqlstr, adCmdText)
                              Set adoRs1.ActiveConnection = Nothing
                              If adoRs1.RecordCount >= 1 Then

                                  ReDim Preserve nowSerNode(oldNumber + adoRs1.RecordCount)
                                  ReDim Preserve nowDesNode(oldNumber + adoRs1.RecordCount)
                                  ReDim Preserve nowBusNode(oldNumber + adoRs1.RecordCount)
                                  For i = oldNumber + 1 To oldNumber + adoRs1.RecordCount
                                      nowSerNode(i) = adoRs1("service_name")
                                      nowDesNode(i) = adoRs1("service_comment")
                                      nowBusNode(i) = adoRs1("business_name")
                                      adoRs1.MoveNext
                                  Next i
                                  oldNumber = oldNumber + adoRs1.RecordCount
                              End If
                              Set adoRs1 = Nothing
                              adoRs.MoveNext
                          Wend
                      ElseIf adoRs("dmsb") >= 0 And adoRs("dmsb1") >= 0 And adoRs("dmsb2") >= 0 And adoRs("dmsb3") < 0
Then
                          gCFlag = 4
                          Call GetChildNodes
                                          Set adoRs = Nothing
                          sqlstr = " select * from test_dmzd where dmlb=(select dmlb from test_dmzd where dmmc='" &
Trim(txtUserOnto.Text) & "') and dmsb=(select dmsb from test_dmzd where dmmc='" & Trim(txtUserOnto.Text) & "') and
dmsb1=(select dmsb1 from test_dmzd where dmmc='" & Trim(txtUserOnto.Text) & "') and dmsb2=-1 and dmsb3=-1"
                          sqlstr = sqlstr & " or dmlb=(select dmlb from test_dmzd where dmmc='" & Trim(txtUserOnto.Text) & "')
and dmsb=-1 and dmsb1=-1 and dmsb2=-1 and dmsb3=-1 or dmlb=(select dmlb from test_dmzd where dmmc='" &
Trim(Trim(txtUserOnto.Text)) & "') and dmsb=(select dmsb from test_dmzd where dmmc='" & Trim(txtUserOnto.Text) & "')
and dmsb1=-1"
                          Set adoRs = AdoCon.Execute(sqlstr, adCmdText)
                          Set adoRs.ActiveConnection = Nothing
                          While Not adoRs.EOF
                              sqlstr                                    =                                        "select
distinct(a.business_key),b.service_key,a.business_name,b.service_comment,b.service_key,b.service_name ,b.service_ontology,b.
user_ontology from Business_information a,Service_information b ,test_dmzd c where    b.business_key=a.business_key "
                              sqlstr = sqlstr & "   and b.user_ontology='" & Trim(txtUserOnto.Text) & "'"
                              Set adoRs1 = AdoCon.Execute(sqlstr, adCmdText)
                              Set adoRs1.ActiveConnection = Nothing
                              If adoRs1.RecordCount >= 1 Then

                                  ReDim Preserve nowSerNode(oldNumber + adoRs1.RecordCount)
                                  ReDim Preserve nowDesNode(oldNumber + adoRs1.RecordCount)
                                  ReDim Preserve nowBusNode(oldNumber + adoRs1.RecordCount)
                                  For i = oldNumber + 1 To oldNumber + adoRs1.RecordCount
                                      nowSerNode(i) = adoRs1("service_name")
                                      nowDesNode(i) = adoRs1("service_comment")
                                      nowBusNode(i) = adoRs1("business_name")
                                      adoRs1.MoveNext
                                  Next i
                                  oldNumber = oldNumber + adoRs1.RecordCount
                              End If
                              Set adoRs1 = Nothing
                              adoRs.MoveNext
                          Wend
                      ElseIf adoRs("dmsb") >= 0 And adoRs("dmsb1") >= 0 And adoRs("dmsb2") < 0 And adoRs("dmsb3") < 0
Then '
                          gCFlag = 3
                          Call GetChildNodes
                                  Set adoRs = Nothing
                          sqlstr = "select dmmc from test_dmzd where dmlb = (select dmlb from test_dmzd where dmmc='" &
Trim(txtUserOnto.Text) & "') and dmsb=(select dmlb from test_dmzd where dmmc='" & Trim(txtUserOnto.Text) & "') and
dmsb1=-1 or "
                          sqlstr = sqlstr & " dmlb=(select dmlb from test_dmzd where dmmc='" & Trim(txtUserOnto.Text) & "')
and   dmsb=-1 and dmsb1=-1 and dmsb2=-1 and dmsb3=-1"
                          Set adoRs = AdoCon.Execute(sqlstr, adCmdText)
                          Set adoRs.ActiveConnection = Nothing
                          While Not adoRs.EOF
                              sqlstr                                    =                                        "select
distinct(a.business_key),b.service_key,a.business_name,b.service_comment,b.service_key,b.service_name ,b.service_ontology,b.
user_ontology from Business_information a,Service_information b ,test_dmzd c where    b.business_key=a.business_key "
                              sqlstr = sqlstr & " and b.user_ontology='" & Trim(txtUserOnto.Text) & "'"
                              Set adoRs1 = AdoCon.Execute(sqlstr, adCmdText)
                              Set adoRs1.ActiveConnection = Nothing
                              If adoRs1.RecordCount >= 1 Then

                                  ReDim Preserve nowSerNode(oldNumber + adoRs1.RecordCount)
                                  ReDim Preserve nowDesNode(oldNumber + adoRs1.RecordCount)
                                  ReDim Preserve nowBusNode(oldNumber + adoRs1.RecordCount)
                                  For i = oldNumber + 1 To oldNumber + adoRs1.RecordCount
                                      nowSerNode(i) = adoRs1("service_name")
                                      nowDesNode(i) = adoRs1("service_comment")
                                      nowBusNode(i) = adoRs1("business_name")
```

```
                                adoRs1.MoveNext
                            Next i
                            oldNumber = oldNumber + adoRs1.RecordCount
                        End If
                        Set adoRs1 = Nothing
                        adoRs.MoveNext
                    Wend
                ElseIf adoRs("dmsb") >= 0 And adoRs("dmsb1") < 0 And adoRs("dmsb2") < 0 And adoRs("dmsb3") < 0
Then
                    gCFlag = 2
                    Call GetChildNodes
                     Set adoRs = Nothing
                    sqlstr = "select dmmc from test_dmzd where dmmc<>'" & Trim(txtUserOnto.Text) & "' and  dmlb
in(select dmlb from test_dmzd where dmmc='" & Trim(txtUserOnto.Text) & "') and dmsb=-1 and dmsb1=-1 and dmsb2=-1 and
dmsb3=-1 "
                    Set adoRs = AdoCon.Execute(sqlstr, adCmdText)
                    Set adoRs.ActiveConnection = Nothing
                    While Not adoRs.EOF
                        sqlstr                                                    =                                    "select
distinct(a.business_key),b.service_key,a.business_name,b.service_comment,b.service_key,b.service_name  ,b.service_ontology,b.
user_ontology from Business_information a,Service_information b ,test_dmzd c where     b.business_key=a.business_key "
                        sqlstr = sqlstr & "and b.user_ontology ='" & adoRs("dmmc") & "' and b.user_ontology='" &
Trim(txtUserOnto.Text) & "'"
                        Set adoRs1 = AdoCon.Execute(sqlstr, adCmdText)
                        Set adoRs1.ActiveConnection = Nothing
                        If adoRs1.RecordCount >= 1 Then

                            ReDim Preserve nowSerNode(oldNumber + adoRs1.RecordCount)
                            ReDim Preserve nowDesNode(oldNumber + adoRs1.RecordCount)
                            ReDim Preserve nowBusNode(oldNumber + adoRs1.RecordCount)
                            For i = oldNumber + 1 To oldNumber + adoRs1.RecordCount
                                nowSerNode(i) = adoRs1("service_name")
                                nowDesNode(i) = adoRs1("service_comment")
                                nowBusNode(i) = adoRs1("business_name")
                                adoRs1.MoveNext
                            Next i
                            oldNumber = oldNumber + adoRs1.RecordCount
                        End If
                        Set adoRs1 = Nothing
                        adoRs.MoveNext
                    Wend
                ElseIf adoRs("dmsb") < 0 And adoRs("dmsb1") < 0 And adoRs("dmsb2") < 0 And adoRs("dmsb3") < 0 Then
                    gCFlag = 1
                    Call GetChildNodes
            End If
        End If
    End If
        Set adoRs = Nothing
        If Len(UserOntoId) = 2 Then
                gCFlag = 1
          Call GetChildNodes
        ElseIf Len(UserOntoId) = 4 Then
            gCFlag = 2
            Call GetChildNodes
                    Set adoRs = Nothing
            sqlstr = "select dmmc from test_dmzd where dmmc<>'" & Trim(txtUserOnto.Text) & "' and   dmlb in(select dmlb
from test_dmzd where dmmc='" & Trim(txtUserOnto.Text) & "') and dmsb=-1 and dmsb1=-1 and dmsb2=-1 and dmsb3=-1 "
            Set adoRs = AdoCon.Execute(sqlstr, adCmdText)
            Set adoRs.ActiveConnection = Nothing
            While Not adoRs.EOF
                sqlstr                                                    =                                    "select
distinct(a.business_key),b.service_key,a.business_name,b.service_comment,b.service_key,b.service_name  ,b.service_ontology,b.
user_ontology from Business_information a,Service_information b ,test_dmzd c where     b.business_key=a.business_key "
                sqlstr = sqlstr & "and b.user_ontology ='" & adoRs("dmmc") & "' and b.user_ontology='" &
Trim(txtUserOnto.Text) & "'"
                Set adoRs1 = AdoCon.Execute(sqlstr, adCmdText)
                Set adoRs1.ActiveConnection = Nothing
                If adoRs1.RecordCount >= 1 Then

                    ReDim Preserve nowSerNode(oldNumber + adoRs1.RecordCount)
                    ReDim Preserve nowDesNode(oldNumber + adoRs1.RecordCount)
                    ReDim Preserve nowBusNode(oldNumber + adoRs1.RecordCount)
                    For i = oldNumber + 1 To oldNumber + adoRs1.RecordCount
                        nowSerNode(i) = adoRs1("service_name")
                        nowDesNode(i) = adoRs1("service_comment")
                        nowBusNode(i) = adoRs1("business_name")
                        adoRs1.MoveNext
                    Next i
                    oldNumber = oldNumber + adoRs1.RecordCount
                End If
                Set adoRs1 = Nothing
                adoRs.MoveNext
            Wend
        ElseIf Len(UserOntoId) = 6 Then
            gCFlag = 3
            Call GetChildNodes
                      Set adoRs = Nothing
            sqlstr = "select dmmc from test_dmzd where dmlb = (select dmlb from test_dmzd where dmmc='" &
Trim(txtUserOnto.Text) & "') and dmsb=(select dmlb from test_dmzd where dmmc='" & Trim(txtUserOnto.Text) & "') and
dmsb1=-1 or "
            sqlstr = sqlstr & " dmlb=(select dmlb from test_dmzd where dmmc='" & Trim(txtUserOnto.Text) & "') and
dmsb=-1 and dmsb1=-1 and dmsb2=-1 and dmsb3=-1"
            Set adoRs = AdoCon.Execute(sqlstr, adCmdText)
            Set adoRs.ActiveConnection = Nothing
            While Not adoRs.EOF
                sqlstr                                                    =                                    "select
distinct(a.business_key),b.service_key,a.business_name,b.service_comment,b.service_key,b.service_name  ,b.service_ontology,b.
user_ontology from Business_information a,Service_information b ,test_dmzd c where     b.business_key=a.business_key "
                sqlstr = sqlstr & " and b.user_ontology='" & Trim(txtUserOnto.Text) & "'"
                Set adoRs1 = AdoCon.Execute(sqlstr, adCmdText)
                Set adoRs1.ActiveConnection = Nothing
                If adoRs1.RecordCount >= 1 Then

                    ReDim Preserve nowSerNode(oldNumber + adoRs1.RecordCount)
                    ReDim Preserve nowDesNode(oldNumber + adoRs1.RecordCount)
                    ReDim Preserve nowBusNode(oldNumber + adoRs1.RecordCount)
                    For i = oldNumber + 1 To oldNumber + adoRs1.RecordCount
                        nowSerNode(i) = adoRs1("service_name")
                        nowDesNode(i) = adoRs1("service_comment")
                        nowBusNode(i) = adoRs1("business_name")
                        adoRs1.MoveNext
                    Next i
                    oldNumber = oldNumber + adoRs1.RecordCount
                End If
                Set adoRs1 = Nothing
                adoRs.MoveNext
            Wend

        ElseIf Len(UserOntoId) = 8 Then
            gCFlag = 4
            Call GetChildNodes
            Set adoRs = Nothing
            sqlstr = " select * from test_dmzd where dmlb=(select dmlb from test_dmzd where dmmc='" &
Trim(txtUserOnto.Text) & "') and dmsb=(select dmsb from test_dmzd where dmmc='" & Trim(txtUserOnto.Text) & "') and
dmsb1=(select dmsb1 from test_dmzd where dmmc='" & Trim(txtUserOnto.Text) & "') and dmsb2=-1 and dmsb3=-1"
            sqlstr = sqlstr & " or dmlb=(select dmlb from test_dmzd where dmmc='" & Trim(txtUserOnto.Text) & "') and
dmsb=-1 and dmsb1=-1 and dmsb2=-1 and dmsb3=-1 or dmlb=(select dmlb from test_dmzd where dmmc='" &
Trim(Trim(txtUserOnto.Text)) & "') and dmsb=(select dmsb from test_dmzd where dmmc='" & Trim(txtUserOnto.Text) & "')
and dmsb1=-1"
            Set adoRs = AdoCon.Execute(sqlstr, adCmdText)
            Set adoRs.ActiveConnection = Nothing
            While Not adoRs.EOF
                sqlstr = "select
distinct(a.business_key),b.service_key,a.business_name,b.service_comment,b.service_key,b.service_name ,b.service_ontology,b.
user_ontology from Business_information a,Service_information b ,test_dmzd c where    b.business_key=a.business_key "
                sqlstr = sqlstr & "    and b.user_ontology='" & Trim(txtUserOnto.Text) & "'"
                Set adoRs1 = AdoCon.Execute(sqlstr, adCmdText)
                Set adoRs1.ActiveConnection = Nothing
                If adoRs1.RecordCount >= 1 Then

                    ReDim Preserve nowSerNode(oldNumber + adoRs1.RecordCount)
                    ReDim Preserve nowDesNode(oldNumber + adoRs1.RecordCount)
                    ReDim Preserve nowBusNode(oldNumber + adoRs1.RecordCount)
                    For i = oldNumber + 1 To oldNumber + adoRs1.RecordCount
                        nowSerNode(i) = adoRs1("service_name")
                        nowDesNode(i) = adoRs1("service_comment")
                        nowBusNode(i) = adoRs1("business_name")
                        adoRs1.MoveNext
                    Next i
                    oldNumber = oldNumber + adoRs1.RecordCount
                End If
                Set adoRs1 = Nothing
                adoRs.MoveNext
            Wend
        ElseIf Len(UserOntoId) = 10 Then
                    gCFlag = 5
            Call GetChildNodes
            '           Set adoRs = Nothing
            sqlstr = "select * from test_dmzd where (dmlb=(select dmlb from test_dmzd where dmmc='" &
Trim(txtUserOnto.Text) & "') and dmsb=(select dmsb from test_dmzd where dmmc='" & Trim(txtUserOnto.Text) & "') and
dmsb1=-1 "
            sqlstr = sqlstr & " or dmlb=(select dmlb from test_dmzd where dmmc='" & Trim(txtUserOnto.Text) & "') and
dmsb=-1 and dmsb1=-1 and dmsb2=-1 and dmsb3=-1"
            sqlstr = sqlstr & " or dmlb=(select dmlb from test_dmzd where dmmc='" & Trim(txtUserOnto.Text) & "') and
dmsb=(select dmsb from test_dmzd where dmmc='" & Trim(txtUserOnto.Text) & "') and dmsb1=(select dmsb1 from test_dmzd
where dmmc='" & Trim(txtUserOnto.Text) & "') "
            sqlstr = sqlstr & " and dmsb2=(select dmsb2 from test_dmzd where dmmc='" & Trim(txtUserOnto.Text) & "')
and dmsb3=-1 or dmlb=(select dmlb from test_dmzd where dmmc='" & Trim(txtUserOnto.Text) & "') and dmsb=(select dmsb
from test_dmzd where dmmc='" & Trim(txtUserOnto.Text) & "') "
            sqlstr = sqlstr & " and dmsb1=(select dmsb1 from test_dmzd where dmmc='" & Trim(txtUserOnto.Text) & "') and
dmsb2=-1 and dmsb3=-1 ) or dmmc='" & Trim(txtUserOnto.Text) & "'"
            Set adoRs = AdoCon.Execute(sqlstr, adCmdText)
            Set adoRs.ActiveConnection = Nothing
            While Not adoRs.EOF
                sqlstr = "select
distinct(a.business_key),b.service_key,a.business_name,b.service_comment,b.service_key,b.service_name ,b.service_ontology,b.
user_ontology from Business_information a,Service_information b ,test_dmzd c where    b.business_key=a.business_key "
                sqlstr = sqlstr & " and b.user_ontology='" & Trim(txtUserOnto.Text) & "'"
                Set adoRs1 = AdoCon.Execute(sqlstr, adCmdText)
                Set adoRs1.ActiveConnection = Nothing
                If adoRs1.RecordCount >= 1 Then

                    ReDim Preserve nowSerNode(oldNumber + adoRs1.RecordCount)
                    ReDim Preserve nowDesNode(oldNumber + adoRs1.RecordCount)
                    ReDim Preserve nowBusNode(oldNumber + adoRs1.RecordCount)
                    For i = oldNumber + 1 To oldNumber + adoRs1.RecordCount
                        nowSerNode(i) = adoRs1("service_name")
                        nowDesNode(i) = adoRs1("service_comment")
                        nowBusNode(i) = adoRs1("business_name")
                        adoRs1.MoveNext
                    Next i
                    oldNumber = oldNumber + adoRs1.RecordCount
                End If
                Set adoRs1 = Nothing
                adoRs.MoveNext
            Wend
        End If
    ElseIf txtSerOn.Text = "" And txtUserOnto.Text = "" Then
```

```
            oldNumber = 0
            sqlstr = "select
distinct(a.business_key),b.service_key,a.business_name,b.service_comment,b.service_key,b.service_name ,b.service_ontology,b.
user_ontology from Business_information a,Service_information b ,test_dmzd c where  b.business_key=a.business_key  and
b.service_name='" & Trim(txtSerName.Text) & "'"
            Set adoRs = AdoCon.Execute(sqlstr, adCmdText)
            Set adoRs.ActiveConnection = Nothing
            If adoRs.RecordCount >= 1 Then

                ReDim Preserve nowSerNode(adoRs.RecordCount)
                ReDim Preserve nowDesNode(adoRs.RecordCount)
                ReDim Preserve nowBusNode(adoRs.RecordCount)
                For i = 1 To adoRs.RecordCount
                    nowSerNode(i) = adoRs("service_name")
                    nowDesNode(i) = adoRs("service_comment")
                    nowBusNode(i) = adoRs("business_name")
                    adoRs.MoveNext
                Next i
                oldNumber = adoRs.RecordCount
            End If
            Set adoRs = Nothing
End If

    J = 0
    For i = 1 To oldNumber
        frmService.lblser(J) = nowSerNode(i)
        frmService.lblDes(J) = nowDesNode(i)
        frmService.Label6(J) = nowBusNode(i)
        J = J + 1
    Next i
    frmService.Show
End Sub
```

Results Display

```
Option Explicit
Private Str As String

Private Sub Label5_Click(Index As Integer)
        Str = "select a.business_ID,a.business_name, b.service_key, b.service_name from Business_information
a ,Service_information b where a.business_name='" & Trim(Label6(Index).Caption) & "' and a.business_key=b.service_ID"
        Set adoRs = AdoCon.Execute(Str, adCmdText)
        Set adoRs.ActiveConnection = Nothing
        If adoRs.RecordCount >= 1 Then
            frmDetails.lblKey.Caption = adoRs("service_key")
            frmDetails.lblownBus.Caption = adoRs("business_name")
            frmDetails.lblOwnKey.Caption = adoRs("business_ID")
            frmDetails.lblSerName.Caption = adoRs("service_name")
              Set adoRs = Nothing
            frmDetails.Show
            frmDetails.SetFocus
        End If

End Sub

Private Sub Label6_Click(Index As Integer)
        Str = "select a.business_ID,a.business_name , b.service_key , b.service_name from Business_information
a ,Service_information b where a.business_name='" & Trim(Label6(Index).Caption) & "' and a.business_key=b.service_ID"
        Set adoRs = AdoCon.Execute(Str, adCmdText)
        Set adoRs.ActiveConnection = Nothing
        If adoRs.RecordCount >= 1 Then
            frmDetails.lblKey.Caption = adoRs("service_key")
            frmDetails.lblownBus.Caption = adoRs("business_name")
            frmDetails.lblOwnKey.Caption = adoRs("business_ID")
            frmDetails.lblSerName.Caption = adoRs("service_name")
            Set adoRs = Nothing
            frmDetails.Show
            frmDetails.SetFocus
        End If

End Sub

Private Sub lblser_Click(Index As Integer)
        Str = "select a.business_ID,a.business_name, b.service_key ,b.service_name from Business_information
a ,Service_information b where a.business_name='" & Trim(Label6(Index).Caption) & "' and a.business_key=b.service_ID"
        Set adoRs = AdoCon.Execute(Str, adCmdText)
        Set adoRs.ActiveConnection = Nothing
        If adoRs.RecordCount >= 1 Then
            frmDetails.lblKey.Caption = adoRs("service_key")
            frmDetails.lblownBus.Caption = adoRs("business_name")
            frmDetails.lblOwnKey.Caption = adoRs("business_ID")
            frmDetails.lblSerName.Caption = adoRs("service_name")
            Set adoRs = Nothing
            frmDetails.Show
            frmDetails.SetFocus
        End If

End Sub

Private Sub XPButton1_Click()
      Unload Me
End Sub

Private Sub XPButton2_Click()
```

```
    Unload Me
End Sub
```

mdlPubFunctions

```
Public Function CheckReserveCode(ByVal CodeStr As String) As Boolean

    If Not IsNumeric(CodeStr) Then
        CheckReserveCode = False
    ElseIf CLng(CodeStr) <> Val(CodeStr) Then
        CheckReserveCode = False
    ElseIf CLng(CodeStr) <= 0 Then
        CheckReserveCode = False
    Else
        CheckReserveCode = True
    End If
End Function
Public Sub DBConnection()

    Dim ConectReturn As String
    Dim sqlstr As String
       GetSetupFile
    ConectReturn = CreateConnection
    If ConectReturn <> "" Then
             frmODBCSetup.ErrorMessages = ConectReturn
        frmODBCSetup.Show
    Else
              sqlstr = "select * from Business_information"
         AdoCon.CursorLocation = adUseClient ' '
         Set adoRs = AdoCon.Execute(sqlstr, , adCmdText)
         Set adoRs.ActiveConnection = Nothing

         Set AdoCmd = Nothing
         Set AdoCmd = New ADODB.Command
         AdoCmd.ActiveConnection = AdoCon
         AdoCmd.CommandText = "select * from Business_information"
         AdoCmd.CommandType = adCmdText
         Set adoRs = Nothing
         Set adoRs = AdoCmd.Execute

    '     LoginSucceeded = True
         Set adoRs = Nothing
         mdiHealthTest.Show
    End If
End Sub
Public Sub GetSetupFile()
'   Dim StringBuffer(10)       As String
    Dim i                       As Integer
         Screen.MousePointer = vbHourglass
    On Error GoTo FileError

    Script = "[DataBase Connection Information for HS Hospital Information System]"
    DSN = "HealthTest"
    UseDSN = 0
    DBMS = "Microsoft SQL Server 2000"
    Server = "localhost"
    DataBase = "HealthTest"
'   DataBase = "999his"
    LogID = ""
    LogPassWd = ""
    DBUserID = "sa"
    DBUserPassWd = ""
    CursorDriver = RdUseOdbc
        Set fsoSetupFile = CreateObject("Scripting.FileSystemObject")
    If Not fsoSetupFile.FileExists(App.Path & "\" & SetupFileName) Then
               PutSetupFile                Else

        Set tsSetupFile = fsoSetupFile.OpenTextFile(App.Path & "\" & SetupFileName, ForReading, False, TristateFalse)

        If Not tsSetupFile.AtEndOfStream Then
            StringBuffer(0) = Get_Value(tsSetupFile.ReadLine, "")
        End If
        i = 1
        While i < 10 And Not tsSetupFile.AtEndOfStream
            StringBuffer(i) = UnLockString(Get_Value(tsSetupFile.ReadLine, "="))
            i = i + 1
        Wend

        Script = StringBuffer(0)
        DSN = StringBuffer(1)
        If Not IsNumeric(StringBuffer(2)) Then
            UseDSN = 0
        Else
            UseDSN = CInt(StringBuffer(2))
        End If
        DBMS = StringBuffer(3)
        Server = StringBuffer(4)
        DataBase = StringBuffer(5)
        LogID = StringBuffer(6)
        LogPassWd = StringBuffer(7)
        DBUserID = StringBuffer(8)
        DBUserPassWd = StringBuffer(9)
        tsSetupFile.Close
    End If

    Screen.MousePointer = vbDefault
    Exit Sub
```

```
FileError:
    Screen.MousePointer = vbDefault
    MsgBox

End Sub
Public Sub PutSetupFile()

    Screen.MousePointer = vbHourglass
    On Error GoTo FileError

    Set fsoSetupFile = CreateObject("Scripting.FileSystemObject")

    Set tsSetupFile = fsoSetupFile.CreateTextFile(App.Path & "\" & SetupFileName, True)
    tsSetupFile.WriteLine (Script)
    tsSetupFile.WriteLine ("DSN = " & LockString(DSN))
    tsSetupFile.WriteLine ("USEDSN = " & LockString(CStr(UseDSN)))
    tsSetupFile.WriteLine ("DBMS = " & LockString(DBMS))
    tsSetupFile.WriteLine ("Server = " & LockString(Server))
    tsSetupFile.WriteLine ("DataBase = " & LockString(DataBase))
    tsSetupFile.WriteLine ("LogID = " & LockString(LogID))
    tsSetupFile.WriteLine ("LogPassWd = " & LockString(LogPassWd))        PassWd
    tsSetupFile.WriteLine ("DBUserID = " & LockString(DBUserID))
    tsSetupFile.WriteLine ("DBUserPassWd = " & LockString(DBUserPassWd)) '
    tsSetupFile.Close

    Screen.MousePointer = vbDefault
    Exit Sub
FileError:
    Screen.MousePointer = vbDefault
    MsgBox ", vbExclamation, "
End Sub
Public Function Get_Value(SearchString As String, SearchChar As String) As String

    Dim Position      As Integer
    If SearchString = "" Then
        Get_Value = ""
    ElseIf SearchChar = "" Then
        Get_Value = SearchString
    Else
        Position = InStr(1, SearchString, SearchChar, vbTextCompare)
        If Position = Null Or Position = 0 Then
            Get_Value = ""
        Else
            Get_Value = Trim(Mid(SearchString, Position + 1))
        End If
    End If
End Function

Public Function UnLockString(LockedString As String) As String

    EncryptTable                                    =                                    "`12sdfg5%$90-=+_)""(*&^#
@!~qw34e8iop[{P]\yu|}OIrt67TREWQahjklKUYJ;':LHFDGScvAzxb/?>m,.NBVC<MnXZ"
    StringLength = Len(LockedString)
    TableLength = Len(EncryptTable)
    For i = 1 To StringLength
        UnlockChar = Mid(LockedString, i, 1)
        For J = 1 To TableLength
            If UnlockChar = Mid(EncryptTable, J, 1) Then

                UnLockString = UnLockString & Chr(J + 31)
            End If
        Next J
    Next i
End Function
Public Function LockString(SourceString As String) As String

    StringLength = Len(SourceString)
    For i = 1 To StringLength
                            LockString = LockString & Mid(EncryptTable, Asc(Mid(SourceString, i, 1)) - 31, 1)
    Next i
End Function

Public Sub pErrHandle(ErrStr As String)

    Dim Msg As String
    If Err.Number <> 0 Then
            & Err.Source & Chr(13) & "    " & Err.Description
        MsgBox Msg, vbCritical + vbOKOnly, ErrStr & "", Err.HelpFile, Err.HelpContext

    End If
End Sub
Public Function CreateConnection() As String

    Dim sConnect          As String
    Dim ErrorMessage      As String

    Screen.MousePointer = vbHourglass
    On Error GoTo AdoError

    If UseDSN = 1 Then
        sConnect = "uid=sa;pwd=;dsn=" & DSN
        'AdoCon.CursorLocation = adUseClient                  '
        Set AdoCon = New ADODB.Connection
        AdoCon.Open sConnect
    Else
        sConnect = "driver={SQL Server};server=localhost;database=MakeMacher;uid=sa;pwd="

'           sConnect = "uid=" & DBUserID & ";pwd=" & DBUserPassWd & ";"
'           sConnect = sConnect & "server=" & Server & ";driver={SQL Server};"
'       sConnect = sConnect & "database=" & DataBase & ";"
'
        Set AdoCon = New ADODB.Connection
        AdoCon.Open sConnect
    End If

    CreateConnection = ""
    Screen.MousePointer = vbDefault
    Exit Function

AdoError:
    pErrHandle
End Function
Public Function ProcessPin(ByVal PassWord As String) As String

    ReDim AsciiPasswWord(1 To Len(PassWord)) As Integer
    ReDim PinString(1 To Len(PassWord)) As Integer
    Dim i As Integer


    If PassWord = "" Then
        ProcessPin = ""
        Exit Function
    End If

    For i = 1 To Len(PassWord)
        AsciiPasswWord(i) = Asc(Mid(PassWord, i, 1))
    Next

    For i = 1 To Len(PassWord)
        If AsciiPasswWord(i) > 250 Then
            PinString(i) = AsciiPasswWord(i) - 251
        Else
            PinString(i) = AsciiPasswWord(i) + 5
        End If
    Next

    ProcessPin = ""
    For i = 1 To Len(PassWord)
        ProcessPin = ProcessPin & Chr(PinString(i))
    Next i
End Function
Public Function Del_Space(ByVal SourceString As Variant) As String

    If IsNull(SourceString) Then
        Del_Space = ""
    Else
        Del_Space = LTrim(RTrim((SourceString)))
    End If
End Function
Public Function Del_Space1(ByVal SourceString As Variant) As String

    If IsNull(SourceString) Then
        Del_Space1 = 0
    Else
        Del_Space1 = LTrim(RTrim((SourceString)))
    End If
End Function
Public Function LocateCmbByIndex(ByRef Cmb As ComboBox, ByVal IndexNum As Integer) As Integer

    Dim Ind As Integer
    LocateCmbByIndex = 1
    With Cmb
        For Ind = 0 To .ListCount - 1
            If .ItemData(Ind) = IndexNum Then
                .ListIndex = IndexNum
                .ListIndex = Ind
                .Text = .List(Ind)
                LocateCmbByIndex = 0
                Exit Function
            End If
        Next Ind
    End With
End Function
Public Sub FillComboxZd(ByVal rs As ADODB.Recordset, Cmb As ComboBox, ByVal Tdmlb As Integer)

    'Tdmlb
    rs.MoveFirst
    With Cmb
        .Clear
        While Not rs.EOF
            If CInt(adoRs("dmlb")) = Tdmlb Then
                Cmb.AddItem (Trim(adoRs("dmmc")))
                Cmb.ItemData(Cmb.NewIndex) = adoRs("dmsb")
            End If
            adoRs.MoveNext
        Wend
        If Cmb.ListCount > 0 Then Cmb.ListIndex = 0
    End With
End Sub
Public Function CompareData(ByVal data1 As Double, ByVal data2 As Double, ByVal data3 As Double) As Double

    If data1 > data2 Then
        CompareData = data1
    Else
        CompareData = data2
    End If
    If CompareData < data3 Then
        CompareData = data3
    End If

End Function
Public Function CompareData1(ByVal data1 As Double, ByVal data2 As Double) As Double

    If data1 > data2 Then
        CompareData1 = data1
    Else
        CompareData1 = data2
    End If
End Function
Private Function mathData1(ByVal mathData As Double) As Double

    mathData = Round(mathData, 1)

End Function
Public Sub showFrm(frm As Form)
    frm.Show
End Sub
```

The SWS Matchmaker Table

```
/*****************************************************************************/
/**************(1) create Business information table                      **/
/**         Created:zy          date:                       **/
/**         Modified:           date:16:12                  **/
/**         Called by:                                      **/
/*****************************************************************************/
create table Business_information
(
    business_key                int             identity (1, 1) not null,
    business_ID                 varchar(50)  not null,
    business_name               varchar(20) not null,
    business_description varchar(40)    null,
    business_category           varchar(100)    null,
    business_uddiServer         tinyint         null,
    bus_con_name                varchar(20)     null,
    bus_con_phone               varchar(20)     null,
    bus_con_email               varchar(20)     null,

    bus_con_Des                         varchar(20)     null,
    bus_con_Postal              varchar(20)     null,
    primary key(business_key)

)
create table Service_information
(
   service_ID                   int             identity (1, 1) not null,
   business_key                         int             not null,
   service_key                          varchar(50)     not null,
   service_name                 varchar(20)         not null,
   service_comment              varchar(40)         null,
   service_category             varchar(100)          not null,
   service_WSDL                 varchar(40)         null,
   service_ontology             varchar(50)         null,
   user_ontology        varchar(50)         null,
   service_ontoid               char(20)        null,
   user_ontoid                  char(20)        null,
   serviceLb                    tinyint         null,
   servicesb            smallint            null,
   servicesb1                   smallint            null,
   servicesb2                   smallint        null,
   servicesb3                   samllint        null,
   userLb               tinyint     null,
   usersb               smallint    null,
   usersb1              smallint    null,
   usersb2              smallint    null,
   usersb3              smallint    null,
   service_uddiServer           tinyint         null,
   primary key(service_key)

)
create table test_dmzd
(
dmlb        tinyint     not     null,
dmsb        smallint    not     null,

dmsb1               smallint    not         null,
dmsb2               smallint    not     null,
dmsb3               smallint    not     null,
srdm        varchar(50) null,
dmmc        varchar(50) not     null,
scope       int         null,

primary key(dmlb,dmsb,dmsb1,dmsb2,dmsb3)
)
insert test_dmzd (dmlb,dmsb,dmsb1,dmsb2,dmsb3,dmmc,scope) values(1,-1,-1,-1,-1,'Electronic Journals',0)
insert test_dmzd(dmlb,dmsb,dmsb1,dmsb2,dmsb3,dmmc,scope) values(2,-1,-1,-1,-1,'Student',0)

create table service_ontology_infromation
(
  service_id        int             identity (1, 1) not null,
  service_ontology                  varchar(20)   not null,
  business_key                      int not null,
  primary key(service_id)
)
create table user_ontology_infromation
(
   user_id            int      identity (1, 1) not null,
   user_ontology                     varchar(20)    not null,
   business_key                      int    not null,
   primary key(user_id)
)
create table Bus_Id
(
   bus_Id            int identity(1,1) not null,
   business_ID              varchar(50)  not null,
   business_tag             int                    not null default 0,
)


/**************************************************************************/
/**No:matchmaker_Test                                              **/
/**Name:sp_test_dmzd_access                                  **/
/**Paremeters:access,dmlb,dmsb,dmsb1,dmsb2,dmsb3,dmmc,pardmmc,type                     **/
/**              access=1:input; access=3:modify;access=2:delet         **/
/**Total:     6 parameters                              **/
/**Call: test_dmzd                                            **/


/**************************************************************************/
create procedure sp_test_dmzd_access
(
@access            tinyint,
@dmlb              tinyint,       @dmsb            smallint,
@dmsb1             smallint,      @dmsb2           smallint,
@dmsb3             smallint,
@srdm              varchar(50),   @dmmc            varchar(50),
@scope             int
)
as
if @access=1
begin
    if exists(select * from test_dmzd where dmlb=@dmlb and dmsb=@dmsb and dmsb1=@dmsb1 and dmsb2=@dmsb2 and
dmsb3=@dmsb3)
        return -9
end
else
begin
    if not exists(select * from test_dmzd where dmlb=@dmlb and dmsb=@dmsb and dmsb1=@dmsb1 and dmsb2=@dmsb2 and
dmsb3=@dmsb3)
        return -8
end
if @access=1
    insert              test_dmzd              (dmlb,dmsb,dmsb1,dmsb2,dmsb3,srdm,dmmc,scope)              values
(@dmlb,@dmsb,@dmsb1,@dmsb2,@dmsb3,@srdm,@dmmc,@scope)
else
begin
    if @access=2
        delete  from  test_dmzd  where  dmlb=@dmlb  and  dmsb=@dmsb  and  dmsb1=@dmsb1  and  dmsb2=@dmsb2  and
dmsb3=@dmsb3
    else
        if @access=3
            update  test_dmzd  set  srdm=@srdm,dmmc=@dmmc,scope=@scope  where  dmlb=@dmlb  and  dmsb=@dmsb  and
dmsb1=@dmsb1 and dmsb2=@dmsb2 and dmsb3=@dmsb3
        else
            return -7
end
if @@error<>0
    return -1
else
    return 0
go
```