UQAC
UNIVERSITÉ DU QUÉBEC
À CHICOUTIMI

MÉMOIRE PRÉSENTÉ À

L'UNIVERSITÉ DU QUÉBEC À CHICOUTIMI

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN INFORMATIQUE

PAR

WEI RAN

B.A.A.

SEMANTIC WEB AND BUSINESS APPLICATION

April 2011

Content

Abstract .... - 1 -

Résumé .... - 2 -

Acknowledgement .... - 3 -

Chapter 1 Introduction .... - 4 -

Chapter 2 Concept of Semantic Web .... - 9 -

2.1. Semantic Web .... - 11 -

2.2. The Resource Description Framework .... - 12 -

2.3. Resource Description Framework Schema .... - 15 -

2.4. Web Ontology Language .... - 15 -

2.5. SPARQL Query Language .... - 17 -

2.6. The Optional Matching .... - 18 -

Chapter 3 Semantic Web Working Process .... - 19 -

3.1. Semantic Web Working Process .... - 20 -

3.2. Description and Examples of RDF .... - 23 -

3.3. XML and RDF .... - 25 -

3.4. Summary .... - 27 -

Chapter 4 A Semantic Web Experiment: The FOAF Project .... - 29 -

4.1. Environment .... - 30 -

4.2. FOAF Project .... - 32 -

4.3. Implementation for a Search Model .... - 33 -

4.3.1. Introduction for the Interactive Search Model .... - 33 -

4.3.2. Sample for the Model .... - 34 -

4.3.3. Sample Model Simulation .... - 35 -

4.3.4. Ambiguity of the Search Result .... - 37 -

4.4. Summary .... - 38 -

Chapter 5 Semantic Web Experiment of a Business Model .... - 39 -

5.1. The Business Model .... - 41 -

5.2. Business Application Characteristic... - 42 -

5.3. Environment .... - 44 -

5.4. Modeling and the Implementation .... - 46 -

5.4.1. Product .... - 46 -

5.4.2. Sales .... - 47 -

5.4.3. Marketing .... - 49 -

5.4.4. Customer Information .... - 53 -

5.4.5. Purchase Report... - 54 -

5.4.6. Generate a Business Plan .... - 58 -

5.4.7 Structure of the application .... - 60 -

5.5. Methodology of Building Semantic Web... - 61 -

5.6. Summary .... - 63 -

Chapter 6 More Technical Aspects of the Semantic Web .... - 65 -

6.1. Security and the Fake Information .... - 66 -

6.1.1. The Authorization .... - 66 -

6.1.2. Access the Data behind RDF .... - 68 -

6.2. Robust and Fault-Tolerant - 69 -

6.3. Artificial Intelligence and Semantic System - 71 -

6.4. Merge the RDF - 72 -

6.5. The Global View of the Semantic Web - 74 -

Chapter 7 Related Work - 78 -

7.1. Semantic Inference Engine - 79 -

7.2. Knowledge Base - 79 -

7.3. Existing Projects Related to Semantic Web - 80 -

Chapter 8 Conclusion - 81 -

Appendix - 84 -

<Appendix A> the Page of the Web Link - 84 -

<Appendix B> Source Code of the Business Model - 85 -

Common.inc - 85 -

Businessinformation.php - 97 -

Customer.php - 101 -

Sales.php - 108 -

Purchase.php - 113 -

Product.php - 116 -

Plan.php - 120 -

Marketing.php - 125 -

Reference - 130 -

List of Figures

Figure 2.2-1 RDF Chart .... - 14 -

Figure 3.2-1 RDF Relation .... - 24 -

Figure 4.1-1 FOAF Project Architecture .... - 31 -

Figure 4.2-1 FOAF display .... - 33 -

Figure 4.3-1 System Structure .... - 34 -

Figure 4.3-2 Query Interface .... - 36 -

Figure 5.1-1 Business Model Structure .... - 41 -

Figure 5.4-1 Product .... - 47 -

Figure 5.4-2 Sales .... - 48 -

Figure 5.4-3 Marketing .... - 52 -

Figure 5.4-4 Customer .... - 54 -

Figure 5.4-5 Purchase Report .... - 57 -

Figure 5.4-6 Product Publication .... - 58 -

Figure 5.4-7 Business Plan .... - 59 -

Figure 6.1-1 Certification Process .... - 67 -

Figure 6.1-2 Security of Semantic Web .... - 69 -

Figure 6.5-1 Multi-level RDF Data .... - 76 -

Abstract

The main topic is about the **Semantic Techniques** and the **RDF** which is used for indicating the resources in the distributed systems over the network (the database). There are implementations for two projects: **FOAF** and a sample by semantic concept. A business model description follows and the technique discussion about the semantic web and the related topics are also shown in the following chapters. A study of the problems and experiences completed on the concepts of a semantic web used to build a small application; not only suggest a practical methodology (that could be used to implement that concept), but also mention the limitations of this new software technology. For a new generation web, the semantic web concept shows a new layer of the web: knowledge level for machines. The new web system offers the opportunity for the computers to understand the content of the data. The thesis demonstrated the machines understand the information and then use it as its own knowledge. This step requires further research which this knowledge standard similar to the complex workings of the human brain. The distributed developing offers the possibility of it and this thesis gives the foundation of the theory.

Résumé

Le sujet principal concerne les **Techniques Sémantiques** et le **RDF** qui est utilisé pour indiquer les ressources dans les systèmes répartis sur l'ensemble du réseau (la base des données). Il y a des mises en œuvre pour deux projets : **FOAF** et un type de concept sémantique. Suit la description du modèle d'entreprise et la discussion sur le réseau sémantique et les sujets connexes sont aussi démontrés dans les chapitres suivants. Une étude des problèmes et des expériences faites sur les concepts d'un réseau sémantique a été utilisée pour intégrer une petite application; non seulement suggérer une méthodologie pratique (qui pourrait servir à exécuter ce concept), mais il y a aussi mention des restrictions de cette nouvelle technologie de logiciel. Le concept de réseau sémantique ajoute un niveau d'abstraction supplémentaire au Web: les machines peuvent manipuler les données comme des connaissances.. Le nouveau système réseau offre l'opportunité aux ordinateurs de comprendre le contenu des données. La thèse démontre que les machines comprennent l'information et l'utilisent comme leur propre connaissance. Cette étape exige une recherche supplémentaire dont la connaissance normale est similaire au complexe fonctionnement du cerveau humain. Le développement de mon sujet offre la possibilité de cela et cette thèse démontre bien la base de cette théorie.

Acknowledgement

Wish to express my sincere gratitude to Professor Paul Girard and Professor Sylvain Boivin, who helped and guided me towards my academic as well as my professional success. This thesis would not have been possible without their grateful help.

For Ryan and Chen.

Chapter 1

Introduction

Late in 1990s, when *google.com* was established, the internet became an amazing tool for everyone in the world. From that time, people searched the information on the internet with more ease and in a quicker manner. Google takes the place of common weather talk or newspaper reading.

As time goes by, the information increases as an explosion every minute. Among that information, the useful information is substantially less in volume. People ask for an intelligent system. Our goal is to talk with the system, and let the system do the most complex job. Up to this point it is the people who do the most complex part and the system completing the search job. By increasing the information on the network, it is more and more difficult to create the list of the keywords that can lead to the right information.

The HTML (Hyper Text Markup Language) was developed to show the multimedia materials to everyone by using a web browser. The advantage of HTML is simple and powerful but it is limited and less flexible. In order to describe the data in different tags (classes), the XML (Extensible Markup Language) was introduced, which is more flexible than HTML. XML is a simplified version of SGML (Standard Generalized Markup Language) and it defines a general way to represent structured data, for example the syntax of a language, regardless of its meaning.

The HTML is derived from SGML and it is intended to represent the content of a Web page (HTML has nothing, except the META, to characterize the nature of the content). Currently there are many applications are designing based on HTML standard.

> SGML is a rather general markup language which was originally designed to enable the sharing of machine-readable documents in large projects in government, law and industry, which have to remain readable for several decades; a very long time in information technology. It has also been used extensively in the printing and publishing industries, but its complexity has prevented its widespread application for small-scale general-purpose use. [SGML][i]

The HTML can be viewed as a special case of XML, when we are talking about the XHTML, actually, XHTML is HTML expressed in XML. In the late 1990s, many considered the future of HTML lay in the creation of a version adhering to the syntax rules of XML. At that time, the HTML 4 was ostensibly an application of Standard Generalized Markup Language (SGML); however the specification for SGML was complex, and neither web-browsers nor the HTML 4 Recommendation were fully conformant with it. By shifting the underlying base from SGML to the simpler XML, HTML would become compatible with common XML tools. Servers and proxies would be able to transform content, as necessary, for constrained devices such as mobile phones.

> Another key advantage was extensibility. By using namespaces, XHTML documents could include fragments from other XML-based languages. Finally, the renewed work would provide an opportunity to divide HTML into reusable components (XHTML Modularization) and clean up untidy parts of the language.[XHTML][ii]

The success of the web was related to the implementation and the power to HTML language. This comes with several limitations regarding the semantic of the documentation contents.

When we are searching the information from the HTML pages, the best that we can do is to search the meta-data from the head with very limited keywords. But it will lead to ambiguous. The RDF (Resource description framework), the definition will be introduced in the following chapter) is not designed for display the content by the explorer but readable for the system. Imagine that we are going to search the information from different web site (servers), if all the information is shown by HTML; the engines are not able to give out the intelligent answer for our query because there are too many formats for those information which are not readable for the system. If we put them into the same format which means the new standards for the contents, which is readable for the engines, it will be easier to search and merge. To obtain the information by the system we need a language that can make web resources more accessible to automated processes. Compare this framework to XML, XML is not an ontology language which means it cannot be understood logically by the system. It is true that the system can read the tags from XML but what do those tags mean to the system? In this framework we define Classes and Properties, Sub/super Classes and Range, domain with the properties, too. The framework can be expressed by XML and the other methods.

RDF is the Resource Description Framework, which is a language expressed in XML, it is used to represent a graph as triples (Subject, Predicate, Object). In semantic web system, the RDF plays an important role. The whole network is considered as an global database in which the semantic system can make the queries and data operation through the help of RDF to explain the data flow.

By reading this report, the readers should be encouraged to establish a new opinion of semantic view of the systems. The difference of the traditional business info-system to semantic web system for business is shown to the readers and the advantages of it are obviously functional for a business with further view of the construction of the information system. For data collections, searching and querying, the semantic web shows a great power for business application.

Chapter 2 introduces the concepts of the semantic web including the basic idea about the semantic web.

After the basic concept introduction, in Chapter 3, it shows the working process of semantic web. Several techniques are used for explaining how the semantic web proceeds and what kind of result that user are expecting.

In Chapter 4 there are the implementation for some semantic web samples, such as FOAF (Friend of a Friend) project and a simulation search model. Following is some information regarding the details of the techniques.

Up to this point the semantic web is still in a theory level but some models have been built to test this concept and evaluate the problems. In Chapter 5 there is a business model which is designed by the idea of the semantic web. The whole chapter will introduce the structure of the business model and its functions. The advantages including the strengths, weakness and difficulties of this business system are discussed after the introduction of the implementation.

In Chapter 6 there are the techniques related to the security, robust, artificial intelligent, and the other discussions about the operations of the semantic web. The related work is described in Chapter 7.

Chapter 2

Concept of Semantic Web

The semantic web is a concept of another layer of the web network. The current web system shows the content by browser directly to the users. The concepts for the web system history are distinguished as web 1.0, 2.0 and 3.0 versions. These versions are not official definitions for World Wide Web but a kind of milestones of the web generations.

> The essential difference between Web 1.0 and Web 2.0 is that content creators were few in Web 1.0 with the vast majority of users simply acting as consumers of content, while any participant can be a content creator in Web 2.0 and numerous technological aids have been created to maximize the potential for content creation.[WDIFF][iii]

The web 1.0 is considered as the classic read-only web system, while web 2.0 is considered as a read-write-able web system. In the mean time, web 3.0 concept was proposed after web 2.0. The web 3.0 concept introduces the semantic intelligent web system. In web 3.0 level, the machine system generates the useful information instead of human brain in a certain part.

By Hypertext Transfer Protocol (HTTP) the data transferred from the server to the client part. The browser resolves the data and display in a certain format so that the user can read it. In the semantic web level, the data transfers from the server to the client, the user does not have to read it. It is the system who reads it. The format of the data has the particular format only for the system to understand. The data is describing the resource in the network. The RDF (Resource Description Framework) is applied for this purpose. The RDFS (Resource Description Framework Schema) is a schema of RDF. These are used for semantic web resource indicating. The OWL (Web Ontology Language) is the knowledge presentation language for semantic web. Also the semantic web layer requires the database operation in order to deal with the data in the RDF format. SPARQL (Simple Protocol and RDF Query Language) is the query language to do it. The algorithm of the data operation can be optimized in diverse methods, in Chapter 2.6 an algorithm is introduced as an example. By the last part of Chapter 2 the problem of the semantic web gives the idea about the developable parts of the semantic web which needs to be solved.

2.1. Semantic Web

The current web structure is based on the web browser or the HTML technology. The whole Internet can be introduced as a global database which contains a huge volume of data. The current system is not intelligent enough to classify the information. The system can tell the weather is sunny and the temperature is -5 C but it does not understand what that means even the system cannot tell the difference of the information from a personal blog or from a weather report centre because they are the same thing for it. One of the goals for semantic web is to maximize the ability of the computer system and minimize the operation by human. Without understanding the information, the system is limited in intelligent treatment of the data. The semantic web concept introduces the method for the system to understand what the data means. Comparing to the traditional method of the web technology, the results that return from an intelligent system should be a list of resources in which the system can make the following treatment. For example, when the user search the key word 'food', traditional search engine gives out the result as a list of URLs (Uniform Resource Locator) such as 'foodnetwork.com' or the 'Food' page in Wikipedia.org. With these information user has to select the proper website to look into it to get the useful information. The system cannot pick up the information from these web sites to research it further. The intelligent system returns the result as a resource list including the resource files which indicates the 'food' knowledge. By understanding what they are and how the resource should be used (which the resource file described) the system is able to give advice to the user that he would be more interesting. Also by reading the resource file which describes the interests of the user, the system can do better adopting job for the returned result. In this example, suppose the user has no idea about Chinese food and search about 'Local Chinese Restaurant'. The system can read his profile from local system or anywhere in the network to understand where he locates and what his interests of the food are. The result from the global database contains the list of Chinese Restaurant resources, which might contains different types but all indicated by the resource list file to explain everything to the system. The system matches the two sides' information to give out the proper suggestion to the user. In this case the system 'selects' for the user instead of leaving everything to the user. Furthermore, the optional operation can include more information about the appropriate result such as contact methods, menu, prices, and comments. A similar model will be introduced in Chapter 3 for the working process. In Chapter 4 there will be an introduction about some simulation of the semantic web application.

Up to this point, there is not a formal definition for semantic web. It can be defined by the goal of the semantic web. The current web is full of data and they are used every day. The semantic web is a web of data, too.

> The vision of the Semantic Web is to extend principles of the Web from documents to data. Data should be accessed using the general Web architecture using, e.g., URI-s; data should be related to one another just as documents (or portions of documents) are already. This also means creation of a common framework that allows data to be shared and reused across application, enterprise, and community boundaries, to be processed automatically by tools as well as manually, including revealing possible new relationships among pieces of data.[SWGL][iv]

The semantic web is not going to replace the current web. Instead, this is another layer for the web contents. It is designed as a parallel expression for the current web. The different is, to express to the machines in order to let them really understand the meaning of the data. For human beings, it is even not necessary to read this data directly. This is something we are able to read as necessary.

2.2.The Resource Description Framework

> The Resource Description Framework (RDF) is a language for representing information about resources in the World Wide Web. It is particularly intended for representing metadata about Web resources, such as the title, author, and modification date of a Web page, copyright and licensing information about a Web document, or the availability schedule for some shared resource. However, by generalizing the concept of a "Web

> resource", RDF can also be used to represent information about things that can be identified on the Web, even when they cannot be directly retrieved from the Web. Examples include information about items available from on-line shopping facilities (e.g., information about specifications, prices, and availability), or the description of a Web user's preferences for information delivery. RDF is intended for situations in which this information needs to be processed by applications, rather than being only displayed to people. RDF provides a common framework for expressing this information so it can be exchanged between applications without loss of meaning. Since it is a common framework, application designers can leverage the availability of common RDF parsers and processing tools. The ability to exchange information between different applications means that the information may be made available to applications other than those for which it was originally created. [RDF][v]

Basically speaking, the RDF data model is not different from classic conceptual model approaches such as Entity-Relationship or Class diagrams. This is based upon the idea of statements about resources; in particular web resources. These are in the form of subject predicate object expressions. These expressions are known as triples in RDF terminology. The subject denotes the resource, and the predicate denotes traits or aspects of the resource and expresses a relationship between the subject and the object. For example, one way to represent the notion "*The sky has the color blue*" in RDF is as the triple: a subject denoting "*the sky*", a predicate denoting "*has the color*", and an object denoting "*blue*".

> RDF is an abstract model with several serialization formats (i.e., file formats), and so the particular way in which a resource or triple is encoded varies from format to format. [RDFOV][vi]

When the concept 'sky' is put into namespace 'www.sample.org/sky #', the graphic shows as the following:

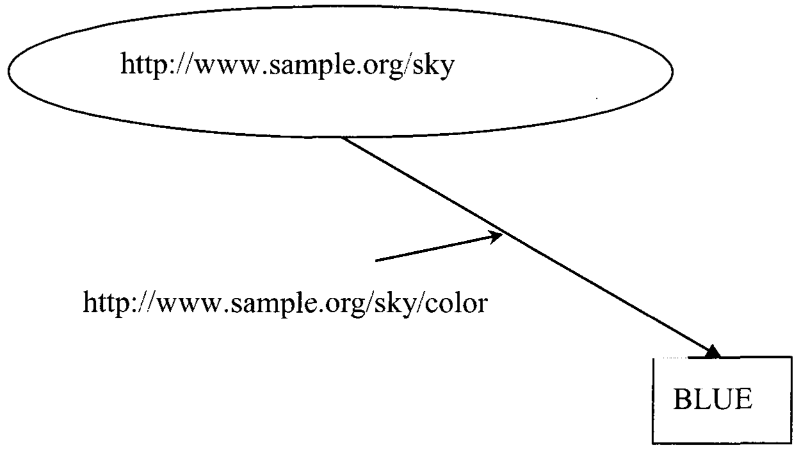


Figure 2.2-1 RDF Chart

This mechanism for describing resources is a major component in what is proposed by the W3C's (The World Wide Web Consortium) Semantic Web activity: an evolutionary stage of the World Wide Web in which automated software can store, exchange, and use machine-readable information distributed throughout the Web, in turn enabling users to deal with the information with greater efficiency and certainty. RDF's simple data model and ability to model disparate, abstract concepts has also led to its increasing use in knowledge management applications unrelated to Semantic Web activity.

A collection of RDF statements intrinsically represents a labeled, directed multi-graph. As such, an RDF-based data model is more naturally suited to certain kinds of knowledge representation than the relational model and other ontological models traditionally used in computing today. However, in practice, RDF data is often persisted in relational database or native representations also called Triple stores, or Quad stores if context (i.e. the named graph) is also persisted for each RDF triple. As RDFS and OWL (Web Ontology Language) demonstrate, additional ontology languages can be built upon RDF.

2.3. Resource Description Framework Schema

RDF Schema (or RDFS) is an extensible knowledge representation language, providing basic elements for the description of ontology, otherwise called RDF vocabularies, intended to structure RDF resources. The first version was published by W3C in April 1998, and the final W3C recommendation was released in February 2004. Main RDFS components are included in the more expressive language OWL. [RDFS][vii]

For example, the following RDFS section shows the sub class relation. This is not to declare the definition but to limit the 'range' of the 'value' for 'earth':

```
<?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#"

xml:base=   "http://www.universe.fake/planet#">

<rdfs:Class rdf:ID="planet" />

<rdfs:Class rdf:ID="earth">

<rdfs:subClassOf rdf:resource="#planet"/>

</rdfs:Class>

</rdf:RDF>
```

2.4. Web Ontology Language

The Web Ontology Language (OWL) is a family of knowledge representation languages for authoring ontology, and is endorsed by the World Wide Web Consortium. This family of languages is based on two (largely, but not entirely, compatible) semantics: OWL DL and OWL Lite semantics are based on Description Logics, which have attractive and well-understood computational properties, while OWL Full uses a novel semantic model intended to provide compatibility with RDF Schema. OWL ontologies are most commonly serialized using RDF/XML syntax. OWL is considered one of the fundamental technologies underpinning the Semantic Web, and has attracted both academic and commercial interest.

> In October 2007, a new W3C working group was started to extend OWL with several new features as proposed in the OWL 1.1 member submission. This new version, called OWL 2, has already found its way into semantic editors such as Protégé and semantic resoners. [OWL][viii]

> The OWL Web Ontology Language is designed for use by applications that need to process the content of information instead of just presenting information to humans. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary along with a formal semantics. [OWLOV][ix]

The Semantic Web will build on XML's ability to define customized tagging schemes and RDF's flexible approach to representing data. The first level above RDF required for the Semantic Web is an ontology language what can formally describe the meaning of terminology used in Web documents. If machines are expected to perform useful reasoning tasks on these documents, the language must go beyond the basic semantics of RDF Schema.

To show the example, we ignore the head part for short:

```
<owl:Class rdf:ID="Winery"/>

<owl:Class rdf:ID="Region"/>

<owl:Class rdf:ID="ConsumableThing"/>

……

<Wine rdf:ID="MikesFavoriteWine>

<owl:sameAs rdf:resource="#StGenevieveTexasWhite" />
```

```
</Wine>

<Wine rdf:ID="MikesFavoriteWine">

<owl:sameAs rdf:resource="#StGenevieveTexasWhite" />

</Wine>
```

OWL files are usually expressed by XML (so far this is the only method).

2.5.SPARQL Query Language

RDF is a directed, labeled graph data format for representing information in the Web. RDF is often used to represent, personal information, social networks, metadata about digital artifacts. This specification defines the syntax and semantics of the SPARQL query language for RDF. This also provides a means of integration over disparate sources of information.

> The SPARQL query language for RDF is designed to meet the use cases and requirements identified by the RDF Data Access Working Group in RDF Data Access Use Cases and Requirements. [SPARQL][x]

SPARQL is an RDF query language; its name is a recursive acronym that stands for SPARQL Protocol and RDF Query Language. It is standardized by the RDF Data Access Working Group (DAWG) of the World Wide Web Consortium, and is considered a component of the semantic web. Initially released as a Candidate Recommendation in April 2006, but returned to Working Draft status in October 2006, due to two open issues. In June 2007, SPARQL advanced to Candidate Recommendation once again. On 12th November 2007 the status of SPARQL changed into Proposed Recommendation. On 15th January 2008, SPARQL became an official W3C Recommendation.

SPARQL allows for a query to consist of triple patterns, conjunctions, disjunctions, and optional patterns.

Several implementations for multiple programming languages exist.

Redland is a general-purpose Resource Description Framework (RDF) library providing APIs to the RDF model and its concepts. In our simulation part, we will show a small project for RDF querying with the optional matching which is supported by the Redland for PHP interface.

The following codes show an example of SPARQL. This is the operation to find out the first result from the RDF or OWL for the one who has the lowest age.

```
SELECT ?name ?sex ?age

WHERE { ?x :name ?name ; :sex ?sex; :age ?age}

ORDER BY DESC(?age)

LIMIT 1
```

2.6. The Optional Matching

When we query something from the database, we always want to get the ‘right’ result. It is possible to ignore something valuable we assume it’s for the reader. For example, we need to find a dentist in a small town and usually, we look up in the yellow page. It might be that there aren’t any registered dentists in this area. Here is the table denoting this:

Province	City	Professional	Title	Name	Address
Quebec	SmileCity	Medication	Doctor	James	20 Rue King

When we query the information, we search the database for ‘dentist’ in ‘Title” of this table but there is only a Doctor in the Medication section. The result can encompass a larger parameter: We can left-join the result for Title, which is the professional column. Our goal is to reveal a useful and educational answer. The left-join method shows how to implement a part for the ‘smart operation’. After this the semantic power should be able to pick up the reasonable answer for the user, which is not only listing all the answers. The optional matching function can be nested looped so that the huge amount of answers will not confuse the user.

Chapter 3

Semantic Web Working Process

Semantic web is still a concept. In this chapter the description about the semantic web will describe what the semantic web is about and how it works.

3.1. Semantic Web Working Process

During the query procedure, suppose that the user is going to Buffalo, NY for a conference. Then the user submits a request as follows:

Request: Buffalo, NY, Conference ABC

After this request is submitted, the system understands that the user is inquiring about a conference in Buffalo, NY for conference ABC. The system sends out the request to the server and searches the database Internet. The returned result will be a list of resources including all kinds of information about the Conference ABC in Buffalo area. These resources can be text, images, voice tracks, videos, related information such as weather report, taxi information, hotels, and restaurants in Buffalo area.

The request can be explained to the system by RDF. The system can understand the parameters about the target goal. There are the branches of the information such as the location, the time, and the other conditions. The returned information includes the information concern these parts. Ideally the system provides the main information about the conference and for the branches part gives out the related advices. The advantage of the semantic search result is that the results are resources that the system can understand. It means that the system can judge which the best matching ones are according to the profile of the user which can be expressed by RDF, too.

For example, the system gets the location information of the user. It knows that the user is not in Buffalo area. The system has to put the transportation and the hotel information into the results. The logic part can be implemented by the specific logical file or by the knowledge base concept which is under research in other projects. The branches information is explored by this kind of logical judgment. The key point is the system has to understand what the resources are so that they can be deal with.

Compared to the semantic web working process, the traditional method offers a query result from a database or from the Internet. The results can be any format but the system understands data results only, it cannot tell the difference between a picture of a bird are from the weight of a steel pipe.. The system does not really understand these 'DATA'. What the system does is offer this information to the user and leave the rest of the work to the human brain.

In this example, Server A, B, C, D, E is the independent servers in a certain network. Each of them offers a different service. Suppose that the services offered from the five servers are:

Server A: Calendar Service

Server B: Air Ticket Reservation Service

Server C: Weather Report Service

Server D: Hotel Reservation Service

Server E: TAXI reservation Service

The agent receives a request:

Client: "I will go to Buffalo area for the weekend if the weather is nice there. Please arrange my trip."

The agent will check the Server A for the date and Server C for the weather in Buffalo area. If the result is positive, the agent will make the reservation for Taxi/Air Ticket/Hotel. During the process, the agent has to be able to find the services among the network and recognize the client's requests. After that the agent should be able to distinguish the differences from the three reservation services and make the right decision. Only when the system fully understands the content of the data from the web, the agent is then able to explore the hierarchy of the plans in order to give out the result including the necessary goals for the request. Also the agent can give the results that distinguished by integrant part and the optional part. The management of the answer depends on the understanding of the knowledge of the agent system.

> RDF Concepts and Abstract Syntax define an abstract syntax on which RDF is based on, and which serves to link its concrete syntax to its formal semantics. It also includes discussion of design goals, key concepts, data typing, character normalization and handling of URI references. [RDFCAS][xi]

> In order to extend effectively the distributed database technology, three problems should be addressed: First, it should be possible to control the placement of data that is based on relationships between data objects. Second problem is the maintenance of referential integrity across sites. The third problem is how to use links that relate data across sites effectively. [DDS][xii]

Usually we do not have a complete database in one system. As the current web, all the information stored in different servers, when we need the information, the search engine will find as much results as it can and we have to judge which results fit our request the best, and this is the job that we want to put in the system instead of our brains. The distributed database all over the Internet could be treated as the database resource. Before we have the web 2.0 standard to be done, it is hard for all the servers to keep the same standard for the data, even after establishing the standards. Because of that, the agent should be robust in functional parts to be able to pick up the useful information in the distribution database. Actually, the semantic web concept is not going to define a new standard for the web, instead, it is going to define a new way to indicate the resources on the internet, which is the way that the machine will be able to understand the data all over the web: where the needed data is, what the data is and how it can be read and used.

Generally, when we talk about the semantic web, what we need is an easy-interactive system. We give only general request and the system gets out all the useful information for us by a certain format which is easy to understand for human beings.

3.2. Description and Examples of RDF

The traditional machine-unreadable content from the web is hard to deal for the machines and they are unable to be inherited by the sub object (class) even after the 'intelligent treatment' for the data. For the semantic web, the RDF file indicates the resource with the triple elements: the subject, the predicator, and the object. There is the way to express the phrase of a natural language, for example: Ryan washes his hand to prevent the H1N1 flu virus. The RDF description is the following:

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

<rdf:RDF

xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

xmlns:prevention="http://www.antiH1N1.org/prevention"

xmlns:exterms="http://www.sample.org/human#">

<rdf:Description rdf:about="http://www.sample.org/huma/ryan">

<exterms:name>Ryan</exterms:name>

<exterms:body>hand</exterms:body>

</rdf:Description>

<rdf:Description rdf:about="http://www.antiH1N1.org/awayfromh1n1">

<prevention:virus>H1N1</prevention:virus>

</rdf:Description>

</rdf:RDF>
```

The RDF shows the resource for the system which is going to understand there is a human whose name is Ryan, hand is a part of his body, and there also exist a virus named H1N1. With this information, the system carries out the treatment for the data by recognizing the resources which are indicated in the RDF file. This is not a simple input, instead, the system understand the meaning of the data. The knowledge structure is shown in the following diagram:

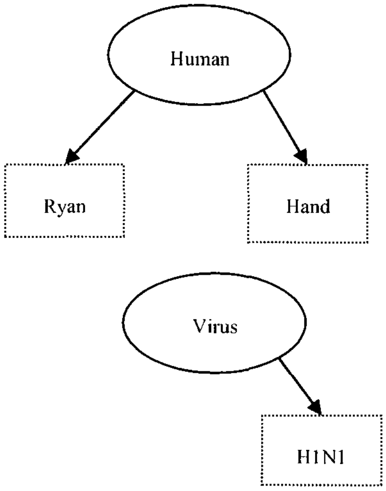


Figure 3.2-1 RDF Relation

The structure is a graph with nodes. It shows the whole structure for the system to understand. The circles stand for objects and the squares stand for the properties (attributes). The lacking verbs can be described by the arrows which can be indicated to a URI to show what this action is, such as 'washing'.

The following graph shows the CD collection in a CD store. It helps the store owner to manage the goods and also helps the clients to choose.

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

<rdf:RDF

xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
```

```
xmlns:cd="http://www.recshop.fake/cd#">

<rdf:Description
 rdf:about="http://www.recshop.fake/cd/Empire Burlesque">
  <cd:artist>Bob Dylan</cd:artist>
  <cd:country>USA</cd:country>
  <cd:company>Columbia</cd:company>
  <cd:price>10.90</cd:price>
  <cd:year>1985</cd:year>
</rdf:Description>

<rdf:Description
 rdf:about="http://www.recshop.fake/cd/Hide your heart">
  <cd:artist>Bonnie Tyler</cd:artist>
  <cd:country>UK</cd:country>
  <cd:company>CBS Records</cd:company>
  <cd:price>9.90</cd:price>
  <cd:year>1988</cd:year>
</rdf:Description>
……
</rdf:RDF>
```

3.3.XML and RDF

This last application provides another point of view. Actually, the RDF is ‘grammatical’ based on XML and it is implemented by XML format. In XML the information are expressed in tags, which are showed as the following format:

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

<xmltags>

    Information

</xmltags>
```

The information within the XML tags is easy to manage for the system when querying.

1. The difference from XML to RDF is the goal. XML is designed to reduce the complexity of the data or information expression during the data exchanging process. That is nothing to do with semantic part. RDF is designed to parse the semantic tree which has the simple triple structure. Even all the RDF can be expressed by XML, it does not mean the XML can be understandable for the system. When the system read the tags from XML, the tags means nothing except a flag without any meaning.

2. In RDF, the node is not only a node with information under a tag; it is always expressing a URI which indicates a resource in the network or database. XML does not have the standards for semantic expressions which RDF has.

3. XML schema is used for constraint of the grammar of XML file. But the RDF schema is used for explaining the meaning of the RDF file elements in it. This is showing the system how to understand the data in the RDF file but there is not any constraint from the RDFS.

4. Even XML can be merged or assembled, rarely people does that. Because during the merging and assembling process of XML, there is strictly constraint from the namespace which RDF has too but does not apply any constraint from the language. Also there comes

the problem ambiguous of XML when merging and the RDF does not have this problem at all because the nodes of RDF are URIs. Though they are pointing to all different contents, they are still in a same type and be understood by the system. But for XML, the system is not able to distinguish the difference of different tags when ambiguous exist and without a complete explanations in a strict grammar.

5. Another difference from XML to RDF is the data model. RDF is a disordered side with the directed graph when XML is an ordered with the tree node.

Generally, RDF is very flexible, and it is designed to represent the resources. All RDF can be expressed by XML but still RDF has other way to express such as a graph or a triple. XML is designed for exchanging information and it has nothing to do with semantic. It is hard to merge and data aggregate for XML when it is an advantage of RDF.

3.4.Summary

The differences between Semantic Web and current web standard are that the current web designing is for people using the different types of the data including the text, pictures, sounds, and videos. And it can be described as a space of a library. The semantic web is designed for the machines to understand what the resources are and what they are about. It is not to say that we are going to replace the current WWW by the semantic web. In fact the semantic web is another layer of the web that it can be considered as a WWW schema. It gives the explanations of the current web contents for the computer systems. A certain level of actions in brain can be implemented by this theory. Another point is the semantic web brings the information revolution that the merged information can show different information. The important point for the RDF technique is to treat every object as a resource.

During a query in the semantic web, as mentioned before, the SPARQL is a tool for queries in the RDF files in which the resources are described. Instead of searching in the web, the RDF query brings faster response. The more important is the RDF result is still an RDF which can be used in another query or data aggregation. It is hard to merge the different information in the WWW but in the RDF files the resources has the same types and can be easily merged.

The data is stored all over the network and the RDF can also be distributed anywhere. That means the entire internet is a huge database for RDF and it is doable to skip the search engine and get to the destination of the data in a query.

By testing an actual implementation such as the FOAF project and a small example for the business model, the readers will understand. In the following chapter there are two implementations showing the RDF working method and the usage of the RDF. The simulation of the semantic web environment shows how the system search in the database using the SPARQL and the how the system distinguish the resources for different purpose. The system should be able to distinguish the different objects (resources) even they are in the same resource name so that the system is able to continue the following process.

The FOAF project in the next chapter shows the base RDF usage and this project can be considered as a connection tools for human resources or even a name card method which shows the profile to the others, in the mean time it is specially designed for the system to understand who one is and what the characteristic it has. This is an ideal methodology for an interactive semantic search engine.

Chapter 4
A Semantic Web Experiment: The FOAF Project

In this chapter, there are two simulation experiments implemented. Based on the two implementations there are some topics to discuss and the later chapters contents are based on the result of these two implementations.

4.1.Environment

The semantic web is an idea about the additional layer for the current web system. It is not designed for a browser to display the content of a web page, instead it is designed for a system to understand the content of a resource and how the resources can be found and used by the system. In the following experiment the FOAF project shows how a person can be found by the others using the FOAF file (which is an RDF file). The system reads the foaf file of a person and displays the content that the system understands. The simulated server is established on a Linux system with Apache 2.2.11 + MySQL 5.1.31 + PHP 5.2.8. For reading the RDF file the system uses the Redland RDF API for PHP interface. It is a client-server architecture which the client part can be any one of the current browser. The way that the system understands the contents is shown by displaying the contents on the web pages.

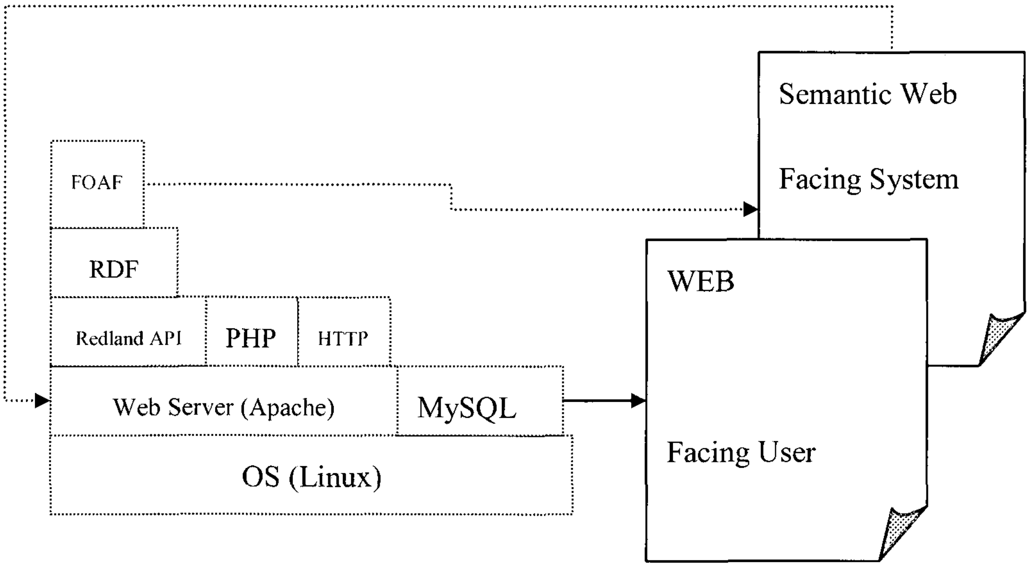


Figure 4.1-1 FOAF Project Architecture

It is not only the browser but also can be any developed client part. The server part can be any web server or even a new type server which offers only the resource lists. When there is a query for a resource the system returns an RDF file including the resource information as the theory shows. For client part the only requirement is to be able to read RDF file and explain the resource to the system.

After the FOAF project, there is a simulation for a search method which shows the working process of a semantic web. By using the FOAF project, the system returns the result of a query in an interactive way and distinguishes the resources for the system requirement. It is implemented under the same system environment as FOAF project.

The query is usually targeted in a limited database. A web search engine such as Google is using the distribution technology to store the information that it seeks from the Internet. The semantic web requires a huge database as well. The different is the semantic web database is a database storing the URL for resources which cost less than traditional data storage. A good classification methodology can solve the unlimited storage problem which considers all the nodes in the internet as a storage node for resources lists. This idea will be discussed in Chapter 6.5.

4.2.FOAF Project

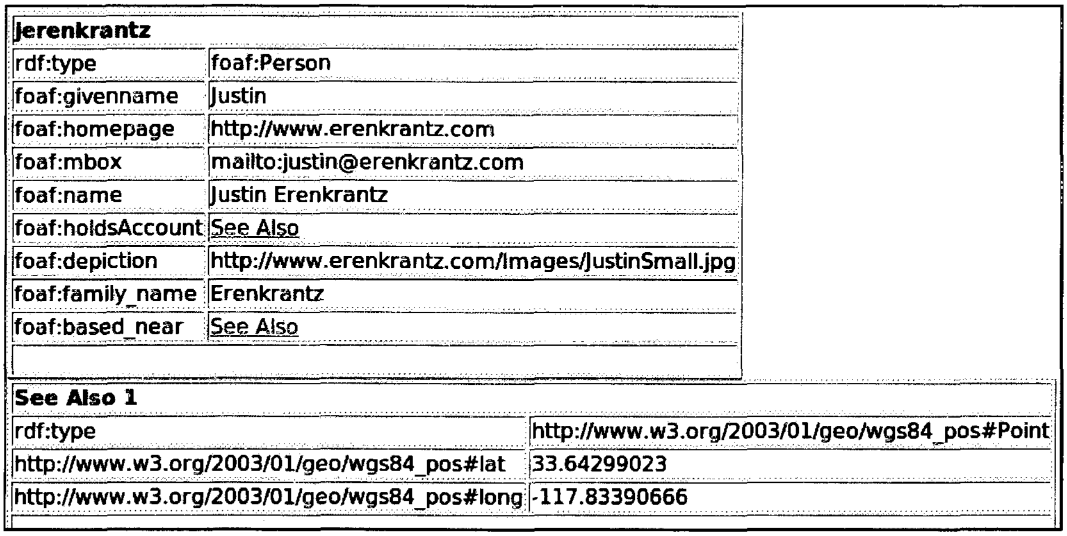
The FOAF project is one of the semantic web experiment project.

> FOAF is about your place in the Web, and the Web's place in our world. FOAF is a simple technology that makes it easier to share and use information about people and their activities (e.g. photos, calendars, web blogs), to transfer information between Web sites, and to automatically extend, merge and re-use it online. The *Friend of a Friend* (FOAF) project is creating a Web of machine-readable pages describing people, the links between them and the things they create and do. [FOAF][xiii]

Currently, there are many developers put their foaf file on their web pages. Some of them use the foaf file as a name card and show the others their connections and activities. For example, when we Google the result for 'foaf', there is a result from the first page: http://www.erenkrantz.com/jerenkrantz.rdf[*] which shows the foaf file of Justin R. Erenkrantz.

[*] This Web URL does not work any more because the author made the modification and the RDF was removed.

In the experiment part we implement a small program to show all the elements in the foaf file.



jerenkrantz	
rdf:type	foaf:Person
foaf:givenname	Justin
foaf:homepage	http://www.erenkrantz.com
foaf:mbox	mailto:justin@erenkrantz.com
foaf:name	Justin Erenkrantz
foaf:holdsAccount	See Also
foaf:depiction	http://www.erenkrantz.com/Images/JustinSmall.jpg
foaf:family_name	Erenkrantz
foaf:based_near	See Also

See Also 1	
rdf:type	http://www.w3.org/2003/01/geo/wgs84_pos#Point
http://www.w3.org/2003/01/geo/wgs84_pos#lat	33.64299023
http://www.w3.org/2003/01/geo/wgs84_pos#long	-117.83390666

Figure 4.2-1 FOAF display

4.3. Implementation for a Search Model

Within this section, there is another implementation shows how the system deal with the query. The system picks up the useful information by using the SPARQL in RDF queries.

4.3.1. Introduction for the Interactive Search Model

To obtain the advantage of semantic web and based on the last experiment, we suppose that one day the semantic web has a future standard which everybody uses. One should have at least one foaf file in RDF to show the other the basic information of oneself. In the foaf file there are names, address, age and the other information. Putting in more information will make it easier for the user to process. In the current situation, our searching methods are always based on the key words. There are many specific databases that face to the certain people. It does not mean anything for the system. The database gives out the right answer as long as the human being requests the proper key words. What will the database do if the results of a ping-pong game are being considered?

The foaf file can show the system what kind of professional person someone is and what his interests are. By knowing that the system understands whether he needs a general introduction or a professional knowledge collection. The query will proceed either in the professional database or the general database as requests.

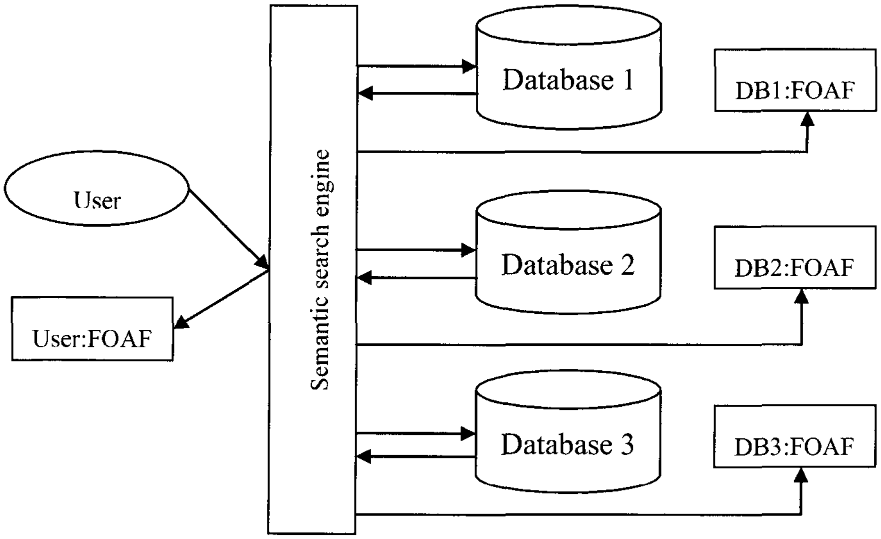


Figure 4.3-1 System Structure

4.3.2. Sample for the Model

Suppose there is one general database which has the content as the following table, and the user is someone who is interested in Music, Classical, Furniture, and Kitchen, following tables shows the content of the database:

Name	Content	property 1	property 2	property 3
Dining table	Dining table	Furniture	Kitchen	[empty]
Poker table	Poker table	Furniture	[empty]	[empty]
Soccer	Soccer	Sports	[empty]	[empty]
Tennis	Tennis	Sports	[empty]	[empty]
Table tennis	Table tennis	Sports	[empty]	[empty]
Swimming	Swimming	Sports	[empty]	[empty]
Enya	Enya	Music	NewAge	[empty]
Yanni	Yanni	Music	[empty]	[empty]
Tchaikovsky	Pyotr Llyich Tchaikovsky	Music	Classical	[empty]
Beethoven	Ludwig Van Beethoven	Music	Classical	[empty]
Sarah Brightman	Sarah Brightman	Music	Opera	Sarah Brightman
Andrew Lloyd Webber	Andrew Lloyd Webber	Opera	Sarah Brightman	[empty]
Scarborough Fair	Scarborough Fair	Sarah Brightman	[empty]	[empty]
Mozart	Wolfgang Amadeus Mozart	Music	Classical	[empty]
Karajan	Herbert Von Karajan	Music	Classical	[empty]
Secret Garden	Secret Garden	Music	NewAge	[empty]
Dvorak	Antonin Dvorak	Music	Classical	[empty]
Sibelius	Jean Sibelius	Music	Classical	[empty]
[empty]	[empty]	Folk	[empty]	[empty]

It is expressed in the foaf file as the following line:

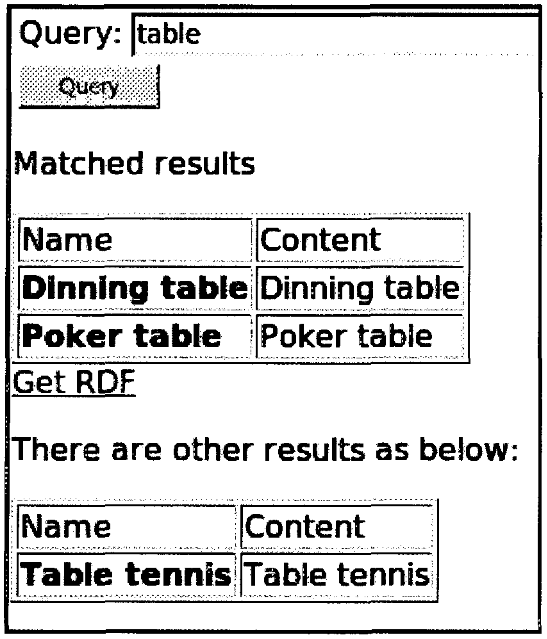
```
<foaf:interest>Music, Classical, Furniture, Kitchen</foaf:interest>
```

As a sample, the foaf is put under an executive position for the search engine in order to simulate the fetching action for the system. There exist the questions about the security reason which will be discussed in the following chapters.

4.3.3. Sample Model Simulation

As in the introduction part, when the system receives a query request, it is not going to perform the search action. Instead, it fetches the foaf automatically from the user side and read all the information that is useful for the searching action, such as the professional area, name, and age.

The most important one that is simulated in this sample is the <interest> part. From the foaf file the system understand that the user is interested in Music, Classical, Furniture, and Kitchen. The system does not understand the relationship among these classification elements until they are explained in another RDF schema file. After these, the system search from the database and get the result of the common query. Actually we ignore the process of getting the foaf file from the database side because it is just the opposite direction of the application method. It is supposed that the only database is the target database from the feedback of fetching the server foaf files. The result of the query will be filtered by the user foaf interest tag content. The output result contains the final result and optional result



Query: table

Matched results

Name	Content
Dinning table	Dinning table
Poker table	Poker table

Get RDF

There are other results as below:

Name	Content
Table tennis	Table tennis

Figure 4.3-2 Query Interface

For example, when query about 'table', there are three results from SQL: "Dinning table, Poker table, and Table tennis." Among the result the table tennis is not related to the interest content of the user. So it is not listed in the main result part, instead, the system put it in the optional part, which can avoid missing result from the similar searching. Also the system offers the RDF version of the description for the result to the user in order to make a further query.

4.3.4. Ambiguity of the Search Result

From the last example there is a question about the ambiguity. The 'table' is not only furniture or kitchen stuff. Also it is a name for a sport. Only from the key words the system is not able to tell the difference among these elements.

By using XML, it is also do-able to put a tag for each of the elements in order to 'tell' the system that it is a 'sport' so that the system will remember it with knowing what it really is. When people talk about the whole semantic web, it is impossible to create a general knowledge base for every single element in the reality because that is an unlimited work. The tags have memory capabilities, but are not readable. They are also difficult to comprehend. <Sports> and <Sport> means to entirely different things.

RDF file is not telling the system what a 'sport' is. RDF indicates the URIs in the node to tell the system what a 'sport' is explained and shown. That is how the system understands it. Spelling different would cause the system understand the similarly because from the network there are more than one resource which concerns the target.

4.4. Summary

The first experiment showed how the FOAF file is read by the system and how the system understands it. The second program showed how the system makes the semantic search in an interactive way and how to avoid the ambiguity problems. By using these two tools, it is visible to establish a formal project for real life.

The foaf project gives an interface for the semantic systems to exchange the basic information for person even things. The query function is the foundation of the search operation, which is always the first step to get into the internet. The new layer of the network gives the system an ability to understand what the people are or what the things are. FOAF gives the name card to the systems and all the information can be treated by the similar way for beginning.

The search model shows the intelligent network search by 'understanding' the meaning of the key words. Actually it is not only limited in a key words, but also can be a new style of searching operation. For example, the user has a part of the characters of the target; the foaf file can be made for this search and the target might be existed in several servers, the agent would be able to find out the proper ones among the huge amount of result by interactive searching and avoid the ambiguity of the search result by understanding what the target really is.

In the next chapter it introduces a complete business model. It is limited for the implementation because the semantic web standard is not established completely yet. Through the business model, it is easy to tell the advantage of the semantic design for business, especially for the information collection and management work it shows its ability and efficiency.

Chapter 5
Semantic Web Experiment of a Business Model

The semantic web technique is widely adapted for many domains such as complex production process management, financial system management. In this chapter, there is a small implementation for a semantic web experiment of a business model. The model is based on a distributor business which is one of the simplest business models in North America.

Usually, the distributors purchase directly from the manufactures in order to reduce the purchasing price. It has to reduce as much cost as possible whatever the internal or external cost, and rise up the price for selling within the range that is acceptable for the clients.

The information system for a business is not only a system to publish the information for the clients, but also a system for it to develop itself. The business researches the data that they own and optimize the business as well.

Each department has its own mission to accomplish. In this thesis some technique related departments are chosen to demonstrate the advantage and future prospects of semantic application. Within the technique layer and based on the introductions above (about the business and semantic techniques), there are discussions about the application for a distributor in reality after the description of the business modeling by department.

A certain number of the departments are chosen in the business model to demonstrate the advantage and future prospects of semantic application. In techniques, the business model can be implemented in the real world. The semantic web standard is not deployed in the whole internet world. It can be done in the near future. The process might be like the HTTP protocol establishment. In the following chapters the general methodology of building a semantic web environment will be introduced in concepts level.

The modeling part is separated into two parts: internal parts and external parts, which is shown in the last paragraphs. During the discussion it is supposed that there exist a certain size of general web which contains at least the foaf files for some servers. Semantic web is deployed only when it is becoming a standard of the current web for the public; otherwise it is only the application within a certain range.

5.1.The Business Model

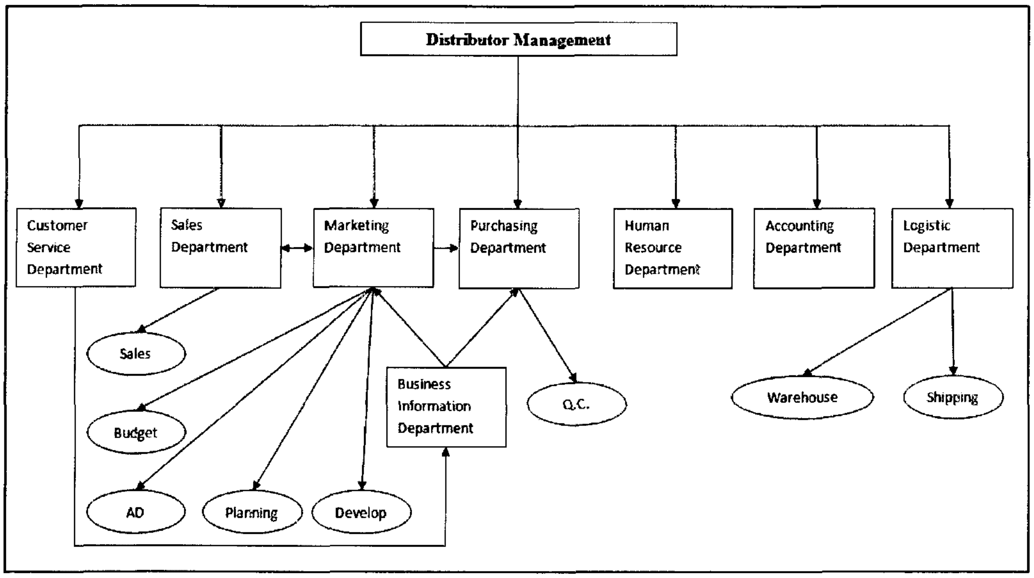


Figure 5.1-1 Business Model Structure

Distributor business is always about the sales. The sales department is the only one that makes benefit in the business. Obviously a good sales team is the key of success. The marketing department makes the marketing plans and the advertisement for sales. The development of new products is completed by the sales team. The business information department collects the information from the market and the customer service department and gives the feedback to the marketing department in order to better serve the sales team. Also the sales team provides feedback to the marketing department when they find any valuable information during the selling process. The marketing department will calculate the necessary information and forecast marketing and purchase plans when the purchasing department obtains the orders from the marketing department it should check the warehouse for the inventory and make a new purchasing order to the suppliers. Buyers should avoid risky decisions regarding shipping terms, quality assurance problems and they should be acutely aware for the social and environmental issues their actions contribute to.

5.2.Business Application Characteristic

This is an example of a business man facing an unfamiliar issue with the area of computer science: It is hard for a business man well understand how to merge the huge volume of the information from the internet. The companies hire the technicians to deal with the professional problems, to monitor the competitors` website, make the survey on the internet, and collect information of politic and economic trends. And there are the consulting companies who offer the precious business information in specific area and it is always expensive. These behaviors can be expensive. Also in the other part such as inventory statistic or sales statistic, the result is always with the inadvertently wrong data which is caused by the mistaken operation by the unprofessional people. These cause the waste of the human resources.

On the other hand, the information from a distributor is important to be shown to the retailers in order to gain more orders from them. The retailers should be encouraged by the information from the distributor that shows the products are competitive. It must be distinguished from the information for the suppliers, because the information of the distributor is confidential among the other business.

The current search engine shows the result that user requires. The user must select valuable information among millions of pieces. Some of the important information is 'hidden' from the combination of many pieces of news.

This is a story about a famous woman: "She ran her restaurants in China for many years and gains a certain amount of money. She was looking for a new project to proceed but she knows nothing others. During a trip in Sichuan Province at the end of 1980s, she saw some children playing Hula hoop. Those wooden inferior hoops made her curious and she asked about it to the children. They told her that they played it just for fun, but someone said this sport is effective for losing weight. She suddenly reminded in Shanghai there were many young girls searched for the drugs to losing weight. After the trip she made a business for the plastic Hula Hoop and sold them in Shanghai which was a very good investment at that time and it leaded a trendy of Hula Hoop in China. After a period of time she quit the market because she found that the plastic Hula Hoops are almost unbroken and the market was saturation of it in Shanghai."

In this case, the lady successfully combined the information such as Hula Hoop is a sport; it is useful for losing weight. Girls in Shanghai are willingly to lose weight. The result of the combination leads to a successful business case. In a certain domain, a certain area, with specific products, there are always many information to be mining. Since it is the information age now, there comes more information day by day. Why do we have the computer doing the simple job such as searching only and we spend our time and make the summary of all the information by ourselves? The system can do better.

This is another feature of the business application: In normal case, the company does not use more than one set of the software for the common staff. That is to say the rights control is also important. The sales team should be able to use the sales function but not allowed to touch anything in the accounting department. Also the logistic department will never reach the human resource department's functions. The management layer must be fully accessible for the system. In the common way people uses a username and a password to control but in the semantic system, the system judges the identification by treating everybody as a resource. The detail of this part will be discussed in the security consideration chapter.

5.3. Environment

The system is designed under Ubuntu linux with Redland API and PHP.

Before the description of the business model, there is an important part for the implementation: the DATABASE.

The database contains the data of the products, the human resource data, the client data, the inventory information and any other related data to the distributor. For the semantic system, the important point is to treat all the data as a resource, and this it could be indicated by the URI.

As there is always an existing database, it is not necessary to create another database to meet the 'resource requirement'. It is already there. When there is a query for data, the semantic search will be executed among the RDF files which have the resources location information. The RDF indicates the resource that not only the data in a database or a web site (service), but also it can be another RDF file which shows next level data. This is the similar to the process in of the DNS system which resolves the domain name during the network visiting. The connections of the RDF files are discussed in the next chapter.

Actually, the database (such as MySQL or Oracle) is not applied. Instead, the system use the RDF file as the database or resources for all the data transferring, because in the real network, the databases are different for types and connections. In semantic web layer, the data is treated as the concept resource. To be understood by the system, the data must be described by RDF, which means the source data and the result should be in RDF explanations. In this experiment, the system uses the RDF as the resource directly, which is one of the abilities of RDF. The RDF can be used directly as a data float, or can be another layer for the data just for the machine to understand.

Redland is a group of C libraries which support the semantic web concept. It contains the functions for RDF operation including the SPARQL as the query language for RDF.

The implementation is designed to show the abilities of the semantic web concept. The business model requires a stable system which is able to deal with the data collected from the systems all over the network. The system is combined by the following parts: Sales Department, Marketing Department, Product Information, Customer Information, Generate the purchase plan, Generate Business information and Generate business plan. It is going to show how the information will be collected from the sources as the resource, also how the system deal with the data to give out the information that human requires. The result from interactive query from tables shows the ability of understanding the data of the system.

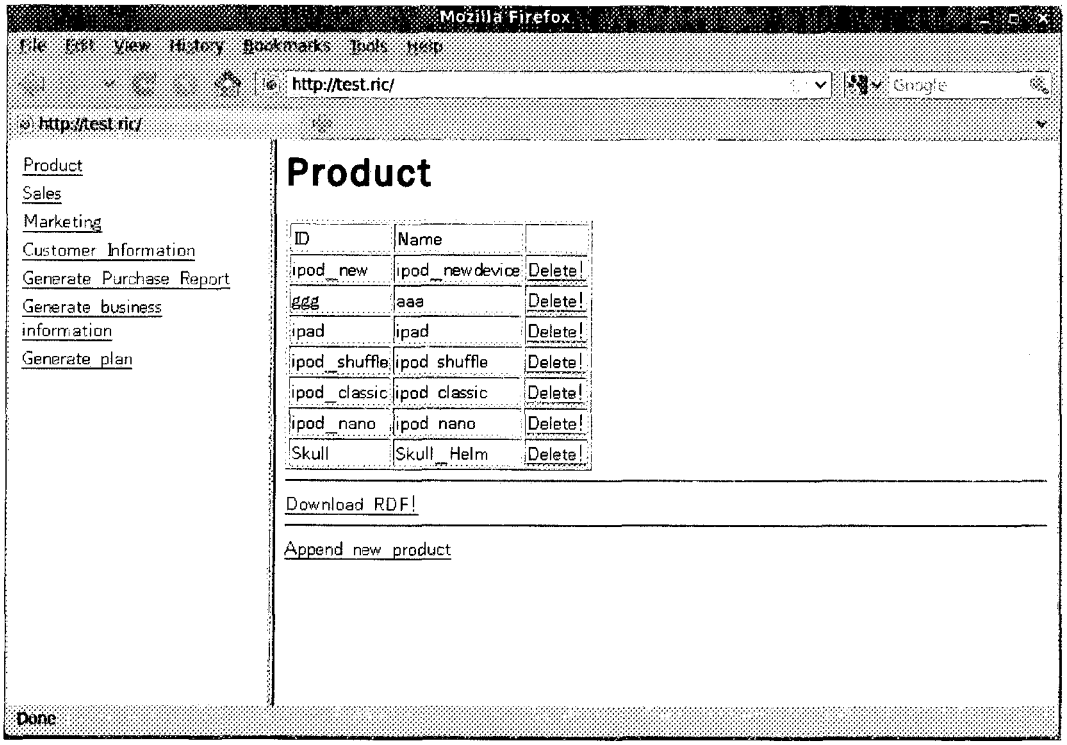
The system is not showing all the details of the distributor business because this is a report concerning information technology. In this part, it is going to introduce the technique detail of the business model and its implementation.

5.4.Modeling and the Implementation

5.4.1. Product

The products are registered in the system by a unique ID and product name for each. The user can choose delete operation or append a new record for the product. In order to avoid the same ID or same product name, the ID and the name will be erased at first and then add the new record every time. The different part from the common module is the output of the module is always in RDF format, which allows the system re-use the information. For example, when the inventory in the warehouse is running out, the customers' request or the analysis of the marketing trends requires the system fetch the data from the database to understand which and what the volume is for purchasing plan. Please note that, the data from the database, is not limited for a certain instance, the format is not limited, because the resource is described by RDF so the system understands what they are talking about. The system returns the valuable plan for the business without human being's intervention. In the mean time, the other modules of the application will be affected by the system actions, such as the customer service part will inform the customers that the goods they want are ordered as expected, or logistic department will get the information of shipping. The biggest advantage of the semantic system is that the system generates the information without the human being's intervention, and it could be re-used by system itself, or even the other semantic systems which can read RDF as the explanation for the data. A common scheme to create an abstract database from existing database, visible on a network through their RDF files.

The following screenshot shows the interface of the product part, this is the center of the data part. All the following data in other functions need this table (data) for further operation.



Product
Sales
Marketing
Customer Information
Generate Purchase Report
Generate business information
Generate plan

Product

ID	Name	
ipod_new	ipod_newdevice	Delete!
ggg	aaa	Delete!
ipad	ipad	Delete!
ipod_shuffle	ipod shuffle	Delete!
ipod_classic	ipod classic	Delete!
ipod_nano	ipod nano	Delete!
Skull	Skull_Helm	Delete!

Download RDF!

Append new product

Figure 5.4-1 Product

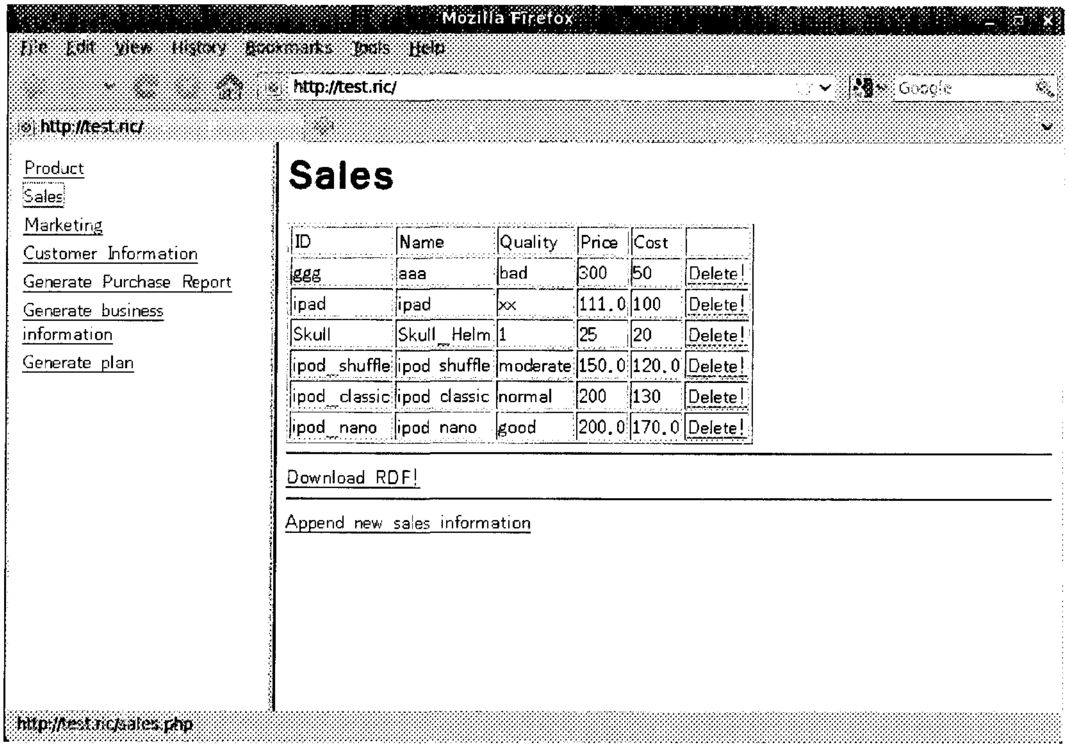
5.4.2. Sales

The sales department is the most important department in a distribution company because this is the only one who makes benefits.

The sales team should have the most accurate, timely and the widest possible information to master the marketing. Also they need a great marketing plan from the marketing department to have outstanding sales.

The information for the sales team begins by the information about the product. In the company's database, there must be the description about all the products that the company sells, including the picture, prices, specification, inventory, back order volume, feedback rate, origin mark, manufacture or the other sections. In order to simplify the model there are only a part of product information in this model to show the concept of the business model.

Sales part gives the user an interface of sales record. The sales team records the sales operation into the system. The administration system gets the information from sales department including the product id and name, quality of the product, the cost and the sales price. Also the sales team can delete or create record when needed. All the record can be exported as an RDF file. This operation is to demonstrate that the data can be reused for other semantic applications directly.



Sales

ID	Name	Quality	Price	Cost	
ggg	aaa	bad	300	50	Delete!
ipad	ipad	xx	111.0	100	Delete!
Skull	Skull_Helm	1	25	20	Delete!
ipod_shuffle	ipod shuffle	moderate	150.0	120.0	Delete!
ipod_classic	ipod classic	normal	200	130	Delete!
ipod_nano	ipod nano	good	200.0	170.0	Delete!

Download RDF!

Append new sales information

Figure 5.4-2 Sales

Every time when sales team changes the data in the sales function, the other function will fetch the new data in order to generate the latest result to guarantee the information is the opportune for business, which is very important for the business application.

The RDF code will be generated as the following:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<rdf:RDF xmlns:product="http://xmlns.com/product/0.1/" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" xml:base="http://xmlns.com/product/0.1/">
    <rdf:Description rdf:about="#ipad">
      <product:Price>111.0</product:Price>
    </rdf:Description>
    <rdf:Description rdf:about="#ipad">
      <product:Cost>100</product:Cost>
    </rdf:Description>
  ………………
```

5.4.3. Marketing

Every plan making is always based on the information that one has. It is the same story for the marketing department. The more information it owns, the more accurate strategy it makes. The information from sales department such as the feedback of the customers, the information from the business information department which may be the routine information collection, consulting company information, competitor monitoring information, and marketing trends can be all described as the resources and be put into the database of the business information. For example, when the salesman talked to the customer, the customer mentioned that he visited the farm in his area and because of the abnormal weather, the crops were terribly growing. This information is also confirmed by some other customers. The system recorded this information and checked up the related information:

```
  ……
  <rdf:Description
  rdf:about="http://www.targetarea.fake/crop/">
```

```
<crop:size>reduce 50%</crop:size>
</rdf:Description>
……
```

The information from the Geography database and the agriculture database shows that in this area, the alfalfa is the main product that supports the animal husbandry in the surrounding area:

```
……
<rdf:Description
rdf:about="http://www.agriculture.fake/targetarea/">
<targetarea:product>alfalfa</ targetarea:product>
</rdf:Description>
……
<rdf:Description
rdf:about="http://www.geography.fake/srarea/">
<srarea:product>Animal Husbandry</srarea:product>
</rdf:Description>
……
```

The market will have certain level of the reaction by the factor of the crops reducing and based on the information from several database, the system conclude the report as:

```
……
<rdf:Description
rdf:about="http://www.distributor.fake/alfalfa/">
<alfalfa:spot>Target Area</alfalfa:spot>
```

```
<alfalfa:price>Rising</alfalfa:price>

</rdf:Description>

……
```

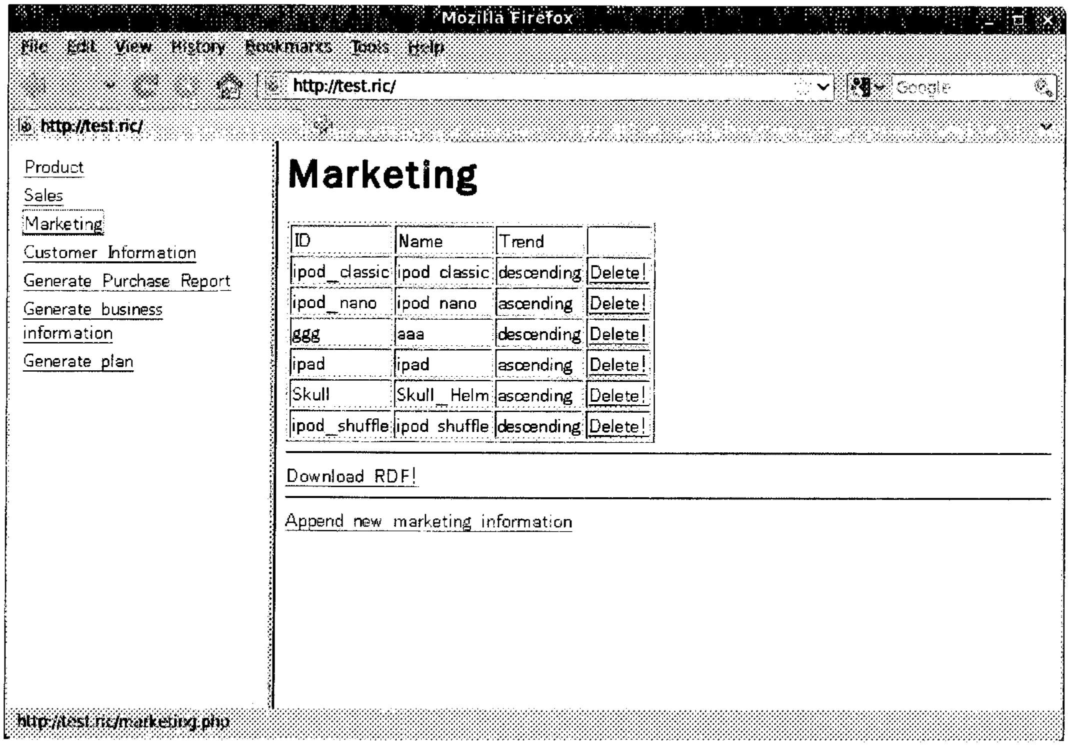
Usually, the market trends changes every minute, the system establish a base line, which can be an average level for the past 3 months, for example. The new data is compared with the base line to give a certain consultant advice, but for different area, different domain, the system uses different base line definition and different comparison methods. The database still stores the data for history queries. The RDF explains to the system how and where to generate the base line and other information for the comparison.

This is the information based on the feedback and the system analysis from the other database. The merging of the information is the ability of RDF and it is the property of the semantic system. The information is always treated as the process as above. The sales team will significantly improve the efficiency by the help of the hints from the system.

There are other responsibilities for the marketing department. The promotion plan should be released when there is a sales season coming or there is a business opportunity showing up. The system gives out the merged information from semantic searching and it should be helpful in the practical executions.

During the process of new product development, the semantic information about the similar products can be from the market even from some one who has nothing to do with this domain and just pop up a good idea and shared with his friends. The system collects the information and treats the data to make a report as the reference for the developer. And the developer can easily make the survey for new products concepts test by selecting the interested people from the system. Or deep dig more information such as whether the concept has already been applied and protected by the copyright protection policy.

In the implementation of the Marketing function, it is a simple part which shows the sales trends of the product in the market for simplification. Actually the real business model for marketing survey is more complex as discussed before. In this model, the marketing model is simplified to suit the technique topic. By RDF expression, it is not only able to show the numbers and the simple trends, but also fit for all kinds of data expression such as percentage of the variety, plans of improve, pictures or even audio tracks.



Marketing

ID	Name	Trend	
ipod_classic	ipod classic	descending	Delete!
ipod_nano	ipod nano	ascending	Delete!
ggg	aaa	descending	Delete!
ipad	ipad	ascending	Delete!
Skull	Skull_Helm	ascending	Delete!
ipod_shuffle	ipod shuffle	descending	Delete!

Download RDF!

Append new marketing information

Figure 5.4-3 Marketing

5.4.4. Customer Information

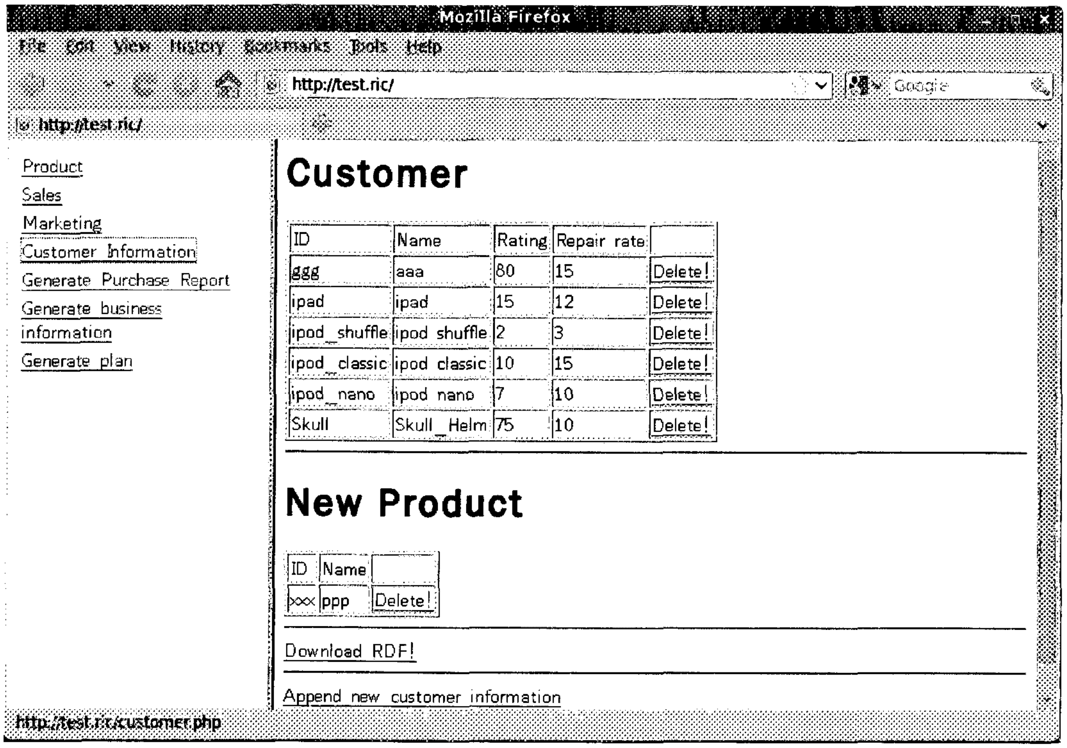
Customer information contains the survey result, which is an important part for marketing department. The feedback from the clients is the best instruction for business to optimize itself.

For marketing department, there is a hard job: make the annual plan for sales. To make a plan for a year requires a lot of information which the semantic system is good at it. There are many unpredictable conditions coming out every year and the marketing department must avoid as much as possible of it. When a low value goal is made, the distributor will lose benefit by lower purchasing plan and lower budget plan. On the contrary, an over estimated goal will dampen the sales team's enthusiasm and waste the resources of the budget.

The proper plan can be well made by the marketing department as long as the plan maker has enough information and the experiences. It is not going to be a perfect or accurate one but instead a reasonable and accountable plan is needed.

During the planning period, the information which includes the business records of last years, economic trends and environment, the primary materials market situation, the policy and politics changing, the human resource situation and any related information should be considered in. There is a huge volume of the information jump into the working process and it is not a simple work to merge everything in a brain of human being especially when the business size grows to a certain level.

The semantic system can put every detail in order and some of the data can be treated by existing economic theories which are to say that it can be solved in the semantic system. There are still the information that the system cannot gives out the quantitative criteria but merging the information will show something different and save the cost of the 'brain' in a human being's body. The power of the semantic system is the resources searching and information merging.



Product
Sales
Marketing
Customer Information
Generate Purchase Report
Generate business information
Generate plan

Customer

ID	Name	Rating	Repair rate	
ggg	aaa	80	15	Delete!
ipad	ipad	15	12	Delete!
ipod_shuffle	ipod shuffle	2	3	Delete!
ipod_classic	ipod classic	10	15	Delete!
ipod_nano	ipod nano	7	10	Delete!
Skull	Skull_Helm	75	10	Delete!

New Product

ID	Name	
xxx	ppp	Delete!

Download RDF!

Append new customer information

http://test.ric/customer.php

Figure 5.4-4 Customer

In this implementation, the interface shows the simplified model which has the rate of the product and the repair rate as the delegate. In the real model it can be extended as discussed before because the power of RDF for the resource concept.

Another point in this implementation, when there comes a new product, the marketing department will mark the New Product with the name and ID for further development.

5.4.5. Purchase Report

The purchasing department receives the order from the marketing department. When there is an order passed, the purchasing department chooses the suppliers from the system by recognizing the product classification, quality standard, production terms, payment methods and quality controlling. It can be implemented by the foaf-like project. Following is the simulation of the purchasing process in complete business model:

After choosing the suppliers, the purchasing department passes the order to the suppliers and starts the production. The quality control can be followed by the log file from the manufacture. The machines can be monitored by the system in the manufacture and they can choose a certain range of the data to be accessible to the distributor in order to make the quality control work. The product's document is a resource in the distributor's database, or it can be described in the manufacture's database with the standard specifications. Everyday's work is recorded into the database and during the production period the system from the manufacture side gives out the daily report automatically to the distributor to show the quality and the quantity of the products.

When there is a problem, the distributor can find it directly and fast. The early recognition of the problem, the easier the job is. The requirement of the product might be changing during the production period by the client or standard changing, the client gives the different requirement and the distributor can response with the information from this real time production management of quality control system to tell clearly whether it can be fixed or changed and the manufacture can receive the requirement at the same time to do the modification.

After the items are shipped, and a quality assurance problem is identified, the semantic system can pinpoint the issue. It becomes easier and clearly for both sides.

For example when an order of 500 tables is requested to a manufacture in China, during the production time the system shows the status as the following:

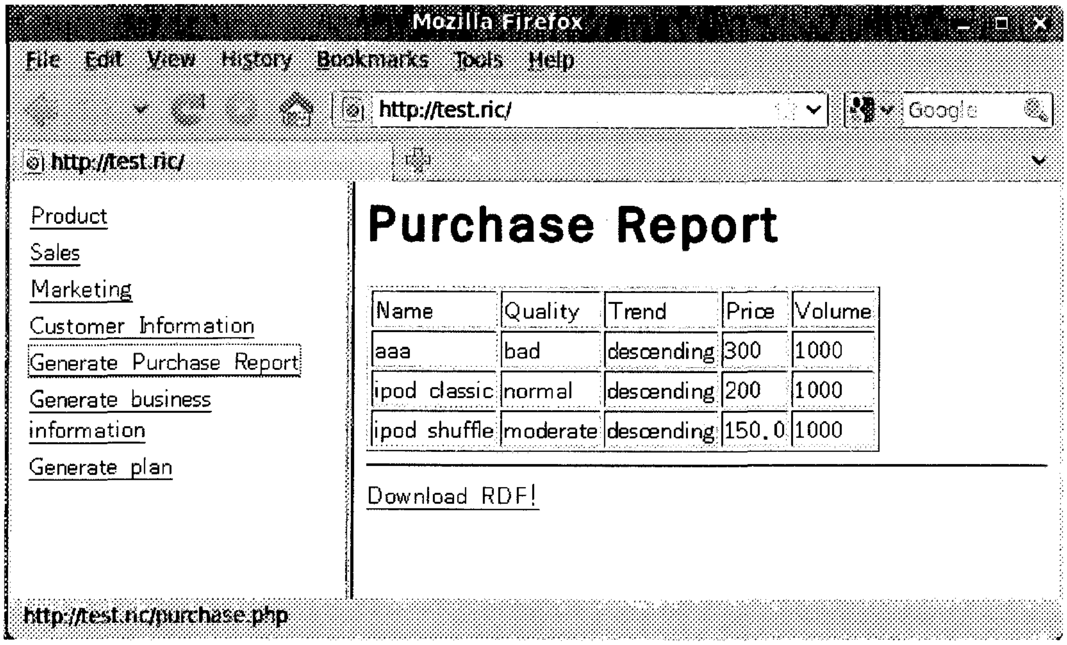
```
<?xml version="1.0"?>

<rdf:RDF

xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

xmlns:pr="http://www.tablemaker.fake/product#">

<rdf:Description

rdf:about="http://www.tablemaker.fake/product/table">
```

```
  <pr:name>Queen Table</pr:name>
<pr:quantity>500</pr:quantity>
<pr:color>brown</pr:color>
<pr:finished>150</pr:finished>
<pr:color-changeable> 200 </pr:color-changeable>
......
</rdf:Description>
</rdf:RDF>
```

The customer requests the color changing to the complex finishing shown in his web site:

The semantic system merges the two pieces of information and gives out the answer immediately saying that only 200 of them can be changed, otherwise it takes more time and more cost. During this process, the distributor does not have to confirm each of the details and the manufacture can receive the new commands in time. Besides, the distributor does not have to show the customer the details of the manufacture in order to protect the business secret which is very important in a distributor business, too.

```
......
<rdf:Description
rdf:about="http://www.customer.fake/Queentable/">
<pr:newcolor rdf:resource= “http://www.customer.fake/kingtable/color” / >
</rdf:Description>
......
```



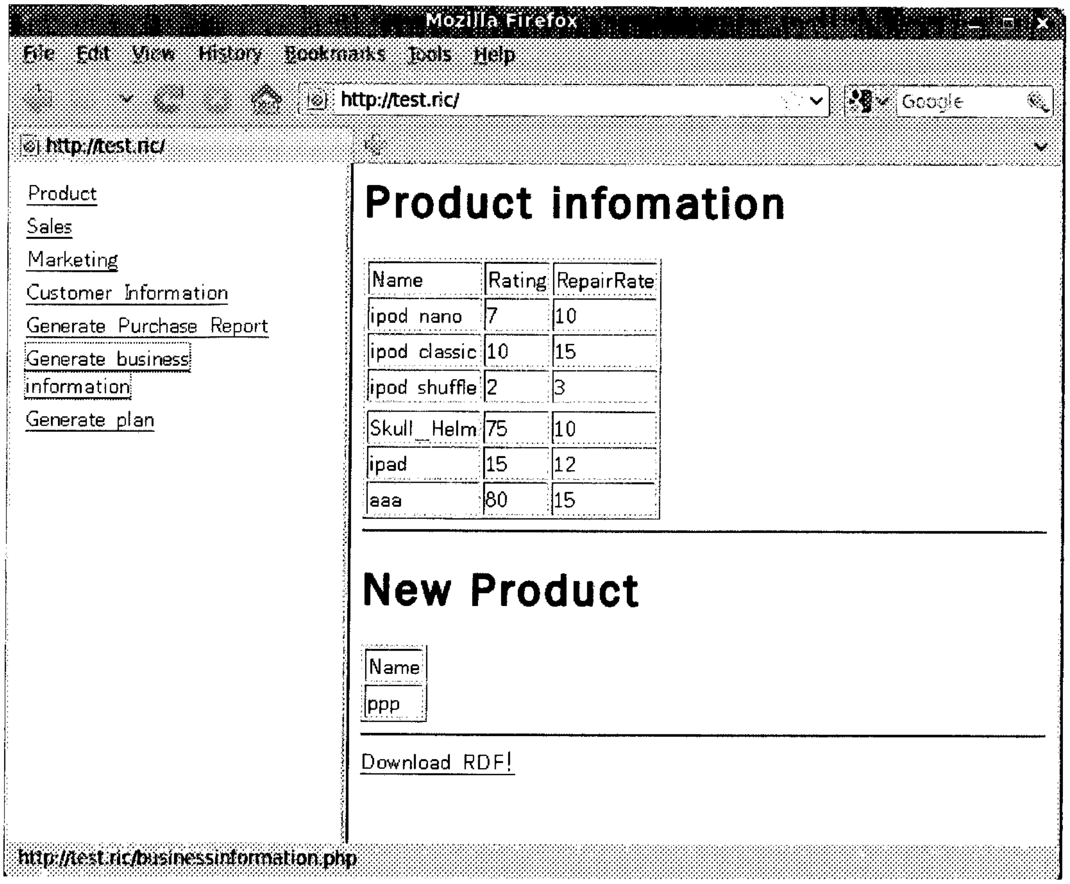
Purchase Report

Name	Quality	Trend	Price	Volume
aaa	bad	descending	300	1000
ipod classic	normal	descending	200	1000
ipod shuffle	moderate	descending	150.0	1000

Figure 5.4-5 Purchase Report

In the implementation, the purchase report is simplified, too. The system generates the purchase report when there is a product with ID and name, price, and marketing trend as 'descending'. The complex designed model which was discussed before has a link to the external elements of the business model (the supplier) and there are many other options for completing the information structure. According to the goal of this thesis, the model is always simplified till it is able to show the concept and the idea about the semantic web.

Business Information page is the feedback page for publication. It offers a place to show the clients what the business has and what is under developing. This is a part of the composition of the Marketing Department.



Product infomation

Name	Rating	RepairRate
ipod nano	7	10
ipod classic	10	15
ipod shuffle	2	3
Skull_Helm	75	10
ipad	15	12
aaa	80	15

New Product

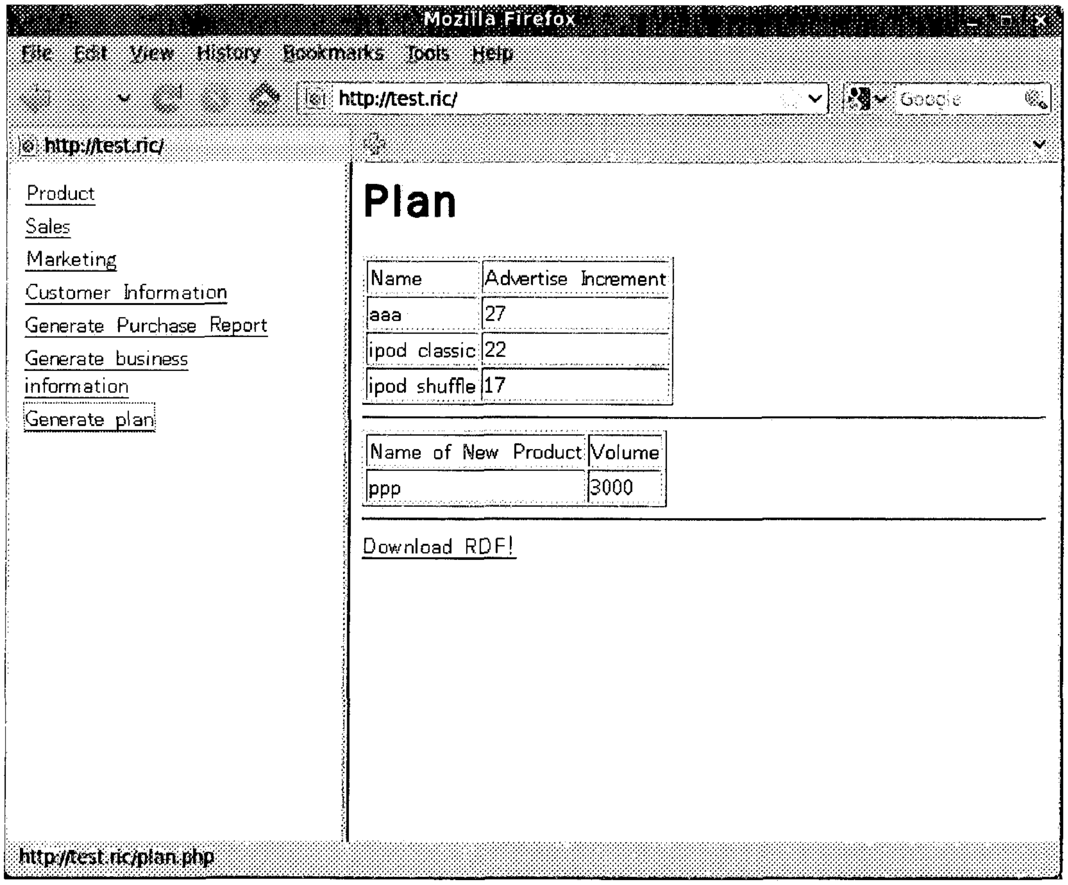
Name
ppp

Download RDF!

Figure 5.4-6 Product Publication

5.4.6. Generate a Business Plan

The development of a business plan is essential for any business. After the Sales Department and the Marketing Department submit the key data, the system is going to generate a business plan for the advertisement budget increasing for the next year, the plan includes the develop plan for the new products:



Plan

Name	Advertise Increment
aaa	27
ipod classic	22
ipod shuffle	17

Name of New Product	Volume
ppp	3000

Figure 5.4-7 Business Plan

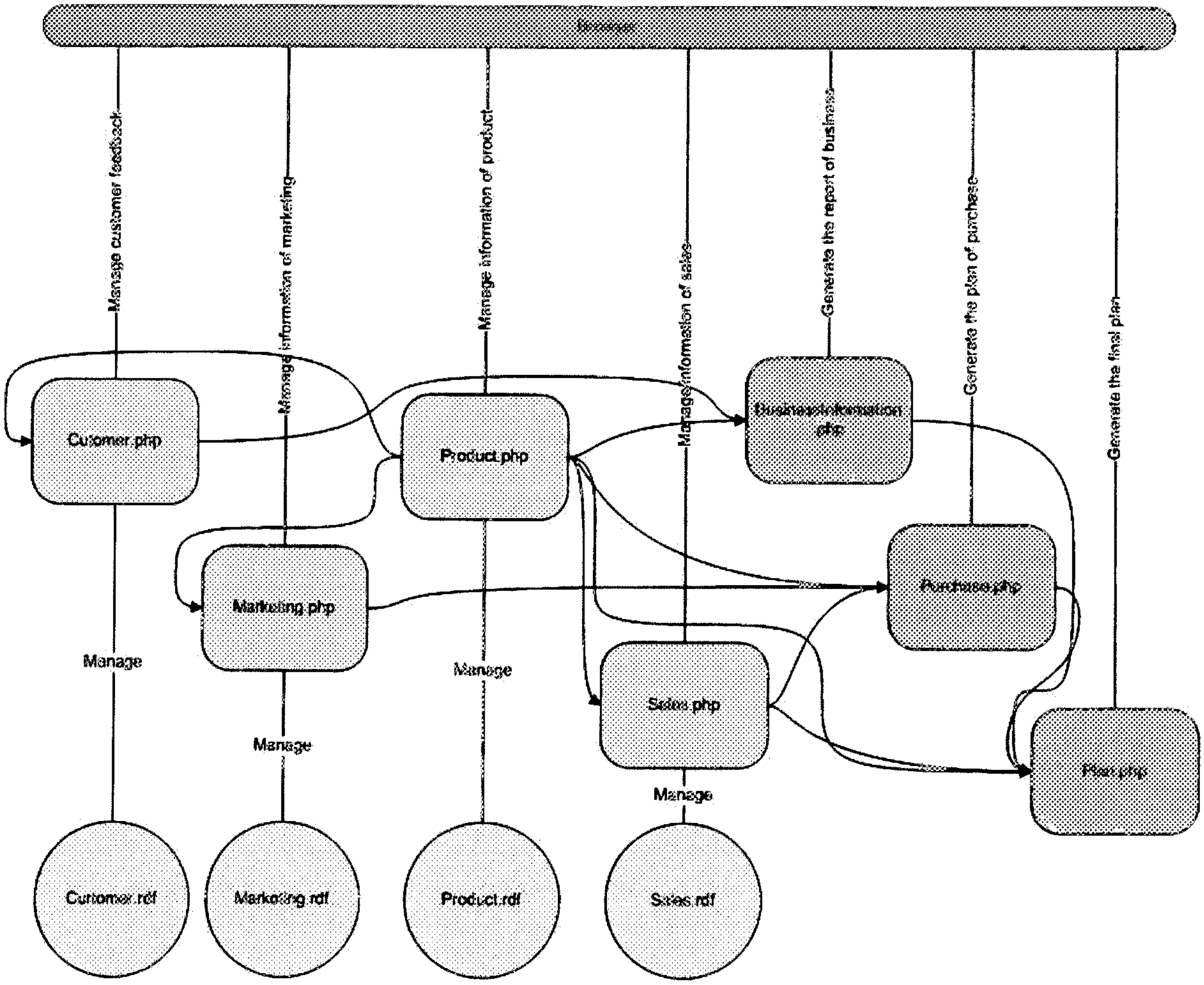
Only the products with enough information such as ID and names, prices, sales trends as 'descending' will be put into this report. Also the system gives out the volume of the new product expected.

This function shows a concept, 'using the resource'. It is a similar process as the query in a database among more than one table. The difference is that the resources are not limited in strict formats and the data is understandable for computer systems. The data can be any type such as numbers, images, audio tracks, even biological mark or magnetic signals. The RDF gives a method for the systems to read the data. The explanation of the data does not necessarily have the same meaning to the human brain, when the data is needed, the computer system will find its own method for the very data by RDF explanation. Also the RDF offers the logical method for relations and calculation which is the direction of the human brain's 'thinking'.

In the model the data is always number or text, and this is limited because of the simplification.

5.4.7 Structure of the application

The whole application structure can be explained in the following graphic:



The source codes can be found in Appendix B.

5.5.Methodology of Building Semantic Web

The business model uses the file as the resource in order to show the structure of the semantic web.

First of all, establish the resource concept of the system. For example, the business model requires the sales record as a resource, the data from the record is fetched by the module of marketing analysis. The module calculates the trends, the prices changing, then, give the output to the user. This is implemented as the data operation in the traditional system. It requires the data must be specified for the module which is going to receive it. But when the resource part is changing, such as a new server, a different operation system, or another database to fetch the data, the module should adapt the new ones that usually require the modification of the system. The semantic web system gives the idea of 'resource' which is explained by RDF. It means that even the source of the 'resource' changed or subjoined, the system can still recognize the data and pick them up from the source. In the sales records part, the system gets the data of sales prices, and trends.

Second, that RDF indicates interpreter of the resources in the business model, but this interpreter is not facing to the user but the system. This is the function part of the semantic web system, which means a disjoint layer from the user interface (before the data meets the final display).

Finally, the semantic web system uses the RDF to explain the resource. Also the system gives out the data (in resource concept, too) to the other module. The output content keeps explained by RDF when the output is facing another module of semantic web system.

To establish a semantic web application for business model the development work is described as the following steps:

Step	Step name	List of activities
1	Resource Concept	Treat all the nodes as a resource, including data, servers/database
2	RDF Lib installation	Get the RDF lib for the code support from http://download.librdf.org/
3	Build RDF	Write, or generate the resource description in XML for each resource (or each kind of resource).
4	Connect to DB	In DB, use the RDF to indicate the URI for resources.
5	Draw Graphic	Use graphic to define the relation among the resources. Also define the connections for different data flow.
6	Query in RDF	Query by SPARQL in RDF to locate the target resource.
7	Output in RDF	Generate RDF file for system to do further operation.

Above formula shows the basic step to establish the semantic web. During the process of definition for the semantic graphic which shows the relationship among the resources, there should be a standard. The standard can be established within a certain range such as within an enterprise or an industrial. The global standard is preferred but it requires certified by W3C org. It is a complex step, because it must be accepted by all the users and the specific domain's requires. The standard establishment is far beyond this article's discussion. It may be discussed in a related project in the future.

5.6. Summary

In this chapter the implementation of the business model was introduced. From a complete application under the semantic web environment better showed the ability and the advantage of the semantic web. The business application usually has the different characteristic from the ones in industrial or other professional area. It requires more general adoption ability because the data from different aspects will be different. The system is not only dealing with the information specific designed for it. For example, in the business model the system might receive the information for competitors. The data can be numbers with images or videos. Before receiving the data, the system has no idea about what they are and how to deal with it. The advantage of the semantic web environment is the system can read from the data to understand what they are.

To establish the semantic web environment, it requires a global standard for semantic web, especially the RDF. The standard establishment needs a certain volume of users for it. The individual user submits all kinds of resources and its explanations in a format of RDF, such as XML format. Then the collection of the whole data collection may leads to a global collection of the resource standard. It is not obligation to strictly follow it but it can still give a pattern of resource concept. It helps a lot in the real application development. The resource concepts will be established naturally in this way with time.

After building up the resource concept, they following steps are easier. Besides the traditional application for users' interactive, some parts of the system will be considered to be more automatically. Some of the operation will be replaced by data fetching and computing in semantic web layer instead of input and output by human or some special designed modules. The semantic application will follow the resource concepts to fetch the useful information (data) from all over the network. The RDF will help the system to find and deal with the data such as URL locating, data classification, and data transformation.

The final step of the semantic web layer is to be sustainability for the data flow. It means that they result from a module which contains the data fetched or dealt with by semantic part, will be able to re-use by the other application or modules. This idea leads the semantic web system becoming more and more automatically.

The traditional web system returns the result for human. The sources of the data are very different, so that the data formats are very different from each other. Without a proper standard, the data cannot be universal readable for the system. RDF solves this problem by explaining the data format for the system, which make the returned value readable for system. This is the most important point for semantic web. The result can be re-used for other queries or tasks. The intelligent network can be based on this recursive process.

With better understanding the data it is the only option for a system to perform the intelligent operation and really ‘thinks’ about what it is going to do. Inevitably, the risks and new challenges exist.

Chapter 6

More Technical Aspects of the Semantic Web

In this chapter, there are the discussions on the technique layer about the security and the fake information, robust and fault-tolerant, artificial intelligence related topic, RDF merge, and how RDF connects with each other.

6.1. Security and the Fake Information

The semantic web technique is good at information searching and merging treatment. There are huge amounts of information but only a small amount one will access. The information that the system collected is the information within the authorization. In the other hand, the information of one's own should be protected, too. Especially for a business, the distributor does not want the clients contact the manufacture directly, otherwise either the distributor lose the customer because the customer purchase directly from the manufacture when he needs a big volume of product, or he gets the purchasing price and that is really powerful in negotiating the price with the distributor. There is the problem about the security consideration in the semantic business model.

6.1.1. The Authorization

In the last chapters, there is the implementation about the foaf project. By reading the foaf file, the semantic application can read the identity of a person. In the business application model, the system has to distinguish the user in order to give the authorization to release the information to the visitor. But the foaf is a free file that anyone can make it in any content. If a manufacturer wants to know about the sales price of the distributor to see whether the distributor is capable to accept a price rising up, he can make a fake foaf file in his client side and visit the resource in distributor's database and get the information he wants. It is the problem of distributor that the information should be well protected by some mechanism.

The digital certificate is a solution for the authorization problem. There are the certificate authorities who offer the service in digital certification release. Some of them are free and some of them are paid. Compare to the username and the password model, the digital certificate method is safer and harder to decrypt.

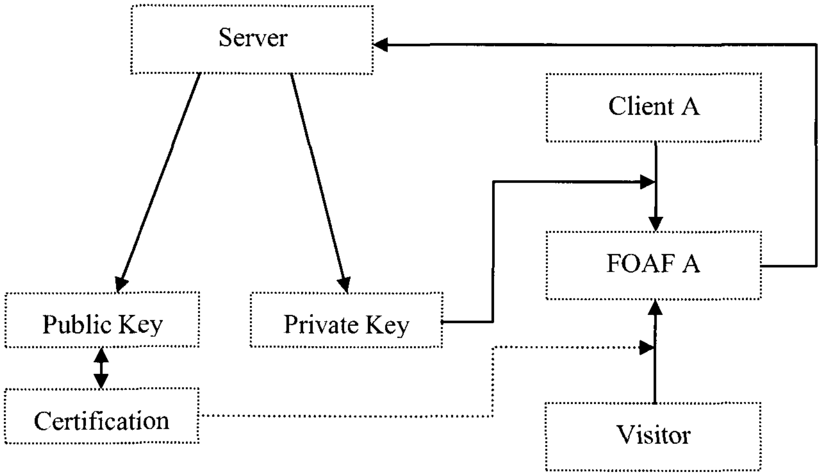


Figure 6.1-1 Certification Process

> The concept of the digital certificate is an electronic document which uses a digital signature to bind together a public key with an identity — information such as the name of a person or an organization, their address, and so forth. The certificate can be used to verify that a public key belongs to an individual.[PBKEY][xiv]

The working process is the server holds a public key which can be distributed in any position and the client keep a private key in secret. During the connection process, the client encrypts a piece of message by the private key in a certain algorithm such as RSA. The server side gives the decrypted message by the public key in the same algorithm. If the decrypted message meets the original message which was used for encryption in the client side, it means that the client side is holding a valid digital certificate so that the server part can treat the client as the one who is authorized to access the certain information in the server side. Even the original message and the public key is plaintext that everybody can read, without the private key it is impossible (depends on the algorithm) to reverse the process.

This theory is demonstrated in a lot of documents and there are many researches on it. In this thesis, the introduction of this theory is not going to take more length.

The little different part is the client can put the original message and the encrypted message in the foaf file so that the foaf is not only a simple name card but also a certificate proof for the identification of the visitor.

The same method can be used in the internal authorization control. As mentioned in the internal part of the business modeling chapter, the internal user rights control is also applicable for the authorization problem.

6.1.2. Access the Data behind RDF

The RDF is indicating the resources all over the network including the local database. The distributor's database contains the sensitive data such as the prices, information about the clients and the suppliers, information about the employees including the salary amount.

In the simple model the visitor makes the query by semantic system and the RDF will indicate the resource to the visitor. After that the visitor will access the data that he found directly. Actually the risk exists whenever there is a direct contact between the visitor and the data server which contains the sensitive data. A firewall is needed for the model.

In the following diagram it shows a structure of the firewall system for the semantic system. The dotted arrows show the original design for the data stream. Instead of that way, the database is separated from the outside connections but through a gateway. When the query to the RDF finished and the user wants to access the data, the request is redirected to the gateway. The system gives the feedback of the data request not directly back to the user but to a simulated database

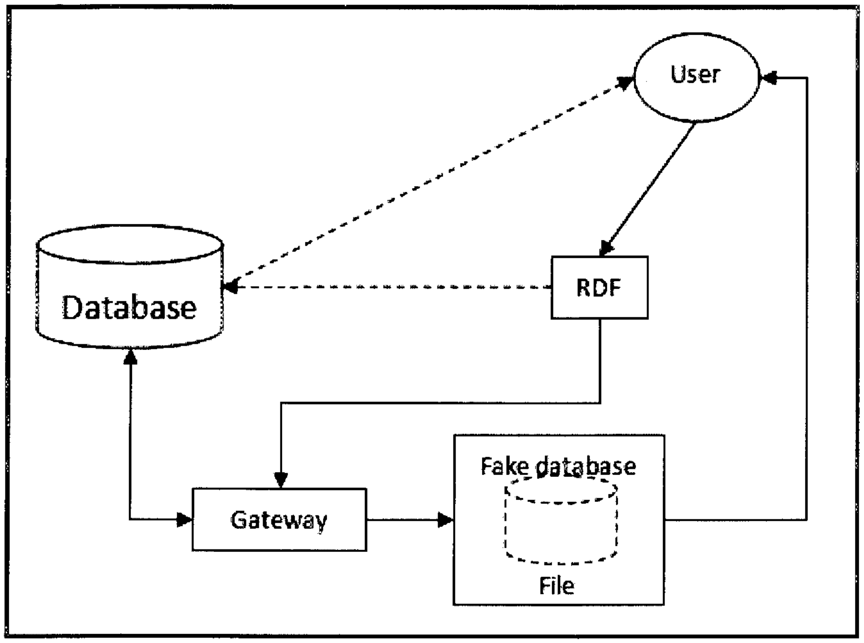


Figure 6.1-2 Security of Semantic Web

which is actually a database or just a file through the gateway.

After establishing the data in the fake database, the user can get the data that he asked for. From the user side it seems the same source because all he requests and the responses are from the gateway which can be disguised as the same URI. [WEBSEC][xv]

6.2. Robust and Fault-Tolerant

From the beginning of the software industrial, the fault-tolerant is the eternal topic. The semantic system is not an exception. The errors of the semantic system can be from the empty resource, wrong URI indicating, information mistaken merged or namespace confusion, the overload of system abilities, hardware or software conflicts.

For the business model, the most dangerous error is the loosing information especially the business information including the clients and products info. To prevent this kind of accident, the redundant backup is a proper solution. The technique is similar to the RAID techniques: backup the resources (data) in the database in another equivalent database. The indicator (RDF) points to the original data during the normal condition. When there is an error in the database, the system releases a re-direction layer which forced the requests to the redundant backup database to access the resource. The two databases should be checked regularly and maintenance in time. Concerning the security mentioned in the last chapter, this protection can be applied within the firewall and the user will not notice.

When there is a false RDF indicator applied, which means the resource that is indicated is not a valid resource, the system throws out an exception. The reason can be a no longer valid resource or mistaken maintenance RDF files. In a simple system throwing an exception will stop the working process and wait for the fixing. In the business model it must be corrected immediately otherwise the function part will delay the whole business process. For example, during the production period the quality controlling system is not able to contact with the real time report in the manufacture side will lead to a quality control. process failure. The distributor will treat the production when it is stopped or causes problems. Before it is fixed, the distributor will not be able to monitor the working process with the manufacture and not be able to give the response to the clients if there is an order changing. The exception treatment must be positive from the manufacture side. When the distributor system receives an exception which is released from the semantic system, it sent an exception query to the manufacture side and the exception treatment starts automatically. It can choose the backup system to send the original information or it must be fixed manually. This exception must be based on the well communicated condition. If the error is caused by the disconnection of the communication, the system can do nothing except send the request again after a certain delay. The exception must be obviously marked and warn the operator (manager, sales team, or purchase department) in time.

If the RDF is pointing to an empty resource, it can be treated as the HTTP exception (404), which shows an empty resource is returned. The operator should check whether the system is quoting the wrong resource or it is a wrong type of resource.

The property of the semantic system is able to merge the information from different resources. With a mistake in the knowledge structure or an incorrect combination the merged information will give out the wrong information. Sometimes it could be meaningless. The system will not be able to recognize the problem. When the case happened the developer must modify the system for the logical part or the knowledge base.

Usually the namespace confusion shows an error in resource indicator because in another namespace, the resource or the properties might be not existed at all. This error will throw a same exception as the empty resource indicating. The system throws out the exception with the information of the namespace so it will be more clearly for the developers to check out.

6.3. Artificial Intelligence and Semantic System

From many years ago people dreamed that one day the system will be able to understand the nature language and do everything fully automatic for human beings. This is the similar thinking about the Perpetual motion machine. But during the efforts of researching on the related projects, there are many advanced techniques born, such as self learning, logical inference (first-order), and knowledge base techniques in a certain domain. By researching, people get more and more techniques.

It is the same story for Semantic Web. The discussion on the Internet about the semantic web is always related to the artificial intelligence. Actually the semantic technique is designed for the machines instead of for the human beings, at least from the 'semantic' layer.

> The concept of machine-understandable documents does not imply some magical artificial intelligence which allows machines to comprehend human mumblings. It only indicates a machine's ability to solve a well-defined problem by performing well-defined operations on existing well-defined data. Instead of asking machines to understand people's language, it involves asking people to make the extra effort.
>
> Even though it simple to define, RDF at the level with the power of a semantic web will be complete language, capable of expressing paradox and tautology, and in which it will be possible to phrase questions whose answers would to a machine require a search of the entire web and an unimaginable amount of time to resolve. This should not deter us from making the language complete. Each mechanical RDF application will use a schema to restrict its use of RDF to a deliberately limited language. However, when links are made between the RDF webs, the result will be an expression of a huge amount of information. It is clear that because the Semantic Web must be able to include all kinds of data to represent the world, that the language itself must be completely expressive. [RDFNOT][xvi]

6.4.Merge the RDF

There are many times in this thesis mentioned about the 'merging the RDF'. Actually it is the aggregation of the data. The node of the RDF is the resources URI. There is not a problem of types for the data. The aggregation can be proceeding properly.

There is an example on a forum: From a Hong Kong Web site there is 'A.RDF':

http://foo.com/JackChen   ex:girlfriend   http://foo.com/QiliWu[†]

[†] This is a URI for an example of the resource expression, not a real link.

Also from a Malaysia web site there is another one 'B.RDF':

http://foo.com/QiliWu ex:hasChild 'Dragon girl'

RDF Aggregator proceeds by the URI http://foo.com/QiliWu and merges the two information and get the conclusion: Jack Chen's girlfriend has a child. If the Aggregator mining deeper:

?x ex:girlfriend ?y

and ?y ex:hasChild ?z

?x ex:hasChild ?z

There will be another conclusion: Jack Chen has a daughter! Actually, he does not.

Before the semantic web conceptions, the XML was designed for data exchanging among different platform or environment. The strictly data format guarantees the correctness of the data exchanging operation. In the semantic concept, the system treats every element as a resource so that the system can merge the information to make an 'inference' (which is not a real inference). It requires the aggregation of the data from different sources so it is hard to guarantee the strict format of the data. By algorithm thinking it is too difficult to combine the different types of the data. So the resource concept came out and solved the problem in a different layer, theoretically.

The RDF can be merged not because it has a loose structure of the data format. When we express the RDF by XML, it seems the similar syntax as the XML does. When the XML is hardly aggregated by the data from different sources, the RDF is not following the same way. The XML expression of RDF is only an optional format. There are other formats such as triples and graphs that describe the same thing. The aggregation of RDF is not formal aggregation of the data in the RDF file; it is the resource that the RDF indicates to be aggregated. For certain system, there certain knowledge structures and processes to do the job.

6.5.The Global View of the Semantic Web

A goal of the semantic web is that the entire web world becomes semantic. Before the ideal query process description, the knowledge base should be classified at least at root level. For example the resource of the knowledge will be distributed all over the network and the classification of the knowledge will be updated to the root server who holds a list of the root level of the knowledge classification, when the system recognized that the current knowledge is not classified yet. The next level servers hold the lists that contain the classification content which under this level. As the network grows and the semantic process proceeds, the multi-level lists will be built up by time. There should be an organization that takes the charge of distributing the lists to the distribution systems all over the network in order to keep it in order an efficient. This is a similar model as the DNS query process. The query will be returned layer by layer. The difference from the semantic web query process is the result is not unique and the return value is the RDF result which indicates the resources by URI in the network. The search engine is not necessary anymore because the entire network is a distribution search engine.

The classification of the knowledge is not strict restrictions for the format. From the research of library science, the knowledge classification has more than one method and the library classification science is a good reference for it. The knowledge classification is not only by the subjects but also from the culturally view, regional view, and sequential view. Also the resource of the knowledge is not restricted that it must belong to only one URI. The returning result will flood to the query user and the system will be self-restraint and filtered the result. The open model for the semantic searching over the network brings new concepts of the searching engine: the search is no longer for mining the data, instead, the data are 'pushed' to the target system and the system has to distinguish the useful data for itself. For a business model or any specific model the database is limited and the system can be maintained manually. When the global network becomes an entire search engine, the data is almost unlimited, the restriction of the data with the classification of the data is the problems for the next generation searching engine. The search centre should have a new task to classify the query for the user.

Also, the RDF technique is good at the data aggregation which means the merged information from difference resources. It usually brings the logical result or the inference result from the multi-data even RDF has nothing to do with inference but the system can be developed into a limited level inference engine. So the search query only by key words is not enough for a semantic search. The related result is also important for the data aggregation. The related data should be referred from the classification server which always understands the connections among the target data, and the connection among the knowledge can be managed from the child-node server to the parent-node server by updating the resource registration.

This is an example to show how the data is found and how the system manages the searching query:

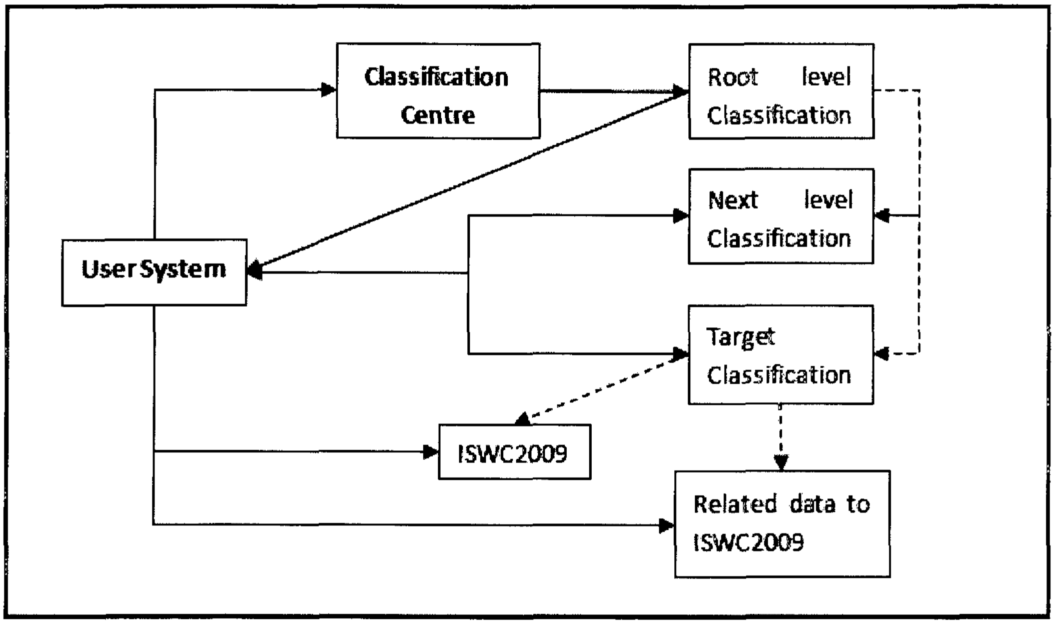


Figure 6.5-1 Multi-level RDF Data

When the user sends a query, for example search about the 8th International Semantic Web Conference 2009, the system submits a query to the classification center (equivalent as the search engine). The classification centre attached the tag for the root level knowledge classification, for example,

<META classification = "http://classification.root/Computer Science#">

The root level receives this request and reply to the user system to tell where to check of the next classification server. The next level classification server does the similar work and attaches the new tags until the query reaches the target classification server which indicates the resources by RDF. Also the target level classification server also gives out the related result which was updated by the administrator of ISWC2009 website. The user system receives the RDF response from the target classification server and the query process is done.[GRIDVINE][xvii]

Chapter 7
Related Work

In this chapter there is the introduction about the related work to the RDF techniques and the business models.

7.1. Semantic Inference Engine

In the previous chapters there were discussions about the data aggregation of the RDF. In fact the data aggregation gives out the merged information which is helpful for human being to understand well. The inference engine can lead further. There is an example from Youyong Zou, Tim Finin and Harry Chen in 2004 in their article:

> A simple example demonstrates the power of inference: Joe is visiting San Francisco and wants to find an Italian restaurant in his vicinity. His wireless PDA tries to satisfy his desire by searching for a thing of type *restaurant* with a *cuisineType* property with the value *Italian.* The goodPizza restaurant advertises its cuisine type as *Pizza.* These cannot be matched as keywords or even using a thesaurus, since *Italian* and *Pizza* are not equivalent in all contexts. The restaurant ontology makes things clearer: *Pizza rdfs:SubClassOf ItalianCuisine.* By using an inference engine, Joe's PDA can successfully determine that the restaurant goodPizza is what he is looking for. [FOWL][xviii]

This example well explained the function part for an inference engine. So far, the inference engine research work is still in a practical level. There are many projects is proceeding not only in the semantic domain but also the other areas.

7.2. Knowledge Base

> Machine-readable knowledge bases store knowledge in a computer-readable form, usually for the purpose of having automated deductive reasoning applied to them. They contain a set of data, often in the form of rules that describes the knowledge in a logically consistent manner. An ontology can define the structure of stored data - what types of entities are recorded and what their relationships are. Logical operators, such as And (conjunction), Or (disjunction), material implication and negation may be used to build it

> up from simpler pieces of information. Consequently, classical deduction can be used to reason about the knowledge in the knowledge base. Some machine-readable knowledge bases are used with artificial intelligence, for example as part of an expert system that focuses on a domain like prescription drugs or customs law. Such knowledge bases are also used by the semantic web. [MRKB][xix]

People did a lot of work in researching the knowledge base for general logic. The knowledge base technique is only applied in the specific domain such as industrial applications, some publication and copy rights controls and some research projects.

7.3. Existing Projects Related to Semantic Web

There are some immature applications developed by the concepts of semantic web. For example in the year 2008 in Xian Province, China, some researchers (Li Baomin, Zhang Na) release an Application of Semantic Intelligence Retrieval in Fruits Domain. They have the introduction in their same name article [ISSN : 1673-9965(2008)03-301-06] 2008, Volume 28, Edition 03.

This is still in the research phase. Some related techniques such as linked-data is developing at the same time which helps each other.

Chapter 8

Conclusion

The semantic web concept was first defined in the year 1998 [Tim Berners-Lee]. After 10 years developing, there are many new techniques joined and so far the semantic technique is growing. Even it is hard to say the semantic web is coming as the Web 3.0 standard, the idea and some of the models are applied in a certain level. The semantic web is still in his infancy, but ready for application.

At the beginning of this article, it mentioned the goal of the whole project. The concepts introduction leads the reader to a new layer of the web system which is the semantic layer for systems. By understanding the data the system makes intelligent move which people called web 3.0. The semantic web works in an equivalent layer as the current web but specific for system itself to understand. The key element for the working process is understanding of the data.

The implementation in this article shows that some of the techniques can be applied in some module such as interactive searching or rich-format automatic identification. When the technique is serving the common life, it means it is coming for perfect. The business model still contains some technique difficulties in the implementation reality but as the technique develops, it is just the problem of the time. The experiment is limited but it shows the basic idea about the semantic web. Even according to the current software environment it is impossible to implement the business model in real life, the concept and the methodology can be applied soon.

The semantic web is working at the back end. Whatever the system understands the data or not, what the user can see is still the content of the web pages. It is a framework or to say it is a foundation project. One of the difficulties is to establish the standard for semantic web and transfer the data/resource description into RDF/OWL. In the certain scale enterprise the huge amount of data has been indexed by the current database. It is not easy to show the advantage of the semantic web environment until the semantic applications become popular. People would ask: 'since the current database works fine, why would I spend a lot for something that I cannot see right away?'

Before answering this question, there is something should be recalled, the resources. The resources do not only mean the products in the market, the information about a physical item, more than that, and the applications (like web service), the next level resources indications. Compare to the resource within a single computer system, or a group of system in a limited network, when the whole internet is considered as a global database, the resource will be unlimited for individual. That is the final goal and the most distant advantage of the semantic web. To achieve this goal, there are many works to do such as semantic web standard establishment and semantic applications which are working on it, distribution system development, cloud computing, and security solutions.

The semantic web may surprise you someday.

Appendix

<Appendix A> the Page of the Web Link

http://www.erenkrantz.com/jerenkrantz.rdf (The first page only)[‡]

```
<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:foaf="http://xmlns.com/foaf/0.1/"
        xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
        xmlns:pm="http://www.web-semantics.org/ns/pm#"
        xmlns:wot="http://xmlns.com/wot/0.1/"
        xmlns:rss="http://purl.org/rss/1.0/"
        xmlns:dc="http://purl.org/dc/elements/1.1/"
        xmlns:doap="http://usefulinc.com/ns/doap#">
  <foaf:Person rdf:ID="jerenkrantz">
  <foaf:name>Justin Erenkrantz</foaf:name>
  <foaf:givenname>Justin</foaf:givenname>
  <foaf:family_name>Erenkrantz</foaf:family_name>
  <foaf:depiction rdf:resource="http://www.erenkrantz.com/Images/JustinSmall.jpg"/>
```

[‡] This Web URL does not work any more because the author made the modification and the RDF was removed.

<Appendix B> Source Code of the Business Model

In the chapter 5.4.7 (P64), there is a graphic showing the whole structure of the business model.

Common.inc

```
<?php
/*
 * common.inc - shared variables and functions
 *
 */
/* hostname of the website */
$webhost = "test.ric";
/* header of the generated RDF */
$header = "<?xml version=\"1.0\"?>\r\n" .
   "<rdf:RDF\r\n" .
   " xml:base=\"http://xmlns.com/product/0.1/\"\r\n" .
   " xmlns:rdf=\"http://www.w3.org/1999/02/22-rdf-syntax-ns#\"\r\n" .
   " xmlns:rdfs=\"http://www.w3.org/2000/01/rdf-schema#\"\r\n" .
   " xmlns:product=\"http://xmlns.com/product/0.1/\"\r\n" .
   ">\r\n";
/* footer of the generated RDF */
$footer = "</rdf:RDF>\r\n";
/* xmlbase of RDF */
$xmlbase = "http://xmlns.com/product/0.1/";
/* generate the prefix of SPARQL */
$prefix = "";
$arrPrefix = Array(
   'rdf' => "http://www.w3.org/1999/02/22-rdf-syntax-ns#",
   'rdfs' => "http://www.w3.org/2000/01/rdf-schema#",
  'product' => "http://xmlns.com/product/0.1/"
);
foreach ($arrPrefix as $key => $value) {
  $prefix .= "PREFIX $key: <$value>\n";
}
/**
 * rdf_new_world:
 *
 * Create an execution environment.
 *
 * Return value: a new world object or NULL on failure
 **/
function rdf_new_world() {
  return librdf_php_get_world();
}
/**
 * rdf_free_world:
 * @world: the world object
 *
 * Terminate the library and frees all allocated resources.
 *
 **/
function rdf_free_world($world) {
  librdf_free_world($world);
}
/**
 * rdf_new_file_storage:
 * @world: the world object
 * @path: the RDF file path
 *
 * Create a new FILE storage.
 *
 * Return value: a new storage object or NULL on failure
 **/
function rdf_new_file_storage($world, $path) {
   return librdf_new_storage($world, 'file', $path, NULL);
}
/**
 * rdf_new_storage:
 * @world: the world object
 *
 * Create a new MEMORY storage.
 *
 * Return value: a new MEMORY storage object or NULL on failure
 **/
function rdf_new_storage($world) {
   return librdf_new_storage($world, 'hashes', 'dummy', "new=yes,hash-type='memory'");
}
/**
 * rdf_free_storage:
 * @storage: the storage object
 *
 * Destroy the specified storage.
 *
 **/
function rdf_free_storage($storage) {
   librdf_free_storage($storage);
}
/**
 * rdf_new_model:
 * @world: the world object
 * @storage: the storage object
 *
 * Create a new model object.
 *
 * Return value: a new model object or NULL on failure
 **/
function rdf_new_model($world, $storage) {
  return librdf_new_model($world, $storage, '');
}
// destroy the specified model
/**
 * rdf_free_model:
 * @model: the model object
 *
 * Destroy the model object.
 *
 **/
function rdf_free_model($model) {
  librdf_free_model($model);
}
// fill `$model' with uris
/**
 * rdf_get_uri
 * @world: the world object
 * @model: the model object
 * @req_uri: the URI string
 *
 * Parse a URI of content into an model object
 *
 **/
function rdf_get_uri($world, $model, $req_uri) {
  global $xmlbase;
```

```
    $parser = librdf_new_parser($world, 'rdfxml', 'application/rdf+xml', null);
    $uri = librdf_new_uri($world, $req_uri);
    $baseuri = librdf_new_uri($world, $xmlbase);
    librdf_parser_parse_into_model($parser, $uri, $baseuri, $model);
    librdf_free_uri($baseuri);
    librdf_free_uri($uri);
    librdf_free_parser($parser);
}
/**
 * rdf_query_model:
 * @world: the world object
 * @model: the model object
 * @sql_string: the SPARQL string
 *
 * Execute a query against the model.
 *
 * Return value: an array containing the query results
 **/
function rdf_query_model($world, $model, $sql_string) {
    $query = librdf_new_query($world, 'sparql', null, $sql_string, null);
    $return = Array();
    $results = librdf_model_query_execute($model, $query);
    while ($results && !librdf_query_results_finished($results)) {
        $cell = Array();
        for ($i = 0; $i < librdf_query_results_get_bindings_count($results); $i++) {
            $name = librdf_query_results_get_binding_name($results, $i);
            $val = librdf_query_results_get_binding_value($results, $i);
            if($val) {
                $nval = librdf_node_to_string($val);
            } else {
                $nval = "";
            }

            $cell[$name] = $nval;

        }

        array_push($return, $cell);

        librdf_query_results_next($results);

    }

    librdf_free_query($query);


    return $return;

}

/**

 * strip_rdfid:

 * @input: the URI string

 *

 * Extract an identifier from the URI string.

 *

 * Return value: an identifier of a RDF entry

 **/

function strip_rdfid($input) {

    return preg_replace("/^.*#(.*)].*$/", "\\1", $input);

}

/**

 * to_rdfid:

 * @input: the identifier of a RDF entry

 *

 * Generate a URI string with the identifier.

 *

 * Return value: a URI string

 **/

function to_rdfid($input) {

    global $xmlbase;
    return $xmlbase . "#" . $input;
}
// get the abbreviate
/**
  * to_abbreviate:
  * @input: the full name of type
  *
  * Generate an abbreviation with the full name, or interrupt the program if a matching type doesn't exist.
  *
  * Return value: an abbreviation
  **/
function to_abbreviate($input) {
    global $arrPrefix;
    $input = trim($input, "[]");
    foreach ($arrPrefix as $key => $value) {
        if (strncmp($input, $value, strlen($value)) == 0) {
            return $key . ":" . substr($input, strlen($value));
        }
    }
    die("Don't know how to abbreviate `" . $input . "'!");
}
/**
  * serialize_model:
  * @world: the world object
  * @model: the model object to be serialized
  *
  * Serialize the model into a string.
  *
  * Return value: a string representing the model object
  **/
function serialize_model($world, $model) {
   global $arrPrefix, $xmlbase;

   $serializer = librdf_new_serializer($world, "rdfxml", null, null);

   $arrUri = array();

   foreach ($arrPrefix as $prefix => $uri) {

      $rdfuri = librdf_new_uri($world, $uri);

      librdf_serializer_set_namespace($serializer, $rdfuri, $prefix);

      array_push($arrUri, $rdfuri);

   }

   $base = librdf_new_uri($world, $xmlbase);

   $result = librdf_serializer_serialize_model_to_string($serializer, $base, $model);

   librdf_free_serializer($serializer);

   librdf_free_uri($base);

   foreach ($arrUri as $uri) {

      librdf_free_uri($uri);

   }

   return $result;

}

/**

 * add_new_type:

 * @world: the world object

 * @model: the model object

 * @id: the identifier of a RDF entry

 * @type: the type of a RDF entry

 *

 * Add a new type entry to the model.

 *

 **/

function add_new_type($world, $model, $id, $type) {

   global $arrPrefix, $xmlbase;

   $node_subject = librdf_new_node_from_uri_string($world, $xmlbase . "#" . $id);

   $node_predicate = librdf_new_node_from_uri_string($world, $arrPrefix['rdf'] . "type");
```

```
    $node_object = librdf_new_node_from_uri_string($world, $arrPrefix['product'] . $type);
    librdf_model_add($model, $node_subject, $node_predicate, $node_object);
    librdf_free_node($node_subject);
    librdf_free_node($node_predicate);
    librdf_free_node($node_object);
}
/**
  * add_new_value:
  * @world: the world object
  * @model: the model object
  * @id: the identifier of the RDF entry
  * @type: the type of the RDF entry
  *
  * Add a new value entry to the model.
  *
  **/
function add_new_value($world, $model, $id, $type, $value) {
   global $arrPrefix, $xmlbase;
   $node_subject = librdf_new_node_from_uri_string($world, $xmlbase . "#" . $id);
   $node_predicate = librdf_new_node_from_uri_string($world, $arrPrefix['product'] . $type);
   $node_object = librdf_new_node_from_literal($world, $value, NULL, 0);

   librdf_model_add($model, $node_subject, $node_predicate, $node_object);
   librdf_free_node($node_subject);
   librdf_free_node($node_predicate);
   librdf_free_node($node_object);
}
/**
  * delete_id_from_model:
  * @world: the world object
  * @model: the model object
```

```
 * @id: the identifier to be removed
 *
 * Delete the entry with the identifier `id'.
 *
 **/
function delete_id_from_model($world, $model, $id) {
  global $arrPrefix;
  $id = to_rdfid($id);
  $sql_string = $prefix .
    " SELECT * " .
    " WHERE { " .
    "   <$id> ?predicate ?object " .
    " } ";
  $result = rdf_query_model($world, $model, $sql_string);
  foreach ($result as $entry) {
    $predicate = trim($entry['predicate'], "[]");
    $object = trim($entry['object'], "[]");
    $node_subject = librdf_new_node_from_uri_string($world, $id);
    $node_predicate = librdf_new_node_from_uri_string($world, $predicate);
    if ($predicate == $arrPrefix['rdf'] . "type") {
      $node_object = librdf_new_node_from_uri_string($world, $object);
    } else {
      $node_object = librdf_new_node_from_literal($world, $entry['object'], NULL, 0);
    }
    $statement = librdf_new_statement_from_nodes($world, $node_subject, $node_predicate, $node_object);
    librdf_model_remove_statement($model, $statement);
    librdf_free_statement($statement);
    librdf_free_node($node_subject);
    librdf_free_node($node_predicate);
    librdf_free_node($node_object);
  }
}
/**
 * save_string_to_file:
 * @filepath: the path of the file to be saved
 * @input: the string to be saved
 *
 * Save the string to the specified file.
 *
 **/
function save_string_to_file($filepath, $input) {
   $fp = fopen($filepath, "wb");
   fwrite($fp, $input);
   fclose($fp);
}
/**
 * download_rdf:
 * @filepath: the path of the file to be downloaded
 *
 * Generate the downloading stream.
 *
 **/
function download_rdf($filepath) {
   $file = fopen($filepath, "rb");
   Header("Content-type: application/rdf+xml");
   Header("Accept-Length: " . filesize($filepath));
   echo fread($file, filesize($filepath));
   fclose($file);
}
/**
 * get_products:
 * @world: the world object
 *
 * Get the information of products.
 *
 * Return value: the information of products.
 **/
function get_products($world) {
   global $prefix;
   $product_uri = "http://test.ric/product.php?download";
   $storage = rdf_new_storage($world);
   $model = rdf_new_model($world, $storage);
   rdf_get_uri($world, $model, $product_uri);
   $sql_string = $prefix .
      " SELECT * " .
      " WHERE { " .
      "   ?id rdf:type product:Product; " .
      "         product:ProductName ?name " .
      " } ";
   $result = rdf_query_model($world, $model, $sql_string);


   rdf_free_model($model);
   rdf_free_storage($storage);
   return $result;
}
?>
```

Businessinformation.php

```
<?php
/*
 * businessformation.php - generate the report of business
 */
include_once("./common.inc");
/* definition of constants */
/* the uri for fetching the information of customers */
$customer_uri = "http://$webhost/customer.php?download";
/* the uri for fetching the information of products */
$product_uri = "http://$webhost/product.php?download";
/* initialization */
$world = rdf_new_world();
$storage = rdf_new_storage($world);
$model = rdf_new_model($world, $storage);
/* fill `$model' with the information of customer feedbacks and products */
rdf_get_uri($world, $model, $product_uri);
rdf_get_uri($world, $model, $customer_uri);
/* build the query string */
$sql_string = $prefix .
   " SELECT * " .
   " WHERE { " .
   "   ?id rdf:type ?type; " .
   "         product:ProductName ?name. " .
   "   OPTIONAL { ?id product:RepairRate ?repairrate } . " .
   "   OPTIONAL { ?id product:Rating ?rating } " .
   " } ";
/* execute the query */
$result = rdf_query_model($world, $model, $sql_string);
```

```
/* free useless resource */
rdf_free_model($model);
rdf_free_storage($storage);
/* generate the result according to the results */
if (isset($_GET["download"])) {
  /* generate RDF file */
  Header("Content-type: application/rdf+xml");
  echo $header;
  foreach ($result as $entry) {
    $id = strip_rdfid($entry["id"]);
    $type = to_abbreviate($entry["type"]);
    echo "<$type rdf:ID=\"$id\">\r\n";
    if ($type == "product:NewProduct") {
      echo "<product:ProductName>" . $entry[name] . "</product:ProductName>\r\n";
    }
    if (isset($entry["rating"]) && strlen($entry["rating"]) > 0) {
      echo " <product:Rating>" . $entry["rating"] . "</product:Rating>\r\n";
    }
    if (isset($entry["repairrate"]) && strlen($entry["repairrate"]) > 0) {
      echo " <product:RepairRate>" . $entry["repairrate"] . "</product:RepairRate>\r\n";
    }
    echo "</$type>\r\n";
  }
  echo $footer;
} else {
  /* generate the HTML page */
?>
<h1>Product infomation</h1>
<table border=1>
<tr>
```

```
<td>Name</td>
<td>Rating</td>
<td>RepairRate</td>
</tr>
<?
   foreach ($result as $entry) {
      $type = to_abbreviate($entry["type"]);
      echo "<tr>\r\n";
      if ($type == "product:NewProduct") {
         continue;
      }
      echo "<td>$entry[name]</td>\r\n";
      echo "<td>$entry[rating]</td>\r\n";
      echo "<td>$entry[repairrate]</td>\r\n";
      echo "</tr>\r\n";
   }
?>
</table>
<hr>
<h1>New Product</h1>
<table border=1>
<tr>
<td>Name</td>
</tr>
<?
   foreach ($result as $entry) {
      $type = to_abbreviate($entry["type"]);
      if ($type == "product:NewProduct") {
         echo "<tr>\r\n";
         echo "<td>$entry[name]</td>\r\n";
         echo "</tr>\r\n";
```

```
    }
  }
?>
</table>
<hr>
<a href="?download">Download RDF!</a>
<?
}
?>
```

Customer.php

```
<?php

/*

 * customer.php - view, append, remove or export the information of customer feedback

 */

include_once("./common.inc");

/*

 * definition of constants

 */

/* the uri for fetching the information of products */

$product_uri = "http://$webhost/product.php?download";

/* the data file for the information of customer feedbacks */

$customer_file = "./customer.rdf";

/* test whether to download the rdf file directly */

if (isset($_GET["download"])) {

   download_rdf($customer_file);

   exit;

}

/* initialization */

$world = rdf_new_world();

$storage = rdf_new_storage($world);

$model = rdf_new_model($world, $storage);

/* fill `$model' with the local data file */

rdf_get_uri($world, $model, "file:" . $customer_file);

/* process `delete' command and save the result to the data file */

if (isset($_GET["delete"])) {

   delete_id_from_model($world, $model, $_GET["delete"]);

   $result = serialize_model($world, $model);

   save_string_to_file($customer_file, $result);

}
```

```
/* modify the information of the specified customer feedback and save the result to the data file */
if (isset($_POST["type"]) && $_POST["type"] == "customer") {
  $id = $_POST["id"];
  $rating = $_POST["rating"];
  $repairrate = $_POST["repairrate"];
  delete_id_from_model($world, $model, $id);
  add_new_type($world, $model, $id, "Product");
  add_new_value($world, $model, $id, "Rating", $rating);
  add_new_value($world, $model, $id, "RepairRate", $repairrate);
  $result = serialize_model($world, $model);
  save_string_to_file($customer_file, $result);
}
/* append a new product to the data file */
if (isset($_POST["type"]) && $_POST["type"] == "newproduct") {
  $id = $_POST["id"];
  $name = $_POST["name"];
  delete_id_from_model($world, $model, $id);
  add_new_type($world, $model, $id, "NewProduct");
  add_new_value($world, $model, $id, "ProductName", $name);
  $result = serialize_model($world, $model);
  save_string_to_file($customer_file, $result);
}
/* fill `$model' with the information of products */
rdf_get_uri($world, $model, $product_uri);
/* build the query string of fetching the information of product and customer feedback */
$sql_string = $prefix .
  " SELECT * " .
  " WHERE { " .
  "    ?id rdf:type product:Product; " .
  "          product:ProductName ?name; " .
  "          product:Rating ?rating; " .
  "           product:RepairRate ?repairrate " .
  " } ";
/* execute the query */
$rating_result = rdf_query_model($world, $model, $sql_string);
/* build the query string of fetching the information of new products */
$sql_string = $prefix .
  " SELECT * " .
  " WHERE { " .
  "     ?id rdf:type product:NewProduct; " .
  "           product:ProductName ?name " .
  " } ";
/* execute the query */
$newproduct_result = rdf_query_model($world, $model, $sql_string);
/* free useless resource */
rdf_free_model($model);
rdf_free_storage($storage);
/* generate the page according to the results */
/* show all customer feedbacks and the list of new product(s) */
?>
<h1>Customer</h1>
<table border=1>
<tr>
<td>ID</td>
<td>Name</td>
<td>Rating</td>
<td>Repair rate</td>
<td>&nbsp;</td>
</tr>
<?
  foreach ($rating_result as $entry) {
    $id = strip_rdfid($entry["id"]);
```

```
        echo "<tr>\r\n";
        echo "<td>$id</td>\r\n";
        echo "<td>$entry[name]</td>\r\n";
        echo "<td>$entry[rating]</td>\r\n";
        echo "<td>$entry[repairrate]</td>\r\n";
        echo "<td><a href=\"?type=customer&delete=$id\">Delete!</td>\r\n";
        echo "</tr>\r\n";
    }
?>
</table>
<hr>
<h1>New Product</h1>
<table border=1>
<tr>
<td>ID</td>
<td>Name</td>
<td>&nbsp;</td>
</tr>
<?
    foreach ($newproduct_result as $entry) {
        $id = strip_rdfid($entry["id"]);
        echo "<tr>\r\n";
        echo "<td>$id</td>\r\n";
        echo "<td>$entry[name]</td>\r\n";
        echo "<td><a href=\"?type=newproduct&delete=$id\">Delete!</td>\r\n";
        echo "</tr>\r\n";
    }
?>
</table>
<hr>
<a href="?download">Download RDF!</a>
```

```
<?php
if (!isset($_GET["append"])) {
?>
<hr>
<a href="?append&type=customer">Append new customer information</a>
<br>
<a href="?append&type=newproduct">Append new product</a>
<?
} else {
/* show the section of appending new customer feedback or new product */
?>
<hr>
<form method=post action=customer.php>
<table border=1>
<?
  if ($_GET["type"] == "customer") {
    $result = get_products($world);
?>
<tr>
<td colspan=2>New customer information</td>
</tr>
<tr>
<td>ID</td>
<td>
<select name=id id=id>
<?
    foreach ($result as $entry) {
      $id = strip_rdfid($entry["id"]);
      echo "<option>$id</option>\r\n";
    }
?>
```

```
</select>
</td>
</tr>
<tr>
<td>Rating</td><td><input name=rating id=rating></td>
</tr>
<tr>
<td>Repair rate</td><td><input name=repairrate id=repairrate></td>
</tr>
<tr>
<td colspan=2>
<input type=hidden id=type name=type value=customer>
<input type=submit>
<input type=reset>
</td>
</tr>
<?
  } else if ($_GET["type"] == "newproduct") {
?>
<tr>
<td colspan=2>New Product</td>
</tr>
<tr>
<td>ID</td><td><input name=id id=id></td>
</tr>
<tr>
<td>Name</td><td><input name=name id=name></td>
</tr>
<tr>
<td colspan=2>
<input type=hidden id=type name=type value=newproduct>
```

```
<input type=submit>

<input type=reset>

</td>

</tr>

<?

  }

?>

</table>

</form>

<?

}

?>
```

Sales.php

```
<?php
  /*
  * sales.php - view, append, remove or export the information of sales
  */
include_once("./common.inc");
/*
  * definition of constants
  */
/* the uri for fetching the information of products */
$product_uri = "http://$webhost/product.php?download";
/* the data file for the information of sales */
$sales_file = "./sales.rdf";
/* test whether to download the rdf file directly */
if (isset($_GET["download"])) {
   download_rdf($sales_file);
   exit;
}
/* initialization */
$world = rdf_new_world();
$storage = rdf_new_storage($world);
$model = rdf_new_model($world, $storage);
/* fill `$model' with the local data file */
rdf_get_uri($world, $model, "file:" . $sales_file);
/* process `delete' command and save the result to the data file */
if (isset($_GET["delete"])) {
   delete_id_from_model($world, $model, $_GET["delete"]);
   $result = serialize_model($world, $model);
   save_string_to_file($sales_file, $result);
}
```

```
/* append the new information of sales */
if (isset($_POST["type"]) && $_POST["type"] == "newsales") {
  $id = $_POST["id"];
  $quality = $_POST["quality"];
  $price = $_POST["price"];
  $cost = $_POST["cost"];
  delete_id_from_model($world, $model, $id);
  add_new_type($world, $model, $id, "Product");
  add_new_value($world, $model, $id, "Quality", $quality);
  add_new_value($world, $model, $id, "Cost", $cost);
  add_new_value($world, $model, $id, "Price", $price);
  $result = serialize_model($world, $model);
  save_string_to_file($sales_file, $result);
}
/* fill `$model' with the information of products */
rdf_get_uri($world, $model, $product_uri);
/* build the query string of fetching the information of sales */
$sql_string = $prefix .
  " SELECT * " .
  " WHERE { " .
  "   ?id rdf:type product:Product; " .
  "         product:ProductName ?name; " .
  "         product:Quality ?quality; " .
  "         product:Cost ?cost; " .
  "         product:Price ?price " .
  " } ";
/* execute the query */
$result = rdf_query_model($world, $model, $sql_string);


/* free useless resource */
```

```
rdf_free_model($model);
rdf_free_storage($storage);
/* generate the page according to the results */
/* show all sales information */
?>
<h1>Sales</h1>
<table border=1>
<tr>
<td>ID</td>
<td>Name</td>
<td>Quality</td>
<td>Price</td>
<td>Cost</td>
<td>&nbsp;</td>
</tr>
<?
foreach ($result as $entry) {
   $id = strip_rdfid($entry["id"]);
   echo "<tr>\r\n";
   echo "<td>$id</td>\r\n";
   echo "<td>$entry[name]</td>\r\n";
   echo "<td>$entry[quality]</td>\r\n";
   echo "<td>$entry[price]</td>\r\n";
   echo "<td>$entry[cost]</td>\r\n";
   echo "<td><a href=\"?delete=$id\">Delete!</td>\r\n";
   echo "</tr>\r\n";
}
?>
</table>
<hr>
<a href="?download">Download RDF!</a>
```

```
<?php
if (!isset($_GET["append"])) {
?>
<hr>
<a href="?append">Append new sales information</a>
<? } else {
/* show the form of appending new information of sales */
  $result = get_products($world);
?>
<hr>
<form method=post action=sales.php>
<table border=1>
<tr>
<td colspan=2>New Sales Information</td>
</tr>
<tr>
<td>ID</td>
<td>
<select name=id id=id>
<?
  foreach ($result as $entry) {
    $id = strip_rdfid($entry["id"]);
    echo "<option>$id</option>\r\n";
  }
?>
</select>
</td>
</tr>
<tr>
<td>Quality</td><td><input name=quality id=quality></td>
</tr>
<tr>

<td>Price</td><td><input name=price id=price></td>

</tr>

<tr>

<td>Cost</td><td><input name=cost id=cost></td>

</tr>

<tr>

<td colspan=2>

<input type=hidden id=type name=type value=newsales>

<input type=submit>

<input type=reset>

</td>

</tr>

</table>

</form>

<?

}

?>
```

Purchase.php

```
<?php

/*

 * purchase.php - generate the plan of purchase

 */

include_once("./common.inc");

/*

 * definition of constants

 */

/* the uri for fetching the information of marketing */

$marketing_uri = "http://$webhost/marketing.php?download";

/* the uri for fetching the information of sales */

$sales_uri = "http://$webhost/sales.php?download";

/* the uri for fetching the information of products */

$product_uri = "http://$webhost/product.php?download";

/* initialization */

$world = rdf_new_world();

$storage = rdf_new_storage($world);

$model = rdf_new_model($world, $storage);

/* fill `$model' with the information of marketing */

rdf_get_uri($world, $model, $marketing_uri);

/* fill `$model' with the information of sales */

rdf_get_uri($world, $model, $sales_uri);

/* fill `$model' with the information of products */

rdf_get_uri($world, $model, $product_uri);

/* build the query string of fetching the information of purchase */

$sql_string = $prefix .

  " SELECT * " .

  " WHERE { " .

  "    ?id product:PriceTrend ?trend; " .
  "       product:Quality ?quality; " .
  "       product:ProductName ?name; " .
  "       product:Price ?price " .
  " } ";
/* execute the query */
$result = rdf_query_model($world, $model, $sql_string);
/* free useless resource */
rdf_free_model($model);
rdf_free_storage($storage);
/* generate the plan of purchase according to the fetched information */
if (isset($_GET["download"])) {
  Header("Content-type: application/rdf+xml");
  echo $header;
  foreach ($result as $entry) {
    $id = strip_rdfid($entry["id"]);
    if ($entry["trend"] == "ascending") {
      continue;
    }
    echo "<product:Product rdf:ID=\"$id\">\r\n";
    echo " <product:Volume>" . "1000" . "</product:Volume>\r\n";   // just purchase `1000' products
    echo "</product:Product>\r\n";
  }
  echo $footer;
} else {
?>
<h1>Purchase Report</h1>
<table border=1>
<tr>
<td>Name</td>
<td>Quality</td>
<td>Trend</td>
```

```
<td>Price</td>
<td>Volume</td>
</tr>
<?
  foreach ($result as $entry) {
    if ($entry["trend"] == "ascending") {
      continue;
    }
    echo "<tr>\r\n";
    echo "<td>$entry[name]</td>\r\n";
    echo "<td>$entry[quality]</td>\r\n";
    echo "<td>$entry[trend]</td>\r\n";
    echo "<td>$entry[price]</td>\r\n";
    echo "<td>1000</td>\r\n";     // just purchase `1000' products
    echo "</tr>\r\n";
  }
?>
</table>
<hr>
<a href="?download">Download RDF!</a>
<?
}


?>
```

Product.php

```
<?php
/*
 * product.php - view, append, remove or export the information of products
 */
include_once("./common.inc");
/* definition of constants */
/* the data file for the information of products */
$product_file = "./product.rdf";
/* test whether to download the rdf file directly */
if (isset($_GET["download"])) {
  download_rdf($product_file);
  exit;
}
/* initialization */
$world = rdf_new_world();
$storage = rdf_new_storage($world);
$model = rdf_new_model($world, $storage);
/* fill `$model' with the local data file */
rdf_get_uri($world, $model, "file:" . $product_file);
/* process `delete' command and save the result to the data file */
if (isset($_GET["delete"])) {
  delete_id_from_model($world, $model, $_GET["delete"]);
  $result = serialize_model($world, $model);
  save_string_to_file($product_file, $result);
}
/* append a new product to the data file */
if (isset($_POST["type"]) && $_POST["type"] == "newproduct") {
  $id = $_POST["id"];
  $name = $_POST["name"];
```

```
    delete_id_from_model($world, $model, $id);

    add_new_type($world, $model, $id, "Product");

    add_new_value($world, $model, $id, "ProductName", $name);

    $result = serialize_model($world, $model);

    save_string_to_file($product_file, $result);

}

/* build the query string of fetching the information of products */

$sql_string = $prefix .

    " SELECT ?id ?name " .

    " WHERE { " .

    "   ?id rdf:type product:Product; " .

    "          product:ProductName ?name " .

    " } ";

/* execute the query */

$result = rdf_query_model($world, $model, $sql_string);

/* free useless resource */

rdf_free_model($model);

rdf_free_storage($storage);

/* generate the page according to the results */

/* show the list of product(s) */

?>

<h1>Product</h1>

<table border=1>

<tr>

<td>ID</td>

<td>Name</td>

<td>&nbsp;</td>

</tr>

<?

   foreach ($result as $entry) {

      $id = strip_rdfid($entry["id"]);
```

```
        echo "<tr>\r\n";
        echo "<td>$id</td>\r\n";
        echo "<td>$entry[name]</td>\r\n";
        echo "<td><a href=\"?delete=$id\">Delete!</td>\r\n";
        echo "</tr>\r\n";
    }
?>
</table>
<hr>
<a href="?download">Download RDF!</a>
<?php
/* show the section of appending new product */
if (!isset($_GET["append"])) {
?>
<hr>
<a href="?append">Append new product</a>
<? } else {
?>
<hr>
<form method=post action=product.php>
<table border=1>
<tr>
<td colspan=2>New Product</td>
</tr>
<tr>
<td>ID</td><td><input name=id id=id></td>
</tr>
<tr>
<td>Name</td><td><input name=name id=name></td>
</tr>
<tr>
```

```
<td colspan=2>

<input type=hidden id=type name=type value=newproduct>

<input type=submit>

<input type=reset>

</td>

</tr>

</table>

</form>

<?

}

?>
```

Plan.php

```
<?php

/*

  * plan.php - generate the plan

  */

include_once("./common.inc");

/*

  * definition of constants

  */

/* the uri for fetching the information of business */

$businessinformation_uri = "http://$webhost/businessinformation.php?download";

/* the uri for fetching the information of sales */

$sales_uri = "http://$webhost/sales.php?download";

/* the uri for fetching the information of products */

$product_uri = "http://$webhost/product.php?download";

/* the uri for fetching the information of purchase */

$purchase_uri = "http://$webhost/purchase.php?download";

/* initialization */

$world = rdf_new_world();

$storage = rdf_new_storage($world);

$model = rdf_new_model($world, $storage);

/* fill `$model' with the information of products */

rdf_get_uri($world, $model, $product_uri);

/* fill `$model' with the information of purchase */

rdf_get_uri($world, $model, $purchase_uri);

/* fill `$model' with the information of business */

rdf_get_uri($world, $model, $businessinformation_uri);

/* fill `$model' with the information of sales */

rdf_get_uri($world, $model, $sales_uri);
```

```
/* build the query string of fetching the information to generate the plan */
$sql_string = $prefix .
  " SELECT * " .
  " WHERE { " .
  "   ?id product:ProductName ?name; " .
  "        product:Rating ?rating; " .
  "        product:RepairRate ?repairrate; " .
  "        product:Volume ?volume; " .
  "        product:Price ?price; " .
  "        product:Cost ?cost " .
  " } ";
/* execute the query */
$result = rdf_query_model($world, $model, $sql_string);
/* build the query string of fetching the information of new products */
$sql_string = $prefix .
  " SELECT * " .
  " WHERE { " .
  "   ?id rdf:type product:NewProduct . " .
  "   ?id product:ProductName ?name " .
  " } ";
/* execute the query */
$new_result = rdf_query_model($world, $model, $sql_string);
/* free useless resource */
rdf_free_model($model);
rdf_free_storage($storage);
/* generate the plan */
$eval = Array();
foreach ($result as $entry) {
  $id = strip_rdfid($entry["id"]);
  $name = $entry["name"];
  $price = intval($entry["price"]);
  $cost = intval($entry["cost"]);
  $repairrate = intval($entry["repairrate"]);
  $benefit = ($price - $cost) * 100 / $cost * (100 - $repairrate) / 100;
  array_push($eval, array(
    "id" => $id,
    "name" => $name,
    "benefit" => $benefit
  ));
}
function benefit_cmp($a, $b) {
  $va = $a["benefit"];
  $vb = $b["benefit"];
  if ($va == $vb) {
    return 0;
  }
  return ($va > $vb) ? -1 : 1;
}
usort($eval, "benefit_cmp");
/* generate the rdf according to the results */
if (isset($_GET["download"])) {
  Header("Content-type: application/rdf+xml");
  echo $header;
  $advertiserate = 27.12;
  foreach ($eval as $entry) {
    $id = $entry["id"];
    if ($advertiserate <= 0) {
      break;;
    }

    echo "<product:Product rdf:ID=\"$id\">\r\n";
    echo " <product:AdvertiseIncrement>" . intval($advertiserate) . "</product:AdvertiseIncrement>\r\n";
    echo "</product:Product>\r\n";
    $advertiserate = $advertiserate * 0.813;
  }
  $max_volume = 3000;
  foreach ($new_result as $entry) {
    $id = strip_rdfid($entry["id"]);
    echo "<product:NewProduct rdf:ID=\"$id\">\r\n";
    echo " <product:Volume>" . intval($max_volume) . "</product:Volume>\r\n";
    echo "</product:NewProduct>\r\n";
    $max_volume = $max_volume * 0.648;
  }
  echo $footer;
} else {
/* generate the page according to the results */
?>
<h1>Plan</h1>
<table border=1>
<tr>
<td>Name</td>
<td>Advertise Increment</td>
</tr>
<?
  $advertiserate = 27.12;
  foreach ($eval as $entry) {
    if ($advertiserate <= 0) {
      break;;
    }

    echo "<tr>\r\n";
    echo "<td>$entry[name]</td>\r\n";
    echo "<td>" . intval($advertiserate) . "</td>\r\n";
    echo "</tr>\r\n";
    $advertiserate = $advertiserate * 0.813;
  }
?>
</table>
<hr>
<table border=1>
<tr>
<td>Name of New Product</td>
<td>Volume</td>
</tr>
<?
  $max_volume = 3000;
  foreach ($new_result as $entry) {
    echo "<tr>\r\n";
    echo "<td>$entry[name]</td>\r\n";
    echo "<td>" . intval($max_volume) . "</td>\r\n";
    echo "</tr>\r\n";
    $max_volume = $max_volume * 0.648;
  }
?>
</table>
<hr>
<a href="?download">Download RDF!</a>
<?
}
?>
```

Marketing.php

```
<?php
/*
 * marketing.php - view, append, remove or export the information of marketing
 */
include_once("./common.inc");
/*
 * definition of constants
 */
 /* the uri for fetching the information of products */
$product_uri = "http://$webhost/product.php?download";
/* the data file for the information of marketing */
$marketing_file = "./marketing.rdf";
/* test whether to download the rdf file directly */
if (isset($_GET["download"])) {
  download_rdf($marketing_file);
  exit;
}
/* initialization */
$world = rdf_new_world();
$storage = rdf_new_storage($world);
$model = rdf_new_model($world, $storage);
/* fill `$model' with the local data file */
rdf_get_uri($world, $model, "file:" . $marketing_file);
/* process `delete' command and save the result to the data file */
if (isset($_GET["delete"])) {
  delete_id_from_model($world, $model, $_GET["delete"]);
  $result = serialize_model($world, $model);
  save_string_to_file($marketing_file, $result);
}
```

```
/* append a new marketing information to the data file */
if (isset($_POST["type"]) && $_POST["type"] == "newmarketing") {
   $id = $_POST["id"];
   $trend = $_POST["trend"];
   delete_id_from_model($world, $model, $id);
   add_new_type($world, $model, $id, "Product");
   add_new_value($world, $model, $id, "PriceTrend", $trend);
   $result = serialize_model($world, $model);
   save_string_to_file($marketing_file, $result);
}
/* fill `$model' with the information of products */
rdf_get_uri($world, $model, $product_uri);
/* build the query string of fetching the information of marketing */
$sql_string = $prefix .
   " SELECT * " .
   " WHERE { " .
   "   ?id rdf:type product:Product; " .
   "         product:ProductName ?name; " .
   "         product:PriceTrend ?trend " .
   " } ";
/* execute the query */
$result = rdf_query_model($world, $model, $sql_string);
/* free useless resource */
rdf_free_model($model);
rdf_free_storage($storage);
/* generate the page according to the results */
/* show all marketing information */
?>
<h1>Marketing</h1>
<table border=1>
<tr>
```

```
<td>ID</td>
<td>Name</td>
<td>Trend</td>
<td>&nbsp;</td>
</tr>
<?
foreach ($result as $entry) {
  $id = strip_rdfid($entry["id"]);
  echo "<tr>\r\n";
  echo "<td>$id</td>\r\n";
  echo "<td>$entry[name]</td>\r\n";
  echo "<td>$entry[trend]</td>\r\n";
  echo "<td><a href=\"?delete=$id\">Delete!</td>\r\n";
  echo "</tr>\r\n";
}
?>
</table>
<hr>
<a href="?download">Download RDF!</a>
<?php
/* show the section of appending new information of marketing */
if (!isset($_GET["append"])) {
?>
<hr>
<a href="?append">Append new marketing information</a>
<? } else {
  $result = get_products($world);
?>
<hr>
<form method=post action=marketing.php>
<table border=1>
```

```
<tr>
<td colspan=2>New Marketing Information</td>
</tr>
<tr>
<td>ID</td>
<td>
<select name=id id=id>
<?
  foreach ($result as $entry) {
    $id = strip_rdfid($entry["id"]);
    echo "<option>$id</option>\r\n";
  }
?>
</select>
</td>
</tr>
<tr>
<td>Trend</td>
<td>
<select name=trend id=trend>
<option>ascending</option>
<option>descending</option>
</select>
</td>
</tr>
<tr>
<td colspan=2>
<input type=hidden id=type name=type value=newmarketing>
<input type=submit>
<input type=reset>
</td>
```

```
</tr>

</table>

</form>

<?

}

?>
```

Reference

[i] [SGML] C. M. Sperberg-McQueen and Lou Burnard, *Chapter A Gentle Introduction to SGML* from *A Gentle Introduction to XML,* Referred by W3C official page: *Introduction to SGML,* Line 6. http://validator.w3.org/docs/sgml.html

[ii] [XHTML] Steven Pemberton, World Wide Web Consortium. (2004-07-21). "*HTML and XHTML Frequently Answered Questions*". http://www.w3.org/MarkUp/2004/xhtml-faq. Retrieved 2008-12-29. The definition of XHTML

[iii] [WDIFF] Graham Cormode and Balachander Krishnamurthy AT&T Labs–Research February 13, 2008, chapter 2. *Key Differences between Web1.0 and Web2.0*

[iv][SWGL] Ivan Herman, "*How would you define the main goals of the Semantic Web?*" Section 1.1, line 5, 2009-11-12 http://www.w3.org/2001/sw/SW-FAQ#swgoals

[v] [RDF] Frank Manola, Eric Miller, "*Introduction of the RDF Primer*" Chapter 1 Introduction first paragraph, 2004-02-10 http://www.w3.org/TR/rdf-primer/

[vi] [RDFOV] Ora Lassila, Ralph R. Swick "*Resource Description Framework (RDF) Model and Syntax Specification*" Chapter 1 Introduction. http://www.w3.org/TR/PR-rdf-syntax/

[vii] [RDFS] R.V. Guha, Dan Brickley, W3C Recommendation, 10 Feb, 2010. "*The introduction of the RDFS from W3C*"chapter 1 Introduction http://www.w3.org/TR/rdf-schema/

[viii] [OWL] James Hendler, Guus Schreiber, 2002, Smith, Michael K.; Chris Welty, Deborah L. McGuinness (2004-02-10). "*OWL Web Ontology Language Guide*". W3C. Retrieved 2008-07-15. Definition of OWL

[ix] [OWLOV] Deborah L. McGuinness, Frank van Harmelen, *OWL Web Ontology Language Overview* Introduction about the OWL. 10, Feb, 2004. http://www.w3.org/TR/2004/REC-owl-features-20040210/#s1,

[x] [SPARQL] Eric Prud'hommeaux, Andy Seaborne, "*SPARQL Query Language for RDF*", W3C Recommendation 15 January 2008, Introduction about the SPARQL. http://www.w3.org/TR/rdf-sparql-query/

[xi] [RDFCAS]Graham Klyne, Jeremy J. Carroll, "*Resource Description Framework (RDF): Concepts and Abstract Syntax.*" 2004 Feb. Abstract http://www.w3.org/TR/rdf-concepts/

[xii] [DDS]Arvols Chan, Umeshwar Dayal, Stephen Fox, Daniel Ries, "*Supporting a Semantic Data Model in a Distributed Database System.*" 1990. 354-355

[xiii] [FOAF] W3C Organization "*FOAF project*", introduction part. http://www.foaf-project.org/about

[xiv] [PBKEY] D. Cooper , S. Santesson , S. Farrell , S. Boeyen , R. Housley , W. Polk , "*Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*" May, 2008, Chapter 6.1.3. Basic Certificate Processing

[xv] [WEBSEC]David Scott, Richard Sharp "*Abstracting Application-Level Web Security*" 2001-11-09, chapter 1 Introduction and chapter 2 Application-Level Security, page 2-3

[xvi] [RDFNOT] Tim Berners-Lee "*RDFNot document*", Date: September 1998. Last modified: 1998/09/17 Section "*A Semantic Web is not Artificial Intelligence*" http://www.w3.org/DesignIssues/RDFnot.html

[xvii] [GRIDVINE] Karl Aberer, Philippe Cudr´e-Mauroux, Manfred Hauswirth, and

Tim Van Pelt, *"GridVine: Building Internet-Scale Semantic Overlay Networks"* 2004, Page 1-2 Chapter Introduction.

[xviii] [FOWL] Zou, Tim Finin and Harry Chen, *"F-OWL: an Inference Engine for the Semantic Web"* Youyong 2004, Example in the Introduction part.

[xix] [MRKB] Reference: Akerkar RA and Sajja Priti Srinivas: *"Knowledge-based systems"*, Jones & Bartlett Publishers, Sudbury, MA, USA (2009) Section The concept of machine-readable knowledge base.