Thesis for the Degree of Doctor of Philosophy

A Framework for User Preference Sharing based on Semantic Web in Personalized Services

by

Ju Yeon Kim

Interdisciplinary Program of Information Security

The Graduate School

Pukyong National University

February 2008

A Framework for User Preference Sharing based on Semantic Web in Personalized Services
개인화 서비스에서 시맨틱 웹 기반의 사용자 선호정보 공유를 위한 프레임워크

Advisor: Chang Soo Kim

by

Ju Yeon Kim

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in Interdisciplinary Program of Information Security, Graduate School,
Pukyong National University

February 2008

A Framework for User Preference Sharing based on Semantic Web in Personalized Services

A Dissertation

by

Ju Yeon Kim

Approved by :

(Chairman) Kyung-Hyune Rhee

(Member) Ki-Ryong Kwon

(Member) Ha-Joo Song

(Member) Am-Suk Oh

(Member) Chang-Soo Kim

February 26, 2008

Contents

List of Figures

List of Tables

개인화 서비스에서 시맨틱 웹 기반의 사용자 선호정보 공유를 위한 프레임워크

김 주 연

부경대학교 대학원 정보보호학협동과정

요 약

사용자의 요구와 선호도에 따라 적합한 정보를 제공해주는 개인화서비스에 대한 많은 연구와 개발이 진행되어왔다. 하지만 기존의 연구들은 단일시스템 내에서 사용자의 선호정보를 관리하기 때문에, 다양한 서비스들 사이에서 이러한 정보를 공유하기 어렵다는 문제가 있다. 특히, 최근 관심이 고조되고 있는 유비쿼터스 환경에서는 주로 제한된 성능, 제한된 사용자 인터페이스 및 낮은 대역폭의 통신환경을 제공하는 단말기에서 분산된 다양한 서비스에 대한 접근이 많이 이루어지기 때문에, 개인화서비스 뿐 아니라 다양한 서비스들 사이에 사용자 선호정보의 공유를 통한 효율적인 개인화서비스 지원이 요구될 것이다. 본 논문에서는 향상된 개인화서비스를 제공하기 위한 하나의 방법으로 시맨틱 웹 기반의 사용자 선호정보 공유 모델을 제안하고 이를 지원하는 사용자 선호정보 공유 프레임워크를 설계 및 구현한다.

본 논문에서 제안하는 사용자 선호정보 공유 모델은 서비스 온톨로지(Service-specific Ontology)들 상에서 사용자가 선호정보를 기술하도록 함으로써 사용자 선호정보가 의미를 기반으로 정확하게 표현될 수 있는 방법을 제공하고, 다양한 개인화 서비스들 사이에서 사용자 선호정보의 의미적 분석을 가능하게 함으로써 강력한 상호운용성(interoperability)을 제공한다. 특히, 본 논문에서는 제안하는 UPDL(User Preference Description Language)은 시맨틱 웹 환경에서 개인의 선호정보가 온톨로지를 기반으로 기술(describe)되어지는 것을 가능하게 한다. 또한, 본 논문에서는 제안한 사용자 선호정보 공유 모델을 지원하기 위한 사용자 선호정보 공유 프레임워크를 설계하고, 미들웨어로서 사용자 선호정보 관리 시스템(UPMS: User Preference Management System)를 구현한다. UPMS는 UPDL을 기반으로 사용자 선호정보들을 의미적으로 기술하고, 유사성 평가를 통해 사용자 선호정보를 다양한 개인화 서비스들 사이에서 공유하기 위한 매커니즘을 제공한다. 마지막으로, 본 논문에서는 응용사례들을 통해 제안하는 프레임워크에 대한 평가와 분석을 제공한다.

본 연구는 다음과 같은 장점을 제공한다. 첫째, 사용자의 선호정보는 서비스 온톨로지상에서 의미기반으로 기술되므로 보다 풍부한 표현력을 반영할 수 있다. 둘째, 개인의 선호정보가 온톨로지 기반의 UPDL에 의해 기술되고 공유되어짐으로써 상호운용성을 제공한다. 셋째, 제안 모델을 통해 제공되는 사용자 선호정보는 정확한 용어에 대한 추천뿐만 아니라 의미적으로 유사한 개념에 대해서도 추천이 가능하다. 넷째, 서비스제공자가 새로운 서비스를 추가하거나 변경하더라도 제안하는 미들웨어인 UPMS를 통해 다른 응용서비스들의 변경 없이 서비스가 가능하다. 즉, 제안방법은 각 서비스를 중심으로 사용자 선호정보를 다양하게 기술할 수 있으면서도 다양한 서비스들 간에 그 정보를 공유할 수 있기 때문에, 기존의 연구보다 효율적인

개인화서비스를 제공할 수 있다는 장점이 있다. 특히, 제안모델은 현재의 인터넷 환경에서 사용자에게 편리한 서비스 이용을 가능하게 할 뿐 아니라 제한된 성능, 제한된 사용자 인터페이스 및 낮은 대역폭의 통신환경을 제공하는 단말기에서 분산된 다양한 서비스에 대한 접근이 많이 이루어질 것으로 예상되는 유비쿼터스 환경에서 더욱 유용할 것으로 본다.

Chapter 1

Introduction

Personalization is a technology that provide users with adaptive information filtered from heterogeneous and numerous information[25]. As growth of the web, the needs of personalized services have increased in the various areas of web-based services such as E-commerce, E-education, and E-business[62]. Also, the personalized services have made human-computer interaction efficient, even in the restricted environments like mobile devices and wireless networks.

Although personalized services have provided users with much comfortable services in the various fields, the heterogeneous and numerous information has raised the problem in exchanging heterogeneous user preferences represented by various services[37][38][41].

In this thesis we propose a Semantic Web approach as a solution to share heterogeneous user preferences among the various and decentralized services. Semantic Web is a technology to add well-defined meaning to information on the Web to enable computers as well as people to understand meaning of the documents easily[13][52][68]. This chapter introduces our challenges and contributions to enhance both heterogeneity and interoperability in exchanging user preferences used in personalized services.

1.1 Research Issues

Personalization is the ability to provide contents and services tailored to individuals based on user's information. With growth of the web, there have been a lot of researches on the personalized services that dynamically adapts services or contents based on the personal information such as preferences, behaviors and other relevant facts. These efforts have enhanced the convenience of accessibility to user-adaptive information and the quality of user interaction in the restricted systems used in wireless environment like ubiquitous.

Although these personalized services provide users with much comfortable access to user adaptive information, current personalized services have imposed the burden of registration of user preferences on users. For instance, users should input their preferences into each personalized service even if they register the same preferences. To resolve the problem, the research on interoperability for user preferences have been proposed. As the most general approach to provide interoperability among information, the standard exchange format and metadata for each specific domain have been studied and defined. For user information, the IEEE PAPI and the IMS LIP have been used to the user exchange user information in education domain, and CC/PP, Dublin Core, and SUMO/MILO have been defined to describe user information[29-31].

Despite of these efforts, they have fundamental problems as follows. First, user preferences cannot reflect heterogeneous terms of each service because the standard exchange format restricts expressiveness about user preferences. Second, current approaches cannot provide concept-based sharing. That is to say, these approaches cannot share the meaning of similar terms and synonym.

1.2 Contributions

We discussed some research challenges to enhance personalized services in the environments with distributed services and heterogeneous contents in previous section. In this section we present our solution and contributions for these challenges.

1.2.1 Overview of the solution

Our goal is to develop a user preference sharing framework that supports the novel personalized services that can access and utilize user preferences not only in own service but also in other personalized services. For the purpose, we propose ontology-based user preference sharing model. We especially specify an ontology for describing user preferences. It can provide interoperability among heterogeneous personalized services by allowing user preferences to be described and shared over service-specific ontologies.

Figure 1.1 shows the conceptual structure of our user preference sharing model for personalized services, and it is organized to provide the following two advantages. First, our model provides rich expressiveness of user preferences by specifying them over service-specific ontologies. Because service-specific ontologies are classified as the purpose and the feature of each service, they can represent user preferences more correctly. Second, our model provides strong interoperability among personalized services because user preference is specified by OWL-based description language for user preferences and service-specific ontologies. Even if a term of user preferences in the user profile is not equal to terms in the ontology of a personalized service currently used, the service can compute similarity from corresponding concept of higher level. The advantages are achieved by our proposed user preference sharing model and the framework to support our model.

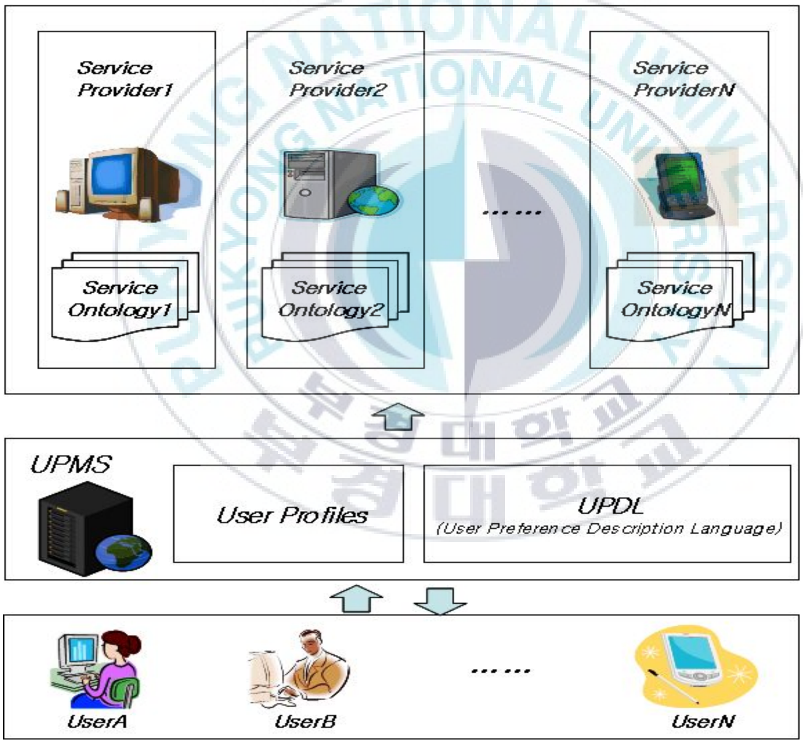


Figure 1.1 The conceptual structure of the user preference sharing model for personalized services.

1.2.2 User Preference Sharing Model

In this thesis, we propose the User Preference Sharing Model that includes a data model for user preferences and its sharing mechanism. For the data model, we specify the User Preference Description Language (UPDL). It is an OWL-based description language that allows users to describe their preference over service-specific ontologies. In our model, we suppose that each personalized service provides own service-specific ontology classifying concepts used in each service and specifying the relationship between the concepts. For the sharing mechanism, we provide a similarity evaluation mechanism. This mechanism computes similarity between two concepts from two different service-specific ontologies based on the hierarchy of ontologies[60].

1.2.3 User Preference Sharing Framework

In this thesis, we design the User Preference Sharing Framework, and realize a middleware for the framework. The framework for our User Preference Sharing Model is based on three-tiered model. In the application tier, personalized services serve their contents based on their service-specific ontologies, and they recommend user-adaptive contents using user preferences received from the middleware. In the data tier, user preferences described by UPDL are stored in the user profile. The User Preference Management System (UPMS) is a middleware to support our User Preference Sharing Model.

UPMS plays the roles of acquisition and sharing of user preferences through three different kinds of managers as follows. User Manager registers new users and creates new user profile. Acquisition Manager provides a unified user interface to acquire user preferences over service-specific ontologies and updates user preferences in the user profile. Access Manager offers application programming interface (API) for personalized services, and it returns the evaluated user preferences and weights computed by Similarity Evaluator.

1.2.4 Main Contributions

The details of main contributions of our approach are belows.

- Rich expressiveness (Heterogeneity): The UPDL ontology provides more expressive and flexible description mechanism for personal preferences by allowing personal preference to be described over various service-specific ontologies.

- Interoperability: Our approach supports the sharing of user preferences among various personalized services because user preferences are described by the UPDL specified based on OWL and shared by the similarity evaluation.

- Flexibility (Scalability): Even if new services are appended or service-specific ontologies are extended, user preferences can be described and shared in the same method without modifying the middle and applications.

- Benefits in personalized services: Our approach provides the advanced mechanism for personal preference profiling. Because our profiling approach is based on hierarchically classified ontologies, it is possible to recommend user-adaptive information not only with explicit keyword but also with similar concept.

1.3 Organization of Thesis

This thesis is organized as follows. Chapter 1 briefly introduces our approach to enhance personalized services and main contributions. Chapter 2 presents some backgrounds for three research fields related with our approach - personalized services, information sharing, and semantic web. Chapter 3 proposes a user preference sharing model to enhance both heterogeneity and interoperability in personalized services. Chapter 4 proposes a framework for user preference sharing among different services. This chapter especially presents the User Preference Management System (UPMS) that is a middleware of our framework to enhance interoperability among heterogeneous services. Chapter 5 realizes personalized services based on several service-specific ontologies and our UPMS, and then we discuss evaluation and analysis. Finally, Chapter 6 contains the conclusions of this thesis.

Chapter 2

Background

In this thesis, we adopt semantic web technologies to personalized services in order to share user preferences in personalized services. This chapter provides some backgrounds on personalized services, information sharing, and Semantic Web. This chapter begins with background research for personalized services that provides users with adaptive information. Then, this chapter introduces the efforts of interoperability to share the information among different systems or services. The Semantic Web that is a technologies to add well-defined meaning to information on Web also introduced in this chapter.

2.1 Personalized Services

2.1.1 Overview of Personalized Services

Recently, there are the various web-based applications that reflect each domain to provide users with the specialized information. For example, there are tourism services that provide users with specialized tourism information and related information [34-36][61][72]. Although the distributed applications on web benefits from numerous and various information, they are difficult to provide users with adaptive information. As a solution, personalization was proposed to overcome the one-size-fits-all by observing the user's needs according to interaction processes and provide users with satisfactions by providing individually optimized access to numerous information.

Personalized services provide users with adaptive information filtered from heterogeneous and numerous information. Therefore, systems for personalization help each individual person to adapt user-tailored contents and services based on the user's information. As growth of the web, the needs of personalized services have increased more in various areas of web-based services such as E-commerce, E-education, and E-business. Blom defined personalization as a process that changes the functionality, interface, information content, or distinctiveness of a system in order to increase its personal relevance to individual user. Thus, personalization has to consider how systems enhance quality of its interaction for personalized services[25][37][38][41][72]. Most systems which provide personalized services take into account user profiles in which information such as user's interests and preferences is generally stored.

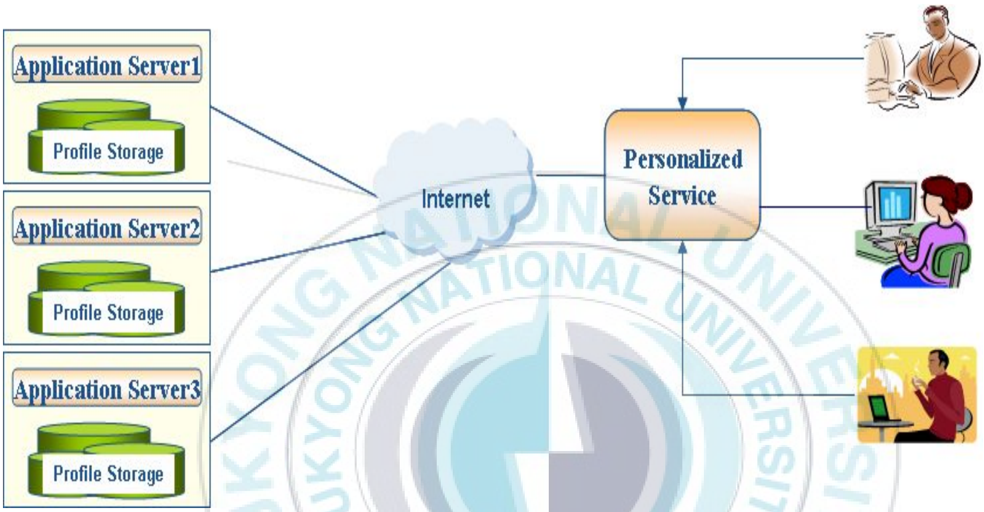


Figure 2.1 Overview of personalized services.

2.1.2 Personalized Applications

Recently, there are some research on enhance quality of interaction as well as contents[5][13]. Personalized applications are managed to solve information overload by tailoring the information presented to individuals users. These applications usually consider how to provide each user with adaptive access or adaptive filtering on numerous information as shown in Table 2.1. The personalized access provides web or filesystem as personalized portals, and the personalized filtering is to filter and rank the information on each service such as newspapers, Usenet news, and recommendation services.

Table 2.1 The types of personalized applications.

Personalization	Description	System
Personalized Access	A access management provides users with personal information space such as bookmark.	Yahoo Google BASAR
Filtering and Rating	F&R management recommend consumers higher interesting items Personalized Browsing, Personalized Search	Personal Wall Street Journal The amazon web site WebWatcher PEA, Siteseer

(1) Personalized Access

As the amount of information rapidly increase in these days, personalized approaches on information access are needed. Personalized access is to manage personal information by updating in bookmarks or cookies files. The approach is usually consider how to access to user-adaptive information as personalized manner. It provides users to hyperlinks to different information such as news, weather, stock market, and television programs, and they allow users to specify topics on user's interests.

For example, Yahoo[1] and Firefly[2] provide the personalized information by finding persons with similar interests based on users' search history stored in system. BASAR(Building Agents Supporting Adaptive Retrieval) provides users with assistant when managing their personal information spaces[53]. The e-commerce is also a common application area of personalization. For instance, amazon.com sends information about new interesting books based on a list of categories that the user enters. User interests in the systems are determined by keywords or reviews they write. These information are typically stored in form of cookies, and it enable personalized systems to recommend adaptive information or services.

(2) Filtering and Rating for personalization

Filtering and rating have important roles in personalized services because they provide filtered information and higher interesting items. For instance, the Personal Wall Street Journal[3] that

1) www.yahoo.com
2) www.firefly.net
3) www.wsj.com

personalize newspapers acquires user interests by clicking categories of interest, and proposes links or articles related to user interests in the user profile. The WebMate[31] helps users to effectively browse and search for the Web. The articles associated with headlines are compared to the user's profile, resulting in a personalized presentation of news. The profile consists of the cluster centers together with their associated documents.

These recommendation services usually present lists recommended according to user's interests[1]. Filtering based on user profiles have been implemented in various recommendation applications and web browsing[57][67]. For example, WebWatcher[53] provides users with lists of keywords at the beginning of browsing to present user's interests in user profiles. As other forms of user profiles, profiles of the PEA essentially have bookmark files, similar to Siteseer. In the system, different folders represent different classes of interest[54]. The user profiles for personalized services are introduced in Section 2.1.3.

(3) Commercial Personalized Services

A few personalized services such as Start-On[4], My yahoo 5.0, and Eye Google are known as personized portal services. Recently, a personalized service in mobile environments is devel-

4) www.start-on.co.kr

oped in order to provide users with popular items. The '1mm Expert Agent[5]' provides users with general recommendation services such as news, weather, TV, movie, and food. However, the services provide only general information like generally popular food or TV programs.

In commercial services area, personalization is proposed according to each domain. For example, music system[22] proposed the music recommendation in a smart office to recommend music based on users' context. The music recommendation is based on the user's favorite genres and the current mood of the user by collecting and analyzing the contextual information such as mood and preferences of the user. The LBS system is another example. It proposed a profile-based approach to improve the efficiency in the location-based services. The XML-based profile specification takes the history of users' activities into account[72].

2.1.3 User Profiles for Personalized Services

In personalized services, user information is gathered into a system in order to delivery user's interesting information to individual user. Many personalized applications provide the users with customized services by using user profile. This approach can

5) www.1mm.com

provide the assured information reflecting the user's explicit preferences for design or display of certain content.

Generally, there are several approaches to provide the profile-based personalized services as follows. First, the user programming approach is programmed according to rules in order to process the related information. The main problem of this approach is that it requires too much effort to recognize proper conditions from individual user. Second, the knowledge engineering approach makes domain-specific knowledge of both the application and the user. However, the approach requires substantial efforts in eliciting domain knowledge because it need programming skills of knowledge engineer instead of the end user. Third, the machine learning mechanisms approach is to acquire adaptive knowledge from users. The approach provide the customized services by learning user's behaviors.

In order to obtain user preferences in the systems, they allow user to input items, record visited pages, or count the number of clicks. For the explicit information about user preference, user preferences can be acquired as following manners. First, An explicit user profiling is elicited from the presented questions to acquire user interests or preferences. Although the approach for explicit user profile can be easily deduced from the provided data, it requires a variety of effort from users. In order to capture properly user preferences, the approach need a long period interaction between users and systems. Next, implicit knowledge acquisition has little impact on the regular activities of users. In order to discover behavioral patterns such as user interests or preferences, users record information that can be used to manage their interests. While implicit knowledge has inherently error prone process, explicit knowledge have generally high confident information. In order to explicit acquisition of user's information, systems need to interrupt with the user. This is no guarantee that the questions asked are answered truthfully, or even that the questions asked are the right ones to obtain the desired information[54][67].

2.2 Technologies for Information Sharing

2.2.1 Overview of Information Sharing

A variety of applications which exist on the network centralize the information of each domain into an application server. Centralized systems are available anywhere, but they need a large database to store the information or contents. Futhermore, the centralized approaches cause high network traffic, and require servers with high performance. According to the problem, systems which provide services on the web are becoming increasingly distributed. However, the information sharing among the distributed services and systems has been required.

Information Sharing can be achieved by interoperability among data in distributed systems. Interoperability can be circumscribed as "a condition that exists when the distinctions between information systems are not a barrier to accomplishing a task that spans multiple systems"[64]. Therefore, interoperability among different systems enables systems to exchange information between different systems. The table 2.2 shows standards for interoperability. In 1960s, there were the efforts for interoperability of the hardware interfaces including operating systems, DBMS and so on. This stage tried to overcome interoperability among different operating system. In 1970s, there were the effort for interoperability of program interface, and the standards for interoperability among the heterogeneous and distributed data were specified from 1980s. After 1990s, there have been the efforts for information sharing to provide interoperability of both data and its meaning[31].

Table 2.2 Efforts for interoperability.

Years	Standard of interoperability
1960 ~	▪ Intergrated services on Web ▪ Hardware Interface
1970 ~	▪ Structure interoperability among different applications ▪ Program Interface
1980 ~	▪ Communication Protocol ▪ Data standards
1990 ~	▪ Modeling Facility ▪ Metadata standards
2000 ~	▪ Semantics & Ontology ▪ Logic, Proof, Trust

The profits of the information sharing have been presented in industry area like follows [48]. First, it allows businesses to ascertain customer needs accurately and meet those needs rapidly and efficiently. Second, it permits consumers to be informed rapidly and at low cost of those opportunities in which they are most likely to be interested. Third, it promotes competition by facilitating the entry of new competitors into established markets, reduces the advantage that large, incumbent firms have over smaller startups, and encourages the creation of businesses specialized in satisfying specific consumer needs. Fourth, it expands consumer access to a wide range of affordable services and products. Fifth, it enhances customer convenience and services. Next, it improves efficiency and significantly reduces the cost of many products and services. Finally, it facilitates the detection and prevention of fraud and other crimes.

2.2.2 Interoperability in Personalized Services

Personalized services should deliver contents to users as effective manners. In order to provide these services, systems for personalization have to be interoperable among existing corporate systems. Existing standard and specification enables users to exchange their information regularly by using element sets that are specifically designed to describe people and their interests. For example, the Friend of a Friend (FOAF) element set provides a set of properties and classes, focusing initially on people, documents, organisations, images etc. FOAF is a simple vocabulary for describing social networks, people, organisations etc. The schema is still under development[5].

The vCARD was also specified. The vCARD is a set of metadata elements defined by the Internet Engineering Task Force (IETF) as a standard for representing information about people and organisations, such as that which is profiled in a common business card. The vCards carry directory information such as name, addresses (business, home, mailing, parcel), telephone numbers (home, business, fax, pager, cellular, ISDN, voice, data, video), email addresses and URLs. vCards can also contain graphics and multimedia (photographs, company logos, audio clips)[29].

Recently, there are several approaches to provide standards of interoperability among systems or web documents. First, as a standard activity of interoperability among different systems, there is the W3C's Composite Capabilities/Preference Profile (CC/PP) specification. The specification addresses the problem of describing device capabilities of mobile devices with highly divergent input, output and network connectivity capabilities as well as user preferences. The basic idea behind the CC/PP framework is content adaptation and thus adaptive hyper media. Second, there are also the PAPI standards in the education field. IEEE Public and Private Information (PAPI) standards deal with several categories for information about a learner. A key feature of the PAPI Learner Standard is the logical division, separate security, and separate administration of several types of learner information. The first type is personal information such as name, address, social security number. Next, there are six types of information such as security information, performance information, and so on[30-31].

Besides, the Dublin Core describes web documents by using metadata elements. The role of metadata elements is to help providers manage information and describe information resources. The element set was originally developed at the Metadata Workshop in Dublin, Ohio in 1995. The Dublin Core Metadata

Element Set contains definitions for properties of elements[64]. It is fifteen elements such as title, creator, subject, description, publisher, and so on. Figure 2.2 shows an example of web page described using Dublin Core metadata.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description rdf:about="http://www.dlib.org">
   <dc:title>D-Lib Program - Research in Digital Libraries</dc:title>
   <dc:description>The D-Lib program supports the community of people
    with research interests in digital libraries and electronic
    publishing.</dc:description>
   <dc:publisher>Corporation For National Research Initiatives</dc:publisher>
   <dc:date>1995-01-07</dc:date>
   <dc:subject>
    <rdf:Bag>
     <rdf:li>Research; statistical methods</rdf:li>
     <rdf:li>Education, research, related topics</rdf:li>
     <rdf:li>Library use Studies</rdf:li>
    </rdf:Bag>
   </dc:subject>
   <dc:type>World Wide Web Home Page</dc:type>
   <dc:format>text/html</dc:format>
   <dc:language>en</dc:language>
  </rdf:Description>
</rdf:RDF>
```

Figure 2.2 A web page described using Dublin Core elements.

In order to personalized services, applications need to provide efficiency and interoperability among systems. According to needs of interoperability, the W3C proposed a specification to describe user information as known PIDL. The Personalized Information

Description Language (PIDL) described propose a unified framework for services to both personalize and disseminate information. Therefore, the PIDL be able to facilitate personalization of online information by providing enhanced interoperability among personalization applications. The PIDL, XML-based document, supports personalization of different media and different delivery methods such as SMTP, HTTP, IP-multicasting, etc. XML is emerging as the standard for data exchange on the Internet. Therefore, the technology enhances the ability of remote applications to interpret on the Internet. However a tagged document is not very useful without some agreement among inter-operating applications that is to say, what the tags mean and it is common DTDs which provide for this. A DTD specifies the structure of an XML document by specifying the names of its elements, sub-elements and attributes[59].

2.3 The Semantic Web

The Semantic Web is a vision for the idea of having data defined and linked on the Web in a way. It can be used by machines not just for display purposes, but for automation, integration, reuse of data across various application[40][42-45]. Therefore, the technology enables computers as well as people to understand meaning of documents on the web. To do this, semantic web process semantically by adding well-defined meaning to information on the Web.

2.3.1 Semantic Web Technologies

HTML-based Web documentations have several problems analogous to Web services across the Internet or within a intranet for group (or corporation). Firstly, information overload occur because of rapid rate of growth in the amount of information available. Secondly, current information technologies are a stovepipe system which all the components are hardwired to only work together. Accordingly, information only flows in the stovepipe and cannot be shared by other systems or organizations. Finally, poor content aggregation is happen. Putting together information from disparate sources is a recurring problem in various aggregation areas[2][4][12][19].

These days keyword-based search engines help people search for needed information on web. It is clear that numerous information provide successfully. However there are also following problems. The first problem is high recall, low precision that result from numerous information on the web. Next, low or no recall. Often it happens that we don't get any answer for our request, or that important and relevant pages are not retrieved[49][52] The key challenge for the Semantic Web community is to push technology in a similar direction. Recently, there have been many approaches of applications and information browsing in semantic web area[23][36][39][42]. The semantic web technologies provide the practical applicability of current research by integrating different technologies[12][15][19][68].

The Semantic Web has the layered structure to enable Web documents to intellectualize. The W3C has been a leader in developing technologies for the Web. The W3C specifies a diagram labeled Architecture on the Semantic Web, called the "Semantic Web layer cake", in Figure 2.4. The Semantic Web consists of Uniform Resource Identifier (URI), UNICODE, Resource Description Framework (RDF), RDF Schema, and Ontology hierarchically. The RDF is an XML-based language to describe resources like images, audio files, or concepts available via the Web. The RDF contents can be searched, integrated, and inferred semantically. Figure 2.3 depict layered technologies of the semantic web[6][15].

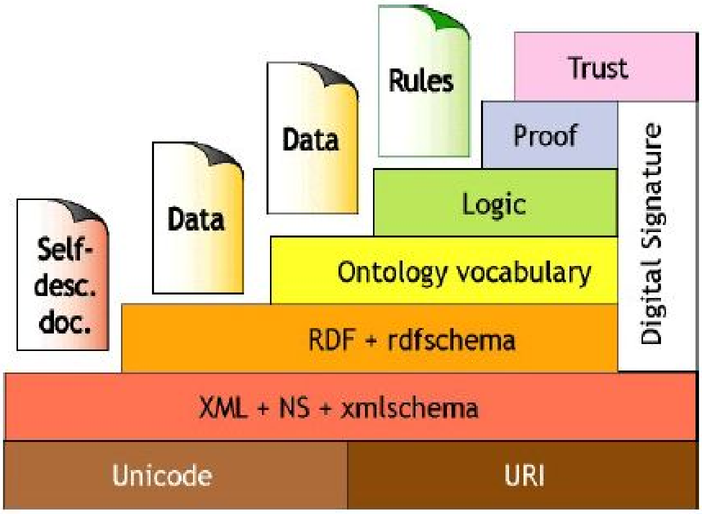


Figure 2.3 The semantic web stack.

2.3.2 Resource Description Framework

RDF (Resource Description Framework) is emerging as one of the primary languages for encoding semantic material on the web. The technology has become based on XML technology known as a universal meta language for defining markup. The Resource Description Framework (RDF) was developed by W3C for modeling semi-structured metadata and enabling knowledge-management applications. A RDF model depicts directed graph to represent its information, and contains its knowledge in form of triples so called subject-predicate-object format[12][29].

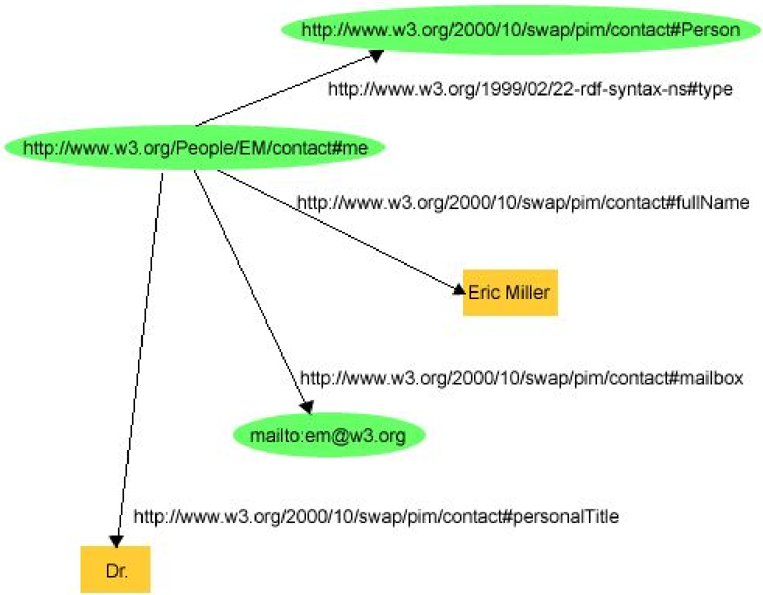


Figure 2.4 Graph of an RDF.

The RDF data model define relationship between concepts as RDF triple format[15].

- Subject : The subject is the noun or noun phrase that is the doer of the action. In the sentence "The person is named John Myers", the subject is "the person." The subject of the sentence tells us what the sentence is about. In RDF, this is the resource that is being described by the ensuing predicate and object.
- Predicate : The predicate is the part of a sentence that modifies the subject and includes the verb phrase. In previous sentence, the predicate is the verb "is named". In order words,

the predicate tells us something about the subject. In RDF, a predicate is a relation between the subject and the object.

- Object : The object is a noun that is acted upon by the verb. Returning to previous sentence, the object is the noun "John Myers." In RDF, an object is either a resource referred to by the predicate or a literal value.

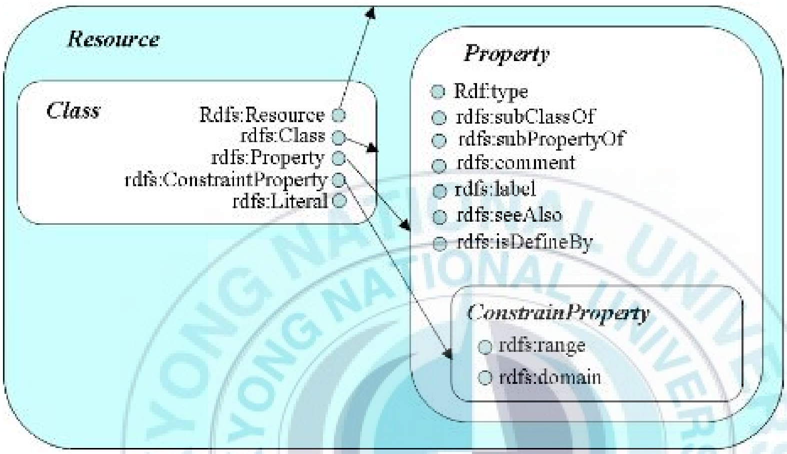


Figure 2.5 Relationship between resource and property.

RDF identify things using Web identifiers such as Uniform Resource Identifiers, or URIs, and resources in terms of simple properties and property values. This enables RDF to represent simple statements about resources as a graph of nodes and arcs representing the resources, and their properties and values[66].

```xml
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">

  <contact:Person rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:fullName>Eric Miller</contact:fullName>
    <contact:mailbox rdf:resource="mailto:em@w3.org"/>
    <contact:personalTitle>Dr.</contact:personalTitle>
  </contact:Person>

</rdf:RDF>
```

Figure 2.6 An example of RDF/XML describing Eric Miller.

Like HTML, this RDF/XML is machine processable and, using URIs, can link pieces of information across the Web. However, unlike conventional hypertext, RDF URIs can refer to any identifiable thing, including things that may not be directly retrievable on the Web (such as the person Eric Miller).

RDF such as above example is a universal language that users describe resources using their own vocabularies. However, RDF does not make assumptions about any particular application domain, nor does it define the semantics of any domain. RDF Schema (RDFS) is up to users to define their own terminology. RDFS[6] define the vocabulary and specify properties objects. Also, the schema describes relationships between objects, and what values they can take[47].

2.3.3 Ontology

Ontologies are a key enabling technology for the Semantic Web. The term of ontology is borrowed from philosophy where an ontology denotes a systematic account of Existence. In the context of the Semantic Web, an ontology denotes a description of the concepts and relationships that exist for a specific domain[16]. Ontologies provide a shared understanding that, together with the declarative data representation, can potentially enable different systems to utilize and understand data written by different users who did not communicate but who both chose to use the same ontology[21]. Ontologies interweave human understanding of symbols with their machine process ability. More recently, the use of ontologies has also become widespread in fields such as intelligent information integration, cooperative information systems, information retrieval, electronic commerce, and knowledge management[33].

Ontologies are now central to many applications such as scientific knowledge portals, information management and integration systems, electronic commerce, and semantic web services[17][19]. Also, ontologies supporting tools offers an opportunity to significantly improve knowledge management capabilities in large organizations. It describes a Semantic Web-based knowledge management architecture and a suite of innovative tools for semantic information processing[21].

```
<owl:Class rdf:about="#associateProfessor">
            <owl:disjointWith rdf:resource="#professor"/>
            <owl:disjointWith  rdf:resource="#assistantProfessor"/>
</owl:Class>
```

```
<owl:TransitiveProperty rdf:ID="is-part-of"/>

<owl:ObjectProperty rdf:ID="eats">
            <rdfs:domain rdf:resource="#animal"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="eaten-by">
            <owl:inverseOf rdf:resource="#eats"/>
</owl:ObjectProperty>
```

Figure 2.7 An example of ontology.

A number of research groups had already identified the need for a more powerful ontology modeling language. OWL (Web Ontology Language)[46] is a W3C project to standardize a more capable ontology framework language than RDFS[6][29]. OWL evolved from DAML+OIL[18], a relatively successful ontology project of DARPA, the United States Defense Advanced Research Projects Agency[28].

The Semantic Web effort has produced OWL, an ontology language for the web. As described in [16]: "OWL is intended to be used when the information contained in documents needs to be processed by applications, as opposed to situations where the content only needs to be presented to humans. Unlike

DAML+OIL, OWL is originating as a World Wide Web Consortium(W3C) sponsored language[70]. The W3C's Web Ontology Working Group was formed in November 2001, and the first official version of OWL is anticipated to be available in early 2003. OWL can be used to explicitly represent the meaning of terms in vocabularies and the relationships between those terms". By defining the basic semantics of the OWL building blocks, theories like description logic can be used to perform reasoning about the information described in OWL[46].

An ontology has to be represented by predefined languages. Currently, there are well-known ontology representation languages such as web-based RDF, XML, HTML[24], and besides, there are ontology language such as OIL(Ontology Interchange Language)[18], KIF(Knowledge Interchange Format). These ontology languages utilize web standards such as XML and RDF schema, or syntax derives from first-order predicate calculus. Recently, these ontology languages can create conveniently by using tools for ontology representations. For example, there are various ontology development tools such as OilEd[18], Protégé-2000[6], SMORE(Semantic Markup, Ontology and RDF Editor) for ontology representation as ontology editor[17][32]. These tools are implemented as JAVA programming language, and also supported to ontology editor including semantic tree viewer, import, or ex-

6) http://protege.stanford.edu/

port of RDF[50][51]. For example, [23] is introduced with pizza ontology to represent ontology-based knowledge modeling, and [64] is introduced on ontology-driven development in the semantic web. Swoogle is known as research on search engine for RDF[66]. Besides, there are various applications based on semantic web technologies[34][38][42][45].

Chapter 3

User Preference Sharing Model

The personalized services provide users with adaptive information based on their personal information such as their objects, preferences, behaviors and other relevant facts. The user preferences have been especially considered one of the most important and efficient elements to filter and rank user-adaptive information from numerous information. Although the personalized services have enhanced user-computer interaction, there is a complex issue that is to provide interoperability for user preferences in heterogeneous personalized services.

In this thesis, we focus on how to enhance both heterogeneity in describing user preferences and interoperability in various personalized services. For the objective, we propose Semantic Web approach for user preference sharing. In this chapter, we describe user preference sharing model including a data model for user preferences and a sharing mechanism for our data model.

3.1 Conceptual Model

Decentralized infrastructures are becoming increasingly popular on the Internet and the Web. A distributed system on the web generally maintain partial views in only its environment. Because these decentralized systems have abundant information, personalized services that provide users with adaptive information is necessary. The user preferences have been considered one of the most important and efficient facts to filter and rank user-adaptive information from numerous information. However, current personalized services have imposed the burden of registration of user preferences on users because users should input own preferences into each personalized service even if they do the same preferences. Therefore, Interoperability of user preferences is required in the personalized services.

Our goal of this work is to enhance interoperability by sharing user's preferences in heterogeneous personalized services. In this thesis, we consider about semantic interchange so as to recommend higher preferences to users. For our goal, we propose a user preference sharing model including an ontology-based data model for user preferences and a sharing mechanism for the data model. Our model enables user preferences to be shared and referred semantically by using Semantic Web technologies.

Figure 3.1 shows the conceptual structure of our model. A main feature of our model is to allow user preferences to be described over various service-specific ontologies.

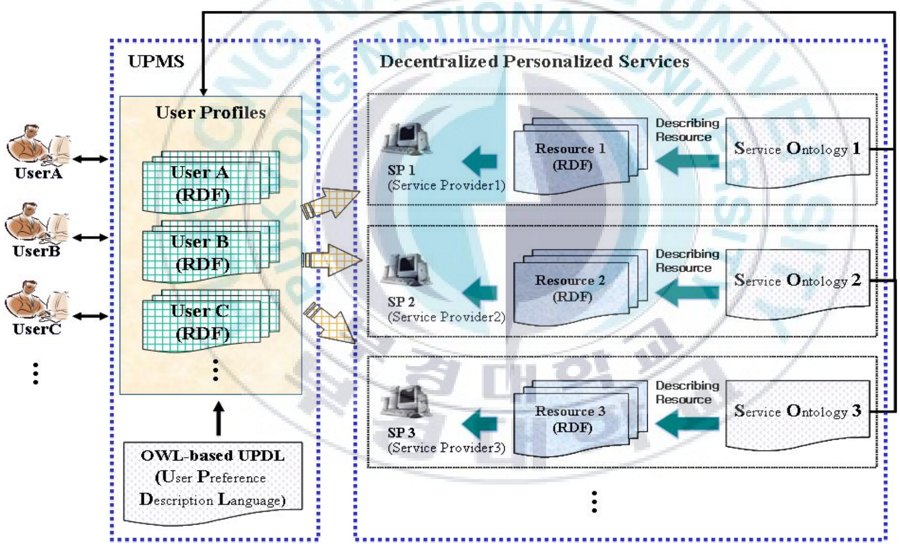


Figure 3.1 Conceptual service model for personalized services.

In our model, each service defines its service-specific ontology based on classification and features of its resource, and the contents of each service are described according to its service-specific ontology (refer to Section 3.2.1). Next, users describe their preferences over various service-specific ontologies. The user preferences are described by UPDL that is an OWL-based description language for user preferences (refer to Section 3.2.2). Finally, the user preferences is shared among personalized services using the Semantic Web technologies and our proposed algorithm for user preferences sharing (refer to Section 3.3).

Our model is organized for the following two advantages. First, our model provides rich expressiveness (heterogeneity) of user preferences by specifying them based on service-specific ontologies. Because service-specific ontologies are classified as the purpose and the feature of each service, they can represent user preferences more correctly. Second, our model offers strong interoperability among personalized services. Because user preferences is specified by OWL-based UPDL and service-specific ontologies, if a term in the user profile is different from terms in the service-specific ontology of the current service, the service can compute similarity from corresponding concept of higher level.

The novel services are provided based on three important components - service-specific ontologies, UPDL (User Preference Description Language), and the sharing mechanism for user preferences. In reminder of this chapter, detail of each component is described.

3.2 Description of User Preferences

In our model, the ontology-based user profiling is used to describe user preferences. Ontology-based user profiling approach takes advantage of the knowledge contained in both general-purpose and domain ontologies instead of acquiring and modeling user profiles from scratch. An ontology is a specification of concepts and their relationships which implies an agreement on vocabulary usage, sharing, and reuse of knowledge. In this user profiling approach, user interests are mapped to concepts in the reference ontology by classifying the instances of user interests into ontological concepts. Then, profiles are represented in terms of which concepts from an ontology a user is interested in irrespective of the specific instances of such interests. The use of ontologies for the representation of user interests promises to close the semantic gap between the previous approaches and the more abstract, conceptual view users may have of their interests.

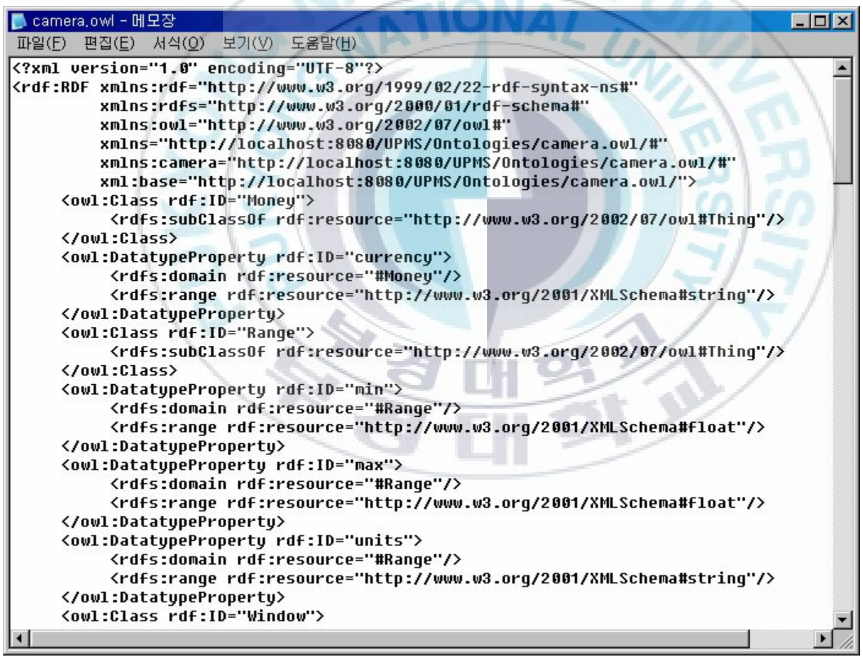
Our model describes user preferences based on UPDL ontology and heterogeneous service-specific ontologies. The service-specific ontologies are defined as a special one of domain ontologies, and it classifies contents as the feature of each service. The UPDL is an OWL-based description language for user preferences. Each ontology has its unique XML namespace. For this section, we use prefixes referring to these namespaces as follows. The prefix 'updl' refers to namespace of UPDL ontology, and service-specific prefixes refer to namespaces of service-specific ontologies. For example, 'acm' refer to the namespace of ACM classification ontology. The remainer of this section details ontologies to describe user preferences.

3.2.1 Service-specific Ontologies

Domain ontologies define concepts for each domain and relationship between them. The domain ontologies can be specified by service providers or Standard Organizations. Because taxonomies of contents are different as their domain, the domain ontologies help domain-specific information to be represented more exactly. Several technologies of domain ontologies are introduced in the study of [5][23][28][58].

In this thesis, we define the service-specific ontology as a special one of domain ontologies, and it describes classification and properties of contents provided in each service. The service-specific ontology provides main concepts and properties to describing contents in a service on the Semantic Web. The service-specific ontology is generated by a machine-readable OWL expressed in

RDF/XML, and metadata of each content in the service is described based on the service-specific ontology. The service-specific ontology is an attempt to link all the information about contents in a service. The goal is to express classification and all relations between contents to help people to browse and find anything about contents. It is based around the use of machine readable information provided by any web site or web service on the Web. The service-specific ontologies are referred by the instances of UPDL ontology described in Section 3.2.2 to specify the service-specific user preferences. Figure 3.2 shows a simple example of the service-specific ontology.




```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
         xmlns:owl="http://www.w3.org/2002/07/owl#"
         xmlns="http://localhost:8080/UPMS/Ontologies/camera.owl/#"
         xmlns:camera="http://localhost:8080/UPMS/Ontologies/camera.owl/#"
         xml:base="http://localhost:8080/UPMS/Ontologies/camera.owl/">
     <owl:Class rdf:ID="Money">
          <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
     </owl:Class>
     <owl:DatatypeProperty rdf:ID="currency">
          <rdfs:domain rdf:resource="#Money"/>
          <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
     </owl:DatatypeProperty>
     <owl:Class rdf:ID="Range">
          <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
     </owl:Class>
     <owl:DatatypeProperty rdf:ID="min">
          <rdfs:domain rdf:resource="#Range"/>
          <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
     </owl:DatatypeProperty>
     <owl:DatatypeProperty rdf:ID="max">
          <rdfs:domain rdf:resource="#Range"/>
          <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
     </owl:DatatypeProperty>
     <owl:DatatypeProperty rdf:ID="units">
          <rdfs:domain rdf:resource="#Range"/>
          <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
     </owl:DatatypeProperty>
     <owl:Class rdf:ID="Window">
```

Figure 3.2 RDF schema for service-specific ontology.

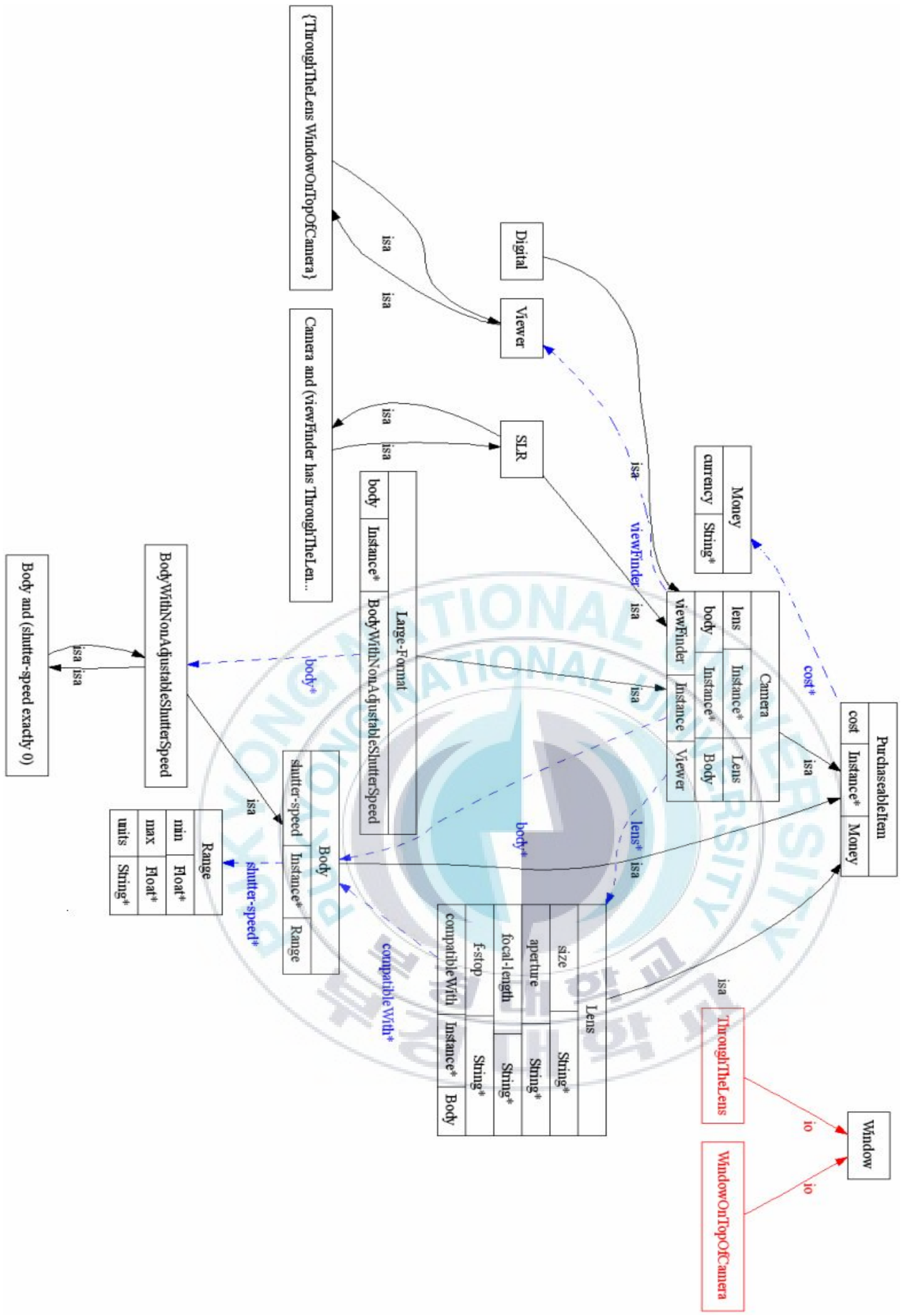


Figure 3.3 Visualization of the service-specific ontology.

To develop the service-specific ontologies, expert knowledge of each service and ontology modeling is necessary. Existing metadata and standards developed by knowledge experts and ontology developers in various areas can help service providers to specify their service-specific ontologies. On the web, there are many sets of metadata that are not commercial use, but open. For example, the ACM Computing Classification System (CCS) can be used as a classification ontology in the field of computer Machinery. Its full classification scheme involves three concepts: the four-level tree, general terms, and implicit subject descriptors. We will show the example of the service-specific ontology of the paper retrieval service generated based on ACM CCS in Chapter 5.

Besides above metadata, The SUMO and MILO are used as upper-level concepts of service-specific ontologies. The SUMO[7] ontology is being created as part of the IEEE Standard Upper Ontology Working Group. The goal of this Working Group is to develop a standard upper ontology that will promote data interoperability, information search and retrieval, automated inferencing, and natural language processing. The MILO[8] ontology is a mid-level ontology that is intended to act as a bridge between the high-level abstractions of the SUMO and the low-level detail of the domain ontologies. An upper ontology is limited to concepts that are

7) Suggested Upper Merged Ontology, http://ontology.teknowledge.com/
8) Mid-Level Ontology, http://ontology.teknowledge.com/

meta, generic, abstract or philosophical, and hence are general enough to address at a high level a broad range of domain areas. Concepts specific to particular domains are not included in an upper ontology, but such an ontology does provide a structure upon which ontologies for specific domains can be constructed. In our model, user preferences can be shared more efficiently when service providers put to use upper-level of a service-ontology by using existing standards or metadata, and then specify and define service ontology depending on each service features.

3.2.2 User Preference Description Language

The User Preference Description Language (UPDL) is an OWL-based description language that allows users to describe their preference over service-specific ontologies. The goal of UPDL is to provide rich expressiveness in specifying items for user preferences and their weighting and to enable the sharing of such preference information. Our model can be satisfied with the use of ontology which specifies the concepts of corresponding domains and their relationship and supports the sharing of above information. UPDL can describe not only the preference for a specific service but the preference for other various services. So, it can deal with the preference for concepts used in any service dynamically, and enables the sharing of information by referencing ontologies.

Users need a way of flexible expression for user preferences to describe the interests concerning various services and items. The UPDL specifies the list for a user's various preferences and each preference separately. And because the relation between the preference list and each preference can be associated with the properties defined in UPDL, the flexible description of user's preferences is possible.

Figure 3.4 shows a simple example of user preferences for the paper retrieval service described based on UPDL and service-specific ontologies.

```
<updl:Profile rdf:ID="UserA">
    <updl:personalInfo rdf:resource="&person;UserA">
    <updl:hasPreference rdf:resource="#UserA_InterestingField1">
    <updl:hasPreference rdf:resource="#UserA_NotInterestingField1">
</updl:Profile>

<updl:Interesting rdf:ID="UserA_InterestingField">
    <updl:item rdf:resource="&papaer;SemanticWeb">
    <updl:weight>10</updl:weight>
<updl:Interesting>
```

Figure 3.4 A simple example of user preferences.

In the example, Profile class is used to describe the list of various preferences for a user who associated with user's personal information and each preference information by two properties explained below. However the details of real preference is described by the instances in the Preference class. User preferences are described based on UPDL ontology and heterogeneous service-specific ontologies each with their unique XML namespace.

- updl:personalInfo: This property is used to associate an instance of Profile class with an instances of person ontology which expresses the personal information. In figure 3.4, this property is used to associate Profile of 'UserA' with his/her instance of Person ontology (&person:UserA) modeled to describe as ontology-based information about the Person such as FOAF metadata or VCard ontology. The instance of Person ontology includes personal information such as id, name, and address [51][59].

- updl:hasPreference: This property is used to list various preference information and to associate corresponding instances of Preference class. The class represents detail specification of user preferences. In the example, this property is used to associate instances in the preference list with two instances of the preference class, "UserA_InterestingField1" and "UserA_NotInterestingField1".

Preference class describes the interesting information together with corresponding item and weight. Practically, a user preference can be described by two subclasses, Interesting and NotInteresting classes which describe the weights of items interested and not-interested respectively.

- updl:item: This property is used to associate an item with a concept which is hierarchically classified in the service-specific ontologies. This property can be used to associate the concepts described in various ontologies by using URI (Unified Resource Identification). Therefore, the property can represent classes defined in various services ontologies. In the example, this property associates "SemanticWeb" concept of "paepr" ontology (&paper) that classifies research field hierarchically.

- updl:weight: This property describes the weight of preference. In the example, the value "10" or "-10" is used as its weights for users firstly registered by users. These weights are computed through acquisition process for user preference. In the example, this property associates the weight for "&paepr:SemanticWeb" is "10".

- updl:hasCreateTime: This property describes the date when the preference was created firstly.

- updl:hasUpdateTime: This property describes the date when the preference was updated lastly.

- updl:hasLastAccessTime: This property describes the date when the preference was accessed lastly.

As we explained in this section, UPDL enables the description of user preferences using ontologies in various services and corresponding URI, and provides a way of flexible expansion of such preferences. Our user preference profiling provide flexibility and extensibility on expressiveness for user preferences, because users can reuse as well as describe a variety of preferences more detail. Hence, UPDL allows user preferences to be shared and represented dynamically. Although UPDL could provide only minimum restrictions, it provides user preferences with excellent flexibility and extensibility.

3.3 Sharing of User Preferences

In this section, we present the sharing mechanism of user preferences based on the data model for user preference described in Section 3.2.

3.3.1 The Sharing Mechanism

As we described in Section 3.2, the heterogeneous preferences of a user are described over various service-specific ontologies by UPDL, and those are stored in the user profile. In our model, each service specifies its service-specific ontology based on classification and features of its contents, and the contents of each service are described according to its service-specific ontology. Next, users describe their preferences over various service-specific ontologies using UPDL. These user preferences can be shared based on the hierarchical structure of the ontology and similarity between two ontologies. To put it more concretely, even if the concept of a user preference in user profile does not exist in the service-specific ontology of the current service, our model can recommend similar preferences by comparing upper concepts of the concept registered in user profile with the service-specific ontology of the current service.

However, since it is likely that the weights of recommended concepts is unequal to the original preference, it is necessary to evaluate weights of the recommended concepts over the service-specific ontology of the current service. An algorithm for similarity evaluation will be introduced in Section 3.3.2. The evaluated weight is used in personalized services as a factor for computing recommendation value.

3.3.2 Similarity Evaluation

The similarity evaluation is an algorithm that computes the evaluated weight from the distance between the original concept and the corresponding concept of higher level in service-specific ontologies of the user preference. The evaluated weight (EW) is the relative value of the selected concept in comparison with the weight of the concept registered in user preferences. When considering classes in an service-specific ontology, those that are lower in the hierarchy can be considered to be more specialized instances of those further up in the hierarchy. Thus, lower classes convey more detailed information and have more specific meaning. Consequently, as the higher concept is recommended, the evaluated weight of the recommended concept decreases. In our model, the evaluated weight of super class has the value of 50% less than that of current class.

Figure 3.5 depicts not only a class, HomeAppliance, but also various subclasses of it, where it is apparent that as the hierarchy is traversed from the top down, subclasses become more specialized than their super-classes. Figure 3.5 also shows the evaluated weight of each class in the hierarchy. This assignment of the weight based on the hierarchical classification has very important advantage that the assignment of the weight for the queries which is different from user's personal preference information is possible.

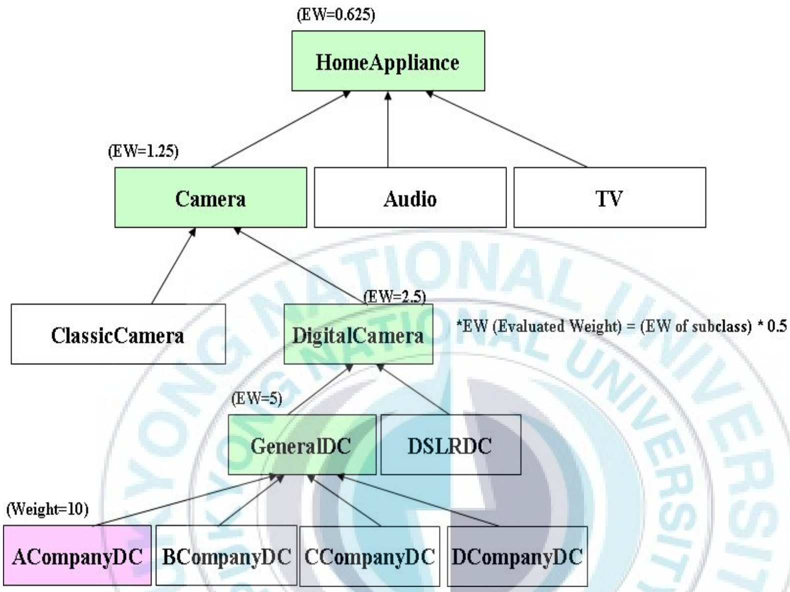


Figure 3.5 Similarity evaluation.

3.4 Discussion

In this chapter, we proposed user preference sharing model including an ontology-based data model for describing user preferences and its sharing mechanism based on Semantic Web. The most important aim of our model is to enhance both heterogeneity in describing user preferences and interoperability in sharing user preferences among various personalized services. An significant point in our model is that our model does not exclude current efforts such as metadata and standards developed in numerous domain, but provides the enhanced sharing mechanism for user preferences using these metadata and standards.

The data model consists of two kinds of ontologies: UPDL ontology to describe user preferences and service-specific ontologies to specify contents of their services. UPDL is especially an OWL-based description language that enable user preferences to be described over service-specific ontologies. Based on the data model, our model shares user preferences by computing similarity between two ontologies: the ontology registered in user profile and the service-specific ontology that the user use currently. We represented a way to compute evaluated weight based on the similarity in this chapter. Some of the main contributions of our model include:

- Heterogeneity in describing user preferences: Our user preference sharing model provides sufficient expressiveness. Although the existing personalized services specify user preferences concretely, they cannot share user preferences among various services. On the other hand, the current standards for user preference sharing cannot describe user preferences correctly. UPDL in our model provides heterogeneity for user preferences by enabling user preferences to be described over various service-specific ontologies.

- Interoperability in exchanging user preferences: Our model supports the sharing of personal preferences among various personalized services. As mentioned above, most of the existing personalized services cannot share user preferences among various services because they are designed as the stand-alone system. Compared with these models, our model provide strong interoperability in exchanging user preferences by computing similarity between service-specific ontologies. For example, if a term in the user profile is different from terms in the service-specific ontology of the current service, our model can recommend similar terms and their evaluated weights in the current service.

The UPDL also has other advantages based on the features of ontology. First, the UPDL is based on standard syntax and descriptions of meaningful user preferences because the model is based on OWL. Second, it provides rich expressiveness for encoding the meaning of user's preferences. Third, it is suitable to express semantic information for data fields because instances or real data of applications are covered as meta data. Also, it allows users to describe concepts into its domain parts because the UPDL is structured as hierarchy concepts. The UPDL enables users to describe semantically his/her preferences in particular domains. The UPDL is well suited to become description language because it has well-formed structure, rich expressiveness, and flexible description.

Chapter 4

User Preference Sharing Framework

This chapter describes a framework for user preference sharing. In the chapter 3, we describe the user preference sharing model including basic concept, description model, and sharing mechanism. Although the model is more heterogeneous and interoperable than current personalized services, it is complex in describing and processing user preferences because of RDF-based profiling unfamiliar to users. Therefore, the middleware to process these complex tasks is necessary. In this chapter, we describe the architecture of the framework for user preference sharing and the middleware to support the framework.

4.1 Overview of the Framework

The User Preference Sharing Framework enhances interoperability among heterogeneous personalized services by sharing user preferences. The framework is organized based on the 3-tiered architecture as shown in Figure 4.1. An advantage of 3-tiered architecture is to provide system with independence among data, process, and application. That is to say, middleware enables applications to be developed regardless of data model and not to be affected by modifying data model.

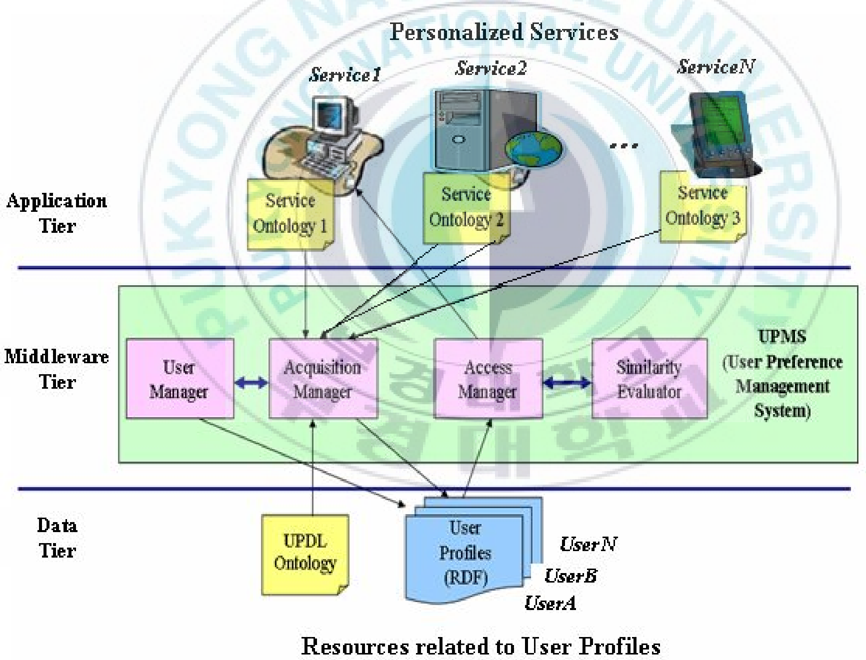


Figure 4.1 Architecture of user preference sharing framework.

In application tier, service-specific ontologies are defined as adaptive features for each service, and contents are published based on the ontologies. In data tier, users describe their preferences over service-specific ontologies using UPDL to share the information semantically. As described in Section 3.2, the UPDL enables user preferences to be described over service-specific ontologies of various services. The middleware tier gives a connection between data tier and application tier, and it takes charge of describing and sharing the user preference in place of applications.

Although our model described in Chapter 3 provides rich expressiveness in describing user preferences and interoperability for user preferences between various services, it is complex to describe and process user preferences in our model due to RDF-based profiling. To resolve these problems, our framework provides the User Preference Management System (UPMS) as a middleware. The UPMS also provides easy development environments to develop applications regardless of data model and complicated processing methods. It is possible by providing the comfortable user interface (UI) to register their preferences and programming interface (API) to share their preference. In addition, the middleware tier protects user information from external access through preventing application from directly accessing the information.

The User Preference Management System (UPMS) consists of user manager, acquisition manager, access manager, and similarity evaluator. The user manager plays the role of the management for new users, including creation of personal information and initialization of the user profile. The acquisition manager provides users interface that enable users to create and update their preferences in the profile based on UPDL and heterogeneous service-specific ontologies. The access manager provides API (Application Programming Interface) that enable various personalized services to access users' profile. The similarity evaluator is used by the access manager, and it computes the relative weights evaluated based on similarity between the original concept and the corresponding one of the requested service. The evaluated weight is used in recommendation services in each personalized service as a factor for computing recommendation value.

Each manager in UPMS is implemented using Jena Toolkit, Joseki, and RDQL[66]. Jena API is a Java application programming interface that creates and manipulates RDF documents, and Joseki is a Java client and a server that implemented the Jena network API over HTTP[50][51]. In addition, we can semantically search the instances of RDF documents through RDQL, a query language for RDF, which is Jena's query language [66].

4.2 User Management

User Manager of UPMS takes charges of creation of personal information and initialization of user profile to register a user and create new user profile of the user.

The first function of user manager is to create personal information. Although UPMS needs only URI to user information to identify a user, our system requires several basic information about user for the additional management in the future. Figure 4.2 shows the user interfaces used in UMPS User Manager. The user manager provides two ways to register user information in our system (Figure 4.2(a)). The first way is that the user creates it in person and registers the URI to the information. With the second way, the user can register own information using the creation module of user information provided by user manager. Figure 4.2(b) shows the interface to register user information, and Figure 4.2(c) shows an example of the created user information by UPMS user manager. To describe user information, a few specifications such as VCard and FOAF(Friend-of-a-Friend) can be used. As growth of many information on the web, personal information has been utilized on the environments. This information benefits connection among person as well as many other available information on the web. Thus, there have been the researches on the specifications to support the information.

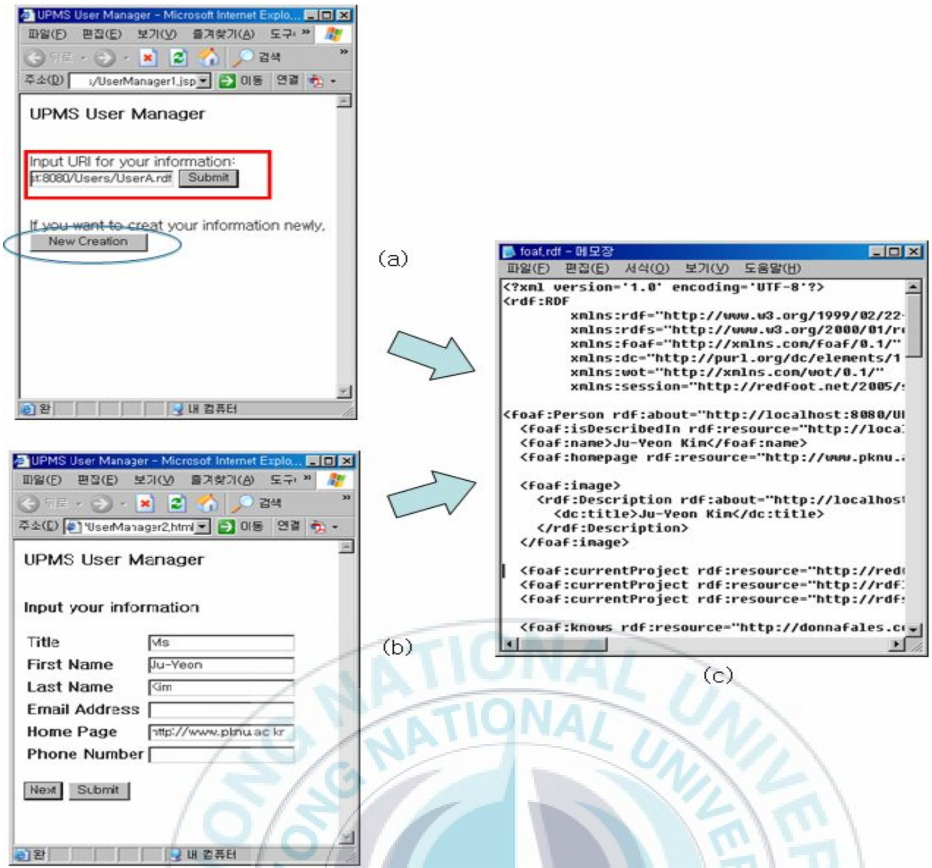


Figure 4.2 The user manager of UPMS. (a) The first registration way to user information. (b) The second registration way to user information. (c) The created user information.

In our system, the FOAF is used to describe user information. The FOAF Specification[9] to describe metadata about personal information is a way of describing a network of friends and others, and providing affiliation and other social information

9) http://xmlns.com/foaf/0.1/

about individual user is used. The FOAF vocabulary defines some useful constructs that can appear in FOAF files. For example, FOAF defines categories ('classes') such as foaf:Person, foaf:Document, foaf:Image, alongside some handy properties of those things, such as foaf:name, foaf:mbox (ie. an internet mailbox), foaf:homepage etc., as well as some useful kinds of relationship that hold between members of these categories. According to the registration form of user information as shown in Figure 4.2, users can enter themselves basic information, and the input values then are automatically stored in personal information file based on RDF.

The second function of the user manager is to create a user profile of a new user with the user information created by either the user or user management. The user profile initialized by user management is updated by acquisition management that will be described in next section.

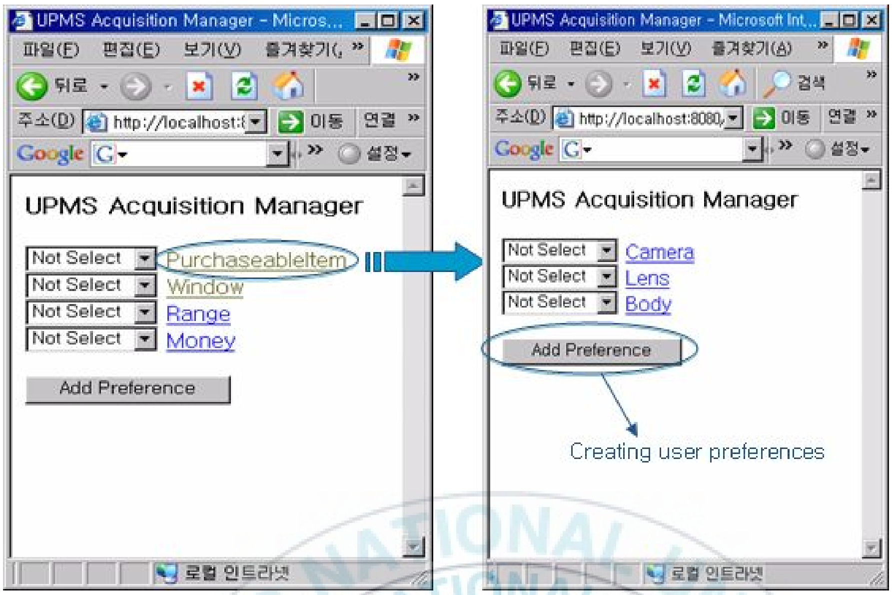
4.3 Acquisition Management

Personalization systems have to acquire certain knowledge about the user preferences. The task is performed in the acquisition management of User Preference Management System (UPMS), and it consists of the acquisition module and the creation module of user preferences. The user preferences acquired by acquisition management are stored in user profile.

4.3.1 Acquisition of User Preferences

The user profile acquisition is typically divided into implicit and explicit methods. Implicit profiling often lacks in accuracy and reliability, because it cannot be made transparent to the user for corrections. On the other hand, explicit user profiling is subject to enter information directly into the system. For instance, the systems generally obtain by filling in questions. Although these methods in general lead to more reliable profiles, the problem is that users can become overstrained by filling in large forms or rating hundreds of items.

To resolve the problem, the structured model for user preferences that spans various items is necessary, and it need to be shared between different services. In our approach, the service-specific ontologies are used as the model to specify user preferences and UPDL is used to describe user preferences over the ontologies as described in Section 3.2. To acquire user preferences, the acquisition manager provides the Web-based user interface as shown in Figure 4.3. The interface enables users to easily and comfortably register their preferences by providing hierarchical access to items of each service based on service-specific ontologies.



(a) (b)

Figure 4.3 The acquisition manager of UPMS.

To collect user's interest, the concepts classified hierarchically by service-specific ontologies are provided through user interface as shown in Figure 4.3 (a). Once a user selects a preference concept, the UPMS acquisition manager displays more specific terms by browsing concepts in the subclasses as shown in Figure 4.3 (b). And then the user can select the closest preference level to his/her preference information. In selecting the preferences, the acquisition manager provides two selections based on binary level of interest - positive interest and negative interest. Introducing two interest levels of Interesting and Not-interesting with the analysis of user preference information can solve the ambiguous selection problem in the system using multi-level of preference. The selected items is described in user profile with the corresponding weights.

This approach has advantages as follows. First, users can obtain knowledge on services without basic concepts or knowledge because contents of services are provided automatically. Second, the interface has directly access contents of services without processing or reading a user's profile. The UPMS enables a variety of services or systems to share user preferences through user interface. Finally, the acquisition manager also provides the same form of browsing interface for all services. It has an advantage that collection of user preference information in various services. It also is possible in the process of user preference acquisition without modifying the program for user interface.

4.3.2 Creation of User Preferences

User preferences acquired by acquisition module is stored in user profile written in RDF form as shown in Figure 4.4. A preference in the user profile is represented as a resource that is the conceptual mapping to an entity or set of entities, and it identified by a specific URI. The corresponding concept and its weight of a preference are described by statements that is defined to be a triple consisting of a subject, a predicate, and an object. These knowledges are represented based on the User Preference Description Language (UPDL) described in Section 3.2. The acquisition manager performs these generation processes automatically without user's recognition.


```
<rdf:RDF
    xmlns:updl="http://localhost:8080/UPMS/Ontologies/updl.owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:base="http://localhost:8080/UPMS/Profile/UserA.rdf#" >
  <rdf:Description rdf:about="http://localhost:8080/UPMS/Profile/UserA
    <updl:hasWeight>10</updl:hasWeight>
    <updl:hasItem>Digital</updl:hasItem>
    <updl:hasItemOnt>http://localhost:8080/UPMS/Ontologies/camera.owl<
    <rdf:type rdf:resource="http://localhost:8080/UPMS/Ontologies/updl
  </rdf:Description>
  <rdf:Description rdf:about="http://localhost:8080/UPMS/Profile/UserA
    <updl:hasWeight>10</updl:hasWeight>
    <updl:hasItem>Body</updl:hasItem>
    <updl:hasItemOnt>http://localhost:8080/UPMS/Ontologies/camera.owl<
    <rdf:type rdf:resource="http://localhost:8080/UPMS/Ontologies/updl
  </rdf:Description>
  <rdf:Description rdf:about="http://localhost:8080/UPMS/Profile/UserA
    <updl:hasWeight>10</updl:hasWeight>
    <updl:hasItem>Camera</updl:hasItem>
    <updl:hasItemOnt>http://localhost:8080/UPMS/Ontologies/camera.owl<
    <rdf:type rdf:resource="http://localhost:8080/UPMS/Ontologies/updl
  </rdf:Description>
  <rdf:Description rdf:about="http://localhost:8080/UPMS/Profile/UserA
    <updl:hasWeight>-10</updl:hasWeight>
    <updl:hasItem>Lens</updl:hasItem>
    <updl:hasItemOnt>http://localhost:8080/UPMS/Ontologies/camera.owl<
    <rdf:type rdf:resource="http://localhost:8080/UPMS/Ontologies/updl
  </rdf:Description>
  <rdf:Description rdf:about="http://localhost:8080/UPMS/Profile/UserA
    <rdf:type rdf:resource="http://localhost:8080/UPMS/Ontologies/updl
    <updl:hasItemOnt>http://localhost:8080/UPMS/Ontologies/camera.owl<
    <updl:hasItem>Large-Format</updl:hasItem>
    <updl:hasWeight>10</updl:hasWeight>
  </rdf:Description>
  <rdf:Description rdf:about="http://localhost:8080/UPMS/Profile/UserA
    <rdf:type rdf:resource="http://localhost:8080/UPMS/Ontologies/updl
```


Figure 4.4 A User profile created by the acquisition manager.

The concept of an RDF graph as shown in Figure 4.5 is used in the actual processes. A user preference consists of specific collections of statements, and it is dealt as the concept of a graph for fast searching. In processing the RDF statements of user preferences, it is necessary to name with URIs of a specific concept of a service-specific ontology and UPDL ontology. That is to say, the acquisition manager has to manipulate specific named ontologies. The Figure 4.5 shows the graphical view of userA's preferences through validation service of W3C that supports elements and attributes of the standard RDF model and syntax specification. It displays a triple representation of the corresponding data model as well as an optional graphical visualization of the data model.

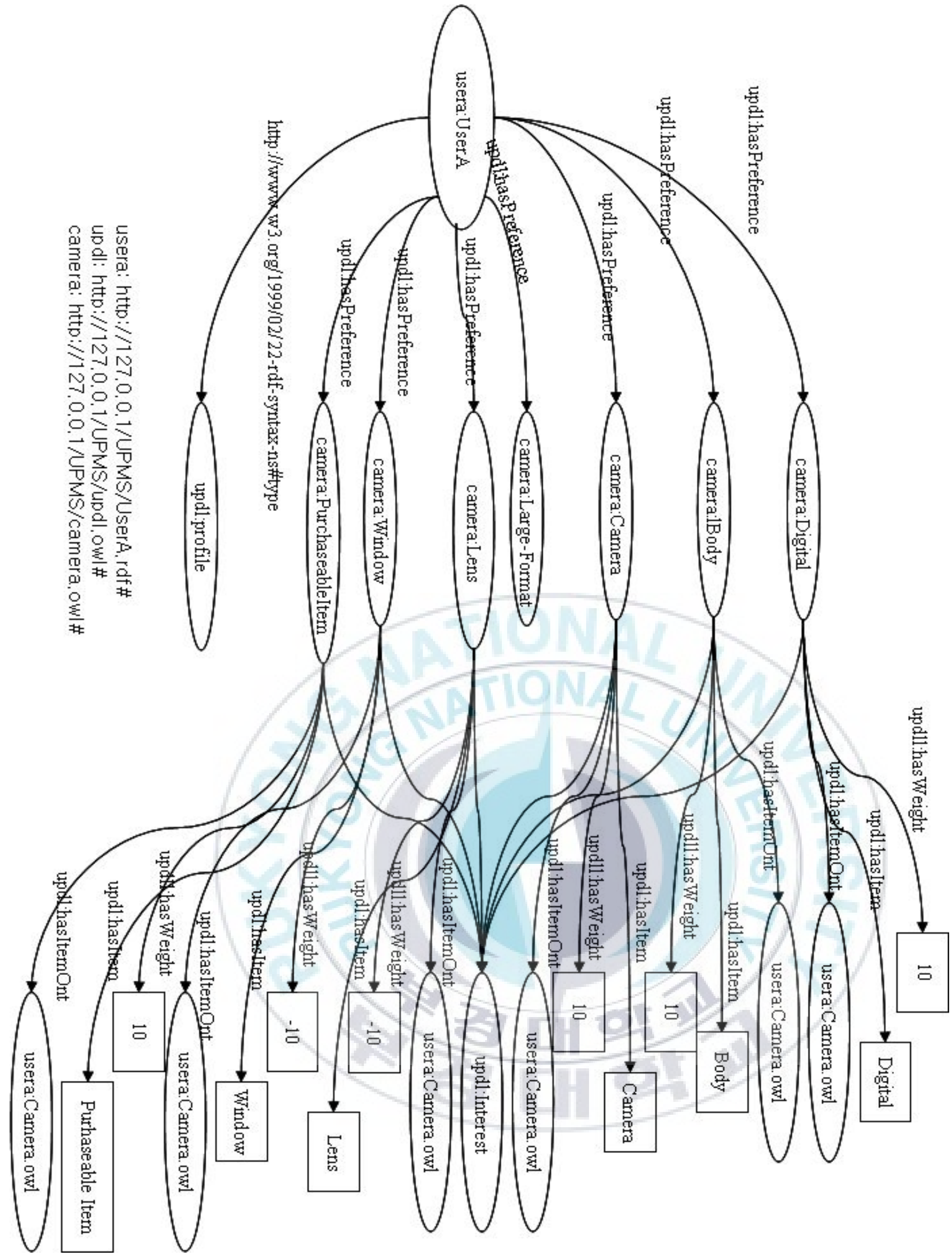


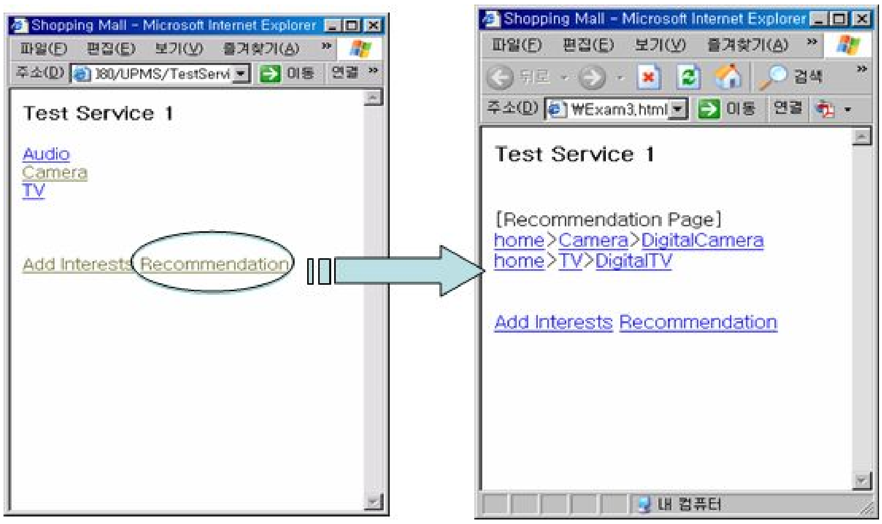
Figure 4.5 Validation of automatically created user profile for UserA.

4.4 Access Management

To access to user preferences, User Preference Management System (UPMS) provides two functions - Access Manager and Similarity Evaluator.

4.4.1 Access to User Preferences

Access Manager of UPMS provides heterogenous personalized services with a programming interface to access to user preferences. Figure 4.6 shows that a personalized service uses the Access Manager to access to user preferences. First, the personalized service provides the Access Manager with own service-specific ontology and requests the Access Manager to return user preferences (Figure 4.6 (a)). Next, Access Manager reads user profile, and it then evaluates user preferences over current service ontology through the Similarity Evaluator of User Preference Management System. Finally, the Access Manager return preferences and weight evaluated over service ontology of the requested service. Figure 4.6 (b) shows a result page recommended by the requested service using the evaluated preferences and weights returned from the UPMS Access Manager.



(a) connection to UPMS. (b) recommended results.

Figure 4.6 UPMS access manager.

4.4.2 Similarity Evaluator

The similarity evaluator is used by the access manager, and it evaluates the relative weight of each preference on the service ontology of the requested service. The similarity is computed by the distance between the concept of requested service ontology and original concept of the user preference as described in Section 3.3. The similarity evaluator computes similarity from distance of corresponding concept of higher level in service-specific ontologies. The value is used in recommendation services in each personalized service as a factor for computing recommendation value.

For items in the "Interest" class, the weight value "10" is assigned and then super classes receive the weight value of 50% less than that. For items in the Not-Interest class, the weight value "-10" is assigned. This assignment of the weight based on the hierarchical classification has very important advantage that the assignment of the weight for the queries which is different from user's personal preference information is possible. Figure 4.7 shows the comparison results between the original preferences/ weights in the user profile and the evaluated preferences/weights computed by the similarity evaluator.

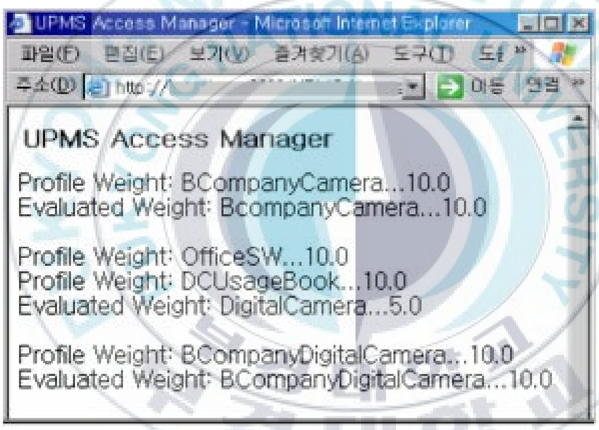


Figure 4.7 The result of the similarity evaluation.

4.5 Security Considerations

Personalized services can offer much convenience to users as introduced in previous chapters. However, the personalized services also involve the threat of an invasion of privacy and security because they should manage the personal information such as user preferences. This section introduces the security consideration in our framework.

4.5.1 Overview of Security in Our Framework

Security is to prevent and detect unauthorized use on information. The personal information like the user preferences is very sensitive information. In a system that involves personal information, it appears to be the best idea to create and store profiles locally like User Preference Management System (UPMS) we proposed. However, considerations for network security is necessary because the user preferences are shared among a number of services through network in our framework. Therefore, the framework has to provide protection as components like authentication, confidentiality, integrity, non-repudiation.

The UPMS has to provide authentication service to validate identity. Mutual authentication means proving the identity of both parties involved in communication, and this is done using special security protocols. In our approach, the authentication is vexatious because users access to UPMS through various services. Although trust among systems is used to protect the problem and sniffing of id/password, it is weak on IP spoofing attack. AS an alternative mechanism, Single Sign On (SSO) that allows the user to only authenticate once to user's client. More details on SSO is introduced in next section.

Authorization means determining a user's permissions. UPMS has to provide mechanisms for finding a user's permissions and roles such as access control lists. A number of systems for personalization are regulated security policy. A security policy is a set of specifications for the processing of the data. A flexible definition of policy serves two purposes. First it enables users to adjust their preferences regarding privacy and to make an informed decision about the use of a user adaptive system. Second, developers of user personalized services are able to gain with user demands regarding privacy and to develop systems that are more user-oriented.

Confidentiality means that information open only by permitted times and methods. When sensitive information such as user preferences is transmitted, keeping it secret is important. It is common practice to satisfy confidentiality requirements with encryption. Integrity means that information received exactly as sent by an authorized entity. In a network, making sure data has not been altered in transit is imperative. Validating a message's integrity means using techniques that prove that data has not been altered in transit. Usually, techniques such as hash codes and MAC (Message Authentication Codes) are used for this purpose. Non-repudiation means that Prevents either sender or receiver from denying a transmitted message. The process of proving legally that a user has performed a transaction is called non-repudiation. Using digital signatures provides this capability. The SSL (Secure Socket Layer) appears to be one of the best selection for these security services because our framework is designed based on HTTP[51].

4.5.2 Single Sign-on

SSO is a technical mechanism that allows the users to only authenticate once to their client, so that they do not have to memorize many user names and passwords of them for other server applications. It has been receiving much attention from many enterprises due to the user convenience through a single authentication.

Figure 4.8 illustrate the general authentication mechanism that a client application connects to the distributed services. For authentication, the id/password of each service is necessary.

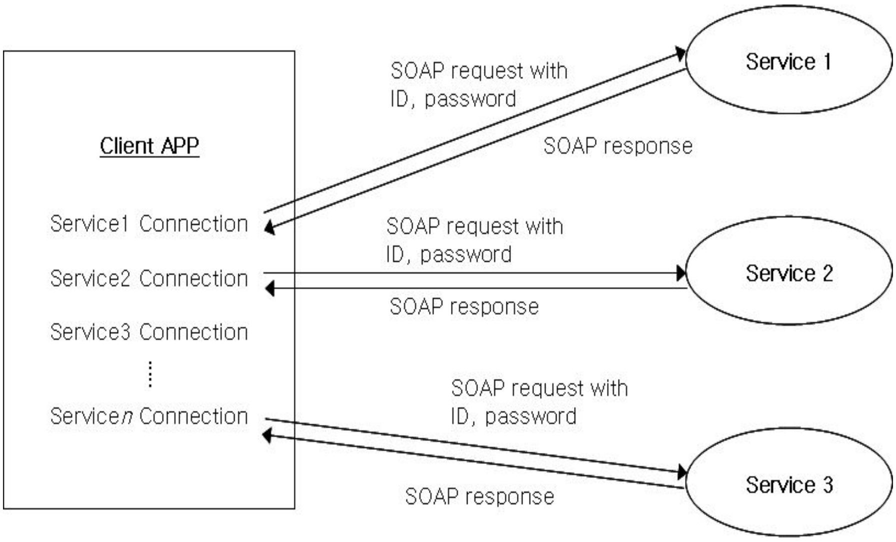


Figure 4.8 The general authentication mechanism.

There are several advantages acquired by using SSO. First of all, aspects of users guarantee security from threats such as hacking of personal information. Existing authentication processes should repeat input user's identification number and his/her password. Next, aspects of managers of systems are able to guarantee security high by controlling access centralized. Also, the managers can process easily creation, deletion of user's number because of intergrated login. In ID and Password of SSO, the password which input by a user didn't transfer through network, and safe from re-attracts because ensured ticket are used[9].

Figure 4.9 show each stage to process certification between Client and Server.

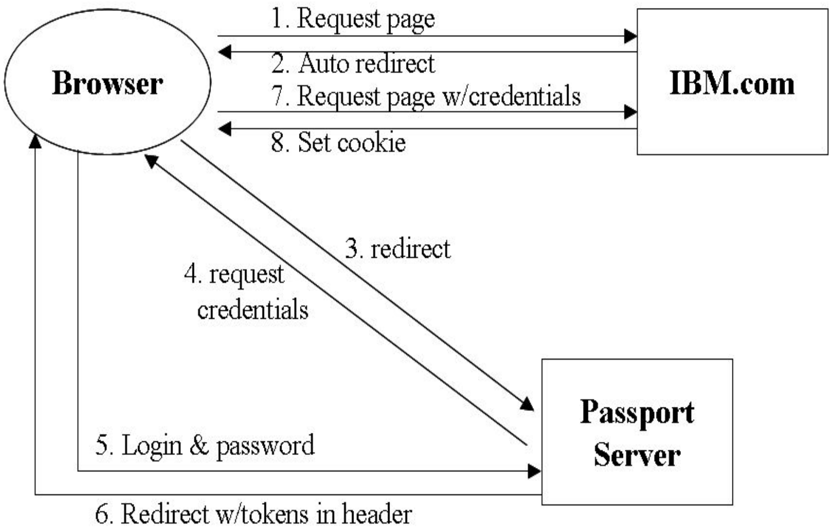


Figure 4.9 Single Sign-On mechanism between client and server.

The process each stage as follow.

(1) Client browser requests a page in web site of IBM.

(2) IBM server is redirected automatically in client browser.

(3) A browser redirects to passport server.

(4) The passport server request credentials.

(5) The client browser sends his/her login and password to passport server.

(6) The passport server redirects tokens in header, and then send the tokens to client browser.

(7) The IBM server requests signed credentials within the page.

(8) The IBM server sends cookies.

In these days, there are several technologies enablers for SSO, including Kerberos, Secure Assertion Markup Language (SAML), and other cryptographic protocols[9]. As one of SSO technologies, Kerberos is a network authentication protocol. It is created by MIT in order to provide authentication of applications by using secret-key cryptography. The Kerberos protocol can prove its identity to a server or to a client across by integrating a session encrypted. Next, there is a Security Assertion Markup Language (SAML) as an authentication of SSO technologies. It is an XML standard to identify authentication among providers. It is also known as a product of the OASIS Security Services Technical Committee.

4.5.3 Authentication in Our Framework

In our framework, the authentication is vexatious because users access to UPMS through various services. For authentication, our framework can be use SSO introduced in previous section. Figure 4.10 shows the authentication mechanism using SSO in our framework. UPMS provides KDC (Key Distribution Center) for authentication and ticket granting.

(1) Service providers register their services with information for authentication to KDC in UPMS.

(2) Users subscribes to KDC in UPMS with information for authentication.

(3) The Authentication Service of KDC in UPMS authenticates a user using Kerberos protocol with public key.

(4) The TicketGranting Service of KDC in UPMS issues the ticket for each service.

(5) The user access to a service with ticket for the service.

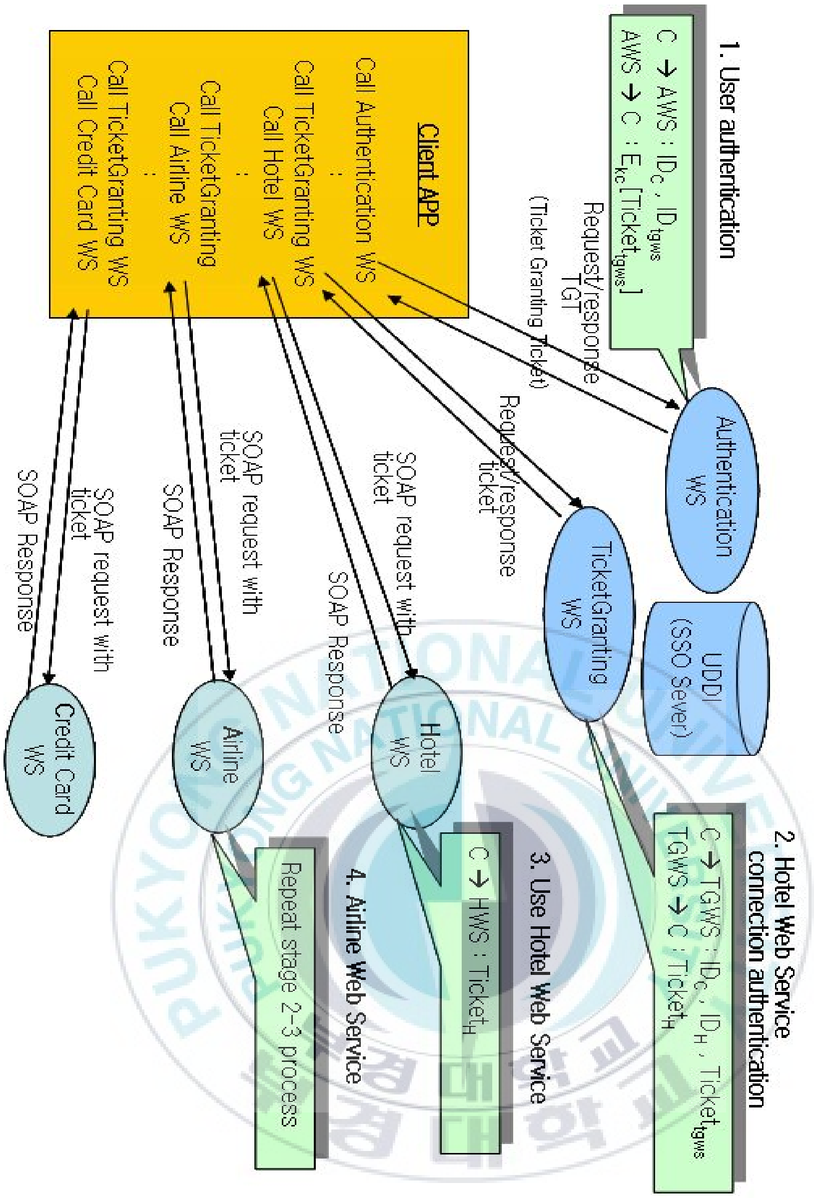


Figure 4.10 An authentication by SSO.

4.6 Discussion

In this chapter, we introduced the framework to support our user preference sharing model described in Chapter 3. The framework is based on 3-tiered architecture, and User Preference Management System (UPMS) as the middleware of the framework provides the management services to process the complex tasks for user preferences. First, the user manager of UMPS manages creation of personal information and initialization of the user profile. Second, the acquisition manager creates and updates user preferences based on UPDL and heterogeneous service-specific ontologies. Third, the access manager provides API to access users' profile, and similarity evaluator used by the access manager computes the evaluated concepts and weights based on similarity evaluation. Finally, The evaluated weight is used in recommendation services in each personalized service as a factor for computing recommendation value.

The framework that we proposed has the following advantages. First, because the framework was designed to support our user preference sharing model described in Chapter 3, it provides heterogeneity in describing user preferences and interoperability in exchanging user preferences as the benefits of the user preference sharing model. Second, the framework provides flexibility in modifying the classifications and features of services using our

UPMS, because the service-specific ontologies are easy to modify as the feature of ontology, and they are referred by similarity evaluator in UPMS in execution time. Lastly, the framework provides users and service providers with accessibility by capsulizing the complex tasks for the user preference management. That is to say, it is possible for users to register their preferences without the knowledge of RDF and for each service to access the evaluated preferences and weights adapted to each service without the knowledge of the similarity evaluation.

Chapter 5

Application and Evaluation

In this chapter, we realize an example application based on our user preference sharing framework, and evaluate our framework under various conditions. Then, we analyze the evaluated results.

5.1 An Example Application

In this section, we realize five prototype services with each service-specific ontology in the area of research information services. First, overview of the application including scenarios and system architecture is introduced. Next, we describe the service-specific ontologies for our application. Then, recommendation in personalized services using UPMS is described.

5.1.1 Overview of the Application

The purpose of Personalized Research Information Services using UPMS is to share the user preferences about research interest of each user among various services. In current services, a user should register own preferences in each service, and then personalized contents are recommended to the user. However, personalized applications in the distributed environments are required to exchange user information among various services. The two typical example scenarios in the area of research of user preference sharing are belows.

- Example Scenario 1. The UserA usually uses a peper search service to find her/his interesting paper about the field of 'ontology'. Now, if the user wants to use the online bookstore service, the service can recommend the links of concepts related to 'ontology' by sharing user preferences specified in the paper search service.

- Example Scenario 2. The UserB is going to submit a paper to the conference of HCI international, and she/he register 'interoperabiltiy' as her/his interesting field to the HCI international web site. Then, if the user wants to find the related works in her/his interesting field with the keyword, 'ontology', the paper search service can recommend the papers related to interoperability within the papers related to ontology.

Figure 5.1 shows the architecture of the Personalized Research Information Services to satisfy the above example scenarios. As described in Chapter 4, users using each personalized service can register their preferences with the service-specific ontology of the current service by accessing to Acquisition Manager of UPMS. Then, each service can acquire the user preferences and weight evaluated according to the service-specific ontology of the requested service by Access Manager and Similarity Evaluator of UPMS. Each service of example application provides two type of recommendation - personalized browsing and personalized search. The personalized browsing is to provide users with links to the recommended concepts based on user preferences like the example scenario 1. The personalized search is to provide users with the ordered contents by ranking based on user preferences as the example scenario 2.

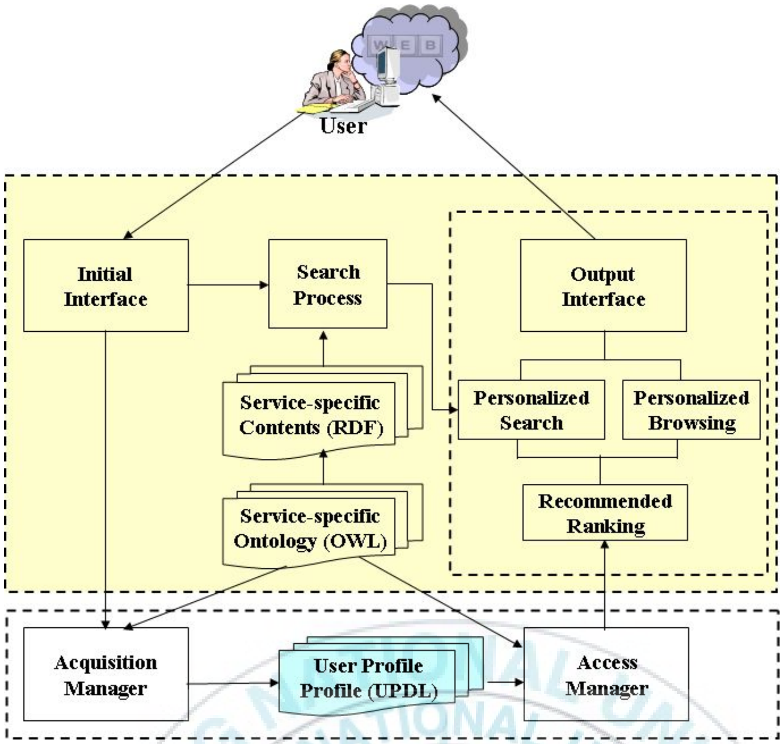


Figure 5.1 The architecture of the personal research information services.

Figure 5.2 shows the user interface of the paper retrieval service. The interface provides two general functions - general browsing based on its service-specific ontology and general search based on keyword. In addition to the general functions, each service provides three functions for the personalized services - the registration of user preferences based on its service-specific ontology, the personalized browsing based on the evaluated preferences from UPMS, and the personalized search based on the keyword and the evaluated preferences from UPMS.

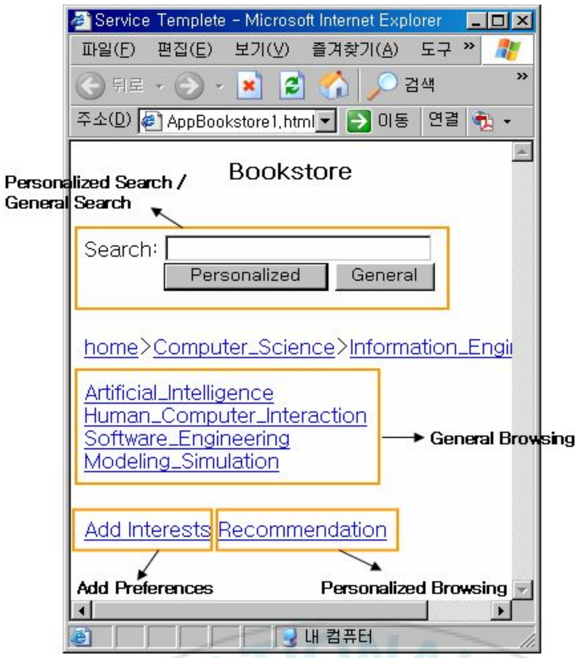


Figure 5.2 The user interface of an example personalized service.

5.1.2 Creation of Service-specific Ontologies

For our example application, we specify a upper ontology for research field and five service-specific ontologies for application services. Our framework requires service-specific ontologies to describe user preferences as described in Chapter 3, and using the upper ontology in specifying the service-specific ontologies enhances the performance of our framework in sharing user preferences as described in Section 3.2. For the application, we define the upper ontology for research field based on metadata of ACM Computing Classification System, and we then specify the service-specific ontology for each service based on the upper ontology we defined.

Figure 5.3 shows the upper ontology based on ACM Computing Classification System used in our application. The full classification scheme of ACM Computing Classification System (CCS) involves three concepts: the four-level tree, general terms, and implicit subject descriptors.

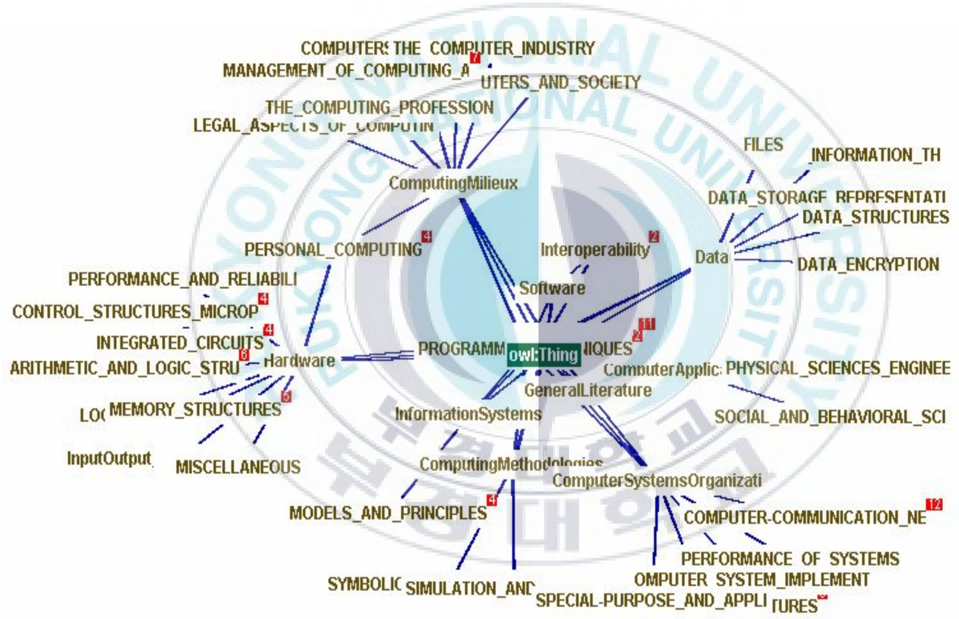
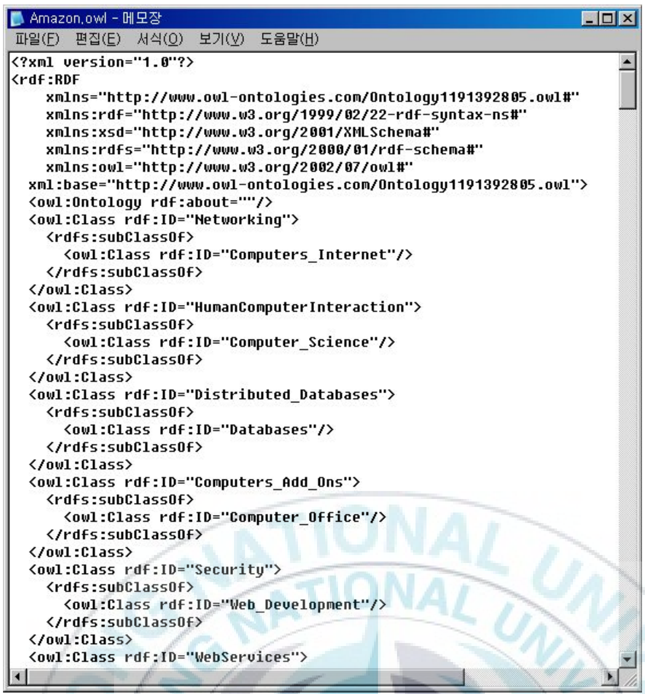


Figure 5.3 The upper ontology based on ACM computing classification system.

Figure 5.4 shows an example of Service-specific ontology based on the upper ontology we defined.




```
<?xml version="1.0"?>
<rdf:RDF
    xmlns="http://www.owl-ontologies.com/Ontology1191392805.owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.owl-ontologies.com/Ontology1191392805.owl">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="Networking">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Computers_Internet"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="HumanComputerInteraction">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Computer_Science"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Distributed_Databases">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Databases"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Computers_Add_Ons">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Computer_Office"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Security">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Web_Development"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="WebServices">
```

Figure 5.4 An example of the service-specific ontology generated based on the upper ontology.

5.1.3 Recommendation in Personalized Services

In our example application, each service to take an advantage of UMPS described in Chapter 4 provides the registration process of user preferences and recommendation process. For the registration process of user preferences, each service provides the link to Acquisition Manager of UPMS as shown in Figure 5.5. As the result of this process, the user preferences are described by

UPDL and appended to the user profile.

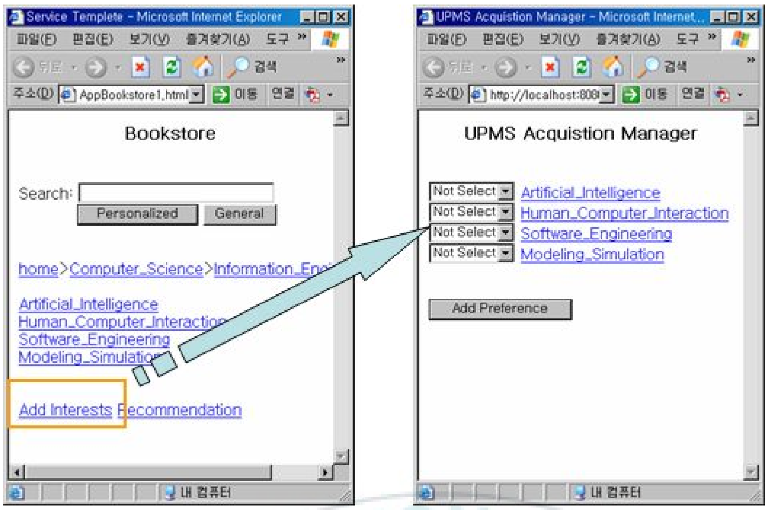


Figure 5.5 An example of access to UPMS.

In the recommendation process of our application, each service provides two types of personalization - personalized browsing and personalized search. To rank the concepts and contents, the two personalizations use the evaluated concepts and their evaluated weights acquired from the Access Manager of UPMS. Compared with the general browsing that provides the simple access to contents of the service based on the service-specific ontology (Figure 5.6), the personalized browsing provides the user-adaptive links based on the user preferences shared among various services by UPMS as shown in Figure 5.7.

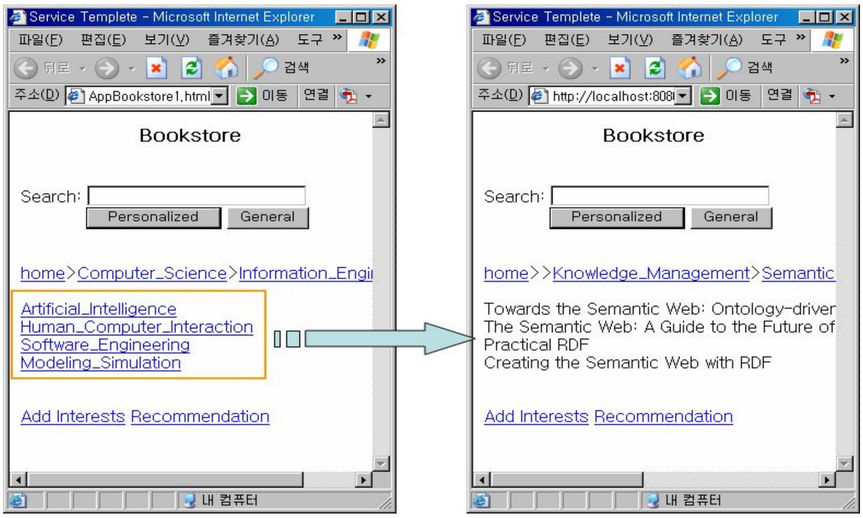


Figure 5.6 The general browsing.

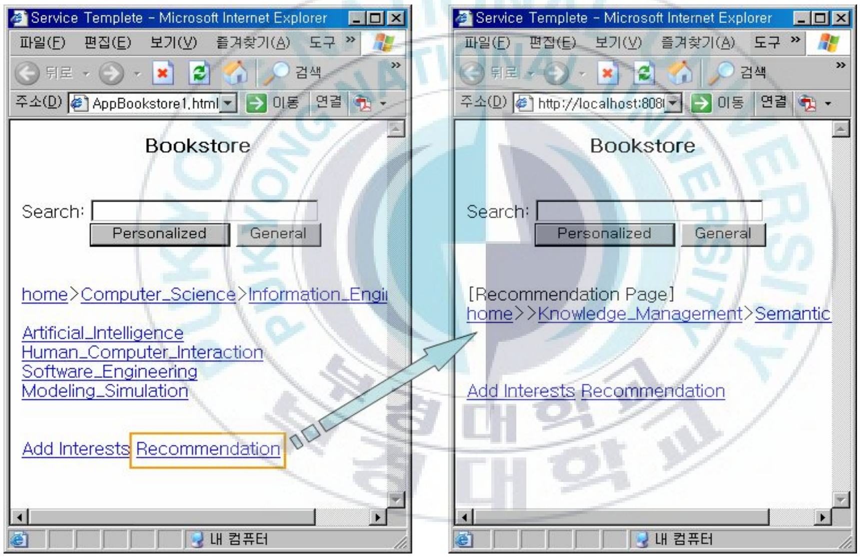


Figure 5.7 The personalized browsing using UPMS

The general search provides the results of the keyword-based search as shown in Figure 5.8. On the other hand, the personalized search provides the user-adaptive contents with ranking of the results of the general search based on the user preferences shared by UPMS as shown in Figure 5.9.

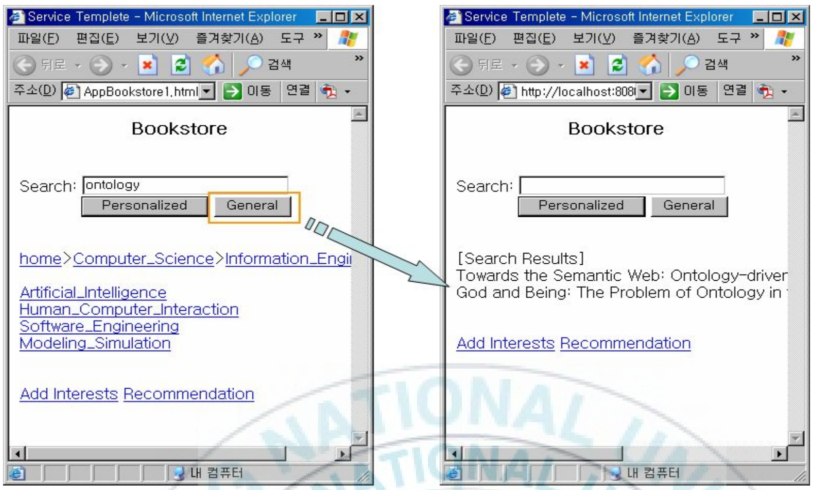


Figure 5.8 The general search.

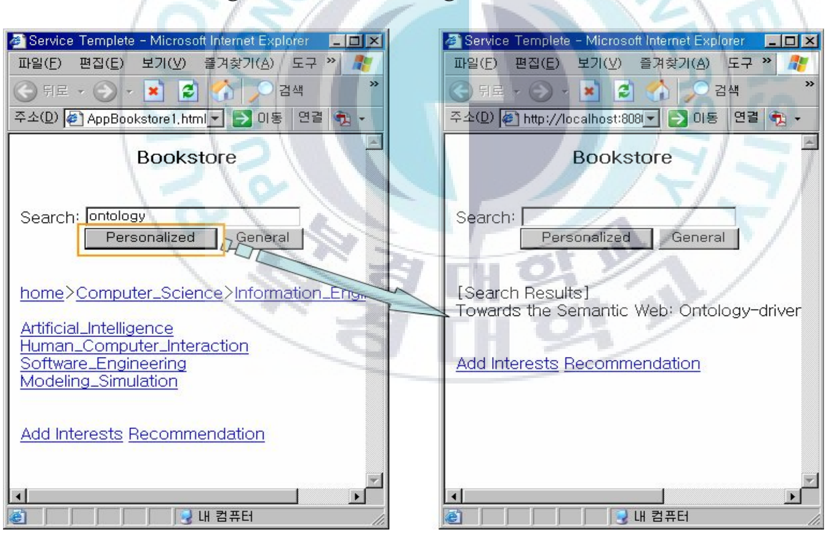


Figure 5.9 The personalized search using UPMS

5.2 Evaluation

In this section, we evaluate heterogeneity/interoperability of our model in exchanging user preferences and precision/recall in recommending user-adaptive contents. For evaluation, we realized five prototype services with each service-specific ontology in the area of research information services as listed in Table 5.1. The services for the evaluation consist of two online bookstores, two paper search services, and a conference service. Then, we registered two user preferences for each service using Acquisition Manager of UPMS described in Section 4.3.

Table 5.1 The services for evaluation.

Services	Service1	Service2	Service3	Service4	Service5
Depth of ontology	5	5	5	5	5
Total Concepts	32	37	33	41	47
Leaf Node Concepts	17	21	18	23	27
Instances	40	40	40	50	50

5.2.1 Evaluation of Our Model

Heterogeneity in current Web with various and distributed services are an important factor because the services treat heterogeneous contents with various classifications and features. Generally, it is difficult to improve interoperability under ensuring heterogeneity of each service due to interoperability-heterogeneity trade-off. Nevertheless, we have tried to overcome the issue by sharing user preferences semantically using the concept of ontology and similarity evaluation. In this section, we evaluate interoperability under ensuring heterogeneity of each service.

Table 5.2 compares the number of available preferences of stand-alone model and our model. In the stand-alone model, the user preferences are used within the only local service because users are permitted to register their preferences within the terms of a service, and the preferences cannot be shared in other services. In contrast, our model can share the user preferences with other services.

Table 5.2 The number of available preferences in each service.

Services	Service1	Service2	Service3	Service4	Service5	Average
No. of available user preferences in stand-alone model	2	2	2	2	2	2
No. of available user preferences with evaluated weight above 1 in our model	8	7	8	6	7	7.2
No. of available user preferences with evaluated weight above 2 in our model	6	6	7	4	5	5.6
No. of available user preferences with evaluated weight above 5 in our model	3	4	4	2	3	3.2

In Table 5.2, the number of the available preferences in our model is measured in cases of the evaluated weight 1, 2, and 5. Table 5.2 shows that the stand-alone model has the same number of preferences as '2' because user's preferences are managed within the system regardless of the number of the services. On the other hand, 10 user preferences can be used in our model because the preferences are shared among all of the services.

However, the number of the available user preferences is from 2 to 10 because the semantically corresponding preferences can exist. Allowing that a preference can be used in current service, only the preference with the weight over the critical value is significant. In the table, the average number of the available user preferences is 7.2 for the evaluated weight over 1, and the average number is 5.6 for the weight over 2. In case of the evaluated weight over 5, the number of available user preferences is 3.2. The result shows that our approach is more interoperable then stand-alone model in all cases.

Figure 5.10 shows the number of user preferences as the number of services. As the number of services increase, the number of available preferences also increase in our approach. Consequently, we can expect that the efficiency of our approach rapidly increase if much more services are attended.

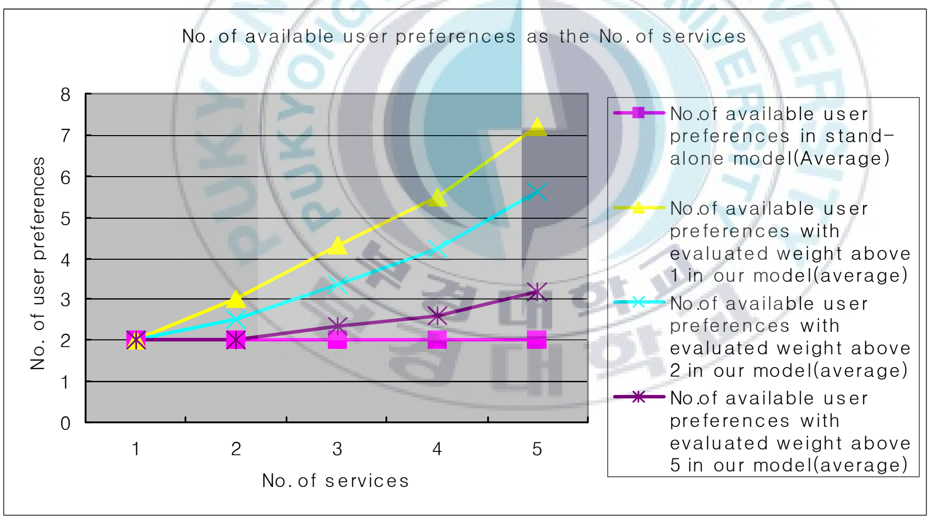


Figure 5.10 Variation of available preferences according to increase of participate services.

5.2.2 Evaluation in Personalized Services

To evaluate the performance our framework in personalized services, we measure the average number of clicks to target content in browsing services and recall/precision of the retrieved contents in search services.

Figure 5.11 shows the number of clicks to target content that a user wants to find. In the general browsing, the user must browse directories from the root directory to target directory containing the target content as the hierarchy of classification of the service. Therefore, the average number of clicks to target content in general browsing becomes the average depth of the service-specific ontology irrespective of the number of services. Compared with the general browsing, the personalized browsing recommends directories related to user preferences based on our user preference sharing framework. As a result, the average number of clicks to the target content is less than the average depth of the service-specific ontology, and it is more efficient as the number of services increases.

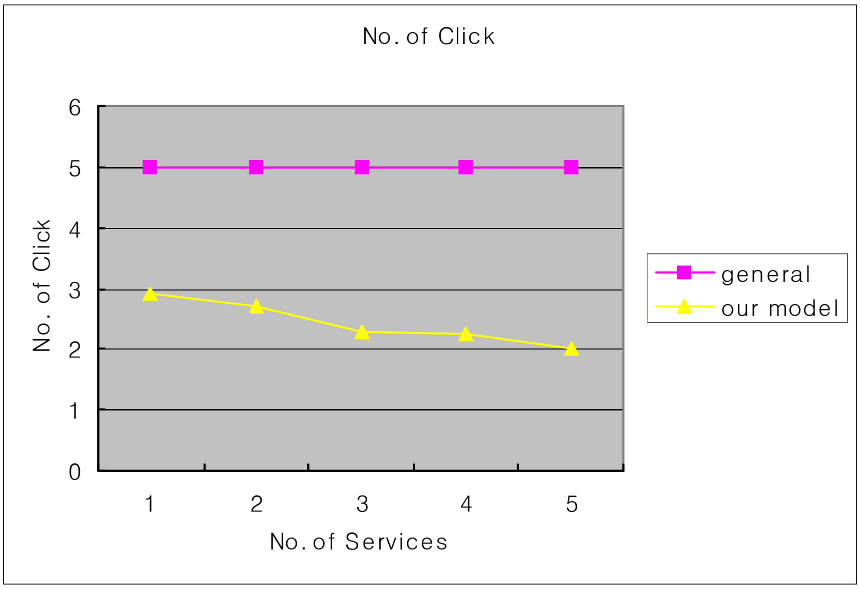


Figure 5.11 The number of clicks to target content.

The recall and precision are generally used to evaluate the performance of information search system. Figure 5.12 illustrates the concept of recall and precision. The recall is the percentage of the relevant entries that also appear as retrieved entities, relative to the total number of relevant entries. The precision would be the percentage of the relevant entries that also appear as retrieved entities, relative to the total retrieved entities.

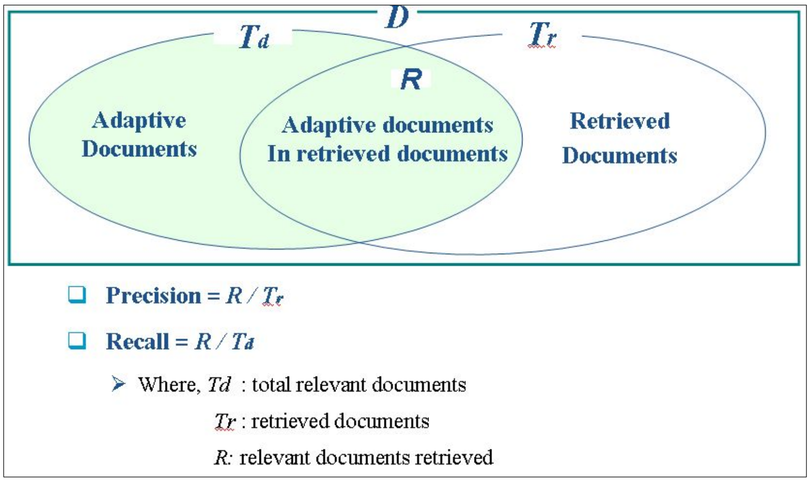


Figure 5.12 The concept of recall and precision.

Figure 5.13 and Figure 5.14 show the recall and precision of the search services respectively. In the general keyword-based search, recall/precision have the same values irrespective of the number of services because user preferences are not considered in search. As shown in Figure 5.13 and Figure 5.14, the personalized search has relatively low recall and high precision while the keyword-based search has high recall and low precision. Moreover, the recall and precision rise in he personalized search based on our framework as the number of services increases.

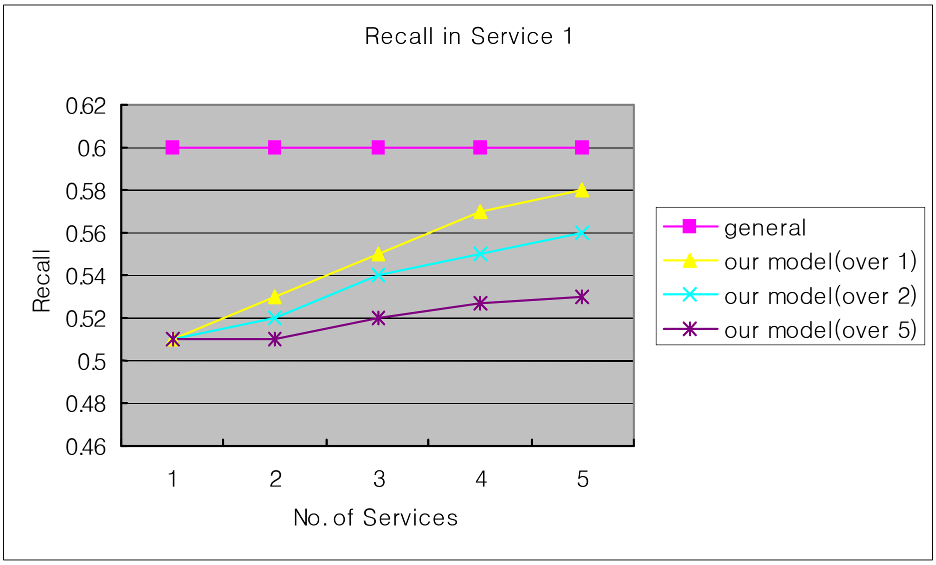


Figure 5.13 The comparison of recall.

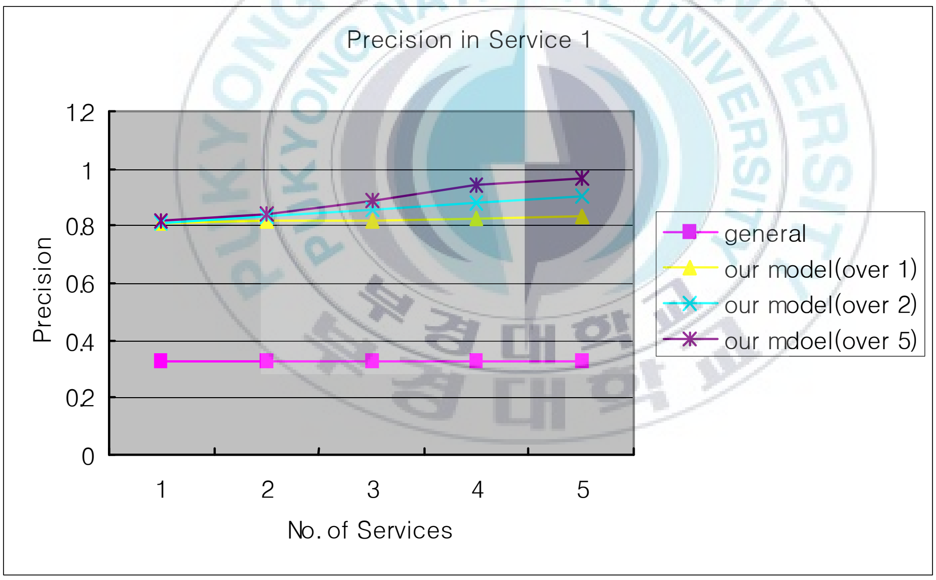


Figure 5.14 The comparison of precision.

5.3 Analysis

In this thesis, we proposed user preference sharing model based on Semantic Web, and realized the framework supporting our model. Our framework enables various services to share semantically user's preferences among various services. Our approach can also enhance human-computer interaction by recommending user-adaptive information using the user preferences specified in other personalized services. For example, if a user usually uses a paper search service to find his/her interesting paper about the field of 'ontology', an e-learning service can recommend its contents for concepts similar to 'ontology' such as Semantic Web when the user accesses the e-learning service.

This chapter implemented an example application with five example services, and evaluated the performance of our framework and benefits in the personalized services. Main contributions of our approach include follows, and we summarize the feature of our user preference sharing framework in Table 5.3.

- Heterogeneity and interoperability of our framework: Our framework supports the user preference sharing among various personalized services because the user preferences are described by the UPDL specified based on OWL. The UPDL ontology especially provides more expressive and more flexible description mechanism for personal preference by allowing personal preference to be described over various domain-specific ontologies. In the evaluation with 5 services, the average number of the available preferences of our framework was improved by 3.2, 5.6, and 7.2, compared with stand-alone model that has value of 2. Moreover, the efficiency of our framework could be improved if much more services are attended.

- Benefits in personalized services: Our approach provides advanced mechanism for personal preference profiling. Because our profiling approach is based on hierarchically classified ontologies, it is possible to recommend not only with explicit keyword but also with similar concept. In the evaluation with 5 services, the average number of clicks in the personalized browsing decreased by 2, compared with the general browsing with value of 5. The efficiency of the personalized search was improved because the precision of our approach increased by about 0.8, 0.9, and 0.98, compared with the general search that has value of 0.6.
- Flexibility: Even if new services are appended and service-specific ontologies are extended, user preferences are described and shared in same method without modifying the UPMS and applications.

Table 5.3 The comparison of the personalized services.

Factors	stand-alone	Standard Exchange Format	Our Approach
Expressiveness	Medium	Low	High
Interoperability	Low	Medium	High
Effectiveness	Medium (in local)	Low	High
Reliability of Results	High (in local)	High	Medium
Application Dependency	High	Medium	Low
Flexibility	High (in local)	Low	High
Processing Time	Low	Medium	High

Chapter 6

Conclusions

Our goal in this thesis was to enhance both heterogeneity and interoperability in exchanging user preferences used in personalized services. For the purpose, we adopted Semantic Web technologies in this thesis. In this thesis, we addressed some research challenges in personalized services. After that, we proposed user preference sharing model and framework based on Semantic Web. This chapter summarizes the contributions of this thesis.

In this thesis, we proposed the user preference sharing model including a data model for user preferences and its sharing mechanism. As the data model, we specified the User Preference

Description Language (UPDL) that is an OWL-based description language for user preferences. Because UPDL was designed to enable user preferences to be described over service-specific ontologies, it provided the heterogeneity in describing user preferences and the possibility of sharing in exchanging user preferences. We also proposed a similarity evaluation as the sharing mechanism. This mechanism computes similarity between two concepts from two different service-specific ontologies based on the hierarchy of ontologies.

Next, we designed the framework to support our user preference sharing model. The user preference sharing framework based on 3-tiered model consists of application tier, middleware tier, and data tier. In the application tier, personalized services serve their contents based on their service-specific ontologies, and recommend user-adaptive contents using user preferences received from middleware. In the data tier, user preferences described by UPDL are stored in the user profile. As the middleware, we implemented the User Preference Management System (UPMS).

UPMS plays the roles of acquisition and sharing of user preferences through three different kinds of managers as follows. User Manager registers new users and creates new user profile. Acquisition Manager provides a uniform user interface to acquire user preferences over service-specific ontologies and updates user preferences in the user profile. Access Manager offers application programming interface (API) for personalized services and, it returns the evaluated user preferences and weights computed by Similarity Evaluator.

In conclusion, our approach provided some contributions as follow. First, The UPDL ontology provided more expressive and more flexible description mechanism for personal preference by allowing personal preference to be described over various domain-specific ontologies. Second, Our approach supported interoperability by sharing user preferences among various personalized services. Third, Even if new services are appended and service-specific ontologies are extended, user preferences was described and shared in same way without modifying the middleware and application services. Finally, our approach provided the advanced mechanism in recommending user-adaptive information preference profiling. Because our profiling approach is based on hierarchically classified ontologies, it is possible to recommend not only with explicit keyword but also with similar concept.

References

[1] Silvana Aciar, Debbie Zhang, Simeon Simoff, and John Debenham, "Recommender System Based on Consumer Product Reviews," Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, 2006.

[2] Antonious and van Harmelen, *A Semantic Web Primer*, The MIT Press Cambridge, Massachusetts London, England, 2004.

[3] Sean Bechhofer, Ian Horrocks, Carole Goble, and Robert Stevens, "OilEd: a Reasonable Ontology Editor for the Semantic Web," Proceedings of KI2001, Springer-Verlag LNAI Vol. 2174, 2001.

[4] Tim Berners-Lee, James Hendler and Ora Lassila. "The Semantic Web," In Scientific American, 2001.

[5] Dan Brickley and Libby Miller, "FOAF Vocabulary Specification 0.9," available at http://xmlns.com/foaf/0.1/, May 2007.

[6] Dan Brickley and R.V. Guha, "RDF Vocabulary Description Language 1.0: RDF Schema," W3C Recommendation. available at http://www.w3.org/TR/rdf-schema/, 2004.

[7] Yi-Shin Chen and Cyrus Shahabi, "Automatically Improving the Accuracy of User Profiles with Genetic Algorithm," In the Proceedings of IASTED International Conference on

Artificial Intelligence and Soft Computing, 2001.
[8] Nam-deok Cho, and Eun-ser Lee, "Design and Implementation of Semantic Web Search System Using Ontology and Anchor Text," ICCSA 2006, Lecture Notes in Computer Science, Springer Berlin, Vol.3984, 2006.
[9] Jin-Tak Choi, "A Study on Authentication Management Technique Used of SSO," KSIAM IT series Vol.10, No.1, 2006
[10] Eastlake, D., Reagle, J., "XML Encryption Syntax and Processing," W3C Candidate Recommendation, 2002.
[11] Eastlake, D., Reagle, J., Solo, D., "XML-Signature Syntax and Processing," 2002.
[12] Michael. Daconta, Leo J. Obrst, and Kevin T, Simth, *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management*, Wiley Publishing Inc., Indianapolis, Indiana, 2003.
[13] Honghua Dai and Bamshad Mobasher, "Using Ontologies to Discover Domain-Level Web Usage Profiles," In proceedings of the second semantic web Mining Workshop at PKDD2001, 2001
[14] John Davies, Dieter Fensel, and Frank van Harmelen, "*Towards the Semantic Web Ontology-driven Knowledge Management*," JOHN WILEY & SONS, LTD, 2003.
[15] Stefan Decker, Prasenjit Mitra, and Sergey Melnik, "Framework for the Semantic Web: An RDF Tutorial," IEEE Internet Computing, 2000.

[16] Ying Ding, "A review of ontologies with the Semantic Web in view," Journal of Information Science, Vol.27, No.6, 2001.

[17] Natalya F. Noy, Michael Sintek, Stefan Decker, Monica Crubézy, Ray W. Fergerson, and Mark A. Musen, "Creating Semantic Web Contents with Protégé-2000," IEEE INTELLIGENT SYSTEMS, 2001.

[18] Dieter Fensel, Frank van Harmelen, Ian Horrocks, Deborah L. McGuinness, and Peter F. Patel-Schneider, "OIL: An Ontology Infrastructure for the Semantic Web," IEEE Intelligent System", 2001.

[19] Tim Finin, James Mayfield, Anupam Joshi, R.Scott Cost, and Clay Fink, "Information Retrieval and the Semantic Web," Proceeding of the 38th Hawaii International Conference on System Sciences, IEEE, 2005.

[20] D. Garlan, R. Allen, and J. Ockerbloom, "Architectural Mismatch or Why It Is So Hard to Build Systems out of Existing Parts," presented at The 17th International Conference on Software Engineering, Seattle, Washington, USA, 1995.

[21] Asunción Gómez-Pérez and Oscar Corcho, "Ontology Languages for the Semantic Web," IEEE Intelligent Systems, 2002.

[22] Donghai Guan, Qing Li, Sungyoung Lee, and Youngkoo Lee, "A Context-Aware Music Recommendation Agent in Smart

Office," FSKD 2006. Lecture Notes in Artificial Intelligence, vol.4223, Springer, Heidelberg, 2006.

[23] Alon Y. Halevy, Zachary G. Ives, Peter Mork, and Igor Tatarinov. "Piazza: Data management infrastructure for semantic web applications," WWW, 2003.

[24] Frank van Harmelen, Ian Horrocks, James Hendler, and Deborah L. McGuinness, "The semantic Web and its languages," IEEE Intelligent Systems, 2000.

[25] N. Henze and M. Kriesell, "Personalization functionality for the semanticc web: Architectural outline and first sample implementation," EAW, 2004.

[26] Johan Hjelm, *Creating the Semantic Web with RDF*, WILEY, Canada, 2001.

[27] R. Housley, W. Polk, W. Ford, and D. Solo. "Internet x.509 public key infrastructure certificate and certificate revocation list," RFC 3280, 2002.

[28] I. Horrocks, F. van Harmelen, and P. Patel-Schneider., "DAML+OIL.,http://www.daml.org/2001/03/daml+oil-index.html", March 2001.

[29] Renato Iannella, "Representing vCard Objects in RDF/XML", W3C Note 22, available at http://www.w3.org/TR/vcard-rdf, 2001.

[30] IEEE P1484.2 "Learner Model Working Group: PAPI Learner," Draft 7 Specification. 2003.

[31] IMS Global Learning Consortium: "IMS Learner Information Package Specification v. 1.0," 2001.

[32] A. Kalyanpur, J. Hendler, B. Parsia, and J. Golbeck, "SMORE - Semantic Markup, Ontology, and RDF Editor," Available at: http://www.mindswap.org/papers/SMORE.pdf.

[33] A. Kalyanpur, N. Hashmi, J. Golbeck, and B. Parsia, "Lifecycle of a Casual Web Ontology Development Process," Proceeding of the WWW2004, 2004.

[34] Jong-Woo Kim, Ju-Yeon Kim, and Chang-Soo Kim, "Semantic LBS: Ontological Approach for Enhancing Interoperability in Location Based Services," OTM2006, Lecture Notes in Computer Science,4277, Springer-Verlag, 2006.

[35] Jong-Woo Kim, Ju-Yeon Kim, and Chang-Soo Kim, "The Semantic Web Approach in Location Based Services," Lecture Notes in Computer Science, Vol. 3481, Springer-Verlag, ICCSA205, 2005.

[36] Jong-Woo Kim, Ju-Yeon Kim, and Chang-Soo Kim, "Location-Sensitive Tour Guide Services Using the Semantic Web," KES2005, LNCS 3682, Springer-Verlag. 2005.

[37] Ju-Yeon Kim, Jong-Woo Kim, and Chang-Soo Kim, "A Study on User Preference Sharing based on Semantic Web in Personalized Services," Journal of Korea Multimedia Society, Vol. 10, No. 10, 2007.

[38] Ju-Yeon Kim, Jong-Woo Kim, and Chang-Soo Kim,

"Ontology-Based User Preference Modeling for Enhancing Interoperability in Personalized Services," International HCI 2007, LNCS 4556, 2007.

[39] Ju-Yeon Kim, Jong-Woo Kim, Sung-Ki Park, and Chang-Soo Kim, "A design of Personalized Services using Ontology-based User Profile," Proceedings of Multimedia Conference, Vol.10, No.1, 2007.

[40] Ju-Yeon Kim, Jong-Woo Kim, Sung-Ki Park, and Chang-Soo Kim, "Integrated Services of Location-based Tour Information in Semantic Web," Proceedings of KISS Youngnam Branch Conference, Vol.14, No.1, 2006.12.

[41] Ju-Yeon Kim, Jong-Woo Kim, Sung-Ki Park, and Chang-Soo Kim, "Design of the Semantic Web-based Information Retrieval System Using Personlized Preference," Proceedings of Multimedia Conference, Vol.9, No.1, 2006.5.

[42] Ju-Yeon Kim, Jong-Woo Kim, and Chang-Soo Kim, "A Semantic Web Approach toward Intelligent Tour Guide Services," ISAE2005, 2005.

[43] Ju-Yeon Kim, Jong-Woo Kim, Hyun-Suk Hwang, and Chang Soo-Kim, "A study on semantic representation for tour information and its efficient retrieval," Proceedings of Multimedia Conference, Vol.8, No.2, 2005.11.

[44] Ju-Yeon Kim, Chang-Soo Kim, Ha-Sik Kim, and Sung-Jun Lee, "Tour Information retrieval using the Semantic Web,"

Proceedings of Multimedia Conference, Vol.8, No.1, 2005.5.
[45] Wooju Kim, SungKyu Lee, and DaeWoo Choi, "Semantic Web Based Intelligent Product and Service Search Framework for Location-Based Services," ICCSA2005, LNCS3483, pp.103-112, 2005.
[46] Holger Knublauch, "Ontology-Driven Software Development in the Context of the Semantic Web: An Example Senario with Protégé/OWL," MDSW2004, 2004.
[47] Deborah L. McGuinness, Richard Fikes, James Hendler, and Lynn Andrea Stein, "DAML+OIL: An Ontology Language for the SemanticWeb," IEEE Intelligent Systems, IEEE, 2002.
[48] Ernst&Young LLP, "Customer Benefits from Current information Sharing by Financial Services Companies," 2000.
[49] James Mayfield, Tim Finin, "Information retrieval on the Semantic Web:Integrating inference and retrieval," Proceedings of the 38th Hawaii International Conference on System Sciences-2005, IEEE 2005.
[50] Brian McBride. "Jena: Implementing the RDF model and syntax specication," In Proceedings of the 2001 Semantic Web Workshop, 2001.
[51] B. McBride, "Jena: A Semantic Web Toolkit," IEEE INTERNET COMPUTING, 2002.
[52] Eric Miller, "The W3C's Semantic Web Activity:An Update," IEEE Computer Society, IEEE, 2004.

[53] Dunja Mladeni, "Personal WebWatcher: Design and Implementation," Technical Report IJS-DP-7472, J. Stefan Institute, Department for Intelligent Systems, 1998.

[54] M. Montebello, W.A. Gray, S. Hurley, "A Personal Evolvable Advisor for WWW Knowledge-Based Systems," Proc. IDEAS, 1998.

[55] E. Niemelä and T. Vaskivuo, "Agile Middleware of Pervasive Computing Environments," presented at Middleware Support for Pervasive Computing Workshop, 2004.

[56] B. D. Noble, D. Narayannan, J. E. Tilton, J. Flinn, and K. R. Walker, "Agile Application-Aware Adaptation for Mobility," presented at The 16th ACM Symposium on Operating Systems Principles, Saint Malo, France, 1997.

[57] D. Pakkala, "Lightweigth Distributed Service Platform for Adaptive Mobile Services," VTT Technical Research Centre of Finland, Espoo, Finland, VTT Publications 2004.

[58] Syng-Seok Park, Chang-Soo Kim, Jong-Woo Kim, Ju-Yeon Kim, and Sung-Ki Park, "The Design of Web-linked Location-based Tour Guide Service Model," Proceedings of Multimedia Conference, Vol.7, No.2, 2004.11.

[59] PIDL, W3C, available at http://www.w3.org/TR/NOTE-PIDL.

[60] Shelley Powers, *Practical RDF*, O'Reilly, Gravenstein Highway North, Sebastopol, CA, 2003.

[61] Francesco Ricci, "Travel Recommender System," IEEE

Intelligent System, 2002

[62] Doug Riecken and Guest Editor. “Personalized views of personalization,” communications of the ACM, Vol.43, No.8, 2000.

[63] M. Roman, C. Hess, R. Cerqueira, A. Ranganathan, R. H. Campbell, and K. Nahrstedt, "A Middleware Infrastructure for Active Spaces," in IEEE Pervasive Computing, 2002.

[64] J. Shavlik, S. Calcari, T. Eliassi-Rad, and J. Solock. "An Instructable, Adaptive Interface for Discovering and Monitoring Information on the World Wide Web," In Proceedings of the International Conference on Intelligent User Interfaces, 1999.

[65] H. Schulzrinne and J. Rosenberg, "Application Layer Mobility Using SIP," ACM Sigmobile. Mobile Computing and Communications Review, 2000.

[66] A. Seaborne, "RDQL-A Query Language for RDF," W3C, available at http://www.w3.org/Submission/2004/SUMB-RDQL20040109, 2004.

[67] Kazunari Sugiyama, Kenji Hatano, Masatoshi Yoshikawa, and Shunsuke Uemura, "User-Oriented Adaptive Web Information Retrieval Based On Implicit Observation," In Proceedings of the Asia Pacific Web Conference, LNCS Vol.3007, Springer, 2004.

[68] York Sure, Pascal Hitzler, Andreas Eberhart, and Rudi Studer, "The Semantic Web in One Day," IEEE Computer

Society, IEEE 2005.
[69] A. Taulavuori, E. Niemelä, and M. Matinlassi, "Evaluating the Integrability of COTS Components - the product family viewpoint," in Building Quality into COTS Components - Testing and Debugging, S. Beydeda and V. Gruhn, Eds.: Springer-Verlag, 2004.
[70] W3C. "http://www.w3.org/2001/sw/WebOnt/. Web-ontology working group.", 2001.
[71] W3C: Extensible Markup Language, W3C Recommendation, available at http://www.w3c.org/XML/.
[72] Shijun Yu, Stefano Spaccapietra, Nadine Cullot, and Marie-Aude Aufaure, "User Profiles in Location-based Services: Make Humans More Nomadic and Personalized," Databases and Applications, ACTA, 2004.