

# Implementing an Ubiquitous Resource Service Architecture Based on the Web and LDAP

Chu-Sing Yang, Chien-Yuan Liu, Sheng-Wen Bai, Ken-Da Wu  
Jr-Houng Chen, Hor-Tzung Lee, and Jr-Houng Wu

Institute of Computer and Information Engineering  
National Sun Yat-Sen University  
Kaohsiung, Taiwan, R.O.C.

email: csyang@cse.nsysu.edu.tw

## ABSTRACT

*Due to the fast growth of the Internet, huge amount of resources have becoming available for the Internet users. A resource service architecture is required to provide ubiquitous and transparent resource services to the Internet users. Thus a user can efficiently and easily access to and utilize the resource services in the Internet. This is especially important for a mobile user. We had proposed an ubiquitous resource service architecture to address the above mentioned problem. The implementation of the prototype system demonstrates the concept of the architecture is feasible in a mobile and heterogeneous environment.*

*Keywords: LDAP, Mobile, Resource Service, Web*

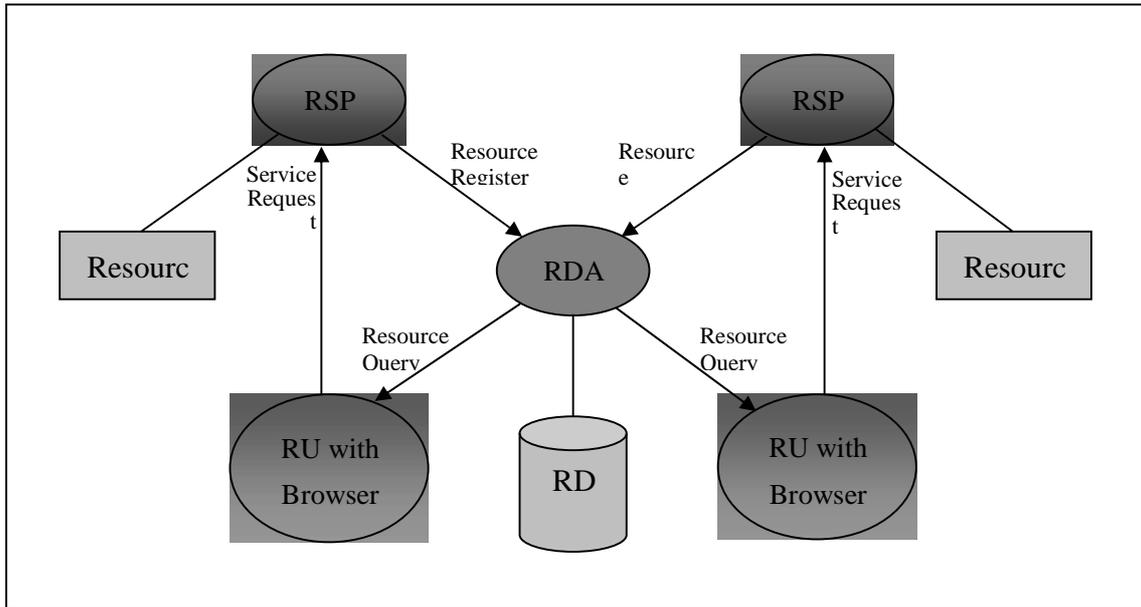
## 1. Introduction

Thanks to the web technology, the Internet is quickly growing in recent years. A web user can easily navigate over the webspace by simply point-and-click operation. This encourages that explosive numbers of users wish to access the Internet for acquiring information and services for their daily life. Since the existing potential markets of the Internet are huge, lots of companies are considering how to provide more and better information and services. Therefore users can obtain more convenient and higher-quality of services.

There are many kinds of information and services in the Internet. We denoted these information and services as the resources. The resources can be classified into following categories:

- **Information or content service** – This type of service provides a user with dissimilar information, such as news, shopping advertisements, research papers, reports, hyperlinks of search engines, and hyperlinks of disparate proxies and servers, etc.
- **Peripheral service** – Desktop users usually use general peripheral services- printer, plotter, scanner, and fax machine etc. in local networks. Sometimes they may need to use some special peripheral services- high-speed or color printer, large-size plotter, and high-resolution scanner etc. in remote networks. More needs of peripheral services shall occur from mobile users.
- **Computation service** – The computation service could provide a greater computing power for a thin client user. Parallel mechanism is a must technology to support such kind of computation services. Supercomputers and clusters are usually adapted as the computation servers.
- **Storage service** – Disk array, tape drivers, and disc jukebox normally have very high storage capacity to keep a gigantic information for users. However, there are also very expensive that most of the users can not afford to own these equipment by themselves. Users usually have to use them in sharing manner.
- **Network service** – Wired, terrestrial wireless, satellite broadband networks are developing to serve the multimedia communications. Interactive and real-time multimedia communications need the guaranteed QoS to offer enough network bandwidth and acceptable network delay time.

Although, these resources are directly or indirectly connected to the Internet, a Internet user can not



**Figure 1: The Ubiquitous Resource Service Architecture**

easily discover which resources are available to them. For most of Internet users, the Internet resources are just like some opaque facilities. Especially for the mobile users, they totally unknown to the current environment. Therefore, a resource service architecture is necessary to provide and manage all the resources efficiently and transparently.

Today there were already some special solutions applied on certain platforms. For example, a Microsoft Windows user can easily find and use a resource in same workgroup or domain. However, an UNIX user is difficult to get any information of the same resource from Windows world. Although some products can provide transparent services across few different system platforms by a specific-integrated solution, that only solves a part of the problem. Furthermore, the specific solution is not general and can only applied to very restricted platforms.

In order to provide a countermeasure to reduce the barrier of resource sharing, we had proposed the ubiquitous resource service architecture [1]. The goal of the architecture is to provide a very convenient environment for resource sharing in a heterogeneous environment. A Internet user, especially a mobile user, can easily request a service without any special knowledge or enough resource information. In addition to the friendly and transparent features, the architecture also considers the flexibility, scalability,

integrity, security, and performance for future applications.

The paper is divided into four sections. Section 1 briefly introduces the categories of the Internet resources and explains the motivation behind the work for the ubiquitous resource service architecture. Section 2 describes the element of the ubiquitous resource service architecture in detail. The functions of every component is explained in short. The implementation and the discussion of the prototype system for the ubiquitous resource service architecture is explained in section 3. Finally, We give a summary conclusion and directions for future work in section 4.

## **2. Ubiquitous Resource Service Architecture (URSA)**

The URSA is consisted of the resource directory, the resource directory agent, the resource service provider, and the resource user. The configuration of the URSA is illustrated in figure 1. Detailed description of each component is depicted as follows:

### **2.1 Resource Directory (RD)**

A resource in the Internet can be represented as an object class with multiple attributes. A resource may be a super class or a sub class of another resource. A

resource usually belongs to an organization or a subdivision and be managed by an assigned administrator. All these resource object classes and attributes and their relationship can be represented by and stored in a directory named RD.

In terms of RD, every resource object is represented by an entry, all entries forms the directory information base (DIB) [2]. The directory information base is almost in tree shape. The tree is called directory information tree (DIT). Every entry has a relative distinguished name (RDN). Each RDN is unambiguous to its direct superior. A RDN plus the RDN of its superior forms a globally unique DN. The object RDN and the information of the object is represented by attributes in the entry. An attribute is consisted of a pair of attribute type and attribute value. An alias entry, which hold a pointer to another entry, is allowed.

The structure of DIT is specified by a directory schema, which defines object classes, attribute types, and attribute syntax. An object class specifies a set of mandatory or optional attributes for the entry of a given object class. From the organizational point of view, the directory is hierarchical and can be subdivided into subdomains. Hence, the schema can

be composed of a number of subschemas, which may be administered by subdivision administrators.

A directory can provide at least two levels of authentication[3]. First one is the simple authentication based on user's DN and password. Second one is the strong authentication using public key cryptographic scheme. The public key infrastructure (PKI) can be fully implemented in the directory service to strengthen the security and the management of the URSA. In the PKI, a secret key is owned by its owner, whilst a public key is announce to every user of the system. One could cipher information with a public key and decipher the encrypted information with the owner's secret key.

## 2.2 Resource Directory Agent (RDA)

A RDA is a software agent which provides the services for registering a resource information into the RD and for querying a resource information from the RD. Before a network user obtains a resource service from the Internet, the user firstly has to connect to the RDA for querying the address and basic information of the Internet resource. For a desktop user, the address of RDA can be

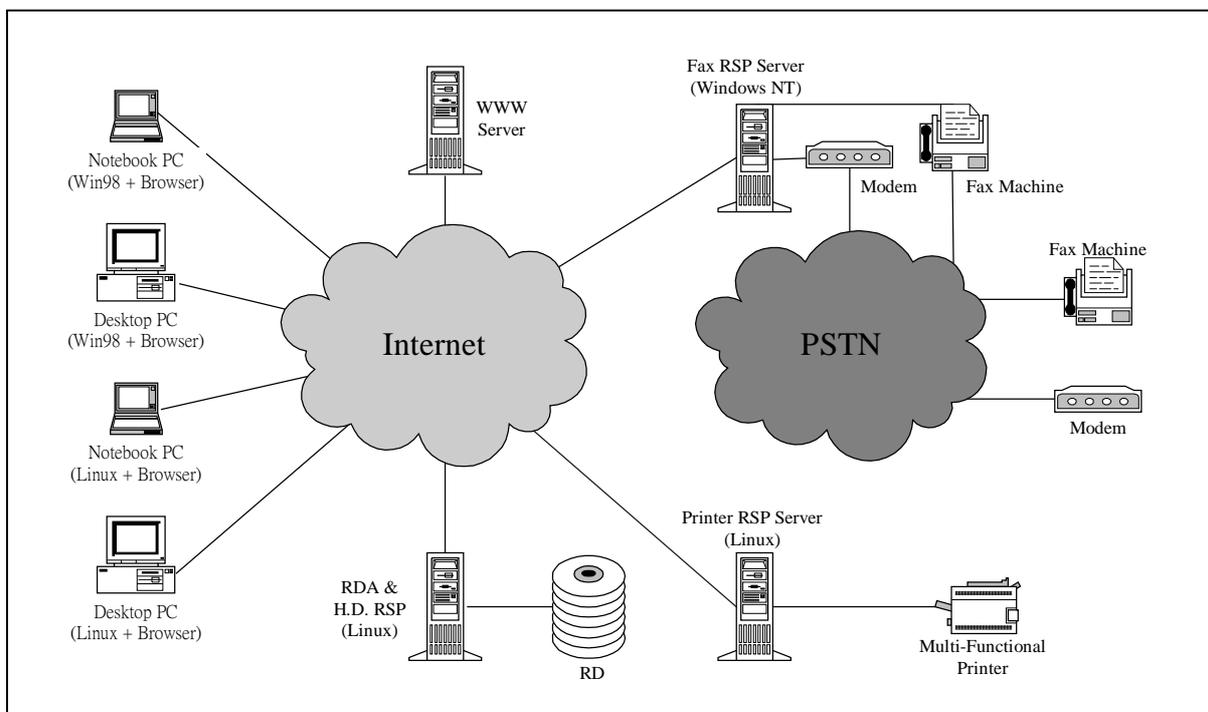


Figure 2: The URSA Prototype System

**Table 1: The Selected Softwares**

Component Name	Operating System	Softwares
RD	Linux (Red Hat 6.0)	UMICH LDAP 3.3
RDA	Linux (Red Hat 6.0)	Apache 1.3.6 + PHP 3.0.6
White pages RSP	collocated at RDA	collocated at RDA
Fax RSP	Windows NT4	Apache 1.3.6 + PHP 3.0.6
Printer RSP	Linux (Red Hat 6.0)	Apache 1.3.6 + PHP 3.0.6
Desktop RU	Linux / Windows 98	Netscape Communicator 4.5 / Microsoft IE5.0
Mobile RU	Linux / Windows 98	Netscape Communicator 4.5 / Microsoft IE5.0

pre-configured into the desktop workstation. As for a mobile node user, the RDA address can be dynamically allocated through using DHCP. After querying the address of the given-type resource, the user can choose one of the nearest available resource for service. Of course, the chosen resource should be a public-sharing resource or be once authorized to the user.

### 2.3 Resource Service Provider (RSP)

A RSP is a kind of software proxy, which can perform a designated service for a request from a resource user. Different kinds of resources need disparate RSP to provide services to the Internet users. For example, a printer RSP provides printing service whereas a fax RSP provides fax machine service. In most cases, the RSP can collocate with the desired resource. This is because of performance consideration. However, a RSP can be assigned to another distributed server, if the distributed allocation is required for flexibility.

For some application, a RSP may need to request further service from other RSP to accomplish a complex service. In this case, The RSP will act as a resource user to solicit further service from other RSP. For example, when a printer RSP receives a printer service request, the RSP checks the details of the request. If the request includes a file format beyond the processing capability of the printer RSP, the RSP may ask another RSP to convert the file format. After that, the printer RSP can process the converted file and issue the print service to accomplish the request.

### 2.4 Resource User (RU)

A RU is an Internet user who intends to utilize the resource service of the Internet. Since most of the Internet users navigate the Internet with web browser, the URSA is decided to leverage the web technology in order to attract the Internet users to utilize the resource services provided by the URSA.

## 3. Implementation

In order to conduct a pilot test for the URSA, we designed and implemented a prototype system. The system consisted of a RD for keeping the resource object information, a RDA for providing resource registration and query services, a white-pages RSP for providing the human directory service, a printer RSP for serving printer service, a fax RSP for providing fax sending and receiving services, and few RUs. Figure 2 illustrates the URSA prototype system.

The prototype system was designed to execute across a heterogeneous environment. The RD is located at a Linux [4] server. The RDA is running on another Linux server. The fax RSP is implemented on a Windows NT server. The printer RSP is assigned to the other Linux server. One desktop PC and one mobile notebook PC each is running with the Linux plus the Netscape Communicator. Another one set of a desktop PC and a mobile notebook PC is running with the Windows 98 plus the Internet Explorer. Table 1 itemizes the selected softwares for the implementation. The complete functional descriptions are given in the following subsections.

### 3.1 RD

The RD is implemented on the lightweight directory access protocol (LDAP) [5, 6] and a LDAP directory server [7]. The reasons why we choose the LDAP as our directory service is based on the following contemplation:

- The LDAP has recently be accepted as an industrial de-facto standard for accessing attribute-value-tuple information from the directory server of networks, especially in the area of the Internet and modern mobile communication networks [8].
- Network manufactures have actively integrated LDAP into their products.[9]
- New lifetime, referral, and replication [10] make the static directories become more dynamic and more strong-related directories, e.g. a hierarchical-directories structure.
- New network applications have built with LDAP to enhance the communication and management capability [11, 12, 13]

The RD kept the schema of all object classes in the prototype system. The object classes included a white pages classes, a printer class, a fax class, etc. The attributes of every object class are shown in figure 3. In figure 3, the dotted line means that the

directory information sub-tree is out of the implementation scope; the real line denotes the scope of the implemetation, and the attributes in *Italic style* mean they are mandatory attributes; whilst the attributes in normal style means they are optional attributes.

### 3.2 RDA

A RDA serves as an agent. It receives the request for registering a resource information from a RSP. At the same time, it transforms the request into a LDAP formatted request for adding the resource information into a corresponding RD. Figure 4.(a) depicts the object interaction model for RSP registration. In order to leverage the most prevalent web technology, the RDA adapts the Apache as its web server. Meanwhile the RDA needs some script programs to be invoked at the web server to work as the traditional CGI [14] programs.

Concerning the CGI programming languages, there are lots of candidates which have been succesfully applied on many existing Apache web servers. PHP [15] is one of the most popular CGI scripting language. In particular, PHP has a bunch of APIs for accessing more than 15 different databases, including LDAP directory service. Therefore PHP is choosen as the CGI scripting language for our Apache web server.

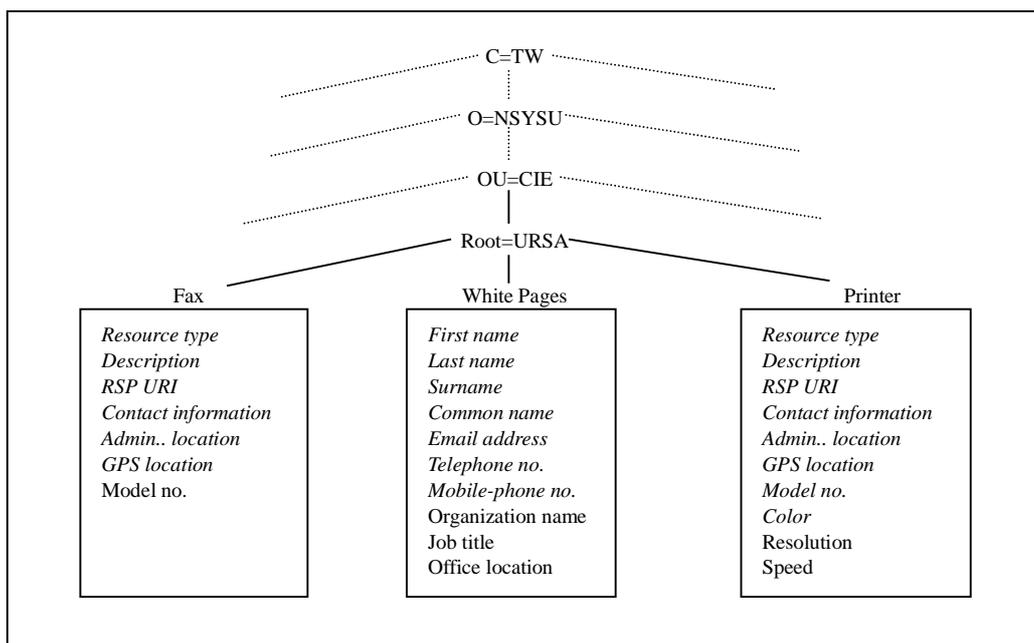
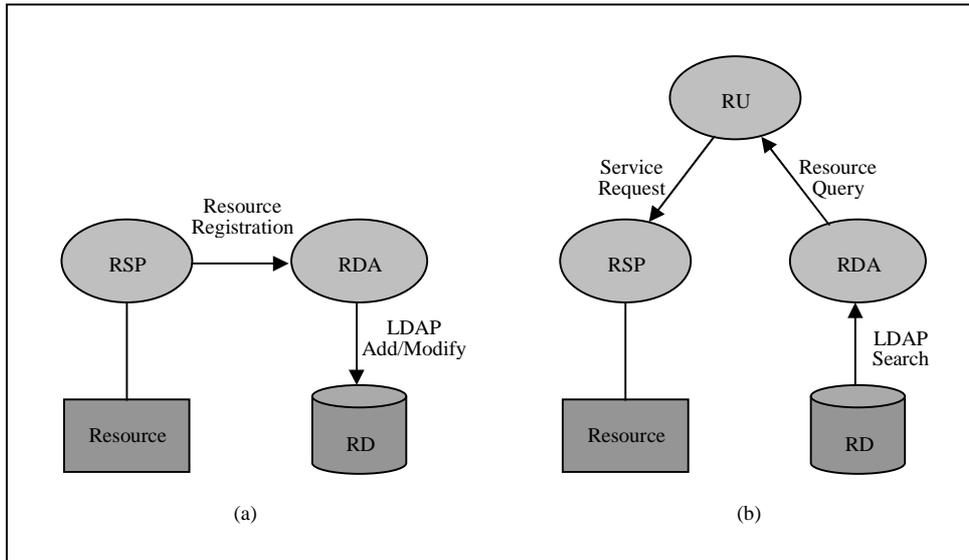


Figure 3: The Resource Schema Diagram



**Figure 4: The Resource Registration / Service Models**

The requests of the resource information registration from a RSP and the resource information query from a RU are in http message format and will invoke the corresponding CGI programs for registration processing and query processing. The CGI programs which are written in PHP scripting language then call LDAP directory service functions to add, modify, and/or delete the resource information in the LDAP directory. After the CGI programs finish the directory manipulations, it return the results back to the originator.

### 3.3 RSP

In the prototype implementation, there are three types of RSPs- white pages RSP, printer RSP, and fax RSP.

- **White pages RSP**

A white pages RSP provides the human directory service for a web user. There is one HTML form located at the RDA for users to register, query, and/or update their basic information, such as name, surname, common name, email address, telephone number, mobile phone number, organization name, job title, office location, etc. The schema for these basic information is defined in a LDAP configuration file located at a RD. A resource user can register a

personal information to or query it from the RD through the web service of the RDA.

- **Fax RSP**

A fax RSP provides the fax sending and receiving services for a web user. A fax RSP itself is a web server with a CGI program in PHP scripting language. In the web server, there is a HTML form for a web user to fill out the designated fax sending information, such as destined fax number, receiver name, subject, cover page information, and the attached file. In addition to the HTML form, there is a corresponding CGI program which responses for the fax sending task. The PHP program executes an external fax service command with certain parameters to serve the requested fax services.

- **Printer RSP**

A printer RSP provides the printer service for a web user. A printer RSP itself is a web server with a CGI program in PHP scripting language. In this web server, there is a HTML form for a web user to fill out the designated request information for a printer service, such as numbers of copy, paper size, printing resolution, and the upload file for printing out. Besides the HTML form, there is a corresponding CGI program which responses for the printing service. The PHP program executes an external printing

command with certain parameters to print out the desired printing result.

### 3.4 RU

A RU likes an ordinary web user. A RU uses a popular web browser, such as Netscape Communicator or Microsoft Internet Explorer, to request for resource services. The object interaction model of resource service is depicted in figure 4.(b). The resource query, service request, and service confirmation procedures are illustrated in figure 5.

For example, if a RU wants to contact with another people called Sam, the RU has to query a RDA in advance for getting the basic information of Sam, like email address or telephone number. Further example, if the RU needs to use a color laser printer service, the RU also has to query the RDA for knowing that where is a color laser printer located at and how to access the printing service. After the RU obtains the resource basic information which includes the uniform resource indicator (URI) of that resource, the RU points and clicks the URI hyperlink to connect to the resource service form page located at the corresponding printer RSP. Then, the RU must fill out the resource service form page and submit the request to invoke the action program- a corresponding printer service CGI program. The detailed action procedures was already described in the printer RSP

of section 3.3.

### 3.5 Discussion

The prototype system is now successfully completed. A RU can directly acquire human's basic information from the human directory service provided by the RDA. A RU also can easily find a right resource from the RDA and can transparently request a service provided by the corresponding RSP. All these queries and requests for services are actually conducted over a heterogeneous environment as our preassumption.

Currently we are extending the prototype system to support a location-awared resource service[16, 17]. A location-awared resource service is very important and convient for a mobile user. Because a mobile user is unable to obtain any available nearby resources without extra help information. For example, when a mobile RU is driving to an unfamiliar city. The RU would strongly require the information about parking place, restrurant, automaic telling machine, hotel, and gasilin station, etc. Further example, a mobile RU goes far away from his office for a significant business meeting and the RU losses one important business proposal and drawing dedicated to the meeting. The RU can first query his home RDA for the document files and then find a nearest high-resolution printer and large-scale plotter.

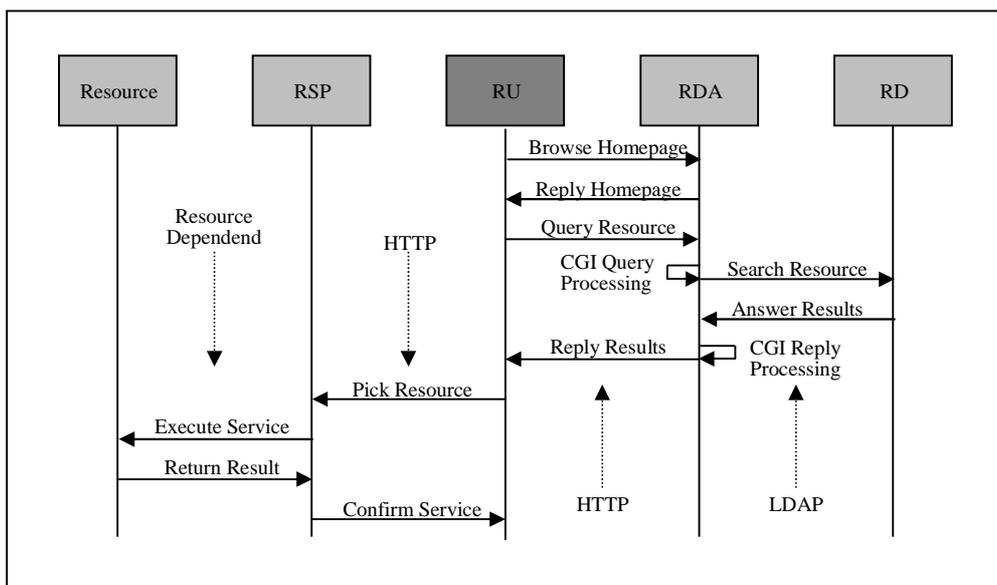


Figure 5: The Resource Service Scenario Diagram

The RU can also find a nearby RDA for querying more precise information of printer and poltter. From the query results, the RU then can issue the request for printing service directly to the corresponding RSP or indirectly through the contact to the administrator of that RSP for further service.

In order to provide the location-aware resource service, the location information must be added into the schemas of all resource classes. When a RU issues a resource query, the RDA not only reply the previous resource information, but also include the location informtion of that resource for RU reference. Therefore, the RU can make a decision concerning the selection of the resource type and the corresponding RSP.

#### 4. Conclusion

Enormous numbers of the Internet resources are existing in the Internet. But the Internet users still suffer from that it is uneasy to use all the Internet resources directly, especially for mobile users. The key point is how to provide resource services transparently and efficiently. It is necessary to come out an service architecture to solve the problem throughly.

We had proposed the URSA to solve this problem. The implementation was conducted to proof that the URSA is feasible. The prototype system was successfully implemented. The system was leveraged with two most widespread technologies, the Web and LDAP. With these two prevalent technologies, the deployment and adaption of the URSA is much easy and achivable

Finally, we are working on the extension of the prototype system to provide a location-aware resource service. The location-aware resource service can provide more intellegient resource information and convinent resource service for a mobile RU. We imagine that the location-aware service is a trend for the future Internet resource service.

#### REFERENCES

- [1] C.S. Yang, C.Y.Liu, and K.D. Wu, "An Ubiquitous Resource Service Architecture Based on Directory Service," CAAPCON, Proceedings 1999, pp. 6.2.1~6.2.5
- [2] "The Directory: Overview of Concepts, Models and Services," ITU-T Recommendation X.500, 1988
- [3] "The Directory: Authentication Framework," ITU-T Recommendation X.509, 1993
- [4] "Linux User's Group," <http://www.linux.org/>, <http://www.linux.org.tw/>, <http://cle.linux.org.tw/>, and <http://www.redhat.com/>
- [5] Timothy A.Howes, "The Lightweight Directory Access Protocol: X.500 Lite," University of Michigan, July 27, 1995
- [6] M. Wahl, T. Howes, S. Kille "Lightweight Directory Access Protocol (v3)," RFC 2251-2256, December 1997
- [7] Timothy A. Howes and Mark C. Smith, "A scalable, Deployable, Directory Service Framework for the Internet," University of Michigan, March 1995
- [8] "Internet Directory Consortium," <http://www.opengroup/idc/index.htm>
- [9] "Cisco, Microsoft Launch Industrywide Initiative for Directory-Enabled Networks," September 23, 1997
- [10] J. Merrells E. Reed, U. Srinivasan, "LDAP Replication Architecture," draft-ietf-ldup-model-01.txt, June 25, 1999
- [11] Kwok-Yan Lam, Tom Salkield, "Implementing a Highly Available Network Directory Service," Journal of System and Software, Vol. 37, April 1997, pp. 41-47
- [12] Steven Fitzgerald, Ian Foster, Carl Kesselman, Cregor von Laszewski, Warren Smith, Steven Tuecke, "A Directory Service for Configuring High-Performance Distributed Computations," IEEE International Symposium on High Performance Distributed Computing, Proceedings 1997, pp. 365-375
- [13] "Novell NDS Directory Service," <http://www.novell.com/products/nds/index.html>
- [14] "CGI resources," <http://www.cgi-resources.com/>
- [15] "PHP3 Hypertext Preprocesor," <http://www.php.net/>
- [16] Henning Maass, "Location-Aware Mobile Applications Based on Directory Services," International Conference on Mobile Computing and Networking, MOBICOM 1997, pp. 23-33
- [17] A.Krikelis, "Location-Dependent Multimedia Computing," IEEE Concurrency, April-June 1999