Integrating Web Information Sources for Mobile Users(1) — System Architecture

3T−8

Wisut Sae-Tung Junkai Ge Tadashi OHMORI Mamoru HOSHI
Graduate School of Information System, University of Electro-Communications

1 Introduction

Currently, information sources tend to provide their information via WWWs embedding their client applications into Web pages. Let us call such an information source *a Web information source* (WIS). The goal of this work is to design a system for integrating data of WISs by a concept called *navigational integration*. Further, we aim at a system that the integration is performed by mobile users. In this paper, we concentrate on describing this system architecture and the domain hierarchy, which is a mechanism to solve semantic conflict of data of different WISs.

2 Navigational Integration

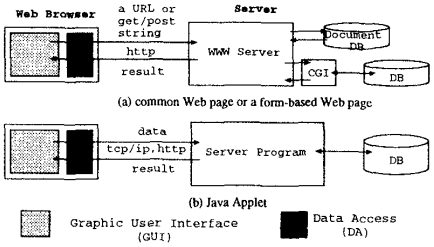

Figure 1: The Component Model of WISs

To apply our approach to a given WIS, the WIS must satisfy some constraints. In general, a client application of the WIS is embedded into a Web page and downloaded into a browser running in a user computer. Let us call such a client application *a Web client application* (WCA). The WCA can be divided into two parts: *a GUI part* (Graphic User Interface) and *a DA part* (Data Access). The GUI part and the DA Part are responsible for gathering information from a user as input data and passing the user's input as conditions to access information from the WISs, respectively. To be able to control these two parts using wrapping approach, they must be distinctly separated as shown in Figure 1. This figure shows two examples of WISs to which our approach can apply: (i) a ordinary Web document server or a form-based CGI Web server, and (ii) a WCA written in the Java language.

Assume there are two WISs. The first WIS provides information about departments, laboratories and their staffs in a university. The second WCA is the CGI-Web page that provides in the detail of laboratories in the university. The *navigational integration* is the process of generating a new WCA from the original WCAs by creating *derived links* that pass the data (laboratory names) from the output page of the first WIS as the input conditions to get the result of the second WIS. It is shown in the Figure 2. However, the format of laboratory data of the first WIS is represented in the different format required by the front end of the WCA of the second WIS. To integrate them together, the semantic conflict problem must be solved.

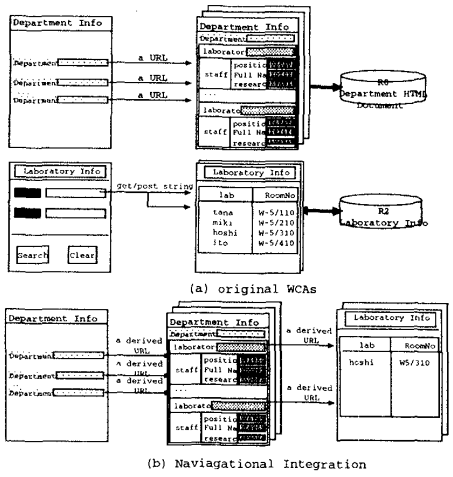

Figure 2: The Navigational Integration

3 System Architecture

3.1 Overview

The system architecture of our approach is shown in Figure 3. This system contains multiple sub-systems called *cells*. A cell is an independently-developed system that manages multiple WISs and provides access to these WISs in our approach. Furthermore, multiple cells form a *region*, which is designed for resolving semantic conflict of data between its cells so that they can exchange data among themselves. All WCAs of the WISs are wrapped and packed into packages. Then, in Figure 3 a mobile user uses our system as follows. While moving, (1) the mobile user can download the packages through a main information resource *(MIR)* acting as a yellow page server in a cell. Next, (2) the mobile user uses the *mediator* module in his mobile computer to explore information from packages, generates a script program and sends it to any MIR (it may be the MIR that he downloaded the package). Then, in the MIR, (3) a module called *a general wrapper* analyzes the execution in the script into sub-executions, and communicates with other general wrappers in other cells for controlling the overall execution. In each MIR, (4) each general wrapper passes

its sub-execution to a corresponding submodule called *an application-specific wrapper* for processing. (5) A new WCA generated as a result of execution is sent back to the user.

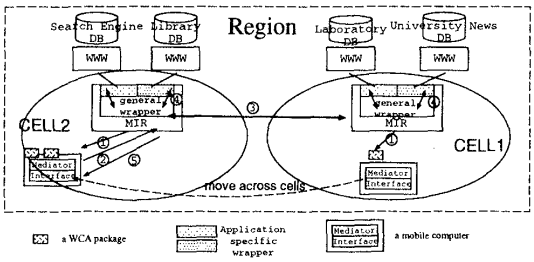

Figure 3: The System Architecture

3.2 A Domain Hierarchy

In the wide-area system, the semantic conflict between WISs must be solved. To do so, we propose a multi-level data representation called a *Domain Hierarchy*. The *domain* in our approach refers to a description of a pool of values that have an exact data-presentation and meaning. The domain hierarchy contains two components as follows:

1) A common domain hierarchy (CDH): it is a set of domains that are used to define the representative data, in meaning and/or format, shared by all cells in the region.

2) A local domain hierarchy (LDH): it is a set of domains that describe data of the WISs in each cell. Therefore, in each cell, one LDH is maintained.

Figure 4 shows these components in a region consisting of two cells.

The domains defined in the CDH and the LDHs are constructed to form a hierarchy structure by *links*. The link in our approach is constructed from a relationship between two domains. One domain has several relationships with other domains, and each relationship has two functions to convert semantic of data between two domains. For a given domain, one link is selected as *a hierarchy link* and the domain associated with this link will be specified as its *parent* domain. The parent domain is determined so that it should represent: (i) a common data format, or (ii) common meaning of the child domain. The other links are called *point-to-point links*. The hierarchy structure of domains in the region is constructed by using hierarchy links in the CDH and the LDHs. For each domain, the hierarchy link, the point-to-point links and their conversion functions will be described in a rule-base language called *a domain rule*.

To wrap a WCA of a WIS into a common data model, we use a common form called a *interface definition*. One WCA is wrapped by a set of interface definitions. The common data model[1] used by the interface definition is a variant of the object-oriented (OO) model extended with the concept of the domain hierarchy. To link data used in the WCA with the domain in the domain hierarchy, domains are attached to attributes and arguments of the methods specified in the interface definitions.

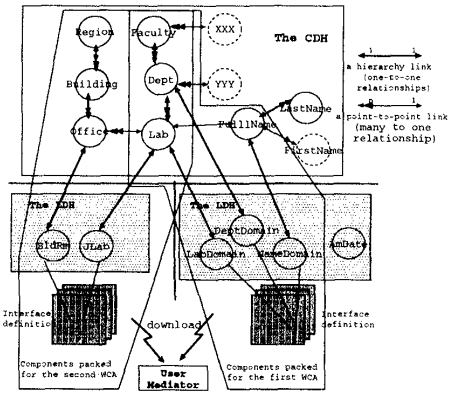

Figure 4: Processing in User Mediator

3.3 Semantic Conflict Resolution

As described earlier, a user who wants to create a new WCA from existing WCAs must download the packages from the MIRs. The package of each WCA consists of the following components:

(1) A set of interface definitions that wrap the original WCA.

(2) Domain rules for those domains which are directly specified in the interface definitions.

(3) The domain rules for the ancestor of the domains of (2).

Next, let us call the path that traverses between a given domain X and its ancestor via their hierarchy links a *hierarchy path* of X. The semantic conflict between two given domains X and Y can be solved by a mediator on the mobile computer if there is a domain that has links with the domains existing on the hierarchy paths of both given domains X and Y. In Figure 4, the paths between $Jlab \rightarrow Lab \rightarrow Dept \rightarrow Faculty$ and $LabDomain \rightarrow Lab \rightarrow Dept \rightarrow Faculty$ are the hierarchy paths of the domain *JLab* and the domain *LabDomain*, respectively. The conflict between the domain *Jlab* and the domain *LabDomain* can be solved because the domain *Lab* has links as described above. The data can be converted in $LabDomain \rightarrow Lab \rightarrow JLab$ order. Likewise, the conversion between data of the domain *NameDomain* and the domain *BldRm* can be solved using the path $NameDomian \rightarrow FullName \rightarrow Lab \rightarrow Office \rightarrow BldRm$.

4 Summary

In this paper, we described the system architecture for mobile users to integrate information among several independently-developed WISs. We also described the domain hierarchy for solving the semantic conflict due to the autonomous heterogeneity of WISs.

References

[1] S.T. Wisut, T. OHMORI and M. HOSHI: Ingetration of Web Information Sources by Mobile Users. *Data Engineering Workshop 99 (DEWS99)*, September 1997