3T−9

# Integrating Web Information Sources for Mobile Users(2) — Document Application

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

## 1 Introduction

We are currently developing a new information-networking environment based on WWWs; our objective is to realize an environment where a mobile user moves in a WWW network, finds multiple autonoumous (and heterogenous) Web-Information Sources (WIS), and integrates them on his mobile computer so that he can use them as a new customized WIS [1]. This paper describes two document applications of our system; one is navigational integration of WISs, and the other is to merge Web-documents.

## 2 Example of Navigational Integration

Firstly, we describe an example of navigational integration of WISs. Let us assume two WISs here; the first one is a normal WWW document server; i.e., it consists of a Web top-page (as a *Web client application*, WCA) and some HTML-pages (under a given DTD) describing departments; further, the top-page includes URL links to these HTML-pages. The second WIS is the CGI-based page (as a WCA), through which a user can input condition parameters to retrieve a dynamic result page. These WISs and their *domain hierarchy structure* are nothing but those displayed in Figure 2 of [1].

As described in [1], our system must wrap a *Web client application* (WCA) part of a given WIS into a common data model called *interface definition*. (Exactly speaking, we must wrap the *Data-Access part* of a WCA of a given WIS.) Here, note that nested data structure in a given WCA can be represented as relationships between interface definitions through the multi-valued attributes. Consequently, because the data access part of the WCA of the first WIS is nothing but a structured document, we can wrap this data-access part by a set of interface definitions. Figure 1 shows the DTD of the first WCA and its equivalent schema in our approach. The schema of the WCA for the second WIS is a relational model, and thus can be represented by one interface definition.

Now, assume that a mobile user finds these two WISs and wants to do navigational integration of them. To do so, we provide a SQL-like query language. The following script is a command to do navigational integration from a given (single) department page of the first WIS to the second WIS:

```
select *
from   D in Department, L in LabBuilding
source Department of ISInformation
       on http://HOST1/cgi-bin/execute.pl,
       LabBuilding of uecdbWeb
       on http://HOST2/cgi-bin/execute.pl
where  L in LabBuilding->getByLaboratory(
       D:LabD2Lab(L:Lab2JLab(
        D->{laboratories}->{labName}))
```

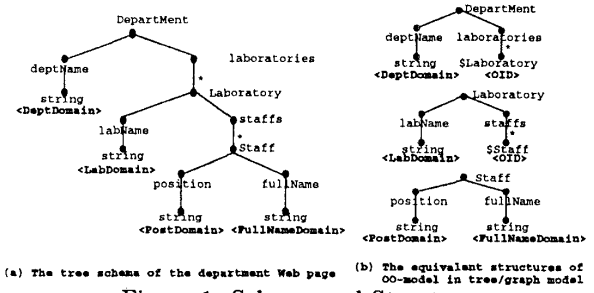


Figure 1: Schema and Structure

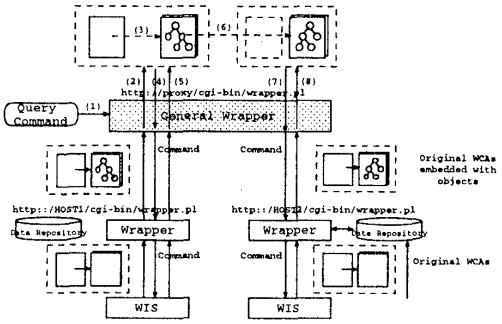


Figure 2: Integration Process

Note that the above script executes a navigational integration between the *Data-Access parts* of the two WISs. As described in [1], there arises a semantic conflict problem here. To solve this problem, we have developed a user-interface with a meditor facility, which is working on a user's mobile computer. That is, behind the user-interface, a *mediator* is running on a user's computer, and it solves the semantic conflict by using the domain hierarchy [1]. Then, by the help of the mediator, a user can generate the above query command easily. For example, in the above SQL-like statement, *getByLaboratory* is the method defined in the interface definition of the second WCA; this method is the major function of this WCA. The LabD2Lab and Lab2JLab functions are the one to convert data of the domain *LabDomain* into the domain *Lab* and the one to convert the domain *Lab* into the domain *JLab*, respectively.

As a result, the above SQL-like script can execute navigational integration (of the Data-Access parts) with resolving semantic conflicts.

On the other hand, in order to build a new WIS from the two given WISs, we must also connect the above SQL-like script with the user-interface part of the first WIS. Fortunately, this connection can be easily done because there arises no semantic conflict. Our current prototype supports a function to execute this connection on a mobile computer.

Until here, we have described how a mobile user collects some WISs and does a navigational integration of them as a new WIS. Hereafter, let us describe what happens when the user executes the newly-built WIS on his browser. In this case, the DA-part (i.e,, the SQL-like script described above) of the WIS is issued to a general wrapper. Then the overall execution is done according to the steps (1) to (8) in Figure 2.

## 3 Example of Web Document Merging

Another example of our system is a case of merging Web documents from different WISs. This topic is intensively discussed in many researches [2], but most of them discussed the case having no semantic conflict. In contrast, we here describe merging heterogenous documents by resolving their semantic conflicts. To do so, we can use the domain hierarchy here.

Assume that there are two Web documents named NIKKEI and ABCNews with different structures existing in two cells. Also, let us assume that a mobile user has collected interface definitions of these documents. Note that these *interface definitions* correspond to the DTDs of the original documents. Structures of these two documents are shown in the following figure.

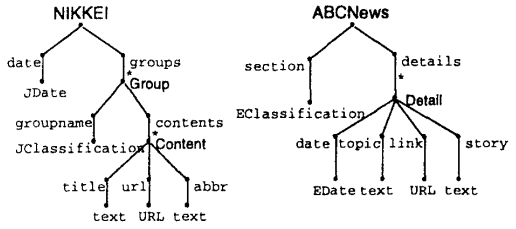


Figure 3: Structures of Original Documents

These two documents have different structures: the first one has the sorted order by the attribute "date" and "groupname", while the other is grouped by "section" of the news. There are also semantic conflict: NIKKEI contains news articles in Japanese, whose issued date of each news has Japanese format of 'yyyy 年 mm 月 dd 日'. On the other hand, date in the document ABCNews has American format of 'mm/dd/yyyy.

The common data model used in our system is described in the paper[1]. Because this data-model can be seen as a nested relational structure, we adopt here nested relational algebra operations on these data objects. We use the following main operators:

- *nest* Sorting and grouping by specified attributes and newly create nested sub objects
- *unnest* Unnest a nested part
- *reorder* Reorder attributes on specified class
- *union* Heterogeneous union

There are other operators for adding, deleting, renaming attributes and filtering by specified conditions.

Using the above preparation, our solution for document integration is the following steps: (1) firstly, we use unnest operator to drop nested part of two original objects into plain tuples until attributes needed in the integration are all in the same level of structure. (2) next, we reorder the attributes of one side to let them have the same structure. (3) After that, by a union operation, we get heterogeneous-union data objects. (4) Finally we use nest operator to restore the nested part. As a result, we can form a new structured document as a final result. Furthermore, when executing these steps, we design our system in such a way that Domain Hierarchy Module[1] is invoked if a nest operator is called and if domains of specified grouping-/sorting-attribute are heterogenous. By this design, we can resolve the semantic conflit automatically according to the domain hierarchy module.

In summary, a mobile user is requested to write the following sequence of nested relational operators in order to realize the integration of NIKKEI and ABCNews:

```
@X0=retrieveDoc("NIKKEI");
@X1=unnest(\@X0, "", groups);
@X2=unnest(\@X1, "", contents);
@Y0=retrieveDoc("ABCNews");
@Y1=unnest(\@Y0, "", details);
reorder(\@Y1, "", "date,section,topic,link,story");
@U1=union(\@X2, \@Y1, 'NIKKEI');
@U2=nest(\@U1, "", "date", "groups[groupname,title,
    url,abbr]");
@U3=nest(\@U2, "groups", "groupname", "contents
    [title,url,abbr]");
```

Domain Hierarchy used for this application is illustrated in the following figure.

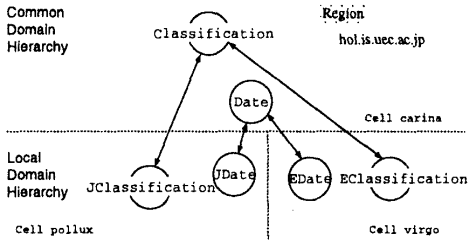


Figure 4: Domain Hierarchy

As a result, by using this domain hierarchy, the above script can merge two heterogenous documents, and generates a new document with a homogenous representation about the domains Date and Classification.

References

[1] Wisut S.T., et al., "Integrating Web Information Sources for Mobile Users(1)", *58th IPSJ Annual Conv.*, 3T-8, 1999.

[2] D.Florescu, et al., "Database Techniques for the World Wide Web: A Survey", ACM SIGMOD Record Vol.27(3), pp.59-74, 1998.