Linking Web Objects with Typed links

Kenji Takahashi[1], Kenji Ota[1], and Kazuchika Sekiya[2]
[1]NTT Software Laboratories, [2]NTT Software Corporation
e-mail: {kt, ota}@nttlab.com, sekiya@ntts.com

1. Introduction

We describe a mechanism to relate web objects with typed links. The typed links enables us to formally represent relationships between web objects. More importantly, we can interconnect a set of web objects with typed links and define a structure or meanings of the set as a whole. Such an interconnected set of web objects could be an object class hierarchy or a knowledge representation. This mechanism incorporates object-oriented computing and knowledge representation frameworks into WWW.

We are developing a hypermedia system for collaborative document production by fully utilizing typed links. The links are best expressed and managed the by object technology, such as ODMS (object data management system). Our implementation includes a client and a server that can handle typed links and submit web objects into an ODMS by using LINK, UNLINK, PUT, POST, LOCK, and UNLOCK methods and related entity-header fields in HTTP.

Our approach is an amalgam of WWW and object technologies — using HTTP for communication, and using object technology for computing and management. The typed link is the glue to tie the two technologies.

2. References vs Links

Typed links are defined as LINK entity-header fields of HTTP. Users can define types of LINK as relationship name, "rel". This is not usually shown in browsers, and servers and browsers can handle the LINK fields without parsing and interpreting the HTML bodies.

Currently relationships between web objects are represented by references, namely `<HREF>` in HTML. For example, suppose that there are two versions of a web objects: V1 and V2. The relationship between the two objects is informally represented by narrative descriptions in HTML as follows. In the HTML body of V1, "`The next version is <HREF V2>`". In the body of V2, "`The previous version is <HREF V1>`".

Whereas using typed links, the relationships are formally represented by two pre-defined links in the headers of web objects: '`previous version`' and '`next version`'.

3. Collaboration Support using Typed Links

By fully utilizing the typed links, we are developing EColabor, a hypermedia system for collaborative document production. EColabor manages all the information generated and referred to during document production, such as documents, discussions about documents, and their change history, in hypermedia. Typically the work process is represented by a series of linked web objects: documents-questions-answers-changes-new versions of the documents. Such information is are interconnected by typed links and stored in an ODMS.

EColabor consists of clients and a server. The clients communicate the server with an enhanced HTTP, and the server interacts the ODMS. The enhanced HTTP includes LINK, UNLINK PUT, LOCK, UNLOCK, and conditional GET.

Several functions are provided using typed links, such as version control and work flow management. For version control, EColabor automatically renews the version number of a web object when the object is put into the ODMS.

For work flow management, EColabor keeps track of changes in topology of the web objects (i.e., how web objects are linked), which represents status of work process. For example, EColabor sends an e-mail message to relevant users when a version of web object is put into the ODMS.

We have developed the prototype of both of the client and server that provide simple version control functions. The client has been developed by using libwww and MetaCard, a GUI builder. The server has been written in PERL and uses EROOT, a persistent object library.

4. Future Work

We are implementing all the functions of EColabor, using a CORBA compliant ODMS, such as ObjectStore, and Java. We will also work on formal reasoning over the web resources linked with typed links. We are preparing proposals of standards for the link related fields and methods in HTTP, such as LINK and UNLINK methods.