UDC 621.395.74:651.55:681.32

XML Projects in Japan and Fujitsu's Approach to XLink/XPointer

●Toshimitsu Suzuki ●Masatomo Goto

(Manuscript received June 2, 2000)

The Extensible Markup Language (XML)[1] is a markup language developed in response to a recommendation by the World Wide Web Consortium (W3C).[2] It is a meta language used to make an information structure. XML's original specification is the Standard Generalized Markup Language (SGML).[3] Now, XML is used not only as a format language but also as a framework in various areas beyond the SGML field. This paper describes the XML technology trends, the current state of XML technology, and some case studies in Japan and at Fujitsu. This paper also describes HyBrick,[4] which is an XML/SGML browser that was demonstrated at SGML'97, and the XLink[5]/XPointer[6] technology.

1. Introduction

The specification of XML has been in use by the W3C since 1996. XML version 1.0 was released as a W3C recommendation in February 1998. XML is a meta language for marking up information. XML's original specification is the Standard Generalized Markup Language (SGML), which is used to mark up documents for archiving and reuse. XML is used not only as a markup language but also as a framework in various areas beyond the SGML field.

The use of XML is more advanced in the U.S. and European countries because there is a much larger body of text documents in those countries and it is easy to adapt the XML technology for the World Wide Web and its contents. In Japan, XML applications have been designed and developed, and newspapers and magazines have carried XML articles since the end of 1997. Last year, many experiments that demonstrated uses of XML, for example, in financial trading and EDI trading, were conducted with the government of Japan. Although XML has yet to be applied to actual systems, various methods for its application are being verified through these experiments.

Up to now, XML has been related to information technology (IT), and because of the IT revolution it is becoming a secure part of the foundation of new IT systems. Last year, a lineup of XML parsers and other XML basic software appeared, establishing a development environment for application software. In addition, the W3C is determining XML-related peripheral specifications. The year 2000 is expected to be the year of full XML implementation.

In 1997, Fujitsu Laboratories demonstrated the world's first XML/SGML browser, HyBrick, at SGML'97 and published its first XML technical book. Since then, we have been committed to XML and its broad implementation. (HyBrick comes from work we started in 1994 to produce an original multimedia browser.) We have been seeking to develop XML specifications both independently and as a member of the World Wide Web Consortium (W3C) working group responsible for link specifications. This paper explains the current status of XML in Japan and the activities, study results, and future direction of Fujitsu and

Fujitsu Laboratories in this area.

2. XML and related work

XML-related specifications are mainly determined by the W3C and the Internet Engineering Task Force (IETF)[7] and are mostly based on XML Version 1.0, which was recommended in February 1998. From the beginning of the design stage, it has been expected that XML will become a protocol having the advantages of HTML and SGML, but not their disadvantages. Consequently, software manufacturers immediately started to develop software that supports XML. The first XML-supporting software to be released was the XML-based Channel Definition Format (CDF)[8] from Microsoft.

Microsoft installed this technology in their Web browser, Internet Explorer (IE), to realize the functions for a new information push service called "Active Channel." As a result, the number of XML studies and implementations began to increase at a fast pace, mainly in the U.S. Many companies noticed the advantages offered by XML and started putting them into their applications.

However, XML does not provide adequate functions to satisfy all of the different kinds of requirements for user applications. Therefore, related specifications were proposed to compensate for functions that were lacking and the specifications were examined extensively. The specifications that extend the ability of XML can be classified as basic specifications, application-dependent specifications, or framework specifications.

2.1 Basic specifications

The basic specifications supporting XML are:

- Namespaces for using several document type definitions (DTDs) in the same XML document,
- schema for extending the definitions of elements and definitions,
- Extensible Stylesheet Language (XSL)[9]/XSL Transformation (XSLT)[10] for defining style sheets,
- XLink/XPointer/XML Base[11]/XInclude[12] for defining links,
- XPath[13] for specifying XML structure application positions, and
- XML Query Language (XQL) and XML-QL[14] as interfaces for easy retrieval from XML documents.

The basic specifications include the Resource Description Framework (RDF),[15] which is a meta-information description framework for retrieval and content rating.

2.2 Application-dependent specifications

Some examples of application-dependent specifications are:

- Scalable Vector Graphics (SVG)[16] for the graphic format,
- XHTML[17] for the document format,
- Synchronized Multimedia Integration Language (SMIL),[18] and
- Mathematical Markup Language (MathML).[19]

In Japan, application-dependent XML specifications are used, for example, for electronic books (JepaX), documents submitted to the Ministry of Welfare about new drugs that companies want to market, legal information, and security market definitions. The Japan Electronic Publishing Association (JEPA) is responsible for JepaX.[20] In addition, Japanese companies are leading the study of domestic and international XML applications that add information to static images (Dig35).[21]

2.3 Framework specifications

XML. ORG,[22] which is hosted by the Organization for the Advancement of Structured Information Standards (OASIS),[23] provides repository and registry functions for multiple DTDs. A similar framework is BizTalk[24] hosted by Microsoft. This framework defines the message format between individual systems.

A variety of industries are promoting the standardization of their own vocabularies as industrial frameworks. Some typical examples are

cXML,[25] ebXML,[26] eCo,[27] and CBL.[28] Various industries in Japan have also started this process of standardization, but not on a full scale. The Japanese Standard Association (JIS)[29] has recognized the necessity of standard vocabularies and has started looking for a common framework.

3. XML in Japan

The New Media Development Association (NMDA)[30] in Japan constructed an electronic application system in 1998; this was the first example of XML system construction and operation. Since then, many other experiments have been carried out. Furthermore, XML-only mailing lists have been set up and run on a volunteer basis and the first vendor of XML-dedicated software has appeared. These events have generated even more XML activity. Details about some of the attempts to establish general support of XML in Japan are given below.

3.1 Project for electronic applications and notices in administrative procedures

The project for electronic applications and notices is aimed at developing a highly efficient system that can electronically process a series of administrative procedures. The procedures are creating, filing, and examining written applications; registering data; and sending notifications after receiving the applications. In this project, a database consisting of application forms was written in XML. Fujitsu Laboratories participated in this NMDA project and took charge of the processing section for application form input and display. We developed a function for XML display in an existing browser and XML input from an existing browser. The system developed in this project was proved to be effective and practical in an experiment at the Database Center for the project. However, electronic applications are not permitted under existing governmental rules and local ordinances about the formats of administrative application forms. Nonetheless, the system appears to be more convenient than the conventional paper-based systems.

3.2 Electronic catalog auto-collection system

NTT Communications is experimenting with an online auto-collection system for XML-based liquor catalogs that breweries create and place on their Web sites.[31] The system automatically collects the catalogs and then provides optimum merchandise information according to retrieval requests from customers.

3.3 Information delivery service

An XML-based personal information delivery service[32] is currently being demonstrated by Dai Nippon Printing Co., Ltd., Otsuka Shokai Co., Ltd., and NTT. The purpose of this service is to customize financial information for personal use and supply that information to individual and institutional investors. This service features style sheets that are used to customize the layout of contents for each user and embedded links and pointers in the supplied information. This enables each user to have continuous access to the latest information.

3.4 Digital contents (BML) for television sets

The Broadcast Markup Language (BML) is an example of an XML application in a TV-contents format for satellite digital broadcasting. Fujitsu is also a member of the BML project. The U.S. is creating BHTML, and the Association of Radio Industries and Businesses (ARIB)[33] in Japan is creating a protocol.

The ARIB proposed the first BML draft in the middle of 1999 to establish acceptance of the final specifications before full-scale implementation toward the end of 2000.

BML produces contents using XML-based XHTML and XSLT. When contents are received, the TV set runs different scripts (ECMScrpt) embedded in the contents and processes data for the

screen layout and a variety of buttons. Another language, BXML, is being created as a contents format that permits the design of arbitrary tags. BXML is designed to be superior to BML. Fujitsu participated in creating specifications for the protocol and is currently developing software for its processing. Experimental broadcasts using BML are scheduled to start in September.

3.5 Electronic disclosure system

The Electronic Data Gathering, Analysis, and Retrieval (EDGAR)[34] database in the U.S. is a well-known system for releasing information to stockholders. A review committee at the Ministry of Finance is proposing the Electronic Disclosure for Investors' Network (EDINET) as a Japanese version of EDGAR. To support EDINET, financial institutions are constructing an XML-based system. This system collects settlement briefs and other materials released electronically from listed companies to supply XML-based information to individual and institutional investors.

3.6 Information sharing between sections

Another example of an XML application by Fujitsu is a system for information sharing.[35] This system uses XML as a common data format for sharing information between several sections within a company. By using this system, designated sections release the latest data internally. Sales staff can thereby obtain necessary information, make estimates from the latest merchandise lists, and check data consistency on-site. Then, notices based on order data can be sent to business support systems in the accounting section, production section, and other sections. This system links business sections to make processing easy and reduce total costs.

4. Our approach to XML

We have been involved in the development of XML technology since we attended an event held for XML developers called SGML'97. Our approach is explained below.

Through the research and development of the XML/SGML browser, Fujitsu Laboratories recognized the importance of basic XML technologies. Our browser used to have an SGML parser for XML processing. However, considering the performance and the need to support high processing speeds and standard interfaces, we designed and installed parsers that had a lightweight W3C standard interface such as Simple API for XML[36] (SAX) or Document Object Model[37] (DOM). Fujitsu Laboratories programmed the parser in Java and also in C++. Because of the Java programming, we participated in the Java Community Process (JCP)[38] proposed by Sun Microsystems and supported a Java API for XML Parsing (JAXP).[39] Our Java-version parser is a completely compatibility parser that supports JAXP.

Fujitsu Laboratories is also participating in other leading edge projects in Japan.

As well as the aforementioned NMDA, we joined the financial foundation project and the trading EDI project proposed in the third supplementary budget of the Information-technology Promotion Agency (IPA) of Japan[40] to become involved in system development using XML technology. The financial foundation project produced a system for processing the United Nations rules for Electronic Data Interchange for Administration (UN/EDIFACT) information in XML. The technologies developed in this project have commercial promise.

Regarding XML-related products, Fujitsu has already developed and marketed an SGML Editor.[41] This SGML Editor supports the Document Style Semantics and Specification Language (DSSSL - ISO 10179)[42] for WYSIWYG previews. This software has a DSSSL style editor so that DSSSL style sheets can be easily created. In addition, since XML can be specified when saving, it is easy to convert an SGML document into an XML document.

5. Our approach to XLink/XPointer

Our research and development activities also cover the XML-related standards, Xlink and XPointer. We have constructed several systems[43),44)] while trying to realize the potential of hyperlinks with these standards. This chapter describes these systems: but first, the characteristics of XLink and XPointer are described, followed by an explanation of the performance realized with the two standards.

XLink and XPointer are description languages essential for constructing hyperlinks. The W3C working group responsible for linking is mapping out XLink and XPointer, but both languages are still in the working draft stage.

XLink is a hyperlink description language that provides flexible and extensible XML-defined hyperlink functions such as HTML's Anchor (A) and Image (IMG) in XML. XLink enables resources to be related or traversed by different methods. An HTML link has a structure consisting of only a link start position and a jump destination. XLink, on the other hand, can describe not only a one-to-one link from one document to another but also a link between three or more documents.

Therefore, even operations that currently have to be defined with a script can be programmed easily using an XLink link. For example, for a function to display a pull-down menu and jump to a specified location, these operations must be described in a Java script and each jump destination must be embedded in the source code as a script. Programming the same operation in XLink, however, only requires each link to be defined as an XML document, like other contents in the same XML document.

XPointer is a language for pointing to a specific location or fragment in an XML document. The pointer can be defined immediately after a Uniform Resource Identifier (URI) (RFC2396) as either a fragment ID or as a character string to describe a link destination using XLink. XPointer will provide essential and important functions in XML.

The XLink Working Group used to define this standard using a unique syntax. However, since XPointer can perform processing equivalent to the expressions used in the XML document conversion language, a common syntax between the two was defined. Therefore, XPointer is currently defined with additional functions and limitations on this syntax. This syntax is called XPath.

We examined the effectiveness of applying these two standards in two specific areas:

– Hyperlink functions running on IE, Netscape Navigator, and other browsers, and
– server software applications running on CGI, ASP, and other kinds of servers.

The HyBrick system and Link Server Page system were constructed to verify the effectiveness. While creating contents, we found the process of defining links to be more difficult than anticipated. We solved this problem by creating a link editor that has a Graphical User Interface (GUI), which is required in any software application for generating links.

However, a more important factor is the lack of individual link functions in the aforementioned three systems. Therefore, an XLink/XPointer link function is incorporated as a common module into each system. This function makes it easy to construct systems.

The four applications explained below contain this link processor.

5.1 HyBrick

Current WWW systems support very simple hyperlinks. Only a one-to-one directional relationship from one specific location to another can be defined. To define multiple links from one specific location, links have to be described in a line of code or script has to be written. In addition, a unique document cannot be created based on references, and a unique link to a document on a CD-ROM or other documents cannot be changed, because such links must be predefined in a resource document. Unique browser functions must be incorporated into a system and provided be-

fore such processing is possible. Incorporation as an application-dependent function is also the only method for a link function that returns control from a hyperlink destination to the original position. To solve these problems, HyBrick has a browser that provides the following three types of XLink-defined hyperlink functions:

– Link descriptions in a document that are different from the displayed descriptions,
– bidirectional links, and
– links to several locations.

HyBrick is a document browser with an XLink/XPointer processing function that uses DSSSL style sheets to display XML/SGML documents. **Figure 1** shows an example of a HyBrick window. The location defined in the link is underlined.

Similar to an ordinary WWW browser, HyBrick jumps (hyperjumps) to the location defined in the link when the link is clicked. However, hyperlink processing differs depending on the link. If the link points to a single target, HyBrick jumps directly to the corresponding page and displays it, similar to an ordinary WWW browser. If the link is connected to several destinations, the browser displays a dialog box listing jump destinations, as shown in **Figure 2**.

For a hyperjump, a destination is selected from the jump destination list. Since this dialog box displays character strings representing jump destinations (defined as link attributes), link destinations can easily be specified. Moreover, functions can be added for security, copyright protection, and other kinds of processing. The security function allows a security level to be set for each jump destination so that jump destinations can be displayed according to their security levels. The copyright protection function automatically embeds a copyright mark into a document when it is accessed. Furthermore, since links from other locations can be defined in a dialog box, several links may be defined together at a specific link location. **Figure 3** shows a link list dialog box.

To jump to a target, a corresponding link is selected from this dialog box, then a jump destination is selected from the locator list dialog box that is displayed next. Thus, links with multiple destinations can be generated from a single resource.

HyBrick is currently available as freeware and has many users.

5.2 Link Server Page

As a server application, a hyperlink effective-

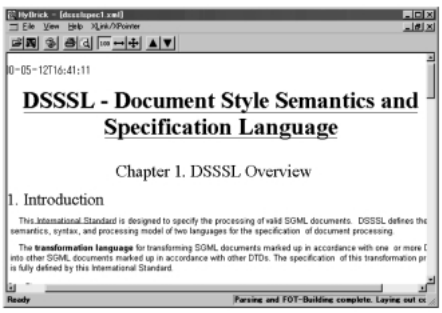

Figure 1
HyBrick window.

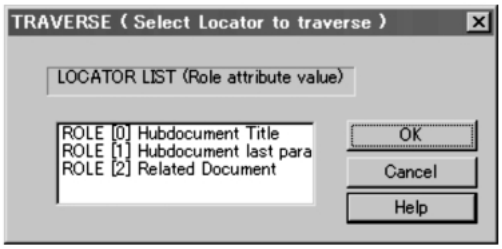

Figure 2
Locator list dialog box.

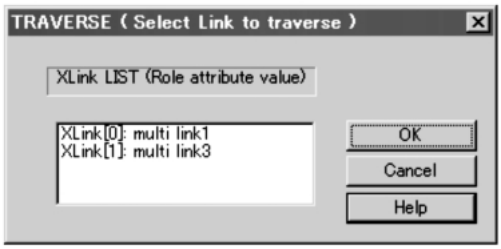

Figure 3
Link list dialog box.

ly solves certain problems in the WWW.

The people who operate the WWW can roughly be classified into three categories:

– The content providers who provide information,
– the portal administrators who organize information before providing it to users, and
– the end users who reference the information on a client.

The XLink system described above effectively serves content providers and portal administrators. Also, this system is useful for end users.

In any system, it is generally very costly to maintain contents so that users continue to receive the latest data. A portal administrator must always create a copy of information transmitted from a content provider in a local server, update the information periodically, and organize information into individualized formats. XLink realizes this processing dynamically. The Link Server Page (LSP) allows a content provider to concentrate on server management and other work. Under this system, any user can access the latest information, even before the local server is updated.

This system is called Link Server Page because its configuration is similar to that of Active Server Page (ASP)[45] and Java Server Page (JSP).[46]

Figure 4 shows a conceptual diagram of LSP operation.

To access a portal site, a client usually specifies a specific URL. This system judges whether a corresponding file is a link document. If the file is a link document (e.g., skelton.xml), the system accesses documents that have contents (e.g., content1.xml, content2.xml, and content3.xml) and then obtains and embeds the necessary information. Then, the newly integrated document (e.g., result.xml) is sent back to the client.

Under this system, a portal administrator only needs to manage link information and portal pages, and a content provider is free to provide contents, irrespective of the format and time. In addition, user information can be managed and used to customize contents for each user.

5.3 Link editor

Since XLink is described in the XML syntax, only programmers and technicians with specific know-how can create hyperlinks with complicated link information. We therefore designed and installed a GUI link editor to help users without that specific know-how to create hyperlinks. As **Figure 5** shows, the link editor consists of two panes.

The link information pane on the left displays link information, and the contents display pane on the right is used to create and browse links.

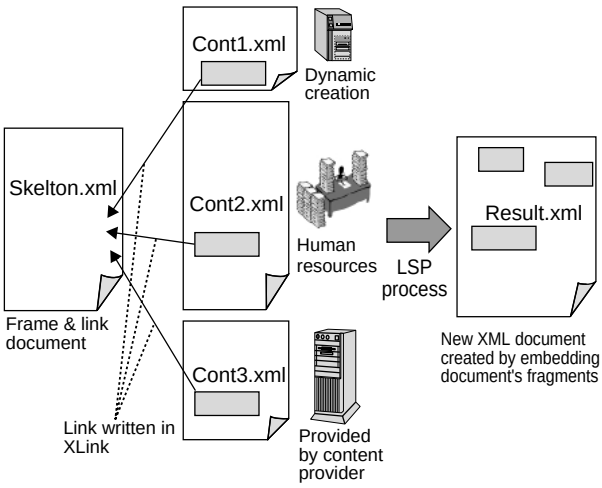

Figure 4
LSP operation.

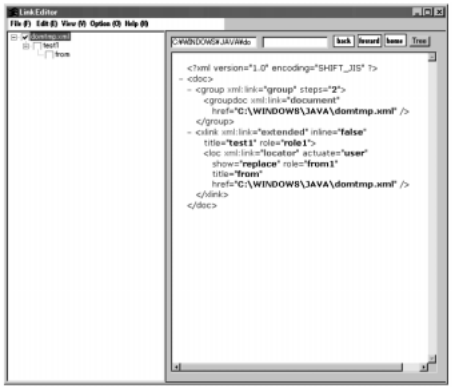

Figure 5
Link editor start window.

When a link document is opened, link information is displayed in a tree format and action in the pane creates a link. The following is an example of link creation.

First, "Create Link" is selected from the menu bar to display the dialog box shown in **Figure 6** and the required information for link creation is entered.

Then, a resource to link to the created link is defined. Next, the document in the contents information pane is displayed and the link destination is selected and highlighted as shown in **Figure 7**.

Then, "Create Locator" is selected from the menu bar to display the Locator creation dialog box shown in **Figure 8** and the required information is entered to create a link.

This semi-automatic GUI system for link description makes it easy to create link documents.

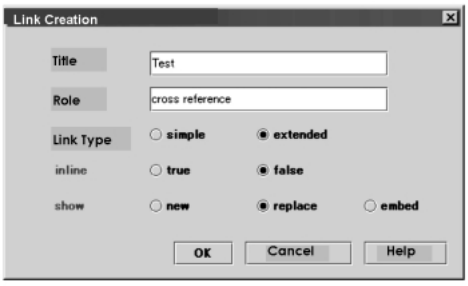

Figure 6
Link creation dialog box.

Figure 7
Selecting a link destination.

5.4 XLink/XPointer processor[40)]

The XLink/XPointer processor (XLP) is installed on 100% Pure Java to process XLink/XPointer on DOM. This processor constructs a link processing model by preprocessing after DOM construction. A DOM-like application programming interface (API) has been set up as a unique link processing model to help describe hyperlinks specific to individual applications. This module is called XLink processor and is available as freeware from Fujitsu's home page.

6. Future work

At present, Fujitsu is developing an XML parser (DOM/SAX), XLink, XPointer, XSLT, and other basic technologies separately in a software development kit (SDK).

Now that XML is poised for explosive growth in the world of Internet technologies, we will efficiently integrate these basic technologies and develop proprietary systems. Especially regarding XLink, we will keep developing practical applications from multiple points of view. We will continue to develop the use of XML as a basic technology for realizing hyperjumps from hyperlinks and document display on the WWW.

XML is now used mainly for data exchange and requires compatibility with document formats. To meet this requirement, a method of combining XSLT or XSL processing must be examined.

7. Conclusion

XML has the dual role of being a document

Figure 8
Locator creation dialog box.

notation and a data exchange format. As described in this paper, XML is mainly used as a data exchange format to access the same data from different kinds of platforms and applications and to execute processing based on this data. New hyperlink applications will conform to this trend.

Currently, XSL is popular as a data exchange format, but if it is standardized, its use for document notation or as a substitute for HTML will expand. However, in the current situation, basic HTML technology for hyperlinks remains essential, and XLink and XPointer will become more important in the foreseeable future.

Lastly, a few words about Minimal XML. Minimal XML is especially aimed at data exchange and features simple and fast parse processing to eliminate attributes and entity references. Keep an eye on the development of this protocol because it is suitable for use in Electronic Data Interchange (EDI).

References

1) World Wide Web Consortium: Extensible Markup Language (XML) 1.0. W3C Recommendation.
http://www.w3.org/TR/REC-xml
2) World Wide Web Consortium.
http://www.w3.org
3) ISO 8879:1986 Information processing – Text and office systems – Standard Generalized Markup Language (SGML).
4) HyBrick: SGML/XML Browser supporting XLink/XPointer.
http://www.fujitsu.co.jp/hypertext/free/HyBrick/en/index.html
5) World Wide Web Consortium: XML Linking Language (XLink) W3C Working Draft.
http://www.w3.org/TR/xlink
6) World Wide Web Consortium: XML Pointer Language (XPointer) W3C Working Draft.
http://www.w3.org/TR/xptr
7) IETF: Internet Engineering Task Force.
http://www.ietf.org
8) Microsoft: Channel Definition Format (CDF).
http://msdn.microsoft.com/workshop/delivery/cdf/reference/channels.asp
9) World Wide Web Consortium: Extensible Stylesheet Language (XSL).
http://www.w3.org/TR/xst
10) World Wide Web Consortium: XSL Transformation (XSLT).
http://www.w3.org/TR/xslt
11) World Wide Web Consortium: XML Base.
http://www.w3.org/TR/xmlbase
12) World Wide Web Consortium: XML Includesions (XInclude).
http://www/w3/org/TR/xinclude
13) World Wide Web Consortium: XML Path Language (XPath). W3C Recommendation.
http://www.w3.org/TR/REC-xpath
14) World Wide Web Consortium: XML-QL: A Query Language for XML.
http://www.w3.org/TR/NOTE-xml-ql
15) World Wide Web Consortium: Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation.
http://www.w3.org/TR/REC-rdf-syntax
16) World Wide Web Consortium: Scalable Vector Graphics (SVG) 1.0 Specification.
http://www.w3.org/TR/SVG
17) World Wide Web Consortium: XHTML 1.0: The Extensible HyperText Markup Langage – A Reformulation of HTML 4 in XML 1.0. W3C Recommendation.
http://www.w3.org/TR/xhtml1
18) World Wide Web Consortium: Synchronized Multimedia Interface Language (SMIL).
http://www.w3.org/TR/REC-smil
19) World Wide Web Consortium: Mathematical Markup Language (MathML).
http://www.w3.org/TR/REC-MathML
20) JepaX.
http://x.jepa.or.jp/ks/dish/jepax
21) Digital Imaging Group (Dig35).
http://www.degitalimaging.org
22) XML.org.
http://www.xml.org
23) Organization for the Advancement of Struc-

tured Information Standards (OASIS).
http://www.oasis-open.org
24) BizTalk.
http://biztalk.org
25) Commerce XML (cXML).
http://www.cxml.org/home
26) Electronic Business XML Institute (ebXML).
http://www.ebxml.org
27) CommerceNet (eCo).
http://www.commerce.net
28) Commerce One (CBL).
http://www.commerceone.com
29) Japanese Standards Association (JIS).
http://202.248.220.3
30) New Media Development Association (NMDA).
http://www.nmda.or.jp/index-english.html
31) Osake no Tansu.
http://mom.abw.ntt.ocn.ne.jp
32) The Internet Personal Direct Mail Example.
http://www.dnp.co.jp/jis/news/99/990305.html
33) Association of Radio Industries and Businesses (ARIB).
http://www.arib.or.jp
34) The Electronic Data Gathering, Analysis, and Retrieval (EDGAR).
http://www.sec.gov/edgarhp.htm
35) XML SDK (in Japanese).
http://www.fujitsu.co.jp/jp/soft/xmlsdk/index.html
36) Simple API for XML (SAX).
http://www.megginson.com/SAX
37) World Wide Web Consortium: Document Object Model (DOM).
http://www.w3.org/TR/REC-DOM-Level-1
38) Java Community Process Program (JCP).
http://java.sun.com/aboutJava/communityprocess
39) Java API for XML Parsing (JAXP).
http://java.sun.com/aboutJava/communityprocess/final/jsr005
40) Information-technology Promotion Agency, Japan (IPA).
http://www.ipa.go.jp
41) Fujitsu: SGML Editor (in Japanese).
http://ikb.solnet.se.fujitsu.co.jp/pcwin/hs_guide/soft/win32/kaihatu/04831000.html
42) ISO/IEC: 10179, Document Style Semantics and Specification Language (DSSSL).
ftp://ftp.ornl.gov/pub/sgml/WG8/DSSSL
43) M. Goto.: An Implementation Design of XLL as a Subset Of HyTime.
SGML/XML Europe'98, pp.461-470.
44) XLink Processor. Japanese page only.
http://www.fujitsu.co.jp/hypertext/free/xlp
45) Microsoft: Active Server Page (ASP).
http://msdn.microsoft.com/workshop/server/asp/ASPover.asp
46) Sun Microsystems: JavaServer Page (JSP).
http://java.sun.com/products/jsp

Toshimitsu Suzuki received the B.E. and M.E. degrees in Electro-Communication Engineering from the University of Electro-Communication, Tokyo, Japan in 1985 and 1987, respectively.
He joined Fujitsu Laboratories Ltd., Kawasaki, Japan in 1987. Since then, he has been involved in research and development of communication services and human interfaces, including hardware architecture. His current research interests include document markup, style sheets, and linking.

E-mail: tsuzu@jp.fujitsu.com

Masatomo Goto received the B.S. degree in Electronic Engineering from Kyushu University, Fukuoka, Japan in 1992.
He joined Fujitsu Laboratories Ltd., Kawasaki, Japan in 1992 and has been engaged in research and development of macro language on a text editor. He moved within Fujitsu Laboratories to Akashi, Japan and since then has been engaged in research and development of SGML, Hytime, and XML systems. He is a member of the XLinking Working Group of the W3C.

E-mail: mg@jp.fujitsu.com