The Valid Web: an XML/XSL Infrastructure for Temporal Management of Web Documents

Fabio Grandi **Federica Mandreoli**

C.S.I.TE. – C.N.R. and Dipartimento di Elettronica, Informatica e Sistemistica
Università di Bologna, Viale Risorgimento 2, I-40136 Bologna, Italy
Tel: +39 051 209.3548, Fax: +39 051 209.3540, Email: {fgrandi,fmandreoli}@deis.unibo.it

Abstract

In this paper we present a temporal extension of the World Wide Web based on a complete XML/XSL infrastructure to support valid time. The proposed technique (borrowed from the basic temporal database theory) enables the explicit definition of temporal information within HTML/XML documents, whose contents can then be selectively accessed according to their valid time. By acting on a navigation validity context, the proposed solution makes it possible to "travel in time" in a given virtual environment with any XML-compliant browser; this allows, for instance, to cut personalized visit routes for a specific epoch in a virtual museum or a digital historical library, to visualize the evolution of an archaeological site through successives ages, to selectively access past issues of magazines, to browse historical time series (e.g. stock quote archives), etc. The proposed Web extensions have been tested on a demo prototype showing, as application example, the functionalities of a temporal Web museum.

1 Introduction

A great deal of work has been done in recent years in the field of Temporal Databases (TDBs) [15, 9, 26, 7, 16]. Due to this effort, a large infrastructure (namely data models, query languages, index structures, etc.) has been developed for the management of data evolving in time, for which successive versions need to be maintained rather than being overwritten or discarded by destructive changes. However, research interests on temporal information have been almost focused on highly structured data (e.g. relational or object-oriented), and, for instance, no textual data or less structured multimedia documents have been diffusely considered for temporal extensions so far.

On the other hand, the World Wide Web (WWW, W3 or Web [2]) is a large distributed collection of hypertextual and multimedia irregular documents, currently formatted according to the HTML standard [20], available on-line on the Internet. The new emerging standard for publishing documents over the Web, which has been recently recommended by the W3C Consortium [25], is the eXtensible Markup Language (XML [18]), which has been designed to overcome the main limitations of HTML (and is also suitable to describe semistructured data). The browsing of XML documents can be best enjoyed by means of provided *stylesheets*: in particular, the eXtensible Stylesheet Language (XSL [19]) allows the definition of XML document transformation rules [30] and the specification of formatting semantics. Although the functionalities of the Web, including markup language potentialities and browser capabilities, have been lately greatly increased, scarce attention has been so far devoted to the *temporal* aspects, whereas Web documents may also contain intrinsically temporal (i.e. *historical*) information.

Our proposal concerns the introduction of valid time into the Web to support the management of historical information, borrowing the basic timestamping and temporal selection techniques introduced in TDB theory. To this purpose, in Sec. 2, we will define a complete XML/XSL infrastructure to embody temporal information into Web documents and put temporal navigation and query facilities at Web user's disposal. The solution we propose does not require changes in the current Web technology as it is based on XML and related standards. The adoption of a suitable XML schema for document timestamping and a provided XSL stylesheet for temporally selective document processing will enable any XML-compliant browser, like Microsoft Internet Explorer 5 (Ie5 [24]), to support temporal documents.

The proposed Web extensions have also been tested on a prototype implementation [5] which will be briefly described in Sec. 3. Our implementation also largely exploits another powerful Web technology supported by Ie5: the Document Object Model (DOM, [17]), which is an application programming interface (API), which allows dynamic manipulation of HTML and XML documents (e.g. via scripting languages like JavaScript [23]).

Conclusions can finally be found in Sec. 4.

2 Integrating Valid Time into the Web

In this Section we outline our proposal concerning an XML/XSL infrastructure for the definition and use of valid-time temporal Web documents. Generally speaking, the adoption of valid time is aimed at allowing the management of historical information (past, present or future). Historical information must explicitly be coded within the Web documents, to be selectively accessed during temporal browsing. To this purpose, distinct parts of a Web document can be timestamped with their own validity during the document creation.

2.1 Defining Valid Temporal Documents

The addition of valid time to Web documents we propose is based on the extensions of the XML markup language [18] with timestamping tags. The functionalities of the new tags can be fully specified by means of suitable XML schemas and stylesheets, without requiring modifications to the Web browsers supporting XML (like Ie5).

In particular, our proposal consists of the addition of a new XML tag, `<valid>`, to define a *validity context*. The validity context is used to timestamp a part of a document, to which it assigns a specific time pertinence which can then be used for temporally-selective document manipulation. The contents embraced in a validity context can be of any allowed kind (namely text, graphic elements, and any other XML structure including nested `<valid>` elements). The timestamps can be specified in a validity context by means of `<validity>` tags, which allow the definition of a temporal interval through its boundaries (i.e. the values of the `from` and `to` attributes of the `<validity>` XML element). In general, multiple intervals can be used: in this case, the timestamp is defined as the union of all the validity intervals specified; formally, the timestamp is a *temporal element* as defined in the BCDM temporal data model [8]. For instance, the following code:

```
<valid>  <!-- definition of a validity context -->
   <validity from="1980-01-01" to="1985-12-31" />
   <validity from="1995-01-01" to="2000-12-31" />

   This is text <b>valid from 1980 to 1985</b>
      but also <b>valid from 1995 to 2000</b>...

</valid>
```

```
<?xml version="1.0" ?>
<Schema xmlns="urn:schemas-microsoft-com:xml-data"
        xmlns:dt="urn:schemas-microsoft-com:datatypes">

   <AttributeType name="from" required="yes" dt:type="date" />
   <AttributeType name="to" required="yes" dt:type="date" />

   <ElementType name="validity">
      <attribute type="from" minOccurs="1" maxOccurs="1" />
      <attribute type="to" minOccurs="1" maxOccurs="1" />
   </ElementType>

   <ElementType name="valid" content="mixed">
      <element type="validity" minOccurs="1" maxOccurs="*" />
   </ElementType>

</Schema>
```

Figure 1: The `ValidSchema.xml` XML schema.

defines a validity context whose validity is $[1980\text{–}1985] \cup [1995\text{–}2000]$. The time constants are specified according to the ISO 8601 format [3], corresponding to the XML `date` data type. The introduction of the validity context, including the definition of the required new tags, is effected via the XML schema, named `ValidSchema.xml`, displayed in Fig. 1. Such a schema must be included in any XML temporal document to enable validity contexts, in the way usual for XML-Data [27] schemas:

```
<?xml version="1.0" ?>
<TemporalDoc xmlns:dt="x-schema:ValidSchema.xml">

        the rest of the document, with timestamped <valid> elements,
        including any XML and (well-formed) HTML mark-up

</TemporalDoc>
```

Notice that the use of an XML schema ensures syntactic checks on the well-formedness of temporal documents to be automatically effected by the XML-enabled browers[1]. The adoption of an XML schema instead of a Document Type Definition (DTD, [12]) is due to its flexibility and extensibility, in addition to the availability of predefined data types. Unlike a DTD, an XML schema is based on an open content model, and thus it can be applied to any kind of documents (irregular data), also containing XML elements defined by other schemas or even not defined anywhere. In this way, our proposal concerns a way of adding the timestamping facility to generic XML documents, also just containing plain HTML code. Any existing HTML-

[1]This also applies to the date constants specified as `from`/`to` attribute values. Unfortunately, the Ie5 parser and DOM do not currently support the `date` type and, thus, do not effect correctness checks on date values when applying the XML schema.

based Web site can thus easily be made temporal, by adding XML `<valid>` timestamps to its documents in accordance to the `ValidSchema.xml` schema above.

Therefore, the purpose of our proposed timestamping schema is (at least) twofold:

- it can be used to make temporal "legacy" Web sites (featuring HTML multimedia documents), in order to support the representation of historical information, and enabling a temporally selective navigation with respect to information validity;
- it can be used to make temporal the emerging deployment of XML for data representation and exchange on the Web, in order to support the management of temporal structured or semistructured data, and enabling the utilization of functionalities as developed by temporal database research (e.g. TSQL2-like temporal query languages);

Furthermore, any other kind of new XML-based application can be made temporal in the same way. Hence, they will benefit from the representation of historical information and temporal data management capabilities.

Referring to our reference application, the standard HTML pages composing the Web Museum must be converted to XML documents making reference to the `ValidSchema.xml` schema as described above. Moreover, their original HTML markup needs some checks for conversion into *well-formed* HTML code. This phase of Web site re-engineering can largely be automated. The human designer intervention is indeed required for the addition of timestamping, as the pieces of information to be enclosed in `<valid>` environments have to be carefully identified and appropriate time values have to be assigned in `<validity>` timestamps.

The order in which the different `<validity>` tags composing a temporal-element timestamp appear (and their individual position within the `<valid>` environment) is irrelevant. Only conformance to the `ValidSchema.xml` schema is required.

2.2 Temporal Browsing and Navigation

The default valid time used for the Web navigation is usually the whole time range, enabling to view the full contents of documents, which corresponds to the basic use of non-temporal (standard) documents. For a selective temporal navigation, the time range can be reduced by the user (e.g. via some browser facility). It does not seem too restrictive to adopt a time *interval* as validity context, since also most queries in valid-time databases are commonly based on the comparison with an interval. Once set up by the user, such validity interval is known by the Web client as a *navigation validity context*. When temporal documents are processed, only the parts whose valid timestamp *overlaps* the navigation validity context are effectively taken into account and displayed. The same is automatically applied when new documents are retrieved by following a link, enabling a full-fledged temporal navigation.

In our proposal, the valid-time selection relies on the adoption of an XSL stylesheet [19], named `Valid.xsl`, to perform a dynamic filtering of the document contents according to the navigation context. Such stylesheet embeds XSL transformations [30] and adopts the XML path language [28] facilities implemented in Ie5 (see [29] for reference). The definition of the stylesheet can be seen in Fig. 2: the first part consists of a simple identity-transformation template, whereas the second part is devoted to the temporal selection of the contents of validity contexts. The processing of the new XML `<valid>` element causes the output of the element contents when a validity selection condition (involving the document element timestamp) is verified. For instance, if the condition has the form:

```
@from[ .  $le$ '1999-12-31' ] and @to[ .  $ge$ '1999-01-01' ] ,
```

```
<?xml version="1.0" ?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">

<!-- identity transformation template -->
<xsl:template>
   <xsl:copy>
      <xsl:apply-templates select="@*|*|comment()|pi()|text()" />
   </xsl:copy>
</xsl:template>

<!-- recursive valid-time selection template -->
<xsl:template match="valid">
   <xsl:choose>
      <xsl:when test="validity[condition involving from and to attribute values]">
         <xsl:copy>
            <xsl:apply-templates select="@*|*|comment()|pi()|text()" />
         </xsl:copy>
      </xsl:when>
      <xsl:otherwise>
         <xsl:apply-templates select="*//valid" />
      </xsl:otherwise>
   </xsl:choose>
</xsl:template>

</xsl:stylesheet>
```

Figure 2: The XSL Valid.xsl stylesheet.

each <valid> element whose validity overlaps year 1999 is included in the stylesheet output: the selection condition matches any <validity> element where the from attribute value is $\leq$1999/12/31 and the to attribute value is $\geq$1999/1/1.

The particular structure of the selection template causes the execution of a test involving the navigation context and all the <validity> timestamps found in the current <valid> element. The conditional processing uses the xsl:choose instruction which provides for an xsl:otherwise case (not supported by the xsl:if XSL element), in order to recursively look for nested validity contexts. The xsl:when instruction is activated if at least one of the intervals (corresponding to a validity element) belonging to the timestamp satisfies the selection condition. The xsl:otherwise instruction is activated only when none of the timestamps of the current <valid> environment satisfies the selection condition.

Notice that, if the navigation validity context is changed by the user during his/her navigation, such a condition should dynamically be changed in the stylesheet. If the stylesheet is then re-applied to the document, also the document visualization changes to reflect the user's action. For instance, in our prototype implementation, which is based on Ie5, the stylesheet is actually changed to update the selection condition and re-applied to the document by means of the DOM methods' functionalities. In this way, the change is actually effected on the stylesheet copy loaded (as an XML document object) in the main memory space managed by the browser

```
// Find the validity test condition in the stylesheet.
var sel = document.XSLDocument.selectSingleNode("//xsl:when/@test");

// Replace its value with the 1999 overlap condition.
sel.value ="validity[ @from[.$le$'1999-12-31'] and @to[.$ge$'1999-01-01'] ]";

// Apply the modified stylesheet to the document, and update the display.
document.body.innerHTML = document.XMLDocument.transformNode(document.XSLDocument)
```

Figure 3: A DOM script for dynamic change of the navigation validity context.

or on a copy of the stylesheet cached on a local disk of the machine on which the browser is running. For example, the overlap with 1999 condition above can dynamically be applied to the displayed document by means of the JavaScript code shown in Fig. 3.

Notice also that more complex temporal selections than the simple overlap could be implemented by applying a different condition or even by defining a different filtering template in the stylesheet. Therefore, also sophisticated temporal query and reasoning facilities could easily be added to a temporal Web site by means of appropriate direct management of the `Valid.xsl` temporal selection condition. As a matter of fact, as it is based on the Ie5 XSL support (with Microsoft XSLT [30] and XPath [28] extensions) our proposal represents a straightforward temporal extension of the XQL query language [13]; extension that has been designed as an application of XQL itself. For instance, in a related application [4], we showed how XSL filters can be used to support all the TSQL2-like selection predicates [14] over XML temporal data.

3 A Reference Application: the Temporal Web Museum

The proposal outlined in this paper concerns the addition of the valid-time dimension to (HTML/XML) Web documents and is finalized to support temporal navigation in virtual environments which are sources of historical information. An extremely appropriate example of such an environment is a Web museum, which has been used as reference application.

In a virtual museum, temporal selective browsing allows the definition of personalized visit paths through centuries and artistic or historical periods within the museum collections. In order to plan a visit (virtual or real), we can act on valid time selection to change the historical period of interest. For instance, we can choose the High Renaissance period, by selecting the validity range 1495–1520. Hence, we may start our virtual visit entering some virtual hall or gallery: only the temporally relevant paintings or sculptures would be present; by changing the validity context we could see some works vanish and some different works materialize. For example, in a hall dedicated to the Italian High Renaissance, we could view the evolution of the painting styles of Leonardo da Vinci, Raphael, Michelangelo and Titian and, say, have a look to works contemporaneous to the Mona Lisa picture.

The interest for museum applications on the Internet is constantly growing. This is shown by the increasing number of available museum sites and by the development of a specific discipline [1], with dedicated journals (e.g. *Archives and Museum Informatics*) and conferences (e.g. *Museums and the Web*). The "Web Museum" [11], authored and maintained by Nicholas Pioch, was one of the very first to open and is probably the most popular virtual museum on-line. It is basically a collection of image data representing famous paintings, heterogeneous as to their origin, which can accessed, for example, via an artist or a theme index. In order to test our proposal, we realized a temporal version of a subset of the Web Museum pages and developed a Web environment for the temporal browsing of its collections.

The temporal Web museum has been implemented as a software prototype to test and validate our proposed valid-time XML/XSL temporal extensions. The prototype, named "The Valid Web" [5], consists of a Web site which can be browsed with Ie5 (see Fig. 5). The Web pages of the site are organized in two frames. A small service frame in the bottom part of the window contains all the required controls to deal with the user interactive specification of the validity context to be used for temporal navigation, including the visualization of the current validity context. All the controls are implemented as JavaScript functions. A larger frame, occupying almost all the browser window space, is used to display temporal documents, that is the results of the temporally selective filtering effected by the `Valid.xsl` stylesheet on timestamped XML documents. The results of such filtering is a plain HTML document which is then rendered by the browser in the "usual" way.

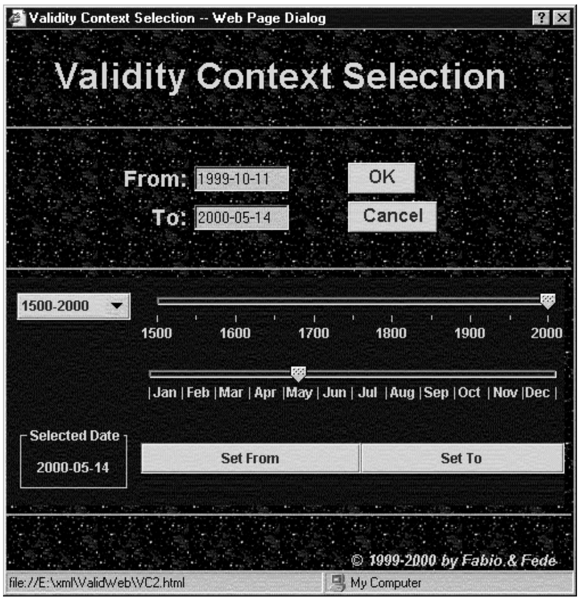

Figure 4: The Java Applet for the selection of a Validity Context.

In general, the valid-time selection implies the choice of an interval. This can be done by an independent choice of the two time points representing the interval bounds. The selection of each interval bound can be based, for instance, on a graphic *scrollbar* or *slider* for analog fine selection of a time-point (at a given granularity level). In our prototype implementation, time-points are dates (i.e. the granularity level is the day) and the selection of an interval can be effected by means of a Java JFC/Swing applet [22, 6], which contains two graphic sliders: the former to select the year and the latter to select the day of the year (see Fig. 4). The former slider, for the user's convenience, has a 500-year range, which can be changed (from 0–500 to 2000–2500) by means of a *multiple-choice menu* available next to the year slider. Assume we have to fix a date, say 1996/3/7. We can start by choosing the year 1996 with the former slider (with the default range 1500–2000 set) and then choose the March, 7 date with the latter slider. The chosen value can then be assigned to the From or To interval bound by means of the corresponding "Set" *button*. However, also editable input fields for direct typing of a valid time value (in the "`YYYY-MM-DD`" string format) are always available in the dialog window containing the running applet. The communication between the applet and the JavaScript control functions in the calling service frame (e.g. to return the selected validity context) is managed by means of the LiveConnect package [10] supported by the Java Plug-in 1.2.2 [21].

Once the navigation validity context choice is confirmed by the user, the temporal selection over the currently displayed document is automatically re-executed by means of the DOM mechanism described in the previous Section. Furthermore, in order to enable a full-fledged

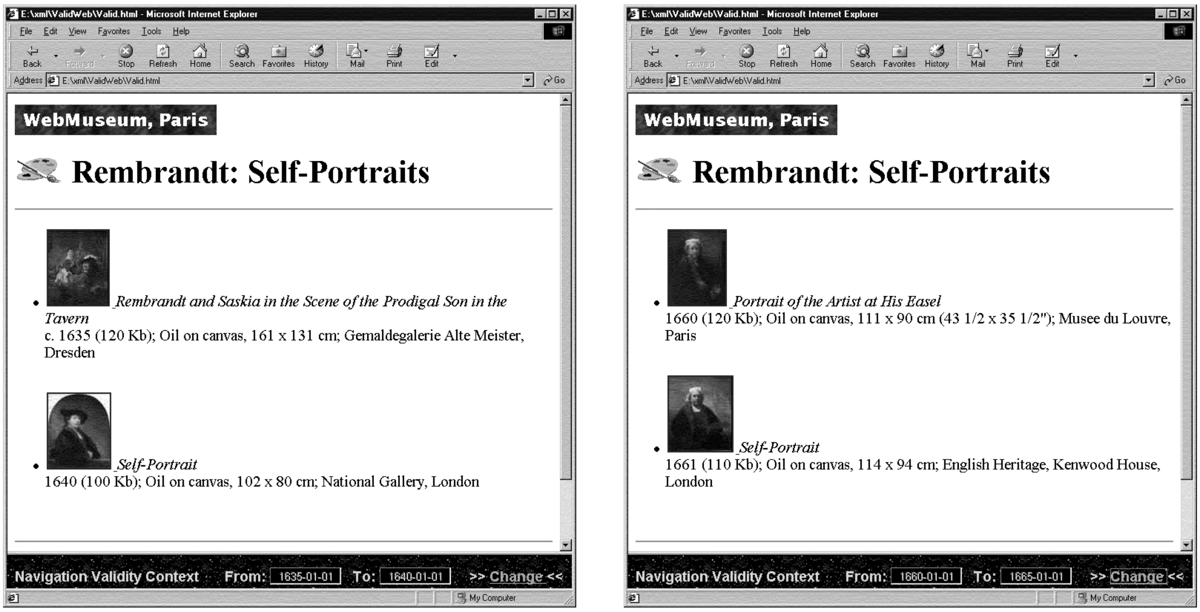

Figure 5: Temporal navigation of the Web Museum.

temporal navigation, each time the user changes the displayed document in the usual way (e.g. by following a link), the current validity context is automatically "inherited" by the newly loaded page, if also the new document is a temporal XML one. This behaviour is forced in our prototype thanks to the dynamic HTML facilities supported by Ie5. In fact, we used a slightly modified `Valid.xsl` stylesheet with respect to Fig. 2. The actual stylesheet implements a dynamic callback mechanism by inserting some JavaScript code in the preamble of the processed document. Such a script provokes, on load of the document by the browser, the immediate activation of the temporal selection functionality: as when the validity context is changed, the `Valid.xsl` filter is updated on the fly (to include the overlap with the current validity context as selection condition) and then re-applied to the displayed document by means of DOM method calls similar to the ones in Fig. 3. In this way, the valid-time selection is automatically propagated, in a transparent way, also to the newly loaded temporal documents.

For example, Fig. 5 shows two snapshots of the navigation of a sample page containing Rembrandt's self-portraits. The full page contains seven pictures, dating from 1629 to 1669. Fig. 5 shows on the left the page when the validity context has been set to [1635–1640] and only two pictures are visible (the third and the fourth one) and on the right the same page with the validity context changed to [1660–1665] and two other pictures have been displayed (the fifth and the sixth one). The current navigation context is always visible in the bottom service frame of the window. The "Change" command on the right is a link that activates the applet of Fig. 4.

4 Conclusions

In this paper we outlined an approach for the integration of valid time into the Web. We addressed the issues concerning the extensions required to document structure and format, document processing and temporal browser usage. We basically sketched some feasible solutions and gave an idea of their potentialities by means of the application to a Web Museum. The solution we proposed relies on simple XML-based extensions: the introduction of a new `<valid>` markup tag with the definition of an XML schema for the creation of temporal documents, and the use of an XSL stylesheet for selective filtering of temporal documents according to a user-defined navigation validity context. We also developed a prototype temporal Web site, accessible with Ie5, which implements the proposed solution and provides tools for the management of the validity context with a friendly user interface. Also legacy HTML-based Web sites can easily be made temporal by converting them into XML documents and adding the required timestamps to the historical multimedia information they contain. Our proposal is fully compatible with the nowadays Web technology and can be enjoyed by millions of users all over the world.

Notice that the adoption and use of the time dimension is a powerful weapon to improve the selectivity of certain information searches over the net, which could also enhance the usefulness and functionality of already available navigation aids (e.g. Web search engines). The setting of a suitable validity coordinate in browsing a Web site containing historical information (e.g. a digital library or a virtual museum, but also an archive of newspaper issues or stock quotes) would improve the search quality: only relevant information would be displayed, instead of being immersed in a mass of temporally heterogeneous and non-pertinent stuff. This is made possible by the temporal semantics, which is carried by the new timestamping XML tags and allows the selection of information pieces in a temporally marked multimedia document via the evaluation of even complex temporal predicates. This is very different indeed from the textual search for matching date strings available with a search engine.

Future work will be devoted to the extension of the presented insfrastructure to include more sophisticated temporal representation and retrieval facilities with the addition, for instance, of a TSQL2-like support [14] for temporal indeterminacy, multiple granularities and calendars.

References

[1] Archives & Museums Informatics, URL: `http://www.archimuse.com`.

[2] T. Berners-Lee, R. Cailliau, A. Lautonen, H.F. Nielsen, A. Secret, "The World Wide Web," *Communications of the ACM*, Vol. 37, No. 8, 1994.

[3] Date and Time Formats, W3C Note, URL: `http://www.w3.org/TR/NOTE-datetime`.

[4] F. Grandi, F. Mandreoli, "The Valid Web: it's Time to Go...", TR-46, a TimeCenter Technical Report, December 1999, *available from*
URL: `http://www.cs.auc.dk/research/DP/tdb/TimeCenter/`.

[5] F. Grandi, F. Mandreoli, "The Valid Web ©", *Proc. of Software Demonstrations Track at the EDBT'2000 Intl. Conference*, Konstanz, Germany, March 2000.

[6] Java Foundation Classes (JFC), Sun Microsystems,
URL: `http://java.sun.com/products/jfc/`.

[7] C.S. Jensen, J. Clifford, R. Elmasri, S.K. Gadia, P. Hayes, S. Jajodia (eds.) *et al.*, "A Consensus Glossary of Temporal Database Concepts - February 1998 Version," in O. Etzion, S. Jajodia and S. Sripada (eds.), *Temporal Databases - Research and practice*, LNCS N. 1399, Springer-Verlag, 1998.

[8] C.S. Jensen, M.D. Soo, R.T. Snodgrass, "Unifying Temporal Data Models via a Conceptual Model," *Information Systems*, Vol. 19, N. 7, 1994.

[9] N. Kline, "An Update of the Temporal Database Bibliography," *ACM SIGMOD Record*, Vol. 22, N. 4, 1993.

[10] LiveConnect, in *JavaScript Guide*, Ch. 5, Netscape Communications, URL: http://developer.netscape.com/docs/manuals/communicator/jsguide4/livecon.htm.

[11] N. Pioch, "The Web Museum," URL: http://www.cnam.fr/wm/.

[12] Prolog and Document Type Declaration, in *Extensible Markup Language (XML) 1.0*, W3C Recommendation, URL: http://www.w3.org/TR/REC-xml#sec-prolog-dtd.

[13] J. Robie, J. Lapp, D. Schach, "XML Query Language (XQL)," *Proc. of QL'98 - The W3C Query Languages Workshop*, Boston, MA, Dec. 1998, URL: http://www.w3.org/TandS/QL/QL98/pp/xql.html.

[14] R.T. Snodgrass (ed.) *et al.*, *The TSQL2 Temporal Query Language*, Kluwer Academic Publishers, Boston, Massachussets, 1995.

[15] M.D. Soo, "Bibliography on Temporal Databases," *ACM SIGMOD Record*, Vol. 20, N. 1, 1991.

[16] A. Tansel, J. Clifford, V. Gadia, S. Jajodia, A. Segev, R.T. Snodgrass (eds.), *Temporal Databases: Theory, Design and Implementation*, Benjamin/Cummings Publishing Company, Redwood City, California, 1993.

[17] The Document Object Model (DOM) Home Page, W3C, URL: http://www.w3.org/DOM/.

[18] The Extensible Markup Language (XML) Resource Page, W3C, URL: http://www.w3.org/XML/.

[19] The Extensible Stylesheet Language (XSL) Resource Page, W3C, URL: http://www.w3.org/Style/XSL/.

[20] The HyperText Markup Language (HTML) Home Page, W3C, URL: http://www.w3.org/MarkUp/.

[21] The Java Plug-in Home Page, Sun Microsystems, URL: http://java.sun.com/products/plugin/

[22] The Java Technology Resource Page, Sun Microsystems, URL: http://java.sun.com.

[23] The JavaScript Resource Page, Netscape Communications, URL: http://developer.netscape.com/tech/javascript/.

[24] The Microsoft Internet Explorer Home Page, Microsoft, URL: http://microsoft.com/windows/Ie/.

[25] The World Wide Web Consortium (W3C) Home Page, URL: http://www.w3.org/.

[26] V.J. Tsostras, A. Kumar, "Temporal Database Bibliography Update," *ACM SIGMOD Record*, Vol. 25, N. 1, 1996.

[27] XML-Data, W3C Note, URL: http://www.w3.org/TR/1998/NOTE-XML-data.

[28] XML Path Language (XPath) Version 1.0, W3C Working Draft, URL: http://www.w3.org/TR/xpath.

[29] XSL Developer's Guide, The Microsoft Developer Network, URL: http://msdn.microsoft.com/xml/xslguide/.

[30] XSL Transformations (XSLT) Version 1.0, W3C Working Draft, URL: http://www.w3.org/TR/xslt.