Extending Temporal Database Concepts to the World Wide Web

Fabio Grandi **Maria Rita Scalas**

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 (0)51 644.3548, Fax: +39 (0)51 644.3540, Email: {fgrandi,mrscalas}@deis.unibo.it

Abstract

The development of the World Wide Web technology on the Internet is a major achievement of computer research in recent years. The great availability of easy-to-access on-line hypermedial documents has been a real revolution in the information world, also for its important social, cultural and economical consequences. However, a scarce attention has been devoted so far to the temporal aspects of the World Wide Web definition and technology. Moreover, although the management of time-varying information is a consolidated research issue in the database field, no integrations between the two worlds has been attempted yet.

In this work, we try to make a first step towards a temporal extension of the World Wide Web, by considering the impact of well-known temporal database concepts on the design and management of Web documents, starting from the very first notions of transaction time and valid time. By means of a reference application (a virtual Web museum of fine arts), we will try to explore the potentialities of the approach and sketch research directions in which new solutions have to be sought.

In particular, the integration of transaction time into the World Wide Web would allow, in a transparent way, the management of successive versions of the same documents. For instance, by acting on transaction time within a "virtual museum," it will be possible to visit different temporary special exhibitions dressed in different times at the museum (i.e. navigate through succesive museum's versions).

On the other hand, the adoption of valid-time versioning would allow the explicit definition of temporal information within documents, whose contents will be selectively indexed on the basis of the validity coded within them. By acting on valid time, it will be possible to navigate through time in a given environment; for instance to visualize the evolution of an archaeological site through successives ages, or cut personalized visit routes for a specific epoch in a museum.

This work has been in part supported by the C.N.R. "Progetto Finalizzato Beni Culturali" (subproject: Hierarchical representation of museum environments and planning of paths for spatio-temporal virtual visits) and by the M.U.R.S.T. 40% "Progetto Basi di Dati Evolute: Modelli Metodi e Sistemi" (subproject: Multimedia databases for museum environments: models and algorithms).

1 Introduction

A great deal of work has been done in recent years in the field of Temporal Databases (TDBs) [18, 11, 21, 9, 20]. Due to this effort, a large infrastucture (namely data models, query languages, index structures, etc.) has been developed for the management of data evolving in time, for which succesive versions need to be maintained rather than being discarded by destructive changes. Two well-known to date kinds of time are usually considered in the literature [9]:

- **Transaction-time**:

  which concerns the data evolution with respect to the system where data are stored;

- **Valid-time**:

  which concerns the data evolution with respect to the application reality that data describe.

By the way, 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.

The World Wide Web (WWW or Web) [3] is a large distributed collection of hypertextual and multimedia semi-structured documents, available on-line on the Internet. The Web documents are written according to the HTML standard [4], which is a markup language based on the SGML formalism [8]. From a system point of view, the Web is based on an open, distributed software architecture based on the client-server paradigm, with network support [6] relying on the TCP/IP communication protocol. In a few words, information servers (network sites) store Web documents and deliver them on request to clients (e.g. Web browsers). Virtual navigation from site to site is made possible, in a transparent way, by following hypertextual links (Uniform Resource Locators) [5] found in browsing the documents.

The rapidly growing popularity of the Internet, chiefly due to the Web itself, has gained momentum for a consistent investment of intellectual (and finacial) resources on research and development of its technology. Although the funtionalities of the Web, including HTML potentialities and browser capabilities, have been lately greatly increased, scarce attention has been so far devoted to the *temporal* aspects.

The only acknowledged necessity in the Web community is for the support of document *versioning*, particularly in a scenario of concurrent document editing [22] (see also the bibliography for previous work). Hence, a working group —on the WWW Distributed Authoring and Versioning (WebDAV) mailing list <w3c-dist-auth@w3.org>— is very active on the subject and is preparing a consensus working document (Internet draft) [16] on versioning support requirements to be submitted to WWW standardization committees.

The kind of versioning considered there is on the system side: the maintainance of versions reflects the history of modifications applied to the documents, as committed by authors to the document store (server). In TDB terminology, the temporal dimension involved is, thus, *transaction time*. However, Web documents may also contain intrinsically temporal (i.e. *historical*) information, where the validity is a constitutive part of the information itself. The temporal dimension involved is, this time, *valid time*. This aspect has not been taken into account with the necessary worth yet.

A third temporal dimension is already of interest in the Web community, that is the time dimension intrinsic to multimedia data like music, movies, animated presentations and others involving, *streams* of information. The issue is not indeed very central in the context of temporal databases (e.g. it has been dealt with in [7]), but has been taken into account in hypertext research. For instance, the HyTime proposal [13] mainly consists of an SGML extension to represent *time-based* documents.

This third time dimensions involves temporal aspects which can be considered as "internal," "local" to the representation and usage of some special types of data. Therefore, such a time dimension is quite orthogonal to the time dimensions considered in this work (it could be labelled a *user-defined* time dimension, in TDB terminology), although possible interactions would deserve a thoroughly investigation in future work.

Last but not least, interaction between standard database and Web technology is already an interesting and very promising research and application field [2, 12]. For example, commercial DBMS commonly now support the functionality of restructuring query results into HTML pages, whereas database techniques can be used for querying information over the Web. Several studies relate to interfacing and interoperating the two worlds, in their current non-temporal status. Therefore, also interaction between temporal databases and possibly temporal Web is a matter of concern in this framework.

This work tries to discuss some basic ideas for integration of temporal capabilities (and temporal database functionalities) into the World Wide Web specification and to draw some simple directions for their integration into the Web client-server system architecture. However, this work has to be considered as a position paper to basically define a new research project in which the new ideas have to be put into reality. We put a challenge that we are also the first to accept, although our project is still at a very preliminary stage.

2 The Reference Application: a Museum Web Server

One of the reasearch directions outlined in this paper concerns the addition of the temporal dimension(s) to hypermedia Web documents, finalized to support navigation in virtual environments. An extremely appropriate example of such an environment is a virtual museum, and will be used as a reference application in the rest of the paper.

The number of Internet museum sites available on the Web is increasing day by day. Recently, several famous real museums opened their official Web sites, as for instance:

- The Louvre Museum in Paris,
  URL: `http://mistral.culture.fr/louvre/`
- The Uffizi Galleria in Florence,
  URL: `http://www.Uffizi.Firenze.it/`
- The Vatican Museums in Rome,
  URL: `http://www.christusrex.org/www1/vaticano/0-Musei.html`
- The Guggenheim Museums (including Manhattan, Soho, Bilbao and Venice sites),
  URL: `http://www.guggenheim.org/`
- The Metropolitan Museum of New York,
  URL: `http://www.metmuseum.org/`
- The MoMa Museum in New York,
  URL: `http://www.moma.org/`

The main purpose of these sites is the presentation of the corresponding real institutions (providing general information, plans, samples from the main collections, etc.), the advertisement of current

initiatives, up to the merchandising of museum gadgets and publications. Also private or corporate art collections may be found on the Web.

On the other hand, also purely *virtual* museums, that is without a "physical" counterpart, are also avalilable on-line. For instance, one of the very first site to appear, and one of the most popular now (mirrored on several servers throughout the world), is the "Web Museum" [15], also known as the "Web Louvre" before the official Louvre site was plugged in. It is basically a collection of image data representing famous paintings, heterogeneous as to their origin, which can accessed, for instance, via an artist or a theme index.

A related application is also the realization of a virtual archaeological site, based on the finds, reliefs and reconstructions of an underlying real site, where virtual sightseeings can be possible. These can be made available through hypermedial plans (e.g. topographic, thematic, historical), or possibly implemented as *virtual reality* environments, for instance based on VRML code [19]. As a virtual reality application makes it possible an interactive spatial trip in the modeled environment, the integration of time into the Web would add also the temporal dimension to the navigation, with the site seen as changing through the ages by the user moving along the time axis.

Transaction time

The integration of *transaction time* into the Web would allow the management of the evolution of Web documents. In such a case, entire HTML files are automatically timestamped by the system when they are created or updated, enabling the rollback of Web sites to past states of their lifetime.

The application of transaction time is the navigation of a Web site through the history of its successive reorganizations. However, such reorganizations are usually due to routine upgrade of documents (e.g. textual information editing, link restructuring, improved graphic design, introduction of new HTML features), but may also reflect a real evolution of the underlying organization. For example, we may think of a museum site in which temporary exhibitions are periodically set up. Therefore, navigation of a museum site through transaction time would allow to rollback the museum to some *expired* temporary exhibition.

Furthermore, an analysis of the restructuring of some Web pages, which are updated very frequently, may give an historical perspective on the underlying organization(s). Finally, once all Web document transaction-time versions were organized in a uniform manner (e.g. as explained in the next section), a rollback navigation with an advanced browser would give interesting snapshots of the overall Web information universe at a past point in time. Tracking the Web evolution would give important cues on its development trends, leaps and stalemetes, and this would be possible through the Web itself.

Valid time

In general, the integration of *valid time* into the Web would allow the management of historical information within Web documents. In such a case, distinct HTML items can be explicitly timestamped with their own validity in the document editing phase, enabling then the selective browsing of the documents based on the validity of the information contained.

The application of valid time is the selective temporal navigation of a Web site, through the history of the "virtual world" it embodies, which may either be a purely virtual entity or may represent some real-world entity. Historical information must be explicitly coded within the Web documents, to be selectively accessed during temporal navigation. In a Web museum, this would

allow to define personalized visit paths through centuries and artistic or historical periods through the museum collections.

For example, assume to be browsing a museum Web site. To plan a visit (virtual or real) we could act on valid time to change the highlighting of halls on a museum graphic plan: only halls containings works relevant for the selected validity period would be evidenced, while the other ones could be "grayed." In this way, we can select, for instance, the paths concerning the High Renaissance period, by selecting the validity range 1495–1520. Furthermore, to start a virtual visit, we may enter one highlighted hall or gallery: only the temporally relevant paintings or sculptures would be present; changing the validity context we could see some works vanish and some different works materialise. In a hall dedicated to the Italian High Renaissance, we could thus view the evolution of the painting styles of Leonardo da Vinci, Raphael, Michelangelo and Titian and, for instance, have a look to works contemporaneous to the Mona Lisa picture.

Database integration

Special-purpose applications require that some designated Web page be intrinsically updated very often, also in a (quasi) periodic manner. This is the case, for instance, of any *news*' pages: newspapers, magazines and journals on the Web, but also stock market closure reports, and even "What's new?" sites. For instance, "What's new?" pages are often used as entry points by casual users, and virtual museum may use them (they can also be thematic, e.g. "What's new in fine arts?") to post advertising of their latest initiatives, new acquisitions, temporary exhibitions, etc. Museal institutions can also have their own bullettin boards, corporate magazines, scientific series, and will have to manage them on the Web. In such cases, we are in the presence of periodic Web page rewritings, probably done in an automatic manner, and often interacting with a companion underlying database. Since the stored information is rapidly changing, but still maintained on-line after the publication of upgrades, the nature of the underlying database and of the Web interface is intrinsically temporal. This kind of interaction between a database and the Web presents very challenging issues, as it shares temporal aspects, real-time aspects (the creation/update of a Web page may have to respect deadlines and temporal consistency constraints, it may likely involve database periodic transactions) and also active aspects (the creation/update of a Web page may be triggered by some database event and may follow complex database rules and/or may be subject to database integrity constraints).

Although all those aspects represent very stimulating and promising research topics, they will not be addressed in this paper, as it would be quite premature at the present stage of our research project. A study of the temporal aspects of the interaction between Web and database technologies will be devoted to our future work.

3 Integrating Transaction Time into the Web

A Web user accesses resources (e.g. HTML documents, JPEG images, Java applets etc.) over the Internet, usually located in files stored on local servers. However, in the prospected scenario, Web resources identified by a file name may be available in different versions on a site. Given the resource name, the resource versions can be identified by some symbolic name, by a version number, or by time. In other words, the granularity of Web data to be considered here for versioning is the *resource*

unit[1].

For the management of Web data versions with resource granularity, an interesting solution based on meta-level links was proposed in [14]. First of all, a version naming scheme is introduced to ensure accessibility from exixting browsers and easy manipulation in directory-based file systems (e.g. UNIX). Versions of an object are associated with a representative URL and with a unique directory name in the file system. Each version is then placed in the directory and can be referred to as:

representative_URL / version_number

For instance, version 3 of "doc.html," located on directory "pub" on the server "www.server.it," can be accessed by legacy browsers via the complete URL:

`http://www.server.it/pub/doc.html/3`

The representative URL alone allows access to the latest version by default. An extension to the HTTP protocol is then introduced to allow version negotiation from version-aware clients, by means of the `Content-Version` field of HTTP headers. For example, the above mentioned version can be requested via the HTTP `GET` method as follows:

```
GET http://www.server.it/pub/doc.html
    Content-Version :  "3"
```

Hence the returned resource can easily be identified on the server file system by the path name "/pub/doc.html/3."

However, in the context of distributed authoring research, also a finer granularity has been taken into account. This is due to the fact that document editing sessions may be limited to a few operations (i.e. insertions and deletions) on very limited chunks of the resource. Therefore, instead of creating a brand new file version each time, versioning information could be added to the modified document chuncks only within a single file. This can be done, for instance, by means of the Versioned Text Markup Language (VTML) proposed in [22]. VTML tags are added (as SGML comments, so that unaware browsers will ignore them) to identify local modifications, in order to code multiple document delta-versions within the same file.

Anyway, we do not initially consider this level of detail in document versioning along transaction time, since our approach deals with the management of *consolidated* resource versions and is not concerned with distributed authoring (also involving concurrency control in document editing by different people). In our present approach, a versioning granularity finer than the resource level will only be considered in valid-time versioning.

In order to identify versions, the WebDAV initiative is concerned with version numbers, also considering branching and merging version chains, giving rise to version directed acyclic graphs associated to single documents. In our perspective, we are indeed interested in time, and do not take into account, for simplicity, branching or merging transaction time. Therefore, the version graphs we consider here are linear, and parallel to the time axis.

The transaction time used for the Web navigation is usually the current time (now), or a past time point for backward navigation (rollback). As in transaction-time databases, a single time point is commonly used for temporal data selection. Such a time point, once set up by the user (e.g. via some browser command) is known by the Web client as a navigation context. When new documents are needed (i.e. when links are followed), such a context must be communicated to the server along

[1] The Web defines the *resource* as the unit of accesible information, as identified by an autonomous URL, regardless of its actual implementation on the server file system. However, a Web resource can always be, and commonly is, memorized on disk as a *file*.

with the requested resource name. The ability of recognize the correct desired resource version and deliver the appropriate file through the network must reside on the server side.

Under these assumptions, the integration of transaction time does not require, in general, HTML extensions, but only a mechanism to manipulate document versions through resource names and version tags by means of server actions. Minimal extensions are indeed required to the HTTP communication protocol to allow negotiation of the temporal context between the client and the server. For instance, a new header request field could be added to the `GET` method to specify the transaction-time context. The `Pragma` header, which is suitable to support protocol extensions, could be augmented with a new `as-of` token for transaction-time information transmission. The request for the Web resource "www.server.it/pub/doc.html" as of 1/31/96 could thus be issued as follows:

```
GET http://www.server.it/pub/doc.html
    Pragma :  as-of = "DATE 01/31/1996"
```

The format of the time constant could be one of the three accepted in the HTTP definition (namely RFC 1123, modified RFC 850, ANSI C's asctime standards). In alternative, as in our example, it could also conform with the TSQL2 syntax, even if this would mean to add a fourth possible time format to HTTP. However, in the proposed extension, the time constant is used strictly outside of the protocol stream, as it just appears within a string literal which is not interpreted by the protocol itself.

Coherently to the phylosophy of transaction-time in the temporal database field, the management of transaction time should be transparent to the user and automatically managed by the system. This means that the version timestamping must not be demanded to the user, and could rely, for instance, on file dates and times of creation/last modification as managed by the underlying file system. Following the above example with the naming scheme proposed in [14], we assume three versions of the "doc.html" file to be available on the "pub" directory of our server as:

```
03-Feb-95  doc.html/1
25-Jan-96  doc.html/2
17-Mar-97  doc.html/3
```

After the request of the version as of 1/31/96, the Web server has to check out that the desired version is the second one, since it has been created on 1/25/96 and has been substituted by the succesive version in 1997. The resource `http://www.server.it/pub/doc.html/2` will finally be returned to the requesting client. Obviously, smarter indexing techniques may be used by Web servers to identify transaction-time file versions rather than a straightforward directory search to match dates.

Moreover, transaction-time selection with the naming scheme adopted is compatible with version selection by version numbers and also with version selection by symbolic names, although symbolic names would give rise to a very non-uniform resource naming situation (every version name would come out as user-defined). A better solution would consist of superimposing symbolic version names as alias on top of the uniform naming scheme based on numbers. The alias support, for instance, could be integrated with the indexing mechanism.

4 Integrating Valid Time into the Web

Valid-time versioning is aimed to the production of *temporal* Web documents, that is containing (historical) information with different validities. Web resources need to be specially designed as

temporal, and timestamping of time-varying information has to be explicitly coded in the document creation phase by the authors.

The addition of valid time requires new HTML extensions, in order to timestamp single items within documents. A *validity context*, for example, can be defined as a new HTML *element,* delimited by a new "valid-time" *tag*, say `vt`. In the valid-time *start-tag*, "from" and "to " *attributes* can be defined to specify the endpoints of the validity interval to be used as timestamp. Thus, a text fragment valid from 1980 to 1985 could be inserted in a HTML document as follows[2]:

```
<!-- valid time timestamping -->
<vt from="DATE 01/01/1980" to="DATE 12/31/1985">

This is text <b>valid from 1980 to 1985</b>...

</vt>
```

As in the example, a syntax conforming to the TSQL2 query language can be used to express time constants. Furthermore, also time-stamping via temporal *elements* (namely disjoint unions of maximal intervals) could be employed by means of multiple interval decalarations, as in the following example:

```
<!-- valid time timestamping with elements -->
<vt from="DATE 01/01/1980" to="DATE 12/31/1985"
    from="DATE 01/01/1987" to="DATE 06/30/1987">

This is text <b>valid from 1980 to 1985</b>
and <b>in the first half of 1987</b> too...

</vt>
```

In both cases, semantic checks on interval/element consistency should be effected by the parser.

Notice that the present markup proposal is similar to the VTML approach [22], but it applies to valid- rather than to transaction-time. Whereas the document element timestamping is automatically performed by the authoring tool in a transparent way during document editing in VTML, it is completely under the user control and responsibility while authoring valid-time temporal HTML documents.

In order to make temporal HTML documents compatible with standard (snapshot) browsers, the safest solution consists in adding HTML extensions as comments (as in VTML). Only valid-time aware browsers would try to interpret any comment as a valid-time command, whereas standard browsers will simply ignore the timestamping. However, existing browsers (e.g. Netscape) do not usually complain at all about "unknown" HTML objects. In the above example, Netscape 3.0 would correctly recognize "vt" as a HTML element, "from" and "to" as HTML attributes, and the two dates as correct attribute values expressed as string literals, as it can be seen from the highlights in the "View Document Source" window. Anyway, their effect is *null* on the document processing, as they are not recognized as meaningful HTML objects yet. Therefore, the proposed addition of valid timestamping to HTML documents would not compromise, in general and in principle, their

[2]We hope that the examples which follow are comprehensible enough also without a great familiarity with the HTML format. The HTML element delimited by tags `<b>` `</b>` is used to markup boldface text.

usability by unaware browsers. In such a case, the complete contents of the document will be visible, regardless of the timestamping inside.

A similar extension, with valid timestamping of single elements within a document, can also be defined also for VRML, enabling spatio-temporal navigation in virtual enviroments, like archaeological sites on the Web. A navigation facility for the fourth dimension could be added to virtual reality browsers (or browser plug-ins) using the same techniques envisioned above (e.g. based on the temporal client paradigm and using reasource valid timestamping).

The default valid time used for the Web navigation is usually the whole time range, enabling to view the full contents of documents. For a selective temporal navigation, the time range can be restricted by the user (e.g. via some browser facility). It does not seem too restrictive to adopt a time *interval* as a validity context for document browsing, since also most queries in valid-time databases are commonly based on the overlap with an interval. Once set up by the user, such validity interval is known by the Web client as a local navigation context. When documents are processed, only the parts whose valid timestamp overlaps the validity navigation context are effectively displayed. Such a technique, that we say based on a **temporal client**, would limit the valid-time integration to the extension of HTML and to the availability of validity-aware advanced browsers. Hence, whereas transaction-time management put emphasis on the server side, valid-time management can limit its impact on the client side.

However an alternative solution may rely on valid-time processing also effected on the server side: when requesting a document, the Web client communicates also the validity context to the server, which makes a temporal restriction to the validity-qualifying parts on the documents before delivering them to the client. The advantage of this approach, that we say based on a **temporal server**, is the delivery of a smaller document (the smaller the more selective is the validity context) along the communication network. The transferred files could also be standard snapshot HTML documents (i.e. temporal HTML extensions need only to be known to the server), also in absence of a temporal context negotiation, to be managed by non-temporal browsers. Anyway, one of the disadvantages is also the charge of computational weight to the server, due to the document temporal pre-processing.

Moreover, when the validity context is changed by the user during the browsing of a document (and this is very likely to happen frequently), with the first solution the system response is trivially demanded to a browser local action on the current document, since different parts of the document are possibly involved by the new temporal selection and a screen refresh is eventually performed. With the second solution, the current document becomes out-of-date and has to be refetched: a further interaction with the server and a consequent new file trasfer over the network are needed before the new valid-time version of the document can be visualized.

If both transaction and valid time are supported, the same mechanism (HTTP extension) introduced for the transaction time can be used for the negotiation of both temporal contexts between the client and the server. Therefore, the **temporal server** solution could be the easiest to implement. On the contrary, if only valid-time support is added, the **temporal client** solution could be simpler, as minimal modifications to the browser are required, while no changes are needed in the transfer protocol and server functionalities. Notice also that with the **temporal client** solution, although transaction- and valid-time supports could share a common HTTP extension, they need different behaviours of the server to be implemented.

In conclusion, the **temporal client** solution seems more promising for efficiency reasons as it aims to reduce the overall communication traffic (for frequent temporal context changes) and to privilege local data processing. On the other hand, the **temporal server** solution may lead to a faster prototipation of valid-time support, once the transaction-time support has already been added.

As a second step, the techniques for valid-time document processing succesfully experimented on the server side could eventually be integrated on the client side. In other words, the **temporal server** solution can be used as a preliminary step to the implementation of the **temporal client** solution.

5 Temporal User Interfaces

Whichever implementation solution is adopted, there is always the necessity to locally define the temporal context(s) to be used for temporal navigation. This means that some kind of visual language for temporal information selection has to be defined and embedded in advanced Web browsers.

Transaction-time selection can be based, for instance, on a *graphic cursor* for fine selection of a time-point (at a given granularity level). The cursor selection can be helped by a *pop-up menu* (e.g. activated via right mouse button clicking) for coarse selection of the granularity level. The scale of the cursor depends on the selected granularity. Therefore, the combined use of the two tools allows the choice of any given time point. Assume we have to fix a date, say 7/3/1996. We can start by choosing the year granularity with the pop-up menu and then select year 1996 by means of the analogue cursor. Hence, we can change the granularity from year to day and finally select with the cursor the desired date. However, also a dialog window for direct typing of a transaction time value should always be avaliable.

The valid-time selection implies the choice of an interval. This can be done by an independent choice of the two interval bounds (e.g with the time-point selection tools described above for transaction time). A more friendly solution could be the use of a "range" cursor, where the cursor index has a thickness (representing the range extension) which can be changed by moving just one of its extremities, or the entire cursor index can be moved rigidly along the time line. The switch between the two index movement options could be regulated by a concurrent action on some combination of the control/shift/alt keys. The granularity in use should be the same for selecting both interval bounds, although (unusual) independent granularity selections might also be supported.

In case a multi-frame document is processed, the temporal context definition may be defined on a per frame basis, that is different frames may have different valid-time contexts selected and, thus, display different kinds of historical information. To respect the semantics of transaction-time indeed, the transaction-time context should be the same for every active window within a single running browser.

As to the semantics of the temporal visual language, we have to implement the solutions presented in sections 3 and 4. To this end, detailed solutions will be completely outlined only when the design of browser extensions will be eventually finalized. However, a first step could consist in experimenting solutions *on top* of an existing commercial browser (e.g. Netscape). This could be done by defining a dummy "container" document, composed of two frames: a large frame for actual document visualization and a tiny frame to be used for time selection via a java applet (implementing the cursors and pop-up menus described above). The request of the appropriate transaction-time document version could be managed by the Java applet by constructing the complete URL corresponding to the desired resource version. The extraction of useful temporal information (valid-time selection) could also be effected by the Java applet, which could pre-process the received document and locally produce a restricted document to be passed to the browser for visualization in the dedicated large frame. In a second step, the temporal functionalities could directly be integrated into evoluted browsers, expressely designed. During this phase, the transaction-time support should also rely on selection mechanisms already implemented on the server side (e.g. via direct HTTP daemon extensions, perl-scripts, cgi-binaries, etc.).

6 Conclusions

The pioneering works in the field of temporal databases already evidenced the pervasiveness of the temporal dimension in the information world (as a reflection of its pervasive role in the reality) and realized the complexity of its uniform management in an automated information system. The first impressions we got from the attempt to introduce the temporal dimension into the World Wide Web, which is currently considered *the universe of network-accessible information, the embodiment of human knowledge*, are not really dissimilar. The situation is even more complicated with the fact that new semantic aspects also need to be considered. While in a database the temporal evolution concerns stored information and the modeled reality, in the Web it may also concern purely virtual realities (i.e. the stored information may have an own existence completely independent from any real world entity). Moreover, multimedia data with intrinsic time dimensions are diffusely employed in Web documents. Conceptual and practical implications arising from the coexistence and interaction of so many heterogeneous temporal aspects deserve thorough and insightful studies.

Other temporal aspects, not covered by our study yet, concern the use of some kind of time attributes in current Web functionalities. The original purpose of such attributes (namely HTTP entity headers `Last-Modified`, `Expires`) is the management of cache-validation mechanisms (or the execution of conditional `GET`s with the header `If-Modified-Since`), to maintain the consistency between Web original documents and their copies locally cached by clients or proxy servers. Such mechanisms are also currently exploited to obtain some kind of special effect. For instance, the delivery of an "already expired" document by a server causes an immediat refetch by the client (client-pull), which can be used to give rise to the visualization of rapidly changing, animated documents. The value that a HTTP server must return for such attributes, can be coded within HTML documents, with the `META` element (and tags `HTTP-EQUIV` and `CONTENT`) of the document `HEAD` preamble. The possibilities open by the use of such attributes, the exact characterization of their temporal nature (thery are validities internally used by the system but which can also be specified by document authors), and their interaction with the "classical" transaction and valid time dimensions, are also very interesting topics to be addressed in future research.

A further temporal dimension is also connected with the user navigation itself: the sequence of hypertextual links followed by the user constitutes a navigation *history*, usually maintained by existing browsers, and which can also be navigated back and forth (e.g. via arrow buttons). The temporal dimension involved here is a sort of transaction time, local to the user action, and is also connected (via browser actions) to the cache-validation process. However, the kind of navigation it allows is completely orthogonal to the transaction-time navigation through resource versions we have took into account so far.

In this paper we tried to outline a first approach to the incorporation of transaction and valid time into the Web. We tried to address some of the issues concerning the extensions required to document structures, protocols and functionality of system components. We basically sketched some feasible solutions and gave an idea of their potentialities by means of the application to Web museums. In particular, for transaction-time extensions, we proposed to adopt a **temporal server** solution, requiring straightforward extenstions to the HTTP protocol and to server functionalities, and relying on a naming scheme which can be easily used on existing file systems. For valid-time extensions, we proposed a **temporal client** solution, based on very simple extensions to the HTML language and to client functionalities. Advanced Web browsers also need limited extensions to their user interface to interactively support temporal selections during navigation sessions. All the proposed solutions are perfectly compatible with the current Web technology, in the sense that all the information available on the prospected temporal Web would still be accessible via unaware browsers, although temporal

navigation facilities would clearly not be present.

Finally, the first conclusion we have to draw (but it has been announced since the Introduction) is that a great deal of work has to be done, since this paper raises more problems than it solves. A great deal of work that can be started from a prototype implementation of the solutions presented here for transaction and valid time integration into the Web. To this purpose, our temporal Web extensions could be tested, for instance, on the two experimental extensible systems for which the public distribution is supported by the Web Consortium: Amaya [1] that is a Web client supporting HTML 3.2, acting both as a browser and as an authoring tool; Jigsaw [10] that is a HTTP/1.1 compliant Web server running on a Java machine.

After all, our ultimate expectation is that new ways of netsurfing could soon experience the excitement, and benefit from the potentialities, of a travel through the time.

Acknowledgments

The authors would like to thank Barbara Pernici and Fabio Vitali for their *timely* comments on a draft of this work.

References

[1] The "Amaya" Web home page,

URL: `http://www.w3.org/pub/WWW/Amaya/`.

[2] P. Atzeni, G. Mecca, P. Merialdo, E. Tabet, "Structures in the Web," Tech. rep. RT-INF-19-1997, University of Rome III, Italy, 1997.

[3] T. Berners-Lee, R. Cailliau, A. Lautonen, H.F. Nielsen, A. Secret, "The World Wide Web," *Communications of the ACM*, Vol. 37, No. 8, pp. 76–82, August 1994.

[4] T. Berners-Lee, D. Connolly, "HyperText Markup Language Specification - 2.0," RFC 1866, MIT/LCS, November 1995.

[5] T. Berners-Lee, L. Masinter, M. McCahill, "Uniform Resource Locators (URL)," RFC 1738, CERN, Xerox PARC, University of Minnesota, December 1994.

[6] R. Fielding, J. Gettys, J.C. Mogul, H. Frystyk, T. Berners-Lee, "Hypertext Transfer Protocol – HTTP/1.1," RFC 2068, U.C. Irvine, DEC, MIT/LCS, January 1997.

[7] R. Hjelsvold, R. Midtstraum, O. Sandstå, "A Temporal Foundation of Video Databases," Proc. Intl. Workshop on Temporal Databases, Zurich, 1995.

[8] International Standard Organization, ISO-8879 Standard: Information Processing - Text and Office Systems - Standard Generalized Markup Language (SGML), October 1986.

[9] C.S. Jensen, J. Clifford, R. Elmasri, S.K. Gadia, P. Hayes, S. Jajodia (eds.), C. Dyreson, F. Grandi, W. Käfer, N. Kline, N. Lorentzos, Y. Mitsopoulos, A. Montanari, D. Nonen, E. Peressi, B. Pernici, J.F. Roddick, N.L. Sarda, M.R. Scalas, A. Segev, R.T. Snodgrass, M.D. Soo, A. Tansel, P. Tiberio, G. Wiederhold, "A Consensus Glossary of Temporal Database Concepts," *ACM SIGMOD Record*, Vol. 23, N. 1, 1994.

[10] The "Jigsaw" Web home page,

URL: `http://www.w3.org/pub/WWW/Jigsaw`.

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

[12] a. Mendelzon, G. Mihalia, T. Milo, "Querying the World Wide Web," Proc. Fourth Intl. Conf. on Parallel and Distributed Information Systems, Miami Beach, 1996.

[13] S.R. Newcombe, N.A. Kipp, V.T. Newcombe, "The "HyTime" Hypermedia/Time-based Document Structuring Language," *Communications of the ACM*, Vol. 34, No. 11, 1991.

[14] K. Ota, K. Takahashi, K. Sekiya, "Version Management with Meta-level Links via HTTP/1.1," Internet draft,

URL: `ftp://ftp.nis.garr.it/internet-drafts/draft-ota-http-version-00.txt`.

[15] N. Pioch, "Web Museum," Worl Wide Web site,

URL: `http://www.cnam.fr/wm/`.

[16] J.A. Slein, F. Vitali, E.J. Whitehead, D.G. Durand, "Requirements for Distributed Authoring and Versioning on the World Wide Web," *ACM Standard View*, 1997, *in press*.

[17] R.T. Snodgrass (ed.), I. Ahn, G. Ariav, D. Batory, J. Clifford, C.E. Dyreson, R. Elmasri, F. Grandi, C.S. Jensen, W. Käfer, N. Kline, K. Kulkarni, T.Y. Cliff Leung, N. Lorentzos, R. Ramakrishnan, J.F. Roddick, A. Segev, M.D. Soo, S.M. Sripada, *The TSQL2 Temporal Query Language*, Kluwer Academic Publishers, Boston, Massachussets, 1995.

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

[19] D. Raggett, "Extending WWW to support Platform Independent Virtual Reality,"

URL: `http://vag.vrml.org/www-vrml/concepts/raggett.html`.

[20] 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.

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

[22] F. Vitali, D.G. Durand, "Using Versioning to Support Collaboration on the WWW," *World Wide Web Journal*, Vol. 1, No. 1, Winter 1996.