Time: A Coordinate for Web Site Modelling

Paolo Atzeni

Dipartimento di Informatica e Automazione
Università di Roma Tre
Via della Vasca Navale, 79
00146 Roma, Italy
http://www.dia.uniroma3.it/~atzeni/
atzeni@dia.uniroma3.it

Abstract. The adoption of high level models has been advocated by many authors as instrumental in Web site development as a strong support to both design and maintenance. We will discuss how the specification of Web sites at the logical level can greatly benefit from the introduction of specific features for the representation of time, which could also support the notions of versions and editions of objects in the site. Moreover, time can be seen as a "coordinate" of Web models, a more general notion that includes various forms of specializations and variations, such as those related to location, language, user preferences, device type.

1 Introduction

The usefulness of high-level models for the intensional description of Web sites has been advocated by various authors, including Atzeni et al. [1, 2], Ceri et al. [3], which both propose logical models in a sort of traditional database sense, and Fernandez et al. [4], which instead propose an approach based on semistructured data. Specifically, we focus our attention on *data-intensive* Web sites, where more structured models can definitely be useful: they can be described by means of schemes (logical hypertext scheme and associated presentation) and can be obtained by applying suitable algebraic transformations to the data stored in an underlying database (see Atzeni et al. [5]).

In this context, we want to consider the issues related to the management of time-varying information, along the lines that in the database field lead to the interesting area of *temporal databases* (see Jensen and Snodgrass [6] for a recent survey). Indeed, work has already been done on temporal aspects in the Web (for example by Dyreson [7] and Grandi and Mandreoli [8]), but mainly in the context of the management of documents and with use of XML. Instead, we would like to see how a structured model for the Web could benefit from the experience in temporal databases. Specifically, we assume that the logical structure of a Web site can be described by means of a complex-object data model with object identity, with some degree of flexibility, but all described in the scheme. We (Atzeni et al. [1]) have proposed such a model, called ADM, with the following features:

- pages with the same structure and different content are described by a *page scheme*, with the URL as identifier and a set of attributes;
- attributes can be (i) simple, with a type that can be a standard one (such as text, number, image) or *link*; (ii) complex: lists, which can be structured (that is, involve several attributes) and nested;
- *heterogeneous unions* can be used to specify that some attributes are optional and alternative to others and *forms* are modelled as virtual lists.

Following the terminology common in temporal databases, we can say that ADM could be used to describe *snapshot* Web sites, that is, conventional Web sites with no explicit management of time.

The goal of this paper is to express requirements and to comment on possible research directions concerning modelling issues. We will not consider query languages, which indeed constitute a major issue in temporal database management, since in our approach they can be seen as an "implementation" issue: our major goal is to understand which could be the right way to organize information to be offered to Web users; in turn, in a data-intensive site, this information should be extracted from a database, probably by means of time-aware expressions, which could be written in a temporal query language or in a standard query language, but automatically generated by means of a CASE tool [5].

Indeed, we have a situation similar to that reported by Snodgrass [9, p.389] with respect to temporal object-oriented databases: the challenge is to augment a logical model for Web sites in order to capture history of pages and their components, and to publish them in a suitable way. Indeed, since, as we argued above, we assume that Web sites are modelled by means a complex-object model with identity, it is clear that the most interesting ideas for temporal models in this framework should come from models with identity, so temporal extensions of the object model (as discussed by Snodgrass [9, p.389]) or of Entity-Relationship models (surveyed by Gregersen an Jensen [10]).

The rest of this paper is organized as follows. In Section 2 we comment on how the known dimensions for time (valid and transaction) are meaningful in this context. In Section 3 we expose the main requirements for the definition of temporal models for data-intensive Web sites, with respect to fixed schemes. Section 4 is devoted to some additional issues, which include those related to schema versioning and others that specifically arise in a Web framework. Finally, Section 5 is devoted to concluding remarks which sketch the idea of considering time just as one possible dimension, together with others that could be dealt with in a similar way.

2 Time Dimensions

As clearly discussed in all surveys on temporal databases (for example Snodgrass [9] or Jensen and Snodgrass [6]), there are various dimensions along which time can be considered. Beside *user-defined time* (the semantics of which is "known only to the user", and therefore is not explicitly handled), we have *valid* *time* ("the time a fact was true in reality") and *transaction time* ("the time the fact was stored in the database").

In a Web site, the motivation for valid time is similar to the one in temporal databases: we are often interested in describing not only snapshots of the world, but also histories about its facts. The slight difference here is that in temporal databases the interest is in storing histories and in being able to answer queries about both snapshots and histories, whereas in Web sites the challenge is on how histories are offered to site visitors, who browse and do not query. Therefore, this is a design issue, to be dealt with by referring to the requirements we have for the site. The natural (and not expensive) redundancy common in Web sites could even suggest to have a coexistence of snapshots and histories.

Transaction time in Web sites is related to the support of the *archival* nature of the Web, which could be stated, with a bit of exaggeration, as: "once a piece of information is published, it should not be retracted." While the above claim could in general be questioned, or at least interpreted in various ways, it is clear that in many cases it refers to an important user need: for example, if our university publishes wrong information about the exam calendar, which is later corrected, then students who were misled could strongly argue against the university. In more plain terms, we would often be interested in documenting what was the content of the Web site at a given point in time. As with valid time, this can be handled in various ways, depending on the actual requirements: history could be documented only for some important pieces of information, and, in contrast with what happens in relational databases, it can be managed by complex structures where new data is appended. This can help in documenting the changes on a Web site.

In the Web, a couple of additional issues emerge for transaction time which are not relevant in databases. First of all, as Dyreson [7] noted, there are no transactions on the Web, and so it is not obvious how to keep track of events. In general, this can be seen as a problem, a solution for which is the idea of *observant systems* [7], which can read data but do not really manage them; both Web servers and browsers are observant systems. However, in a data-intensive Web site, the problem can be reasonably handled, if the underlying database is suitably monitored—another motivation for a strong correspondence between a Web site and a database. A second issue, also noted by Dyreson [7], is that, while transaction-time in temporal databases is bounded by the current time, in a Web framework it could also refer to the future, to be used for planning publication; in some sense, it turns out that transaction time in the Web site need not be the same as transaction time in the underlying database, but we believe that we can still take advantage of the correspondence between the two.

3 Modelling Time

The basic ideas are here very similar to those mentioned by Snodgrass [9] about the representation of valid-time support in object oriented databases. The most natural solution to follow is the direct incorporation of time into the data model, with the introduction of specific constructs, which could include:

- the distinction between temporal and non-temporal page-schemes (with respect to the existence of their instances, not with the variability of their content); for example, it is likely that the home-page of a site is non-temporal, because it always exists (during the life of the site), whereas the instances of the course page-scheme in our department site have a lifespan corresponding to the time when the respective courses exist;
- the distinction between temporal and nontemporal attributes: during the lifespan of an object, some attributes have a fixed value, whereas others are allowed to change; given the nested nature of the model, this distinction should be allowed at any level of nesting, but with some limitations: each attribute should either be nontemporal or be temporal either because it is declared temporal or because (exactly) one of the higher level attributes that include it is temporal;
- the time granularity associated with each temporal page scheme or attribute: it could be convenient to have different granularities: for example, in a school calendar we could have the academic year as the granularity for courses and the week for the seminar schedule;
- the way we are interested in offering these pieces of information on the site:
  - by means of snapshots on specific instants; for example, in our calendar example, we could be interested in seeing the page of a course with this year's (or last year's) instructor and content;
  - by offering the history of valid values: in the same example, a list of the instructors for a given course throughout the years;
  - by means of a combined, redundant approach: an index with a list (essentially describing a history) and with snapshot pages;
  - by a more compact list showing only changes in the values.

  In each case, links have to be coordinated suitably, if needed; for example, if also instructor pages are temporal, then we could decide to have the instructor link in last year's version of a course page to last year's version of the instructor page (provided that the granularity is the same). Clearly, solutions are not obvious, and designer's choices are important (and the model should leave space for them).

4 More Time-Related Issues

Many other aspects that concern time could be relevant in modelling Web sites. Let us comment on some of them.

4.1 Version Management

In temporal object-oriented databases (Snodgrass [9]), version management is discussed with respect to transaction time. It is not obvious that the same should hold for Web sites. For example, what does it mean to have versions of the hypertext organization, or versions of the graphical layout? We could be interested in seeing information which was valid last year with today's organization, or viceversa. This has little to do with transaction time. In some sense (we will come back to this point later), versioning can give rise to different coordinates (at least partially independent), which correspond to the oft cited components of a Web site, data, hypertext, presentation [11, 4, 5]:

- changes in the presentation;
- changes in the hypertext structure (with possible consequences on the presentation);
- changes in the database structure (with consequences on hypertext structure and presentation).

This issue clearly deserves specific attention. It should be noted that, if the design process is supported by a tool that allows the specification of transformation primitives [5], then the possible transformations can be known *a priori*, in the same way as suggested by Kim and Chou [12] for version management of object-oriented databases.

With schema versions, the need for accurate management of links becomes even more delicate than in the cases discussed earlier.

4.2 Documenting the Degree of Currency of Information

We often see in Web pages the comment "last updated on ..." While these notes are useful, they only fulfill a small fraction of the user needs in this context. Indeed, especially when a page is generated out of a database, what does such a date mean: the last date the page was generated? or the last date a piece of information appearing in the page was changed? if so, what changed when? Also, in most cases, we would like to know what is the "currency reliability" of a page, which could be expressed as: "when was the information in the page last verified"?

These issues are once again delicate and dependent on user needs, they also interact with granularity (which pieces of information do we want to tag in this way, with the right compromise between legibility and detail?), and can benefit from the support of an underlying database. Once more, the crucial issue is to understand how to support the designer in the specification of these features.

4.3 Time Support Provided by a Content Management System

It is now widespread the idea that Web sites should be supported by *content management systems*, which allow users to update the information in the site, by means of specific services supported by the site itself (with suitable protection). In a data intensive site, this would mean that the updates to the database could be controlled in a tight manner, and not appear in an unrestricted way. In this framework, most of the issues we discussed above could be supported in an interesting way, including the decision on what should be temporal and what should not, and on which is the appropriate level in nested structures with which time should be attached. Also, the management of the degree of currency could be effectively supported.

5 Conclusions: a More General Perspective

One way to support time (be it valid time or transaction time) in Web sites is by providing support to the navigation of snapshot Web sites: the user specifies in some way the instant he/she is interested in, and then navigation can proceed with reference only to that instant. Indeed, we have implemented a prototype which offers this feature, as well as support for histories and versions (Del Nostro [13]), in a simple way: each page is dynamically generated with a parameter that specifies the instant of interest. We have used a conventional database as a backend, and so the implementation has required effort and efficiency has not been considered.

However, this approach suggests an interesting point: time can be see as a coordinate for Web sites, in the same way as others, which could include language (a multilingual site has pages that depend on one parameter, the language), device, personal choices, and so forth. Similar issues were considered in the past with the idea of *annotations* (Sciore [14]) for object-oriented databases and with the management of metadata properties for semistructured data (Dyreson et al. [15]); the notion of context in WebML is also related (Ceri et al. [11]).

In conclusion, we do believe that there is a lot of interesting issues to be dealt with in modelling time and other relevant “dimensions” in Web sites, and that they can be pursued in an interesting way if logical models are taken into account, both at Web site level and at an underlying database level: the management of transformations between the two levels could be a major benefit.

Acknowledgments I would like to thank Ernest Teniente and Paolo Merialdo, with whom I had some discussions on the topics of this paper, and Pierluigi Del Nostro, who explored some of these concepts and implemented a prototype within his master’s thesis. I am very indebted to Fabio Grandi and Carlo Combi who helped me in finding interesting references.

References

1. Atzeni, P., Mecca, G., Merialdo, P.: To weave the Web. In: VLDB’97, Proceedings of 23rd International Conference on Very Large Data Bases, August 25-29, 1997, Athens, Greece, Morgan Kauffman, Los Altos (1997) 206–215
2. Atzeni, P., Mecca, G., Merialdo, P.: Managing Web-based data: database models and transformations. IEEE Internet Computing **6** (2002) 33–37
3. Ceri, S., Fraternali, P., Bongio, A.: Web Modeling Language (WebML): a modeling language for designing Web sites. WWW9/Computer Networks **33** (2000) 137–157
4. Fernandez, M., Florescu, D., Levy, A., Suciu, D.: Declarative specification of Web sites with Strudel. VLDB Journal **9** (2000) 38–55

5. Merialdo, P., Mecca, G., Atzeni, P.: Design and development of data-intensive web sites: The Araneus approach. Technical report, Dipartimento di Informatica e Automazione, Università di Roma Tre (2000) Submitted for publication.
6. Jensen, C., Snodgrass, R.: Temporal data management. IEEE Transactions on Knowledge and Data Engineering **11** (1999) 36–44
7. Dyreson, C.: Observing transaction-time semantics with TTXPath. In: Proc. of 2nd Int'l Conference on Web Information Systems Engineering (WISE'01), Kyoto, IEEE Computer Society (2001) 193–202
8. Grandi, F., Mandreoli, F.: The Valid Web: An XML/XSL infrastructure for temporal management of Web documents. In: Proc of Int'l Conf. on Advances in Information Systems (ADVIS 2000), Izmir, Turkey, LNCS 1909, Springer-Verlag, Berlin (2000) 294–303
9. Snodgrass, R.: Temporal object-oriented databases: A critical comparison. In Kim, W., ed.: Modern Database Systems: The Object Model, Interoperability, and Beyond. ACM Press and Addison-Wesley (1995) 386–408
10. H., G., Jensen, C.: Temporal entity-relationship models—a survey. IEEE Transactions on Knowledge and Data Engineering **11** (1999) 464–497
11. Ceri, S., Fraternali, P., Bongio, A., Brambilla, M., Comai, S., Matera, M.: Designing Data-Intensive Web Applications. Morgan Kauffman, Los Altos (2002)
12. Kim, W., Chou, H.: Versions of schema for object-oriented databases. In: Fourteenth International Conference on Very Large Data Bases, Los Angeles (VLDB'88), Morgan Kauffman, Los Altos (1988) 148–159
13. Del Nostro, P.: La componente temporale nello sviluppo dei siti web (in Italian). Tesi di laurea in ingegneria informatica, Università Roma Tre (2001)
14. Sciore, E.: Using annotations to support multiple kinds of versioning in an object-oriented database system. ACM Transactions on Database Systems **16** (1991) 417–438
15. Dyreson, C., Böhlen, M., Jensen, C.: Capturing and querying multiple aspects of semistructured data. In: VLDB'99, Proceedings of 25th International Conference on Very Large Data Bases, September 7-10, 1999, Edinburgh, Scotland, UK, Morgan Kauffman, Los Altos (1999) 290–301