T-Araneus:
Management of temporal data-intensive Web sites

Paolo Atzeni, Pierlugi Del Nostro
Dipartimento di Informatica e Automazione
Università Roma Tre
Via della Vasca Navale, 79
00146 Roma
{atzeni,pdn}@dia.uniroma3.it

Abstract

T-Araneus is a tool for the generation of Web sites with special attention to temporal aspects. It builds on previous experiences in the management of data-intensive Web-sites, and is based on the use of high level models throughout the design process. The main feature is the adoption of a logical model for temporal Web sites, T-ADM, which allows the definition of page-schemes with temporal aspects (which could be related to the page as a whole or to individual components of it). The design process follows a development that starts with a traditional E-R scheme; the various steps lead to a temporal E-R scheme, to a navigation scheme and finally to a T-ADM scheme. The tool automatically generates both the relational database (with the temporal features needed) supporting the site and the actual Web pages, which can be dynamic (JSP) or static (plain HTML), or a combination thereof.

1. Introduction

The effective support to the management of time has received a lot of attention in the database community leading to the notion of *temporal database* (see Jensen and Snodgrass [6] for a recent survey and Snodgrass [9] for a textbook discussion). The need for an analogous support arises also in Web sites, as time is relevant from many points of view: from the history of data in pages to the date of last update of a page (or the date the data in a page was last validated), from the access to previous versions of a page to the navigation over a site at a specific past date (with links coherent with respect to this date). In general, various aspects of a Web site often change over time: (i) the actual content of data (the instructor for a course); (ii) the types of data offered (at some point we decide to publish not only the instructor, but also the TA for a course); (iii) the hypertext structure (we could have the instructor in a list for all courses and the TA only in a separate detail page, and then change, in order to have also the TA in the summary page); (iv) the presentation. In the same way as with databases, it could be interesting to refer to both *valid time* (the time when the information holds in the real world) and *transaction time* (referring to when the information is actually recorded and shown on the site). It is important to note that most current sites do handle very little time-related information, with past versions not available and histories difficult to reconstruct, even when there is past data.

We believe that the management of time in Web sites can be supported effectively by leveraging on the experiences made in the database field, and precisely by the combination of two areas: temporal databases on the one hand and model-based development of Web sites on the other. Let us briefly comment on the latter. The usefulness of high-level models for the intensional description of Web sites has been advocated by various authors, including Atzeni et al.[1,7], 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. With respect to *data-intensive* Web sites (that is, sites with large amount of data and significant hypertext structure) more structured models can definitely be useful: these sites 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 Merialdo et al. [7]).

The goal of this paper is to extend the experiences in the Araneus project [1,2,7] in order to propose a tool, called T-Araneus, which supports the development of temporal data-intensive Web sites. We now have a first version referring to temporal aspects of the content (and not structure) and to valid time, but this is interesting enough to validate this approach. Future extensions will consider additional degrees of change as well as the interaction with a Content Management System handling the updates.

This short paper is organized as follows. Section 2 is devoted to a brief review of the aspects of the Araneus approach that are need to understand this proposal, whose principles will then be illustrated in Section 3, with some details on the tool and the demo in Section 4.

2. The Araneus models and methodology

The Araneus approach (Merialdo et al. [7]) is focused on data-intensive web sites domain and it proposes a design process (with an associated tool) that leads to a completely automatic generation of the site extracting data from a database. The design process is composed of several steps each of which identifies a specific aspect in the design of a Web site. Models are used to represent the intensional features of the sites from various points of view: (i) the Entity Relationship (ER) model is used to describe the data of interest at the conceptual level (then, a translation to a logical model is handled by the tool in a transparent way), (ii) a "navigational" variant of the ER model (originally called NCM and then N-ER) is used to describe a conceptual scheme for the site (with major nodes, called *macroentities* and navigation paths, expressed as *directed relationships*), and (iii) a logical scheme for the site is defined using the Araneus Data Model (ADM), in terms of *page schemes,* which represent common features of pages of the same "type" with possibly nested attributes, whose values can come from usual domains (text, numbers, images) or be links to other pages.

The design methodology (Atzeni et al. [2]), supported by a tool called Homer, (Merialdo et al [8]) starts with conceptual data design, which results in the definition of an ER scheme, and then proceeds with the specification of the navigation features, macroentities and directed relationships. The third step is the description of the actual structure of pages (and links) in terms of our logical model, ADM. A fourth step is the specification of the presentation aspects, which are not relevant here. In the end, since all the descriptions are handled by the tool and the various steps from one model to the other can be seen as algebraic transformations, the tool is able to generate, in an automatic way, the actual code for pages, for example in JSP or in plain HTML, with access to a relational database built in a natural way from the ER scheme.

3. Management of temporal aspects

The tool we are proposing here handles temporal aspects in each of the models, by means of features that are coherent with the focus of the phase of the development process the model is used in.

The features we have developed so far refer to the temporal evolution of the content of a site, and do not refer to evolution of schemes.

Let us start by briefly illustrating the temporal features available in ADM, as this is the actual goal of the design process.

The temporal extension of ADM (hereinafter *T-ADM*) includes the possibility of distinguishing between temporal and non-temporal page schemes (in terms of the interest in describing the life-cycle of the page instances), and for each page, the distinction between temporal and non-temporal attributes; since the model is nested, this distinction is allowed at various levels in nesting, with some technical limitations. Temporal pages and attributes can have additional pieces of information attached, including the following:

- LAST MODIFIED: the date/time (at the granularity of interest) of the last change
- VALIDITY INTERVAL: the interval of time during which the element is/was valid
- VERSIONS: a link to a page containing all the versions of the element
- TIME POINT SELECTOR: in a temporal page scheme, a means to access the instance valid at a certain instant.

A special page scheme is also provided to be referred by VERSIONS attributes: it is called VERSION LIST, and it is used to contain links to the various versions.

The other models used in the design process require only higher-level temporal features, following proposals in the literature (see Gregersen and Jensen [5] for a survey). The temporal ER model allows the designer to specify which are the entities, the relationships, and the attributes to be considered as temporal, with little more details. Similarly, the temporal N-ER model allows the indication of the temporal macroentities and navigation paths.

4. The T-Araneus methodology and tool

The best way to see how our tool supports the designer is to refer to its original, non-temporal version: a “snapshot” site is a special case of a temporal site where all elements are non-temporal. Also, we can see a temporal site as the “extension” or evolution of a snapshot site. The development process follows the same path as we discussed in Section 2, with some extensions. The tool supports all phases, and therefore the demonstration we are proposing would follow the same development.

In the conceptual design phase, after the definition of the ER scheme, the temporal features are added, by indicating which are the entities, relationships and attributes for which the temporal evolution is of interest. This (beside being the input to the subsequent phases, to be discussed shortly) causes the generation of a relational database with the needed temporal features (we are using a standard relational database, as the temporal features are all generated by the tool and so require no specific competence to the designer).

The hypertext conceptual design phase requires the specification of the temporal features over hypertext nodes (macroentities) and paths (directed relationships). In the N-ER model a macroentity is considered to be temporal if it includes temporal elements from the T-ER model. The main choices in this phase are related to the facets of interest from the temporal point of view, including the choice among alternatives, such as (i) the last version with a timestamp; (ii) versions at a given granularity; (iii) all versions.

The logical design of the site, with the goal of producing the actual T-ADM scheme, is based on the choices made in the previous phases. Specifically, the decisions here mainly concern page organization: the main, new choice concerns the organization of the temporal attributes for which versioning has been chosen in the previous phase There are various alternatives:

1. All versions in the same page, each of which with the validity interval.

2. The current value, possibly with a LAST MODIFIED attribute, in the page scheme, with a VERSIONS attribute that points to a VERSION LIST page scheme which will present a list of all versions of the temporal element. A VALIDITY INTERVAL attribute can be attached to each temporal value.

3. A time point selection: the page will show all attribute values where the validity interval include the time point. Constant attributes will always be presented. The validity interval can be shown using the VALIDITY INTERVAL attribute. If a directed relationship is marked as temporal in the N-ER model, navigation has also to be coherent with the time instant of interest.

At the end of the design process, the tool can be used to generate the actual site, which can be static (that is, plain HTML) or dynamic (JSP); actually some of the features (such as the time point selector) are allowed only in the dynamic environment.

References

1. P. Atzeni, G. Mecca, P. Merialdo, "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. P. Atzeni, P. Merialdo, G. Mecca: Data-Intensive Web Sites: Design and Maintenance. *World Wide Web* 4(1-2): 21-47 (2001)
3. S. Ceri, P. Fraternali, A. Bongio, M. Brambilla, S. Comai, M. Matera, "*Designing Data-Intensive Web Application*" Morgan Kauffman, Los Altos (2002).
4. M. Fernández, D. Florescu, A. Levy, D. Suciu "Declarative Specification of Web Sites with STRUDEL" *VLDB Journal* 9(1): 38-55 (2000)
5. H. Gregersen, C. S. Jensen, "Temporal Entity-Relationship Models a Survey", *IEEE Transactions on Knowledge and Data Engineering* 11 (1999) 464–497
6. C.S. Jensen, R. Snodgrass, "Temporal data management", *IEEE Transactions on Knowledge and Data Engineering* 11 (1999) 36–44.
7. P. Merialdo, P. Atzeni, G. Mecca: "Design and development of data-intensive web sites: The araneus approach". *ACM Trans on Internet Technology* 3(1): 49-92 (2003)
8. P. Merialdo, P. Atzeni, M. Magnante, G. Mecca, M. Pecorone: Homer: a Model-Based CASE Tool for Data-Intensive Web Sites. *ACM SIGMOD Conference* 2000: 586
9. R. T. Snodgrass, *"Developing Time-Oriented Database Application in SQL"*, Morgan Kaufmann, 2000.