Association for Information Systems

AIS Electronic Library (AISeL)

AMCIS 1998 Proceedings

Americas Conference on Information Systems (AMCIS)

December 1998

Conceptual Modeling for Web Site Development

Reinhard Jung
University of St. Gallen

Robert Winter
University of St. Gallen

Conceptual Modeling for Web Site Development

Reinhard Jung
Robert Winter
Institute of Information Management
University of St. Gallen

Abstract

Today's web site development practice lacks suitable software engineering principles. In this paper we argue to adopt state-of-the-art software engineering principles and tools in order to improve web site development.

Introduction

Early software development in the 60ies and 70ies can be described as follows: large scale IS have been implemented by editing programs with available programming languages. Typically, the result was a spaghetti coded mixture of data and functional aspects and therefore an information system that was difficult to maintain (Martin, 1982). Thus, we refer to this time as the "software crisis". A rather similar approach can be observed when looking at today's web site development practice (Schranz, 1998). By coding web sites with HTML authoring tools we create complex multiple-file systems that mix up various very different aspects, e.g. appearance and contents. By adopting state-of-the-art software engineering principles web site development might become a much easier and cost-efficient task. Two general approaches can be differentiated: On the one hand, state-of-the-art CASE environments can be extended to cover not only traditional systems, but also web sites. On the other hand, existing development tools for web sites can be extended to support early development stages and large development projects.

Conceptual Modeling for Web Site Development

Hypertext have been known long before the success of the world wide web (WWW). However, hypertext development took on a different complexion because of two web-specific characteristics: the integration of multimedia elements and the high update frequency. The latter is necessary to attract the surfer's attention continually. Therefore, like conventional systems development, HTML hypertext (in the following we call it hypertext for short) development requires conceptual modeling in order to keep maintenance easy and cost-efficient (Jung and Winter, 1998).

The language HTML was "invented" in 1986 as a derivative of SGML (Bradley, 1997); in fact SGML was used as a meta language to define HTML. As HTML 1 mixed up document appearance and contents it was a simplification of SGML which separated these aspects by so-called document type definitions (DTDs). It took twelve years to revive this well-proven concept in HTML: the recommendation for the next language standard (version 4.0) proposes style sheets (Pozadzides and Quinn, 1997) which enable the developer to define general layout properties at a single point (within a style sheet). Thus, consistent modifications are much easier to accomplish.

In the following we present our proposal of a modeling approach and a corresponding toolset for hypertext development and maintenance (cf. Figure 1). The core components of the toolset are: a Site Diagrammer, a Layout Template Designer, a Contents Editor, a Constistency Checker, and a HTML Generator.

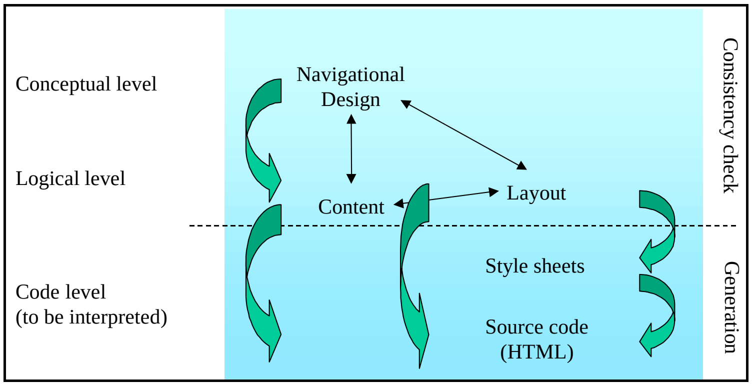


Figure 1. Levels of Abstraction for Web Sites

- *Site Diagrammer:* It should be possible to design a web site graphically by drawing, for example, rectangles which represent pages. The properties of a page comprise its physical filename, references to one or more layout templates to be used for the page, and references to the content for each layout template element. The navigational structure of the site could be depicted by arrows between the rec-tangles. Each arrow represents a hyperlink pointing from one page to another page or to an anchor within another page.
- *Layout Template Designer*: The elements of the various layout templates (e.g. buttons, headings, simple text paragraphs) could be created and arranged within the template by state-of-the-art (graphical) aids. A layout template might be used for different pages.
- *Contents Editor*: A special editor could be used in order to specify the content of a layout template element within a certain page.
- *Consistency Checker*: A seperate or integrated check of the model's constistency is necessary to avoid, for example, layout template elements for certain pages without associated contents. Another possible inconsistency is a navigational element (an arrow between two rectangles) without a corresponding layout template element within the "pointing" page.

- *HTML Generator*: The Generator is finally used to produce the HTML files.

By using those tools or something similar, development and maintenance of hypertexts could benefit in some ways:

- Inconsistencies within the site (broken links, missing pictures etc.) can be avoided by generating the HTML code.
- Modification of the site's appearance could be done at the layout template level without touching the content.
- Modification of the contents of a single page could be done without touching the layout.
- Extensions/deletions within the site could be done by simply adding/removing rectangles and arrows. Of course, in case of extensions layout and content have to be specified.

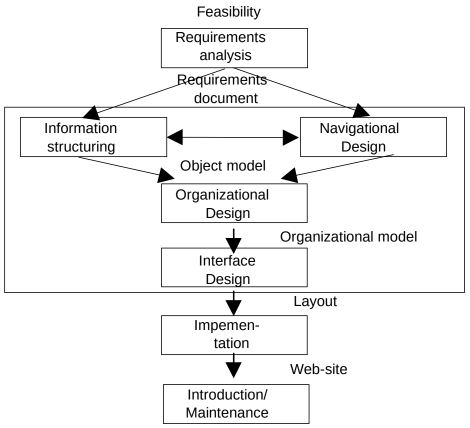
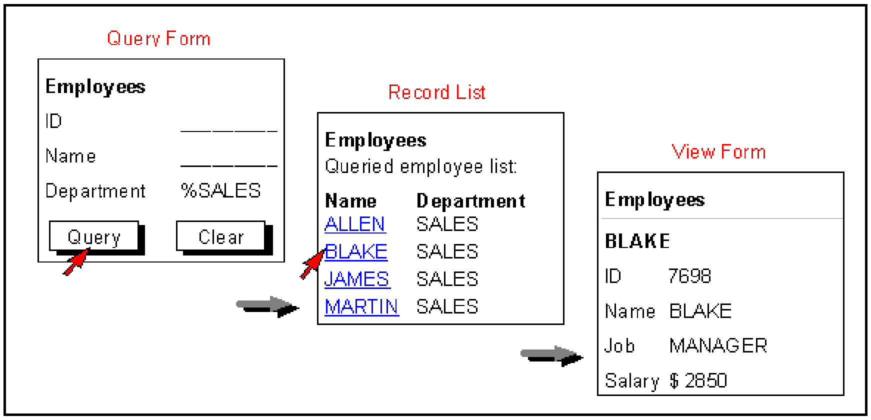


Figure 2. W3DT Development Process
(Bichler and Nusser, 1996)

Most of the components mentioned above are available within some advanced web design tools, e.g. Microsoft FrontPage 98 and NetObjects Fusion 3.0. However, there is no tool that is comparable to CASE toolsets as regards the coverage of early development stages like requirements analysis and information structuring. Only the development process of W3DT considers those analysis tasks (cf. Figure 2). Another shortcoming of today's web design tools is their handling of transactional databases. Typically, the tools make use of interfaces on the level of database tables (e.g. ODBC and MS-IDC) in order to access the data, i.e. an implementation level dependency between web site and database is introduced. Therefore, database modifications are likely to require corresponding manual modifications within the web site.

An alternative way of web site development is the extension of available CASE toolsets as far as web design is concerned. Toolsets such as, for example, Designer/2000 are already capable of generating simple web servers that access transactional data (cf. Figure 3). The result is a set of web pages that allow for navigation through retrieved data. Relationships between data items can be represented by hyperlinks that lead from data items in one page to related data items on another page. However, the major shortcoming of CASE toolsets in this context are their restricted web design capabilities. A possiblity to overcome the shortcomings of both web design and CASE toolsets as regards the efficient development of web sites might be the integration of the respective core features.



Employees

ID ______

Name ______

Department %SALES

Employees

Queried employee list:

Name	**Department**
ALLEN	SALES
BLAKE	SALES
JAMES	SALES
MARTIN	SALES

Employees

BLAKE

ID 7698

Name BLAKE

Job MANAGER

Salary $ 2850

Figure 3. Sample Results of Designer/2000's Web Server Generator

Related and Future Work

The most promising and cost-efficient strategy to improve web site modeling and development seems to be the integration of specific concepts for hypermedia systems into state-of-the-art CASE technology such as, for example, Designer/2000 (Winter, 1996). The basic idea about CASE is a common repository that comprises information about the stages and views of systems development. In order to cover web site development, certain repository extensions have to be made. It is especially neccessary to define an information model for web sites (Isakowitz et al., 1995; W3DT-Team, 1997) which comprises all relevant aspects (navigational design, contents, and appearance).

References

Bichler M. and Nusser S. "SHDT-The Structured Way of Developing WWW-Sites" *Proceedings of the 4th European Conference on Information Systems*, Dias Coelho J. et al. (eds.), Lisbon, 1996, pp. 1093-1101.

Isakowitz T., Stohr E.A., and Balasubramanian P. "RMM: A Methodology for Structured Hypertext Design"; *Communications of the ACM* (38:8), 1995, pp. 34-44.

Jung R. and Winter R. "CASE for Web Sites: Towards an Integration of Traditional CASE Concepts and Novel Development Tools" *Proceedings of the Symposium on Applied Computing 1998*, J. Carroll et al. (eds.), New York, 1998, pp. 726-731.

Martin J. *Application development without programmers*, Prentice-Hall, Englewood Cliffs, 1982.

Pozadzides J. and Quinn L. "Guide to Cascading Style Sheets", <http://www.htmlhelp.com/reference/css/> (26 Nov 1997).

Schranz M. "Engineering Flexible World Wide Web Services" *Proceedings of the Symposium on Applied Computing 1998*, J. Carroll et al. (eds.), New York, 1998, pp. 712-718.

Winter R. "Integrated CASE Environments for Information Systems Development - State of the Art and Future Challenges" *Proceedings of the 2nd Americas Conference on Information Systems*, J.M. Carey (ed.), Phoenix, 1996, pp. 698-701.

W3DT-Team "W3DT – WWW Design Technique", <http://wwwi.wu-wien.ac.at/ w3dt/> (26 Nov 1997).