Chapter 5

Conceptual Modeling of Large Web Sites

Bernhard Strauch and Robert Winter
University of St. Gallen, Switzerland

Current web site development is still dominated by technical issues. In order to enable efficient communication between developers and to provide a stable foundation for adopting new technologies, web sites should be derived from conceptual models. Based on the state-of-the-art of conceptual modeling as implemented in current CASE environments as well as web site development tools, the “essence” of a web site is identified, and an adequate conceptual meta model is proposed. Appropriate web site models are intended to capture not only hierarchical document structure and hypertext semantics, but also dynamical page generation from databases as well as explicit and implicit navigation. It becomes evident that web sites can be regarded as supersets of traditional information systems, thereby requiring conceptual modeling to include various additional features. The proposed meta model comprises several classes of information objects, various types of associations, design rules, and quality checks. For illustration purposes, the model is applied to an existing web site. Commercial web site development tools are analyzed with regard to the extent to which they support conceptual web site modeling.

INTRODUCTION

Early systems development was dominated by using authoring tools (“editors”) to manually edit procedural program code. As a consequence, hand-writ-
ten code mixing up data usage and functional aspects was difficult to maintain (Martin, 1982). Besides of expensive quality control and communication problems among developers, the resulting code suffered from various implementation dependencies, thereby forcing developers to re-do large portions of the development process when technical details (e.g., file structures, access paths) change. A rather similar approach can be observed when looking at today's web site development practice (Rosenfeld & Morville, 1998). By creating web sites using HTML authoring tools, complex code is created that does not only mix up appearance and contents, but also depends widely on implementation details. Moreover, the utilization of different authoring tools complicates communication between developers. As an example, the following problems usually occur with regard to navigation when different web sites have to be integrated:

- Navigation is interpreted and implemented in different ways depending on the authoring tool in use. Different tools use identical terms for different concepts or different terms for identical concepts.
- Navigation is not based on user requirements for optimal access to information objects or associations between information objects (see e.g., Morville & Rosenfeld, 1998; Richmond, 1999). Instead, implementation details like various frame variants dominate design.
- As a consequence, similar navigational concepts are implemented (and specified) in different ways so that explicit integration efforts are necessary to identify common structures and implement them consistently.

In order to enable efficient communication between developers and to provide a stable foundation for adopting new technologies, conceptual modeling of web sites is essential. We understand web sites as server components of distributed applications which use the HTTP protocol to exchange data between servers and clients ("browsers"). By this definition, the principal problem of web site development becomes apparent: Even the relevant class of application components is defined by technical attributes (HTTP protocol, server functionality) instead of conceptual differences. Conceptual modeling should be independent of all technical and application dependent details. Before analyzing conceptual issues in web site design, proposing an appropriate conceptual model, and checking support potentials of current web site development tools, therefore, we should discuss whether our initial definition is appropriate, i.e., to what extent web sites conceptually differ from traditional information systems.

**Web Sites vs. Traditional Information Systems**

Traditional business applications support business processes by implementing data entry and data manipulation of business transactions. Web applications go beyond this functionality by integrating different media for information represen-
The Anatomy of the ArchiMate Language
www.igi-global.com/article/anatomy-archimate-language/40951?camid=4v1a