



Chapter IV

Integrating Hypermedia Functionality into Database Applications

Anirban Bhaumik, Deepti Dixit, Roberto Galnares, Aparna Krishna,
Michael Bieber, Vincent Oria, Firas Alljalad and Li Zhang
New Jersey Institute of Technology, USA

Michalis Vaitis and Manolis Tzagarakis
University of Patras, Greece

Qiang Lu
New Jersey Institute of Technology, USA and
Suzhou University, Peoples Republic of China

ABSTRACT

The general goal of our research is to automatically generate links and other hypermedia-related services to analytical applications, such as geographic information systems and decision support systems. Using a dynamic hypermedia engine (DHE), we propose to automate the following features for database systems, both on and off the Web. First, we automatically generate links based on the database's relational (conceptual) schema and its original (non-normalized) entity-relationship specification. Second, the application developer can specify which kinds of database elements are related to diverse elements in the same or different database application, or even another software system. Our current DHE prototype illustrates these for a relational database management system. We propose integrating data warehousing applications into the DHE. We

also propose incorporating data mining as a new kind of automated link generation. Passing the application element selected by a user, a data mining system would discover interesting relationships for that element. DHE would then map each relationship to a link. DHE's linking is based on the structure of the application, not keyword search or lexical analysis based on the display values within its screens and documents. DHE aims to provide hypermedia functionality without altering applications by building "application wrappers" as an intermediary between the applications and the engine.

INTRODUCTION AND MOTIVATION

Database queries typically return results in a plain text format. Some applications on the World Wide Web generate link anchors for database elements, but these anchors normally hold a single link to the most obvious destination for the dominant type of user.

We could consider each element within a database application as a potential starting point for information exploration. Each element could have multiple links, each representing a different relationship (schema-based or otherwise). The ability to explore a piece of information in more detail could help users resolve doubts about or simply better understand that item, as well as the analysis or display of which it is a part. Users may wish to dig deeper around data values and symbols, labels on graphs or user input forms, options in pop-up lists or even on the menu commands they can invoke.

To complicate the developer's job, users often have different mental models of an application and its underlying domain than the developer. Even when developers work closely with users, the end result might not be intuitive for all users or serve each user's individual tasks equally well. Many people visit a given application's screen aside from the most dominant type of user(s) for which it was developed. These include other users of the application, customer service representatives, company analysts, managers, trainees, people inside the company designing new databases or applications based on the current one, external analysts and stockholders, among others. Each may be interested in different aspects of application elements, according to their current task-at-hand. Customization is one solution, but even so, users might often wish to explore several different relationships from a given anchor, and therefore should have several links available.

The purpose of this chapter is to explore all aspects of hypermedia support for database applications. We base much of our discussion on our experience designing the Dynamic Hypermedia Engine (DHE). DHE automatically generates anchors, sets of links and metadata within database applications, as well as supporting users with other types of hypermedia structuring, navigation and annotation functionality, including guided tours and annotation. As we explain later, the links provide direct access to a broad range of relationships among application elements, including those

29 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/integrating-hypermedia-functionality-into-database/8272

Related Content

Benchmarking and Data Generation in Moving Objects Databases

Theodoros Tzouramanis (2005). *Encyclopedia of Database Technologies and Applications* (pp. 23-28).

www.irma-international.org/chapter/benchmarking-data-generation-moving-objects/11117

On Efficient Evaluation of XML Queries

Sherif Sakr (2011). *Theoretical and Practical Advances in Information Systems Development: Emerging Trends and Approaches* (pp. 239-293).

www.irma-international.org/chapter/efficient-evaluation-xml-queries/52959

Assumptions Underlying Agile Software-Development Processes

Daniel Turk, France. Robertand Bernhard Rumpe (2005). *Journal of Database Management* (pp. 62-87).

www.irma-international.org/article/assumptions-underlying-agile-software-development/3342

Reverse Engineering from an XML Document into an Extended DTD Graph

Herbert Shiuand Joseph Fong (2009). *Journal of Database Management* (pp. 38-57).

www.irma-international.org/article/reverse-engineering-xml-document-into/3403

Vertical Database Design for Scalable Data Mining

William Perrizo, Qiang Ding, Masum Serazi, Taufik Abidinand Baoying Wang (2005). *Encyclopedia of Database Technologies and Applications* (pp. 736-739).

www.irma-international.org/chapter/vertical-database-design-scalable-data/11232