Generating Hypermedia Applications for Volatile Database Output

Geert-Jan Houben and Paul De Bra
Eindhoven University of Technology, Department of Computing Science
PO Box 513, NL 5600 MB Eindhoven, the Netherlands
E-mail: {houben,debra}@win.tue.nl

Abstract: This paper reports on work in progress on the generation of hypermedia (Web) presentations for the results of arbitrary queries to multimedia databases. In [Houben & De Bra 1997] we have proposed a heuristic algorithm for generating navigation structures for multimedia database output, based on ideas from RMM [see Isakowitz et al. 1995]. We have shown that only part of the RMM design methodology can be automated deterministically, and have provided both heuristics and a way to override them through additional specifications in a database query. This paper describes our approach to automatically generate a presentation for multimedia database output, in addition to the generation of navigation for such output described in [Houben & De Bra 1997].

1. Introduction

In [Houben & De Bra 1997] we have shown an approach to generating World Wide Web applications for volatile database output. The motivation for this approach is that the use of a hypermedia platform such as World Wide Web can help to represent the less structured and not purely textual information one typically finds in applications such as employee databases, museum databases, geographic information systems, and mail-order catalogs and services.

Since the design and construction of a hypermedia application involves the representation of relationships between information objects, the approach for generating the *navigation structure* is based on the ideas of RMM [see Isakowitz et al. 1995]. RMM combines elements from the Entity-Relationship model [see Elmasri & Navate 1990] and HDM [see Garzotto et al. 1991] to effectively manage relationships between objects. The core of our approach is to use as much as possible the relevant aspects that are embedded in the data definition (of the volatile information) and in the query specification. In [Houben & De Bra 1997] we have shown how we can deduce the navigation structure for the hypermedia application from the data definition and query specification.

The subject of this paper is the generation of the *presentation* (through WWW browsers) of volatile database output. This includes the presentation of the records (from the query result) and the presentation of the relationships with other records. The information contained in a query result generally depends on the data structures underlying the volatile information, and on the given query specification. We have shown that RMM and its supporting tool RMCase [see Díaz et al. 1995] do not help in the case of results of arbitrary queries. The main problem is that the dynamically generated structure of a query result cannot be (trivially) translated into a hypermedia presentation.

2. Record Presentation

Based on the concepts of RMM, the lowest level of representation of information is the level of the *slices*. Just like a pizza may be too large to eat all at once, an object may be too large to present all at once. It is therefore divided into slices, which are essentially groups of attributes that belong together (semantically). This grouping is used to divide a record into information chunks that can be elegantly displayed by a World Wide Web browser, without too much information becoming "invisible" due to a lack of space on the screen.

It is obvious that a designer of a hypermedia application would specify the presentation of all slices explicitly contained in the data structures. In order to do so, our approach asks the designer to do two things:

1. *Attribute presentations* are chosen for every attribute contained in the slice.
2. A *slice presentation* is constructed using the chosen attribute presentations.

We assume that the attribute presentations are specified in a way that enables the layout algorithm to measure or estimate what (size) rectangular screen area is needed for an attribute. The presentation of a slice can be encoded in HTML using tables. However, such an encoding does not specify what the relationship is between attributes, and thus what the possible reasons are for positioning attributes the way they are. Therefore, we propose to generate the HTML code from an internal representation which specifies *relative* positions of attributes. For instance, we can specify that a "subtitle" attribute must be positioned under a "photo" attribute (and that it must be centered under the photo), and that a "description" attribute must be positioned to the right of the "photo". A translator script will combine size information and these positioning constraints into a layout that is then converted to HTML code.

Both size information and positioning constraints may be hard or soft. An image may have a fixed size and the subtitle really must be positioned immediately under the image. But a text paragraph may be presented in a wider or narrower column and if there is no room we may allow the description to be positioned under the image and subtitle. See [Borning et al. 1997] for the use of constraints for automatic user-interface layout generation.

Another issue involved in the attribute presentation is the choice and presentation of index attributes that are used to provide navigation to slices. We propose to provide two presentations for index attributes:

- a *compact* presentation used to help the user in identifying the records he is searching for;
- a *complete* presentation used to supply the user with all the information available on the selected record.

These two types of presentations can be used to handle different sizes of query results. Our typical target user wants to obtain an efficiently "browsable" set of records. Using the compact attribute presentations first, it becomes efficiently possible to identify the records that are (potentially) interesting. Asking for the complete presentation for those records makes a subsequent thorough inspection of the information possible.

3. Relationship Presentation

In [Houben & De Bra 1997] we have defined the generation of access structures for a query result. This includes the definition of record relating structures (index, guided tour, indexed guided tour or grouping) and the relationships between slices. It did not include any specification of how these relationships are represented.

In some situations the user asks for information that is represented by slices involved in multiple relationships. For example, the representation of a "multiple join" means in general that a slice can give access to other slices contained in multiple guided tours. For the presentation to the user it is essential that the presentation of the navigation and the presentation of the actual record contents are generated simultaneously because both parts of the presentation share (fight for) the same screen real-estate.

We have therefore augmented our approach with practical guidelines to specify how navigation and actual contents are displayed together. The basic idea is that the designer specifies:

1. How the browser page is divided in different frames for navigation and contents;
2. Which navigation information is displayed (not all navigation information is accessible at all time);
3. How this navigation information is displayed.

The presentation of new slices must always be *generated*, when we deal with new slices (parts of existing slices or combinations of existing slices). Sometimes, the presentation of access structures can be *designed*, whenever we can use presentation aspects defined for existing slices. For instance, if a query results in a single indexed guided tour through a set of new slices, the page layout for that result can probably be the same as that for single indexed guided tours through a set of existing slices. But for complicated multiple joins generation of the presentation of the navigation structure may still be needed.

4. Conclusion

In this paper we have discussed two important aspects extending our approach. The first aspect is the attribute presentation and the composition of attribute presentations into record (or slice) presentations. We have shown what elements need to be added to the approach to facilitate the proper design of applications for ad hoc query results. The second aspect is the presentation of relationships that represent navigation mechanisms. We have shown how the designer can fix a number of details in order to produce an effective application. The presentation of navigation structures is more an issue of design than of automatic generation.

5. References

[Borning et al. 1997] Borning, A., Marriott, K., Stuckey, P., Xiao, Y. (1997). Solving linear arithmetic

constraints for user interface applications. *Proc. ACM Symposium on User Interface Software and Technology.*
[Díaz et al. 1995] Díaz, A., Isakowitz, T., Maiorana, V., Gilabert, G. (1995). RMC: A Tool to Design WWW Applications. *Proc. Fourth International World Wide Web Conference,* 559-566.
[Elmasri & Navate 1990] Elmasri, R. & Navate, S. (1990). *Fundamentals of Database Systems.* The Benjamin/Cummings Publishing Company, second edition.
[Garzotto et al. 1991] Garzotto, F., Paolini, P., Schwabe, D. (1991). HDM - A Model for the Design of Hypertext applications. *Proc. ACM Hypertext '91 Conference*, ACM Press, 313-328.
[Houben & De Bra 1997] Houben, G.J. & De Bra, P. (1997). World Wide Web Presentations for Volatile Hypermedia Database Output. *Proc. WebNet97, the World Conference of the WWW, Internet,and Intranet*, AACE, 229-234.
[Isakowitz et al. 1995] Isakowitz, T., Stohr, E., Balasubramanian, P. (1995). RMM: A Methodology for Structured Hypermedia Design. *Communications of the ACM*, 38 (8), 34-44.