HERA: Automatically Generating Hypermedia Front-Ends for Ad Hoc Data from Heterogeneous and Legacy Information Systems

Geert-Jan Houben [1,2]

[1] Eindhoven University of Technology, Dept. of Mathematics and Computing Science
PO Box 513, 5600 MB Eindhoven, the Netherlands
g.j.houben@tue.nl
[2] University of Antwerp (UIA), Dept. of Mathematics and Computer Science
Universiteitsplein 1, 2610 Antwerpen, Belgium

Abstract. The generation of hypermedia (or Web-based) presentations plays an important role in information management on the World Wide Web. In applications for Web modeling and querying, information extraction and integration, or Web site construction and restructuring the generation of hypermedia aspects for given data is essential. Specifically, the design of dynamic Web pages is an issue that needs more research. The HERA research project aims at developing software that generates hypermedia presentations for (semi-structured) data retrieved from heterogeneous, legacy information sources. While traditionally the target applications use relational databases, their Web-based successors integrate information from both database and non-database information sources. In this paper we propose key functional aspects of the HERA software architecture. We thus identify relevant issues concerning the integration involved in automatically generating hypermedia front-ends for data from heterogeneous information systems.

1 Introduction

In applications for Web modeling and querying, information extraction and integration, or Web site construction and restructuring the generation of hypermedia aspects for given data is essential. The use of modern *hypermedia* or *Web-based* platforms can help information systems in presenting their information to end-users in a more elegant and more effective way than before using the metaphor of the World Wide Web. The use of hyperlinks enables the addition of different kinds of relationships to the information, thus offering more semantics to the users.

Take for example an application that a *real estate agent* uses to show his clients the properties on sale that satisfy the criteria of the client. Traditionally, such an application is based on one or more databases and thus the information is presented in tables of records. While these tables contain all the information that is asked for, the presentation can be improved significantly when using navigation principles standard in the Web-based context. By adding additional relationships, implemented via hyperlinks, the user obtains a *web* of information that allows multiple ways of

navigating through the information. Specifically, in applications where the legacy records are too large or complex to show in a simple table row, the use of a *Web-like navigation platform* can significantly increase the ease with which a user can navigate through the information. In the real estate example the agent usually asks the system to show the client a collection of properties on sale: the client navigates through that collection looking for interesting properties; usually, the client navigates from property to property based on different parts of the descriptions, fully exploiting the internal structure of the data.

A typical phenomenon here is that the agent cannot exactly foresee what data to present. In general, this leaves the system with the task to *automatically generate* (derive) a hypermedia presentation for ad hoc data. This generation process should not only translate the data into the hypermedia format. The main purpose is to obtain an *added value* for the user who is inspecting the query result. This implies that several kinds of metadata are exploited during the generation process to get to a presentation of the data that offers this added value.

In the *HERA* research project we are designing and developing software that supports this generation process (see [5,6]). In this research we focus on the *functional* aspects of the process of (semi-)automatically generating hypermedia front-ends for a heterogeneous information system with data from multiple (database) applications: important questions concern the *adaptation* to queries and data, the kinds of *metadata* involved and their *specification*. In this specific paper we propose aspects of the *design* for the associated software tool, also called HERA, as relevant for the identification of key research issues. These *open* issues include the *integration* aspects involved in dealing with data from a heterogeneous information system: while HERA's main investigations are independent of the integration aspects, the ultimate goal of the project requires the integration aspects to be covered also.

2 Target Application Data

HERA's *target applications* contain data with an internal structure that the system can exploit to provide a suitable hypermedia presentation. Concrete examples are applications for real estate sales, mail order catalogs, auctions or (used) car sales, while we use the application of the *Electronic Program Guide* (EPG) as the carrier for our research with Philips and CWI (in the joint Dynamo research project). An EPG provides descriptions of programs broadcast on different TV channels.

Usually, the data that play a part in these applications do not contain just small textual values, but they contain typical hypermedia or multimedia values. Moreover, the structure is not always as rigidly defined as in traditional strongly structured models. Think of data on a TV program, with the time of broadcasting, the subject, information on the people involved, references to related programs, background stories and perhaps a short preview. In this context HERA assumes the data to be *semi-structured* in the sense that the data include (part of) their own definition.

This assumption also benefits the next characteristic aspect. Since the information in the target applications usually is retrieved from different, possibly heterogeneous sources, the architecture must allow for an *integration* of the different data sources, such that the system virtually operates on a single collection of data. Consider our

EPG application where both TV data and EPG data could come from the entire *Internet*. That application aims at constructing an EPG based on data available across the Internet from different information *providers*.

3 Presentation Generation

3.1 Generating Maximum Structure

The nature of the target data is such that a philosophy of generating a "*maximum structure*" benefits the navigation process. In legacy (relational) database applications data are presented usually in a format close to the storage format, e.g. a list of records. Then, the user is confined to the *limited* navigation space implicitly defined by the list concept. The data in our target applications contain typical *hypermedia* elements which makes their presentation a rather complex issue: to help the user navigate through the information appropriate navigation structures should be made accessible. By offering a large set of relationships the user obtains a multitude of ways to explore the data. This represents the required *added value* mentioned in the introduction.

3.2 Adaptation Aspects

The task of the *generation process* is to produce a hypermedia presentation for data that are generally the result of a *query* that is asked. Whether the user asks this query explicitly or implicitly is not relevant here. A hypermedia presentation needs to be constructed for the query result, and this process is influenced by a number of aspects:

- *General presentation heuristics*: General heuristics embedded in the application represent the knowledge of the application designer on the generation of relevant hypermedia presentations. These heuristics contain a number of rules on the traversal of large collections of objects. For example, in certain situations access to a set of objects through an index or through a guided tour seems favorable, while in other cases several objects can be displayed on one page.
- *Personal preferences*: The end-user can express personal preferences as to how the generation is performed. If, for example, the real estate client knows that the system generally produces collections of properties accessed through an index, but prefers guided tours, then the system could acknowledge this preference.
- *Platform information*: The generation process can acknowledge the platform being used. If the real estate client views the information on a novel browser based information appliance with a small screen, e.g. a PDA, it would make sense to leave out some of the more graphic elements of the data and to concentrate on the textual elements (leaving the graphics for a subsequent session on a machine with a suitable screen).
- *Application intelligence*: The application designer or owner can add knowledge that does not represent general presentation heuristics, but expresses the specific

goals that the application wants to reach. In the context of our real estate example the application could add real estate properties to the query result for commercial reasons (of the real estate agent), e.g. "well, the client asked for houses under 200K, but let's include some houses slightly over that limit".

Above we described different kinds of *adaptation* [3] to the data and its presentation. This adaptation intelligence is represented by a set of transformations that specify how query and data are transformed into a hypermedia presentation for the browser. Here the exact heuristics or platform dependencies are less relevant than the fact that transformations are specified that need to be executed by the system.

3.3 Navigation Generation

We exploit the structure in the data not just as a way to convey a meaning, but also as a way to navigate through the data when presented in a hypermedia format. Note that the "additional" relationships serve two purposes. First, they help to add more *semantics* to the data: that is a contents issue. Secondly, they can be of a syntactic nature aimed at producing a more elegant *layout* and *visual presentation* of the data.

One of the starting points for this research was *RMM* (see [4,7,8]). The use of *slices* (or m-slices) as a way to decompose data records into smaller units designed to be presented in a hypermedia application matched our approach. Moreover, it inspired us to consider *inter-* and *intra-record* structure as the basis for the user's navigation.

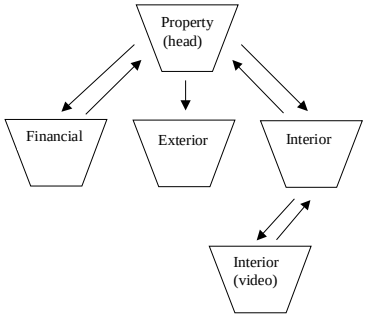

Fig. 1. Example slice diagram for the Property concept

The above figure shows the *intra-record structure* for the property concept. It specifies how the data are divided into separate units each designed to be elegantly presented in a browser's window. There are separate *slices* for the general data on the property (its "head" slice), for the financial data, for descriptions of the interior and exterior and for a video description of the interior. Moreover, it shows how *slice links* (intra-record relationships) relate the different slices within the records on a property.

The data contain *inter-record relationships* to provide the user with the possibility of going through *collections* (sets) of properties. Inspired by RMM[1], the standard

[1] Note that in Extended RMM the slice concept, there called m-slice, is slightly more advanced, but for reasons of presentation we have chosen to use the original slice concept here.

mechanisms for inter-record relationships will be the simple *hyperlink*, the *index*, the *guided tour*, the *indexed guided tour*, and the *grouping*.

Our current *heuristics* are based on experimental experiences (more elaborate experiments are being set up) on the traversal of the generated hypermedia objects.

3.4 Ad Hoc Queries

In our target applications we assume that for the results of the typical *standard* queries elegant hypermedia presentations have been designed. For *ad hoc* queries it is not possible to foresee exactly how the presentation should look like. Therefore, we want a system that *automatically* (or perhaps semi-automatically, with a little help from an intermediate user) derives a hypermedia presentation for an ad hoc query. One of the basic heuristics embedded in our software is that the presentation for the result of an ad hoc query is derived from the presentation for a standard query result.

One aspect is the construction of "new" data elements. Typically in the adaptation data elements are left out or introduced into the given set of elements, and the system has to figure out how to construct an elegant presentation. While we assume that we have designs available for the standard slices, whenever an ad hoc query asks for an ad hoc slice (called a *volatile* slice) we need an intuitive and elegant presentation for that slice: we do not want a slice with some empty spots. This problem will be solved using *constraints* that specify the relative position of attributes (see [1]).

3.5 Software Architecture

The current HERA software developments are based on the assumption that the data are expressed in *XML*. We use *XSL(T)* to express the different transformations. Note that we really would want to use the de facto standard for a Web query language, for example a language like XML-QL. However, waiting for a standard we have chosen to go with XSL, without too many problems in the first prototyping experiments. While XSL has been designed primarily to deal with layout and formatting aspects, the recent changes in the language have given us the possibility to use it also to deal with the querying aspects.

The current developments for the HERA software aim at an architecture that is illustrated by Figure 2. In this paper we will not go into the details, but the figure shows the separation of concerns in the *Presentation Manager* and the *Data Manager*. Current software developments try to accommodate the other aspects that were not considered in a previous version of the software. The first experiences from this development show that it is vital that the different parts of the process are very well separated. In the spirit of [3] we now distinguish:

- A *domain* model (DM) that describes the user-independent details of the data being used in the application.
- A *user* model (UM) that stores the relevant aspects of the user that the system uses to produce a presentation that suits the user.
- An *application* model (AM) that stores the application intelligence, both general intelligence (heuristics) and application or domain specific intelligence.

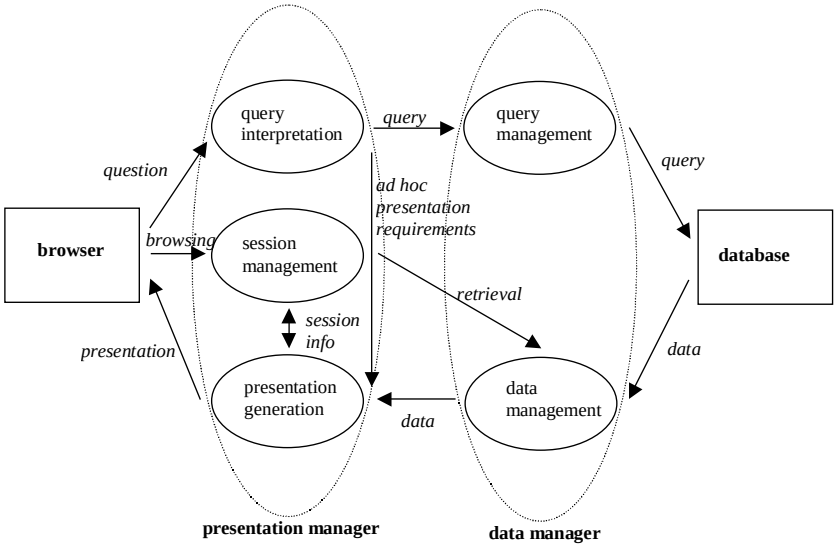

Fig. 2. Conceptual architecture HERA

4 Multiple Heterogeneous Sources

As stated in the introduction this paper describes key aspects of the HERA investigations, identifying problems to solve concerning the integration of data from different sources. Currently, the investigations try to deduce what the consequences are of using a heterogeneous information system (instead of one single data source).

4.1 Interface Heterogeneity

The nature of the target applications implies the use of a heterogeneous set of information sources, e.g. in the EPG application. While we neglect the technical aspect here, we concentrate on the *interface* heterogeneity and the *logical* heterogeneity.[2]

Assuming that different sources require different access methods, the presentation generation process has to decide which data are to be retrieved from the respective sources in order to compose the complete presentation. This *interface* aspect will be dealt with by a layer containing *mediators* and *wrappers*. Their task is to hide the heterogeneity from the core generation process. In the current developments we assume that restrictions associated with the differences in access do not influence the generation process.

[2] For a definition of terminology on aspects of integration, see [2].

The data and the metadata describing their properties are in XML. The task of the wrappers is to translate the data from the individual sources to the universal format that can be exploited in the rest of the process. In a given application *domain-dependent* XML tags could be used to capture domain-specific elements, providing a way to efficiently construct a file with the appropriate meta-data, for example through domain-related interpretation of the tags.

General HERA research questions concerning this wrapping process are:

- Does an effective wrapping process require an interaction with the core presentation generation process?
- Do we want to facilitate the end-user or expert user in changing or overriding the wrapping process on an ad hoc basis?

4.2 Logical Heterogeneity

Logical heterogeneity, *semantic*, *schematic* or *structural*, causes problems for the presentation generation process. Since we assume that the schemas of the data elements are not always available beforehand, and are carried inside the (semi-structured) data, we allow for the possibility that after the retrieval the retrieved data is transformed to deal with this heterogeneity. This means that this heterogeneity is partially dealt with in the *wrapper* layer, but also partially in the (tight) *federation* layer. In the core HERA architecture we have first a phase of *query transformation*, then the actual data retrieval, and then a phase of *data transformation*. The first phase applies the right adaptation to the *query*. The second phase adapts the actual *data*, the query result, based on the information included in the data. If, for example, some TV programs do not contain the expected short preview video, the system might decide to offer a photo instead.

While we do consider semi-structured data (within some restrictions), we do not plan to use *unstructured* data. The core of the generation process depends on the availability of metadata representing information on how the data are to be used.

For HERA this subject implies general questions like:

- Specifically with the Web, data sources can be integrated in an information system without them being aware of this. What are the consequences of this kind of autonomy, e.g. in relation to changes to the data?
- The end-user expresses queries on the basis of an *Application Diagram* that represents a conceptual structure. How does this global Application Diagram relate to the data structures involved? Does the end-user or expert user have to know anything about this relationship?
- What are the consequences of design choices regarding materialization of intermediate query results and presentations, specifically regarding the EPG?

4.3 Metadata

In the core generation process *metadata* on the data are exploited to decide on how to adapt the query and its result. This primarily concerns the already mentioned user-

related metadata, infrastructure metadata and semantic metadata. Several other kinds of metadata, e.g. technical metadata and semantic metadata, play a role in integration.

Two related research questions for HERA are:

- Should the integration-related metadata influence the presentation generation process, or should there be transparency for this process?
- For Internet EPG sources what kind of metadata should be available in order to be able to use the data effectively in the creation of the EPG?

5 Conclusion

In this paper we have considered an important aspect of applications that manage information on the World Wide Web: the automatic generation of hypermedia presentations (front-ends). We have proposed a number of key issues from the HERA research project. That project aims at developing software to deal with semi-structured data retrieved from possibly heterogeneous, legacy information sources. We have addressed characteristic aspects of the software, and thus identified subjects of investigation regarding the integration aspect, e.g. heterogeneity and metadata.

Current software developments aim at serving more of the purposes identified in this paper. This implies that several existing software tools are integrated into one new tool. In the context of the Dynamo project this tool will be applied for the automatic generation of EPGs, which leads to a more domain specific approach.

References

1. Borning, A., Marriott, K., Stuckey, P., Xiao, Y.: Solving linear arithmetic constraints for user interface applications. Proc. ACM Symposium on User Interface Software and Technology, ACM (1997) 87-96.
2. Busse, S., Kutsche, R.D., Leser, U., Weber, H.: Federated Information Systems: Concepts, Terminology and Architectures. Research report TU Berlin 99-9 (1999) http://cis.cs.tu-berlin.de/Publikationen/
3. De Bra, P., Houben, G.J., Wu, H.: AHAM: A Dexter-based Reference Model for Adaptive Hypermedia. Proc. ACM Hypertext'99, ACM (1999) 147-156.
4. Díaz, A., Isakowitz, T., Maiorana, V., Gilabert, G.: RMC: A Tool to Design WWW Applications. Proc. Fourth International World Wide Web Conference (1995) 559-566.
5. Houben, G.J., De Bra, P.: World Wide Web Presentations for Volatile Hypermedia Database Output. Proc. WebNet97, the World Conference of the WWW, Internet, and Intranet, AACE (1997) 229-234.
6. Houben, G.J., De Bra, P.: Retrieval of Volatile Database Output through Hypermedia Applications. Proc. 32nd Hawaii International Conference on Systems Sciences, IEEE Computer Society, (1999) CD-ROM.
7. Isakowitz, T., Stohr, E., Balasubramanian, P.: RMM: A Methodology for Structured Hypermedia Design. Communications of the ACM **38** (8) (1995) 34-44.
8. Isakowitz, T., Kamis, A., Koufaris, M.: The Extended RMM Methodology for Web Publishing. Working Paper IS-98-18, NYU, Center for Information-Intensive Systems, (1998).