Specification of Adaptive Behavior Using a General-purpose Design Methodology for Dynamic Web Applications

Peter Barna, Geert-Jan Houben, and Flavius Frasincar

Technische Universiteit Eindhoven
PO Box 513, NL-5600 MB Eindhoven, The Netherlands
{pbarna, houben, flaviusf}@win.tue.nl

Abstract. Methodologies for the design and engineering of web applications evolve to accommodate the increased dynamic nature of modern web applications. In this paper we show and demonstrate the similarity between the dynamics in web applications and adaptive hypermedia systems using a general purpose model-driven web design methodology (Hera). To do so we use a simple example. We also stress advantages of specifying adaptivity within models defined on the schema level.

1 Introduction

The dynamic nature and increasing complexity of modern web applications require rigorous design methodologies, e.g. WebML [4], OOHDM [10], Hera [11], and UWE [7], that need to evolve to accommodate the increasing demands. When we observe the recent advances in these methodologies, and we do so here in terms of Hera, we see the similarity between dynamic web applications and adaptive hypermedia systems (AHS) where a dynamically updated user model is used to affect the navigation and presentation structure. The goal of this paper is not to introduce yet another reference model for AHS (which is what Hera is not), but to stress and demonstrate with an implementation the fact that existing general-purpose web design methods can be successfully used for the design of adaptive web applications at model level.

2 Related Work

Next to systems originally developed for the purpose of adaptive hypermedia (e.g. AHA! [6], InterBook [3], or KBS HyperBook [8]) and next to reference models for such systems (e.g. AHAM [5], the Munich model [9]), there are general purpose web design methodologies taking into account adaptation and personalization aspects (at different levels). For instance, in WebML [4] the user model (in AHS terminology), there called the personalization sub-schema, is a part of its data model. The content management model specifies how this information is dynamically updated based on user actions. In OOHDM [10], most personalization and adaptation mechanisms are captured in the conceptual (class) model e.g. by means of the user and user group models.

3 Design of Dynamic Web Applications in Hera

3.1 Methodology and Architecture

Hera is a model-driven methodology for web applications that distinguishes a number of design phases, where every phase results in the construction of a particular model describing a specific aspect of the web application:

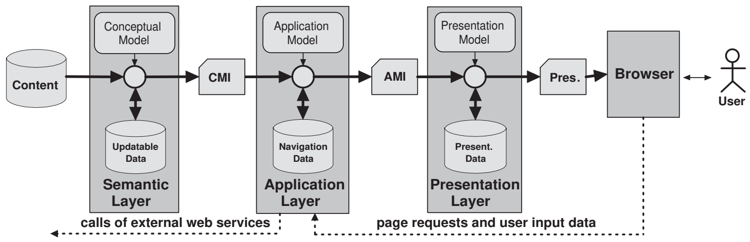

Fig. 1. Overall Hera architecture for dynamic applications

- In the conceptual design phase the Conceptual Model (CM) defining the domain data is constructed.
- In the application design phase the Application Model (AM) defining the structure and behavior of the navigation view over the domain (conceptual) data is constructed.
- In the presentation design phase the Presentation Model (PM) defining the layout of generated hypermedia presentations is constructed.

Figure 1 shows a view of a system hosting dynamic web applications based on the Hera architecture. Hypermedia presentations are dynamically generated page by page, by transforming data from the content subsequently through CM and AM instances to the final format (e.g. HTML) using Hera models. Updatable data (semantic layer), navigation data (application layer), and the presentation data (presentation layer) can be influenced by user inputs and store state information of appropriate layers. All models in Hera are defined in RDFS [1].

3.2 Implementation Example

The example is an excerpt of a virtual shopping basket used in an on-line book shop. Next to it, users' interests in particular authors are modelled and used.

Conceptual and Navigation Data Models Figure 2 (a) shows an excerpt of the shop's conceptual model (the *Book* and *Author* concepts) and a simple virtual shopping basket specified as a part of the navigation data model (the updatable *User* and *Interest* concepts and their properties). The *Interest* concept determines an interest of a user in particular author and is a part of the user model here.

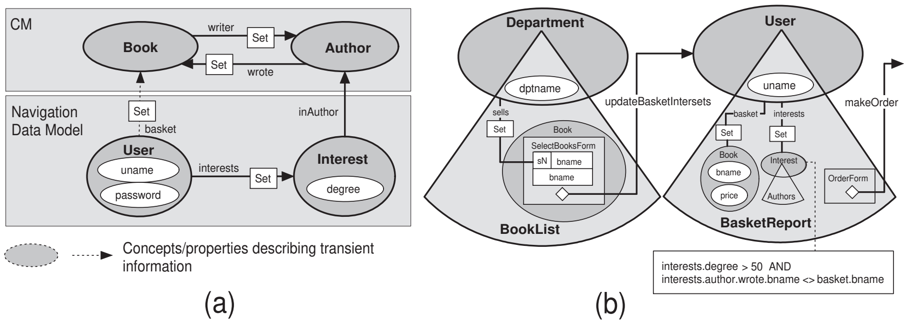

Fig. 2. Conceptual and Navigation Data Models (a), and Application Model (b)

Application Model AM describes the navigation structure of the application and its dynamics in terms of slices (navigation nodes or their parts), their relationships, appearance conditions, and data manipulation operations associated with slice reference relationships. AM is built on the top of CM, updatable data, and navigation data. Figure 2 (b) shows the initial *Department.BookList* slice allowing the user to select a number of books from a list, and the *User.BasketReport* slice showing selected books (the content of the basket) and authors (via the *Interest.Authors* sub-slice) in which the user's interest exceeds a threshold value (50 in this case). The *updateBasketInterests* operation increases and decreases *Interest.degree* depending on the user's selections.

4 Adaptivity Specification with Hera

In our example we placed the user model into the navigation data model, because the perceived state of the user influences the navigation over the data content and typically this does not belong to the content. In most methodologies like WebML and OOHDM all the user-related data is put together with the content. Hera allows to make a strict separation of concerns regarding the modelling of domain content (CM), navigation view over the content (AM and its navigation data model), and presentation of the navigation (PM and its presentation data model, both out of the scope here).

All Hera models define a schema view on data. Hence, compared to the specification of models on instance level it does not limit the extent of the data content that can be conditionally included. In AHAM for example, the designer must specify which instances are included under what circumstances, whereas in Hera the designer specifies appearance conditions without referring to concrete instances, so the concrete data content doesn't have to be known to authors during the design of the system.

All models in Hera, including data models for updatable and navigation data, and the adaptivity conditions in AM, are specified in RDFS [1]. One of the reasons for choosing RDF(S) is that it is a flexible (supporting schema refinement and description enrichment) and extensible (allowing to define new resources) format that enables interoperability. Operations are expressed in SeRQL [2], one of the most advanced RDF(S) query languages.

5 Conclusions

Although the Hera methodology aims to support the design of general purpose web applications, we demonstrated how its support for dynamics in web applications is particularly fit to specify the design of AHS. While the current version of the languages of the Hera methodology (and the associated software for tool support) offers a limited range of adaptation techniques (e.g. conditional inclusion of page fragments), it brings the advantage of schema-level adaptation specification.

References

1. Brickley, D., Guha, R.V.: RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommandation (2004)
2. Broekstra, J., Kampman, A., van Harmelen, F.: Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema, The 1st International Semantic Web Conference, Springer Verlag (2002)
3. Brusilovsky, P., Eklund, J., and Schwarz, E.: Web-based education for all: A tool for developing adaptive courseware. The 7th World Wide Web Conference (1998)
4. Ceri, S., Fraternali, P., Bongio, A., Brambilla, M., Comai, S., Matera, M.: book Designing Data-Intensive Web Applications, Morgan Kaufmann Publishers (2003)
5. De Bra, P., Houben, G.J., Wu, H.: AHAM: A Dexter-based Reference Model for Adaptive Hypermedia. The 10th ACM Conference on Hypertext and Hypermedia, ACM Press (1999)
6. De Bra, P., Aerts, A., Houben, G.J., Wu, H.: Making General-Purpose Adaptive Hypermedia Work, The WebNet 2000 World Conference on the WWW and Internet, AACE (2000)
7. Koch, N., Kraus, A., Hennicker, R.: The Authoring Process of the UML-based Web Engineering Approach. The 1st International Workshop on Web-Oriented Software Technology (2001)
8. Nejdl, W., Wolpers, M.: KBS Hyperbook - A Data Driven Information System on the Web. The 8th World Wide Web Conference (1999)
9. Koch, N., Wirsing, M.: The Munich Reference Model for Adaptive Hypermedia Applications. The 2nd International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, Springer Verlag (2002)
10. Schwabe, D., Rossi, G.: An Object Oriented Approach to Web-Based Application Design. Theory and Practice of Object Systems 4(4) (1998)
11. Vdovjak, R. Frasincar, F., Houben, G.J., Barna, P.: Engineering Semantic Web Information Systems in Hera, Journal of Web Engineering, 2(1&2) (2003)