An RMM-Based Methodology for Hypermedia Presentation Design

Flavius Frasincar, Geert Jan Houben, and Richard Vdovjak

Eindhoven University of Technology
Eindhoven, The Netherlands
{flaviusf, houben, richardv}@win.tue.nl

Abstract. Due to the rapid growth of the Web, there is an increasing need for methodologies that support the design of Web-based Information Systems (WIS). After investigating the application of existing hypermedia design methodologies in the context of automated hypermedia presentation design we propose a specification framework for this context. The framework considers the possibility of dynamically gathering information from a collection of structured, but also possibly heterogeneous sources (relational or object-oriented databases, XML repositories etc.). The methodology associated with the framework shows two levels of abstraction: the logical level, and the presentation level. At the logical level the application diagram captures the design of slices, thus specifying the content related grouping of data elements and their relationships. At the presentation level, the presentation diagram bridges the logical level and the actual implementation by specifying how the design of slices is translated into hypermedia mechanisms, e.g. hyperlinks.

1 Introduction

From its introduction in the early 90's the World Wide Web (WWW) is in a continuous development. Its rapid expansion results in an increasing number of Web-based Information Systems (WIS) [3] being developed, especially with sources that contain frequently changing information such as databases (relational or object-oriented databases, XML repositories etc.). This leads to the fact that there is a higher need to automate, at least partially, the design process of hypermedia presentations as used in a WIS. Although there are methodologies like Relationship Management Methodology (RMM) [1] and Object Oriented Hypermedia Design Methodology (OOHDM) [2], these methodologies have been originally developed for a manual hypermedia design process, they are not particularly well-suited in the context of automated hypermedia design.

RMM focuses on highly structured applications with high information volatility. Moreover, it provides guidelines that can facilitate the automated design process of hypermedia applications. RMM is based on the popular Entity Relationship (E-R) model.

These characteristics are significant for our target application area where sources contain dynamical information. RMM has a specification at the logical level, which groups presentation issues (e.g. navigational links) with semantical issues (e.g. slice attributes). Besides, it lacks a proper specification at the presentation level. In this paper we argue that the separation in two distinct levels is useful and we address them both in the context of the proposed framework.

The results of this research will be exploited in the HERA [12] project that investigates software architectures for the generation of presentations for (semi-structured) data retrieved from heterogeneous data sources. In order to experiment with the proposed methodology a prototype has been built. The prototype uses XML to store the data and XSL to specify transformations between consequent steps. Similar model-driven Web systems inspired by Hypermedia Design Methodology (HDM) [7] were developed in Autoweb [6] (uses relational databases for data storage) and WebML [5] (uses XML for data storage) projects.

Part of this work is also embedded in the Dynamo project, in which Eindhoven University of Technology collaborates with Philips and CWI, both in the Netherlands. The Dynamo project targets the (semi-)automatic generation or design of multimedia presentations. Characteristic for the generation process is the support for adaptation.

As results from [4], [10] show, adaptation can be specified by distinguishing the domain model, the user model, and the adaptation model. The last model describes the actual adaptation being performed (based on the assumed knowledge of the user inferred from the user model). In terms of this project it means that conditions are specified on the existence of objects that are part of the presentation.

2 RMM

RMM [1], [11], [13], [14], [15] proposes a sequence of steps that have to be followed in the design process. For each step there are design guidelines for the designer or builder of the application. Figure 1 depicts the RMM steps that are reused in the proposed framework.

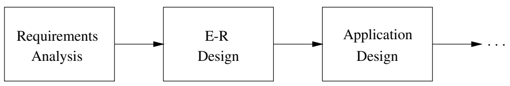

Fig. 1: RMM design methodology

The first design step is to represent the application domain using traditional E-R modeling techniques.

In the second step, the E-R model is enriched with the concepts of m-slices and relationships to build a diagram that describes the application. M-slices

[13] are created by grouping together attributes from the E-R diagram; this can be done hierarchically using previously defined m-slices: the 'm' comes from 'Matreyshka', the nested Russian doll, suggesting the possibility to nest slices in one another. In the rest of the paper we will use the term slice instead of m-slice. The most simple type of m-slice is an attribute ensuring thus the uniformity of the hierarchical slice model.

3 Methodology

We developed a specification framework based on RMM to support the automation of a hypermedia presentation design. Figure 2 illustrates the methodology related to this framework.

Although requirements analysis, E-R diagram (domain analysis), and implementation are important phases in the software engineering life cycle they are beyond the scope of this paper.

The arrows in Fig. 2 show the sequence of steps in the process [1]. Our methodology focuses on the design steps shown in the shaded area of Fig. 2: platform independent application design, platform dependent application design, and presentation design.

The artifacts of each design step in the order in which they appear in the methodology are:

- E-R diagram
- platform independent application diagram
- (platform dependent) application diagram
- presentation diagram

A platform independent application diagram presents common information of application diagrams that is going to be displayed regardless of the platform. A platform dependent application (also referred as application diagram) takes into consideration a particular hypermedia platform.

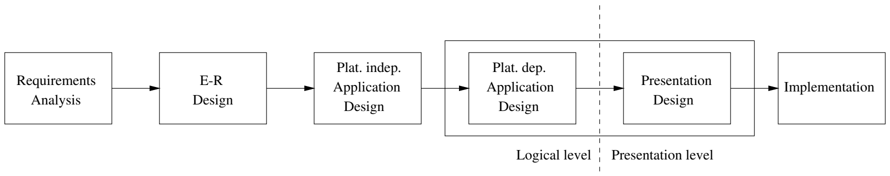

Fig. 2: Design methodology

In the methodology we distinguish two levels, the logical level and the presentation level, based on the 'separation of concerns' principle.

[1] It is possible to have feedback loops in the process.

3.1 Logical Level

At the logical level the methodology is concerned with providing data and relationships/access structures of the information to be presented. Related information is grouped together into meaningful presentation units (slices). Slice relationships provide a means to access information belonging to different units. The logical level is based on the E-R model, in the sense that it exploits the entity structure (attributes) and the relationships among them.

Relationships derived (based on relation's transitivity) from the E-R model are taken into account to minimize the number of slice relationships that need to be traversed in order to reach a meaningful item related to a presentation unit.

The logical level bridges the data source level and the presentation level. The designer is concerned with mapping elements from the data sources (at abstract level entities from E-R model) to slices.

Besides structuring information the notion of slice facilitates reuse. A slice can be reused at application level or even among applications if the applications share the same data model. At application level a previously designed slice can be nested in another slice so that the design effort involved in a new slice creation is considerably reduced.

The logical level is expressed in the application diagram which will be described in Sect. 4.

3.2 Presentation Level

The design choices made at the logical level yield a presentation specification in terms of slices. Such a specification however, stays at the logical level and does not describe the presentation in detail. It does not say anything about how presentation elements should be organized with respect to each other or with respect to the screen; it says neither when, nor in which order presentation elements should appear. These kinds of design decisions were usually made ad hoc during the implementation phase, though they clearly do not belong there.

The presentation level bridges the logical level and the actual implementation. Complementary to the logical level, where the designer is concerned with organizing the overall presentation structure and identifying what attributes from entities are to be included in slices, at the presentation level the designer specifies how and when the identified slices should be displayed.

The presentation level should not be considered only as a detailed logical level but rather as a view from a slightly different angle. During this process, slices from the logical level are mapped into regions and design decisions are made to specify a concrete way (navigational, spatial, and temporal) [9] how rather abstract slice relationships will appear in the presentation. In other words the designer specifies in a conceptual way the navigation, layout, and timing of the presentation.

The presentation level is expressed by means of the presentation diagram which will be described in Sect. 5.

4 Application Diagram (AD)

The application diagram is used to model at the logical level the hypermedia application [15]. An application diagram consists of slices and the relationships among them.

In order to illustrate the framework specification an example of a virtual Web site that presents information about a set of real-estate agencies is used.

Figure 3 describes the underlying E-R diagram without attributes. The application domain is simplified in order not to burden the reader with too many application details. In the example we introduce four entities, house agency, house, neighbourhood, and room, and three relationships, expressing that a house is owned by a house agency, it is placed in a certain neighbourhood, and contains several rooms. All relationships are of type one-to-many.

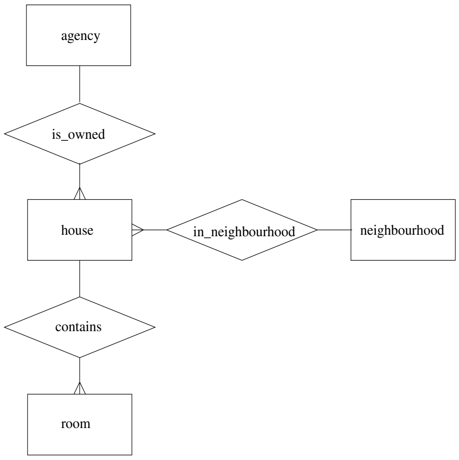

Fig. 3: E-R diagram

4.1 Slices

The application diagram is based on the concept of slice, which groups together attributes and possibly other slices. Each slice belongs to an entity and it can also contain attributes or slices from other entities. Slices provide flexible aggregation and contextual navigation, useful features for intelligent Web navigation.

Slices are linked together with slice relationships. Access structures (e.g. index, tour, indexed guided tour) are used to build complex internal slice structures. Relationships/access structures have associated conditions [2] that show what slice instances are to be connected.

[2] A user adaptation module can use these conditions in order to inhibit or to allow certain links based on the knowledge a user has regarding a certain concept.

The application diagram consists of all the designed slices and provides a global view of the application.

Figure 4 presents a slice (as it is in RMM) which is an element of the larger application diagram.

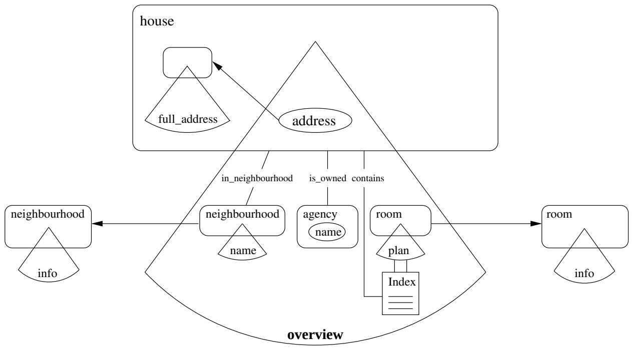

Fig. 4: Slice

4.2 Slice Relationships

Slices are linked together by relationships. A slice relationship associates slices (belonging possibly to different entities) some being sources and others targets. A simple slice relationship connects just two slices while a complex slice relationship connects multiple slices. Each slice relationship can be have conditions (logic predicates) that determine which instances of an entity are to be accessed. Conditions support user and platform adaptation. Figure 5 (left) presents a simple slice relationship.

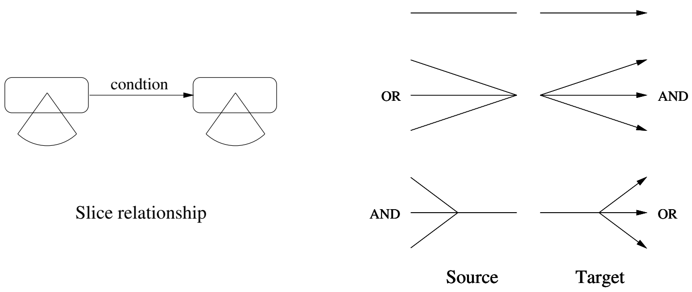

Fig. 5: Slice relationship. Sources and targets for slice relationship

Figure 5 (right) depicts the possible types of sources and targets of a slice relationship. We distinguish the following cases:

– simple source/target
– multi-and source/target, all sources/targets must be triggered/invoked
– multi-or source/target, at least one of the sources/targets is triggered/invoked

One can observe that the proposed notation has as consequence that an 'OR' for sources has the same graphical representation as an 'AND' for targets (and the other way around).

The RMM slice relationship is a navigational relationship (hyperlink) between two slices. In our application diagram a slice relationship represents navigational, spatial, or temporal relationships at a higher level of abstraction. In RMM, slice relationships are always crossing the physical boundaries of a slice. In the proposed framework we allow slice relationships to be embedded in one slice (we call them internal slice relationships). In the case of internal slice relationships the source slice can vanish or can be preserved. Figure 6 presents the associated graphical notation for the two cases.

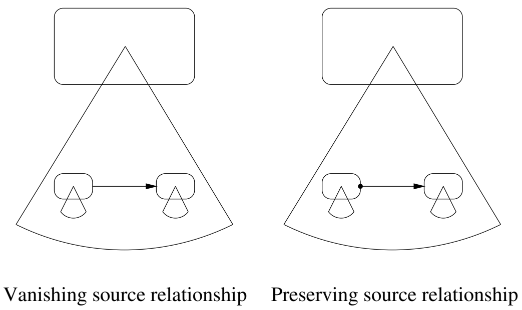

Fig. 6: Vanishing source and preserving source slice relationship

Figure 7 presents a slice diagram that uses the newly introduced concepts of internal relationships and preserving source relationship. The empty slice called 'h_n' is an anchor for two internal relationships (both the anchor and the target belong to the more general 'description' slice). This is an example of relationships that are preserving their source (the bullet anchored on the empty slice). That means that when such a relationship is traversed, the description slice context is kept and the 'h_n' anchor is still present.

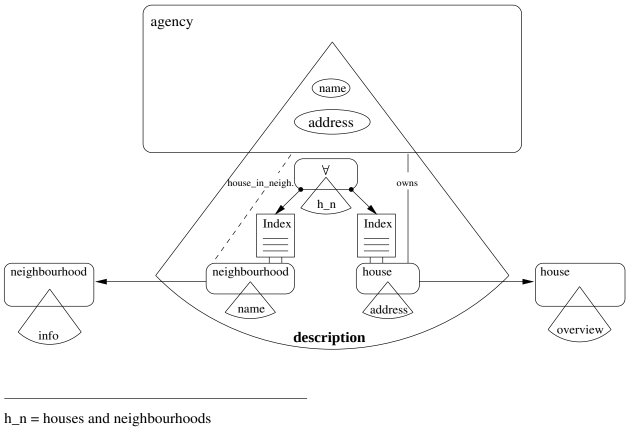

Fig. 7: Slice

5 Presentation Diagram (PD)

Similarly as the logical level is described by means of an application diagram, the presentation level is described by a presentation diagram. A presentation diagram consists of presentation elements called regions and the relationships among them. The presentation diagram specifies how the regions are organized from the navigational, spatial, and temporal point of view. Figure 8 shows an example of a presentation diagram composed from several regions.

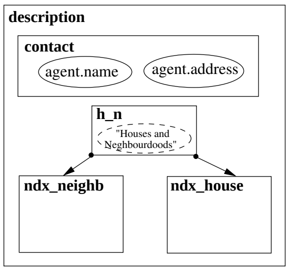

Fig. 8: Presentation diagram

During the process of creating the presentation diagram, the slices introduced at the logical level are mapped to regions and possibly new regions are introduced (it is not always a one-to-one mapping, some slices can be split to more regions and vice versa).

Subsequently, slice relationships from the logical level are materialized with navigational, spatial, or temporal relationships.

5.1 Regions

Regions, together with the relationships that interconnect them, are the main building blocks of a presentation diagram. A region is defined as a collection of attributes and possibly other regions. This recursive definition facilitates nested regions thus providing a means to reuse regions in a component-like manner. Every region is associated with an area where it is displayed. An area is of a rectangular shape having as properties its dimensions and location. When we refer to a region we mean both the region and its associated area.

As mentioned before, a region contains attributes. Attributes are properties that relate concrete values (from a given domain) to the region. Attributes come in two flavors: entity attributes and constant attributes. An entity attribute acquires its value from the entity to which it is mapped and thus it changes its value throughout the presentation. A constant attribute, as its name suggests, does not change its value; the value is determined beforehand and stays constant. Different syntax is used to differentiate the two types. Graphically, entity attributes are depicted as solid line ovals and constant attributes as dashed line ovals as shown in Fig. 8.

Note that though the concept of region is similar to the slice concept, there are some differences:

- A slice is always associated with an entity that owns it, while a region, as a presentation element, is not owned by an entity; it is sufficient that each entity attribute includes the information from which entity it comes from.
- A region has an associated area where it is displayed.

5.2 Region Relationships

All aspects of the abstract concept of slice relationships introduced at the logical level are used also at the presentation level with the difference that the sources and the targets are regions, not slices. Moreover, three instances of this concept are introduced: navigational, temporal, and spatial relationship. Each of them is having a different syntax and different semantics associated with its realization.

Graphically, relationships are depicted as arcs with arrows. Similar to the logical level, an arc starting with a solid circle indicates a persistent (not vanishing) source region. The graphical style of the arc indicates the type of the relationship: navigational, temporal, or spatial as depicted in Fig. 9.

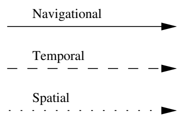

Fig. 9: Types of relationships

Navigational. The navigational relationships were introduced to represent the classical hyperlinks (so called 'click-able' links). After the user clicks on the source region the target region is invoked and the source region either vanishes or stays, depending on its type.

There is a possibility to guard the execution of the link with a condition (the link is followed only if the condition is satisfied) and the possibility to specify an event when the link should be followed. The default event for navigational links is the mouse-click event, but the designer can choose different events (e.g. mouse-over). Navigational links are graphically depicted with solid arcs.

Temporal. Temporal relationships were introduced in order to express the notion of time in the specification of multimedia presentations, while the original RMM was focused on hypermedia presentations (no notion of time). Temporal links can be used to describe both intrinsic delay (a duration of a video or audio clip [8]) and presentation time (temporal links introduced by the designer). There is a time-out event associated with every temporal relationship; after the time elapses the relationship is realized (the destination region is invoked) [3].

Although from semantic point of view the temporal relationships can be considered similar to the navigational links we suggest to make a distinction mainly because of the different nature of the two (navigational links require some user interaction while temporal links do not).

Temporal links can be organized in parallel or sequential compositions. The graphical syntax is rather intuitive; temporal links are graphically described with dashed arcs.

A sequential composition is depicted as a chain of links with one starting node (starting source region) and a chain where the destination region of one link is the source region for the next link.

Two temporal relationships are considered to be in parallel composition if they are not included in the same sequential composition. In other words what is not sequential is parallel.

Figure 10 shows an example of three sequential chains (A*, B*, C*) which are organized into a parallel composition and synchronized before they enter the region D. Note that normally the starting region from a sequence is shown at once (A1, B1) but it is also possible to specify a delay with respect to the top region (C1).

[3] The synchronization can sometimes introduce an additional delay (the time-out of one 'track' elapses but it still has to wait for the other with which it is synchronized).

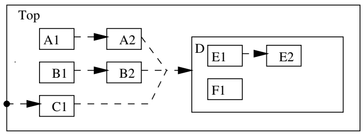

Fig. 10: Sequential and parallel composition with synchronization

Spatial. There are two reasons for introducing spatial links: the first is to provide a means of specifying the layout characteristics of a presentation. The second reason is to facilitate the description of more complex presentation elements like indices, guided tours etc. A spatial link describes a spatial relationship between two regions. The meaning of the link orientation is that the link describes a relative position of the target with respect to the source, the point of reference always being the left upper corner. The semantics of spatial links is slightly different compared to the one of navigational and temporal links. The notion of following a link is missing here, e.g. both the source and the target of a spatial link are displayed at once provided the condition and the event associated with the link permit so.

Spatial links are graphically depicted with dotted arcs.

5.3 Multi-Dimensional Synchronization

Except of the synchronization of temporal relationships our framework offers also multidimensional relationship synchronization where relationships of different kinds are synchronized. This gives the designer the opportunity to specify new types of dependencies among relationships. For example in Fig. 11 a navigational link from the region 'Text' is synchronized with a temporal relationship coming from the region 'Audio'. The target region is invoked only after both events from the involved relationships (mouse-click, time-out) occur.

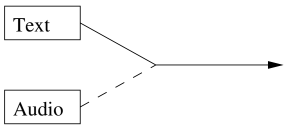

Fig. 11: Multidimensional relationship synchronization

5.4 Ordering of Relationships

A region with an entity attribute is instantiated during the presentation to several regions with concrete values of the attribute.

In order to be able to specify relationships among these instances we introduce the so-called self-relationships at the type level. Since there might be more of such relationships needed to describe the desired presentation structure, there must be a unique order in which they are applied. To facilitate this, the ordering of relationships was introduced. In Fig. 12 we provide two examples to illustrate this approach. Presentation diagrams are depicted on the left and the respective screen renderings on the right.

Figure 12 (top) describes a bulleted index region. It consists of a region with a constant attribute depicting the bullet ('·') right to which is positioned the region 'item' containing the 'N.name' attribute. From this region there is a link to the region called 'info' containing the 'N.description' attribute. The 'bullet' and the 'item' region are grouped together in the region called 'enum', which has a spatial relationship (with the lower priority 1) to itself saying that the next pair bullet-item will be placed below. In other words the constraint introduced by this relationship is taken into account after applying the constraints from the relationships that have the higher priority (0).

Figure 12 (bottom) describes a bookcase region. It consists of the 'shelf' region, which contains the 'thumb' region that has a navigational link to the 'picture' region and also a link to itself saying that the next thumb will be placed to its right. This rule is applied until there is no room left in the shelf (the horizontal dimension of the 'shelf' region). When the shelf is full we proceed with the next one by applying the self spatial relationship of the 'shelf' region which has the lowest priority (2).

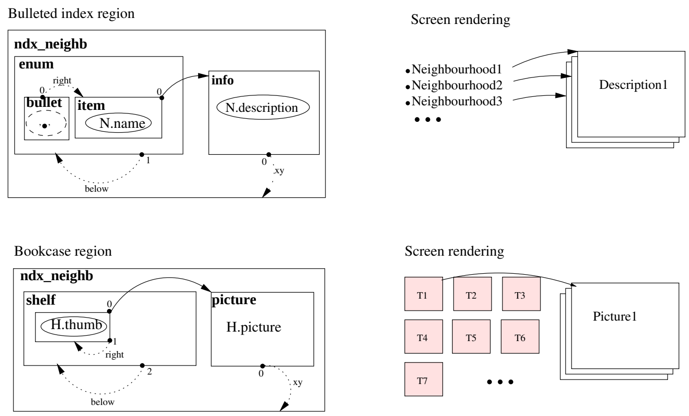

Fig. 12: Bulleted index region. Bookcase region

6 User Adaptation

In our framework we try to address the (semi-) automatic generation and design of multimedia presentations. Characteristic for the generation process is the support of adaptation.

The system can adapt to the user by considering the user's browsing behavior. By observing the navigation behavior the system can infer (based on intelligence specified by the application's author) the user's knowledge and adapt the presentation on the basis of this assumed knowledge.

There is also adaptation to the platform the user is using (platform dependence): if for example the system detects that the user is using a PC with a large screen or a small hand-held device, then the system could adapt the presentation to this platform.

Slices or slice relationships are available or accessible only after the conditions hold. While the specification of the actual adaptation process is beyond the scope of this paper (AHAM model from [10]), the concepts (in terms of AHAM) are specified in this framework as slices and slice relationships. Hence, these slices and slice relationships carry (in the conditions) part of the knowledge on which the adaptation is based. In order to specify hypermedia presentations, especially in the context of automatic design, it is necessary to describe which concepts are part of the design. Our framework allows to specify exactly that.

7 Conclusions

The research described in this paper targets the support of automated hypermedia presentation design (in the context of Web-based Information Systems). Specifically, for applications involving data that change frequently, the design of hypermedia presentations requires a structured approach. This paper discusses a methodology that guides through the process of specifying the presentation generation process. The methodology is inspired by RMM and overcomes some of RMM's shortcomings for the application to automated presentation generation. It also shows the specification techniques used in the steps of the methodology.

Our methodology distinguishes three steps in the design process. First, a logical application design is made to organize the data elements at a logical level, even independent of the platforms used for the presentation. Secondly, at the logical level some of the specific details related to the platform that is used are taken into consideration: this leads to a platform-dependent logical application diagram. The third step in the process realizes the logical decisions made in the first two steps, by carefully choosing representations for the different constructs. At this level, the dimensions of hyperlink navigation, space and time are included in the specification. This paper concentrates on the specification of the different design decisions and illustrates them using the diagrams.

In further work we will extend these specification techniques with other constructs that allow for the specification of some of the other relevant details. In the Dynamo project we are experimenting with different kinds of applications such as the Electronic TV-Program Guide. In these experiments we learn how different aspects play a role in the automated hypermedia presentation design process. We aim at including more of these aspects in this specification framework. One example relates to the use of constraints (rules) in the specification of spatial relationships. Such constraints might not (easily) lead to a solution, or even influence other design decisions (made earlier in the design process). Other aspects relate to the combination of different kinds of relationships, or the construction of 'virtual' slices or regions.

In the Dynamo project the adaptation is an important feature. The conditions associated with slices and regions allow for implementing different aspects of adaptation, such as user adaptation (navigation) and platform adaptation. The Hera prototype is developed incrementally, adding new features to the methodology enables further verifying whether this mechanism suffices in concrete applications.

References

1. Balasubramanian, P., Isakowitz, T., Stohr, E.A. : RMM: A Methodology for Structured Hypermedia Design. Communications of the ACM, 38(8):33-44, August 1995.
2. Barbosa, S.D.J., Rossi, G., Schwabe, D.: Systematic Hypermedia Application Design with OOHDM. Hypertext, 1996.
3. Bieber, M., Isakowitz, T., Vitali, F.: Web Information Systems. Communications of the ACM, 41(7):78-80, July 1998.
4. Brusilovsky, P., De Bra, P., Houben, G.J.: Adaptive Hypermedia: From Systems to Frameworks. ACM Computing Surveys, 31(4es), Article No. 12, December 1999.
5. Ceri, S., Fraternali, P., Bongio, A.: Web Modeling Language (WebML): a modeling language for designing Web sites. WWW9, 2000.
6. Fraternali, P., Paolini, P.: Model-Driven Development of Web applications: the AutoWeb system. ACM Transactions on Information Systems, 18(4):323-382, October 2000.
7. Garzotto, F., Paolini, P., Schwabe, D.: HDM - A Model-Based Approach to Hypertext Application Design. ACM Transactions on Information Systems, 11(1):1-26 , January 1993.
8. Bulterman, D.C.A., Hardman, Mullender, K.S.L., Rutledge, L., van Ossenbruggen, J.: Do You Have the Time? Composition and Linking in Time-based Hypermedia. Hypertext and Hypermedia, 1999.
9. De Bra, P., Houben, G.J.: Automatic Hypermedia Generation for ad hoc Queries on Semi-Structured Data. Digital Libraries, 2000.
10. De Bra, P., Houben, G.J., Wu, H.: AHAM: A Dexter-based Reference Model for Adaptive Hypermedia. Hypertext and Hypermedia, 1999.
11. Diaz, A., Isakowitz, T.: RMCase: A Tool To Design WWW Applications. Fourth International World Wide Web Conference, 1995.
12. Houben, G.J.: HERA: Automatically Generating Hypermedia Front-Ends for Ad Hoc Data from Heterogeneous and Legacy Information Systems. Engineering Federated Information Systems, 2000.
13. Isakowitz, T., Kamis, A., Koufaris, M.: Extending RMM: Russian Dolls and Hypertext. HICSS, 1997.
14. Isakowitz, T., Kamis, A., Koufaris, M.: Reconciling Top-Down and Bottom-Up Design Approaches in RMM. Workshop on Information Technologies and Systems, 1997.
15. Isakowitz, T., Kamis, A., Koufaris, M.: The Extended RMM Methodology for Web Publishing. New York University, 1998. Available online at `http://rmm-java.stern.nyu.edu/rmm/`.