Dynamic web sites

M. Gabbrielli
Università di Pisa *

M. Marchiori
W3C/MIT§

F.S. de Boer
Universiteit Utrecht **

Abstract

We propose a formalism for dynamic web sites, based on the RMM model, an existing methodology for the design of hypermedia applications. We provide an abstract definition of its data and navigation model; then we investigate some limitations in the expressive power of such a model and we extend RMM by introducing some simple, dynamic features. This enhances the navigation model, and makes interaction with databases much more effective.

1 Introduction

The recent growth of the Internet and the World Wide Web has considerably augmented the relevance of hypermedia applications. On one side, the Web itself can be seen as a hypertext domain whose global nature introduces several novel theoretical issues and requires new programming formalisms. From a more pragmatic perspective, the continuing request of new Web pages led to demand for methodologies for the fast and reliable development of hypermedia applications. These methodologies are substantially different from the traditional software development techniques, since hypermedia applications typically involve structuring complex information domains and describing sophisticated navigations among data.

A few proposals in this field appeared recently: the Hypermedia Design Model (HDM) by Garzotto et al. [4] allows to describe complex multimedia structures and their dynamic properties, however it does not provide explicitly a methodology for the hypedmedia design and development. This is provided by the Relationship Management Methodology (RMM) defined by Isakowitz et al. [8]. In fact, RMM defines a data model based on HDM and describes how such a model should be used in the different phases arising when developing an application. RMM was defined by using a graphical notation and (borrowing the database terminology) it can be seen as tool for the conceptual and (partially) for the logical design of a large hypermedia application. Even though the final result of such a design is HTML (XML) code, clearly a more abstract (graphical) model is helpful.

A different structured approach to the hypermedia design has been described by Gaulding and Katz [5], while Object-Oriented methodologies have been studied by Lange [9] and by Schwabe and Rossi [10].

In this abstract we focus on the RMM approach. More precisely, we consider its data and navigation model, that is the formalism used to represent the data, their relationships and the navigation structure. In order to simplify the terminology, in the following we will identify such a formalism with RMM.

Since RMM has been described only informally so far, we first provide an abstract, formal definition for it. For the sake of brevity we do not use graphical notation here, however clearly all the concepts introduced in the following have an immediate graphical correspondent. Then we propose a simple extension which increase the expressive power of RMM. The need for an extension of the original definition

*Dipartimento di Informatica, Università di Pisa, Corso Italia 40, 56125 Pisa, Italy. gabbri@di.unipi.it.

§545 Technology Square, Cambridge, USA massimo@w3.org

**Department of Computer Science, Universiteit Utrecht, Padualaan 14, De Uithof, 3584 CH Utrecht, The Netherlands. frankb@cs.ruu.nl

in [8] was soon recognized by the authors themselves [7]: After a few experiments they realized that RMM was not adequate for many typical applications, notably for the design of non-trivial Web pages. For example, it was not possible to model the situation in which a single Web page contains information deriving from several different "entities" used to describe the Web site, and it was not possible to specify what information was the content of the anchor to the current page. This limitations did not allow to express many natural navigational patterns: In many cases, a (conceptual) hyperlink had to be splitted into several ones, thus forcing the user to visit several different pages before reaching the interesting one. In order to solve these problems RMM was then substantially extended [7] by replacing its basic data modeling primitive, the slice, for a new one, called m-slice. Differently from slices, m-slices can describe nested structures, can contain both data and access structures and, in general, allow to model many common navigation patterns. However, as argued in the following, they are still unable to describe some quite simple situations which arise in the design of Web pages. As a simple example, consider the case of a Web page listing the names of some authors. Each name is an anchor to another page containing further information on that specific author (e.g. the affiliation) together with an anchor to her/his recent publications. Clicking on this latter anchor one gets a third page listing the titles and the abstracts of the publications of the author. It is quite natural to require that this third page contains also an anchor "author", linking back the list of publications to the page containing the author's information. However, this is not possible with the current version of RMM unless we modify the third page by introducing some redundancy (e.g. by including all the names of the co-authors as anchors) and therefore making the application more cumbersome and inefficient.

As discussed more precisely in the next Section, the problem here is that the access structures are statically determined in RMM. In other words, given an anchor one can always statically determine the value of its target. As esemplified above this is restrictive, since in some cases one one needs also some dynamic capabilities: The value of the target of the anchor "author" depends dynamically on the previous navigation path, in fact it is the last visited instance of the page containing author's information.

In order to allow the modeling of these navigation patterns we extend RMM by introducing some, very simple, dynamic features. We first provide an abstract formalization of m-slices and of the related navigation modeling structure, called RMM diagram. Then we extend diagrams by introducing the notion of "dynamic link" which, intuitively, allows to obtain dynamically the value of the target of an anchor, on the basis of the m-slices previously encountered in the navigation path. This provides a smooth extension of the existing RMM framewrok and hence allows us to retain the existing design and development methodology.

The resulting model can be seen as a first step toward the definition of a uniform framework for integrating the three main aspects arising in the design of a web site, namely the creation of HTML (XML) documents, the the design of the user interface and the access to semi-structured information. In fact the m-slice construct, together with the dynamic features we introduce (possibly extended by adding some Datalog-like capabilities) allows one to model all these three aspects. In particular, concerning the third aspect, currently the tools based on RMM are interfaced with Databases by using the standard technology based on the generation of SQL queries (the automatic generation of queries from graphical m-slices is currently invesitgated by the authors of RMM). However a more promising approach seems that one of using the data model of RMM to describe directly both traditional (structured) data and semi-structured data: These are defined in [1] as those data which are "neither raw nor strictly typed, i.e. not table-oriented as in the relational model or sorted-graph as in object databases". These data typically arise in Web application were several different kind of (structured) data need to be integrated and where often data are structured by using a large degree of irregularity. Consider, for example, the Web site of a CS department: the HTML documents describing the staff have some common basic structure, however often the web pages of different people are rather different (some fields are missing, the same information is presented in different ways and different order etc.). Some theoretical model for describing semi-structured data have been studied (see for example the Object Exchange Model in [2]) and also some pragmatic solutions have been proposed (notably the use of RDF for processing metadata). In this respect, the model based on m-slices could be proposed as an alternative solution. In particular, the (possibly extended) dynamic features we introduce should simplify the task of defining a query language which allow to extract information from semi-structured data described in terms of the RMM data model.

2 A limitation of RMM

From an abstract point of view, RMM can be seen as an extension of the Entity-Relationship (E-R) model obtained by introducing a new primitive for data-modeling (the m-slice) and a new kind of relation for representing hyperlinks. We recall here that an E-R diagram is a data model consisting of a set of entities and of a set of relationships between entities. An entity is an uniform set of abstract or phisical objects (e.g. the authors) which are called entity instances and usually it is specified by defining its name and a set of attributes for its instances (e.g. name, affiliation etc.). In a sense and entity can then be considered as a type, and in the following we will identify an entity with its name. A relationship associate the instances of two entities and can be one-to-one or one-to many (many-to-many relations can be represented by pairs of one-to many relations). For example, given the entities author and publications one can have the relation is_author_of associating an author with his/her publications. In the context of hypermedia applications it is also important to identify the conceptual object grouping the information to be presented to the user (e.g. as the content of an Web page). Such an object cannot be identified with an entity, since an entity usually has a large number of attributes of different nature and only a small part of them has to be presented. Moroever, often one would like to combine on the same presentation unit attributes from different entities (e.g. the name of an author and a list of the titles of its pubblications). For these reasons the slice and (afterwards) the m-slices constructs were introduced in RMM as the basic primitives which allows to group information from entities and which formally represent the information to be presented to the user. In the next Section we will provide a formal definition of m-slices and of the way to combine them into an RMM diagram. For the time being, we just observe that m-slices have to be connected into a diagram in order to represent the relations existing among them. Clearly such a diagram has to be defined on the basis of the E-R diagram which we assume being already defined. However, the RMM diagram contains also informations on the hyperlinks existing among m-slices in order to model the navigation structure. This information, which is expressed in terms of anchors and of their targets, is typical of hypermedia models and is missing in the traditional E-R diagrams. For example, the title of a pubblication can be represented as an anchor whose taget is the m-slice describing the page which contains the abstrat.

3 Dynamic RMM

In this section we first formalize the notion of m-slice and then introduce our notion of dynamic RMM diagram. Here and in the following we assume given an underlying E-R diagram consisting of a set of entities $E = \{e_1, \ldots, e_n\}$, where each entity e_i is a set of entity instances, and a set $R = \{R_1, \ldots, R_m\}$ of binary relations on entities, so $R_i \subseteq e_j \times e_h$ for each $i \in [1, m]$ and for some $j, h \in [1, n]$ with $j \neq h$. Note that we abstract from the attributes here, since these are not relevant in our context. In the following we will use e, e', $e_1 \ldots$ as typical elements for entities, ι, ι', $\iota_1 \ldots$ for entity instances and m, m', $m_1 \ldots$ for m-slices. For the sake of convenience we assume that E contains a special "unspecified" entity, denoted by $e_\perp$, and that R contains unspecified relations, denoted by $R_{e,\perp}$, such that $R_{e,\perp} \subseteq e \times e_\perp$ for each $e \in E$.

Definition 3.1 (m-slices)
Given a E-R diagram D, we define inductively an *m-slice* as an object of the form

$$e(\langle R_1, m_1\rangle, \ldots, \langle R_n, m_n\rangle) \qquad (n \geq 0)$$

where e denotes an entity in D, called the *owner* of m, $R_1, \dots, R_n$ are (symbols denoting) relations in D and $m_1, \dots, m_n$ are m-slices such that, for each $i \in [1, n]$, $R_i \subseteq e \times e_i$ holds where e_i is the owner of m_i. If $n = 0$ then the m-slice is called *basic* while if $e = e_\perp$ then the m-slice is called *empty* and in this case we assume that $n = 0$. □

As previously mentioned, an m-slice is used to group information from different entities into meaningful information units to be presented to the users. Intuitively, one can consider an m-slice as the formal representation of the information to be visually rendered by a Web page. As a such, an m-slice has to specify which is the relevant information, i.e. which attributes have to be selected, and how such an information is structured and presented. However, since here we are interested in an abstract representation of m-slices and of their computational behaviour, we do not take into account these issues which are complitely ortogonal to our formalization.

According to the syntax introduced above, an m-slice $m = e(\langle R_1, m_1 \rangle, \dots, \langle R_n, m_n \rangle)$ is essentially a tree whose root is labelled by the *owner entity* e and whose immediate sub-trees[1] are the m-slices $m_1, \dots, m_n$, while the symbols R_i can be seen as labels for the edges connecting the root to the immediate sub-trees. An m-slice is used to group information concerning its owner entity. Clearly, this can involve the use of information from other entities which are related in the E-R diagram to the owner entity. Indeed, according to previous definition, an m-slice m can inductively contain other m-slices which are owned by different entities, provided that these entities are related to the owner of m by the relations R_i in the given E-R diagram. Thus, in a sense, these relations are used to specify the internal structure of the m-slice. An empty m-slice is used to represent constant information and for this reason is has $e_\perp$ as owner.

It is worth noting that this nested nature of m-slices allows the design of rather complex hypermedia applications. Indeed, as previously discussed, m-slices were introduced in [7] to replace the slice construct, originally used in RMM [8], since the latter allowed to represent only a flat group of attributes from the same entity.

According to the original definition in [7], *index m-slices* are identified as a special case: An index is an access structure which acts as a table of content, thus allowing a direct access to a list of instances of the same entity. In terms of the E-R diagram an index corresponds essentially to a one-to-many relationship, where all the values related to a given one are directly accessible. We do not distinguish index m-slices, since these can be treated uniformly as the others.

In the following we adapt to m-slices (a slight modification of) the terminology in use for trees. So, given an m-slice $m = e(\langle R_1, m_1 \rangle, \dots, \langle R_n, m_n \rangle)$ we say that m is the parent of m_i and that m_i is an immediate sub-slice of m, for each $i \in [1, n]$. Moreover we define the relation "sub-slice" as the transitive closure of the relation "immediate sub-slice", i.e. we define inductively m' a sub-slice of m if either m' is an immediate sub-slice of m or m' is a sub-slice of an immediate sub-slice of m. We also say that m' is included in m if either $m = m'$ or m' is a sub-slice of m.

If an m-slice is not contained in any other m-slice, then it is called a *top m-slice*.

In order to represent the network which corresponds to the hypermedia application to be designed, m-slices are organized into a so called RMM-*diagram*. This specifies hyperlinks existing among them on the basis of the given E-R diagram. Here and in the following, given a generic relation R we often write $m \; R \; m'$ to mean $(m, m') \in R$.

Definition 3.2 (RMM-diagram)
Let D be the given E-R diagram. An RMM-diagram consists of

1. a (finite) set M of m-slices defined on D;
2. for each relation R in D a relation $\xrightarrow{R} \subseteq M \times M$, called link, such that $m \xrightarrow{R} m'$ iff m has the form $e(\langle R_1, m_1 \rangle, \dots, \langle R_n, m_n \rangle)$ where $m' = m_i$ and $R = R_i$ for some $i \in [1, n]$;

[1] A tree T is an immediate sub-tree of T' if T is rooted in a son of the root of T'.

3. for each relation R in D a function $\stackrel{R}{\longmapsto} \subseteq M \times M$, called *hyperlink*, such that if $m \stackrel{R}{\longmapsto} m'$ then the following conditions hold:

   (a) m is a basic m-slice;

   (b) m' is a top m-slice;

   (c) m is not included in m';

   (d) there is no other m'' and relation R' such that $m \stackrel{R'}{\longmapsto} m''$.

   Moreover, if $m \stackrel{R}{\longmapsto} m'$ we say that m is the anchor and m' is the target of the hyperlink.

□

Note that an RMM diagram can be seen as a directed graph, the m-slices being the nodes and the (hyper)links being the edges. Therefore in the following we will use for RMM diagrams some terminology of graphs, in particular we will use the standard notion of (directed) path.

The links describe the internal structure of m-slices, according to the relations R_i previously introduced. The hyperlinks define the external connections for m-slices, which are also defined on the basis of the E-R diagram: If there exists an hyperlink $m \stackrel{R}{\longmapsto} m'$ then the owner entities of m and m' are related in the E-R diagram by the relation R.

The hyperlink allows one to activate the target m-slice thus modeling the navigation behaviour. More precisely, as formally described in the next Section, the activation of an hyperlink $m \stackrel{R}{\longmapsto} m'$ is obtained by "clicking" on the anchor m, thus selecting an instance on the owner of m, and causes the "activation" of the target m', thus instantiating the owner entity of m' to a specific set of instances, according to the relation R in the underlying E-R diagram. Some attributes of these instances can then be displayed and presented to the user according ot the definition of the m-slice m'.

As discussed in the previous Section, this computational mechanism provides only a limited form of navigation it does not allow to model some quite natural situations.

To overcome this limitation we introduce a new kind of (hyper)link which allows us to obtain *dynamic definitions*. We introduce the new special symbol Δ to denote a special one-to-one relation in the E-R diagram. Accordingly we have also special relation $\stackrel{\Delta}{\longmapsto}$ in the (dynamic) RMM diagram which represent dynamic hyperlinks.

Definition 3.3 (Dynamic RMM diagram)
Assume given an E-R diagram D and a set of m-slices M. A dynamic RMM diagram is an RMM diagram according to Definition 3.2, provided that D has been extended with a special one-to-one relation symbol Δ different from all the other symbols. □

Differently from the usual relations appearing in an E-R diagram, Δ can relate also several pairs of different entities. Moreover, according to Definition 3.1, the presence of the relation Δ affects also the definition of m-slices by permitting dynamic links.

Roughly, the activation of an hyperlink $\stackrel{\Delta}{\longmapsto}$ consists in the activation of the target m-slice which "inherits" the value for its owner entity from the navigational path followed by the user. More precisely, when activating a dynamic link $m \stackrel{\Delta}{\longmapsto} m'$ where e is the owner entity of the target m', the specific instance of e to be selected is given by the instance of the last occurence of e which has been encountered in the navigational path to m'. Analogously, the relation Δ in the definition of an m-slice is used to specify the fact that values of some (nested) m-slices depends dynamically on the previous path.

In order to guarantee that such an "inheritance" mechanism is well defined we need some further restrictions on the RMM diagram. So we have the following definition where, according to the graph intepretations of a diagram, a path is a simply a sequence of m-slices.

Definition 3.4 [Well Definedness]
Assume given a dynamic RMM-diagram D. Call a path $(m_1, m_2, \ldots, m_n)$ in D *e-free* if e is not the owner entity of m_j, for each $j \in [2, n-1]$. Then we say that D is well defined if for each Δ link l in D the following holds: if e is the owner entity of the target m of l then there exists a unique m-slice m', denoted by $\mathcal{D}(m)$, such that its owner entity is e and there is an e-free path from m' to m. □

References

[1] S. Abiteboul. Querying Semi-Structured Data. *ICDT*, 1997.

[2] S. Abiteboul, D. Quass, J. McHugh, J. Widom and J. Wiener, The Lorel Query Language for Semistructured Data. *International Journal on Digital Libraries* , 1996.

[3] F. de Boer, M. Gabbrielli and M. Marchiori. Dynamic RMM. Draft. 1997.

[4] F. Garzotto, P. Paolini and D. Schwabe. HDM: A model-based approach to hpertext application design. *ACM Trans. Off. Info. Syst.* 11(1): 1–26. 1993.

[5] J. Gaulding and B. Katz. Hypermedia application design: A structured approach. In *Designing User Interfaces for Hypermedia*. W. Shuler, J. Hannemann and N. Streitz eds., Springer-Verlag, ESPRIT series, 43-52. 1994.

[6] L. Hardman, D. Bulterman and G. van Rossum. The Amsterdam hypermedia model. *Communications of the ACM*, 37(2): 50–63. 1994.

[7] T. Isakowitz, A. Kamis and M. Koufaris. Extending RMM: Russian Dolls and Hypertext. In *Proc. of the 30th Hawaii Int'l Conf. on System Sciences*. 1997.

[8] T. Isakowitz, E. Stohr and P. Balasubramanian. RMM: A Methodology for Structured Hypermedia Design. *Communications of the ACM*, 38(8): 34–43. 1995.

[9] D.B. Lange. Object-Oriented hypermodeling of hypermedia-supported information systems. In *Proc. of the 26th Hawaii Int'l Conf. on System Sciences*, 380-389. 1993.

[10] D. Schwabe and G. Rossi. The Object-Oriented Hypermedia Design Model. *Communications of the ACM*, 38(8): 45-46. 1995.