Authoring a Literary Hypermedia Encyclopedia CD-ROM Using Hypermedia Modeling Technique

Fabio Nemetz
Jose Valdeni de Lima*
Altair Cardoso Borges

Abstract

This paper presents the authoring process for a Literary Hypermedia Encyclopedia CD-ROM. The process follows a modeling and design technique for hypermedia applications called HMT (Hypermedia Modeling Technique). First, we describe some problems related to the implementation of this kind of application. Next, we describe the object and the hyperobject models, as well as the navigation model, that are appropriate for hypermedia. Last, we present the implementation of a hypermedia application that deals with literary information in Brazil to show the viability of the HMT technique described.

1 INTRODUCTION

After three decades of research in the hypermedia field, many identified problems haven't been completely solved yet. For example, problems dealing with disorientation, cognitive overhead, interface quality, interactivity and components structure. Great attention has been given to the user's disorientation.. Disorientation happens when the user, during navigation in the hypertextual network [1, 10]:

- doesn't know where he is;
- doesn't know how to move on to another place;
- doesn't know where he came from, and/or,
- doesn't know what could be seen from a particular point.

This happens because users of large hypermedia applications have to deal with a huge structure of connected nodes, but the interface typically shows only a restricted view of this structure [9]. Some solutions were proposed to solve this navigation problem [8]: local and global maps, fisheye views, bookmarks, backtrack, etc. However, these solutions do not guarantee an efficient navigation. They assume that disorientation is intrinsic to navigation, and therefore the user should be offered a mechanism that allows him to be reoriented.

*valdeni@inf.ufrgs.br - Instituto de Informática, UFRGS, Porto Alegre, RS, Brasil

To avoid or to diminish disorientation, it is necessary that the users understand the existing relations among the nodes of the hyperdocument. To achieve this, the developer has to organize the hyperbase in coherent structures. Further more, he has to make these structures clear to the user. Based on hypermedia application development research [2, 3, 4, 5,13] and in the object oriented paradigm [11], HMT - Hypermedia Modeling Technique - was created. It allows the modeling and design of high quality hypermedia applications.

In this paper, we will show how HMT can contribute/help in the construction of hypermedia applications where the risk of disorientation is low (or even non-existent). We will also show how this technique was applied in the literary domain, describing a CD-ROM that we have built.

2 THE HMT

A modeling and design technique for hypermedia applications should help the developer to answer three fundamental questions [4]:

1. how to divide the information domain in nodes;
2. how these nodes are connected, and
3. how the user interacts with the application.

The HMT uses four models to describe an application: the object model describes the domain application objects and their relationships; the hyperobject model refines the object model, adding more semantics to the relationships; the navigation model describes the links and access structures; and the interface model contains the descriptions on how the user will perceive the hypermedia objects.

2.1 The Object Model

The HMT object model uses the concepts and the notation of the OMT (Object Modeling Technique) [11] object model. This model describes the structure of objects in a domain - their identities, relationships to other objects, attributes and operations. An object class describes a group of objects with similar properties, common behavior, common relationships to other objects, and common semantics. The classification of objects depends on the purpose of each application. The purpose of object modeling is to describe objects. This is achieved by the use of object diagrams, as is done in OMT. The concepts of class, attributes, operations, association, cardinality, generalization and aggregation are used. One of the main advantages that the object model brings to hypermedia is the fact that the relationships are treated as explicit constructions, and not as attributes in each class.

The object model captures the application domain semantics. In the point of views of hypermedia, each class of the object model is a candidate to be mapped in one or more

types of nodes; but also many classes could originate only one type of node. Associations are candidates to be links or access structures, depending on the cardinality and the desired kind of access. The important thing is that this kind of design decision is not done in the object model.

2.2 The HyperObject Model

The hyperobject model is a refinement of the object model. As there are a lot of applications that already have the object model, the hyperobject model could reuse it. Besides this, many project decisions are included in the hyperobject model. The emphasis here is:

- defining new classes and associations that define desired paths (not captured by the object model);
- identifying the media that will be used, and
- identifying abstract classes.

2.3 The Navigation Model

Associations are the "glue" of the object model [11], providing access paths between objects. Associations are conceptual entities useful for conceptual modeling. However, during the navigation modeling phase, some strategies must be created to guide the implementation of these associations. In the object model, an association is an abstraction that indicates that one class is related to another one. But, in the design perspective, how should an association be represented? As a link? As a group of links? As an access structure? We should carry out a careful analysis of each association, taking into account the way they will be used in the application.

Besides this, we define navigation contexts in this model. These contexts are used not only to enhance the links semantics, but also as input to the interface model. The idea of navigation contexts is inspired in the "Art Gallery" application [6] and in [12]. Every object is part of a default context and could be associated with other contexts. The navigation can be context-sensitive or context-independent. In the first case, navigation respects the current context. In the second one, while traversing a link, it takes up the default context of the destination object.

Finally, we have to define the application entry-points. Each context can define one entry-point, according to the application requirements. These points indicate from where it is possible to start navigation.

2.4 The Interface Model

Users could be seriously constrained by the divisions of the material predetermined by the author, where a particular piece may be viewed in many different contexts, or arrived at different routes [14]. Designers must consider not only the structure, which must be shown explicitly to the user, but also the way information should be presented, considering the

user context in any part of the hyperbase. Therefore, some mechanisms are necessary to keep a sense of consistent and absolute orientation in the information labyrinth.

Using the interface model, the developer describes how information will be presented to the user. Interface design involves layout definitions of the screens, the appearance of the objects and the visual identity. This definition is based on the hyperobject and on the navigation model.

3 THE LITERATURE CD-ROM

A literary application was developed - modeled, designed and implemented - by using HMT. This was done to evaluate the practical use of the technique. The result became a CD-ROM: the multimedia encyclopedia of Brazilian literature. In this section, we will describe this development, as well as the results. The interface model, not presented in this paper, can be found in [7].

3.1 The application

The application implemented should allow the user to access information about literature, more precisely, Brazilian literature. It should be possible to check authors and books. Authors are classified in periods: *XIX Century, Regionalism, Romance of the 30's*, and *Contemporary Fiction*. Besides this, it should be possible to access associative information, such as authors that are related to others, or authors from a particular literary period. For each book, there should be a review and a summary. .And, if convenient, some excerpts that illustrate a specific aspect. Also, a full-text search should be possible. The book covers should be presented, as well as pictures of the authors. And the passages should be read by a narrator.

3.2 The Object Model

The object model is built as is recommended in [11]: (1) identify object classes; (2) identify associations; (3) identify attributes of objects and associations, and (4) organize and simplify classes using inheritance. The object model for this application is shown in Figure 1.

3.3 The HyperObject Model

To build the hyperobject model, we must analyze each class defined in the object model, checking if all desired access paths are part of the conceptual model. If the answer is no, then we have to add new associations and/or classes. In our application, if we want to know a specific book from an author, we should access his literary production. In many cases, this path will be very common: from an author, we find his production and, then we explore the books that we want. However, we should anticipate one possibility of access that makes a direct association between the author and a particular book. So, to represent this intended new path, an association between *Author* and *Book* should be added.

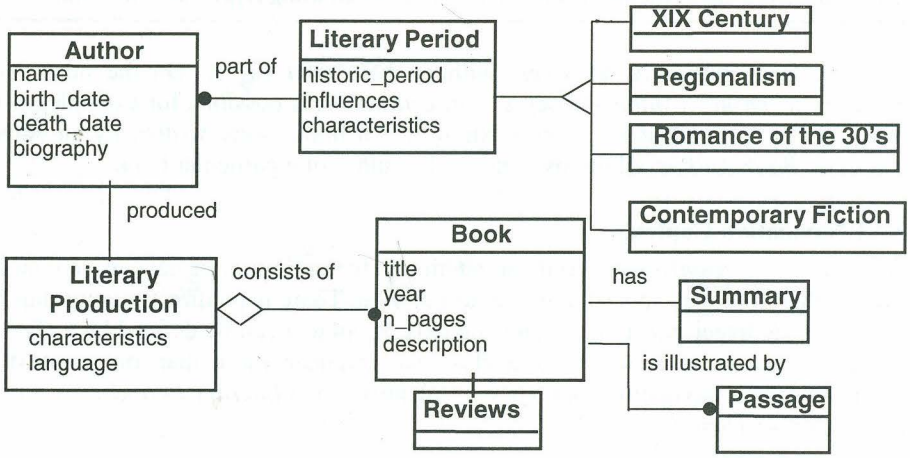

Figure 1 - The Object Model

Another aspect treated in this model is multimedia. Using the hyperobject model, the developer expresses his decisions about several media that take part in the final application. We should then analyze the classes that will have multimedia data. In our model, three classes will have multimedia representation: (i) an author can be seen by a portrait and, if possible, by a video passage; (ii) a book is presented also by its cover, and (iii) a passage should be presented in audio format, besides text. HMT represents these decisions using the aggregation concept, as shown in Figure 2, for the *Passage* class.

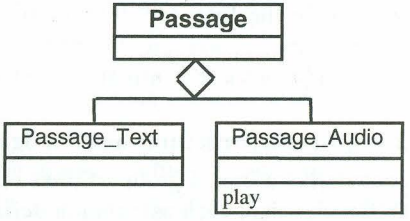

Figure 2 - Adding different media

3.4 The Navigation Model

In the navigation model, each association is analyzed according to its cardinality and direction. The cardinality (1:1, 1:N or N:M) is directly extracted from the hyperobject model. But the direction of the association - uni or bi-directional - is a project decision made in the navigation model. An association is uni-directional if it is only possible to traverse it in one way. An association is bi-directional if it can be traversed in both ways. The association "refers to" (*Author-Author*) is defined as uni-directional. This means that, from an author A, it is possible to access other authors A "refers to". However it is not

possible, from author A, to access authors that "refer to" A. On the other hand, the association "wrote" (*Author-Book*) is bi-directional. It is possible, for example, to traverse from *Author* to *Book*, if we want to know which books were written by an author. The converse, *Book-Author*, allows us to know the author of a particular book.

3.4.1 Navigation Contexts

One way to decrease (or avoid) disorientation is the definition of navigation contexts. We should, therefore, pay special attention to this step. There is no direct relationship between classes of the hyperobject model and contexts. A context can be defined by a class or by a group of classes. In some cases, a class can originate more than one context. In our application, four navigation contexts were identified: (1) *Literary Periods*, (2) *Authors*, (3) *Books* and (4) *Themes*.

After the definition of contexts, for each class of the model a default context should be assigned. For example, the default context of the *Author* class is *Authors*. However, the *Author* class can be associated with other contexts. For example, if we are consulting the period *Romance of the 30's* and we want to know the authors from that period, we must remain in the *Romance of the 30's* context, and the authors accessed are from that period.

A book has *Authors* as its default context, even though we could imagine it would be *Books*; in this case, a book is associated with its author, before it is a book per se. Other contexts associated with *Book* are *Literary Periods*, *Themes* and, of course, *Books*.

Using the concept of navigation contexts, we could define context-sensitive navigation and context-free navigation. In the first one, it takes into account the current context, so the user does not get lost. In the second one, the new context is the default context. It is used when there is no relation between the current context and the new one.

3.4.2 Types of Links, Access Structures and Navigation Diagram

Each association can derive a link, a group of links or an access structure (e.g., an index or a guided tour) [4]. Besides that, each association defines a type of navigation (context-free or context-sensitive). These decisions are made in the navigation model. Furthermore, we define the entry-points, or in other words, the initial access to the application. Finally, we draw a diagram called navigation diagram (Fig. 3). We use the symbols defined in [4].

4 RESULTS

The results we have obtained by using this technique are very impressive. Users feel very comfortable navigating in the hyper-space. There is no cognitive overhead, and the risk of disorientation is almost inexistent. The interface is completely based on the object, hyperobject and navigation models. Therefore, it is consistent, uniform and simple (Fig. 4).

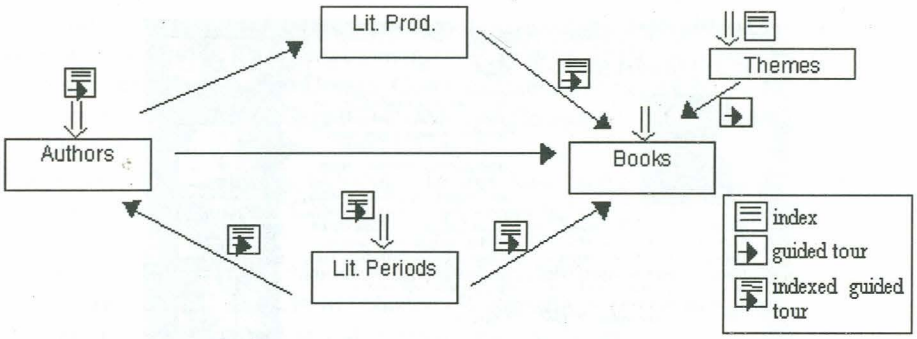

Figure 3 - Navigation diagram

5 CONCLUSIONS

A modeling and design technique - HMT (Hypermedia Modeling Technique) was presented. Through the development of a CD-ROM, it proved to be a powerful tool.

In the past few years, there were several proposals of techniques, models and methodologies for the development of hypermedia applications: HDM [2], HDM2 [3], RMM [4], EORM [5] and OOHDM [13]. HMT contributes with these proposals through these features:

- identification of four models for specifying a hypermedia application: object, hyperobject, navigation and interface;
- use of the object model of OMT to capture the application domain semantics;
- explicit treatment of the multimedia aspect in the hyperobject model, with specific object classes;
- specification of the navigation through semantics enrichment of the associations and the definition of navigation contexts (context-free and context-sensitive).

Our first experience using HMT, the development of a Literary Hypermedia Encyclopedia CD-ROM, was very successful. Our emphasis on research now deals with the interface model, which was not treated in this paper. Another aspect that should be investigated is the use of HMT in different domains, such as cooperative work. We are using the operations of the object model to specify the behavior dimension of a cooperative work tool.

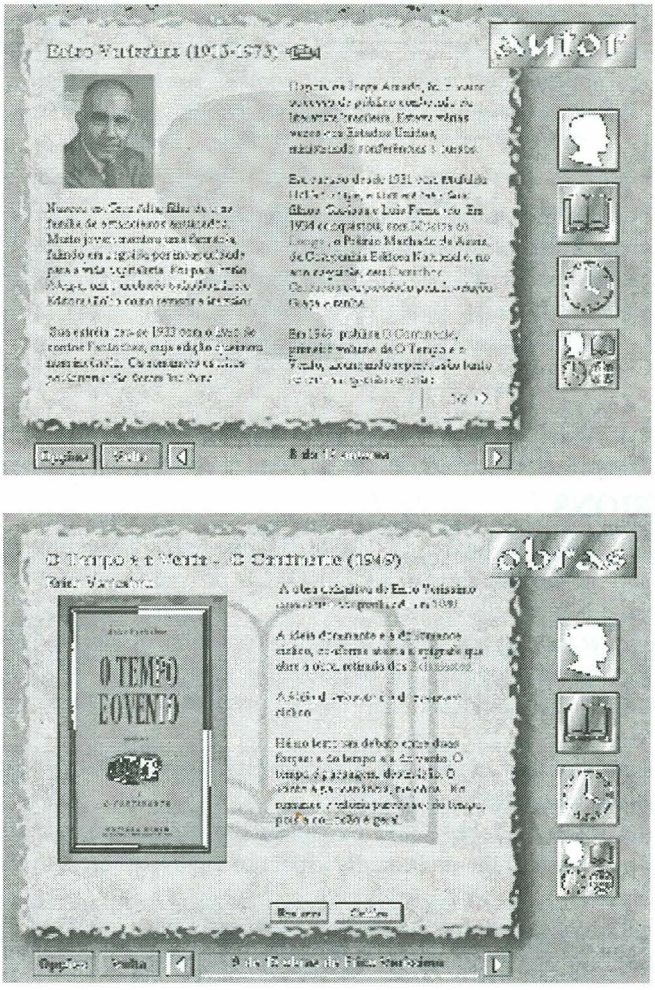

Figure 4 - Examples of an Author and of a Book

References

[1] Conklin, J. (1987). Hypertext: An Introduction and Survey. IEEE Computer, 20 (9), 17-41.

[2] Garzotto, F.; Paolini, P.; Schwabe, D. (1993). HDM - A Model Based Approach to Hypermedia Application Design. ACM Transactions on Office Information Systems, 11 (1), 1-26.

[3] Garzotto, F.; Mainetti, L.; Paolini, P. (1993). Navigation Patterns in Hypermedia Data Bases. International Conference on System Sciences, 1993, Maui, Hawaii. 257-269.
[4] Isakowitz, T.; Stohr, E.; Balasubramanian, P. (1995). RMM: A Methodology for Structured Hypermedia Design. Communications of the ACM, 38 (8), 34-44.
[6] Microsoft. (1993). Art Gallery CD-ROM: The Collection of the National Gallery, London. Microsoft Home, 1993.
[5] Lange, D. (1994). An Object-Oriented Design Method for Hypermedia Information Systems. International Conference on System Sciences, 1994, Maui, Hawaii. 1994. 366-375.
[7] Nemetz, F. (1995). HMT: Modelagem e Projeto de Aplicacoes Hipermidia. Porto Alegre: CPGCC da UFRGS (Master's Dissertation - in portuguese).
[8] Nielsen, J. (1990). Hypertext and Hypermedia. San Diego, CA: Academic Press.
[9] Parseye, K.; Chignell, M. (1993). Intelligent Database Tools & Applications: Hyperinformation Access, Data Quality, Visualization, Automatic Discovery. New York, NY:John Wiley & Sons.
[10] Rivlin, E.; Botafogo, R.; Shneiderman, B. (1994). Navigating in Hyperspace: Designing a Structure-based Toolbox. Communications of the ACM, 37 (2), 87-96.
[11] Rumbaugh, J.; et al. (1991). Object-Oriented Modeling and Design. Englewood Cliffs, NJ: Prentice Hall.
[12] Schwabe, D.; Barbosa, S. (1991). Navigation Modeling in Hypermedia Applications. Technical Report, PUC-RJ, Brazil, 1994.
[13] Schwabe, D.; Rossi, G. (1995). Building Hypermedia Applications as Navigational Views of Information Models. International Conference on System Sciences, 1995, Maui, Hawaii. 231-240.
[14] Woodhead, N. (1991). Hypertext and Hypermedia - Theory and Applications. England: Sigma Press.

Acknowledgements
This work was supported by CNPq and FAPERGS.

Este artigo descreve o software "Enciclopédia da Literatura Rio-Grandense", premiado com 2o. lugar de Melhor Pacote Multimídia, no I Concurso Nacional de Software Universitário, realizado em Minas Gerais, 1996. O artigo foi originalmente publicado no ED-MEDIA & ED-TELECOM 96 (World Conference on Educational Multimedia and Hypermedia & World Conference on Educational Telecomunications), Boston, MA., USA, junho de 1996.