FLEXIBLE GUIDED-TOURS IN HYPERTEXTS: A WAY OF CONTROLLING THE USER IN LEARNING APPLICATIONS

Maria das Graças V. Nunes
Renata Pontin M. Fortes

Dept. of Computer Science
University of São Paulo
{mdgvnune, renata}@icmsc.sc.usp.br

Maria do Carmo Nicoletti

Dept. of Computer Science
Federal University of São Carlos
carmo@dc.ufscar.br
Brazil

Abstract: This paper proposes and discusses a user control strategy in learning applications which is based on the use of flexible guided-tours. The flexibility arises from the use of nested context nodes combined with node attributes, which allow authors to create guided-tours which impose on learners suitable degrees of freedom aiming at learning. It describes an authoring and navigation environment, named SASHE, which implements these ideas.

Keywords: Hypertext Systems, Flexible Guided-tours, Learning Applications in Hypertext, Nested Context Nodes, Authoring Tool.

1. Introduction

Hypertext technology makes available a new way of organising information and provides a flexible approach for users to access it. This flexibility arises mainly from the navigation process which gives users several different options, allowing them to choose which way to follow when reading the hypertext. From the hypertext author's point of view, freedom and flexibility can also be achieved during the authoring process since in this process there are no strict rules which should be followed.

Authors can create a hyperdocument according to their own goals and according to a particular view they have about the information to be made available. Also when creating a hyperdocument, they can take into consideration the end users that will make use of it and their proper ways of dealing with the information.

This technology, however, can give rise to some problems both to users and authors. On one hand, by allowing freedom for organising a hyperdocument, this technology contemplates the creativity of the author and, on doing that, it promotes the opportunity for creating a richer hyperdocument. On the other hand, as a consequence of the same freedom, it becomes harder for the user to capture from a hyperdocument the author's underlying message. The message capturing process is especially important when the hyperdocument is part of a shared pool of documents which can be reused to compose new ones.

A well known side effect of the freedom when navigating a hyperdocument is the disorientation the user faces. This problem lead us to think that an automatic mechanism able to infer parameter values from a document structure could be effective in the recovery of the author's message. The parameters would be defined by the author and its possible range of values given according to a predefined strategy which would aim at a full comprehension of the author's message, by the user.

One possible attempt to overcome the user's disorientation problem is by using what is called a guided-tour. This kind of resource, also called path or script, is supposed to be created by the author and basically consists of a fixed sequence of hyperdocument nodes or pages to be followed during navigation.

It is true that many authoring environments do not distinguish guided-tours from any other type of node-based network. A guided-tour consists of a sequence of nodes selected from the hyperdocument, with implicit *next* and *previous* links. Guided-tours can be created aiming to impose different degrees of freedom during navigation. On one extreme, a guided-tour can be very restrictive, preventing the user to access anything else except the pre-established previous/next node; all the original links inside a node are inhibited. On the other extreme, it can keep each and every link created by the author, allowing the user a very high degree of freedom. It is obvious that a very "broad" guided-tour permits the user to be autonomous when it comes to where to go next and this is very risky: the author's message can easily be missed.

We believe that an ideal solution could be achieved by something in between these two extreme approaches. A guided-tour for learning applications should keep the original node links if they

- can promote a more flexible and consequently a more informative reading,
- can contribute to the achievement of the user's goals and
- do not misguide the user by allowing dead-end, unusual or weird paths.

By proceeding like that, it is possible to exercise a light and almost transparent control over the learners, driving them through the author's intended path and, at the same time, allowing them a certain degree of freedom, so they can still "explore", but within controlled frontiers.

This work focuses on the above problems and in order to solve them, it proposes to qualify the nodes of a hyperdocument and to organise them into virtual nodes named context nodes. Besides, it presents the design and implementation of an authoring and navigation environment which creates context-based guided tours for existing hyperdocuments, to be used in learning applications.

The paper is organised as follows. In Section 2 we discuss few problems related to hyperdocument authoring and point out the use of nested context nodes, by means of flexible guided-tours, combined with the use of node attributes, as a possible solution to these problems. Section 3 presents the description of SASHE, an environment for authoring and navigation which allows the creation and use of flexible guided-tours. In Section 4 the possibilities of generalising SASHE for other domains are considered and finally, in Section 5, we draw our conclusions and outline the next goals we intend to pursue.

2. Contextualizing and Qualifying Hyperdocuments Components

Due to authoring freedom, users generally are left with the responsibility of finding their own ways within a hyperdocument in order to achieve their goals. However, when authors have specific goals as well, authoring freedom can become a burden on them: how can authors ensure that their message has been captured by users, without imposing a very heavy control on the navigation process? In learning applications this burden seems even heavier because there is a very precise goal to be achieved, which strongly relies on ways the learner is guided through the available educational material.

If the authors' goal is to produce a guided-tour for an existing hyperdocument, it would be very convenient for them, to be able to count on tools which could help selecting and presenting relevant nodes, as well as organising them, in a very quick and efficient way.

Although current hypertext environments make available tools which support the authoring process such as graphical browsers and outline viewers, these tools are neither suitable for supporting the construction of context-based guided-tours nor to deal with qualified nodes. In this work we are concerned with domain independent tools, i.e., tools of general use which are especially designed for constructing guided-tours.

What is aimed at is to find ways of ensuring authors the possibility of creating a precise message, which would allow the inclusion of whatever seems essential and relevant to convey their goals, without been necessary to discard what can be considered secondary or complementary, provided there is no conflict involved. We believe that by providing ways of qualifying nodes and constructing guided-tours this can be achieved.

2.1 Using Nested Context Nodes

A logical grouping of nodes which are somehow related to the same topic is known as a context node (or composition node); this kind of node plays an important role in this work. The creation of context nodes can help organising the information and can make, as a consequence, its structure clearer to the author. Many hyperdocument models, including HDM [1], AHM [2], RMDM [3] and OOHDM [4] have the notion of context nodes. Some of them, however, have limitations (not handling nested context, for example). The Nested Context Model (NCM) [5] is a conceptual hypermedia model which handles context nodes and recursively nested context nodes; this model has, in some ways, subsidised some of the ideas discussed in this work which are implemented in SASHE.

Our proposal of flexible guided-tours is based on the notion of nested context nodes. By using this kind of node, the author can define degrees of freedom associated to terminal nodes (which can be text, sound, image, etc.) and by selecting a terminal node which is part of a nested context, the author has ways to find out which is the most convenient context to be allowed in a certain guided-tour. This means that even if a certain group of nodes has not been directly selected for composing a guided-tour, the fact that it belongs to a chosen context makes all its links and corresponding anchors accessible and visible during the reading of the selected terminal node. Intuitively speaking, by choosing which context will be part of a guided-tour, authors are allowing different degrees of controlled freedom to the user. We call degree of freedom of a guided-tour associated to a terminal node to the level of the context which contains the terminal node chosen by the author. The use of nested context nodes is one important issue of our proposal; the other is to qualify nodes by using node attributes.

2.2 Selecting Qualified Guided-tour Components

The existing hypertext models usually do not offer many options for qualifying nodes, with respect to their contents. For instance, the use of keywords is a very useful and common practice but it is not much of a help when a more refined selection is needed, as it happens when selecting nodes for composing a guided-tour.

By allowing the creation of qualitative attributes associated to nodes, we intend to be able to provide a way of categorising a node on the basis of the information related to its contents. An attribute can reflect some peculiarity of an application and can be used both for constructing guided-tours and as an additional support for easing navigation. If a tool is supposed to implement these features, it should first be independent of the application domain and second, it should allow the definition and the use of application-customised attributes.

If the nodes of a hyperdocument have been previously qualified according to some application-customised attributes, this information can be used as an additional (besides the link) control strategy. Based on the qualification of nodes, authors can define alternative ways of interacting with users. In learning applications, for instance, this approach can be used by authors in order to define different control strategies associated to different degrees of difficulties faced by learners. So, if a student finds a topic very hard, he/she can interact with the system by activating a function (e.g. by means of pushing a button) which would cause a certain implemented author's strategy to take over.

Since the tool we designed was meant to be a general authoring tool, it is assumed that the author, when using it, would have previously defined strategies according to parameters based on different node attribute values. It should be noted however that in order to make this possible, the navigation system only needs two information: the values of attributes which should guide the search and what it should be done in case of failure. Based on that, it is not difficult to make this process parametric, which is the first step towards making it general so to serve to any application. Of course the additional costs of adding this process to the navigation system and consequently to the authoring system are directly related to the benefits the whole system can provide the users.

The use of nested context and attributes for creating guided-tours and few other additional features for helping the user during navigation satisfy the main goals established in the previous section, since:

- the relevance and the adequacy of nodes which compose a guided-tour can be easily contemplated by means of establishing a limiting-context (degree of freedom) and by using attributes for carrying out the node selection process;
- users can have a more flexible, more informative and *softer* reading because there is always the possibility of enlarging the degree of freedom so to promote these aspects when interacting with the system and
- authors have alternative ways of composing their messages and learners have alternative ways of recovering them.

Next section presents a description of a tool named SASHE, which can be considered an environment for authoring and navigation which implements the main ideas discussed in this section.

3. Controlling the User in Learning Applications: the SASHE Environment for Authoring and Navigation

The use of hypermedia technology in learning applications usually implies free navigation through the hypertext containing the instructional material. There are several arguments which support the "free-navigation" style; nevertheless we are concerned here with the negative aspects of its use. The lack of system control asks for an extra effort from the learner, who must now decide what to do next, when to do it and where to go. The learner needs to play an active role in designing learning strategies [6].

As experiences have shown, not all students get benefits from this freedom [7]. The question which arises from this discussion is how to establish a balance between the learner and the system control. We strongly believe that by carefully constructing flexible guided-tours this balance can be achieved.

SASHE (a Portuguese acronym for System for Hypermedia Authoring and Support for Education) is a computational environment which supports hypermedia authoring focusing on learning applications. As shown in Figure 1, it has two main functional modules: the *Authoring Module*, which supports the authorship of both hyperdocuments and guided-tours, and the *Navigation Module*, which provides an user-friendly environment when used for educational purposes. The earlier version of the *Authoring Module* presented in [8] has been extended with the implementation of the ideas presented in this paper. An hyperdocument created by SASHE can offer additional functions, available through interface buttons, which can give users, among other things, orientation help related to *where* they are and *within which context*.

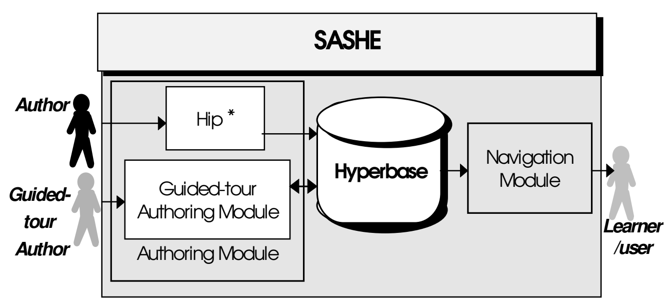

Figure 1 - General architecture of SASHE, showing its functional modules.

The current authorship environment (Hip*) is an extension of that presented in [8], and allows to qualify nodes by the attribution of node attribute values. These attributes can be used later by the author, for selecting an appropriate group of nodes to be part of a guided-tour.

SASHE allows two types of nodes: terminal nodes (such as video, text, sound, graphic) and compositional nodes (tracks, user context and public hyperbase).

While compositional nodes can be classified only with relation to the subject they are about, terminal nodes can be classified by their didactic function (motivation, introduction, definition, example, exercise, abstract), by degrees of difficulty (low, medium, high) and by keywords which identify their contents. If necessary, these attribute values can be reviewed and extended. The SASHE window for creating a terminal node requires values for its attributes *Name, File, Didactic Function, Difficulty and Keywords.*

For creating context nodes, *Name* is the only attribute value required. In a context node, the value of the attribute *Name* indicates the subject the node is about. The system assumes by default that all nodes which are part of a certain context are related to the same subject. Due to that, when organising a hyperbase, one should keep on mind that the inner contexts should always represent more specific subjects.

When creating a link object, besides defining its source and destination nodes, two other attribute values can be assigned to this object: *simultaneous presentation* and *presentation of terminal context nodes*. The first attribute indicates if the destination node should (or should not) be presented simultaneously to the presentation of the source node. The second indicates if during the presentation of a context node (the destination of the link) it should also be displayed all the terminal nodes which are part of that context node, or instead, if the *Graphical Browser* should take over. So, there are two possibilities for visualising a destination context node. The first shows all the terminal nodes which belong to that context node; the second (assumed by default) allows a graphical presentation of all the nodes shown as part of a graph.

From an existing hyperdocument and by using the same *Authorship Module*, authors can define one or more guided-tours, aiming at a certain group of users. Authors compose the graphical sequence of the guided-tour nodes, by selecting nodes and contexts from the hyperdocument, with the help of attribute-based searching functions. At this point, an author can define the degree of freedom associated to each node, by choosing the largest context which contains the referred node. By proceeding like that, the visibility associated to that node is determined and, as a consequence, it is established how far the user can go away from the original track, without compromising the relevance of the educational material.

Figure 2 - SASHE window for Student Interface

The SASHE interface for creating guided-tours is composed by three main modules, which are used for consulting, listing and visualising. The consulting module allows authors to specify node attribute values which will guide the node selection process; the listing module displays the selected nodes and allows authors to choose which will be part of the guided-tours and the visualisation module shows a graphical diagram of it.

By using the *Navigation Module* learners have ways to partially control their interaction with the system as shown in Figure 3. By making use of the "*it is easy*" (Está Fácil!) and "*it is difficult*" (Está Difícil!) buttons, the user triggers the associated learning strategy, which is programmed for increasing and decreasing, respectively, the level of difficulty. Information concerning the learning domain can be directly obtained by the learner. That is the case, for example, of *Bibliography* (Bibliografia) and *Exercises* (Exercícios). Functions which help users to guide themselves within the system are available through the "*where am I?*" (Onde Estou?), "*history*" (Histórico) and "*context-where am I?*" (Contexto) buttons. The "*where am I?*" button locates the current node in the guided-tour and graphically shows the position of the current node with relation to the original sequence of nodes programmed by the author. The "*more information*" (Mais Informações!) interface button triggers a search in the whole hyperdocument, for those nodes which have one or more keywords in common with the current node, and the "*help*" (Ajuda) button seeks, in the current context, for a node which has the same attribute containing questions and answers related to the current topic.

4. Generalizing SASHE for Other Domains

From the experience of developing SASHE and from the results obtained with the creation of public hyperbases for learning applications, we came to the conclusion that the qualification of nodes, by using attributes, is an issue which should be independent of the application under consideration.

By allowing the author to define a set of domain-dependent attributes as well as the values they can assume, SASHE can be generalized to be used with any other application domain. So context-based guided-tours can be created in applications others than learning. In order to do that, it is necessary first to define a formal language which would allow defining the new set of attributes and their values according to a predefined syntax.

5. Conclusion

Controlling the user in hypermedia applications is a largely discussed issue and opinions about this subject can be found elsewhere in literature [9], [10], [11], [12]. However, when learning applications are concerned, the same issue is highly controversial and there is no consensus on which should be the best approach to adopt. In spite of that, we strongly believe that the balanced control provided by flexible guided-tours is a very convenient and friendly way of guiding users through the learning process, being quite assertive, without being too restrictive.

We intend next to exhaustively test and validate the SASHE prototype described in this work, considering two different views: the author-user and the student-user views. We believe that by doing that, we will be able to gather data which would allow us to refine and adapt SASHE for other application domains, namely, the Software Engineering domain and then, to pursue its final generalised version which would suit any application domain.

Acknowledgements: This work was partially supported by CNPq, Procs. N. 680063/94-3, 680077/94-4, 680076/95-6 and Fapesp, Proc. N.1996/10119-8

References

[1] F. Garzotto; P. Paolini; D. Schwabe "HDM - A Model-Based Approach to Hypertext Application Design", ACM Transactions on Information Systems, Vol. 11, No. 1, January 1993, pp. 1-26.

[2] L. Hardman.; D. C. A. Bulterman; G. Van Rossum, "The Amsterdan Hypermedia Model: Adding Time and Context to the Dexter Model", Communications of the ACM, Vol. 37, No. 2, February 1994, pp. 50-62.

[4] T. Isakowitz; E. A. Stohr; P. Balasubramanian "RMM: A Methodology for Structured Hypermedia Design", Communications of the ACM, Vol. 38, No. 8, August 1995, pp. 34-44.

[4] D. Schwabe; G. Rossi; S. D. J. Barbosa "Systematic Hypermedia Applications Design with OOHDM", In Proceedings: Hypertext'96, Washington DC, USA, March 1996, ACM Press, 1996, pp. 116-128.

[5] M. A. Casanova; L. Tucherman; M. J. Lima; J. L. Rangel; N. R. Rodriguez; L. F. G. Soares, "The Nested Context Model for Hyperdocuments", In Proceedings: Third ACM Conference on Hypertext, Texas, December 1991, pp. 193-201.

[6] A. Large "Hypertext Instructional Programs and Learner Control: a Research Review", Education for Information, Vol. 14, No. 2, 1996, pp. 95-106.

[7] E. R. Steinberg "Cognition and Learner Control: a Literature Review", Journal of Computer-Based Instruction Vol. 16, No. 4, 1989, pp. 117-121.

[8] M. G. V. Nunes; F. M. C. Vieira; R. Hasegawa "The HIP/Windows Environment for the Authoring of Multimedia Hyperdocuments" (in Portuguese), In Proceedings: II Workshop em Sistemas Hipermidia, UFC, Fortaleza, Brazil, May 1996, pp. 130-138.

[9] E. T. Freitag; H. Sullivan "Matching Learner Preference to Amount of Instruction: An Alternative Form of Learner Control", Educational Technology Research and Development, Vol. 43, No. 2, 1995, pp. 5-14.

[10] R. D. Hannafin; H. J. Sullivan "Learner Control in Full and Lean CAI Programs", Educational Technology Research and Development, Vol. 43, No. 1, 1995, pp.19-30.

[11] A. Large "Hypertext Instructional Programs and Learner Control: a Research Review", Education for Information, Vol. 14, No. 2, 1996, pp. 95-106.

[12] E. C. Shin; D. L. Schallert; W. C. Savenye "Effects of Learner Control, Advisement and Prior Knowledge on Young Students'Learning in a Hypertext Environment", Educational Technology Research and Development, Vol. 42, No.1, 1994, pp. 33-46.