

HM-Card: A New Approach to Courseware Production

Vanessa Mayrhofer

Institute for Information Processing and Computer Supported New Media (IICM)
Graz University of TechnologyGraz, Austria
Email: vmayr@iicm.tu-graz.ac.at

Nick Scherbakov

Institute for Information Processing and Computer Supported New Media (IICM)
Graz University of TechnologyGraz, Austria
Email: nsherbak@iicm.tu-graz.ac.at

Keith Andrews

Institute for Information Processing and Computer Supported New Media (IICM)
Graz University of TechnologyGraz, Austria
Email: kandrews@iicm.tu-graz.ac.at

Abstract: In this paper we analyze current trends in courseware development, focusing on the fact that most authoring systems today require too much of an investment of effort to produce multimedia courseware. We offer a new authoring paradigm which makes production of courseware easier and more productive. We also describe a particular system, called HM-Card, which supports this paradigm. This system, as well as help files and a number of courseware examples are available via ftp: <ftp://iicm.tu-graz.ac.at/pub/hmcard15>

1. Introduction

It is clear that interactive multimedia courseware would be very useful in education [Vaughan 1994]. By interactive, we mean multimedia where students can alter the flow of execution with their own input and thus get individualized feedback. This would be far more instructive than a simple linear presentation, which would take the same course regardless of the student's level of ability. However, as useful as interactive multimedia would be, the goal of using it in the classroom has not been fulfilled to nearly the extent it could be. While non-interactive multimedia documents, such as a text with pictures, can easily be produced by anyone using the software tools currently available, instructors would have to make a significant investment of time to learn to make satisfactory interactive multimedia courseware. The immediate goal then, if we are ever to realize the ultimate goal of using interactive multimedia in education to its fullest extent, is to make it realistic for instructors to prepare useful interactive courseware even if they have no prior knowledge of programming [Goodman 1990].

What is needed is a new authoring paradigm, one which allows authoring to be done with minimal training, and which allows the flexible re-use of previously prepared materials to produce courseware which can be aimed at different groups with different levels of ability. In this context we offer one possible authoring paradigm which might solve the currently existing problems with the difficulties of authoring interactive multimedia courseware. This paradigm is called the HM-Data Model [Maurer et al. 1994b].

Normally, authoring packages require programming to provide a reasonable level of interactivity, thus making the production of multimedia courseware feasible only for professional programmers. The scripting paradigm (see, for example, [Sims 1994] and [Hall 1995]), is one in which a programming language is used to specify multimedia elements by filename. Similarly, in the well-known icon paradigm, the flow of execution is specified in a form very close to ordinary "if .. then" statements. In both cases, in-program editing of multimedia elements and on-line modifications of courseware are rather restricted.

Basically, the HM-Data Model provides a new paradigm to specify interactions: it is a non-programming approach, which makes the creation of interactive multimedia courseware easy and productive for educators who don't happen to be skilled in programming [Maurer and Scherbakov 1996].

The model operates on two kinds of entities: multimedia pages and computer-navigable S-collections. Multimedia pages are the ultimate basis upon which courseware modules are built. The HM-Card editor provides a way to quickly produce customized pages using a wide range of types of multimedia object. Text can either be entered directly or can be imported from an external application. Images in different formats can be imported using different techniques. Drawings using lines, curves and simple shapes can be made with its vector graphics facilities. A number of objects on an existing page can be saved as a group and then reused in other pages with the click of a button. A beep or a flash can be incorporated into a page as a warning or to attract the user's attention. Any visible object or group of objects can be animated. The page editing in HM-Card follows the WYSIWYG metaphor. Of course such multimedia pages are rather conventional objects which can be found in many different authoring environments.

A really innovative feature of the HM-Data Model is that courseware content and structure are treated separately [Maurer et al. 1995] meaning that previously prepared courseware fragments can be re-used in newly created courseware modules. Simply speaking, human-machine interaction can be seen as a metaphorization of all actions available to a user in a particular current state of courseware execution and the options for changing to another state when an action has been carried out. This situation can be modeled as a large number of multimedia nodes which represent states of the courseware, interrelated by means of computer-navigable links representing possible changes of state [Berk and Devlin 1991]. So-called anchors visualize links in a particular form suitable for human-machine communication (for example push-buttons, check boxes, "hot" words, clickable areas, input fields and so on). The remainder of this paper is dedicated to describing the HM authoring paradigm.

2. The HM-Data Model

With the HM-Data Model, information is organized into units called structured collections, S-collections or simply collections. A collection can be seen as a closed environment which contains a structure of navigable links between the nodes (which will henceforth be called members) contained in it. There are several predefined types of collection. An envelope has links back and forth between every combination of two members in the collection. A folder joins members into a list, with links to next and previous members. In a menu, one specific member (called the collection head) has links to and from each of the other members. Lastly, with freelinks, users can link the collection's members in any manner they wish.

Not only do menus have collection heads, but so does every other type of collection. The collection head is a member designated to be seen first when the user starts browsing a collection. This, as we shall see, is important with respect to the navigational mechanism in this paradigm. Also, a collection can have associated with it a label, a page which describes the content of that collection. These collections, which can be seen as separate containers in which link following can only be done between members of that container, can also themselves be members of other collections and can have either other collections, simple pages or both as members.

The nesting of collections can be arbitrarily deep. Recursive membership is possible, i.e. a collection can have itself as a member or contain a collection which has it as a member. Also, each collection is required to have a unique name in a given HM-Card data base.

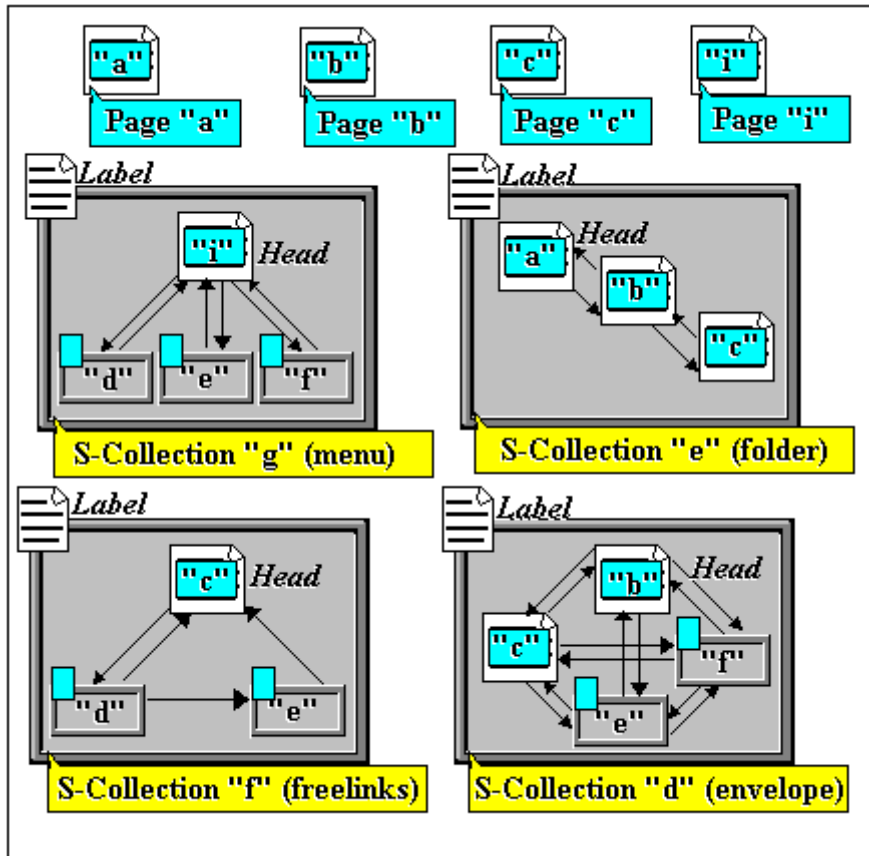


Figure 1: Internal Structure of S-collections

Now that we have discussed the basics about structure in the HM- Data Model, it is time to talk about its navigational operations. There are three important operations, which are called OPEN, CLOSE and ACCESS. The OPEN operation lets users do exactly that- they open a collection with it. As mentioned earlier, each collection has a certain internal structure. When one is opened, the environment the user is in is switched from one collection to another. The internal structure of the opened collection is there for the user to browse, so until another collection is opened, the user can only navigate within this collection, always starting with the head of that collection. The CLOSE operation is the opposite of OPEN. It causes the user to leave the browsing environment of one collection and go back to the context of the collection in which it is contained. The third operation, ACCESS, is what the user uses to navigate between the members within a collection. It is similar to following a hyperlink, except that when users access a member of a collection, that member could be a simple page or it could be another collection, and in this case, what they see first is its label, which gives them information about what is inside so they can decide whether to open it and explore its internal structure or access the next member. An important advantage of this data model when used for the preparation of educational software is that it allows flexible re-use of material. To be more specific, each collection which is prepared, whether it is simple or contains many levels of nested collections, can be used in as many other collections as the courseware author wishes. This powerful feature means that an author can treat collections which have already been created as modules which can be inserted as appropriate into new presentations. This can be done easily, and as often as the author likes. There is no limit on how often a collection can be re-used, nor is there any need for the author to know anything about the underlying mechanism that makes this possible. Programming approaches to interactive multimedia authoring cannot offer this kind of flexibility.

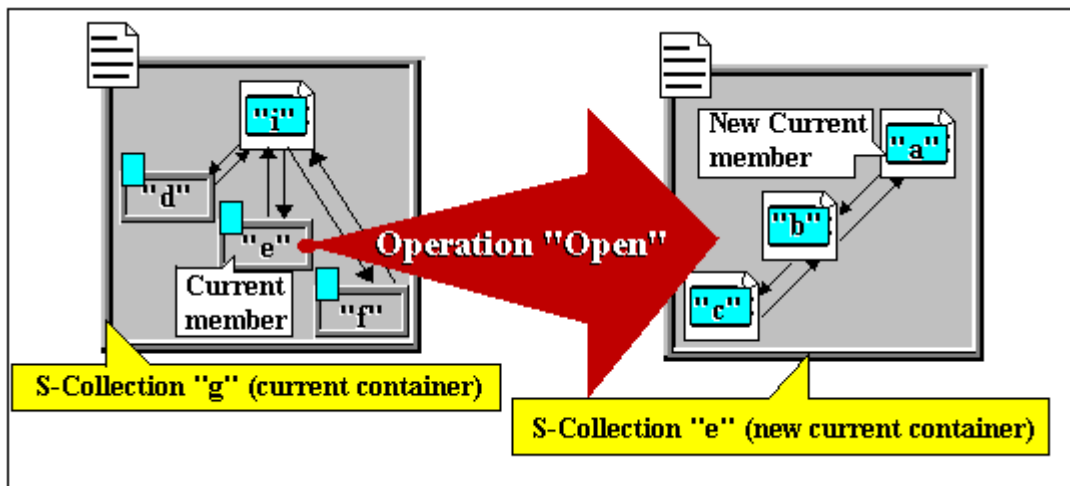


Figure 2: Operation "OPEN S-COLLECTION"

3. Creating Collections with HM-Card

This is to give an idea of how the HM authoring paradigm contrasts with forms of authoring which require programming (see also [Maurer and Scherbakov 1996] and [Maurer et al. 1995]).

Basic steps in creating a collection:

1. The author clicks on the "new collection" button
2. The author specifies the type of collection (menufreelinks, etc.)
3. The author chooses a collection from the set of existing collections to be inserted as the head of this collection. This is done by looking through a list of the unique names of these collections and clicking on them.
4. The author chooses a page as label for this collection.
5. The author inserts new members, either pages or collections, into the new collection with the INSERT_MEMBER operation. The members are linked automatically according to the chosen structure (unless its type is freelinks, in which case the author has to connect the members with the INSERT_LINK and REMOVE_LINK operations). As can be seen from these steps, this method is not difficult to learn and is at a much higher level of abstraction than programming methods. Thus the courseware author is not distracted by implementation details and can concentrate on the goal of making interactive multimedia courseware.

4. Different Views for Different Groups

Now that the basics of the structure and navigational operations have been explained, another advantage of the HM-Data Model can be discussed, namely the ability to use it to easily produce interactive presentations aimed at students of different levels using, at least partially, the same material. The important point here is that links in an HM-Card data base are context dependent. Links are never embedded in content, but exist in the context of a collection. Links are not global to an entire data base, but are followable according to which collection the user is in at a specific time. When the user is in a certain collection and he accesses (follows a link to) a certain member, it doesn't matter how that particular member is linked to other members of other collections, the only links that are visible to the user when he is navigating in a collection are the links which specifically belong to that collection. Thus different collections can be used to present the same information in different ways for students who have different levels of ability or different interests. This and the fact that a collection can be used as often as desired make it possible to make new presentations with existing material with minimal investment of time. This concept may be slightly difficult to imagine at first because in most multimedia paradigms, the links between objects are global, that is, they can be seen at all times. Thus the attempt to use the same objects to produce two different views of the same material with these paradigms would be pointless, since the links present in those objects would ultimately lead users to objects they were not originally intended to see. With the HM- Data Model, the user has no chance of accidentally following a stray link and of course cannot "get lost in hyperspace."

5. An Example

Here is a simple example to help make clear how an HM-Card data base can be structured, and how this structure can be advantageous when preparing courseware. Say, for example, an instructor would like to prepare two different collections about birds (see Fig. 3). For one group, she would like to give a general overview of birds and for the other, she would like to present a more specialized view, focused on flightless birds. She prepares several collections, with the names ROBIN, PENGUIN, CARDINAL, OSTRICH and KIWI. These collections may contain information about the area of the world where this particular bird lives, the bird's appearance, what it eats, etc. It doesn't matter what the content or internal structure is like, once a collection about a particular type of bird is prepared, it can be seen as an individual unit of information which can be re-used in one or more other collections.

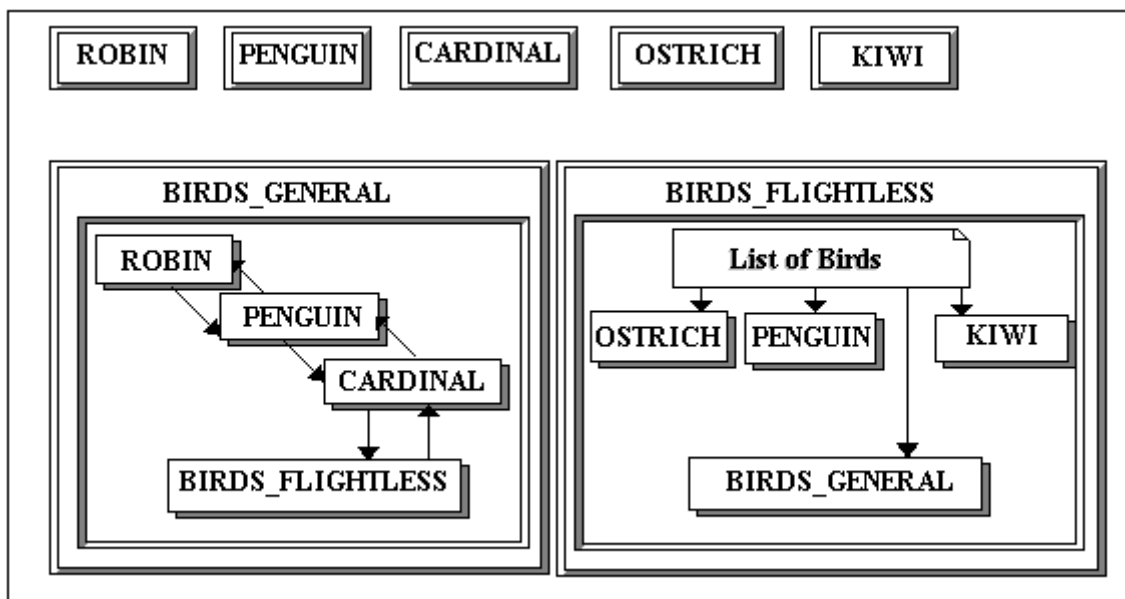


Figure 3: A sample HM-Card Database

The instructor can now use these collections in her courseware for the two different groups. She creates two collections, named BIRDS_GENERAL, of type folder, and BIRDS_FLIGHTLESS, of type menu. In BIRDS_GENERAL, she chooses the collection ROBIN to be the collection head and then inserts PENGUIN and CARDINAL. In BIRDS_FLIGHTLESS, she prepares a list of birds to be the head and inserts OSTRICH, PENGUIN and KIWI. Now she has created two different collections which re-use some of the same material. However, she also wants to make it possible for the group specializing in flightless birds to have an opportunity to go back and review some facts about birds in general. Thus she simply inserts BIRDS_GENERAL into BIRDS_FLIGHTLESS. Also, she would like the other group to be able to move on to more specialized information about flightless birds once they have some general knowledge about birds and thus she inserts BIRDS_FLIGHTLESS into BIRDS_GENERAL. Now the groups have two different views of the same material using two different styles of interaction. Members of one group can only turn pages (folder) and members of the other can look at a list and freely choose which type of bird they would like to learn more about. As can be seen from this example, the HM-Card authoring paradigm allows the re-use of collections in the context of different collections which are aimed at groups with different interests. It also allows recursive membership, as in this case where BIRDS_GENERAL contains BIRDS_FLIGHTLESS, which again contains BIRDS_GENERAL, thus easily handling the instructor's wish to give each group a particular lesson, as well as to permit them to eventually refer to further information about the same topic.

6. Conclusion

As can be seen from its description, the HM authoring paradigm is simple to learn and at a far higher level of abstraction than programming-based authoring paradigms. It allows flexible re-use of previously created

materials in new presentations, thus allowing courseware authors to produce useful interactive multimedia with a minimal investment of time. The fact that links are local to a collection rather than global allows the same material to be seen in different contexts. Because HM-Card is so easy to use and so powerful, we conclude that it has the potential to remove the existing obstacles to the effective use of interactive multimedia courseware in the classroom.

7. References

- [Berk and Devlin 1991] Berk E., Devlin J.(1991) Hypertext/Hypermedia Handbook. McGraw-Hill, New York.
- [Goodman 1990] Goodman D. (1990) The complete HyperCard 2.0 Handbook. Bantam Computer Books, New York.
- [Hall 1995] Hall T. L. (1995) Utilizing MultimediaToolbook 3.0 Boyd Fraser Publishing, New-York.
- [Maurer and Scherbakov 1996] Maurer H. and Scherbakov N. (1996) Multimedia Authoring for Presentation and Education: The Official Guide to HM-Card.Addisson-WesleyPubl.Co. Bonn.
- [Maurer et al. 1994a] Maurer H., Philpott A. and Scherbakov N. (1994) Hypermedia systems without links. Journal of MCA 17 (4), 321-332.
- [Maurer et al. 1994b] H. Maurer, N. Scherbakov, K. Andrews (1994) Object-Oriented Modeling of Hyperstructure: Overcoming the static link deficiency. Information and CoursewareTechnology 36 (6), 315 - 322.
- [Maurer et al. 1995] Maurer H., Scherbakov N. and Schneider A. (1995) HM-Card - a new Hypermedia Authoring system. Journal for Multimedia Tools and Applications, 1 (3), 305-326.
- [Sims 1994] Sims R. (1994) Authorware with Style. Bantam Computer Books, New York.
- [Vaughan 1994] Vaughan T. (1994) Multimedia--Making it Work. Osborne Publishing (McGraw Hill), New York.