Links in Hypermedia:
the Requirement for Context

Lynda Hardman, Dick C.A. Bulterman and Guido van Rossum
CWI
P.O. Box 4079
1009 AB Amsterdam
The Netherlands
lynda@cwi.nl,
dcab@cwi.nl,
guido@cwi.nl

ABSTRACT

Taking the concept of a *link* from hypertext and adding to it the rich collection of information formats found in multimedia systems provides an extension to hypertext that is often called *hypermedia*. Unfortunately, the implicit assumptions under which hypertext links work do not extend well to time-based presentations that consist of a number of simultaneously active media items. It is not obvious where links should lead and there are no standard rules that indicate what should happen to other parts of the presentation that are active.

This paper addresses the problems associated with links in hypermedia. In order to provide a solution, we introduce the notion of *context* for the source and the destination of a link. A context makes explicit which part of a presentation is affected when a link is followed from an anchor in the presentation. Given explicit source and destination contexts for a link, an author is able to state the desired presentation characteristics for following a link, including whether the presentation currently playing should continue playing or be replaced.

We first give an intuitive description of contexts for links, then present a structure-based approach. We go on to describe the implementation of contexts in our hypermedia authoring system CMIFed.

KEYWORDS

Hypermedia links, context for links, structured multimedia, hypermedia presentation

FOLLOWING LINKS: AN INTRODUCTION TO CONTEXTS

What does it mean to follow a link in a hypermedia presentation? The combination of dynamic and static media into one presentation increases the choice of what can happen on following a link. In order to understand the options that are available, it is useful to differentiate the behaviour of links in the three cases: hypertext, multimedia and hypermedia (shown schematically in Fig. 1— note that for simplicity composite nodes have been omitted from the figure). We discuss the three models briefly, looking at their notions of links and what happens on following a link.

In conventional hypertext (Fig. 1(a)), the model of a link is clear [Hala90]: when a link is followed, one 'leaves' the information in the source node and 'goes to' the information in the destination node. The presentation details of what it means to 'leave' a node vary from system to system, but the two most common options are that the information in the source node is replaced (e.g. HyperCard 1.0 [Appl87]) or a new window for the destination information is created (e.g. Intermedia [Pala90]). Most hypertext systems, however, lack a way of choosing between these options—the choice is determined by the destination of the link.

Multimedia systems allow the combination of multiple items of static and dynamic media into a presentation. Many of these systems also support some notion of a link, for example [Macr90], where, when the link is followed, the presentation leaves the point in time that the presentation is playing (that is, the screen is cleared) and a new point in the presentation is activated (Fig. 1(b)). The key is that there is no choice other than clearing the complete window then filling it with new information.

Hypermedia systems can be described as having the discrete, node/anchor/link properties of hypertext with the choice of media types of multimedia. In most current hypermedia systems the addition of dynamic media occurs only at “leaf” nodes of the hypertext (i.e. there are no links leading from the audio or video), and there is no composition of the different media. There are some exceptions, such as [Ross93], [Ogaw90] and [Buch92]. We are concerned here with the more complex case where hypermedia presentations consist of both static and dynamic media items which are grouped together as sets of composite entities (Fig. 1(c)). What should then happen to these components when a link is followed? Should they remain active, should they be frozen, should they be replaced? And what happens when the destination node consists of a collection of components—are all of them referenced by the link, or should a subset be activated? The richness of hypermedia presentations, in which a wide variety of information types can be simultaneously active, means that there is no single ‘right’ answer to these questions. As a result, no single model of link behavior should be defined and implemented. A more flexible approach is to define a set of rules that allow the author of a hypermedia presentation to specify the behavior that is required for a particular presentation.

In this paper, we present a concept for modelling the scope of information that is affected on following a link. We call this concept a *context*. The following section discusses the problems solved by using explicit contexts and gives examples of the use of contexts in existing hypertext systems. We then describe an example hypermedia application and use this to illustrate the choices available to hypermedia presentation authors. We go on to describe the detailed theoretical foundation for the context model, then finish by giving details of our implementation for contexts as part of our existing hypermedia authoring system.

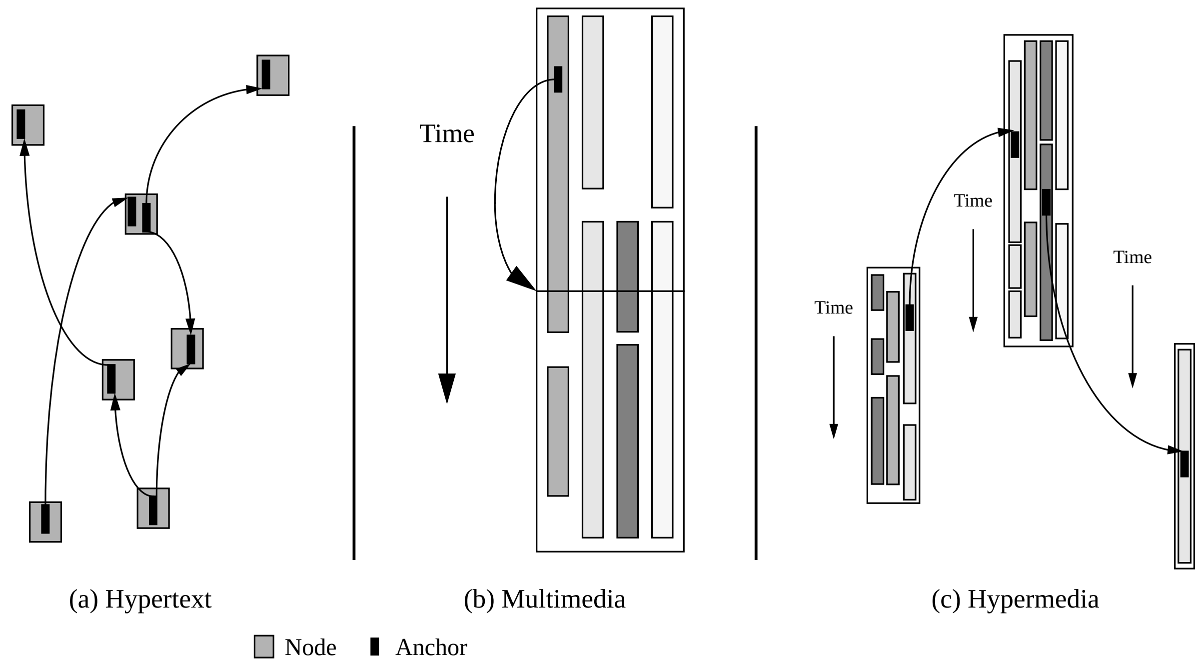

Figure 1. Hypertext, multimedia and hypermedia

DISCUSSION OF RELATED WORK

A number of different issues are relevant to deciding what the source and destination of a link in a complex hypermedia presentation should be. Different groups have come up against some of these issues in the construction of their own systems. Without going into our notion of context in detail, we give some examples of related work to give an impression of which problems we are aiming to solve in making the source and destination contexts of a link explicit.

[Pala90] describes the problem of what might happen when the reader follows a link where its destination demands the "simultaneous playing of active anchors"— in our words, when the destination is a complex hypermedia presentation. They give an example of wanting to coordinate the playing of a series of paintings of Napoleon's retreat with the playing of Tchaikovsky's *1812 Overture*. Their proposed solution is to have multiple anchors described at the destination of the link. We feel that this is insufficient, in that the anchors specify only parts of nodes, and it is the nodes themselves which need to have timing, and preferably structural, relations defined among them. The reader will also want to know which the relevant anchors are at the destination. Our solution is to specify a context along with the anchor(s) at the destination of the link, where this destination context is a structured hypermedia presentation.

In the Videobook system, [Ogaw90], the destination of a link is a complete scene, and thus the timing constraints among the items are all specified. They do not, however, allow the specification of anchors at the destination of the link. This corresponds roughly to the picture shown in Fig 1(b). The source of the link is an anchor ("trigger" in their terminology), and the complete scene is cleared when the link is followed. There is no discussion of whether the scene at the destination of the link replaces the scene at the source of the link or not. By making the source and destination contexts explicit, along with the relation between the two, we give the author more control over the specification of the presentation.

Other structured approaches to hypermedia, for example [Fuji91] and [Buch92], concentrate on the relations between objects at a fine-grained level. These state structural and timing relations within one hypermedia presentation explicitly, but do not explore the options when following links between presentations. We address the issues of which part of the currently playing presentation is affected on following a link by specifying a source context, and what can happen to the source context on following the link.

Examples of context have been implemented in a number of hypertext systems, although they are not referred to as such. HyperCard [Appl87], for example, has the two levels of foreground card and background to a card. On following a link the foreground card changes, but the background may remain or may also change. Although this option exists, the author has no explicit control over this — it depends entirely on the destination card of the link.

The Guide system has implemented a form of context via the use of a "source region" [Brow92]. One link type (the "replace-button" in Guide terminology) takes the reader deeper into the hierarchical structure and the author can specify, via the use of a source region, how much of the surrounding material should be replaced. In our terminology this Guide example defines the source context which is replaced by the destination context when the link is followed. The display option for the destination of a link can also be specified, for example by using the "definition" structure type. This allows the destination of a link to be added to other definitions in a window separate from the source of the link. A further display option for the destination of a link is the "note button" (p 91 in [Niel90]), where a pop-up window appears next to the source of the link.

In the case of hypermedia, where presentations are built out of collections of items and timing plays a fundamental role, it becomes more important to state the source and destination contexts for a link explicitly, along with the relation between the two, and what happens to the source context on following the link.

Figure 2. A hypermedia presentation

The main contents screen of a walking tour of Amsterdam, comprising three text nodes and a picture. Two of the text nodes are linked to other arts of the presentation.

TOURING AMSTERDAM: AN EXAMPLE HYPERMEDIA PRESENTATION

As a starting point for our discussion in this section, Fig. 2 shows an example hypermedia presentation, in this case a tour of the city of Amsterdam. This presentation consists of a number of media items displayed on screen or played through loudspeakers. Some of the media items have anchors with which the reader can interact. The reader can watch the presentation passively, or can jump to another part of the presentation by selecting an active anchor. In the latter case, a new part of the presentation will start up.

A particular presentation may need to support several types of links, each providing a different type of behavior with respect to the active media in the presentation. We illustrate the three possibilities for following links in Figs. 3 to 5.

In Fig. 3, a reader selects the anchor labelled CWI; doing so provides a (short) audio description of our institute. In this figure, we see that the musicians scene can continue playing along with the spoken commentary about CWI.

Fig. 4 shows a second option. Here, the musicians scene pauses during the audio commentary. After the commentary has finished the musicians scene continues from where it was before the link was followed.

The third option is shown in Fig. 5. Here, on following the link from the *Contents* button the musicians scene is removed completely and replaced by the contents screen for the Amsterdam tour. (This is traditionally the most common approach.)

These three examples show relatively straightforward cases of the behavior of links. Unfortunately, by simply looking at the media components that are active at any one time, it is impossible for an author to identify relationships among the items in a presentation. This is the motivation for the definition of source and destination contexts. These are developed in the following section.

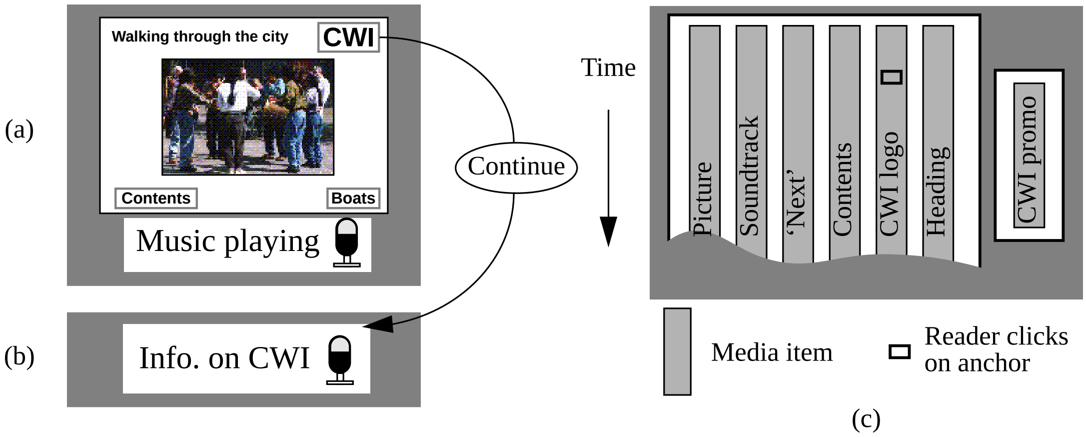

Figure 3. Continuing the source context

A scene with a video of musicians and accompanying music is playing in (a). The reader clicks on the *CWI* button, and the scene continues playing while a spoken commentary (b) is delivered. The same interaction is seen from a time-based point of view in (c)

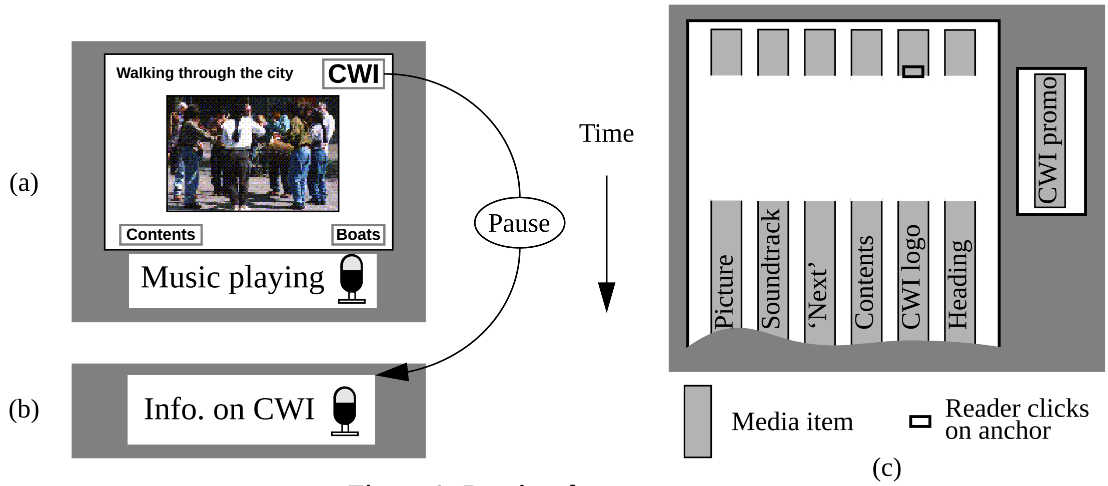

Figure 4. Pausing the source context

A scene with a video of musicians and accompanying music is playing in (a). The reader clicks on the *CWI* button, and the scene pauses while a spoken commentary (b) is delivered. The same interaction is seen from a time-based point of view in (c)

CONTEXTS IN HYPERMEDIA — THEORY

The *source context* for a link is that part of a hypermedia presentation affected by following a link, and the *destination context* is that part of the presentation which is played on arriving at the destination of the link. Source and destination contexts for a link are defined as part of the Amsterdam Hypermedia Model [Hard93a]. This model is an extension of the Dexter hypertext reference model [Hala90] developed to integrate time at a fundamental level in the model, thus leading to the definition of contexts for a link. The Amsterdam Hypermedia Model specifies two forms of composition—parallel and choice. Parallel composition indicates that the children of the node should be played in parallel (constrained by the addition of timing specifications for playing sequences). Choice composition indicates

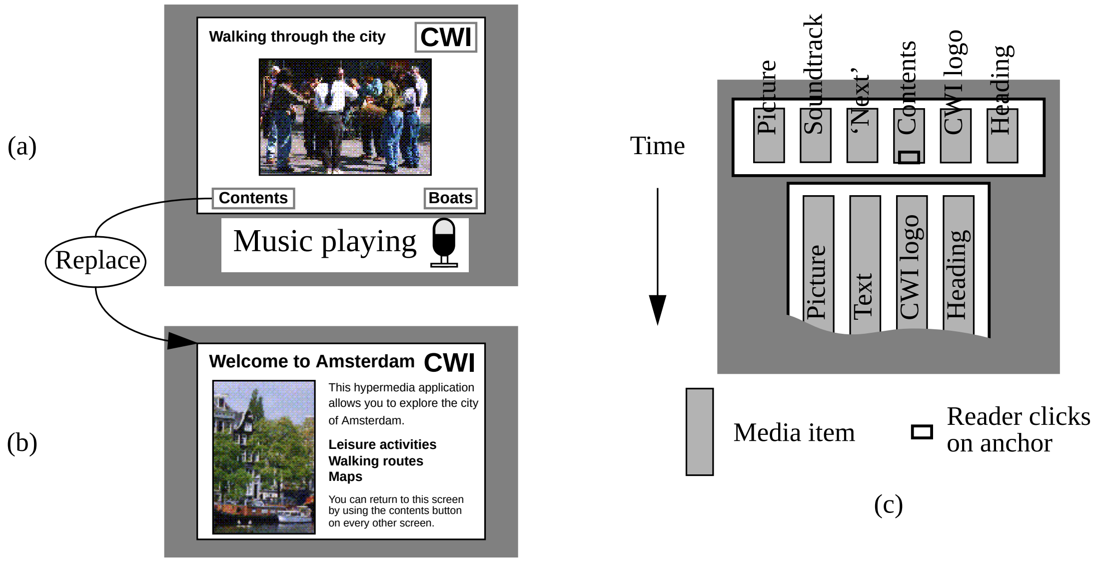

Figure 5. Replacing the source context

A scene with a video of musicians and accompanying music is playing in (a). The reader clicks on the *Contents* button and the scene is replaced by the contents screen of the Amsterdam tour, (b). The same action is seen from a time-based point of view in (c) that at most one of the node's children is played at any time. These forms of composition allow the creation of presentations which retain nodes at higher levels in the structure on the screen (illustrated in Figs. 7, 8 and 9).

In order to support contexts, the link structure in the Dexter model must be extended to include the associated context for the ends of the link, and to define what happens to the source context when the link is followed. The contexts for a link are specified by associating each source or destination of the link with an anchor and a composite component. The composite component defines the context for the link-end. The anchor is a reference to one or more anchors in descendants of the composite. Eventually the de-referenced anchor will reference a part of an atomic component (data node) at a leaf node of structure. These are the areas which can be used on-screen for denoting the end of the link (in the cases of text, graphics or video). Allowing an anchor in a composite component to be de-referenced to multiple leaf-node anchors means a link can be followed from a number of places.

An alternative way for storing context in the model is that the link itself has multiple "data" anchors, and one reference to the composite component which is the context. Neither storage model implies a user interface for specifying the contexts. For example, at some point during the creation of a link between anchors the author could select a composite higher in the structure for defining the context (at one end of the link).

Once we have established the context associated with the ends of a link, we need to define the display options associated with following the link. The choice is whether the destination context is displayed next to or instead of the source context, or in a new window independent of the source context. This display option for the source context also needs to be stored with the link. If the source context is replaced then the destination context will be displayed where the source context had been presented (this will require checking mechanisms in the authoring system to ensure that the destination context can be played within the resources freed up by the source context). If the source context is retained then the destination context requires extra resources (most likely an additional window) in order to be displayed, and may be displayed next to the source context or independently of it. The source context may continue playing, or it may pause.

The options for displaying the destination context in hypermedia are slightly more limited than those for hypertext. Since text is essentially one-dimensional, the destination context can be inserted in-line with little disruption in the text flow (this is used in Guide's "replace-buttons" [Brow92]). A hypermedia presentation, with its use of two screen dimensions plus time, can replace the source context with the destination context, or create a new window for the destination context, whereas trying to insert the destination context next to the source context is unlikely to be successful.

Figure 6 shows a structure-based view of two hypermedia presentations, where the source and destination contexts are in separate presentations. Here the choices are whether to replace the source context (as in Fig. 5) or to retain it. When the source context is retained it can continue playing (as in Fig. 3) or it can pause until the destination context has finished playing (Fig. 4).

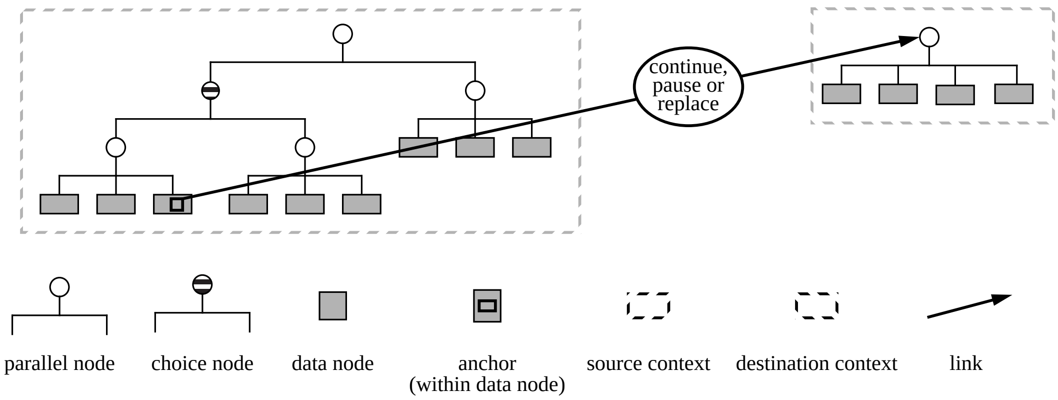

Figure 6. Structure-based view of source and destination contexts in different presentations

The destination context is in a separate presentation from the source context. The source context can be replaced by the destination context, e.g. Fig. 5, or retained and continue playing, Fig. 3, or paused, Fig. 4.

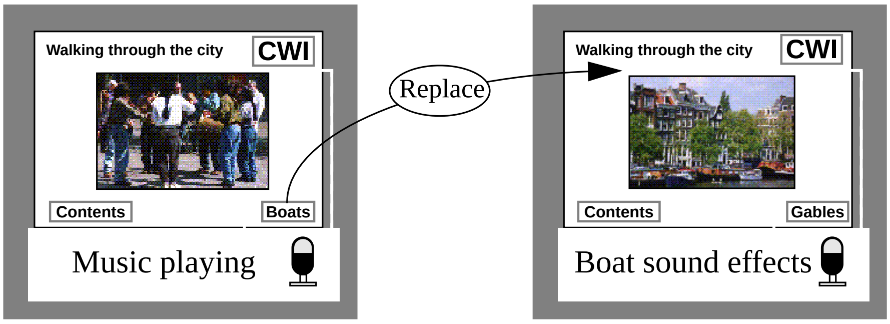

Figure 7. Screen-based view of source and destination context within same presentation

The *Boat*s button takes the reader to the following scene along the walking route. The nodes belonging to the source and destination contexts are enclosed by the dotted lines.

The source and destination contexts need not be in separate presentations. A hypermedia presentation may be built out of different levels of structure, where the higher levels of the structure remain on screen while the lower levels change — for example, when the reader follows links around the structure. Figure 7 shows what happens when the reader follows the *Boats* link from the presentation in 3(a). The higher levels of the presentation, the *Contents*, *CWI logo* and *Heading* nodes remain on the screen, while the nodes specific to the musicians are replaced by those for the boats. The time-based view for this is shown in Fig. 8. This behaviour is supported through the use of a *choice node*, where instead of defining all the nodes to be played in a presentation a number of different structures are given, from which one is chosen. The structure corresponding to Figs. 7 and 8 is shown in Fig. 9. In this situation, where the source and destination contexts for a link are within one choice node, there is no option on following the link—the destination context replaces the source context.

CONTEXTS IN HYPERMEDIA — PRACTICE

The previous section describes our theoretical model of contexts and in this section we describe their implementation as part of our hypermedia authoring system, CMIFed. A hypermedia presentation in CMIFed has a hierarchical structure whose leaf nodes are atomic components which are played in the presentation, and whose non-leaf nodes are composite components containing a collection of other composite components and/or atomic components. The hierarchial structure gives a choice of playing the child nodes (composite or atomic) in parallel or sequentially. This then

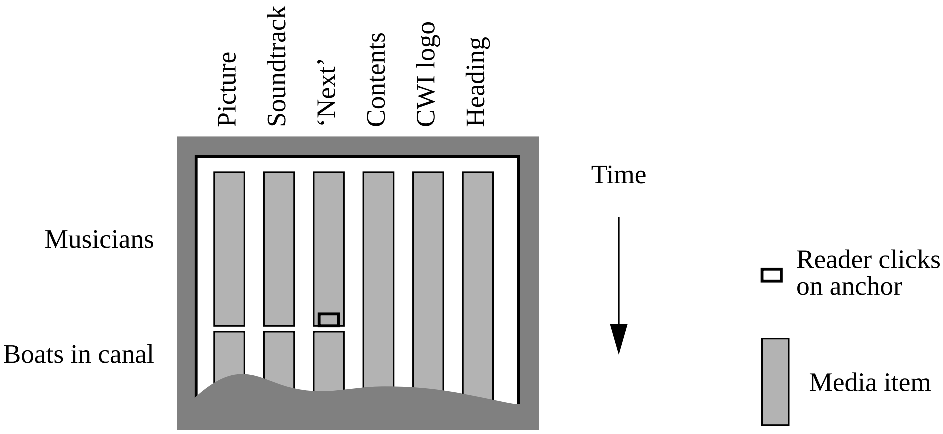

Figure 8. Time-based view of source and destination context within same presentation

This time based view corresponds to the screen-based view in Fig. 7. The *Contents* button, *CWI logo* and *Heading* remain on the screen throughout. The *'Next'* buttons take the reader to the following scene along the route, in this case from the musicians scene to the boats scene.

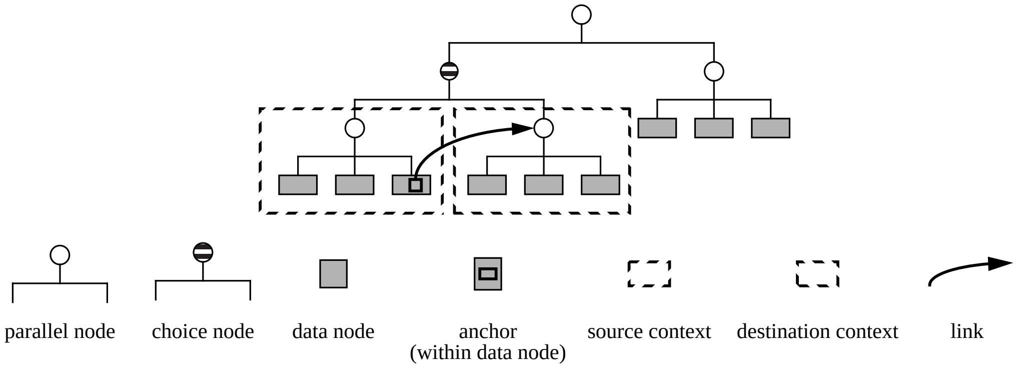

Figure 9. Structure-based view of source and destination context within same presentation

This structure based view corresponds to the screen-based view in Fig. 7. The source and destination contexts are alternative choices for the same parent node, so that the source context cannot be retained.

---

defines coarse timing information for the presentation. Each atomic component is assigned to a channel, a logical output device which is mapped by the player at runtime to a physical output device—e.g. an area on the screen or a loudspeaker channel. See [Ross93] for a description of the full implementation of CMIFed.

The description of contexts above states the specification and the use of contexts in the most general way possible, from which we have implemented the core ideas within the bounds of the currently existing implementation of our authoring system. The model prescribes the definition of source and destination context for a link. We currently derive the source and destination context from the structure around the ends of the link. The context is taken to be the child of the nearest ancestor choice node containing the end of the link. For example in Fig. 9 the source of the link is an anchor in a data node, and the source context is the node's parent; the destination of the link is a composite node, which is also the destination context. (The source of a link in CMIFed is an anchor within an atomic component, the destination can be an anchor, or an atomic or composite component.)

While deriving the context from the structure may seem strange at first, we have managed to take into account the most probable uses for context in a hypermedia presentation. Common wished-for actions on following a link are:

(a) a presentation is playing and extra pieces of information can be played alongside (Fig. 3);
(b) the reader jumps to a completely new presentation (Fig. 5);
(c) a section in a presentation contains several sub-parts through which readers can navigate (Fig. 7);
(d) jumping back to the beginning of a presentation.

In (a) the destination context is a descendant of the source context, and the source context is retained and continues playing on following the link. In (b) the source and destination contexts are in two separate presentations and the source context is discarded (Fig. 6 with the "replace" case). In (c) the source and destination contexts are children of the same choice composite component, so that the source has to be replaced by the destination (as shown in Figs. 8 and 9). In (d) the destination context is an ancestor of the source context, and playing the destination context cancels the source context that was playing; the source context is then re-played as part of the destination context.

CONCLUSIONS AND FUTURE WORK

In this paper we have introduced the notion of context for a hypermedia link, showing examples of where context has been used, but not explicitly stated, in existing hypertext and hypermedia systems. The advantage of making contexts explicit is that authors are given control over how a presentation should behave on following a link, rather than being constrained by the choices hard-wired in a particular authoring system. We have presented a model for contexts which requires the specification of: source and destination contexts for a link; what happens to the source context on following the link; and where the destination context is played. We have implemented a subset of this model in our hypermedia authoring system CMIFed.

One of the future directions for our work is to generate hypermedia presentations on the fly, based on selecting data nodes from a store and assembling them into a presentation. By providing high-level constructs and making the issues in creating a presentation explicit, such as context for a link, we will be able to reduce the search space for constructing a presentation automatically.

ACKNOWLEDGEMENTS

The work reported in this paper was carried out within the Multimedia Kernel Systems group at CWI. Jack Jansen and Sjoerd Mullender have been involved with the design and implementation of contexts in CMIFed. We would also like to thank one of the conference reviewers for their extremely detailed and constructive comments.

REFERENCES

[Appl87] "HyperCard" authoring system for the Macintosh, Cupertino, CA

[Brow92] P J Brown, "UNIX Guide: lessons from ten years' development", ECHT '92 Proceedings (Milano, Italy, November), 1992, pp 63 - 70

[Buch92] M Cecelia Buchanan and Polle T Zellweger, "Specifying Temporal Behavior in Hypermedia Documents," ECHT '92 Proceedings (Milano, Italy, November), 1992, pp 262 - 271

[Bult91] Dick C A Bulterman, Guido van Rossum and Robert van Liere, "A Structure for Transportable, Dynamic Multimedia Documents," USENIX Proceedings (Nashville, TN, June) 1991, pp 137 - 155

[Fuji91] Kazutoshi Fujikawa , Shinji Shimojo , Toshio Matsuura, Shojiro Nishio and Hideo Miyahara, "Multimedia Presentation System 'Harmony' with Temporal and Active Media," USENIX Proceedings (Nashville, TN, June) 1991, pp 75 - 93

[Hala90] Frank Halasz and Mayer Schwartz, "The Dexter Hypertext Reference Model," NIST Hypertext Standardization Workshop, Gaithersburg, MD, January 16-18 1990

[Hard 93a] Lynda Hardman, Dick C A Bulterman, Guido van Rossum, "The Amsterdam Hypermedia Model: Extending Hypertext to Support *Real* Multimedia," Hypermedia 5(1) 47 - 69

[Hard93b] Lynda Hardman, Guido van Rossum, Dick C A Bulterman, "Structured Multimedia Authoring," Multimedia '93, (Anaheim, CA, August) 1993, also CWI report CS-R9304

[Macr90] Director version 2.0, MacroMind 1990 (dynamic media authoring tool for the Apple Macintosh)

[Niel90] Jakob Nielsen, "Hypertext and Hypermedia." Academic Press, 1990

[Ogaw90] Ryuichi Ogawa, Hiroaki Harada and Asao Kaneko, "Scenario-based Hypermedia: A Model and a System," ECHT '90 (INRIA, France, November) 1990, pp 38 - 51

[Pala90] Murugappan Palaniappan, Nicole Yankelovich and Mark Sawtelle, "Linking Active Anchors: a Stage in the Evolution of Hypermedia," Hypermedia 2 (1) 47 - 66

[Ross93] Guido van Rossum, Jack Jansen, K Sjoerd Mullender and Dick C A Bulterman, "CMIFed: a Presentation Environment for Portable Hypermedia Documents," Multimedia '93, (Anaheim, CA, August) 1993, also CWI report CS-R9305