User Interaction Styles in Museum Hypermedia

Franca Garzotto, Luca Mainetti, Paolo Paolini
Hypermedia Laboratory
Department of Electronics and Information - Politecnico di Milano
Piazza Leonardo da Vinci 32 - 20129 Milano - Italy
Phone: +39-2-2399.3623 Fax: +39-2-2399.3411
e-mail: {garzotto, mainetti, paolini}@elet.polimi.it

Abstract

This paper will analyse a variety of interaction styles that can be designed in hypermedia applications, will point out the need of using them consistently, and will analyze the possible design trade-offs between richness of interaction and easiness of use. Examples will be taken from a museum hypermedia we are currently developing in co-operation with the Poldi Pezzoli Museum in Milano.

Introduction

Multimedia and hypermedia can have a potentially enormous exploitation in museums, addressing different categories of users and serving for a variety of purposes. Just to mention few examples, we may think of hypermedia interfaces to museum databases - to support the activity of museum operators, or hypermedia information points, catalogues, or exhibitions - locally installed for museum visitors, distributed on CD-ROM's [19, 22] or on the Internet for education or research use.

Whatever the intended use of a museum hypermedia, its ultimate goal is *comprehension* and *usability*. In other words, it must allow users to *understand* the application content, the application goals and "message", and to *use* the system in a easy and natural way. Comprehension and usability both depend on a number of factors: the quality of contents (the clearness of written texts, the incisiveness of multimedia elements), the elegance of the lay-out, and, last but not least, the quality of *user interaction*.

The term "interaction" refers to how users operate on the various information structures and functionalities of a hypermedia application by acting on lay-out elements. In this paper, we will focus on interaction issues and will analyse a variety of interaction styles that can be provided in hypermedia applications. Our goal is to provide designers with a *taxonomy* of "hypermedia interaction styles," among which they can choose the most appropriate ones for their applications and the requirements of their users.

In hypermedia systems, the user interacts with the application at various levels, from operating on single information pieces (e.g., video elements) to acting on larger structures such as sequences or trees of nodes, in order to access information. Interaction is strictly related to both the nature of the application content and of the information structures used to organize the content itself. Interacting with media such as video and animation, for example, that are intrinsically "active" and "reactive," is quite different from interacting with media such as images or text, which are basically passive and can be at most scrolled and zoomed. Similarly, accessing information organized in linear structures (such as Guided Tours) can be much simpler than interacting with topologically more complex structures such as trees or arbitrary networks.

In order to make a precise analysis on hypermedia interaction styles, we will need an unambiguous terminology to describe data types and structural elements of hypermedia applications. For this purpose, we will adopt the **H**ypermedia **D**esign **M**odel HDM [4, 6, 8, 9, 16], which provides a rich set of primitives that are appropriate to this purpose.

The discussion on the various interaction styles will be exemplified by analysing a museum hypermedia we are currently developing in co-operation with the Poldi Pezzoli Museum in Milano. This application concerns the so-called "Agostinian Polyptych" by Piero della Francesca - one of the most mysterious and fascinating masterpieces of this artist - and, more generally, Italian Polyptychs in the Renaissance. The studies on the "Agostinian Polyptych", by a team of the researchers at the Poldi Pezzoli Museum, have involved the analysis of other works by Piero della Francesca and by other artists of the same period, as well as the analysis of many ancient and modern documents, in order to track the vicissitudes of the "Agostinian Polyptych" along the centuries (e.g., its possessors, its various restorations, etc.). In addition, the researchers have investigated a number of artistic and cultural events related to sculpture, texture, fashion, jewellery, every-day life, religious life in the Renaissance, that together can contribute to a deeper understanding of Piero's work and, more generally, of the Italian Polyptychs in the Renaissance period:

The results of these three-years of research will be presented during a "traditional exhibition" that is under organisation, and will be held at the Poldi Pezzoli Museum in Winter 1995. In parallel, we are developing a hypermedia version, that will provide a multimedia presentation of all the relevant material that is related to the "Agostinian Polyptych" research and have been investigated or discovered in these years. The hypermedia will allow the museum visitors to interactively explore both the material presented at the exhibition and a number of additional artworks, documents, research results, that will not be physically shown within the exhibition. A CD-ROM version is planned for the mid of 1996, to make the application available to a larger public for education or research.

The rest of this paper is organised as follows. Section 2 provides a short summary of the HDM model; it will define the terminology on which we will found the analysis of various styles of hypermedia interaction, described in Section 3. This section will also discuss some criteria of usage of the various interaction styles. Section 4 will draw the conclusions.

Hypermedia Structural Modelling with HDM

The Hypermedia Design Model HDM provides a set of primitives to describe the structures of an existing or to-be-developed hypermedia application, in a concise way, abstracting from implementation issues. This section is a short synthesis of main features of HDM. The interested reader is referred to the bibliography [4, 6, 8, 9, 16] for a more complete presentation of the model.

HDM primitives allow descriptions of hypermedia applications according to two levels: *in-the-large*, where large granules of information are considered, and *in-the-small,* where small granules of information are taken into consideration. We will describe first the features for modelling in-the-small, and later the features for modelling in-the-large.

HDM primitives in-the-small

Slot, frame and *node* are the structural primitives in-the-small of the HDM model.

A **slot** represents an atomic piece of information. It can be of a simple type, such as an "integer" (e.g., representing an historical period, say, 1400), "text" (e.g., describing a painting), "image" (e.g., showing a painting), or of a complex type, such as, for example, a video synchronised with a sound track.

Structurally speaking, a **frame** is an aggregate of slots, put together in order to "present" them in a co-ordinated fashion. All the slots related to a given painting, for example, can be put together in a frame.

A **node** is a navigational unit. In HDM, a node is always associated to a frame that represents the node content. A node is interconnected to other nodes, but the organisation of several interconnected nodes belongs to the in-the-large realm, discussed in the following section.

HDM primitives in-the-large

Entity, *component*, *entity type*, *collection*, *link*, and *link type*, are the *structural* primitives in-the-large that we consider [1].

Entity

An entity is a representation structure that corresponds to some real-world object. A Painter or an Art Period, are all example of entities. An entity is made of a set of **components**.

A component groups together a number of nodes, in a granule that corresponds to a constituent of a real world object. For example, the various life periods of a given artist can be modelled as component of a Painter entity.

The arrangement of the nodes, within a component, and of components within an entity can vary, according to the topology of the entity itself: they can be arranged in a sequence, in a set, in a tree, etc. Sometimes, the various nodes within a component denote different *perspectives*, i.e., different ways to look at the same piece of information.

Entities that correspond to domain objects of the same class are grouped in the same **entity type**, and all share the same topology.

Example

The content of the Poldi Pezzoli hypermedia (shortly presented in section 1) has been organized in twelve entity types: Restoration, Reconstruction, Renaissance Fashion, Texture, Polyptych, Jewellery, Sculpture, Archive Documents, Bibliography, Painter Life, Glossary, and General Comment.

For some entity types, entities are single-component (i.e. they are represented by one component only); in other cases, entities have a linear structure, a tree-shape structure, or a lattice structure. Usually, each component is made of several nodes; one node, that we call "main node of the component," summarizes the content associated to the component itself. The other nodes are called "details nodes" and contain a deeper discussion of some specific issue, e.g., details on a specific portion of a Painting presented in the main node. In most cases, each component also contain a node called "visual presentation" which provides a visual perspective of the main node, and only stores a large image concerning the component subject, an image caption, and a short sound comment.

As an example, let us analyse the entity type "Polyptych." Each entity of this type is made of various components (arranged as a tree-shaped structure). The (single) node of the root component has the purpose of providing an introduction to a Polyptych. The frame associated to a root node contains the

1 Entities, components, and collections are examples of **composites**, in the sense defined by the Dexter Hypertext Reference model [12, 13]

following slots: *title* (the name of the Polyptych), *description* (a short descriptive text), *audio* (a sound comment), *image* (a picture showing the Polyptych, and *animation* (which creates some visual effects to show, for example, the geometrical structure of the Polyptych, the perspective axes, the symmetry of the figure, the different layers of restoration, etc.).

The nodes of the first child component discuss the *shape* of the Polyptych, under different perspectives. The other child components describe the "Collection Vicissitudes" of the various fragments of a Polyptych (i.e., the persons or institutions who have possessed the different parts of the Polyptych along the centuries[2]). Each of these components is made of a sequence of nodes.

Collection

A collection groups together a set of objects called **members** of the collection [7]. Members of a collection can be entities, constituents of entities (i.e., components or nodes) or other collections (nested collections). Objects can be organised into collections in order to represent a taxonomy, according to some objective criteria, or can be grouped together simply to improve application readability and help the user to better find his/her way around in the application.

A collection also holds a distinguished node, locally associated to the collection itself and called **centre-collection-node** (or **centre**, for simplicity). The purpose of the centre is of illustrating the content of the collection itself, and also to provide an access path to the members of the collection. Finally, a collection has a **topology**, which defines how members are mutually related and related to the centre.

Example

In the design of the Poldi Pezzoli hypermedia, we have defined two collections for each entity type: a "long collections" and a "short collections".

A "long collection" of an entity type T groups all entities of type T, and organises in a sequence all information represented in each entity. Basically, a long collection can be regarded as an exhaustive linearization of the content of all entities of a given type, that allow users read, in a linear fashion, all the material related to a given topic. A node in the centre of a long collection of type T is made of two

2 Along the centuries, many Renaissance Polyptychs were split in pieces, to be sold separately to different collectors, and are now dispersed around in private or public institutions in different parts of the world. For example, portions of the Agostinian Polyptych are today exhibited at the Poldi Pezzoli Museum in Milano, the Frick Collection in New York, at the National Gallery in Lisbon, the National Gallery in Washington. Many pieces, unfortunately, have been lost.

slots: a set of text slots listing the titles of all entities of type T, and arranged on the screen so that they represent the entities' topology, and an audio slot of approximately 30 seconds, storing two synchronised sound tracks - a voice comment and a low-volume Renaissance music in the background. All long collections are grouped in a "master" collection. The centre of the master collection contains, among other slots, a short text slot for each entity type that briefly introduce the subject matter of the type.

"Short collections" are basically a shortened version of long collections. They have been designed for users who want to have a quick overview of a given topic. A short collection groups only the nodes of type "visual presentation" of entities of type T; its centre node is very similar to long collections, but the title slots of the entities are arranged in linear fashion.

Other collections are less homogeneous than the ones above described. For example, the collection "Restoration of the Agostinian Polyptych", describes the restoration vicissitudes of the Agostinian Polyptych, and the motivations for some restoration choices, etc. Again, the centre introduce the collection content; the members are arranged in a linear order and include pieces of information from several entity types (e.g., Reconstruction, Restoration, Renaissance Fashion, Texture, Polyptych, Jewellery, Sculpture, Archive Documents, Bibliography, Painter Life) that together help understanding the restoration history of this art work. Members of type General Comment are used as a rhetorical device to glue these different pieces of information, and to provide "context" and explanation for the various topics, in order to make the collection more readable and understandable.

In the Poldi Pezzoli application, Long and Short collections have a linear topology (sequences). In some of the other collections, we arrange the collection members according to two alternatives topologies, that are intended for different categories of users.

The linear topology, in which the centre and the members are arranged in linear order, has been designed for "ordinary users" non necessarily expert on the application domain. Tree or lattice topology exploits semantic dependencies or relationships among the collection members, and has been designed for users that are assumed to be expert on the application domain. The two topologies co-exist at the same time, but only the expert user (once he has identified himself) can exploit both.

Figures 1 and 2 show the linear and lattice topology of the members of Poldi Pezzoli Collection "Textiles".

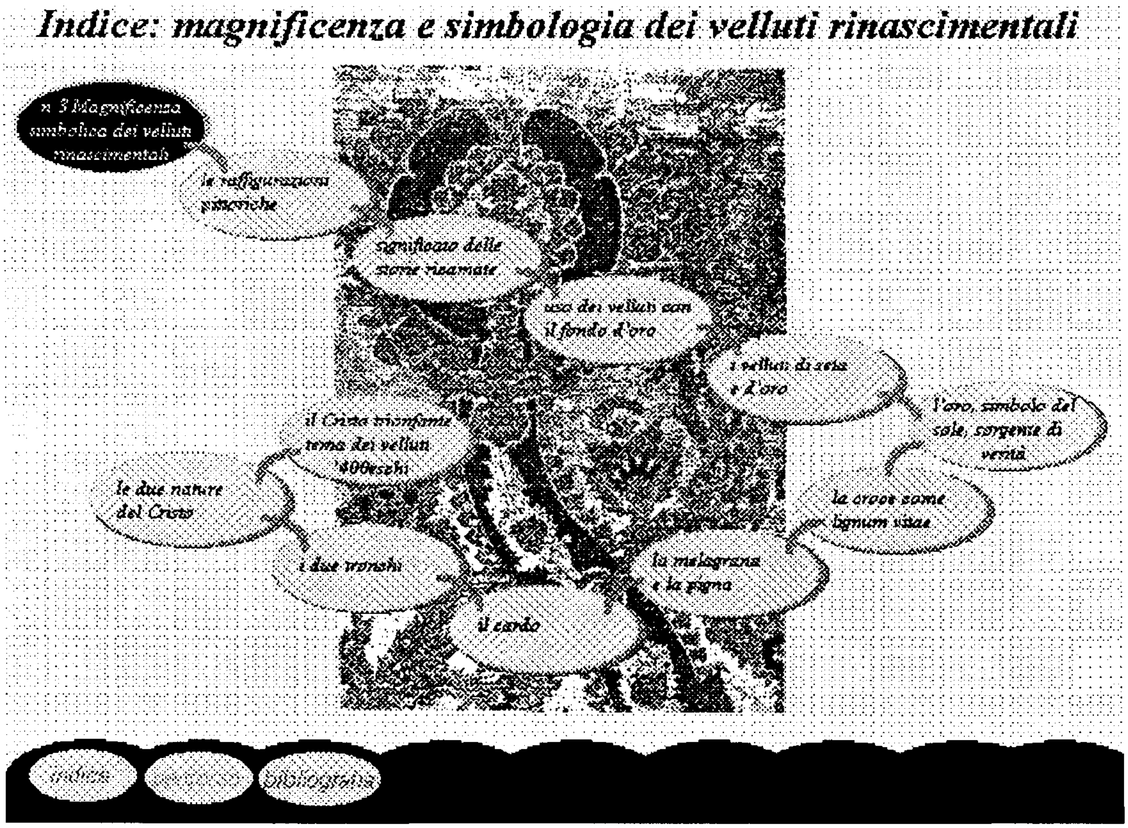

Figure 1 - Linear Topology of Poldi Pezzoli Collection "Textiles"

Links

In HDM, links are connections among nodes, entities, or collections. We consider four categories of them: *perspective links, structural links*, *collection links*, and *applicative links*.

Perspective links connect different nodes of the same component.

Structural links tie together the nodes of the different components belonging to the same entity; they are determined by the structural features of the entities (tree, sequence, etc.). **Collection links**, in general, connect the collection node to the collection members (and vice versa), and also the members among themselves; the features of these links depend upon the topology of the collection.

Applicative links connect together two "objects" according to some intended relationship. At either sides of a link there can be a node, an entity, or a collection. An applicative link, for example, may connect a painter to the historical period when he lived, or a painter to the collection of his major paintings (and vice versa).

Links that have similar meaning and connect objects of the same category or type are grouped under the same **link type**.

Example of these various categories of links will be discussed in the next section, from the viewpoint of user interaction.

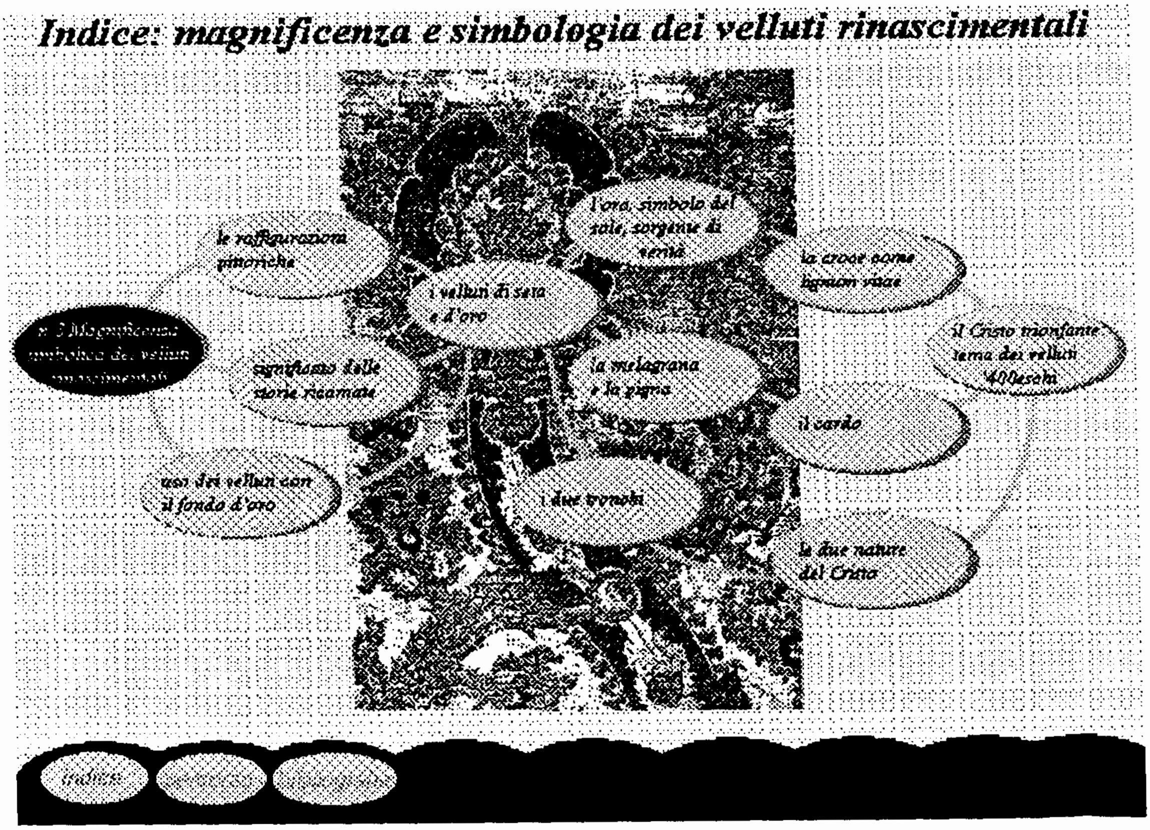

Figure 2 - Lattice Topology of Poldi Pezzoli Collection "Textiles".

Collection members are the same, but their organization is different (compare it with Figure 1)

Hypermedia Interaction Styles

If we go back to the history of hypermedia, we can observe that in hypertext - the ancestors of hypermedia systems - interaction was mainly referred to navigation, i.e., the way by which users can traverse links and browse around across different pieces of information. More recently, hypertext navigation has been combined with queries "a-la-database", and content-based search, with relative relevance depending upon the intended use of the application. The introduction of multimedia - which has lead to today's "hypermedia" - has induced new styles of interaction due to the intrinsically dynamic nature of multimedia data values.

In this section, we will introduce a *taxonomy* of hypermedia interaction styles, that are related to the granularity of the structures the user interacts with, and to the nature - static or dynamic - of the media involved.

We will first distinguish between *interaction in-the- small* and *interaction in-the-large*.

Interaction in-the-small concerns the way user can operate within an individual node, i.e., with its slots, or within nodes of the same component.

Interaction features in-the-large concern the way user can explore, by navigation or by query based search, the information stored within entities or collections.

Interaction in-the-small

Within interaction-in-the-small, we will distinguish among *single slot interaction* and *multiple slot interaction*.

Single slot interaction

A single active slot in a node it may be played *by default* when the user enter the node, or it might be played *upon request*, as an option to the reader. Slot interaction is related to the media type of the slot itself. Interaction with static, i.e., passive media (the only available in traditional hypertext) is often not existing, or it is relatively simple. At most, a picture can be zoomed in or out, and a text can be scrolled. The interaction with a slot storing a value from a dynamic value such as video, sound, animation, etc. can have sophisticated features. Dynamic media are active, i.e., their state can autonomously evolve along the time, and re-active, that is, their state can change based on external stimuli, i.e., on user's actions. A user can halt, suspend, restart a video or an audio track. These are all examples of interaction at slot level. It should be noticed that in most hypermedia applications, multimedia is confined at slot level, in the sense that the peculiar features of active media are used to build complex, but conceptually atomic objects [3, 5, 11].

More sophisticated uses of multimedia, at the level of interaction in-the-large, will be discussed in the following of this section.

Multiple slots interaction

When a node stores several active media slots (say, multiple sound tracks, videos, animation, etc.), they can have some synchronisation properties, i.e., some of them must be played in parallel. Multiple parallel slots can be activated by default when the user enters a node. Alternatively, user can interact with multiple slots and play them upon request. In the latter case, the mutual control among different slots must be defined.

For example, the activation of a video slot might automatically play a musical comment which cannot be played unless the video has been activated. Alternatively, the music might be activated independently from the video. Video and sound might be suspended simultaneously, i.e., suspending the video implies the suspension of the sound, or each of them can be suspended independently from the other one.

Examples

In the Poldi Pezzoli hypermedia, nodes concerning the Restoration of a Painting contain, among other slots, a painting image, an animation that shows (with an overlay effect on the painting image) various restoration operations on the painting, a voice comment that explains the animation, and, optionally, a short (low volume) music piece. When the user has access to a node of this type, all static slots are presented but only the music slot is played by default. The user can activate the animation, which is synchronised with the voice comment, while the music remains in the background. Animation and voice can be stopped, suspended, or re-started (both from the beginning and from the time position in which they have been suspended), but these interactions have no effect on the musical background. Music can be stopped and restarted upon request, without effecting the parallel play of animation and voice.

Interaction in-the-large

Hypermedia interaction-in-the large concerns the way by which user searches for information across the representation structures of an application[3].

Most hypermedia applications provides two basic ways of search: navigation-based and query-based.

3 Hypermedia applications may provide functionalities other than navigation or search (e.g., tools such as "print", "annotate", etc.), that increase the complexity of user interaction and can be analyzed from an in-the-large viewpoint. Since these functionalities are not standard, and are not hypermedia-specific, we will omit them from our discussion.

Navigation-based interaction

Navigation consists in traversing the links connecting the various structures. We can assume that the definition of link traversing is the standard one: activating a link from its source leads to the activation of its destination and the de-activation of its source. However, this basic concept needs some extensions when active, time-dependent media are taken into account. If active slots were being played in the source node, for example, it must be defined the state in which the source itself is left (e.g., the original state, or the state reached at the moment of departure from the node) [14]. If, for example, a video and a sound were being played, the designer must specify whether they are to be suspended, or reset at the beginning, or whether one of them must continue playing [7]).

Whatever the behaviour of link activation, different categories of links discussed in the HDM model induce different *navigation patterns*. We will distinguish among *within-component navigation, structural navigation, applicative navigation, collection navigation, history navigation.*

Within-component navigation

Different nodes of the same component can be connected by perspective links (see session 2). Perspective navigation is the simplest example of navigation. The effect of perspective navigation is, from a cognitive point of view, very intuitive for a reader, since it corresponds to visiting the same information element under a different perspective, without changing the current focus of attention, i.e., the current component of the current entity.

Structural navigation

The natural way of exploring an entity is to follow the structural links for **structural navigation.**

Structural navigation is activated by selecting a structural link. Structural navigation allows the reader to browse around chunks of information belonging to the same entity, i.e., concerning the same topic. Navigationally, it is a little more complex than perspective navigation, but is still cognitively simple, since it does not change the entity instance of interest.

Moving among different in the large objects - entities of collections - can be performed with *applicative navigation* or *collection navigation.*

Applicative navigation

Applicative navigation exploits application links, and allows visiting independent, but semantically related, information segments. From the user standpoint, this type of navigation can be the most disorienting. As the reader, in fact, traverses application connections, he/she will perceive that the information environment (i.e., the topic he/she is exploring), has abruptly changed (e.g. from a painting to a restoration method). In fact, application links, in general, connect nodes defined in different entities.

Examples

In the Poldi Pezzoli application, perspective links allow users to move from the main node of a component to a visual presentation node (see section 2), or to details-nodes about the component subjects, or from there back to the main node.

Structural links connect, for example, the root component of a Polyptych to its Collection Vicissitudes or to the discussion of its shape. They allow user to look at different topics strictly related to the Polyptych they are exploring.

Applicative links allow users jump across objects of substantially different nature, connecting, for example, a Polyptych to an entity of type Jewellery or of type Texture.

Collection navigation

As far as a collection is concerned, there are two navigation styles, that can be intermixed: *index navigation* and *guided tour navigation* [10].

In the **index** navigation, a collection node is used to access the member via collection links, and from a member it is possible, in general, to navigate back to the collection node.

In the **guided tour** navigation, one can move from one member directly to another member (again following collection links) without need to traverse each time the collection node. Typical interaction commands in linear guided tours are "Next", "Previous", "First" and "Last", that allow the user to directly access the following or previous node, from the current one, or the first or last member in the collection.

A typical combination of index and guided tour allows the random selection of a member, and from this the subsequent exploration of other members [24].

A guided tour can be executed either **manually** (i.e. the user controls the transfer from one member to another) or **automatically** (i.e. some time dependent synchronisation provokes the transfer from one member to another one).

Automatic guided tour exploration of a collection can be strongly affected by the presence of multimedia slots. If no active media are involved, simple time-outs can be used to control the automatic transition from a member to another one. If the members, instead, contain active media slots, these can be used to control the navigation. If each member node, for example, contains an audio-track, the transfer to the next member can occur when the audio-track of a member has been fully played, which also determined the activation of the audio-track of the new member. Things can get more complicated if members have more than one active slot, and if the collection nodes themselves have their own active media slots [7]; the different slots must be synchronised, i.e., either played in parallel, or played in some kind of sequence, with a control left to the reader, or with a mixture of the methods.

Complex issues arise when a member of a collection is itself a collection, i.e., when we have **nested collections**. Nested collections induce to navigate at different levels, i.e., from a member collection to another member collection, and *within* individual member collections. Navigating from a member collection to another member collection, either as with index or as guided tour style, means moving across their centre collection nodes. The situation might become disorienting for the user when he or she starts exploring one of the member collections.

Let us assume for example that a user starts a guided tour collection about "Restoration Techniques" and within it we start a guided tour on "Technique 1". One of the design problems is to define what is the proper interaction when the user reaches the end of the current guided tour (Technique 1) and tries a Next operation, since various interpretations can be provided for this action.

A simple interpretation is to consider the Next operation as not executable in the described situation. Another choice can be to interpret a Next request as "go to the next member collection (at the current level of nesting)", thus having the user jump to the centre collection node of collection "Technique 2". Another possible interpretation can be to take the user to the first node of the current collection, that is, to make him "cycling" around the current second level collection. Finally, a further design choice can be to provide all the above alternatives, and to allow the user choice between them. This multiple interpretation offers the advantage of a fast interaction for the user, if he/she knows what can be done, but has the disadvantage of adding additional complexity to the interaction.

Examples

Collections in the Poldi Pezzoli application are either flat sequences of nodes, or two level nested collections (i.e., collections of flat collections) , or have tree or lattice topology (see section 2).

In flat collections, we provide both index and guided tour navigation. Collection link connect the centre node to each member, and each member to the next, previous, last and first member. No "next" link is available from the last member.

In a nested collection, the centre node at the upper level is connected to the centre nodes of the member collections. From the centre of a second level collection, there is a link to each member, and the other way around. From each of the member nodes of a second level collection, there are collection links to the centre node of the next, previous, last and first member collection. No link allows to jump from a member of a lower level collection to a node of another collection. Consistently with flat collections, no "next" navigation is available from the last node.

In the Poldi Pezzoli hypermedia, we have designed a number of automatic guided tours. For example, in the tour "Short Introduction to the Agostinian Polyptych Texture Analysis", the transition from a node to the next one is controlled by a voice comment. When the first node of the collection is activated, its voice comment starts by default; when the voice is over, the link to the next node is automatically executed. At any time, the user can switch from the automatic navigation to the manual navigation (and vice versa).

As already discussed in section 2, tree or lattice topologies for collection have been designed for expert users only. In both cases, the navigation pattern allows both index and guided tour access. However, given the non-linear structure, the meaning of a "next" (in lattices) or "previous" (in trees and lattices) navigation command is not deterministic, since a member might have more than one "next" or "pervious" element. In this case, in our interaction mechanisms the effect of a "next" or "previous" command is presents to the user the list of all next or previous elements, among which the user can choose.

History navigation

History navigation allows the user explore previously visited nodes. Most hypermedia applications provide only "previous" links on the history, that allow just to backtrack along the navigation session. Even in the case of the simple "go-back" interaction style, however, there are several possible variants in the interpretation of this operation.

In a destructive approach, the nodes that are visited during non history navigation are accumulated on a stack, and user can back-track along this stack, down from the top node down; each time a previously visited node is access, it is removed (popped) from the stack.

In a different approach, the nodes visited during non-history navigation are stored in a circular list; the nodes visited during history navigation are not removed from the list, and once the user goes-back from the first visited node he can return to the node where he has started history navigation.

To provide a more powerful styles of interacting with the history of a navigation session, interaction, we can model the **history** as a **run-time collection**, i.e., a collection which is dynamically built on the fly as the user explores an application. With this interpretation, we can provide all the interaction facilities defined for collections: the possibility of showing the collection node displaying the whole set of visited nodes, to navigate back and forth as a guided tour, or to select directly any of the previously visited node, as in index navigation.

Examples

In the Poldi Pezzoli application, we provide both a history of all visited nodes, called Global History, and multiple histories, in which the visited nodes are organised by entity type. All these histories can be navigated as indexes or linear guided tours. With multiple histories, the user can select the type of previously visited information he is interested to, requiring, for example, to explore again only previously visited nodes of type Polyptych, and explore again the polypthycs he has already visited.

Query based interaction

A query (in any environment, not just hypermedia) specifies conditions which must be satisfied by information structures of a given type. Query evaluation returns a set of information objects for which the query conditions hold.

In most hypermedia applications, the simplest form of query is keyword-based - the user specifies one or more terms, and the result of evaluating the query returns a list of items that identify all the nodes in which those terms occur. More sophisticated mechanisms of query specification can be envisioned, similar to those found in data base systems.

Whatever the query specification language, however, hypermedia queries must be interplayed with navigation. In most cases, the user does not simply want to inspect the list of node identifiers returned by the query. Nodes can be complex multimedia objects, and the user may need to select a node of interest and to explore it.

An effective way to interplay query and navigation is to use again the notion of collection, and to interpret the result of a query as a *run-time collection*, the members of which are the nodes selected by the query, and the collection node contains the list of member "identifiers"; With this interpretation, the user can explore the query results either as an index or a guided tour, or both. An interesting use of this approach is to allow user store the query-based collection, for future use, by providing an explicit "save-collection" command, or to filter the current run-time collection by refining the query specification.

Example

In the Poldi Pezzoli application, we will use data base techniques that through a query-by-example-interface, allow the user to define simple queries such as "retrieve all nodes concerning 49Optychs

Conclusions

Multimedia and hypermedia offers a variety of interaction styles that might disorient both the designer who has to choose the most appropriate one for the application tasks, and the end user who uses them. In this paper, we have introduced a set of reference concepts for analysing hypermedia data types and structures from the perspective of user interaction. In addition, we have attempted to provide a taxonomy of interaction styles, against which to compare, or toward which to reduce, various interaction modes found in exiting or to be developed hypermedia.

We will conclude with two considerations. First, different interactions styles must be provided in a *consistent* way, in order to reduce user disorientation. Consistency can be synthesised in a simple generic rule, that states that *conceptually similar elements should be treated by the user in a similar fashion, while conceptually different elements should be treated differently*. Consistency is considered by several authors [1, 14, 17, 18, 20, 21, 23] as one of the crucial criteria for usability [2].

The second consideration is that a balance must be found between richness of interaction, intuitiveness, and easiness of use, avoiding to transform a hypermedia application in a pure show of technology.

References

1. Balasubramanian P., Isakowitz T., Stohr T., "RMD: A Methodology for the Design of Hypermedia Applications", In *Comm. ACM,* August 1995 (in print)

2. Bearne M., Jones S., Bearne J. S-F. M., "Towards Usability Guidelines for Multimedia Systems", In *Proc. ACM Multimedia '95*, S. Francisco, CA, Oct. 1995

3. Buchanan M.C., Zellweger P., "Automatic Temporal Lay-out Mechanisms". In *Proc. ACM Multimedia '93*, Anaheim, Ca, 1993

4. Cavallaro U., Garzotto F., Paolini P., Totaro D. "HIFI: Hypertext Interface for Information Systems". In *IEEE Software,* Nov. 1993

5. De Mey V., Gibbs S., "A Multimedia Component Kit". In *Proc. ACM Multimedia '93*, Anaheim, Ca, 1993

6. Garzotto F., Paolini P., Schwabe D. "HDM - A Model Based Approach to Hypermedia Application Design" In *ACM Trans. Inf. Syst.*, 11 (1), Jan. 1993

7. Garzotto F., L. Mainetti, P. Paolini "Adding Multimedia Collections to the Dexter Model". In *Proc. ECHT'94 - ACM Conference on Hypermedia Technology,* Edinburgh, UK, Sept. 1994

8. Garzotto F., Mainetti L., Paolini P., "Hypermedia Application Design: A Structured Approach". In *Designing User Interfaces for Hypermedia*, W. Schuler, J. Hannemann, N. Streitz (eds.), Springer Verlag, 1994 (in press).

9. Garzotto F., Mainetti L., Paolini P. "Hypermedia Design, Analysis, and Evaluation Issues". In *Comm. ACM,* August 1995 (in print)

10. Garzotto F., Mainetti L., Paolini P. "Navigation in Hypermedia Applications: Modeling and Semantics". In *Journal of Organizational Computing* (in print)

11. Gibbs S., Breiteneder C., Tsichritzis D., "Data Modeling of Time-Based Media". In *Proc. ACM SIGMOD,* Minneapolis (USA), May 1994

12. Gronbaek K., Trigg R.H., "Design Issues for a Dexter-Based Hypermedia System". In *Comm. ACM, 37 (2),* Feb. 1994

13. Halatz F., Schwartz M. "The Dexter Hypertext Reference Model". In *Comm. ACM, 37 (2),* Feb. 1994

14. Hannemann W, Thuering M, Haake A., "Designing for Comprehension: A Cognitive Approach to Hypermedia Development" In *Comm. ACM,* August 1995 (in print)

15. Hardman L., Bulterman D.C.A., Van Rossum G., "Adding Time and Content to the Dexter Model". In *Comm. ACM, 37 (2),* Feb. 1994

16. HIFI, HDM+ User Manual, ESPRIT Project HIFI, 1993

17. Hodges M., Russell M. Sasnet, "Multimedia Computing - Case Studies from MIT Project Athena", Addison Wesley, 1993

18. Kahn P., "Global and Local Hypermedia Design in the Encyclopaedia Africana". In *Proc. of IWHD - International Workshop on Hypermedia Design*, Montpellier, France, June 1995

19. Microsoft, "Art Gallery CD-Rom - The Collection of the National Gallery, London", Microsoft Corporation, 1993

20. Nielsen J., "Usability Engineering", Academic Press, 1993

21. Preece J., "Human-Computer Interaction", Addison Wesley, 1994

22. Rubistein B., "The Micro Gallery at the National Gallery". In *Proc. of ICHIM'91 - International Conf. on Hypermedia and Interactivity in Museums"*, Cambridge, UK, Sept. 1991

23. Shneiderman B., "Designing the User Interface", Addison Wesley, Mass. 1987

24. Trigg R.H., "Guided Tours and Tabletops: Tools for Communicating in a Hypertext Environment". In *ACM Trans. Inf. Syst.* 6 (4),1988