Patterns for Time-based Hypermedia Artifacts

Rui Lopes, Luís Carriço
LaSIGE & Department of Informatics
University of Lisbon
{rlopes,lmc}@di.fc.ul.pt

Abstract

With the incresing presence on the Web of hypermedia contents with temporal constraints (time-based hypermedia), such as multimedia presentations, there must be a precise form to create and manage this type of contents, due to its inherent complexity. Simple mistakes in the design process can result in unusable user interfaces, by posing unnecessary cognitive pressure on the user. This paper presents a set of design patterns towards the time-based hypermedia domain. These patterns gather several guidelines found in related literature and successful applications, ready to be used by system developers and application designers, at different levels (content, navigation, presentation, and interaction design). By following the good practices presented, better applications can be delivered to the final user.

1 Introduction

Patterns synthesize good practices in a given domain. Initially it was a concept introduced in the urban planning field [AIS+77], but since then has been largely embraced by computer science, especially in software development and methodologies [GHJV95]. These patterns have been described [Sch95] as ways to reuse software architectures, standardize the vocabulary used by software development teams, capture knowledge used implicitly, as well as provide a way to specify tradeoffs and design alternatives in a given domain. As each software development domain matures, several patterns emerge as the way to solve domain-specific problems.

Hypermedia is a well-known domain (its most successful implementation is the *World Wide Web*), defined by its inherent non-linearity in static (non time based) content structure. This network-alike structure has a heavy focus on navigation capabilities, based on notions such as anchors and links, thus becoming quite complex, posing difficulties both to the content creator and to the end user (the "lost in hyperspace" problem) [EH99]. The hypermedia domain has been targeted by several design methodologies, such as RMM (Relationship Management Methodology) [ISB95], OOHDM (Object-Oriented Hypermedia Design Method) [SRB96], UWE (UML Web Engineering) [BKM99], amongst others. These methodologies define several layers on application design (e.g. content design, navigation design, presentation design), as a way to minimize incoherences in hypermedia applications. To further extend these methodologies, several concepts were introduced by pattern systems [GRS97, Ber98], such as providing advanced navigation mechanisms and interface guidelines for hypermedia application designers.

Multimedia applications also have their specific concerns. Typically, these applications are defined through the composition of different pieces (text, image, sound, video) along a presentation timeline, with sparse support for navigation capabilities. A typical example relates to playback a sequence of animations with attached subtitles. Some work has been done in finding a pattern system targeted to multimedia applications design [CL98], focusing both on spatial composition of content and providing time synchronization capabilities.

On a higher level of application abstraction comes the interaction domain, which is a cross-cutting concept to all previously presented application domains. This level's focus can range from defining common interaction mechanisms for well-known devices (e.g. mouse, keyboard), to complex multimodal interaction mechanisms, such as combining voice activated commands with gesture recognition capabilities, and weighting each modality's relevance with different criteria. Such range of capabilities and possibilities leverages the need for interaction design pattern systems. A very succesful approach has been presented in [Tid99], synthesizing good practices for designing interactive artifacts. By having coherence amongst different applications' interaction mechanisms, higher usability rates can be achieved (as users tend to get used to the metaphors defined by the patterns).

Time-based hypermedia emerges as a mix of all the domains described earlier, by adding time management capabilities inherent to multimedia (such as timelines and time-based composition) and interaction mechanisms to the traditional hypermedia domain. Formal models for time-based hypermedia appeared [HBR94], and concrete document formats for the Web using them [Jef01]. An initial approach for developing time-based hypermedia systems and designing time-based hypermedia applications would be to follow the guidelines provided by all the pattern systems enumerated earlier. However, the inherent complexity of mixing multimedia content with navigation mechanisms, time constraints, presentation capabilities and interaction mechanisms poses critical difficulties to developers and designers, such as volatile links (available during just a short amount of time). This will result in poor systems and applications, which will be reflected in critical usability issues. Therefore, a pattern system targeted for this domain should help system developers and application designers to better understand the domain and, in conjuction with the previously referenced pattern systems, deliver better systems and applications of the time-based hypermedia domain.

The good practices synthesized by the patterns presented in this paper have been widely used in time-based hypermedia systems, whether in speech based hypermedia [Aro91], video based hypermedia [SBS96, CG02], multimedia [HBR94, Jef01], or even in vertical domains, such as Digital Talking Books [CGD+03, CDL+05]. These patterns should be used by anyone who has a good knowledge of hypermedia or multimedia domains, either being developing time-based hypermedia systems, or designing applications – using complex authoring tools (such as Macromedia Director), automated content production frameworks [Lop05], complex time-based hypermedia formats (e.g. the HTML+TIME language), or even simple document formats (e.g. SMIL language).

Some specific vocabulary of the time-based hypermedia domain is used throughout the pattern system, associated with time-based navigation capabilities, thus being described as follows (for clarification purposes):

- *Time modifier link*: A navigation link whose travessal will trigger a change in the presentation timeline;
- *Link opportunity*: The timeframe in which an anchor is presented (e.g. anchor embedded into a video resource during a few seconds);
- *Anchor manifestation*: The rendering of a link in a specific media (e.g. blue underlined text *vs.* sound beep);
- *Anchoring context*: The content surrounding the presentation of an anchor which helps the user to understand the context in which the anchor is manifested (e.g. the whole phrase where a text-based anchor is found);
- *Anchor deactivation*: Event that occurs when a link opportunity is no longer available;

A specific pattern classification has been defined and used as the structure in Section 2 of this paper. This classification is based on [RSG97], where the pattern space is divided in two categories (as outlined in figure 1): Hypermedia Systems and Hypermedia Applications. The first category relates to patterns that are supposed to be followed by system developers, whereas the second should be used by application designers. A thinner sub-division was defined on the latter category, reflecting the different phases present in hypermedia design methodologies, as well as integrating interaction design into the development process.

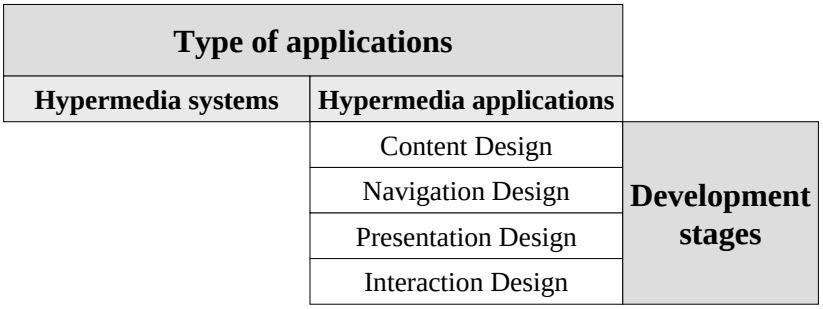

Figure 1: Redefined hypermedia patterns space

The pattern language format used in this paper follows the good pratices presented in [MD96], towards time-based hypermedia systems developers and application designers. Each pattern states its *Context* for applicability, the *Problem* it tries to solve, which *Forces* may lead to using the pattern, the *Solution* provided, amongst other complementary information. In addition, a new element has been introduced, *Overrides*, which states when a pattern replaces another one specified by other pattern systems (as all the previously referenced systems should be taken into account when developing time-based hypermedia systems and applications design).

2 Patterns

2.1 Patterns for System Developers

2.1.1 Time Composition

Context: The system must present several time-based contents, potentially in parallel and relative to each other.

Problem: What composition features should be at the disposal of a time-based hypermedia application designer that lower the burden of arranging big sets of multimedia items along a timeline?

Forces:

- Repeating a sequence of time-based hypermedia media contents along a timeline is very time-consuming in the design process, especially when the sequence has to be redefined.
- Reuse of composed contents allows for a more flexible approach on designing time-based hypermedia applications.
- A designer does not know *a priori* the duration of a specific content (e.g, a continuous video stream), cancelling the ability of specifying an absolute time for another content to be presented afterwards.
- Having just the ability of arranging multimedia contents along an absolute timeline poses several difficulties on aligning media, as each content's time duration must be known by the designer.

Solution: The system developer should provide different time composition constructs, either through absolute timestamps (e.g., *play X at 01m:02s*) or higher level relations (e.g., *play Y after X*), therefore supporting both quantitative and qualitative time references, as described in [PLL95].

Implementation of time composition primitives in time-based hypermedia systems should provide recursion mechanisms (as seen on figure 2). By supporting these type of relationship management (as seen on figure 3, and formally described in [All83]), designers that target their applications to this type of systems have more powerful expression constructs.

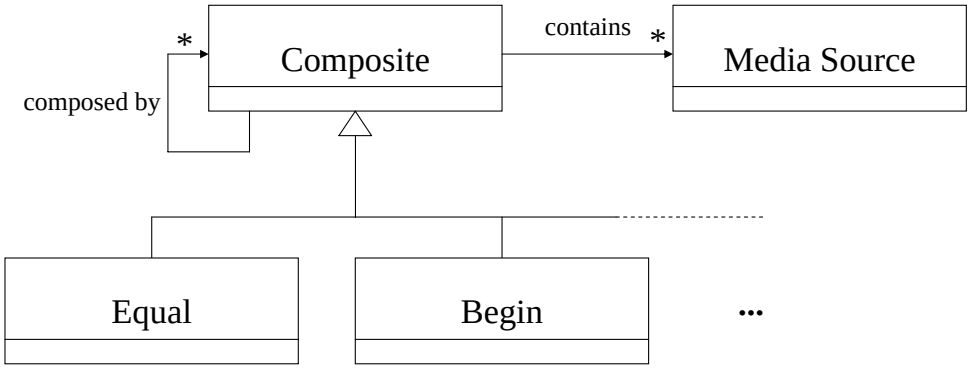

Figure 2: Time Composition pattern

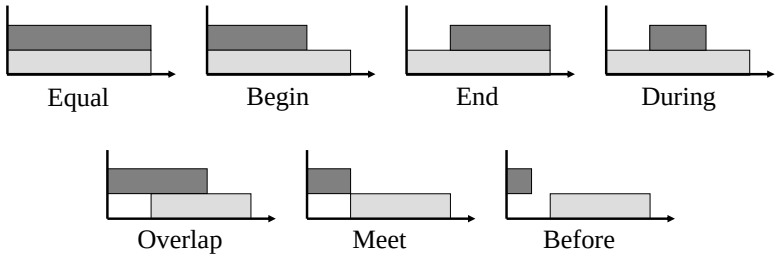

Figure 3: Qualitative time relations

Known Uses: SMIL players support time composition through `par`, `seq` and `excl` elements and relative timing constraints, recursively.

2.1.2 Always Synchronized

Context: The system has to synchronize multiple time-based contents that require intensive use of computer's resources.

Problem: Synchronization tasks specified by the application designer (such as parallel playback of audio and animations) start to fail, resulting from resource overloading of the playback platform.

Forces:

- Out-of-sync problems require strong cognitive efforts from users to comprehend the content.
- Network throughput stability can not be predicted when streaming media.
- Some computers do not have enough capabilities to present high-quality media and, consequently, not able to synchronize multiple media contents in playback time.
- A computer may be running several applications at the same time, therefore reducing the resource availability to playback multimedia contents.

Solution: When synchronizing several resource-intensive contents, the system should supply buffering mechanisms to each content that correspond to a given time interval. On playbacking the synchronized content (e.g. defined by the parallel time composition of two contents), the system must be aware of possible synchronization deviations between the contents, pausing the overall playback instead of just pausing the misbehaved contents. Detecting out-of-sync problems should be implemented through polling mechanisms (e.g. by querying each object with its current playback timestamp). As this goal may not be always possible to achieve, the system may support a mechanism to choose adequate versions of a given content to be played in a specific playback environment. This way, an application designer can provide multiple versions of some content to be chosen by the system. Regarding streaming media over network, start playing only when enough data has been acquired into a data buffer.

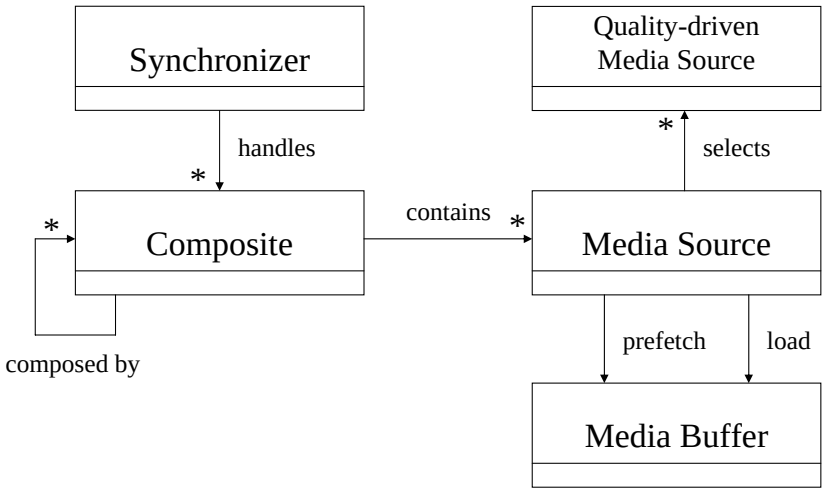

Figure 4: Always Synchronized pattern

Known Uses: Several SMIL [Jef01] players (e.g., RealPlayer, AmbulantPlayer) support the `prefetch` and `switch` content control module elements, to minimize out-of-sync playback of multimedia contents.

Related Patterns: This pattern can only be used if *Time Composition (2.1.1)* has been implemented in the system. Regarding the time-based hypermedia presentation design stage, *Strict Synchronization Presentation (2.4.2)* can help reducing out-of-sync related cognitive problems when full synchronization is not possible to guarantee.

2.1.3 Quick Response

Context: Time-based hypermedia applications that have user interaction capabilities (e.g. allowing a user to follow a link) allow the creation of volatile link opportunities.

Problem: If the system's response to a user generated event over a time-based link opportunity (e.g. trying to click its anchor manifestation) is slow, the link may not be available anymore, thus refusing the action the user tried to do.

Forces:

- A user wants to follow a time-based link, and starts clicking its manifestation, but the system does not respond, thus missing the link opportunity.
- A speech recognition technology is being used for user interaction, but an event's processing finishes after the anchor deactivation.

Solution: Store into memory the timestamp when the user event was generated (i.e. its starting time), coupled with the link's destination anchor. After processing the event, jump to the stored destination anchor. This mechanism certifies that user interaction is triggered at the right time and processed accordingly. To minimize the lag between the system's responsiveness, user interface and user interaction modules should be splitted into different threads (see figure 5). Priority-based command queues should also be used, where user interaction commands have the highest priority (in comparison with system generated events), thus reducing the cognitive pressure imposed by link opportunities.

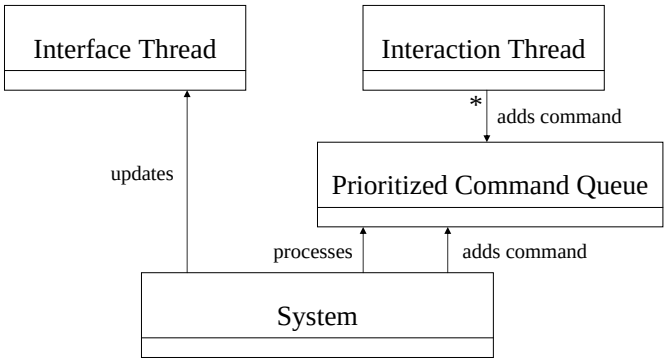

Figure 5: Quick Response pattern

Related Patterns: Cognitive pressure related patterns, such as *Link Opportunity (2.3.3)* and *Link Opportunity Manifestation (2.4.3)*, become more predictable if this pattern is implemented in time-based hypermedia applications, reducing the amount of ajustments over time intervals. Time-sensitive algorithms (such as Hidden Markov Model) can help reducing the computation time of user generated events.

2.1.4 Timeline

Context: Typical time-based hypermedia applications have long duration contents, and navigation mechanisms may be used to skip parts of the content, jump to specific content locations, etc.

Problem: The user should always be aware of the current playback time, as a way to provide helpful contextualization regarding the overall content duration. This should be done independently from the navigation mechanisms supplied by the presentation being played, as well as independently from the playback platform device capabilities, without overlapping the content playback.

Forces:

- The user wants to know the current presentation time quickly.
- The user wants to perform absolute and relative navigation strictly based on time, whether the presentation is playing, paused or stopped.
- Audio centred hypermedia imposes severe limits to non-crucial content presentation to the user, as opposing to visual presentation layers.
- Sometimes content length is not explicit in a document structure.
- Limited dimensions on a visual medium (e.g. a mobile phone screen) may require having alternate forms for informing about the current playback time.
- Some stream-based contents do not have duration (e.g. uninterrupt TV broadcast).

Solution: Provide timeline capabilities in the time-based hypermedia system, related to the presentation overall duration. When content length is not explicit, the system must retrieve that information from the media resources.

Implementation: On a visual medium, implement this pattern through a visual guide (i.e. typical progression bar). When facing constraints on a visual medium (either by reduced spatial dimensions or audio only systems), signal periodically the current time (like clocks and digital watches do) and provide non-visual mechanisms for content navigation (e.g. hardware buttons, gesture recognition, speech-based interaction).

Known Uses: The majority of media players implement the notion of an interactive visual timeline.

Related Patterns: The *Timeline (2.1.4)* pattern is a particular case of *Timed Active Reference (2.3.2)*, but generalized to reference the whole content, thus allowed to be implemented by system developers.

2.2 Content Design Patterns

2.2.1 Content Decoupling

Context: Your are creating a time-based hypermedia application defined by different contents related to each other in different ways, where some content items are going to be reused in different parts of the application.

Problem: How to circumvent the high coupling established by the relationships between different content items, allowing for content reuse and easier composition tasks?

Forces:

- You want to reuse a specific content in different presentation contexts (e.g. in a virtual art museum, displaying some painting in the context of a painter's biography *vs.* art movement).
- While producing some content, different auxiliar contents should be present (e.g., footnotes, table of contents, etc.), but inlining these into the main content pose maintenance and reuse problems.
- Different time composition instances should be allowed to be applied to some content, regarding different user needs.
- Different structures can provide help in semantic content navigation, as different ways to perceive the main content, but may no be possible to be displayed side by side with the main content.

Solution: The system developer should provide a data structure that allows the decoupling between all content parts. Each content part should keep its anchoring information, while relations should be decoupled. This type of content structuring is goes along the lines of the Dexter Hypertext Model [HS94]. Relationship between anchoring points should be manipulated later in navigation design, through traversal links and synchronization arcs. Figure 6 presents a reuse scenario of a content C_1 paired with other contents (C_A and C_B), reflecting the decoupling scheme of figure 7.

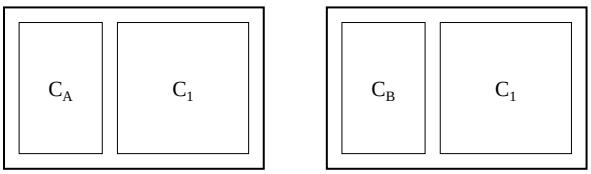

Figure 6: Content Decoupling usage scenario

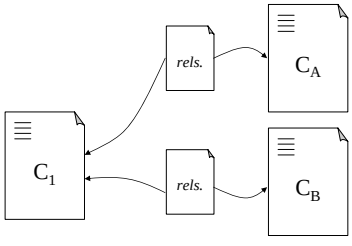

Figure 7: Content Decoupling pattern usage

2.2.2 Multiple Content Sources

Context: You want to produce multimedia content targeted to users with different needs and uses in different usage situations.

Problem: How to overcome the accessibility issues that different users have regarding content comprehension?

Forces:

- A visually impaired person is not able to interpret a visual medium.
- Audio impaired users should be presented with textual transcripts as an alternative to speech based content.
- Learning situations benefit from having multiple representations of some content, such as, while learning the english language, the word *building* should be associated with its spoken audio counterpart and an image representative.
- A user with little time to capture some textual content (e.g. newspaper) can benefit from listening to its audio counterpart while performing some other task.

Solution: Provide different media sources regarding the same content structure, by providing an additional layer of indirection between the content structure and the actual related media. Ideally, content should be available as text, image, video (or animation), and sound. If not possible or feasible, a given content must be, at least, in audio and visual formats. Later on, the application designer can choose whether this type of content is fully available to the end user, or just one subset.

Related Patterns: *Multiple Channel Presentation (2.4.6)* must be taken into account when creating content with *Multiple Content Sources (2.2.2)*.

2.3 Navigation Design Patterns

2.3.1 Integrated Navigational Contexts

Content: You want to integrate different navigation possibilities over a given content, whose presentation is changing as the primary content is being played.

Problem: How can multiple navigation structures be integrated in such a way, that they will always be synchronized with the content currently being presented to the user?

Forces:

- Different navigation mechanisms should be made present to the user, as a way to help understanding a content (e.g. table of contents, index, site map, bookmarks, etc.).
- A user should be able to be guided through the content by different perspectives.
- A user annotates a specific content and later wants to go back to that content, through the annotation binding.

Solution: Provide different navigation contexts for the content. When the time flow changes (either by time shifts specified in the application or triggered by the user through some navigation mechanism), all navigation structures should reflect this change accordigly, in a synchronous way.

Known Uses: Some Digital Talking Books players support the notion of Integrated Navigational Contexts, such as *Rich Book Player* [DC06]. Figure 8 presents Table of Contents and Annotations based navigation contexts over a book's main content.

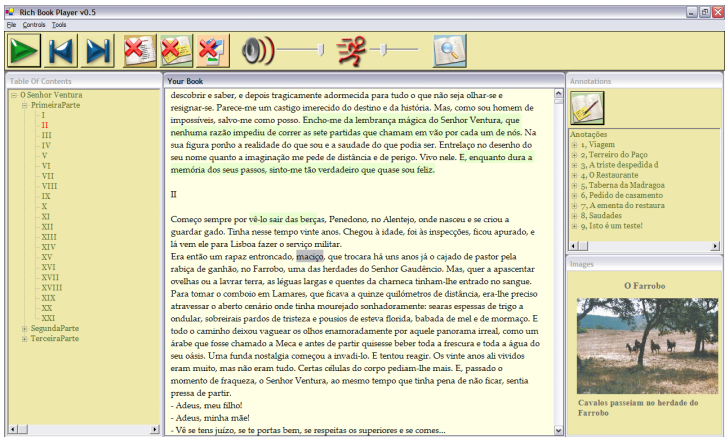

Figure 8: Integrated Navigational Contexts pattern applied on *Rich Book Player*

Related Patterns: *Content Decoupling (2.2.1)* eases the integration of different navigational contexts. If each navigation space provides its own content, *Multiple Channel Presentation (2.4.6)* must be used to check media channel collisions, and then proper navigation patterns should be selected (i.e., *Timewarp Linking (2.3.4)*, *Montage Time Linking (2.3.5)*, and *Time-break Linking (2.3.6)*).

Overrides: *Navigational Context* [GRS97] supplies the same notion of *Integrated Navigational Contexts*, but does not consider multiple context nodes with time related issues.

2.3.2 Timed Active Reference

Context: Complex contents and multiple navigation facilities open the way to a huge set of exploration paths for the user.

Problem: How to minimize "lost in time and space" issues on the user, created by following several links in a time-based hypermedia content?

Forces:

- A user follows several links and wants to get back to some previous content (not present on navigation history).
- History mechanisms do not provide sufficient information about exploration paths.
- Time introduces a new dimension to be taken into account by the user's cognitive map.
- Not all navigation is triggered by user events, time dynamics can provide automated navigation facilities over a time-based hypermedia content.

Solution: As time-based content presentation is inherently dynamic, navigation structures must be updated constantly by presenting the current navigation status within each navigation structure, whether if triggered by user events of just through the application's time flow. If possible, different navigation structures should be related and presented hierarchically (e.g. table of contents presenting a set of sections, with the current section expanded with the user's bookmarks from the same section). Keeping navigation hierarquies always present to the user helps solving "lost in time and space".

Related Patterns: *Timeline (2.1.4)* provides a *Timed Active Reference* related to the global content time (albeit it does not have an associated hierarchy).

Overrides: *Timed Active Reference* adds time constraints to *Active Reference* [GRS97].

2.3.3 Link Opportunities

Context: Without timing constraints, users are able to navigate through any content at their own will. However, you are developing time-based hypermedia content, you may need to specify time dependent links.

Problem: The volatile nature of time limited links easily creates stress to the end user, as link opportunities may vanish before the user's decision on following the link.

Forces:

- A time-based link must be available only within its own context.
- The user must have the ability to follow a time-based link without rush.
- Time-based links are ephemeral within the presentation timeline.
- Spatial-based links embedded into video contents can not be available all the time, as the content is changing along the timeline.

Solution: Provide large time intervals for each link opportunity. If possible, keep active each link opportunity until reaching the next one. If two intervals overlap, provide alternative ways to navigate thourgh them (e.g., menus).

Known Uses: HyperCafe [SBS96] provides temporal link opportunities in hypervideo.

Related Patterns: *Timed Active Reference (2.3.2)* provides alternative ways to follow timed links, if anchors are extracted into an auxiliar navigation structure.

2.3.4 Timewarp Linking

Context: You have created a link between two anchors in a time-based hypermedia content that will trigger a time shift.

Problem: How to update the navigation state of the link's destination content, if there is no purpose to get back to the link's source content?

Forces:

- Navigating between two contents should not trigger the presentation of both contents at the same time, if they are related to different subjects.
- Jumping to another content subject in the application should dismiss the current content presentation.
- Displaying at the same time two contents which require a significant cognitive effort should be avoided (e.g. two speech based audio tracks).

Solution: Perform a shift in the presentation's current time, stopping the playback in the source anchor context and starting the playback in the destination anchor (as seen on figure 9).

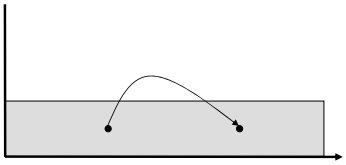

Figure 9: Timewarp Linking pattern

Related Patterns: *Montage Time Linking (2.3.5)* and *Timebreak Linking (2.3.6)* provide alternative time navigation possibilities to *Timewarp Linking*. *Content Viewport Update (2.4.5)* must be applied in presentation design time. *Context on User Event (2.5.2)* must be taken into account if the destination anchor is defined in the middle of a given content part (posing severe cognitive issues to the user, regarding content comprehension).

2.3.5 Montage Time Linking

Context: You have created a link between a content item and some complementary content, in the time-based hypermedia presentation.

Problem: How should navigation be updated if both contents are related to each other and they do not require a significant cognitive effort from the user, to capture the overall presentation?

Forces:

- A user has the ability to perceive two different content sources without cognitive overload.
- Complementary content must be present to the user, in a synchronous way, to enable content correlation from the user's cognition point of view.
- Complementary contents (such as sidenotes, list of links, images, etc.) help the user on content comprehension and further exploratory activities.

Solution: Allow an expansion on the presentation towards two distinct contents (as seen on figure 10). Both contents must be presented to the user at the same time. If the overall presentation time is changed with the introduction of the secondary content source, the user must be informed about this (as contextualization may occur from the timeline's length and current playback timing).

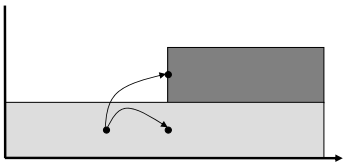

Figure 10: Montage Time Linking pattern

Related Patterns: *Timewarp Linking (2.3.4)* and *Timebreak Linking (2.3.6)* provide alternative time navigation possibilities to *Montage Time Linking*. *Multiple Channel Presentation (2.4.6)* must be taken into account, regarding cognitive overload issues. *Timed Information on Demand (2.4.4)* requires the usage of this pattern.

Overrides: *Montage* hypertext pattern [Ber98].

2.3.6 Timebreak Linking

Context: You want to create a link to some content inside the time-based hypermedia presentation, triggering a shift in the time.

Problem: How should navigation mechanisms be defined, that they will minimize the cognitive effort of presenting two contents, yet allowing the user to get back to the previous presentation's state?

Forces:

- A user wants to explore complementary content, relatively to the main content.
- A distinct path in some subject has been followed, but the user wants to get back to the main content.
- Short-term memory is quickly overloaded.
- Cognitive problems arise when a user is presented with several content sources in the same media channel.
- The presentation's timeflow may require presenting an additional content related to the current content being presented.

Solution: Pause the playback of the source content, and begin the playback of the target content. Afterwards, resume the source content (as seen on figure 11).

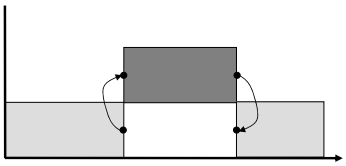

Figure 11: Timebreak Linking pattern

Related Patterns: *Montage Time Linking (2.3.5)* and *Timewarp Linking (2.3.4)* provide alternative time navigation possibilities to *Timebreak Linking*. *Context on User Event (2.5.2)* can be used when the target content playback floods the short-term memory, whether it results from long playback times or distinct content subject (if the link was activated by the user).

2.3.7 Neighbourhood Linking

Context: You want to provide time-based navigation capabilities between consecutive content parts.

Problem: How to allow time-based navigation to behave like traditional forward/back navigation mechanisms?

Forces:

- A user wants to perform a sweep browsing between different reading points.
- A user skips a portion of the content, beginning reading the next part.
- Contents that spread into a significant time interval require inner navigation mechanisms.

Solution: Provide a navigation structure related to the beginning time instance of each content part that will behave like forward/back mechanisms, through associated interaction mechanisms (such as back/forward buttons). This way, at any presentation time, the user will be able to skip the current content part or go to the previous content part (as seen on figure 12). When a content's time interval is long, navigation structures should be created in such a way that each navigation item references a part of the given content. This way, *Neighbourhood Linking* can be applied to the content accordingly.

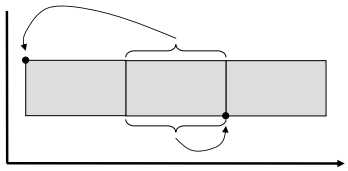

Figure 12: Neighbourhood Linking pattern

Overrides: Chains in the *Navigable Spaces* pattern [Tid99].

2.4 Presentation Design Patterns

2.4.1 Synchronization Animation

Context: You want to present to the user some visual-only content, where different parts of it are synchronized with some time-based content, at different presentation times.

Problem: How to guide the user through the content, making noticeable the time flow on the visual content, regarding the time-based content?

Forces:

- Static visual-based contents (i.e. images, text) do not have inherent time dynamics and may relate to a significant part of the timeline.
- The user is confronted with a big sized content from a time-based presentation, difficulting the task of finding which part of the content is associated with the synchronized time-based content.
- The user is distracted by some external event. When returning to pay attention to the content, a big effort is needed to understand which part of the displayed content is currently synchronized with some time-based content.

Solution: Provide visual helpers that will guide the user through the different parts of the visual content that are being synchronized along the presentation time flow. The helpers' animation results from the inherent dynamics of the time flow, making the synchronization easily perceived by users. Therefore, it will help users to quickly grasp the current playback location within the visual content.

Related Patterns: *Strict Synchronization Presentation (2.4.2)* is based on the notions of *Synchronization Animation*.

2.4.2 Strict Synchronization Presentation

Context: You are presenting two synchronized contents, strictly correlated, possibly pertaining to two expressions of the same content (e.g. a video and its subtitles, an animation and a complementary text, etc.).

Problem: How to guide the user through the synchronized content without raising issues over cognitive problems which may arise when the synchronization process starts diverging?

Forces:

- The user is paying attention to some textual content, while its audible counterpart is synchronized.
- Synchronization fails, with undetermined lag between each content source.
- You want to display a video with some complementary textual content, but keeping track of which part of the text content is associated with the current video time.

Solution: Provide lesser strict synchronization guides, that will help the user to know (with a good certainty level) which part of the content is currently being played (see examples in figure 13). In extreme cases, do not present all the associated visual content, just the essencial part (regarding the current presentation time).

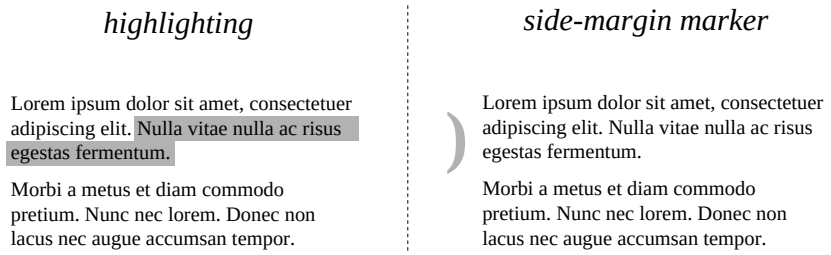

Figure 13: Strict Synchronization Presentation pattern example

Related Patterns: *Always Synchronized (2.1.2)* provides system features that minimize the needs of *Strict Synchronization Presentation*.

2.4.3 Link Opportunity Manifestation

Context: You want to present to the user a link opportunity in the time-based hypermedia presentation.

Problem: How to make perceivable to the user (in time-based mediums) that a link is nearby, currently active, or going to disappear?

Forces:

- Time-based content (such as videos or audio tracks) poses limitations to the designer regarding the presentation of link opportunities.
- Audio-based link presentation must not interfere in the main content comprehension.

Solution: Audio-based link opportunities can be made more presentable to the user through doppler effect[1] tweaking of the anchor manifestation audio resource. Its audio duration can identify the duration of the target content it refers to. On the visual medium, markers can be placed nearby the timeline, identifying a link opportunity's starting and ending points (as seen on figure 14), as described in [CG02].

Figure 14: Link Opportunity Manifestation pattern example

Related Patterns: *Link Opportunity (2.3.3)* describes timed links, and their duration. *Timeline (2.1.4)* can provide link opportunities presentation.

Overrides: *Navigational Feints* in Hypertext Patterns [Ber98].

[1] The doppler effect is the apparent change in the pitch of a sound, when the sound source's distance changes. See more information in http://en.wikipedia.org/wiki/Doppler_effect

2.4.4 Timed Information on Demand

Context: You want to present to the user a complementary content, regarding the current presentation content.

Problem: How to minimize the cognitive problems that arise when complex contents are presented at the same time, even if these are complementary?

Forces:

- The user is presented with complementary content, but its complexity poses cognitive problems in the main presentation content comprehension (e.g. a big sized textual content complementing an audio track).
- Overload in presentation channels may be verified when two contents are presented at the same time, such as presenting two videos side by side, playing two speech audio tracks, etc.

Solution: Present to the user a simplified version of the complementary content, as well as the opportunity to reach its full content with a link opportunity. Make sure that the simplified version's interference is smaller that its full content counterpart (e.g. text caption *vs.* video). Figure 15 presents an example usage scenario of this pattern, where complementary content is in a minimized state (e.g. at time instance t), and presented when a specific part of the content is reached (i.e. $t + \Delta$). When time-based complementary content must be presented, a static version must be provided in its simplified version (e.g. image for a video content, text for an audio content).

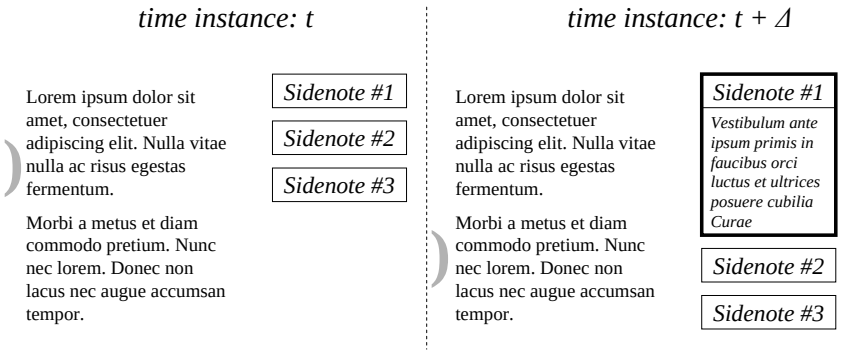

Figure 15: Timed Information on Demand pattern example

Related Patterns: *Multiple Channel Presentation (2.4.6)* limits which presentation instances need to be tweaked with *Timed Information on Demand*. *Link Opportunity Manifestation (2.4.3)* provides ways to signal the user of complementary content availability. *Montage Time Linking (2.3.5)* implements the basic behaviour for *Timed Information on Demand*.

Overrides: *Information on Demand* [GRS97] does not cope with time related issues.

2.4.5 Content Viewport Update

Context: A visual based content is presented to the user and synchronized with some other time based content, after a time related link has been triggered.

Problem: Arriving to a link destination in a time-based hypermedia application may pose a significant cognitive effort on the user, caused by time flow dynamics and lack of surrounding context helpers.

Forces:

- The user follows a link and is presented with new content, somehow related to the destination content's context.
- Visual identification of a given content has a strong affection on user's content perception.
- Long and complex content visual presentation does not fit in a typical display device (e.g. text flows displayed on a screen with limited size).

Solution: Provide automatic content display, coupled with some presentation of its surrounding visual contents (e.g. while presenting an item from a list of paragraphs, displaying the previous paragraph should help on contextualization). Designers must be cautious on surrounding contents display, regarding overloaded cognitive efforts. Figure 16 presents an example where this pattern can be applied: a video is being presented, with complementary images of important screencaps of the video. The current image (*Image n*)is emphasized, but its siblings are also displayed (*Image n-1* and *Image n+1*).

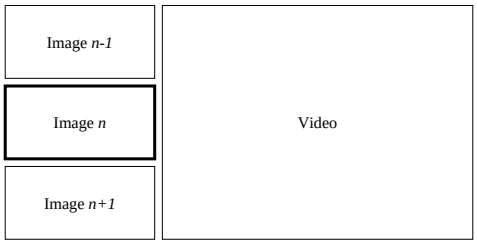

Figure 16: Content Viewport Update pattern example

Related Patterns: *Context on User Event (2.5.2)* can be helpful regarding content comprehension and provide an alternative solution to *Content Viewport Update*. Siblings display must be carefully crafted, taking into account the issues raised in *Multiple Channel Presentation (2.4.6)*. *Neighbourhood Linking (2.3.7)* provides navigation mechanisms with the required semantics for establishing sibling relations.

2.4.6 Multiple Channel Presentation

Context: You want to present simultaneously two multimedia sources to the user, as a way to provide more information within a time frame.

Problem: How to avoid possible cognitive overloads that may appear when playing two or more contents?

Forces:

- Image and video usually require small cognitive effort.
- Text and voice usually require more cognitive effort.
- The user can acquire more content and learn more information if multiple related content sources are presented, in a synchronous way.
- Some media sources can not be composed, as they overload a user's cognitive resources.

Solution: Present multiple content sources to the user as long as the cognitive effort required to capture each content does not create cognitive pressure. Each media type requires different efforts, thus being reflected in media composition through different channels (e.g. aural, visual, etc.). [Mar02] presents the correctness of media type combinations (see Table 1) as *Ok* if the combination is easy for a normal user, *Wrong* if the combination is hard for a normal user, and *Subject* if the combination is easy if both items' subjects are closely related, and hard otherwise, as follows:

	Sound Icon	Music	Speech	Video	Image	Text
Text	*Ok*	*Ok*	*Subject*	*Subject*	*Ok*	*Ok*
Image	*Ok*	*Ok*	*Ok*	*Ok*	*Ok*	–
Video	*Ok*	*Ok*	*Subject*	*Wrong*	–	–
Speech	*Ok*	*Ok*	*Wrong*	–	–	–
Music	*Ok*	*Wrong*	–	–	–	–
Sound Icon	*Wrong*	–	–	–	–	–

Table 1: Correctness of media type combinations

Related Patterns: The time-based hypermedia system must have the *Time Composition (2.1.1)* and *Always Synchronized (2.1.2)* patterns implemented.

2.5 Interaction Design Patterns

2.5.1 Direct Information Access

Context: You want to provide interactivity in a time-based hypermedia presentation that has several contents being displayed and synchronized.

Problem: How can content be reached independently from the navigation structures provided by the designer, as a way to minimize the cognitive effort required to trigger common navigation tasks?

Forces:

- The user does not want to waste time on interaction processes with the presentation.
- Time-related cognitive pressure requires quick interaction processes.
- Common navigation tasks should require little cognitive effort from the user, as content perception must be the user's main task.

Solution: Provide standardized user interaction tasks (e.g. following link opportunities, skipping some content, etc.). The user must also be able to jump directly into any part of the content that can be displayed outside the current timeframe (e.g. a scrollbar related to an already playing audio track). Such methods can be implemented, e.g., through voice activated commands, simple keyboard shortcuts, or plain mouse clicks over the content (instead of using traditional navigation mechanisms to reach some content, such as table of contents).

Related Patterns: *Link Opportunity (2.3.3)* interaction can benefit if the *Direct Information Access* pattern is applied to the presentation.

2.5.2 Context on User Event

Context: You have provided to the user powerful navigation possibilities, such as direct content navigation or timeline based navigation, allowing the user to jump directly into a random part of a content (e.g. middle of an audio track, middle of a paragraph), instead of jumping through the application's provided navigation structures.

Problem: When jumping into a random location in the timeline, the user may be confused by the content's presentation, as it may appear out of context (e.g. middle of a text paragraph, end of a spoken phrase on an audio track, etc.).

Forces:

- The ability to navigate to specific points in a time-based hypermedia presentation allows the user to jump with more precision into some content.
- A user-defined bookmark or annotation is referring to a specific content location. Navigating to those specific points imposes recontextualization from the user (as it may not be attached to the beginning of the content's context scope).

Solution: Provide the user enough content to enable contextualization (e.g. a whole text phrase or paragraph, an overview of a full picture, rewind a time-based content a few seconds, etc.). This can be done by rewinding the presentation timeline towards the initial time of the content scope. Nevertheless, interaction mechanisms must be provided to enable the user to go to the dismissed content location, if desired.

Related Patterns: *Content Viewport Update (2.4.5)* must be applied if *Context on User Event* is used.

3 Summary

Time-based hypermedia contents are inherently complex. Users are targeted with different media sources, composed and presented in different ways, with rich navigation features. As developing systems that support this type of applications is difficult, as well as creating time-based hypermedia applications, a pattern system that supports all these variables, which can be applied at different levels, helps maximizing user experience. Table 2 presents a summary of pattern relations (where each relation was further described previously in *Related Patterns* sections). These relationships can be characterized either as complementary (marked with the symbol •), or exclusive (marked with the symbol ◦).

	Time Composition	Always Synchronized	Quick Response	Timeline	Content Decoupling	Multiple Content Sources	Integrated Navigational Contexts	Timed Active Reference	Link Opportunity	Timewarp Linking	Montage Time Linking	Timebreak Linking	Neighbourhood Linking	Strict Synchronization Presentation	Synchronization Animation	Link Opportunity Manifestation	Timed Information on Demand	Content Viewport Update	Mutiple Channel Presentation	Direct Information Access	Context on User Event
Time Composition																					
Always Synchronized	•														•						
Quick Response									•							•					
Timeline								•													
Content Decoupling																					
Multiple Content Sources																			•		
Integrated Navigational Contexts					•					•	•	•							•		
Timed Active Reference				•																	
Link Opportunity								◦													
Timewarp Linking											◦	◦						•			•
Montage Time Linking										◦		◦					•		•		
Timebreak Linking										◦	◦										•
Neighbourhood Linking																					
Synchronization Animation															•						
Strict Synchronization Presentation		•																			
Link Opportunity Manifestation									•												
Timed Information on Demand											•					•			•		
Content Viewport Update													•						•		•
Multiple Channel Presentation	•	•																			
Direct Information Access									•												
Context on User Event																		•			

Table 2: Summary of pattern relations

References

[AIS+77] Christopher Alexander, Sara Ishikawa, Murray Silverstein, Max Jacobson, Ingrid Fiksdal-King, and Shlomo Angel. *A Pattern Language*. Oxford University Press, 1977.

[All83] J. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26:832–843, 1983.

[Aro91] Barry Arons. Hyperspeech: navigating in speech-only hypermedia. In *HYPERTEXT '91: Proceedings of the third annual ACM conference on Hypertext*, pages 133–146, New York, NY, USA, 1991. ACM Press.

[Ber98] Mark Bernstein. Patterns of hypertext. In *Hypertext 98*, pages 21–29, Pittsburgh, PA, 1998. ACM.

[BKM99] Hubert Baumeister, Nora Koch, and Luis Mandel. Towards a UML Extension for Hypermedia Design. In *Lecture Notes in Computer Science*, volume 1723, pages 614 – 629. Springer-Verlag, 1999.

[CDL+05] Luís Carriço, Carlos Duarte, Rui Lopes, Miguel Rodrigues, and Nuno Guimarães. Building Rich User Interfaces for Digital Talking Books. In *Computer-Aided Design of User Interfaces IV*. Kluwer Academic Publishers, 2005.

[CG02] Teresa Chambel and Nuno Guimarães. Context perception in video-based hypermedia spaces. In *HYPERTEXT '02: Proceedings of the thirteenth ACM conference on Hypertext and hypermedia*, pages 85–94, New York, NY, USA, 2002. ACM Press.

[CGD+03] Luís Carriço, Nuno Guimarães, Carlos Duarte, Teresa Chambel, and Hugo Simões. Spoken books: Multimodal interaction and information repurposing. In *Proceedings of HCII'2003, International Conference on Human-Computer Interaction*, pages 680–684, 2003.

[CL98] Jakob Cybulski and Tanya Linden. Composing multimedia artifacts for reuse. *Proceedings of PLoP'98: Pattern Languages of Programs'98*, 1998.

[DC06] Carlos Duarte and Luís Carriço. A conceptual framework for developing adaptive multimodal applications. In *IUI '06: Proceedings of the 11th international conference on Intelligent user interface*, Sydney, Australia, 2006. ACM Press.

[EH99] Deborah M. Edwards and Lynda Hardman. *Lost in hyperspace: cognitive mapping and navigation in a hypertext environment*, pages 90–105. Intellect Books, Exeter, UK, UK, 1999.

[GHJV95] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns*. Addison-Wesley, 1995.

[GRS97] A. Garrido, G. Rossi, and D. Schwabe. Patterns systems for hypermedia, 1997.

[HBR94] Lynda Hardman, Dick C. A. Bulterman, and Guido Rossum. The Amsterdam hypermedia model: adding time and context to the Dexter model. *Commun. ACM*, 37(2):50–62, 1994.

[HS94] Frank Halasz and Mayer Schwartz. The dexter hypertext reference model. *Commun. ACM*, 37(2):30–39, 1994.

[ISB95] Tomas Isakowitz, Edward A. Stohr, and P. Balasubramanian. Rmm: a methodology for structured hypermedia design. *Commun. ACM*, 38(8):34–44, 1995.

[Jef01] Jeff Ayars, Dick Bulterman, *et al.* Synchronized Multimedia Integration Language (SMIL 2.0). W3C Recommendation, 2001.
http://www.w3.org/TR/SMIL2.

[Lop05] Rui Lopes. Ditabbu: Digital talking books builder. Technical report, Faculty of Sciences, University of Lisbon, 2005.

[Mar02] Oscar Rosell Martinez. Design dependencies within the automatic generation of hypermedia presentations. Technical report, "CWI", 2002.

[MD96] Gerard Meszaros and Jim Doble. A Pattern Language for Pattern Writing, 1996. http://hillside.net/patterns/writing/patterns.htm.

[PLL95] M. J. Pérez-Luque and T. D. C. Little. A temporal reference framework for multimedia synchronization. In *Proceedings of the International Workshop Multimedia Synchronization*, Tysons Corner, Virginia, May 1995.

[RSG97] Gustavo Rossi, Daniel Schwabe, and Alejandra Garrido. Design reuse in hypermedia applications development. *ACM Hypertext Conference 1997*, 1997.

[SBS96] Nitin Sawhney, David Balcom, and Ian Smith. Hypercafe: narrative and aesthetic properties of hypervideo. In *HYPERTEXT '96: Proceedings of the the seventh ACM conference on Hypertext*, pages 1–10, New York, NY, USA, 1996. ACM Press.

[Sch95] Douglas C. Schmidt. Using design patterns to develop reusable object-oriented communication software. *Communications of the ACM*, 38(10):65–74, Oct. 1995.

[SRB96] Daniel Schwabe, Gustavo Rossi, and Simone D. J. Barbosa. Systematic hypermedia application design with oohdm. In *HYPERTEXT '96: Proceedings of the the seventh ACM conference on Hypertext*, pages 116–128, New York, NY, USA, 1996. ACM Press.

[Tid99] Jenifer Tidwell. COMMON GROUND: A Pattern Language for Human-Computer Interface Design, 1999.
http://www.mit.edu/ jtidwell/common_ground.html.