Composing Multimedia Artefacts for Reuse

Jacob L. Cybulski and *Tanya Linden*
j.cybulski@dis.unimelb.edu.au *t.linden@dis.unimelb.edu.au*
http://www.dis.unimelb.edu.au/staff/jacob *http://www.dis.unimelb.edu.au/staff/tanya*
ph: +61 3 9344 9244 *ph: +61 3 9344 9250*
fax: +61 3 9349 4596 *fax: +61 3 9349 4596*

Department of Information Systems
University of Melbourne
Parkville, Vic 3052, Australia

Abstract

This paper describes a pattern language that is used to define a multimedia authoring environment capable of producing and utilising multimedia components. We believe that the effective construction of multimedia material should refrain from the common practice of building new documents and presentations entirely from scratch. Hence, the proposed pattern language emphasises the process of making new multimedia artefacts by reusing the existing components to suit the requirements of new applications. Each of the presented patterns describes one well-known approach to multimedia authoring, e.g. joining and breaking artefact groups, defining and filling in templates, arranging and re-arranging artefact collections, creating and holding presentations, synchronising multiple multimedia channels, etc. The paper also provides some insight as to the direction of the pattern language development, i.e. it lists six dimensions of multimedia authoring and reuse and shows what areas of functionality the final form of the pattern language must ultimately address.

1. Introduction

Multimedia product is a software system or an information environment that uses computer technology to integrate text, graphics, images, video and audio to deliver information. In addition, interactive multimedia allows its user to communicate with the presentation system to control delivery of multimedia information. Recent developments in multimedia cover such issues as multimedia authoring, presentation and collaboration, approaches to structuring, representation and storage of multimedia information, hardware and software technology to support multimedia transfer and delivery, and more [5].

Our multimedia project, Teacher's MATE (Multimedia-Assisted Teaching Environment) [3], adds yet another aspect to the field of multimedia research. In our work, we focused on the construction, organisation and management of *reusable multimedia components*. In our project, we proposed new methods and techniques for handling multimedia and we constructed tools that assist the users of multimedia authoring systems to more effectively identify, represent, generalise, classify, store, search and retrieve, select, adapt and integrate multimedia components and processes that manipulate them.

In our research, as it clearly shows in the remainder of this article, we have adopted terminology that is a characteristic mix of multimedia and software reuse vocabulary. Therefore, we refer to reusable multimedia components and processes as *artefacts*. Some artefacts have proprietary format inaccessible to our authoring and reuse tools, and hence, they require special software for their creation and modification. We call such artefacts *atomic*. Existing multimedia components can be aggregated with the use of our repository services into larger *composite* artefacts. Artefacts are held together in a composite artefact with the assistance of *glue*, i.e. special data structures and processes

that define the layout and the organisation of components in their container. Reusable multimedia components can be organised into elaborate *arrangements* and *presentations* that can subsequently be delivered to the user over one or more *synchronised* communication *channels*.

1.1 Atomic Artefacts

There are many different atomic artefact types that can be used in the construction of multimedia documents and presentations, e.g.

- text document (in Mac, PC and Unix text, HTML, RTF, PostScript),
- sound (e.g. WAVE, MIDI, RealAudio, AU),
- bitmap (e.g. GIF, JPEG, TIFF),
- picture (e.g. Windows Meta Files or Mac Pictures),
- animation (e.g. animated GIFs, QuickTime, MPEG),
- video (e.g. MPEG, QuickTime, RealMedia, VDO, AVI),
- scriptlet (e.g. JavaScript, HyperCard, Tcl/Tk, Macromedia, Perl/Tk),
- applet (written in Java or ActiveX),
- control (e.g. buttons, sliders, options), etc.

As these artefacts cannot be directly created and modified by our repository services, we have to utilise special purpose software to do so. Although the artefacts of different types require different kinds of authoring software, the similarities of services provided call upon the utilisation of *multimedia authoring idioms* in their implementation.

From the reuse perspective, the most important idioms result in the creation of new artefacts and in the alteration of exiting artefacts and their parts. Such idioms look at the aspects of:

- Creating an atomic artefact of a given type
- Destroying an existing atomic artefact
- Combining a number of components into a new atomic artefact
- Combining an atomic component with the contents of another
- Splitting an atomic artefact into a number of fragments
- Extracting a smaller part of a large component
- Removing a part of a large component
- Erasing a part of a component
- Rearranging component parts

We also need to show and play audio-visual components, as well as to query their properties. This leads us to a number of additional multimedia authoring idioms, e.g.

- Determining a set of attributes appropriate for a given atom type
- Modifying an attribute of an artefact or its part
- Checking for an attribute value of an artefact or its part
- Presenting an atomic component
- Controlling and interacting with an atomic component

The authoring idioms lay the details of possible actions, operations and transformations applicable to all kinds of multimedia artefacts. They provide a useful abstraction for multimedia artefacts regardless of differences in their media, representation and function. They define the multimedia authoring terminology and they specify a virtual machine for the multimedia component construction, editing and presentation. The idioms, however, do not offer any insight into the relationships that artefacts of different media types could form, the groups they may be part of, the approaches to their composition and the methods of their reuse. These issues can only be addressed in respect to the composite multimedia artefacts that are described in detail in the next section.

1.2 Composite Artefacts

There exist a rich selection of the off-the-shelf packages that can be used to produce sophisticated graphics, sound, animation and video, e.g. Adobe PhotoShop, Mixman Technologies' Mixman, Apple QuickTime, and Ulead Media Studio respectively. The effective multimedia production, however, can only be achieved, when the multimedia objects generated by such packages can subsequently be placed into new documents or presentations, and hence, combined into larger groups of multimedia objects. Many of the commercial software products allow their users to group and alter components of mixed media either directly or by means of various software integration and communication utilities, e.g. Microsoft OLE, Apple OpenDoc or CORBA. Products such as Adobe Illustrator, CorelDraw or Microsoft FrontPage and PowerPoint are at the forefront of multimedia contenders, still, only a few of them consider reusability of their multimedia artefacts as an essential aspect of their functionality. Some address the issue of reuse only in a very narrow sense, e.g. by the use of clipart, document templates or search engines. However, any software tool that claims to embrace reusability principles must also take into consideration many other aspects of component processing which can be structured into three facets of multimedia reuse [2]:

i. *analysis of existing multimedia artefacts*, which includes identification and extraction of reusable components from existing multimedia documents, understanding reusable components by their documentation and representation, and their generalisation to widen the scope of their applicability to new multimedia products;

ii. *organisation of multimedia artefacts in the reuse repository*, which involves classification and indexing of multimedia components, their storage, search and retrieval of candidate components for the inclusion in new multimedia products; and finally,

iii. *synthesis of new multimedia products*, which consists in the selection of multimedia components from the set of retrieved candidates, modification and adaptation of artefacts, and finally their composition and integration into new multimedia products.

Some search engines, such as Yahoo Image Surfer, collect and index large collections of images. Web management tools, e.g. FrontPage, define reusable HTML templates, allow creation and management of reusable WWW links and documents. However, none of these tools addresses multimedia reuse in a comprehensive way.

Other tools, such as those produced as the result of MelbourneIT's cMILE project [4] or Melbourne University Teacher's MATE project [3], allow to create, describe, index, compose and manage educational course material.

In our multimedia-reuse project, we investigated these aspects and we proposed ways of dealing with these issues in an effective manner. In doing so, we defined a collection of *multimedia-reuse patterns* that address six dimensions of multimedia authoring and reuse (Cf. Figure 1), i.e. the contents and quality of artefacts, their arrangement and presentation, and the processes leading to their construction and utilisation. The resulting pattern language is currently used as a guide to the definition and implementation of a multimedia-authoring environment that actively supports reuse of multimedia components.

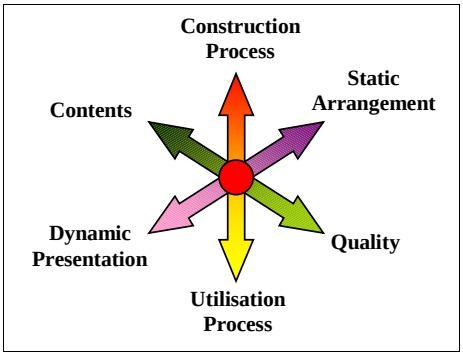


Figure 1 - Dimensions of multimedia authoring and reuse

The patterns presented in this article address only the issues that we considered as the most urgent for the builders of a multimedia reuse tool, i.e. artefact construction, contents, arrangement and presentation. The issues of quality

and utilisation of multimedia artefacts have not been fully explored at this point of time. Nevertheless, we have identified a number of concerns that must be investigated before we finalise a thorough and complete pattern language for multimedia authoring and reuse (for further discussion, see the summary section of this paper).

2. Multimedia-Reuse Pattern Language

In this section, we describe 5 patterns, which constitute a small part of a pattern language aiming at supporting the construction of reusable multimedia artefacts (Cf. Table 1).

Pattern Language	**Contents**	**Quality**	**Construction**	**Utilisation**	**Arrangement**	**Presentation**
Glue	X		X			
Template	X	X	X	X	X	
Components Layout			X	X	X	
Define And Run Presentation	X		X	X		X
Synchronise Channels			X	X		X

Table 1 - Multimedia reuse pattern language vs. six dimensions of multimedia reuse

The first, i.e. `Glue`, is concerned with defining the contents of composite artefacts, their construction and destruction. This pattern introduces the concept of *glue* that enables joining sub-components based on their physical, logical and temporal properties. Glue can be used as a mechanism for many other operations on composites, e.g. addition and removal of components (unglue, add and glue again), union and intersection of collections (unglue, union or intersection followed by glue), attaching semantic attributes to artefacts (by gluing these two kinds of information artefacts), etc.

The `Template` pattern defines reusable *templates* containing *gaps* that can be filled in with other components. This pattern specifies the methods of template creation and use. Templates can also be used to define controls (as collections of possible gap fillers), forms (as a template with textual gaps), hot links between documents (when the gap is filled in with components of another composite artefacts), or hyperlink (a navigable hot link).

`Define And Run Presentation`, defines a notion of a delivery channel or a group of parallel channels through which the artefacts are sent for display. If these channels are connected to the gaps of the predefined template, this pattern should be used in conjunction with `Template`.

`Synchronise Channels` is a pattern that describes temporal relationships between artefacts sent through parallel channels. The synchronisation may be responsible for starting and stopping video or sound, adjusting of temporal position, changing the presentation speed, etc.

`Components Layout` describes audio-visual relationships between components. It defines the order of displayed artefacts and the ordering method. Different methods of ordering artefacts will determine the look and behaviour of displayed artefacts, e.g. occlusion order may allow one artefact to be hidden by another, XOR, AND and OR ordering can define bitmap operations on the overlapping artefacts, partial orders may enforce artefact collision on attempted overlay, etc.

There are many other patterns that extend this subset of multimedia reuse pattern language. Some are listed in the Table 2, others have been explored by researchers in multimedia, e.g. the `Navigation Observer` pattern by Rossi, Garrido and Carvalho [6], Interaction Design Patterns by Tidwell [7], etc.

2.1 Glue

Problem Description: There is a need to join a number of multimedia artefacts into a single composite artefact.

Pre-condition: The artefacts should be designed in such a way so that their physical, logical, or temporal properties match when joined.

Post-condition: The result is a composite artefact. All joined artefacts become the components of the composite artefact but they can be easily decoupled.

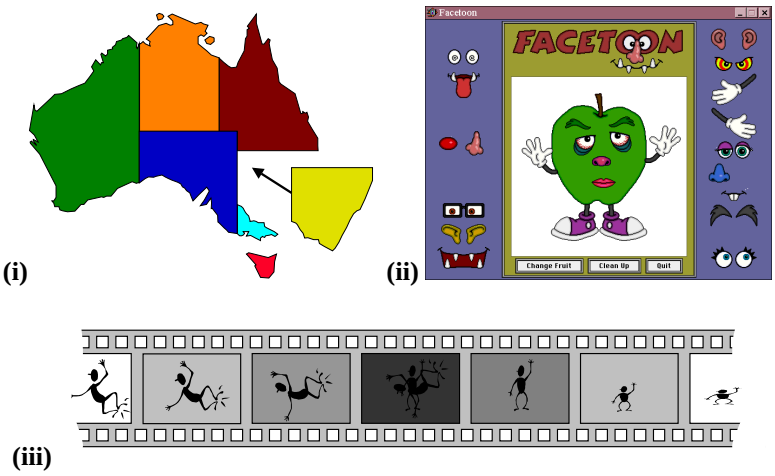


Figure 2 - Composite artefact "glued" to match their physical (i), logical (ii) and temporal properties (iii).

Solution: Put the artefacts together and apply special *glue* to join them into a composite object. The glue also determines which of the artefact properties are needed to match for the components to hold together. If there is a need to extract a component out of a composite artefact, just remove the glue and the components will be available for extraction.

When to use this pattern:

- ✓ to create a composite image based on the physical properties of its components (e.g. a jigsaw puzzle or a map, see Figure 2-i).
- ✓ to create a composite image based on the logical properties of its components (e.g. photo-fit, see Figure 2-ii).
- ✓ to create a synchronised composite sound or video based on the temporal properties of their components (e.g. joining video sequences with fading effect, see Figure 2-iii).
- ✓ to extract a video frame from the video sequence (see Figure 2-iii).
- ✓ to extract a component from a composite image (e.g. from a jigsaw puzzle, see Figure 3).

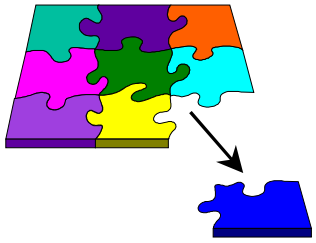
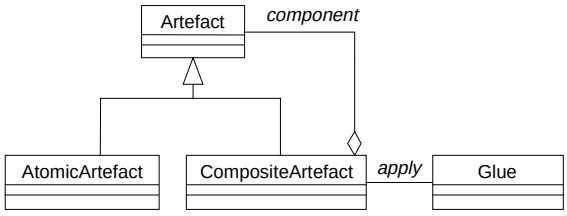


Figure 3 - Extracting the required component from the composite artefact.

Sample structure:



Examples: Different types of glue can be found in many existing software products. In some graphic editors, joining of artefacts can be achieved with special authoring operations, such as “Group”, “Insert”, “Join” or “Link-and-Embed”, examples of which follow.

– Grouping is often used in Microsoft products (e.g. PowerPoint and Word) for creating composite images. “Ungroup” operation as opposed to “group” is an act of removing glue and making components of composite images available for selection, cutting or copying.

– While grouping of components creates a new composite object, insertion of components into an artefact implies that the artefact itself is a composition of its parts. For example, we can insert an image into a word processing document, a header file into a C program, a video into a PowerPoint slide, etc.

– Join can be formed with a special bonding object, e.g. drawing canvas, HTML or JavaScript, Java applet, hyperlink, video stream or web browser frames.

– In all previous examples, the component becomes dependent on the composite artefact containing it, in a sense that to be modified the component should be extracted, edited and subsequently re-inserted. In linking-and-embedding the components maintain their independence. They can be part of several composites. If such a component is altered, the change is immediately effective in all composites.

Extracting of components is easy if a component can be selected and copied while being part of another component, which is often possible with text or image type components. Sometimes, however, to select a component from the composite artefact, the artefact has to be ungrouped first (the case of glue removal as it was mentioned earlier).

Pro’s:

- Glue allows applying certain operations to the artefact as a whole, e.g. resizing, deleting, changing colours.
- Glue facilitates matching of components properties.
- Glue provides a way of encapsulating components.

Con’s depend on the type of glue:

- Altering the actual composition may be more difficult.
- Modifying attributes of individual components may prove to be difficult in certain cases, e.g. in some packages, to change the colour of an image component or to modify text within the composite artefact, glue has to be removed first (by ungrouping the artefact).
- Glue may introduce components coupling, e.g. if a component is linked and embedded into several artefacts, its modification in one composite may cause unwanted changes in other composites.
- The semantics of a component in a group may differ from that of the same artefact out of context, e.g. text inserted into a word processing document may change its meaning; added textual description of an image may alter its interpretation.
- Glue may cause the necessity to change some of the physical properties of artefacts. For example, addition of text to a word processing document may require massive reformatting; when joining several images, the colour scheme of some of the components may have to be adjusted to match the general colour scheme.

Comments: Note that `Glue` can be responsible for a wide range of useful behaviour, e.g. addition and removal of sub-components, extraction of a subset of sub-components, union and intersection of composites, etc.

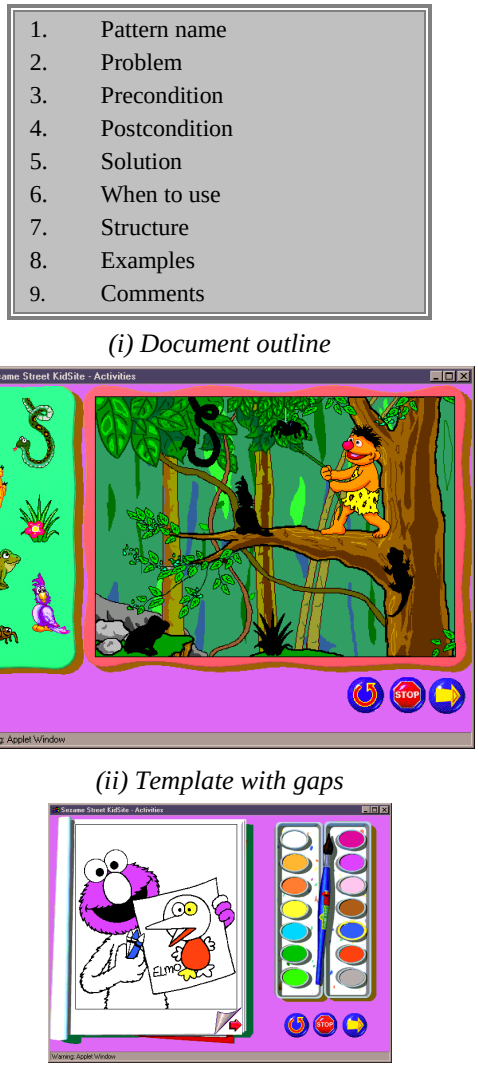
2.2 Template

Problem Description: There is a need to produce a collection of composite artefacts similar in structure and contents.

Pre-condition: There exists a class of similar composite artefacts.

Post-condition: A composite artefact representing the generalisation of similar composites is constructed (Cf. Figure 4). It is then used to create a set of instances, each of which includes all shared components and components specific to the instance.

Solution: Define a special kind of a composite artefact, called a *template*, which contains all shared components and *gap* components. A template can then be used to generate new instances by filling in or replacing the gaps with non-common artefacts.



(i) Document outline

(ii) Template with gaps

(iii) Template with changeable attributes

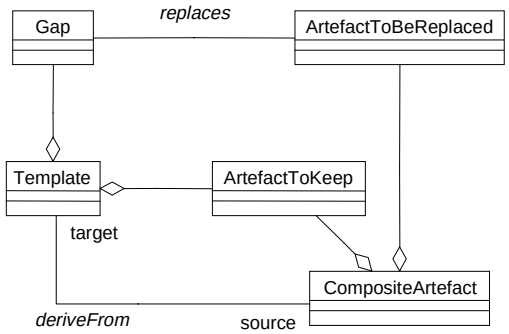
Figure 4 - Different forms of document templates

When to use this pattern:

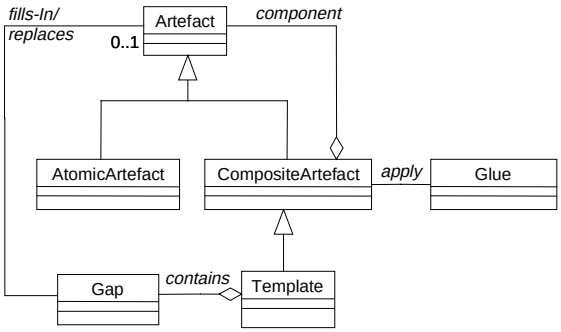
- ✓ When a collection of legacy composites is to be generalised for the reuse purposes.
- ✓ When a multimedia template is to be derived from its prototype.
- ✓ When a template is designed in the participatory process that results in multiple alternatives.
- ✓ When creating a document standard for the collection of similar documents (Cf. Figure 4-i).
- ✓ When defining a form to be filled in with other documents and components (Cf. Figure 4-ii).
- ✓ When creating a graphical object that could be customised by filling in missing components (Cf. Figure 4-ii) or by setting the properties of its partially defined components (Cf. Figure 4-iii).
- ✓ When establishing a hot link or a view between a template and another artefact, e.g. an OLE link.
- ✓ When creating puzzles and games in which components have to match gaps in a template (Cf. Figure 4-ii).

Examples: Templates are commonly used in the processing of both text and graphics. They can be used to define web document skeletons, document outlines and masters (Cf. Figure 4-i), and partially defined graphic objects (Cf. Figure 4-ii, iii) - all such multimedia documents that can be easily adopted and adapted to the new multimedia application. A presentation/slide layout that proved to be effective can be turned into a reusable presentation/slide template. A reusable system documentation template may be designed by giving an example of such a document first and then generalising it by removing the systems-specific information.

Sample structure for creating templates:



Sample structure for creating instances:



Note that in both sample structures, a template is a composite artefact that can be derived in the process of generalising pre-existing composite artefacts.

Related Patterns: `Synchronise Channels`.

Pro's:

- Artefacts derived from the same template are visually consistent.
- A template generalises a collection of composites.
- A template groups common reusable components and puts them in one location, which is a single point for future reference, modification and reuse.
- Gap artefacts preserve the integrity of a collection of artefacts after the removal of variable components.
- Gaps allow live association (like hot links) between two components, e.g. OLE links.
- Gaps allow parametrisation of composite artefacts.

Con's:

- Certain types of glue may require the construction of gaps that match the property of the common components.
- Some types of multimedia artefacts have "built-in" gaps that need to be referenced in a unique and consistent fashion, e.g. Windows, XOR groups, views.

Comments: A template can be derived from a base artefact that is used as a prototype for a collection of similar artefacts. Alternatively a template can be produced from scratch.

The solution suggests three ways of creating instances from a template:

- by *filling-in* gaps with artefacts by association;
- by *replacing* gaps with artefacts;
- by *changing properties* of common artefacts.

This pattern is used to create templates for generalisation and for creation of static collections of multimedia artefacts. This is in contrast with the `Synchronise Channels` pattern that allows the creation of artefacts synchronised in time.

2.3 Define And Run Presentation

Problem Description: An organised collection of artefacts needs to be shown in sequence to the user.

Pre-condition: All artefacts are organised according to a partial order, e.g. hierarchical or network.

Post-condition: A selection of multimedia artefacts that match their logical and temporal properties is serialised so that they could be shown to the user in sequence.

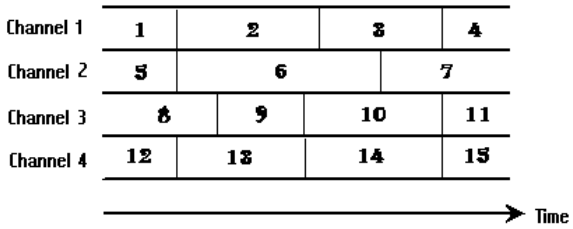


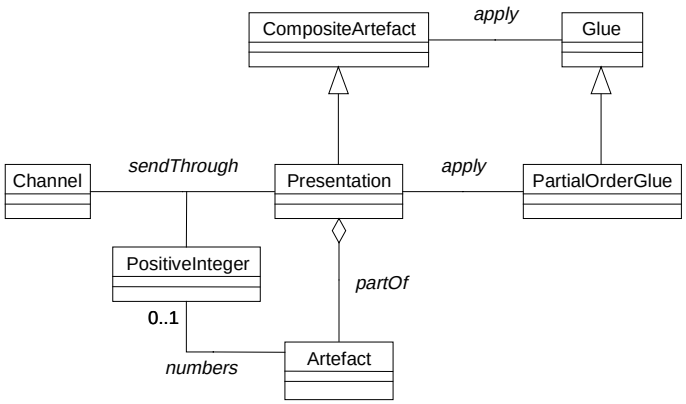
Figure 5 - Various components traveling through channels over time

Solution: Plan the presentation of multimedia artefacts by organising them into a partial order using a special glue. Define a delivery channel that will carry the multimedia message as a series of artefacts. The choice of presented artefacts can be made either during the planning or delivery stage. The delivery can happen via a single channel or multiple channels.

When to use this pattern:

- ✓ When a collection of multimedia artefacts needs to be shown to the user in sequence one at the time.
- ✓ When groups of multimedia artefacts need to be shown to the user simultaneously, but the multiple series are asynchronous (so that they cannot be shown in a single channel presentation of a series of composites).

Sample structure:



Note that in this sample structure the presentation is defined as an ordered composite artefact, which is composed of atomic and composite components.

Example: A slide presentation can be given as a series of visual artefacts sent across a single presentation channel. Graphics can be overlaid with sound in two-channel presentation. A template can be utilised to design multi-channel presentation of multimedia contents (one channel per gap).

Related Patterns: This pattern can be used in conjunction with the `Synchronise Channels` and `Template` patterns, e.g. to deliver a multi-frame presentation of synchronised multimedia components by attaching individual channels to template gaps.

Pro's:

- This patterns allows to present a collection of artefacts that otherwise cannot be viewed simultaneously.
- The serialisation process may be controlled by either the presenter or the viewer.
- The order of artefacts can be defined either at the design or presentation time.
- The pattern does not impose any restrictions on the use of multiple channels whether synchronous or asynchronous.

Con's:

- Placing artefacts in one presentation may require changes to their properties, e.g. the audio fading affect may have to be used to facilitate smooth transition between two sound artefacts.
- This pattern does not take into consideration the temporal requirements of different types of artefacts, e.g. video artefacts may be played while being sent (stream) or they may start playing only after the whole artefact arrives.
- The presenter has restricted control over the viewer's side of a channel, e.g. the presenter may send and play a video artefact but there is no feedback on its progress or control over it.

Comment: Note that since the presentation is defined as a composite artefact, it is also possible to parametrise presentations (in temporal domain) with the use of *temporal templates* and gaps.

A partial order between artefacts may be imposed by the physical order of artefacts, hyperlinks or random access indices.

Multiple parallel channels may be used to deliver truly multimedia presentations, where artefacts of different media types are shown to the user at the same time. (Cf. Figure 5)

The content can be dynamically created during the presentation.

2.4 Synchronise Channels

Problem Description: A number of continuous artefacts (such as video or sound) and non-continuous artefacts (such as text or images) sent through multiple channels need to be synchronised into a multi-channel presentation.

Pre-condition: Artefacts are sent along parallel channels. Each continuous artefact defines a sequence of temporal positions. Temporal properties, such as playing speed or position within an artefact, can be modified.

Post-condition: The change to an artefact's temporal property in one channel is reflected in temporal properties of artefacts in other channels. (Cf. Figure 6)

Solution: For non-continuous artefacts define *delay* artefacts. Delays define the sequence of temporal positions within the composites of continuous and non-continuous artefacts. The solution supports multiway indexing [1].

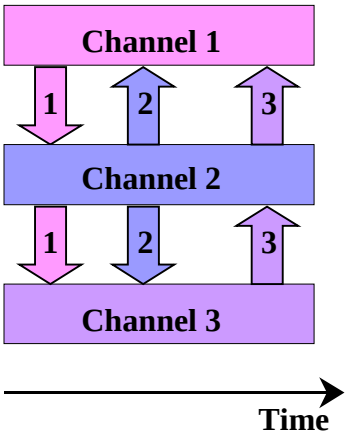


Figure 6 - Synchronising a set of temporal artefacts using a position from any other artefact as a reference.

– Create indexes of sequences and store them as a *synchronisation glue*, i.e. a table of temporal cross-references. The table is used as a look-up to adjust artefacts positions in every channel to the position of the selected artefact.

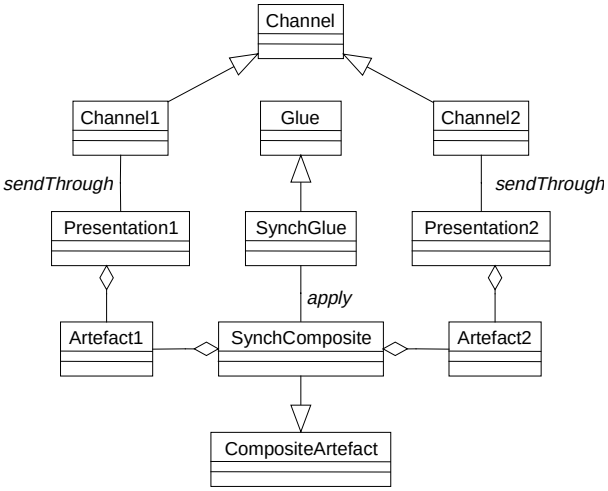
– Alternatively, mapping functions can be used to adjust the artefacts positions.

– Synchronised artefacts form a *synchronisation composite*.

When to use this pattern:

✓ When multichannel parallel presentations need to be synchronised, e.g. text with graphics, voice with video, etc.

Sample structure:



Example:

– Synchronisation of voice with the text to be annotated.

– Synchronisation of voice and graphics to allow narrative of graphical presentations.

Related Patterns: `Define And Run Presentation`.

Pro's:

- In a multiple channel presentation displayed artefacts can be synchronised.
- The presentation can be interactive by providing viewers and presenters with control over channels and feedback on the presentation events, e.g. one channel can be used for the interaction with the user and the remaining channels could provide feedback.
- Continuous and non-continuous artefacts can be presented and synchronised in the uniform fashion.
- This pattern deals with temporal properties of continuous artefacts and introduces the time dimension for non-continuous artefacts.

Con's:

- This pattern may deal only with those continuous artefacts that allow for the modification of their temporal properties. For example, if we have a continuous artefact that cannot be controlled (e.g. certain cases with sound or video), it cannot be synchronised with other artefacts.

- The introduction of time dimension and interaction significantly increases the complexity in the planning and delivery of presentations.

Comments: Note that synchronisation may be necessary in two cases:

– on the boundary of two artefacts, e.g. when voice annotates images the first image should be delayed until the first voice annotation completes and the second image is displayed simultaneously with the second voice annotation;

– within a continuous artefact, e.g. in case of video and sound being sent through separate channels.

2.5 Components Layout

Problem Description: Several artefacts need to be arranged in respect to their audio-visual properties.

Pre-condition: All artefacts have to be visualised or played and their relative positions can be established.

Post-condition: Multimedia artefacts are placed in their respective positions or they are moved to new positions (Cf. Figure 7).

Solution: Compose the artefacts into a *layout*, arrange them with the use of *arrangement* glue to associate each component with its *position* within the layout. Artefact positions can be altered in three ways:

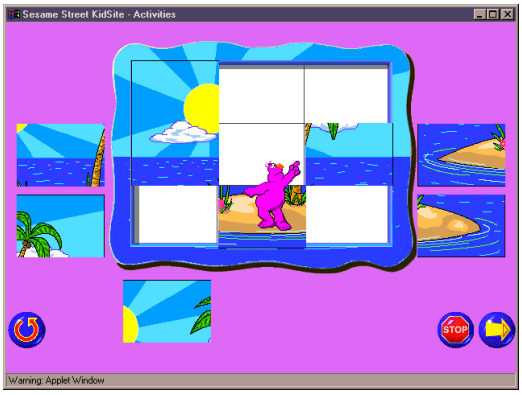


Figure 7 - Arranging components in a layout that defines artefact positions

– Spatial properties of the artefact position may be changed, e.g. x and y coordinates;

– Associations between artefacts and their respective positions may be reassigned, e.g. placing an artefact in a predefined position (Cf. Figure 7);

– The method of arranging artefacts in the layout may be redefined, e.g. by changing the arrangement glue from the Cartesian to the polar coordinate system.

When to use this pattern:

✓ When several artefacts, such as text, images, animation, need to be visually arranged.

✓ When a template layout needs to be defined, e.g. the positions of gaps within the layout are based on the positions of variable components in the template instances.

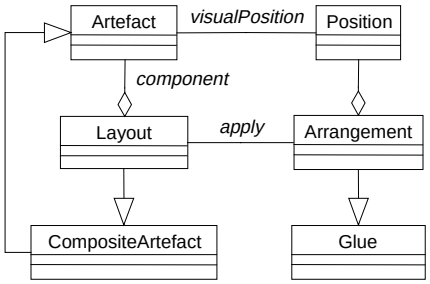
✓ When several presentation channels need to be arranged for simultaneous viewing.

✓ When resizing visual composites, e.g. resizing the browser window may require to repositon text and images.

✓ When the order of overlapping artefacts needs to be changed, e.g. to show hidden objects.

✓ When the method of artefacts visual arrangement is to be altered, e.g. changing from 2D presentation to 3D presentation.

Sample structure:



Examples:

- All multimedia products such as slides, web pages, text and graphic documents need to be arranged for viewing.
- Many existing graphical editors allow positioning and repositioning of graphic components. The components can be moved across the canvas, they can be sent into background or foreground, some also allow changing the viewing mode (e.g. design vs. preview, or 2D vs. 3D) or the perspective (one point vs. two point).
- Text editors and word processors also allow repositioning of textual components, changing the sections order in the outline, or changing the display mode (e.g. page view, outline, print preview).

Related Patterns: `Template`, `Define And Run Presentation`, `Synchronise Channels`

Pro's:

- The position within the layout is independent of the artefact. This means that changes to the layout do not affect the internal structure of components.
- Arrangement glue can be changed without altering relative position of artefacts. This allows dynamic mapping of the coordinate systems, dimensions, magnification, perspective or graphical precision.
- Positions can be defined in non-numeric terms, e.g. gaps in a visual template.
- Component visibility can be controlled either by the existence/non-exitence of their layout positions, sharing positions with occluding objects, or the visibility property of layout positions.

Con's:

- Changes in the layout may force the designer into changing certain properties of participating artefacts, e.g. moving an image to a new place may require it to be resized or moving text to a new position may suggest changes to its font.

Comment: Note that in this pattern an artefact's position becomes a visual attribute of the artefact whereas in the `Template` pattern the location of the artefact is defined by the gap it is filling or replacing. While `Define And Run Presentation` is also concerned with the presentation of multimedia artefacts, the pattern focuses on the temporal rather spatial arrangement of artefacts.

Note that changes to the artefacts layout do not affect the composition of the layout artefact, whereas the modification of the layout composition will always affect the arrangement of the remaining artefacts.

1) **Manipulation of the contents of composite artefacts, e.g.**
   a) Adding an artefact to a collection;
   b) Grouping a collection of artefacts;
   c) Splitting a group of artefacts;
   d) Removing an artefact from a collection;
   e) Union of two artefact collections;
   f) Intersection of two collections;
   g) Extracting a subset of a collection;
   h) Selecting an artefact from a collection;
   i) Performing an operation on the selected artefacts in a collection;
   j) Listing all semantic attributes of an artefact in a collection;
   k) Setting a semantic attribute of an artefact in the collection;
   l) Checking the value of a semantic attribute of an artefact in the collection;
   m) Deleting a semantic attribute of an artefact in a collection.

2) **Imposing quality requirements on the processes manipulating artefacts and on the properties of the resulting composition, e.g.**
   a) Designing new artefacts with reuse in mind, i.e. follow the reuse life-cycle of analysis, organisation and synthesis;
   b) Generalise artefact features;
   c) Maximising semantic cohesion in composite artefacts, i.e. all components should support the central function of a collection;
   d) Minimising semantic coupling between composite artefacts, i.e. components should interface via an explicit "glue" rather than having hidden dependencies;
   e) Clarifying artefacts composition, representation and semantics, i.e. each composite artefact should have an adequate documentation;
   f) Simplifying artefact design, i.e. making sure that compositions are simple in terms of their visual and semantic properties;
   g) Matching artefact semantic properties, i.e. artefacts properties are defined in such a way that they could be manipulated and matched to establish artefacts connectivity;
   h) Facilitating easy navigation between artefacts, i.e. artefact semantic relationships and visual connectedness can be easily tracked;
   i) Keeping track of design decisions, i.e. the steps taken in the process of building multimedia artefacts could be recorded and explained for future reference.

3) **Changing artefacts arrangement by setting and resetting their static audio-visual relationships, e.g.**
   a) Joining artefacts by matching audio-visual properties of components;
   b) Breaking off composite artefacts along the joining properties;
   c) Taking a part of a composite artefact;
   d) Hiding / showing artefacts;
   e) Rearranging the audio-visual properties of artefacts in relationship to others (e.g. rotation, resizing, pitch, playing speed, etc.);
   f) Setting the relative position of artefacts (in respect to the layout scheme);
   g) Changing the layout method in a collection (e.g. flow, grid, pack-it, bag).

4) **Changing the presentation sequence by altering the temporal relationships between artefacts, e.g.**
   a) Presenting artefacts concurrently in parallel channels;
   b) Presenting artefacts one after another in sequence;
   c) Presenting artefacts repetitively in a loop;
   d) Picking an artefact in random for presentation;
   e) Taking user's input to determine the next artefact to present;
   f) Using a grammar to determine the next artefact to present;
   g) Moving along presentation history;
   h) Changing artefact sequence;
   i) Changing artefact ordering scheme;
   j) Selecting an artefact off the map;
   k) Synchronising artefacts presentation.

5) **Defining the process of constructing new artefacts using existing multimedia components, e.g.**
   a) Creating a reusable template for the construction of new artefacts;
   b) Instantiating a template by filling it with other artefacts;
   c) Parametrising an artefacts, i.e. inserting into an artefact a gap, a view or a window onto other artefacts, so that we could view other artefacts through it (this is similar to defining a formal argument in the procedure);
   d) Passing artefact via parameter, i.e. instantiating a parametrised artefact with an artefact that matches the formal parameter;
   e) Copying and adapting an artefact;
   f) Clipping an artefact, i.e. creating an artefact that is defined in terms of another artefact's subset while preserving the integrity of the source artefact, so that an active link between them is established;
   g) Adding control information to components, i.e. making artefacts respond to mouse and keyboard action and effectively turning them into controls.

6) **Defining the method of artefact utilisation and interaction, e.g.**
   a) Clicking a hyperlink and moving the control focus to another artefact;
   b) Following the chain of history;
   c) Accessing menu;
   d) Filling-in a form;
   e) Invoking controls;
   f) Communicating with the remote process;
   g) Showing a hint or status information;
   h) Free interaction.

Table 2 - Required capabilities of authoring tools to support six dimensions multimedia reuse

3. Summary and Conclusions

In this paper we proposed an approach to composing multimedia products. Our approach emphasises the reuse of multimedia artefacts created in the process. We identified a number of multimedia authoring idioms dealing with atomic artefacts and we suggested several multimedia reuse patterns for handling composite artefacts. Our patterns address the issues of artefact contents, quality, arrangement and presentation, construction and utilisation. They are concerned with composition and decomposition of artefacts, their generalisation through the templates, their visual arrangement and presentation over the communication channels, as well as, the quality model dealing with the analysis, organisation and synthesis of reusable multimedia artefacts.

We are currently working on a comprehensive multimedia pattern language and the presented patterns are a small part thereof. Table 2 itemises the aspects of multimedia authoring and reuse that we are considering as the research directions for our immediate future. We are also involved in the development of an authoring tool that relies on the multimedia-reuse pattern language to produce a high quality and effective tool in multimedia design and delivery.

4. Bibliography

1. Buford, J.F.K. (1994): *Multimedia Systems*: ACM Press.
2. Cybulski, J.L. (1996): *Introduction to Software Reuse*, Technical TR 96/4, The University of Melbourne: Melbourne.
3. Cybulski, J.L. and M. Mackowiak (1997): *Teacher's MATE: Multimedia Assisted Teaching Environment*. in *"Doing IT at Melbourne" Symposium*. University of Melbourne, p. 56-61.
4. Hart, G. and A. Gilding (1997): *Virtual Tutorials, Virtual Lectures, Virtual Prisons?* in *ASCILITE*. Curtin University of Technology, Perth, WA, Australia: http://www.curtin.edu.au/conference/ASCILITE97/papers/Hart/Hart.html, p. .
5. Haskins, D. (1994): *The Complete Idiot's Guide to Multimedia*. Indianapolis: Alpha Books.
6. Rossi, G., A. Garrido, and S. Carvalho (1996): *Design Patterns for Object-Oriented Hypermedia Applications*, in *Pattern Languages of Program Design*, J.M. Vlissides, J.O. Coplien, and N.L. Kerth, Editors. Addison-Wesley Publishing Company. p. 177-191.
7. Tidwell, J. (1998): *Interaction Design Patterns*, Web Report http://www.mit.edu/~jtidwell/interaction_patterns.html, InConcert, Inc.