Assisting Web Site Development with Reuse and Patterns

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 systematic approach to Web site design that emphasises reuse of multimedia objects and at the same time addresses quality and productivity of the development process. Our approach is described with the aid of pattern language addressing six dimensions of multimedia authoring and reuse, i.e. contents, quality, construction, utilisation, arrangements and presentation of multimedia products. This paper focuses on methodological aspects of multimedia design and reuse. We present our ideas in a form of the "methodology pattern" and provide an extensive example of Web site development.

1. Introduction

In recent years we have observed multimedia to gain entry into personal computing and business systems. Now non-programmers have access to tools supporting development of multimedia products, e.g. teachers create interactive multimedia learning environments, publishers present high quality multimedia marketing material, technical writers enhance technical reports, manuals, and on-line documentation. With the advent of World Wide Web, multimedia becomes the standard way of presenting information in a globally distributed environment. Nevertheless, currently utilised tools for the construction of multimedia material are still in the stage of their infancy. We find multimedia applications to suffer from many problems, e.g. [3]:

- lack of software designed to construct, manage and use user-friendly human-computer interfaces that would seamlessly combine various media;
- lack of software to satisfactorily support non-programmers in multimedia products development;
- lack of readily available support technology in the area of multimedia artefacts storage, selection and retrieval;
- lack of sophisticated search techniques for locating required multimedia artefacts of various types in large repositories.

Considering these problems, we are currently developing of a Web-based authoring tool that will use a large multimedia repository with potentially great number of audio-visual objects. We also propose guidelines and methods useful for the developers of authoring tools to support effective building of multimedia products. Since we are specifically interested in

the issues of multimedia development for the Web, in this paper we are using terms “multimedia product” and “Web product” interchangeably.

The proposed authoring tool shall be able to manipulate various types of audio-visual objects that are commonly incorporated in Web pages such as text, images in GIF and JPEG format, animation, video (e.g. AVI, MOV and MPEG) and sound in WAV, AU or MIDI format. We refer to these audio-visual objects as multimedia artefacts. Some of these artefacts will be interactive, others will be presented in a passive way. The tool will allow combining multimedia artefacts to form multimedia products, presentations, and rich multimedia environments. It will also allow changing the artefact attributes such as colour, size, position, speed of presentation, etc. Processes manipulating audio-visual artefacts, e.g. those responsible for the presentation, storage or synchronisation, can also be combined and their attributes can be modified. Due to these similarities between audio-visual objects and processes manipulating them the proposed authoring tool shall treat them in a uniform way. Thus we consider the processes to be multimedia artefacts as well. The authoring tool shall be able to describe, classify and store both objects and processes in a common multimedia repository.

2. Development with Reuse

Development of multimedia products is extremely labourious and costly. It often requires expensive software tools and professional services [3]. Multimedia design methods are still very immature and informal, hence they sometimes lead to incomplete, inconsistent, difficult to use and ill-documented products. Multimedia design methods currently in use have been adopted from the fields of publishing, technical writing and art design. These methods may not be fully effective when applied to modern multimedia development. Traditionally they focus specifically on artwork but they largely ignore the development process. At the same time multimedia products are very complex. They have sophisticated structures and interdependencies that necessitate the use of software tools and large repositories of components. To deal with the complexity of multimedia products, developers need special design methods and tools to support them. Even the design approaches used with some success in other fields, e.g. software engineering, are not always appropriate for dealing with this complexity.

A typical approach to the construction of multimedia products is to develop them entirely from scratch. Considering that multimedia developers usually have at their disposal large collections of ready-made components, such an approach is not optimal. We believe that pro-active reuse of multimedia can provide significant savings in time and production costs. Effective reuse of multimedia will have to address issues related to artefacts’ analysis, organisation and synthesis [4], which at the very least imply the aspects of artefacts’ storage and sharing, search, retrieval and their composition.

In today’s multimedia systems, especially commercial ones, sharing of artefacts is based on informal repository facilities, e.g. ClipArt databases and World Wide Web. Typical access to these collections is usually achieved with the use of unsophisticated methods, which include hyperlinks, bookmarks, and query languages. Existing search engines such as Alta Vista, Yahoo, Lycos and InfoSeek, though more complex, are heavily dependent on textual representation of searchable artefacts and are, hence, less applicable to multimedia objects.

Even Yahoo Image Surfer requires entering textual descriptions and keywords for classification of images. Although commercially available authoring tools provide facilities for the creation of multimedia products, they support only the most simplistic methods of artefacts' composition, such as grouping, insertion and containment. We believe that the existing facilities in authoring tools do not offer sufficient functionality to fully support reuse of multimedia. Serious exploration of practical reuse of multimedia artefacts is therefore urgently required.

Generally speaking, *reuse* is the use of previously acquired concepts or objects in a new situation. It involves matching of new and old situations, duplication of already developed objects and actions, and their adaptation to suit new requirements [5].

A *reusable artefact* is a certified, fully developed, possibly generic component, available for the integration into a multimedia product under development. Certified means that a reusable component satisfies some quality and reusability criteria. Generic means that the reusable component is abstract enough to be adapted to particular situations, e.g. a template. [8]

A controlled collection of reusable artefacts constitutes a reuse *repository*. Reuse repositories are expected to provide certain types of services to their users, e.g. storage, searching, inspecting and retrieval of artefacts from different application domains, loading, linking and invoking of stored artefacts, specifying artefacts relationships, etc.

"*Domain analysis* is the process of identifying and organising knowledge about some class of problems -- the problem domain -- to support the description and solution of those problems" [13].

Box - Reuse Concepts and Terminology

The majority of work in reuse to date has been done in the area of software engineering (See Box). Initially the emphasis of reuse effort was on code reuse, both in the binary and source form. Recently, however, reuse found many new applications such as reuse of analysis and design work products [2, 6, 11, 12], educational material [10, 14], multimedia components [7] and processes [15]. In this paper we focus our attention on the latter.

In a typical approach to software reuse, development takes place in two distinct phases: development-for-reuse and development-with-reuse. The focus of the first phase is on development of reusable components, their detailed description, classification and storage in the repository for future reuse. The second stage focuses on the construction of software products with the utilisation of reusable components drawn from the repository. Frequently the two phases are hard to separate as developers discover new components in the process of product development. We consider a similar approach can be taken when *developing reusable multimedia.*

While constructing Web products or reusable multimedia components, developers may take a number of different approaches.

- One approach can be called "find and apply" when a product designer browses through various existing products and collections available on the World Wide Web in search of appropriate components. When found, the components are incorporated in the future Web product.
- Another approach taken by some developers is to maintain collections of multimedia components over long periods of time. We call it "find-store-apply". With this approach developers add new components to the collection when they come across anything that may prove to be useful for the future product. The drawback of this approach is that it relies on the developers managing multimedia resources themselves. These collections are usually maintained in the ad-hoc fashion without proper indexing or documentation of

artefacts. Frequently the potentially useful components are stored in a system of directories and subdirectories. This is not an ideal strategy to support object composition and sharing due to the general limitations of a filing system [16]. The search in such collections can only be conducted by artefact name or its contents.

- Other developers create specialised multimedia repositories where artefacts are classified and indexed, e.g. using enumerated schemes or facets [17], then stored and retrieved as required. This approach follows several steps that involve finding, organising, and applying artefacts, hence, we call it "find-organise-apply". In any case a developer has to view an artefact to determine if it meets the requirements.
- Our approach is more systematic, i.e. we actively analyse existing products and collections for the opportunity to identify reusable multimedia components. We properly document, classify and index them and check whether or not a component can be generalised. We call it "analyse-organise-synthesize".

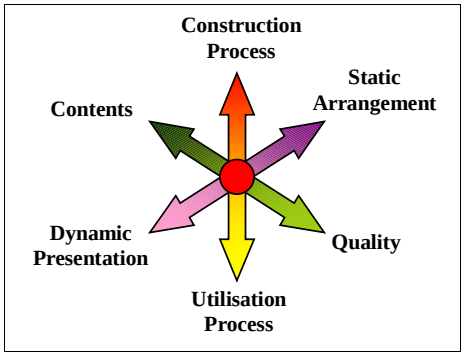

Figure 1 - Dimensions of multimedia authoring and reuse

To guide multimedia developers, who would be interested in using the last approach, we describe our method in a form of a pattern [1]. This format enables us to formally describe the recurring issues arising in the process of multimedia product development, it provides examples of situations in which the proposed development process is applicable, and specifies the consequences of using our approach.

The proposed pattern belongs to a cohesive and consistent group of patterns, "the multimedia reuse pattern language", which provides guidelines for effective multimedia reuse. This pattern language addresses issues of artefacts creation, presentation and interaction. These issues fall into 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 [4]. In our previous work we presented the patterns focusing on artefact construction, contents, arrangement and presentation. The problems of quality of multimedia artefacts are explored in this paper.

3. Multimedia Process Quality Pattern

The patterns, in their structure and expression, aim at the precision, consistency, and clarity. There exist several formats for pattern description depending on the level of its abstraction but all of them contain these essential elements:

- **Name** - usually a noun, identifies the pattern and expresses what a pattern deals with.
- **Problem description** - provides a general description of the recurring situation.
- **Solution** - discusses a general approach to solving the problem

Typically patterns also describe the problem context and forces. The problem context implies the circumstances in which the problem recurs most frequently, whereas forces influence the problem and its solution. Forces have an impact on problem solution as they

define trade-offs that must be considered before applying the solution. Both forces and context precisely describe the circumstances in which application of a pattern could be justified. The resulting context is the context of the solution. It describes the state of the system after the pattern has been applied.

Some pattern authors customise the pattern format to suit their needs depending on the level of the patterns' abstractness [1, 9, 15]. To describe our patterns, we decided to add the following elements to the core of the pattern format:

- **Pre-condition** - conditions to be met before the solution can be applied.
- **Post-condition** - the situation resulting from the successful application of the solution.
- **When to use this pattern** - generalised description of the situations where the pattern is applicable
- **Pro's** and **Con's** - trade-offs to consider before using the pattern
- **Comments** - additional information on the situation and application of the pattern.
- **Examples** to show the cases when the problem occurs and to illustrate the use of the pattern to solve the problem.

With this format, the pattern's context is covered in two sections: Problem Description and Pre-condition. Thus potential users of the pattern could decide if this pattern is applicable to their situation just by reading only these two sections.

We use the described format to present the multimedia process quality pattern, `Analyse Organise Synthesize`. This pattern describes a methodology that addresses issues of multimedia product development and discusses maintenance of a multimedia repository. By applying this pattern developers may improve the quality of individual components. All components stored in the repository are properly documented and, thus, can be easily found and adapted to new requirements. The repository also contains previously tested and thoroughly documented processes that support multimedia artefacts design and implementation. All these factors are contributing to the overall quality of the product.

Analyse Organise Synthesize

Problem Description: A multimedia product needs to be developed. Where to start and how to achieve the quality of the final product and assure high productivity of the process?

Pre-condition: Existence of -

- ✓ some legacy products;
- ✓ project requirements;
- ✓ repository of reusable multimedia components.

Post-condition: Results in -

- ✓ high quality multimedia product composed of reusable multimedia artefacts;
- ✓ extended repository of reusable of multimedia artefacts.

Solution: Apply a *reuse* and *repository* based approach to artefacts' construction, maintenance and utilisation. The approach describes and prescribes a number of activities in the artefacts' life cycle (Cf. Table 1). Each activity proposes several tasks that may be undertaken in the processing of an artefact. These activities may be performed individually, in a sequence, or overlap.

Analyse	Organise	Synthesize
• Identify • Describe and represent • Generalise	• Classify and index • Store • Search • Retrieve	• Analyse • Select • Adapt • Create • Integrate

Table 1 - Activities in the artefacts' life cycle

1. ***Continually analyse the existing multimedia products:***
   - Identify reusable components in legacy systems that could be potentially useful.
   - Describe and represent the features of artefacts identified as potentially reusable.
   - Generalise, if necessary, and document artefacts selected for reuse to widen the scope of their applicability to suit development of new multimedia products.
2. ***Organise a repository of reusable artefacts:***
   - Classify and index artefacts to be stored in the repository. This can be based on facets, keywords or taxonomia.
   - Store reusable artefacts in a multimedia repository.
   - Search for the candidate artefacts suitable for the new project. The search techniques are dependent on the repository facilities and proper documentation of artefacts.
   - Retrieve the candidate artefacts for further processing determined by the search.

   Faceted classification relies on several well-defined attributes, called facets. Facets specify a controlled vocabulary used in the artefact classification. To refine artefact matching, each facet specifies a metric to determine closeness of classification terms, which are commonly broadened with the use of synonym lists and a thesaurus [5].
3. ***Synthesize a new multimedia product:***
   - Analyse the requirements of the project.
   - Select the necessary artefacts from the collection of candidate artefacts.
   - Adapt the selected artefacts, if necessary, to meet the requirements of the current project.
   - Create new artefacts as required, document them and add them to repository.
   - Integrate selected and newly created artefacts into a final product.

When to use this pattern:

✓ When creating a multimedia product, e.g. a Web site, multimedia presentations, multimedia materials on CD-ROM.

- ✓ When adapting existing multimedia documents and their components for reuse.
- ✓ When assembling and maintaining a large collection of multimedia artefacts.
- ✓ When searching for artefacts suitable for inclusion in the new multimedia products.

Comments: The pattern requires that the multimedia repository has already been established. This forces the potential users of the pattern to design and implement the repository.

Pro's:

- The users of this pattern will enrich their multimedia repositories with high quality reusable multimedia artefacts that are easily adaptable, possibly generalised, and properly documented. These components can be shared between multiple products.
- The acquisition of artefacts is conducted on a continuing basis, not only during the product development process.
- Application of the pattern increases developers' productivity in the long term.
- Application of the pattern facilitates search of repository and selection of matching artefacts.

Con's:

- Copyright issues should be considered before storing multimedia artefacts in the repository, modifying and using them in products.
- Application of the pattern may meet opposition from developers, who may think that their approaches are better than those proposed.
- Adoption of reuse practices may increase developers' workload.
- The multimedia repository may be costly to manage in short term. As for any repository maintenance, setting it up and entering data may seem to be boring.

Example:

This example illustrates how the `Analyse Organise Synthesize` pattern can be used to build a Web site. For this example we selected the official site for the French Government Tourist Office "France Tourism and Travel : French Tourist Office" (URL: http://www.fgtousa.org/ - cf. Figure 2). We picked this site only to illustrate the application of the pattern because it has great number of potentially reusable components. We did not participate in the development of the site. The question is how this site could be created with the help of multimedia patterns and `Analyse Organise Synthesize` in particular.

When we face the task of developing a Web site, we usually start with the questions posed in the pattern's problem description. Where do we start? How to develop a Web site that attracts attention from the first sight, conveys information to the customer (quality of the product) and at the same time allows increase in productivity, saving time and cost reduction.

We will start with domain analysis. First we identify required artefacts and existing artefacts, e.g. text files, hand-written ideas, images in electronic form and on paper, videos and animations, sound files, etc. Hardcopy documents will need to be transferred into electronic form, e.g. by scanning text and images. We also need to establish a repository of multimedia artefacts. For example, an object database such as Jasmine or relational database

Figure 2 - Some Web pages from the site http://www.fgtousa.org/

with BLOB facility can be used to support this project, e.g. Microsoft Access or Oracle. Thus, the pre-condition - existence of some legacy products, a multimedia repository and availability of project requirements - has also been met. Hence, we will follow the solution described by the pattern.

Figure 3 - Logo image shared by all pages

Analysis stage. Looking at the site design, we can immediately see that it uses a number of components such as frame structure, transparent table layout (Cf. Figure 5), several images including travel office logo (Cf. Figure 3) and some advertisements. Those components are shared between many pages.

Figure 4 - Menu/image buttons used through the pages of the site

To the developers of this site it will be beneficial to make these components reusable because their productivity will increase and less time will be required for implementation of Web pages. The transparent table can be filled with various information resulting in new Web pages with consistent layout. Menus used throughout several Web pages allow reuse of image buttons with associated links (Cf. Figure 4). Image buttons constituting menus are used as section headers on the main page (Cf. Figure 2). The advertisements are displayed with the help of some Javascript functions that allow changing images at certain intervals of time and maintaining image links to the relevant Web sites (Cf. Figure 9).

During this analysis process we would need to

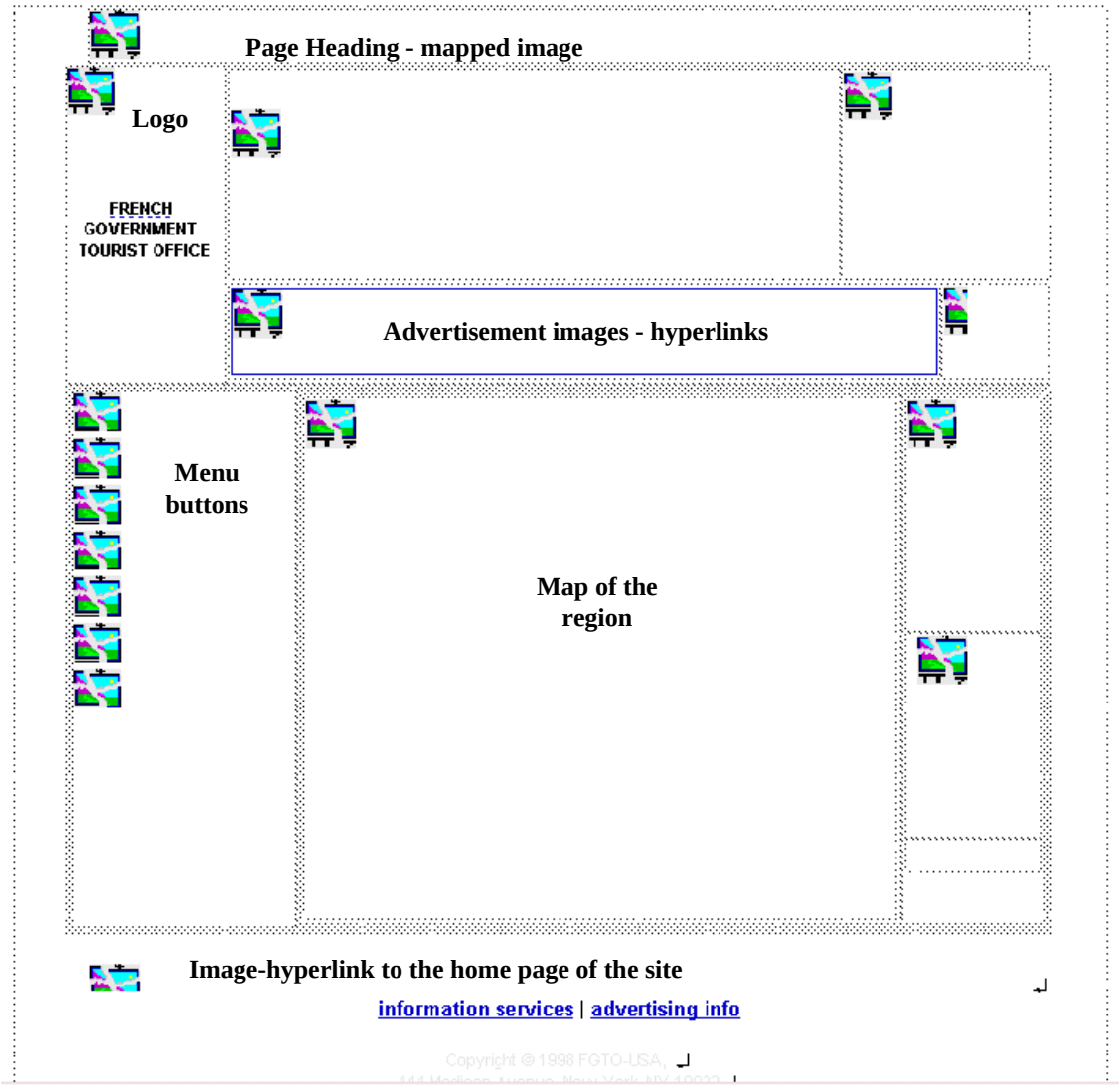

Figure 5 - Transparent table to support consistent layout of pages of the site

Figure 6 - Web pages that could be produced with the same layout template

describe all existing artefacts and note down the ones that could be reused without modifications and the ones that have to be modified or generalised. For example, the pages of the site have consistent layout, which is achieved by using a transparent table (Cf. Figure 5). This table could be used as a template with some cells maintaining the same contents through all or most pages and other cells having variable components. For example, Web pages in Figure 6 could be produced with this template. Image buttons actually differ by text only thus the generalisation can take place in the form of image template creation (Cf. Figure 7). The same applies to the image headers at the top of each page. For now we just generalise components whenever possible and properly document them. Documentation may include artefact's type, label, , list of descriptors and attributes and thumbnail as some sort of preview.

Figure 7 - The template to produce menu items/headings: consists of the background image (common to all components) and the text box to enter text of the items

Organisational stage. During the next stage we classify of candidate components and storing them in the multimedia repository. We use an image of map of Brittany (Cf. Figure 8) to illustrate how this can be done. Assuming the use of a relational database as multimedia repository, the following fields may need to be filled:

- artefact number,
- artefact URL,
- artefact type (image),
- label (map of Brittany),
- contents description (one colour region map with the label Brittany),
- keywords (e.g. Brittany, map, region, France),
- file type (.gif),
- file size,
- preview (the field that allows the preview of the image).

Figure 8 - Map of Brittany

Fields to be indexed to speed up search through the repository are artefact type, label, contents description and keywords.

Synthesis stage. The next step in the project would be selection of multimedia artefacts suitable for the project. The selection could occur either by visual inspection of artefacts, or their attribute values, or based on previous applications of an artefact. Some artefacts will need further adaptation. For example,

- headers/hyperlinks should be generated from the template described above (Cf. Figure 7);
- images may need resizing,
- colours of existing artefacts may need to be changed to support the selected colour scheme, in this case the colours of the French flag,
- some sequences of images may be transformed into animations using packages such as Microsoft Gif Animator or Gif Construction Set or by applying Javascript. This site uses Javascript code for animation that can be reused for animation effect using different arrays of images (Cf. Figure 9).

At the same time new artefacts will be created. These new artefacts should be properly documented and added to the repository.

The new objects can be composed with HTML, scripts, frames and by grouping them. For example, for this Web site HTML is used to hold components together and Javascript is used to produce animation (advertisement images replacing each other).

At the stage of integrating components into a composite, one may need to match their properties (e.g. resize images, change text fonts, change colours of original components). Developers of multimedia products have to do similar things in the implementation process.

Our approach facilitates both development-for-reuse and development-with-reuse, which is promoted and used with the

```
onload="setTimeout('load_image()',TIMER);">

<script>
var ID = 0;
var MAX_ID = 3;
var TIMER = 2000;

picture = new Array(MAX_ID);

picture[0] = "ewban.gif";
picture[1] = "orionbar.gif";
picture[2] = "abotelban.gif";

locations = new Array(MAX_ID);

locations[0] = "http://www.europeanwaterways.com/indexfgto.htm";
locations[1] = "http://www.apartmenthotels.com/indexfgto.htm";
locations[2] = "http://www.abotel.com";

function load_image() {
  if (document.images) {
     ID++;
     if (ID == MAX_ID) ID = 0;
     document.images["testimage"].src = "adbanners/" + picture[ID];
     setTimeout('load_image()',TIMER);
  }
}

function load_image_url() {
     window.location.href = locations[ID];
}

</script>

<a href="javascript:load_image_url();">
<img src="adbanners/ewban.gif"
  alt="European Waterways / ORION Apartment Hotels /
                                  Abotel Boutique Hotels"
  border="1" width="460" height="55" name="testimage">
</a>
```

Figure 9 - Javascript to produce animation from images that are hyperlinks to Web pages

certain degree of success by some software-engineering practitioners. As a result of applying this approach, one develops a well-organised multimedia repository, which can be viewed as an asset to host organisation.

In view of the previous experience in the successful implementation of reuse, we put emphasis on work and management practices rather than technological issues.

4. Summary and Conclusions

In this paper we proposed an approach to constructing and maintaining multimedia products and repositories. Our approach emphasises the reuse of multimedia artefacts created in the process and addresses the quality and productivity of the development process. We presented this approach using pattern format where the solution is described in a sequence of steps. Some of the steps described, such as identification, representation, classification, search, selection and adaptation, can be automated and that's what our project is aiming for.

In our work, we focus on the construction, organisation and management of reusable multimedia components. We are aiming at the creation of a comprehensive multimedia-reuse pattern language. We defined a collection of multimedia-reuse patterns that address six dimensions of multimedia authoring and reuse. 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. It is also intended to 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.

5. Bibliography

1. Appleton, B. (1997): *Patterns and Software: Essential Concepts and Terminology*, : http://www.enteract.com/~bradapp/docs/patterns-intro.html.
2. Biggerstaff, T. and C. Richter (1987): *Reusability Framework, Assessment and Directions.* IEEE Software(July): p. 41-49.
3. Buford, J.F.K. (1994): *Multimedia Systems*: ACM Press.
4. Cybulski, J. and T. Linden (1998): *Composing Multimedia Artefacts for Reuse*. in *Pattern Languages of Programming*. Allerton Park, Illinois, USA: http://jerry.cs.uiuc.edu/~plop/plop98/final_submissions/P38.pdf
5. Cybulski, J.L. (1996): *Introduction to Software Reuse*, Technical TR 96/4, The University of Melbourne: Melbourne.
6. Cybulski, J.L. (1998): *Reuse of Software Requirements.*, in *Information Systems*, University of Melbourne: Melbourne.
7. 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.
8. D'Alessandro, M., P.L. Iachini, and A. Martelli (1993): *The Generic Reusable Component: an Approach to Reuse Hierarchical OO Designs*. in *Second International Workshop on Software Reusability "Advances in Software Reuse"*. Lucca, Italy: IEEE Computer Society Press, p. 39-46.

9. Gamma, E., R. Helm, R. Johnson, and J. Vlissides (1995): *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, MA: Addison-Wesley.

10. Ip, A., R. Canale, P. Fritze, and G. Ji (1997): *Enabling re-usability of courseware components with Web-based "Virtual apparatus"*. in *ASCILITE 97*. Curtin University of Technology, Perth, WA, Australia: http://www.curtin.edu.au/conference/ASCILITE97/papers/Ip/Ip.html

11. Li, H. (1993): *Reuse-in-the-Large: Modeling, Specification and Management*. in *Second International Workshop on Software Reusability "Advances in Software Reuse"*. Lucca, Italy: IEEE Computer Society Press, p. 56-65.

12. Lubars, M.D., ed. (1988): *Wide-Spectrum Support for Software Reusability*. . Computer Society Press.

13. Prieto-Diaz, R. and G. Arango (1991): *Domain Analysis and Software Systems Modeling*. Los Alamitos, California: IEEE Computer Society Press.

14. Roschelle, J. and J. Kaput (1997): *Educational Software Architecture and Systemic impact: The Promise of Component Software*, Web Report http://www.simcalc.umassd.edu/simcalc/library/S&SImpact.hqx, University of Massachusetts: Dartmouth.

15. Rossi, G., D. Schwabe, and A. Garrido (1997): *Design Reuse in Hypermedia Applications Development*, : http://www-lifia.info.unlp.edu.ar/~garrido/.

16. Shih, T.K., H.-C. Liu, C.-H. Kuo, and T.-F. Kuo *A Distributed Multimedia Database Supports Collaborative Learning*, Web Report http://www.uis.edu/~mcc97/html/papers/kuo/, Tamkang University: Tansui, Taiwan.

17. Sorumgard, L.S., G. Sindre, and F. Stokke (1993): *Experiences from Application of a Faceted Classification Scheme*. in *Second International Workshop on Software Reusability "Advances in Software Reuse"*. Lucca, Italy: IEEE Computer Society Press, p. 116-124.