Proceedings of Sixth European Conference on Pattern Languages of Programs EuroPLoP, Irsee, Germany, 4 - 8 July 2001. Note the change of title in response to workshop feedback.

Planning the Process of Multimedia Development

Tanya Linden and *Jacob L. Cybulski*

tanyal@unimelb.edu.au
http://www.dis.unimelb.edu.au/staff/tanya
ph: +61 3 8344 9272
fax: +61 3 9349 4596

jacob@unimelb.edu.au
http://www.dis.unimelb.edu.au/staff/jacob
ph: +61 3 8344 9244
fax: +61 3 9349 4596

Department of Information Systems
University of Melbourne
Vic 3010, Australia

Abstract

This paper looks at some problems that commonly hinder the multimedia development process, e.g. difficulties in finding, accumulating, refining, and organising multimedia artefacts. The suggested solutions to these problems are described with the aid of a set of patterns, which could alleviate these difficulties and at the same time improve developers' productivity and enhance quality of multimedia products and their individual components.

1. Introduction

Construction of multimedia products involves development and composition of artefacts of various media types including text, graphics, animation, video and sound [1, 6, 19]. This process requires special methods, techniques, and tools capable of handling media specific features. Such development is labour intensive and costly, and demands special skills ranging from software development and technical writing to graphic design and sound engineering. The time and cost of such undertaking can be reduced by reusing multimedia components across multiple projects and by sharing them between various development groups. Some of the useful approaches to sharing and reusing multimedia are commonly borrowed from software engineering [2, 11, 21], e.g. use of templates, components, and parameterisation. Unfortunately the generic reuse methods fail to consider features specific to multimedia components, for example, colour, shape and texture for images, motion parameters for video, rhythm signature, chord, and melody for music. Dealing with these features is especially difficult due to subjectivity of their perception.

Multimedia developers need repositories of reusable components that can be shared across projects or organizations. Some modern multimedia tools (e.g. Microsoft FrontPage) address this issue by providing multimedia repositories that could be stocked with frequently used multimedia components. Over time multimedia repositories can significantly grow in size [13]. The size of multimedia repositories, the peculiarities of multimedia components and the subjective nature of their attributes perception pose many challenges in the process of multimedia storage, search and retrieval.

In our previous work [3] we proposed to address some of these challenges relating to the aspects of constructing and presenting reusable multimedia. In this paper we aim at addressing other facets of multimedia reuse, such as finding, accumulating, refining and organising multimedia artefacts.

2. Patterns Facilitating Reuse in Multimedia

2.1 Squirrel

Problem

How do you ensure the availability of quality artefacts in multimedia development?

Context

Over the years, organisations involved in training, consulting and publishing establish vast archives of multimedia documents and components. For example, educational institutions store collections of teaching and assessment material, which includes lecture presentations, printed notes, tutorial exercises, laboratory examples, demonstrations, case studies, project descriptions, assignments and exams. Such information comes in a variety of electronic media types, e.g. text, images, sound, videos and executable programs. Alternatively, some material may be available in hardcopy form only. When creating new products, multimedia developers need to effectively and efficiently find, combine and create multimedia components.

Forces

Reuse awareness. Development of multimedia products necessitates reuse. However, multimedia reuse cannot be fully automated, hence it is developers who have to be aware of multimedia reusability in all of their development tasks and like squirrels thinking about food for winter, accumulate potentially reusable components.

Personal attitude. Although developers realise that reuse is important, there is a subjectivity issue in the accumulation process: what to accumulate and what to ignore is based on developer's judgement, which comes from their level of expertise and experience.

Component harvesting. Designing components with reuse in mind from the very beginning creates considerable overhead in cost and effort, however, harvesting potentially reusable components and making them available for future use proved to be beneficial.

Component identification. Reuse of multimedia components relies on their effective finding and identification in existing multimedia products, and the subsequent representation of their information. However, representation of multimedia features of graphics, sound, movies, animation or executable programs may be a difficult task [18].

Component understanding. Successful reuse of product components needs tool support, however, apart from component storage and retrieval, full automation of multimedia reuse process is difficult [24]. Activities such as describing, indexing, classifying and finding multimedia components require developer intervention. On the other hand, these activities face certain difficulties due to the subjectivity factor in describing multimedia properties [13].

Solution

Multimedia developers should continually accumulate potentially reusable artefacts found in multimedia products:

- Scan existing multimedia products and legacy systems with the view to identify potentially reusable components. History of previous uses, if available, may help in the identification process. Some decisions may be made based on the current project requirements.

- Consider the reuse potential of products under construction and their components. Recording information on intended use of such components may help in future selection process.
- Describe and represent the features of artefacts identified as potentially reusable to support future decision-making on components’ selection.

Resulting Context

An organisation will have a stockpile of potentially reusable artefacts. Legacy documents will be transferred into electronic form. Components will be properly documented which will simplify their understanding and classification. With artefacts being properly documented, classified, and indexed they can be easily found and possibly shared within the organisation, which in turn should increase productivity of product development.

Pro’s

The acquisition of reusable artefacts is conducted on a continuing basis.

Previously used components with the history of being useful are accumulated.

Newly developed components are considered as potentially reusable.

Proper documentation simplifies the process of components understanding and selection decision making.

Con’s

Copyright issues should be considered before acquiring and modifying multimedia artefacts.

Unless there exists a quality assurance process, accumulated artefacts may not be of satisfactory quality.

Unless the accumulated collection is properly organised, finding relevant artefacts may be difficult.

Example

Special software is being developed to scavenge the WWW in order to accumulate links to useful components, e.g. crawlers, spiders and robots.

Related patterns

The Nut Cracking pattern may be applied as a follow up procedure after components’ acquisition.

Components need to be properly stored in order to be easily found thus we may need to apply the Stockpile pattern.

2.2 Nut Cracking

Problem

How do you widen the scope of multimedia components applicability?

Context

A collection of components has been accumulated (see the Squirrel pattern). New components were developed for use in a current project from scratch (possibly without considering their reuse in future projects). Later a special team may harvest potentially reusable components and consider how to widen the scope of their use.

A component is not necessarily reused as is but rather after certain modifications to make it suitable for new requirements, e.g. its parts may be extracted and used or the component could be used after its properties are changed (like a squirrel who would eat dried mushrooms whole but would deal differently with nuts, the shells of which need to be removed before the kernels could be eaten).

Forces

Creating versus modifying. It is possible to create all required components from scratch, however it may be much simpler to modify existing components, yet multimedia components that have not been specifically designed for reuse do not easily yield to modification and generalisation.

Use context. It is convenient to have a collection of components for reuse, however it is often not obvious that a component developed for one use context is also suitable for another unless it is generalised.

Just right? If a component is too specific, it is rich in functionality but it may be usable only in very specific contexts. If a component is too general it may fit wider selection of contexts but it would have very little functionality to make it useful enough.

Solution

Multimedia developers should generalise, if possible, and document potentially reusable artefacts to widen the scope of their applicability to new multimedia projects:

- Establish a desired level of generality to maximise applicability of a component without reducing its usability.
- Decompose a composite artefact and store its constituents (remove shells and use kernels).
- Parameterise an artefact to simplify changing its property values.
- Analyse and document the artefact's existing or intended use contexts.
- Improve artefact quality by applying relevant standards, e.g. in graphic design, video and sound production.

Resulting Context

Components are restructured to support their future reuse and the subsequent adaptation to new requirements. A record of artefact's valid use contexts is kept. For example educational institutions may end up with a collection of templates for lecture slides, on-line assessment, and tutorial handouts, which can be shared across subjects.

Pro’s

With generalisation and documentation artefact quality significantly improves.

Past and intended use contexts could serve as examples of possible artefact applications.

Con’s

Copyright issues should be considered before modifying multimedia artefacts.

Related patterns

Before generalising components, a developer may need to acquire them first, thus the Squirrel pattern may need to be applied.

Some generalised components become templates - see pattern Template in [3].

Storage issues of generalised high quality components are addressed by the Stockpile pattern.

2.3 Stockpile

Problem

How do you organise a collection of multimedia artefacts to be able to find required artefacts effectively?

Context

Patterns Squirrel and possibly Nut Cracking have been applied, i.e.

- Reusable artefacts have been collected in electronic form.
- They are well documented, decomposed and generalised when applicable.

Forces

Component availability. Sharing and reuse of components in multimedia projects increases productivity [5] but when components are scattered across the organisation, developers may not know about their existence.

Component access. There exists a large collection of multimedia artefacts, which are scattered across the organisation, however the size of the collection creates problems with finding required artefacts.

Components quality. While applying the Squirrel pattern over time a developer obtains a huge collection of multimedia artefacts, however there is a danger of having quality components mixed and lost among not so useful artefacts.

Solution

Multimedia reuse can be facilitated with a central repository of shared and reusable components. **Multimedia developers should organise, maintain and share a repository of reusable quality artefacts:**

- Assure that artefacts to be stored in the repository are in the quality form.
- Store reusable artefacts in a multimedia repository.
- Classify and index artefacts. A variety of classification and indexing methods are described and used by developers, e.g. facets [12], keywords, enumerated schemes [21], multimedia properties [16, 20] or media-indexes [7]. Incorporate relevance feedback to deal with individual perception and preferences [15].

Resulting Context

High quality artefacts that are properly classified and indexed can be easily retrieved and then reused within an organisation. For example, in an educational institution where some teachers lack sufficient experience in multimedia design, teaching staff could benefit from the use of a repository with multimedia templates for development of learning material. This repository may also contain a set of case studies, which could be used in lectures, tutorials, projects or revision classes across related subjects (e.g. Programming and Systems Analysis and Design). Less experienced staff may rely on the repository of case studies and feel more confident knowing that all stored artefacts are quality assured.

Pro’s

Well-organised and maintained multimedia repository facilitates sharing of components between multiple products.

The richer the repository, the more likely it is that the developer will find most of the required artefacts in there.

With the assured quality form of artefacts, less experienced developers will benefit from such repository

By incorporating relevance feedback information in multimedia artefacts’ classification, the results of the search and retrieval process continually improve with the use of artefacts.

Con’s

Copyright issues should be considered before storing multimedia artefacts in the repository, modifying them and using them in products.

The multimedia repository may be costly to manage in the short term.

As for any repository maintenance, setting it up and entering data may seem to be boring.

Related patterns

Squirrel and Nut Cracking provide context for application of this pattern.

2.4 Query by Example

Problem

How do you describe multimedia properties to reduce the gap in articulation between the system and a user during query formulation?

Context

Multimedia repository. An organisation maintains an extended repository of reusable multimedia artefacts. With artefacts being properly documented, classified, indexed and generalised where possible, they can be retrieved and reused within the organisation.

Requirements. Any new multimedia project normally starts with the statement of requirements. These requirements determine the intended context of artefact’s use and thus help in formulating queries in order to find reusable artefacts in the repository.

Forces

Subjectivity. To find artefacts suitable for the task, a developer submits a query based on task requirements, however different people may interpret some multimedia properties differently, e.g. colour purple may be described as violet. This becomes a problem if perception of people classifying multimedia artefacts differs from that of developers searching for them.

Unsatisfactory textual description. Multimedia properties may be hard to capture in the form of textual description, however it may be easy to point out the requirement from the list of samples.

Solution

Apply QBE[1] approach:

- Provide facilities to submit not only textual queries but also multimedia samples as queries defined using project requirements and intended context of use, i.e. the searcher should be able to give example of either an artefact or its use context as input to a search engine.
- Apply CBR[2]-based search engine.

Resulting context

The query can be formulated by providing samples of sought artefacts or the intended use context.

Pro's

The process of query formulation is simplified.

The problem of misinterpreting textual descriptions of multimedia attributes is eliminated.

Developers searching repositories will be getting more accurate search outcomes, possibly with lower number of artefacts being on the retrieved list.

Con's

There exist a variety of systems and algorithms for content-based search and retrieval of multimedia artefacts so it may be hard to decide which one to use.[3]

It's difficult to apply Boolean operations to attributes in a QBE query.

Sometimes query construction may require sophisticated skills and be very labour intensive, e.g. searching for a musical composition by example may require composing a piece.

Related Patterns

Queries are submitted to a repository search engine, thus we may need to apply the Stockpile pattern first to have multimedia artefacts collection organised as a repository.

The purpose of the search process is to find required components but then there is an issue which ones to select from the resulting list, thus we may need to apply the Use Context pattern.

Examples/Known uses

The following multimedia search tools allow submission of a multimedia sample rather than textual description to initiate the search process: QBIC [4], El Niño [17], Photobook [10], MARS [14].

[1] QBE stands for query by example, an established approach frequently used in user-friendly software.

[2] CBR stands for content based retrieval

[3] Even for the same feature such as texture of images several different algorithms may be available [10, 14, 18].

2.5 Fit for Use

Problem

How do you improve artefacts classification based on their use?

Context

Multimedia repository. An organisation maintains an extended repository of classified multimedia artefacts.

Use Context. A developer is searching for artefacts to fit a particular use context.

Process. The pattern is applied at the time of manual selection of artefacts from the list retrieved and ranked by the search engine.

Forces

Component relevance. Components need to be assessed for their suitability to the multimedia development task, but the suitability cannot be properly judged without taking use context into consideration [8, 22].

Subjectivity. Also formulating and satisfying requirements describing intended use context might be hard due to individual perception of and preferences for multimedia properties [15].

Solution

Development with reuse focuses on constructing products with the utilisation of reusable components drawn from the repository. **Multimedia developer should apply intended use context in order to improve selected and rejected artefacts' classification:**

- Provide relevance feedback on the intended context of use for selected artefacts.
- Provide relevance feedback on the use context the rejected artefact did not match.[4]
- Use provided feedback to improve classification of candidate artefacts.

Resulting Context

Classification of artefacts is improved with every selection or rejection due to relevance feedback information. As a result developers are likely to find relevant artefacts more often and thus the repository becomes a valuable resource.

Pro's

By incorporating relevance feedback information in multimedia artefacts' classification, the results of the search and retrieval process continually improve with the use of artefacts.

Relevance feedback can also deal with subjectivity in the process of formulating requirements for multimedia products and finding and selecting multimedia components to satisfy these requirements.

[4] A component is selected from the retrieved list and is incorporated into a template. The resulting composition may reveal why other components from the list were not selected, e.g. they did not match the colour scheme.

If after being incorporated into the composition the selected component is removed and replaced with another selection, it may have its classification information updated to change its position in the ranking order when it is retrieved next time for a similar query.

Previous uses reflect developers' preferences and thus may help novice developers in making decisions on which artefacts are most suitable in similar circumstances.

Con's

Providing relevance feedback information may be time consuming.

Feedback on artefacts relevance is subjective and, thus, may not be applicable to artefacts' classification.

Related patterns

Squirrel, Nut Cracking and possibly Stockpile have been applied and their resulting context becomes part of a context for this pattern.

This pattern may be used in conjunction with Query by Example, especially to specify aspects of use context that are hard to express verbally, e.g. size of gaps in a template or colour scheme to be matched.

Examples/Known uses

- Relevance network model [8] is based on context-sensitive indexing and retrieval. An adaptive information retrieval mechanism memorises relevance of references to artefacts based on user feedback. This information is then stored in the user profile and it is also generalised so that other users can benefit from the history of previous uses.
- Footprints [23] is a tool that allows recording users navigational history in virtual environments, such as World Wide Web. With this tool its author Alan Wexelblat demonstrated that users were able to find required information with less effort and with less time spent compared to users doing the same tasks without using interaction history.

3. Future Work

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, which provides multimedia developers with guidelines for effective multimedia reuse. This pattern language addresses six dimensions of multimedia authoring and reuse, i.e. the contents and quality of artefacts, their arrangement and presentation, and the processes leading to their construction and utilisation. In our previous work [3] we presented the patterns focusing on artefact construction, contents, arrangement and presentation. The patterns presented in this paper address the quality dimension. In the future, the resulting pattern language is intended to assist the users (and developers) 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.

4. Acknowledgments

We would like to thank Alan O'Callaghan for his insightful comments, help and encouragement provided during the shepherding process. We are also very grateful to participants of EuroPLOP 2001 for their suggestions on improvement of our patterns.

5. Bibliography

1. Buford, J. F. K., *Multimedia Systems*. SIGGRAPH, ed. S. Cunningham. 1994, New York, NY, USA: ACM Press.
2. Cybulski, J. L., *Introduction to Software Reuse*. 1996, The University of Melbourne, http://www.dis.unimelb.edu.au/staff/jacob/mate/publications/reuse-intro-abstract.htm.
3. Cybulski, J. L. and Linden, T., *Composing Multimedia Artefacts for Reuse*, in *Pattern Languages of Program Design*, N. Harrison, B. Foote, and H. Rohnert, Editors. 2000, Addison-Wesley: Reading, MA.
4. Flickner, M., et al., *Query by Image and Video Content: The QBIC System*. IEEE Computer, 1995. **28**(9): p. 23-32
5. Garzotto, F., Malinettit, L., and Paolini, P. *Information Reuse in Hypermedia Applications*. in *The seventh ACM conference on Hypertext*. 1996. Bethesda, MD USA
6. Haskins, D., *The Complete Idiot's Guide to Multimedia*. 1994, Indianapolis: Alpha Books.
7. Hirata, K., et al. *Content-oriented Integration in Hypermedia Systems*. in *Hypertext '96*. 1996. Washington DC, USA, http://www.cs.unc.edu/~barman/HT96/P42/Hirata.htm.
8. Mathe, N. and Chen, J. *A User-Centered Approach to Adaptive Hypertext Based on an Information Relevance Model*. in *4th International Conference on User Modeling (UM '94)*. 1994. Hyannis, MA: http://ic-www.arc.nasa.gov/ic/projects/aim/papers/um94/cidum94.html
9. Nielsen, J., *Designing Web Usability: The Practice of Simplicity*. 2000, Indianapolis: New Riders Publishing. 432.
10. Pentland, A., Picard, R. W., and Sclaroff, S. *Photobook: Content-Based Manipulation of Image Databases*. in*: SPIE Storage and Retrieval Image and Video Databases II*. 1994. San Jose, CA, USA
11. Prieto-Diaz, R., *Classification of Reusable Modules*, in *Software Reusability: Concepts and Models*, T. J. Biggerstaff and A. J. Perlis, Editors. 1989, Addison-Wesley Pub. Co.: New York, NY. p. 99-123.
12. Prieto-Diaz, R. and Freeman, P., *Classifying Software for Reusabililty*. IEEE Software, 1987. **4**(1): p. 6-16
13. Rui, Y., Huang, T. S., and Chang, S.-F. *Image Retrieval: Past, Present, and Future*. in *International Symposium on Multimedia Information Processing*. 1997. Taipei, Taiwan, http://www.research.microsoft.com/~yongrui/html/publication.html.
14. Rui, Y., Huang, T. S., and Mehrotra, S. *Content-based Image Retrieval with Relevance Feedback in MARS*. in *IEEE International Conference on Image Processing*. 1997. Santa Barbara, California, http://www.research.microsoft.com/~yongrui/html/publication.html.
15. Rui, Y., Huang, T. S., and Mehrotra, S. *Relevance Feedback Techniques in Interactive Content-Based Image Retrieval*. in *IS&T and SPIE Conference on Storage and Retrieval for Image and Video Databases VI*. 1998. San Jose, CA, USA: SPIE, http://www.research.microsoft.com/~yongrui/html/publication.html.
16. Santini, S., Gupta, A., and Jain, R. *Querying Multiple Perspective Video*. in *SPIE Vol. 3656 Storage and Retrieval of Image and Video Databases VII*. 1998. San Jose, CA, USA, http://vision.ucsd.edu/~ssantini/articles/vsam/spie98vsam.ps.Z.
17. Santini, S. and Jain, R. *The "El Niño" Image Database System*. in *IEEE International Conference on Multimedia Computing and Systems*. 1999. Florence, Italy, http://vision.ucsd.edu/~ssantini/articles/imgdb/mmconfdb.ps.Z.
18. Santini, S. and Jain, R., *Integrated Browsing and Querying for Image Databases*. IEEE Multimedia, 2000. **7**(3), http://vision.ucsd.edu/~ssantini/articles/imgdb/Multi.ps.Z.

19. Shin, Y.-F. and Alessi, S. M., *Effects of Text versus Voice on Learning in Multimedia Courseware*. Journal of Educational Multimedia and Hypermedia, 1996. **5**(2): p. 203-218

20. Smith, J. R. and Chang, S.-F. *VisualSEEk: A Fully Automated Content-Based Image Query System*. in *ACM Multimedia 96*. 1996

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

22. Wexelblat, A. *History-Based Tools for Navigation*. in *32nd Hawaii International Conference on Systems Sciences (HICSS '99)*. 1999. Hawaii: IEEE Computer Society Press, http://wex.www.media.mit.edu/people/wex/HICSS-32.html.

23. Wexelblat, A. and Maes, P. *Footprints: History-Rich Tools for Information Foraging*. in *CHI'99*. 1999. Pittsburgh, Pennsylvania, USA, http://wex.www.media.mit.edu/people/wex/CHI-99-Footprints.html.

24. Wu, J.-K., *Content-Based Indexing of Multimedia Databases*. IEEE Transactions on Knowledge and Data Engineering, 1997. **9**(6): p. 978-898