Supporting Open Source Design for Multimedia Content

Alexandros Margaritis
Dept. of Digital Systems
University of Piraeus
Piraeus, 18534, Greece
amargar@unipi.gr

Nikitas M. Sgouros
Dept. of Digital Systems
University of Piraeus
Piraeus, 18534, Greece
sgouros@unipi.gr

Abstract
Open source principles and methodologies allow open access to both the design process and its products. This paper discusses a number of issues for supporting open source design of multimedia content. These environments should allow an unlimited number of users to modify existing media content and post their contributions on the net. In addition, they should allow users to visualize the current state of development in each project, select a subset of the various contributions and dynamically compose, view and share with other users new content versions containing all the selected contributions. This design process is supported through a web-services-based implementation developed in C# that is now freely available on the Web.

Keywords
Open source design, multimedia systems

ACM Classification Keywords
H.5.1 [Multimedia Information Systems]: Evaluation/methodology; H.5.4 [Hypertext/Hypermedia]: User issues.
Introduction

One of the most interesting developments with regards to the production and design of information technologies is the adoption of open source principles and methodologies that allow open access to both the development process and its products [1,2]. Although widely adopted in software development this approach has not yet achieved similar popularity in the production of multimedia content. This is unfortunate because the rise of multiple content distribution platforms (e.g. massive publishing sites like youtube, peer-to-peer repositories) has changed dramatically the way content is being distributed, managed and reviewed by its audience. More specifically, content consumption and review is now immediate and massive, leading to the creation of a large amount of content meta-information including audience reviews that may suggest useful ways to improve its quality. Consequently new creative possibilities for the design of multimedia material using open source methodologies open up, as experience has shown that there exist a large number of people with creative talent willing to alter or extend original multimedia content in order to improve its impact. These processes can be compared with the incremental creative processes by which folk cultural material and open source software are developed.

Furthermore, the adoption of open source methodologies in the generation of multimedia content can provide a financially viable alternative to the prevailing need for multimedia content. Supporting various forms of open source authoring can help multimedia distributors to tap into and support a vast pool of multimedia developers in financially meaningful ways.

Unfortunately, open source design of multimedia material is hampered by the scarcity of appropriate development environments that can provide geographically distributed creative teams with affordable and efficient ways of implementing their work and manage the fusion of original multimedia content with the modifications proposed and implemented by content users. This paper describes the design process supported by PATMOS1, a web-services-based multimedia editing and management system that allows users to:

- Post their reviews of available media content and synchronize the presentation of their comments with that of the original multimedia material.
- Visualize, select and access all or a subset of the reviews posted for a multimedia piece during its presentation.
- Enrich/alter available media content with additional material and post their modifications appropriately annotated.

PATMOS is freely available for testing and evaluation [3].

The Case Study

The open source design process we are looking into is open-ended, decentralized, incremental, participatory and asynchronous. More specifically, design is open-ended in the sense that there is not necessarily any explicit central message that the authors want to communicate. In our case the

---

1 Open-source Authoring Multimedia environments
message is emergent and it may change over time as more people get involved in a particular project. This means that developing a coherent message rests on the viewer of the work. Consequently and contrary to traditional multimedia delivery environments, viewing an open source project is active in the sense that the viewer is responsible for selecting those aspects of the work that are consistent with his own interpretation. In order to facilitate this process, PATMOS provides appropriate means to all the authors involved to describe their intentions and how they relate to the rest of the contributions. These descriptions allow the viewer to select parts of the available contributions and compose his own final content version.

Open source design is decentralized in the sense that there is not necessarily any central authority that will decide on the final version of the work during design. Furthermore, open source design is incremental and participatory since it allows authors to build on previous work and form creative teams. In this sense, open source authoring is a superset of existing forms of collaborative authoring and in order to work it should explicitly represent the history of contributions and the way they relate to each other.

Figure 1 displays an example of our conceptual version of the open source design process. In this case Author0 has supplied the original content version on which authors 1-3 have made their contributions. The contribution of Author1 uses not only the original content version but the contribution of Author2 as well. The original content version and all author contributions comprise the media content. Viewer1 composes a personalized version from the original content version plus the contribution of Author1. Because this contribution uses that of Author2 as well, Contribution2 will also be included in the personalized version generated by Viewer1.
PATMOS supports the conceptual scenario described above through a 3-tiered distributed system consisting of a database server, an expandable set of Web Services that can be hosted on other computers that have access to the Database Server and finally a set of applications that can reside in any computer(s) with access to the Web Services. Each Web Service is responsible for providing the user with content or for storing content provided by the users. In addition, each Web Service sends to a central Database Server that is invisible to the user meta-data describing the user contributions it has received. Furthermore, each Web Service communicates with other web services via the Database Server in order to receive content or content meta-data stored in them and relay it to the user.

The system implements the conceptual scenario described above as follows. Author$_0$ supplies to the system the original content version (OCV$_0$) which authors 1-3 subsequently modify through their contributions. PATMOS allows Author$_0$ to select and store his original content version along with any meta-data created by him to a Web Service from those available in the system. This Web Service automatically sends the meta-data information to the PATMOS Database Server. PATMOS stores only the meta-data describing the contributions made by authors 1-3 to OCV$_0$ in Web Services selected by these authors. These Web Services automatically send these meta-data to the PATMOS Database Server.

The user can compose and view a personalized version of this content by selecting a subset of the available contributions. For example, PATMOS allows Viewer$_1$ to compose a personalized version consisting of the original content version (OCV$_0$) modified by the contribution of Author$_1$. Because this contribution depends on that of Author$_2$ as well, PATMOS will automatically include Contribution$_2$ in the personalized version generated by Viewer$_1$. PATMOS then automatically composes in Viewer$_1$’s computer a new version of the material consistent with all these contributions and presents it to Viewer$_1$. The composition process is aided through the use of a 2-D tree view depicting all the existing design contributions for a content file organized as a dependency tree.

Lessons Learned

While posting novel author contributions is straightforward in an open source authoring environment, managing the removal or the updating of such contributions by their authors is significantly more complex. The main dilemma here is whether the system will continue to offer or not other contributions that depend on the ones selected for removal or replacement. Continuing to offer these dependent contributions means that the environment will have to maintain the contributions selected for removal or replacement against the will of their authors in order to preserve the consistency of the dependent work. On the other hand, the removal of deleted or replaced contributions will respect the will of their authors but the presentation of the work of other developers building on them will become problematic.
Another significant issue concerning the management of author contributions has to do with the desired distribution policy specified by each content provider. For example, each author should be free to specify the parts of his work that are open for modifications by other authors or to negotiate an agreement covering its use.

The design system should present alternative visualization methods to the authors and viewers able to capture both the dependencies, the timeline relations and the popularity of the proposed contributions in order to facilitate user participation and active viewing in the system. In addition the system should automate as widely as possible the annotation process needed to capture the dependencies between newly posted and already existing content in the system in order to facilitate user participation in the design process.

Conclusions
This paper discusses a number of significant research issues for the creation of a computational environment (PATMOS) supporting open source authoring and active consumption of media content. By presenting the final product along with its development history PATMOS encourages a style of viewing in which the process of creation is as significant as the final product. This can be an interesting alternative to traditional forms of multimedia content as the popularity of director’s cut releases of well-known movies has shown. A pilot version of PATMOS using the C# language and MySql is currently available on the Web for testing and evaluation [3].

Citations