

DiMaS: Distributing Multimedia on Peer-to-Peer File Sharing Networks

Tommo Reti, Risto Sarvas
Helsinki Institute for Information Technology HIIT
P.O. Box 9800, FIN-02015 HUT, Finland
[tommo.reti, risto.sarvas]@hiit.fi

ABSTRACT

This demonstration presents the Digital Content Distribution Management System (*DiMaS*). *DiMaS* proves as a concept that it is possible to make a system for multimedia producing communities to publish their work on highly popular P2P networks, and importantly, the system enables producers to insert content metadata, to manage intellectual property and usage rights, and to charge for the consumption. All this can be done *without* introducing another new content or metadata file format and a dedicated client application to read the format.

Categories and Subject Descriptors

H.3.1 [Information Storage and Retrieval]: Information Search and Retrieval; H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems

General Terms

Design, Languages.

Keywords

Multimedia, Metadata, Peer-to-Peer, Decentralized, Wireless

1. INTRODUCTION

Peer-to-peer (P2P) networks offer an alternative approach to distribute information products such as multimedia content. Unlike distributing information products in a central server architecture, decentralized P2P networks offer high availability, better bandwidth through many users' wideband connections, and better scalability without central servers as bottlenecks. The impact of P2P in file sharing is significant (see, *e.g.*, [2]). Hundreds of millions of copies are already in use making file sharing the most popular application on the World Wide Web. So, P2P networks create a huge content sharing base and a complex value network, which offers new business opportunities and models to various actors, such as network operators, and also, to new actors like multimedia producing communities. By the communities we mean amateur and professional movie, music, image *etc.* producers who are looking for alternatives to the traditional media publishing and distribution channels. Our example for this demonstration is a micromovie community that creates short movies for handheld devices.

Content communities want to distribute all kinds of multimedia in all kinds of file formats (*e.g.*, animations, e-books, movies and documents). However, the content producers want to make the viewing or consumption of the multimedia as easy as possible for the consumers. One way to do this is to produce multimedia that can be consumed without a dedicated client application.

At the same time, the content producers want to improve the searchability of the content by adding descriptive metadata, and to retain control over their content in the form of licenses, user rights, and usage charging. All this while using decentralized content distribution on uncontrolled P2P networks. Hence, before the P2P file sharing networks can really be harnessed and leveraged by multimedia producing communities there are several problems to be addressed: How to facilitate the input of content descriptions and control data into standardized metadata? How to bundle the metadata and the actual content into one package rather than having them in separate files? How to make this package easily accessible by the end-user without creating a new file format and another client application?

This demonstration presents the technical implementation of the Digital Content Distribution Management System (*DiMaS*) that takes into account the issues mentioned above. Since there is no single metadata language that provides universal and standard descriptors for content, rights, and pricing information for all content types, *DiMaS* codes the user input descriptions into XML metadata using specific standards for each separate set of information (*e.g.*, MPEG-7 for content description (see [1]), RDF for license text, and ODRL for user rights description and enforcement). This approach creates several different, very informative and browsable metadata files, while the actual content file is encrypted against unwanted usage. Even if the actual content is not encrypted, still metadata improves searchability and gives a chance to add a user license and rights to the content (*e.g.*, a CreativeCommons.org type of license). *DiMaS* introduces the concept of Distribution Package to bundle the metadata with the actual content into one executable file. The Distribution Package has a built-in logic and user-interface for browsing all metadata, viewing an optional preview file, and for purchasing the actual content for viewing. The system requires no dedicated client software for the end user other than Java Standard Edition Runtime Environment. In the following sections we describe the system design for *DiMaS* and the demonstration use case showing the basic use of the system.

2. SYSTEM DESIGN

The system is open-source and it has been produced using open-source tools. Figure 1 shows the main functional parts of the *DiMaS* with the initial data flows: 1) Content Import User-Interface, 2) Publishing System, 3) Distribution Package, and 4) Verification Server. There are also external interfaces to the system, various databases, and tools needed for administration.

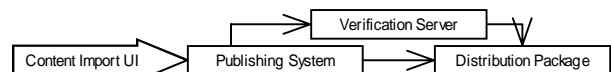


Figure 1. The four main functional system parts.

The Content Import User-Interface is the front-end for a content provider who wishes to publish his multimedia content through the system. The user-interface includes user registration, user information database, login, user manual, and the actual content import. The system encourages the content provider to fill in as many metadata fields as possible by appealing to the provider's need to promote his content. The fields are presented as a chain of web forms where the user can easily move back and forth, like in web store shopping carts. The user can also insert an optional preview file, which can be anything from introducing the content to pushing otherwise the user to purchase the actual content file. When all metadata and files are inserted, they are stored to the system until the user decides to publish the inserted content by pressing the respective button. An administrator of the system can view all the inserted and published content, and optionally moderate the publishing if necessary.

After publishing, the inserted content file, the preview file, and the respective metadata are forwarded to the Publishing System which bundles them into a Distribution Package and executes the actual publishing to computer networks, such as P2P. The Publishing System is a group of separate modules, which encrypt the content file, create a hash out of the encrypted content file, and turn the inserted metadata into machine-readable XML. All this and the logic code are implemented into an executable JAR file, which is called the Distribution Package (see Figure 2). The XML description and the decryption key are sent to the Verification Server for enabling secure purchasing and metadata updates even after the publishing.

Both Publishing System and Distribution Package have a flexible structure that can be customized to better fit a multimedia community's requirements. For example, by changing the routine in a single module the rights expression language can be changed from the used ODRL to another well-known alternative, XrML. The Distribution Package is made available to as many P2P and other distribution channels as possible. Because the actual content file is encrypted, Distribution Packages can be stored even on servers that are not under the publisher's control. This creates a decentralized storage with an mirroring system, which guides a searching user to the nearest source of the desired content. Usually multimedia content is found from a P2P file sharing network by a file name. However, the Distribution Package supports also the future situations where P2P applications can use the included metadata and offer better searching and rights ruling capabilities. For now, the system's Content Search user-interface simulates such metadata driven search by combining view and metadata based search and browsing (see, e.g., [3]).

In the last step of the content distribution process, the goal is that wherever a user finds a Distribution Package, he launches it on a terminal with a single double-click. In other words, the user needs no system specific client program to view the Distribution

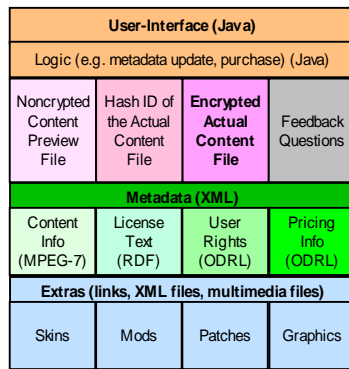


Figure 2. The structure of the Distribution Package.

Package. At the moment the system supports a limited number of content types (e.g., the file types of MS Media Player) and only terminals with Java run-time environment, such as PCs and PDAs.

3. DEMONSTRATION USE CASE

In the use case, a content producer browses via PC onto the main page, where he registers into the system. After registration he logs on and wants to import a micromovie that he has created on his PC. He goes through the steps of the content description import pages, and at the end approves the inserted information and media files. Looking at his personal page he can see the names of the imported content. He decides to publish the newest content with the descriptions he just inserted and the preview file he uploaded. The system announces that the publishing is successful and a Distribution Package is created successfully. The content producer logs off.

In the second part of the use case, another user, a consumer, browses via PC the content selection on the Content Search page. She tries several different search categories (e.g., action and video), browses descriptions of the contents, checks one micromovie preview, and finally decides to download an actual micromovie (i.e., Distribution Package). When she has finished downloading, she moves the content to her attached PDA. Later, she opens the micromovie on her PDA with one click. The user-interface, which is similar to a DVD menu (see Figure 3), opens



Figure 3. Distribution Package on iPAQ.

and it asks if the user wants to update the content metadata. This means that the Distribution Package has found a network connection and offers to update the content description. The user browses through the content description, including the pricing information. Before a purchase decision she wants to check the preview again, and after that she buys the content by pressing the "Buy" button on the user-interface. The PDA retrieves a decryption key from the Verification Server and informs of the successful purchase. The user notices how the price page shows the new licenses she just bought, i.e., the number of times the user can view the micromovie, and the play button activates from gray to green. The user presses the "Play" button and waits for the content to decrypt. After watching the micromovie, the user fills in a feedback form, which is submitted to the Verification Server at once, and the metadata on the PDA is updated considering the new feedback, (e.g., the rating can be changed due to the user's feedback). The user closes the Distribution Packet on her PDA.

4. REFERENCES

- [1] Martínez, J.M., Koenen, R., Pereira, F. MPEG-7: The Generic Multimedia Content Description Standard, Part 1. IEEE Multimedia 9(2), 2002.
- [2] Strumpf K., Oberholzer F., The Effect of File Sharing on Record Sales: An Empirical Analysis, working paper at www.unc.edu/~cigar/papers/FileSharing_March2004.pdf
- [3] Yee, K-P., Swearingen, K., Li, K., and Hearst, M., Faceted Metadata for Image Search and Browsing, In Proc. of CHI 2003, ACM Press.