A Metadata Model for Peer-to-Peer Media Distribution

Christian Timmerer[1], Michael Eberhard[1], Michael Grafl[1], Keith Mitchell[2], Sam Dutton[3], and Hermann Hellwagner[1]

[1] Klagenfurt University, Multimedia Communication Group
{*firstname.lastname*}@itec.uni-klu.ac.at
[2] University of Lancaster, Computing Department, InfoLab21
k.mitchell@lancaster.ac.uk
[3] BBC R&D Prototyping
Sam.Dutton@bbc.co.uk

Abstract. In this paper we describe a metadata solution for a Peer-to-Peer (P2P) content distribution system termed NextShare. We outline the key motivating factors for our approach, detail the overall generic architecture we have developed and present the workflow for delivering metadata through Peer-to-Peer based content distribution. The paper also presents the metadata model we have developed and we describe in detail how all the content can be packetized and distributed using NextShare. Finally, a description of the core and optional metadata attributes which may be utilized within the system is provided.

Keywords: Metadata, peer-to-peer streaming, social networks, payment and cashing, advertisements, MPEG-21 Digital Item, API.

1 Introduction

The Internet is increasingly being used to distribute both real-time and on-demand high bandwidth multimedia content to large audiences due, in part, to the increase in bandwidth available within the last mile. The server and bandwidth costs for provisioning adequate resources to facilitate high Quality of Experience (QoE) for streaming services to the end user are rapidly increasing as High-Definition (HD) content becomes increasingly dominant. One alternative to the traditional client-server or Content Distribution Network (CDN) approach is provided by Peer-to-Peer (P2P) distribution systems. In a decentralized P2P system the notion of a server does not really exist and instead all peers/nodes within a network are capable of distributing content to other peers while simultaneously consuming content. Thus, the distribution cost is shared amongst the peers themselves, potentially significantly reducing the distribution costs to content providers.

Although the number of available P2P systems is already very large and new P2P systems are frequently developed, the metadata utilized within P2P systems is usually proprietary and has not yet been standardized. In this paper, we propose a metadata model that builds upon existing standards and extends them to support P2P-specific requirements. The requirements are motivated by our work in the EU funded P2P-Next project [1] which seeks to develop an open source, standards-based P2P content

distribution platform, herein referred to as *NextShare*. The metadata model provides a solution to describe the content in P2P systems as well as a solution for structuring and packetizing the metadata and the actual audiovisual (A/V) content.

The remainder of this paper is organized as follows. Section 2 describes the general architecture and metadata workflow before we present the metadata model with its core and extensions in Section 3. Section 4 provides details on the metadata specification we have developed which is followed by an overview of the API for creation and access of metadata in Section 5. Some concluding remarks are presented in Section 6 along with some areas of future work.

2 Architecture and Workflow

This section describes a generalized architecture and workflow enabling P2P media distribution as depicted in Figure 1. The media to be distributed via P2P consists of (A/V) content and metadata. The metadata is divided into core metadata and additional (optional) metadata (i.e., extensions of the core metadata; cf. Section 3 for more details). Finally, the relationship between the A/V content and the metadata is described by structural metadata that represents the declaration of the P2P media. The combination of all these assets is referred to as the P2P-Next Item depicted in Figure 1 and is defined as follows: A/V content + metadata (core + extensions) + structure. The torrent file is generated based on these assets from the P2P-Next Item and is used for distribution via the NextShare platform [1]. Additionally, the metadata of the P2P-Next Item is used within the Atom/RSS store for the presentation/interactivity layer at the content consumption side.

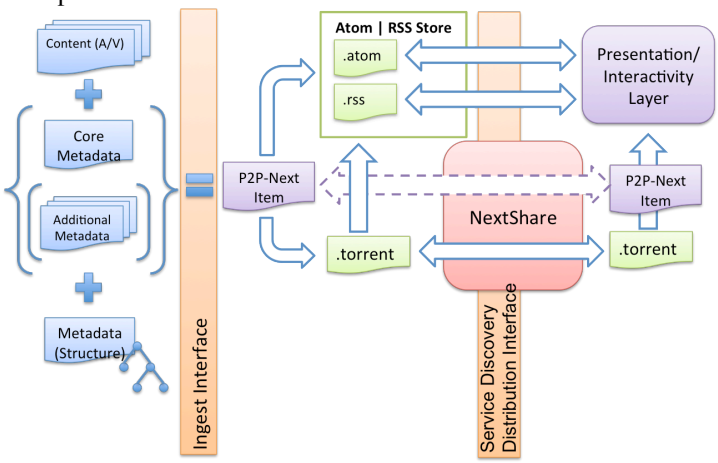

Figure 1. Architecture and Workflow for P2P Media Distribution.

A key requirement for the P2P-Next project is to ensure backwards compatibility with other BitTorrent clients. Thus, a torrent file compatible with the BitTorrent protocol [2] needs to be provided as top-level information. However, torrent files contain only a small portion of the metadata (i.e., the core metadata) needed to represent rich media content. The high-level structure of the P2P-Next Item is shown in

Figure 2. The torrent file contains the references and hash values for the media resources, i.e., the MPEG-2 Transport Stream (TS) containing the video and the audio content, and possibly the Scalable Video Coding (SVC) enhancement layers in case the video content is scalable. In particular, an MPEG-21 Digital Item Declaration (DID) [3] is included in the torrent file which might be encoded as binary XML or just provided as plain XML. The DID included in the torrent file contains the core metadata from the P2P-Next Rich Metadata specification [4] and references to other, optional, metadata and resources packaged separately. One way to package the optional content would be to packetize it into an MPEG-21 file (.m21) [5], including an additional DID that describes the individual optional content items. Alternatively, the additional DID could be provided on its own (i.e., as plain XML) and reference the optional content items, which could be, e.g., distributed through the NextShare system. The main reason for storing only the core metadata directly in the torrent file is to keep the size of the torrent file as small as possible (by referencing the other data) and still provide sufficient data to enable search on the content of the torrent file. A more detailed description of the structure of the DID is provided in the next section.

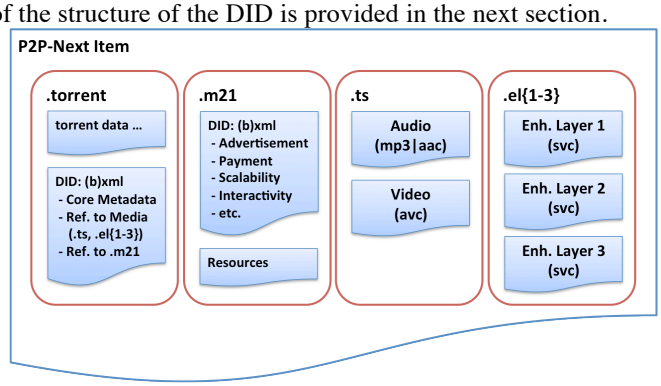

Figure 2. High-Level Structure of P2P-Next Item.

The DID for the optional content (packetized into an .m21 file in Figure 2) provides access to optional metadata related to advertising, media rating/review, content provider, payment, scalability, or interactivity. Furthermore, the metadata required for interactivity references additional resources (text, images, small audio/video clips, etc.) which could be included within an MPEG-21 file. Note that the MPEG-21 file format is based on the ISO base media file format that also provides the foundation for the well-known MP4 file format.

The actual A/V content is multiplexed within an MPEG-2 TS and is encoded with Advanced Video Coding (AVC) and MPEG-1 audio layer 3 (MP3) or Advanced Audio Coding (AAC) respectively. Additionally, enhancement layers for the video content are provided as separate bit-streams which are encoded with SVC having the base layer as part of the MPEG-2 TS.

3 Metadata Model

The P2P-Next Item contains all media resources, metadata, and possible additional data related to a single digital object. The P2P-Next Item is described by a DID which structures content and metadata. The DID contains relevant metadata and provides references to media resources and distributed metadata. The core metadata from the P2P-Next Rich Metadata (RM) specification are directly included in the top levels of the DID. All optional RM, such as for payment, advertising, media rating/review, and scalability, are referenced via XML Inclusions (XInclude) [6], as these metadata are only required for specific services and might be provided separately, e.g., on a secure server for payment. Additionally, data for interactivity is also referenced by means of XInclude. Within the P2P-Next project, the interactivity layer is termed LIMO (Lightweight Interactive Media Objects). LIMO data uses features of HTML 5 [7] and may include HTML, JavaScript code, style sheets, and/or additional media resources. As it is difficult to include media resources into XML files (i.e., base64 encoding is not a feasible solution for large media resources), the LIMO content can be packetized into the MPEG-21 file format, which contains another DID that references all the LIMO content within the MPEG-21 file. In this way, the part of the DID that is stored directly in the torrent file is kept as small as possible while retaining a complete DID structure and conveying the RM core metadata inside the torrent file for increased search performance. Thus, there are two (or more) DIDs within the NextShare system representing the P2P-Next Item.

The main (or master) DID conveys the overall structure of the P2P-Next Item and core metadata. It is stored inside the torrent file. Any additional DID can be stored in the XML box of an MPEG-21 file and can contain the additional metadata as well as data for LIMO. The torrent file references that MPEG-21 file. The main document references parts of the additional document as described above. However, the conceptual DID model for the P2P-Next Item should be seen as one entity, only its physical representation is split into two (or more) documents. Please note that the usage of an MPEG-21 file is not mandatory. The second DID document does not necessarily need to be stored within an MPEG-21 file. If it is more advantageous to distribute the LIMO content and the additional metadata in separate files, the DID just needs to reference these files and they can be provided separately, e.g., through the NextShare system or on traditional servers. This approach is especially useful when only parts of the LIMO content might be required for playback for the user. The MPEG-21 file just provides one way to store all the additional data together, if such a packaging mechanism is desired.

Figure 3 outlines the metadata model based on MPEG-21 DID for a P2P-Next Item and, thus, for P2P media distribution. The building blocks of a DID are shown as part of the legend at the bottom of the figure. Shapes with a dotted outline indicate data that are not included directly within the main DID, but are included by reference, in order to ensure the small file size of the torrent file.

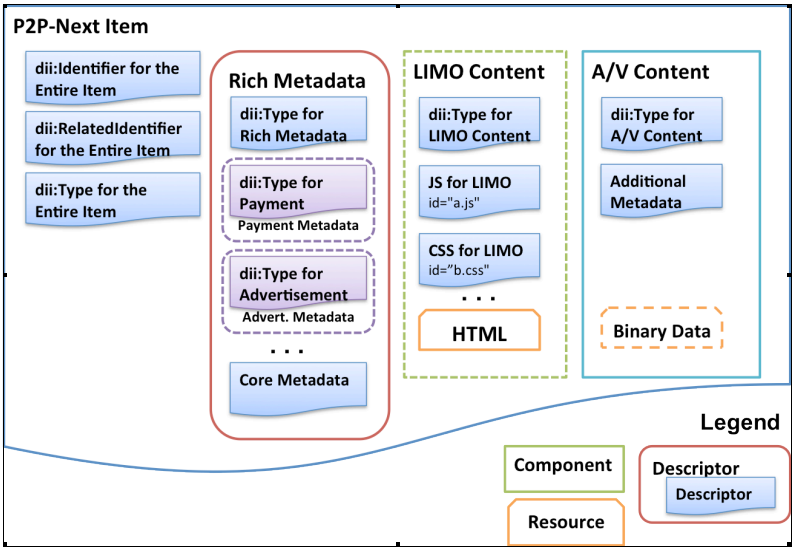

Figure 3. Metadata Model for P2P Media Distribution.

An `Item` represents the P2P-Next Item. The `dii:Identifier` is an MPEG-21 Digital Item Identifier (DII) [3] for the entire item, e.g., the Uniform Resource Name (URN) "urn:p2p-next:item:bbc-bbcone-b00n9p5x". The `dii:RelatedIdentifier` enables identification of the underlying work described by a Digital Item (DI). In this case, the `RelatedIdentifier` defines an "isAbstractionOf" relation to the underlying media content, e.g., identified by "urn:bbc:bbcone-b00n9p5x". This underlying media content is independent of the P2P-Next system. The `dii:Type` identifies the structure of this DID. It is set statically to the URN "urn:p2p-next:type:item:2009", thus determining the position within the DID and which building blocks are allowed. The structure of this DID is defined as part of [8].

The RM is represented following the Rich Metadata specification [4]. The actual RM is contained within a `Descriptor` and is structured as follows. The core metadata are contained within a `Statement` which is typically the last element in this `Descriptor`. Some nested `Descriptors` precede that `Statement`. The first one contains the `dii:Type` for the core metadata. This `Type` is set statically to "urn:p2p-next:type:rm:core:2009" and identifies the structure of the RM core metadata within a P2P-Next Item.

All additional RM assets are contained in further nested `Descriptors` which are referenced via XInclude. For each Descriptor, an `xi:include` element points to a `Descriptor` in the additional DID document. Each of these `Descriptors` has a `dii:Type` to identify its structure and consequently its purpose. Each `Descriptor` contains a `Statement` which conveys the RM representation of the appropriate RM part. More details on the URNs defined and utilized within NextShare are provided in Annex A of [8].

There are two `Components` in the DID for the P2P-Next Item. The first `Component` contains all data for LIMO. It is stored in the additional DID document inside a separate MPEG-21 file. Furthermore, it is included into the main document by means

of an xi:include element. The dii:Type in its first Descriptor is set statically to the URN "urn:p2p-next:type:limo:2009". Further Descriptors contain resources required by the actual LIMO resource (i.e., the HTML content). These resources may be JavaScript files, Cascading Style Sheets (CSS) as well as JPEG or PNG images. Each Descriptor contains an id attribute, uniquely identifying that resource within the P2P-Next Item. It is proposed to use the original file name of a resource in all lower-case characters for the id value if applicable. The resources with text content are contained directly in a CDATA section of the Resource. On the other hand, resources with binary content, such as JPEG images, are bundled in the MPEG-21 file. The Resource of the Component contains the HTML page representing the LIMO content. As the required resources are not in actual files but rather in one MPEG-21 wrapper file, all references to the original files have to be replaced in the HTML document by the corresponding ids. For example, the reference to the file "script.js" would be replaced by "#script.js", a reference to the Descriptor with the id "script.js". These replacements could be accomplished by means of an XSL Transformation (XSLT). The HTML document for LIMO is contained in the CDATA section of the Resource.

The second Component in this Item represents the actual media content. The media content shall typically be packed into an MPEG-2 TS. The Component contains a Descriptor with a dii:Type which is set to "urn:p2p-next:type:content:2009". Furthermore, a second Descriptor may be present, conveying technical metadata about the TS (such as bitrate, size, etc.). The actual binary data of the TS is referenced through a Resource within the Component.

For examples of such DIDs the interested reader is referred to [8].

4 Metadata Specification

This section provides an overview of the core and optional metadata utilized within the NextShare system. More information on the presented metadata can be found in [4].

4.1 Core Metadata

The core metadata contains the essential content-related information that is required to search for a specific P2P-Next Item. All of the core metadata attributes need to be static, as these attributes are included within the torrent file and cannot be changed during the lifecycle of the torrent's swarm. The core metadata attributes include among others the title, the description, the genre, the creator, and also technical attributes like the resolution and the frame rate of the A/V content.

4.2 Optional Metadata

The optional metadata contain information that is related to the content and required for specific services, but might not always be needed for a certain P2P-Next Item. The individual optional metadata types are briefly described below.

– *Advertising* describes which types of advertisement are suitable and may be included in the P2P-Next Item. The attributes include the formats and advertisement types that may be used within the P2P-Next Item as well as information about the target group for advertising.

– *Payment* describes if the content is available for free or has to be paid for. The attributes include the price, the payment options and recipient, and information about possible donations.
– *Scalability* is only required if the video content is scalable and describes the properties of the scalability layers.
– *Media Review* reflects the perception of content by the audience such as user ratings. RM supports two kinds of media reviews. Individual media reviews performed by individual users and aggregated media reviews which summarize sets of individual media reviews. A media review comprises annotation of content by means of keyword tags and textual reviews, rating of content, rating of identity (e.g., to mark inappropriate content or spam), and rating of quality. The individual media review is always linked to its reviewer in order to prevent abuse.
– *User Profile* consists of core information (such as name and contact information) and additional information. The additional information ranges from usage preferences over usage history to social metadata. Social metadata comprises various kinds of information that a user might want to share with a specific subset of users, e.g., personal interests, education, employment, and information on social relations. User profile information is not included within the P2P-Next Item but is conveyed through other channels (e.g., a secure server). However, the P2P-Next Item contains a reference to its content provider.

5 Application Programming Interface

In order to utilize the above metadata model we have defined an Application Programming Interface (API) that serves two major purposes: Firstly, during the creation of the torrent file, the API is utilized to create the MPEG-21 DID describing the content package. Secondly, during the distribution of the content in the NextShare system, the API is utilized to access the different parts of the DID such as the core metadata. The API is implemented in C++ using the CubeWerx BXML library [9] and provides Python bindings for usage with the content ingestion tools implemented in Python (see Section 3.2.3 of [8]).

The API for NextShare provides two major interfaces. The *DIDCreator* enables creation of MPEG-21 DIDs containing the core metadata, references to optional metadata and to media resources, while the *DIDParser* allows extraction of the content from DIDs.

Due to space restrictions only an excerpt of the DIDParser API is provided below; for details the interested reader is referred to [8].

Method	Description
`const CW_XML_NODE* getContentByDiiType (CW_XML_SCAN* xmlScan, const char* diiType)`	Extracts the content from the DID document based on the specified `dii:Type`.
`const CW_XML_NODE* getContentByID (CW_XML_SCAN* xmlScan, const char* id)`	Extracts the content from the DID based on the `Descriptor`'s `id`. The `Descriptor` `id` is only utilized for LIMO content.

Interestingly, as the `getContentByDiiType` method returns parts of the DID identified by a standardized URN (i.e., based on the specified `dii:Type`), these parts can be accessed (i.e., parsed) by a software module conforming to the API that corresponds to the given `dii:Type`. For example, the parts of the DID representing the core rich metadata (i.e., identified by "urn:p2p-next:type:rm:core:2009") are handed over to the core RM API implementation and LIMO (i.e., identified by "urn:p2p-next:type:limo:2009") is forwarded to the LIMO implementation. In this way, interoperability is guaranteed through the usage of standardized (meta-)data formats and well defined APIs and URNs.

6 Conclusions and Future Work

In this paper we have presented the architecture and metadata model utilized within the NextShare system. In particular, we have shown how the A/V content, the core and optional metadata as well as the metadata structure can be packetized into a P2P-Next Item and distributed through the NextShare system. The major advantage of our approach is *interoperability* – thanks to the usage of existing, standardized representation formats for both media and metadata – and *backwards compatibility* to the well-known BitTorrent protocol.

NextShare including our metadata model and the core metadata (cf. Section 4.1) has been successfully demonstrated at various events (e.g., IBC'09, NEM-Summit'09; cf. [1] for details) and is currently being evaluated within the P2P-Next Living Lab [10]. Future work includes the complete definition, design, implementation, and validation of the optional metadata as outlined in Section 4.2 and full support of LIMO content within our metadata framework.

Acknowledgments. This work was supported in part by the European Commission in the context of the P2P-Next project (FP7-ICT-216217).

References

1. The P2P-Next Project, http://www.p2p-next.org
2. B. Cohen, "The BitTorrent Protocol Specification", http://www.bittorrent.org
3. Burnett, I.S., Davis, S.J., Drury, G.M.: MPEG-21 Digital Item Declaration and Identification - Principles and Compression. IEEE Transactions on Multimedia, vol. 7. no. 3, pp. 400--407 (2005)
4. P2P-Next D5.3.1, "Tools for Rich Metadata and Signposts", 2009, http://www.p2p-next.org/download.php?id=C48E37B2671AE3F3B8459942B2C7E484
5. Singer, D., Visharam, M.Z.: MPEG-21 File Format White Paper, ISO/IEC JTC 1/SC 29/WG 11N7925, Montreux (2006)
6. W3C Recommendation, "XML Inclusions (XInclude) Version 1.0", http://www.w3.org/TR/xinclude/
7. W3C Working Draft, "HTML5", 25 February 2010, http://dev.w3.org/html5/spec/Overview.html
8. P2P-Next D5.2.1, "Content Packaging, Ingestion, and Adaptation", 2009, http://p2p-next.org/download.php?id=98079FB406C2CBD78766E20BBF531F4E
9. CubeWerx BXML library, http://www.cubewerx.com/bxml
10. P2P-Next Living Lab, http://www.livinglab.eu/