MPEG-21 Digital Items in Research and Practice

Christian Timmerer
Klagenfurt University
Universitätsstrasse 65-67
A-9020 Klagenfurt
+43 (463) 2700 3621
christian.timmerer@itec.uni-klu.ac.at

Hermann Hellwagner
Klagenfurt University
Universitätsstrasse 65-67
A-9020 Klagenfurt
+43 (463) 2700 3612
hermann.hellwagner@itec.uni-klu.ac.at

ABSTRACT

The aim of the MPEG-21 standard, the so-called Multimedia Framework, is to enable transparent and augmented use of multimedia resources across a wide range of networks, devices, user preferences, and communities, notably for trading (of bits). As such, it provides an important step in MPEG's standards evolution, i.e., the transaction of Digital Items among Users. This paper provides an overview of applications making use specifically of MPEG-21 Digital Items and a more in-depth presentation of a few selected applications in research and practice.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software – *distributed systems, information networks.*

General Terms

Your general terms must be any of the following 16 designated terms: Management, Design, Standardization, Languages.

Keywords

MPEG-21, Digital Item, Usage, Application, UPnP, Digital Library, DANAE, ENTHRONE, P2P-Next.

1. Introduction

The core MPEG-21 specifications [1] have been available for many years. However, apparently their wide scope and complexity seem to make it difficult for enterprises to exploit MPEG-21 technologies for their business directly.

To the best of our knowledge, only a few companies exist that provide MPEG-21 tools or base their products on MPEG-21. Two companies that seem to use a wide range of MPEG-21 standards are Adactus [2] and Enikos [3]. Adactus, with their Mobilize platform, offer a content packaging, adaptation, and delivery system mainly targeting mobile devices. Several media resources can be packaged into a Digital Item; during distribution, e.g., commercials can be added as metadata; the content can be adapted (transcoded) to the end user device; and the commercials can be personalized to the end user or based on his/her current location, according to the Adactus Website. Apparently, this application is mainly based on Digital Item Declaration (DID) [4] constructs, but also Digital Item Adaptation (DIA) [5] descriptions seem to be made use of. Enikos provides the Enhanced Media Platform (EMP), a solution by which content creators and distributors can bundle their digital audio and video media, metadata, and other digital assets, e.g., comments, links to dynamic content, and advertising, to form a rich and partially interactive multimedia experience for the end user. The content can also be contextualized and repurposed (adapted) for the user and his/her device, respectively. Apparently, again DID and DIA constructs seem to be utilized in the EMP.

This paper provides an overview of applications making use of MPEG-21 and a more in-depth presentation of a few selected applications and use cases.

Before going into details, Section 2 describes the abstract model of the Digital Item in order to provide the necessary background, i.e., the foundations on which MPEG-21 and Digital Items are built upon.

In Section 3, we will present one of the first adoptions of Digital Items, namely Universal Plug and Play's (UPnP) DIDL-Lite [6] which is derived from a subset of MPEG-21 DIDL. It is basically used as a container format within UPnP's content directory and enhanced with UPnP-specific data (e.g., media resource attributes such as bit rate, resolution, size, etc.) and Dublin Core metadata. We will provide a brief description and a critical review thereof, in particular, whether and how interoperability between DIDL-Lite and DIDL can be achieved.

Section 4 will review a significant application of MPEG-21 core concepts, namely of Digital Item Declaration (DID), Digital Item Identification (DII) [4], and Digital Item Processing (DIP) [7], which interestingly is not in the core multimedia area. Rather, the use of MPEG-21 technology for representing, storing, managing, and disseminating complex information assets in a digital library has been reported in the literature and will be described here [8][9].

Several EC-funded projects (e.g., DANAE [10], ENTHRONE [11], and P2P-Next [12]) have adopted a wide range of MPEG-21 technologies and provided reference applications on top of it. In Section 5 we will describe these adoptions from an application's point of view.

Finally, Section 6 will provide the conclusions.

2. Abstract Model of the Digital Item

In the following, we will review the abstract DID model which is defined using the Extended Backus-Naur Form (EBNF) [13] and shown in Listing 1.

We will only give a brief description of the elements required for the following sections; for the exact semantics, the interested reader is referred to [14].

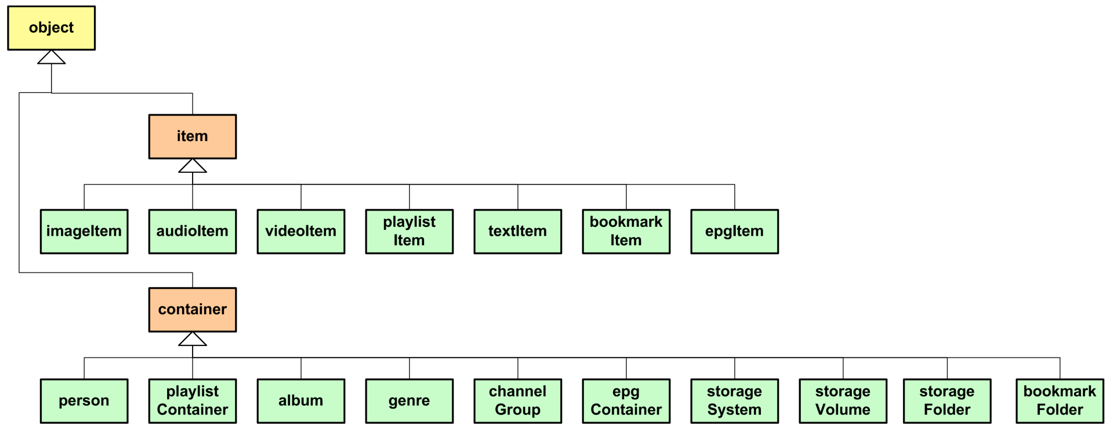

Figure 1. UPnP DIDL-Lite Metadata Model.

Listing 1. Abstract DID Model.

```
container  ::= descriptor* container* item*
item       ::= condition* descriptor* choice*
               (item|component)* annotation
component  ::= condition* descriptor* resource
               anchor
anchor     ::= condition* descriptor* fragment
descriptor ::= condition* descriptor*
               (component|statement)*
condition  ::= predicate+
choice     ::= condition* descriptor* selection+
selection  ::= condition* descriptor* predicate
annotation ::= assertion* descriptor* anchor*
assertion  ::= predicate*
```

In practice, a Digital Item is referred to as a collection of *items* and *containers*, respectively, which may comprise further *items* and *containers* in a hierarchical way. The *container* allows for defining groups of *items* and/or *containers* which may form logical packages or shelves whereas the *item* is primarily used as a collection of *components* bound to a set of relevant *descriptors*. However, an *item* may contain sub-*items*, which is then considered as a compilation of *items*.

A *component* is the binding of a *resource* (i.e., assets such as a movie, video/audio clip, image, text) to a set of *descriptors*. These *descriptors* contain usually structural information (e.g., bit rate, frame rate, resolution) about the resource, but do not describe its content.

A *descriptor* associates information (i.e., metadata) with the enclosing element (e.g., *item*, *container*, *component*). Interestingly, a *descriptor* may contain a *component* including a *resource* that provides descriptive information about the enclosing element such as a thumbnail of an image, summary of a video/audio clip, or text in form of a PDF document.

The combination of *choice*, *condition*, *predicate*, and *selection* enables a so-called choice/selection mechanism which may be used to make the static declaration of a Digital Item configurable. That is, parts of a Digital Item can be declared in a way that they are only conditionally available depending on a predefined context. For example, a Digital Item could be declared such as to deliver a high/medium/low-quality version depending on the user's subscription type (e.g., high-quality for paying customers vs. low-quality free of charge) and/or end device capabilities (e.g., flat screen vs. smart phone). Thus, these elements provide a very powerful mechanism for introducing flexibility within the media value delivery chain including associated networks.

The abstract DID model has been implemented using XML Schema resulting in the Digital Item Declaration Language (DIDL).

There is one noticeable difference between the abstract DID model and DIDL which is the `DIDL` root element. The `DIDL` root element provides the entry point for the Digital Item Declaration (DID) which may be either a `Container` or an `Item` and possibly prefixed by `DIDLInfo` and/or `Declarations`. The former is used to convey application domain-specific information about the DID and the latter is used for declaring a set of DIDL elements without instantiating them, e.g., for later use within this document through a well-defined referencing/inclusion mechanism. `Container` and `Item` are more or less an implementation of the abstract DID model as shown in Listing 1 without noticeable differences.

Finally, it is important to note that MPEG deliberately has not defined which kind of resources or metadata shall be included within a Digital Item and how. It is expected that this should be specified within the respective application domains. Thus, the MPEG-21 Digital Item model is very powerful, generic, and flexible that shall allow for the adoption in almost every application domain including those not related to multimedia (e.g., cf. Section 4).

3. DIDL-Lite

One of the first adoptions of MPEG-21 concepts was within the UPnP forum where DIDL-Lite [6] has been proposed shortly after the specification of the abstract Digital Item Declaration (DID) model and its implementation, namely the Digital Item Declaration Language (DIDL) [4]. In fact, the DIDL-Lite schema aims to be an implementation of the abstract DID model using XML Schema, but with a slightly different syntax compared to the DIDL schema. The metadata model of UPnP DIDL-Lite is shown in Figure 1.

The base class from which all other classes are derived is called *object* which is an abstract class and, thus, cannot be instantiated similarly to `DIDBaseType` as defined within `urn:mpeg:mpeg21:2002:02-DIDMODEL-NS`. Direct sub-classes of *object* are *item* and *container* which can be instantiated through the corresponding DIDL-Lite schema elements `item` and `container` respectively. *Item* and *container* classes can be further refined as shown in Figure 1 and are indicated by the `upnp:class` element which is a child element of `item` and `container`. That is, the nature of the *item* or *container* can be specified, e.g., a photo, music track, movie, photo album, etc., which are defined as sub-classes of leaf nodes in Figure 1 but not shown due to space constraints. Note that only leaf nodes (not shown in Figure 1) are allowed within the `upnp:class` element. This functionality provides an extension of the DID model with respect to the application domain within which UPnP operates.

Further extensions of UPnP DIDL-Lite compared to the original MPEG-21 DID model are the adoption of Dublin Core metadata (dc) [15], various properties (`upnp:forContainer`), and certain attributes for the actual resources (`res-attributes`). A simplified EBNF syntax for UPnP DIDL-Lite is shown in Listing 2.

Listing 2. Simplified EBNF for UPnP DIDL-Lite.

```
upnp:forContainer ::= <some properties>
upnp:forItem ::= upnp:forContainer
allowed-under-container ::= upnp:forContainer | dc
        | desc | item | container | res
allowed-under-item ::= upnp:forItem | dc | desc |
                       res
upnp:class ::= [object.item]
  [object.item.imageItem] ...
  [object.container] [object.container.person] ...
  [object.container.bookmarkFolder]

container ::= dc:title allowed-under-container*
              upnp:class allowed-under-container*
item ::= dc:title allowed-under-item* upnp:class
         allowed-under-item*

res-attributes ::= protocolInfo [importUri] [size]
   [duration] [bitrate] [sampleFrequency]
   [bitsPerSample] [nrAudioChannels] [resolution]
   [colorDepth] [tspec] [allowedUse]
   [validityStart] [validityEnd] [remainingTime]
   [usageInfo] [rightsInfoURI] [contentInfoURI]
   [recordQuality] [protection]
res ::= anyURI res-attributes

desc ::= any

DIDL-Lite ::= ( item | container | desc )*
```

While the `DIDL-Lite` root element has some similarities with the `DIDL` root element, it becomes apparent that the `container` and `item` elements, although present in both models, are quite different from their MPEG-21 DIDL counterparts. In particular, a `container` may equally include resources and a nested structure of further `containers` and/or `items` while an `item` may contain only resources (i.e., descriptive information is excluded for the moment). That is, the UPnP DIDL-Lite `item` element comes close to an MPEG-21 DIDL `Component` element and the UPnP DIDL-Lite `container` represents a combination or mixture of MPEG-21 DIDL `Container` and `Item` elements.

The UPnP `res` element has additional attributes providing structural information about the resource which is similar to MPEG-21's `Descriptor` within a `Component`. Thus, the UPnP `res` element can be seen as equivalent to the MPEG-21 `Component` but with a restricted number of attributes which are shown in Listing 2.

UPnP DIDL-Lite, item, and container may have a `desc` element which is equivalent to the MPEG-21 `Descriptor` element except that the UPnP version can only convey XML data whereas MPEG-21 offers more possibilities (e.g., a `Component` with a `Resource`).

Finally, each UPnP `item` or `container` may include a set of properties providing descriptive information about the enclosing element – maybe a reason why `desc` is not used in practice. These properties include information related to the contributor (e.g., artist, actor), affiliation (e.g., genre, album), associated resources (e.g., album cover art, lyrics), general description (e.g., playback count, last playback position), etc.

4. Information Asset Management in a Digital Library

Digital libraries today must host, and make accessible, a vast amount of material that is usually structured as complex digital objects. Such objects aggregate the information assets proper, which may be of a wide variety of media types, as well as metadata that describe the information assets and support their identification, retrieval, dissemination, and the management of the digital rights associated with them. The format to represent, store, manage, and disseminate the complex digital objects is therefore of significant importance for the flexibility and success of a digital library architecture.

Interestingly, researchers of the Los Alamos National Laboratory (LANL) Research Library have chosen MPEG-21 concepts as important building blocks in one of their projects, the aDORe repository effort [16]:

— Digital Item Declaration (DID) as one of the formats for representing (and serializing) complex digital library objects and

— Digital Item Identification (DII) for the identification of DIDs and assets therein [8][9][16].

— Also, Digital Item Processing (DIP) was proposed to dynamically add processing information to DIDs when disseminating digital objects [9].

The LANL Research Library mainly accommodates scholarly data, locally stores Terabytes of raw material, and has to deal with tens of millions of digital assets to be managed by the digital library. As of 2008, the aDORe Archive repository contained more than 100 million digital objects [18].

The main reasons for adopting MPEG-21 concepts were:

— The ability of DIDs to represent media data of any type together with metadata in a structured way, based on a well-defined abstract model.

— The applicability of the open, modular MPEG-21 multimedia framework to the requirements of digital libraries. This mainly pertains to DID and DII for the representation of digital objects and the inclusion of digital library specific metadata. Moreover, this also holds, e.g., for DIP and Digital Item Adaptation (DIA) for associating functionality with objects and adapting them according to the delivery context. According to [11], concepts remarkably similar to DIP and DIA have been proposed by digital library projects. Finally, [12] shows how MPEG-21 and other standards, such as OAI-PMH (Open Archives Initiative Protocol for Metadata Harvesting) and NISO OpenURL, are integrated in the repository architecture.

— The potential impact of an MPEG standard.

Given the many possibilities to structure and nest DIDs, a fundamental design decision of the LANL researchers was how to map the complex digital library objects onto DIDs.

It was decided to use a simple three-level hierarchy comprising `Container`, `Item`, and `Component`. A DID may hold a `Container` or one or more `Item`s directly. Each `Container` or `Item` entity can contain one or more subordinate entities. Each `Component` contains and/or references one or multiple `Resources`. A DID of this type can only grow in breadth, but not in depth.

A second important design aspect is to treat descriptive metadata and media data alike. This choice is mainly motivated by digital preservation concerns: both media data and descriptive metadata need to be preserved and hence deserve to be treated equally.

Due to their importance in digital library applications, identifiers became a core element in the design of DIDs at LANL. Two types of identifiers are used:

— DID identifiers for the identification of the DID XML documents as well as of the contained XML elements that represent core DID entities.

— Content identifiers for the identification of assets contained in the DIDs.

DID identifiers are introduced for the identification of the DID XML documents themselves; they are dynamically assigned when a DID is created and ingested into the repository. A DID identifier is conveyed as an attribute from a LANL-defined namespace at the DID root element. Also during the ingestion process, DID `Container`, `Item`, and `Component` entities receive XML IDs that are attached as attributes to the corresponding XML elements. Also, the `Descriptor/Statement` constructs containing MPEG-21 DII `Identifier`s can be used to integrate community-specific identification schemes smoothly into the DIDs.

Content identifiers identify the information assets contained in DIDs, usually within Items, and they are conveyed using `Descriptor/Statement` and `Identifier` constructs. Content identifiers are typically derived from the information assets proper. In many cases, the identifier naturally attached to an asset during its creation or publication is adopted as the content identifier [8][9].

The same approach, i.e., the use of `Descriptor/Statement`, is heavily used to provide further information pertaining to the content. For example, relationships between entities contained in DIDs are conveyed by special-purpose `Descriptor`s containing RDF statements expressing relationships like "is member of" or "is translation of". Creation date and time, datastream format information, and W3C XML Signatures which are used to verify (e.g., authenticate) the DID itself and the information assets, can be embedded in this manner as well. More details are available in [16].

Finally, the LANL library researchers proposed to make specific use of DIP in their repository architecture [9]. In general, Digital Item Methods (DIMs) are embedded in a DID with the intent to provide specific functionality or services when the DID is retrieved from the repository and disseminated to the requester(s). However, at LANL, it was felt that static DIMs in DIDs would overly restrict the flexibility of the retrieval and dissemination process. After all, while content is of rather static, archival nature, the requirements or services of content dissemination will likely evolve over time. Freezing the functionality by embedding DIMs into DIDs already upon their ingestion into the repository would make it necessary to touch every DID and exchange the embedded DIMs whenever new dissemination services were devised and implemented as DIMs. This was felt as too much overhead, given the existence of tens of millions of DIDs in the repository.

Instead, a level of indirection and a late, dynamic binding mechanism for DIMs was introduced. Rather than placing DIMs directly in a DID, so-called PlaceHolder elements from a LANL-defined namespace are embedded initially. Upon retrieval of a DID from the repository, a specific module of the DIP framework (the DIM Inserter) performs a matching of PlaceHolder values with DIMs conveyed in a special-purpose registry of services (the DIP Table) and dynamically adds actual DIMs to the DID before delivering it to the requester.

Since this matching and dynamic DIM insertion process is beyond the scope of MPEG-21, it is not further dealt with in this chapter. The interested reader is referred to [9] which describes the operation of the DIP engine in the LANL repository architecture in more detail. However, at the time of writing it is unclear whether or not this proposal was pursued further; in more recent publications [16][17], this specific use of DIP is not mentioned anymore.

It should be noted finally that the LANL aDORe team released a Java toolkit for constructing, validating, serializing, and de-serializing MPEG-21 DIDs [18].

5. Selected Research Projects

5.1 Introduction

Several EC-funded projects have adopted MPEG-21 technologies and among them we will briefly describe three projects:

— Section 5.2 describes ***an advanced MPEG-21 infrastructure*** as developed in the context of the DANAE project [10]. It was one of the first projects that aimed at the integration of several MPEG-21 parts enabling the distribution of Digital Items within a heterogeneous environment.

— In Section 5.3 we will briefly highlight the ENTHRONE project [11] that demonstrated the usage of Digital Items enabling ***end-to-end Quality of Service (E2E-QoS) management***.

— Finally, the adoption of ***Digital Items within next generation peer-to-peer (P2P) networks*** is described in Section 5.4 and is developed as part of the P2P-Next project [12].

5.2 An Advanced MPEG-21 Infrastructure

In the project called DANAE (Dynamic and distributed Adaptation of scalable multimedia coNtent in a context-Aware Environment), funded by the European Commission in the 6th Framework Programme in the time frame 2004-2006, the consortium developed an advanced MPEG-21 infrastructure for offering, delivering, personalizing, and adapting multimedia content in an interoperable way [7]. Digital Item Declaration (DID), Digital Item Processing (DIP), and Digital Item Adaptation (DIA) play key roles in this architecture, and implementations of various MPEG-21 peers (server, proxy, and client) have been completed.

The major application of the DANAE MPEG-21 infrastructure was envisaged to be in a museum context. The main objective in this scenario is that museum visitors carrying MPEG-21 enabled mobile devices are delivered multimedia content (1) pertaining to the section of the exhibition they are currently in, (2) personalized to their preferences and needs, and (3) adapted to the capabilities of their device and the network they are connected to. It was further envisaged that the museum will have installed special presentation devices, e.g., large flat screens, which visitors might at times prefer over their device and migrate their active multimedia session onto (a session mobility case addressed by MPEG-21). While a full installation of this scenario in the museum that was among the project partners could not be realized due to practical and financial constraints, on a technical level the objectives have been fully achieved and demonstrated.

In the sequel, the architecture of the MPEG-21 adaptation/streaming server, as developed in the DANAE project for the above application scenario, will be discussed. The DANAE server can be seen as a typical MPEG-21 peer. The experiences gained from this design and implementation effort as well as from the deployment of the DANAE MPEG-21 infrastructure can be valuable input for the development of MPEG-21 systems.

It must be noted that, apart from server, a terminal – not discussed here due to page limits – and a third type of MPEG-21 conformant node has been developed within DANAE, namely an adaptation node/gateway that can be incorporated into a distributed adaptation system. A scenario in which distributed adaptation is important is an extra (proxy) server for a highly popular museum section that relieves the central adaptation/streaming server from servicing clients in that section (load balancing). Since most of the components and functionality of the separate adaptation node are in common with the central server, this extra node will not be discussed here. Additional issues that have to be considered, though, are how to partition and stream metadata to the adaptation node, synchronized with the media data, and how the client devices get to know about and contact the extra server, for instance. Additional information can be found in [19].

The MPEG-21 adaptation and streaming server is depicted in simplified form in Figure 2. The components and functionality of the server are briefly described as follows.

The *streaming server*, based on the open source Darwin server, delivers media data to its counterpart on the terminal side, the multimedia player, using the RTP/RTCP, RTSP, and UDP protocols. Media data has to be packetized and depacketized as well and, in the distributed adaptation case when the metadata must be delivered to the separate adaptation node, also metadata has to be (de-) packetized. Metadata here comprises descriptions as defined within MPEG-21 DIA, including generic Bitstream Syntax Descriptions (gBSDs), Adaptation QoS (AQoS) information, and Universal Constraints Descriptions (UCDs).

The *context aggregation tool* is responsible for receiving, aggregating, and storing context as well as forwarding context information to server modules requesting it, mainly to the adaptation engine. The context aggregation tool receives context DIDs or fragments (updates) thereof from the client DIP engine, extracts the relevant parameters, and maintains a database of context information, specific for each user, terminal, and session.

The *adaptation engine* is at the core of the MPEG-21 server. It is designed to perform three types of adaptation: gBSD-based media

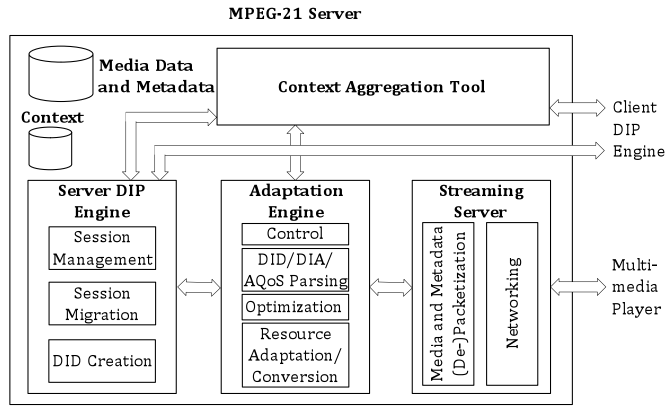

Figure 2. DANAE MPEG-21 Server Architecture.

resource adaptation, scene adaptation, and modality conversion. The adaptation component basically consists of the *optimizer* and the *resource adaptor* proper. The optimizer is a decision-taking engine that accepts content and context DIDs (including DIA descriptions, specifically AQoS information, from the context aggregation tool) as input, in order to come up with appropriate adaptation decisions. The resource adaptor actually performs the types of adaptation listed above. The adaptation engine is supported by parsers and validators for the XML documents involved.

The *server DIP engine* is in control of the application on the server side, very much like its counterpart on the client side. The DIP engine basically manages the open sessions, retrieves or generates and customizes museum catalogue and content DIDs according to the context of a user, delivers the DIDs, and invokes the adaptation engine if required. Clearly, the server side DIP engine is also involved in session migration activities.

5.3 End-to-end Quality of Service for Digital Items across Heterogeneous Environments

In order to enable the provisioning of end-to-end Quality of Service (E2E-QoS) for Digital Items across heterogeneous environments several actors along the delivery path are involved and play a crucial role that will be briefly highlighted in the following (see also Figure 3):

— The *Content Provider (CP)* prepares the actual multimedia content as MPEG-21 Digital Items facilitating scalable coding formats and metadata formats.

— The *Service Provider (SP)* provisions and offers multimedia services to end-users and enriches the multimedia content with additional metadata with respect to Service Level Agreements (SLAs) [20], taking into account constraints imposed by access networks for service provisioning towards the content consumer.

— The *Adaptation Provider (AP)* operates in close relationship with the SP and the NPs. Its goal is to provide improved QoS of content delivery while optimizing available system and network resources across the end-to-end chain. It takes content adaptation decisions according to the a-priori-known as well as dynamically received context information. Note that the actual content adaptation is done by the Content-/Service-/Network Providers.

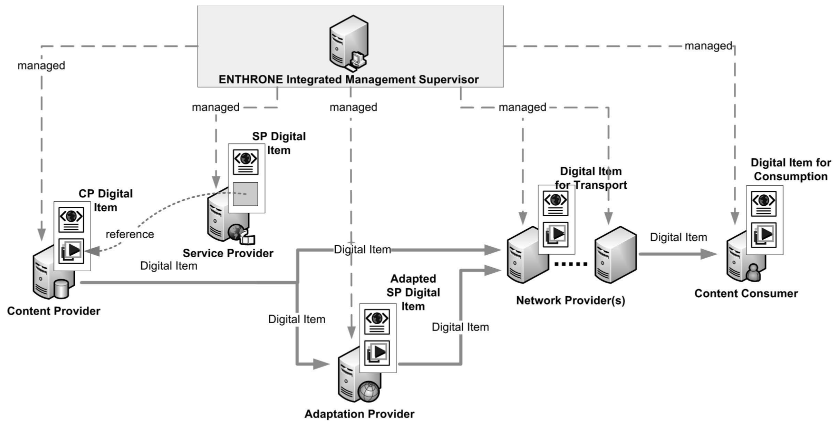

Figure 3. End-to-End QoS for Digital Items across Heterogeneous Environments.

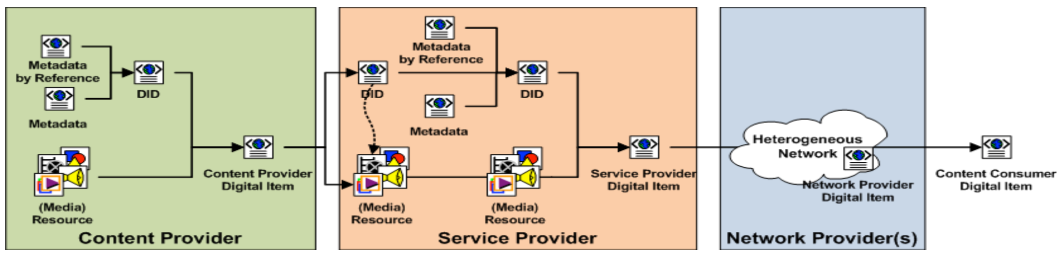

Figure 4. Provisioning of End-to-End Quality of Service for Digital Items [22].

— The *Network Provider (NP)* offers QoS-based network connectivity services at its autonomous domain level. Co-operation is needed among NPs for providing inter-domain QoS-based network connectivity services.

— The *Content Consumer (CC)* requests the services provided by the SP and consumes them on his/her end device. The actual end device functions depend on the business model.

These actors (or entities) are managed through the distributed ENTHRONE Integrated Management Supervisor (EIMS) that das been developed in the course of the ENTHRONE project [11]. In particular, the EIMS provides a set of management subsystems, i.e., EIMS Managers, with predefined functionalities and interfaces – based on Web Services – which enable the construction of ENTHRONE-based services according to the requirements of various scenarios [21].

As there are several actors involved, the Digital Item may undergo various – sometimes significant – changes while being delivered from the CP towards the CC as illustrated in Figure 4. However, the main changes will happen within the CP/SP domain. Hence, we will focus on the requirements coming from CP and SP, e.g., system-wide identification, temporal availability, encoding characteristics, adaptation possibilities and expected resulting qualities, constraints to be considered, etc.

As shown in Figure 4, the CP creates the initial Digital Item including the actual media resource and associated metadata. In this context, this Digital Item is referred to as the CP Digital Item. An SP may enrich this CP Digital Item with additional information pertaining to a particular service offered to the CC. This may, for example, include further metadata or an adapted version of the original Digital Item which fits the a-priori known requirements of certain CCs. The NPs are responsible for appropriate transfer of the so-called SP Digital Items which may undergo well-defined modifications that optimize the transmission over heterogeneous networking infrastructures. Finally, the CC receives the desired Digital Item appropriate to her/his context, i.e., anywhere, anytime, and on any device.

This data model differentiates between composed and final items. A composed item comprises sub-items which can be themselves composed or final whereas a final item does not contain any further sub-items but components as shown in Listing 3.

Listing 3. ENTHRONE Digital Item data model.

```
DIDL
  Declaration(s)  (referable descriptors)
  Container
   Descriptor(s) (top-level container descriptors)
   Item           (composed item)
    Descriptor(s) (top-level item descriptors)
    Item          (final item)
     Descriptor(s) (item-level descriptors)
     Component(s)
      Descriptor(s) (component-level descriptors)
      Resource

   Item(s) (further composed or final items)
Item(s)  (further composed or final items)
```

In this structure,

— `Declaration` elements may include `Descriptors` that are used by reference (instead of duplicating them),

— `Descriptor` elements provide the metadata for the DI at different levels (i.e., `Container`, Composed `Item`, Final `Item`, `Component`), and

— One or more `Component` elements each represent a variation of semantically equal media resources (but with, e.g., different bit rates, resolutions, qualities).

The `Descriptor` elements may contain standardized metadata (e.g., MPEG-7, TVAnytime, MPEG-21) but also proprietary data, mainly required for the delivery as defined within this framework. This proprietary metadata is used for the coordination of various entities within the delivery chain (e.g., servers, proxies, adaptation gateways) in order to provide an agreed level of quality to the end user. It is important to note that this proprietary metadata does not hamper the consumption of media resources or open standards-based metadata at the receiving terminal.

A more detailed review of the usage of MPEG-21 Digital Items within ENTHRONE can be found in [22].

5.4 Digital Item in Next Generation Peer-to-Peer Networks

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 is rapidly increasing as High-Definition (HD) 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 and 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. Therefore, we propose a metadata model that builds upon existing standards and extends them to support P2P-specific requirements. The requirements are based on the P2P-Next project [12] 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.

In particular, the P2P-Next project defines a P2P-Next item (i.e., A/V content + metadata (core + extensions) + structure) that is compliant to an MPEG-21 Digital Item and used for distribution within P2P networks. The torrent file is generated based on these assets from the P2P-Next Item and is used for distribution via the NextShare platform.

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 [23] 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 5. The torrent file contains the references and hash values for the media resources, i.e., the MPEG-2 Transport Stream 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

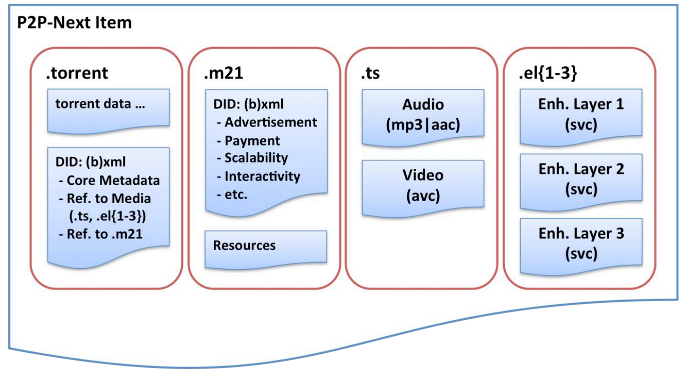

Figure 5. High-Level Structure of the P2P-Next Item.

(DID) [4] 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 [24] 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) [25], 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.

The DID for the optional content (packetized into an .m21 file in Figure 5) provides access to optional metadata, relating 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 Transport Stream 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 Transport Stream.

6. Conclusions

In this paper we presented a brief overview of the usage of MPEG-21 Digital Items in research and practice. The concept of Digital Item and the multimedia framework allowing the transaction therefore is powerful, generic, and flexible for a plethora of use cases and application domains. Experiences so far seem to indicate that the vision of providing the multimedia framework for widespread end-to-end use has not become reality though. There seem to be several reasons behind this.

— First, interoperability on a large scale in practical settings is difficult, if not impossible, to achieve.

— Second, the experience from the various research projects has shown that MPEG-21 support results in complex middleware and intricate interplay between various layer and levels (e.g., application, transport, network, system, etc.).

— Third, it seems to be difficult to identify clear benefits for a single stakeholder in the multimedia value chain (User in MPEG-21 terminology) of adopting MPEG-21 concepts as compared to using proprietary technologies.

— Finally, potential users might still be insufficiently aware of the MPEG-21 family of standards. Probably this chapter will have pointed to interesting MPEG-21 technologies and applications for these users.

However, there is still hope as MPEG has identified some of these issues as well and developed a comprehensive middleware comprising application programming interfaces (APIs) and protocols [26] that allow for interoperable access and exchange of all kind of MPEG-related information (coding, description, systems, etc.) within heterogeneous environments. With such APIs and protocols it shall be possible to hide the messy details and complexity of various MPEG technologies and foster the application development independent of the actual underlying standards-based implementation.

7. Acknowledgments

This work was supported in part by the EC in the context of the DANAE (FP6-2002-IST-1-507113), ENTHRONE (FP6-IST-1-507637), and P2P-Next projects (FP7-ICT-216217).

8. References

[1] Burnett, I.S., Pereira, F., Van de Walle, R., and Koenen, R., (eds.), 2006. The MPEG-21 Book, John Wiley & Sons Ltd.

[2] Adactus, www.adactus.no (last access: Mar. 2010).

[3] Enikos, www.enikos.com (last access: Marc. 2010).

[4] Burnett, I.S., Davis, S.J., and Drury, G.M. 2005. MPEG-21 Digital Item Declaration and Identification – Principles and Compression. IEEE Transactions on Multimedia. 7, 3 (Jun. 2005), 400-407.

[5] Vetro, A. and Timmerer, C. 2005. Digital Item Adaptation: Overview of Standardization and Research Activities. IEEE Transactions on Multimedia. 7, 3 (Jun. 2005), 418-426.

[6] UPnP Forum, 2006. ContentDirectory:2 Service Template Version 1.01, May 2006. Available: http://www.upnp.org/specs/av/UPnP-av-ContentDirectory-v2-Service-20060531.pdf (last access: Marc. 2010).

[7] De Keukelaere, F., De Zutter, S., and Van de Walle, R. 2005. MPEG-21 Digital Item Processing. IEEE Transactions on Multimedia. 7, 3 (Jun. 2005), 427-434.

[8] Bekaert, J., Hochstenbach, P., and Van de Sompel, H. 2003. Using MPEG-21 DIDL to Represent Complex Digital Objects in the Los Alamos National Laboratory Digital Library, D-Lib Magazine, 9, 11, (Nov. 2003) Available: http://www.dlib.org/dlib/november03/bekaert/11bekaert.html

[9] Bekaert, J., Balakireva, L., Hochstenbach, P., and Van de Sompel, H. 2004. Using MPEG-21 DIP and NISO OpenURL for the Dynamic Dissemination of Complex Digital Objects in the Los Alamos National Laboratory Digital Library. D-Lib Magazine, 10, 2, (Feb. 2004) Available: http://www.dlib.org/dlib/february04/bekaert/02bekaert.html

[10] DANAE, http://danae.rd.francetelecom.com/ (last access: Mar. 2010).

[11] ENTHRONE, http://www.ist-enthrone.org/ (last access: Mar. 2010).

[12] P2P-Next, http://www.p2p-next.org (last access: Mar. 2010).

[13] ISO/IEC 14977:1996, Information technology -- Syntactic metalanguage -- Extended BNF, Dec. 1996.

[14] ISO/IEC 21000-2:2005, Information technology -- Multimedia framework (MPEG-21) -- Part 2: Digital Item Declaration, 2nd edition, Oct. 2005.

[15] Kunze, J. and Baker, T., (eds.). 2007. The Dublin Core Metadata Element Set, IETF RFC 5013, Aug. 2007.

[16] Bekaert, J., De Kooning, E., and Van de Sompel, H. 2006. Representing digital assets using MPEG-21 Digital Item Declaration, International Journal on Digital Libraries, 6, 2, (Apr. 2006), 159-173.

[17] Van de Sompel, H., Chute, R., and Hochstenbach, P. 2008. The aDORe federation architecture: digital repositories at scale, International Journal on Digital Libraries, 9, 2, (Nov. 2008), 83-100.

[18] LANL aDORe DIDL Tools. Available: http://african.lanl.gov/ aDORe/projects/DIDLTools/ (last access: Mar. 2010).

[19] Hutter, A., Amon, P., Panis, G., Delfosse, E., Ransburg, M., and Hellwagner, H. 2005. Automatic Adaptation of Streaming Multimedia Content in a Dynamic and Distributed Environment, in Proc. International Conference on Image Processing (ICIP 2005), Genova, Italy, (Sept. 2005)

[20] Verma, D. C. 2004. Service Level Agreements on IP Networks, Proceedings of the IEEE, 92, 9, (Sep. 2004), 1382-1388.

[21] Timmerer, C., et.al., 2008. An Integrated Management Supervisor for End-to-End Management of Heterogeneous Contents, Networks, and Terminals enabling Quality of Service, Proceedings 2nd European Symposium on Mobile Media Delivery (EuMob), Oulu, Finland, (Jul. 2008).

[22] Timmerer, C., Andrade, M.T., Carvalho, P., Rogai, D., Cordara, G. 2008. The Semantics of MPEG-21 Digital Items Revisited", in Proc. ACM Multimedia 2008, 2nd International Workshop on the Many Faces of Multimedia Semantics, Vancouver, Canada, (Oct./Nov. 2008).

[23] Cohen, B. The BitTorrent Protocol Specification, http://www.bittorrent.org (last access: Mar. 2010).

[24] P2P-Next D5.3.1, Tools for Rich Metadata and Signposts, 2009, http://www.p2p-next.org/download.php?id=C48E37B2671AE3F3B8459942B2C7E484 (last access: Mar. 2010).

[25] Singer, D., Visharam, M.Z. 2006. MPEG-21 File Format white paper, ISO/IEC JTC 1/SC 29/WG 11N7925, Montreux (2006)

[26] Timmerer, C., Chiariglione, F., Preda, M., and Rodriguez-Doncel, V., 2010. Accelerating Media Business Developments with the MPEG Extensible Middleware, to appear in the FIA Valencia Book 2010 (working title), IOS Press, (2010).