Implementation of a Mobile MPEG-21 Peer

Shane Lauf
University of Wollongong
Wollongong NSW 2522 Australia
+81 80 6544 5000
shane@jshane.net

Ian Burnett
University of Wollongong
Wollongong NSW 2522 Australia
+61 2 4221 3399
ianb@uow.edu.au

ABSTRACT

The MPEG-21 Multimedia Framework aims to realize interoperable access to content across heterogeneous networks and devices. Within the Framework, the concept of Digital Items is introduced as a structured digital representation for multimedia. To demonstrate the applicability of MPEG-21 to seamless multimedia interactions on limited platforms, the authors have produced an implementation of MPEG-21 for a mobile device, in Java 2 Micro Edition (J2ME). This paper examines the design and implementation of the Mobile MPEG-21 Peer, including a specialized architecture and processing mechanisms specific to the J2ME platform.

Categories and Subject Descriptors

H.5.1 [**Information Interfaces and Presentation**] Multimedia Information Systems – *evaluation/methodology.*

H.5.2 [**Information Interfaces and Presentation**] User Interfaces – *evaluation/methodology, input devices and strategies.*

H.5.5 [**Information Interfaces and Presentation**] Sound and Music Computing – *methodologies and techniques, systems.*

General Terms

Algorithms, Performance, Standardization.

Keywords

MPEG-21, Multimedia, Mobile Applications.

1. INTRODUCTION

Significant effort is being devoted to the realization of seamless multimedia access; the goal is to enable user access to multimedia content anywhere. This vision holds much in common with that of MPEG-21 [1] (the 'Multimedia Framework') which seeks to facilitate interoperable access to content using Digital Items as a universal currency for multimedia. For MPEG-21 to be realistically applicable to seamless access, its technology must be implementable on lowest-common-denominator platforms. Thus, this work seeks to provide an implementation of MPEG-21 on a mobile device with limited processing power, network access and storage capability; the result is an MPEG-21 application solution implementable across a wide range of terminals.

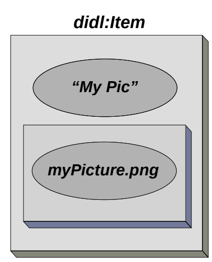
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
MM'05, November 6–11, 2005, Singapore.
Copyright 2005 ACM 1-59593-044-2/05/0011…$5.00.

2. DIGITAL ITEMS AND MPEG-21

Digital Items are the currency within the MPEG-21 framework, encapsulating multimedia content and related metadata. The Digital Item Declaration (DID) Model [2] defines a set of entities in Digital Items, and principal examples of these are:

- *Resources* which link to individually identifiable assets (multimedia content), such as audio or video files
- *Components* encapsulating and linking metadata with *Resources*
- *Items* encapsulating *Components* and sub-*Items*
- *Containers* encapsulating a collection of *Items*

The Digital Item Declaration Language (DIDL) provides a Representation of the DID Model, allowing the makeup, structure and organization of a Digital Item to be conveyed in a Digital Item Declaration (DID), which is an XML document. DIDL includes elements corresponding to the entities of the DID Model and defined in the normative XML Schema for DID. Figure 1 shows an example Digital Item, containing a picture, an associated text descriptor, and its representation in DIDL.




```
<Item>
 <Descriptor>
 <Statement mimeType="text/plain">
    My Pic</Statement>
 </Descriptor>
 <Component>
 <Resource ref="myPicture.png"
    mimeType="image/png"/>
 </Component>
</Item>
```

Figure 1: Example Digital Item

Other parts of MPEG-21 include Digital Item Identification (DII), which specifies *Identifiers* used to identify Digital Items, and parts and collections thereof [3]; the MPEG-21 Rights Expression Language (REL) which expresses the rights, terms and conditions applied to content [4]; and Intellectual Property Management and Protection (IPMP) Components, which provides for the management of rights and intellectual property through the use of protected Digital Items [5].

3. IMPLEMENTATION OBJECTIVE

The objective of this work was the development of a Mobile MPEG-21 Peer, designed for mobile phones running the MIDP 2.0 Profile of Java 2 Micro Edition [6]. The key aim was to demonstrate an architecture which, while encompassing a sufficient range of MPEG-21 technology to realize the consumption, authoring, and transmission of Digital Items, was efficient and flexible enough to run on the limited processing resources of MIDP 2.0 devices. The K700 handset and Wireless Toolkit from Sony Ericsson were used in development.

4. APPLICATION FUNCTIONALITY

The specific functionality planned and realized in the Mobile MPEG-21 Peer was: Web access for Digital Item download; Digital Item browsing and storage; Presentation of assets within Digital Items; Digital Item authoring; IPMP DIDL protected Digital Item processing. These functionalities are now considered in detail.

4.1 Web access for Digital Item Download

The implementation of Digital Item download via the web presented some challenges, as the security model of MIDP 2.0 does not allow an application to leverage the internal web browsing functionality of the mobile device. Rather than write a web browser from scratch, the authors decided to implement web access through an adapted version of the open-source PicoBrowser [7], which is optimized for pages translated to cHTML, the standard for i-mode. By modifying the PicoBrowser to recognize links to Digital Items by filename extension and return them to the application core for parsing and display, functionality was realized allowing users to seamlessly navigate to a target Digital Item via the web, select it for download, view its contents and store it locally.

4.2 Digital Item Browsing and Storage

Given the screen space restrictions of mobile devices, the Mobile Peer required an efficient and intuitive mechanism for user browsing of Digital Items. The key advantage of Digital Items is their structured expression of multimedia content, and for this reason the authors implemented a hierarchical browsing mechanism, which holds much in common with i-mode navigation. Figure 2 shows the hierarchy display at the level of an *Item* with title “Album Tracks”. To improve navigability, the Peer scans each sub-*Item* (corresponding to a track, in this case) for a title located within a *Descriptor*, and also displays the first image it finds associated with each sub-*Item* within a *Descriptor* or *Component*.

Figure 2: Display of Digital Item hierarchy at Item level

When a user chooses to store a Digital Item for later use, the Peer serializes both the DIDL and the associated Asset Manager cache (see 7 below) to a Digital Item database built on the MIDP Record Management System, allowing offline access later.

4.3 Presentation of Assets within Digital Items

Once a user selects an image, sound or video asset from a Digital Item, the Mobile Peer invokes the MIDP 2.0 Media API to present the asset to the user. Mechanisms for image and audio presentation are relatively stable, though the range of MIME types which can be rendered varies; video playback technology, however, remains immature on many devices including the K700, and it was found that subtle encoding differences can lead to unexpected device crashes. The Mobile Peer was designed to query the device (in advance) as to its multimedia rendering capacity, and also monitors for exceptions during rendering due to media errors or memory shortage, to ensure system stability.

4.4 Digital Item Authoring

On devices equipped with a built-in camera, users may author (create) a Digital Item containing photographs taken from within the Mobile Peer. Each photo is assigned a unique filename according to the date and time of the photograph. As J2ME does not provide for saving image files, the image data is assigned a pseudo-URL and injected directly into the Asset Manager cache (see 7). A templated Digital Item hierarchy is then built around the photographs, including serial numbers embedded in *Descriptors*, and *Resources* linking to the image data via pseudo-URLs. The resulting Digital Item is added to the list of Digital Items currently in memory, and may be stored by the User for later access.

4.5 IPMP DIDL protected Digital Item processing

The Mobile Peer includes the first prototype implementation of IPMP Components technology allowing for user access to protected rather than clear Digital Items. When the Peer encounters parts of a DID hierarchy expressed in protected IPMP DIDL, it examines the REL Grants possessed by the user and determines whether the user is authorized to access the protected content, by checking for a Grant conferring the right to Play on the User identified by the handset IMSI (International Mobile Subscriber Identity). Once the authorization is confirmed the Peer uses associated information on the protection (e.g. decryption keys) to process DID hierarchy within the protected area.

5. HIGH-LEVEL DEVELOPMENT CONSIDERATIONS

Given the limitations of the target device platform, a number of key, high-level architectural considerations were identified:

- Modular and decoupled architecture:
- Optimized parsing of elements across relevant MPEG-21 namespaces
- Robust DIDL/IPMP DIDL class design

The outcome of design work based on these considerations is now considered.

6. APPLICATION ARCHITECTURE

The authors recognized that to provide maximum portability across devices of varying specifications and to facilitate extensibility, the architecture needed to be modular and decoupled. An overview of the resulting application package structure is shown in Figure 3. A decoupling is maintained between the Application package (edu.uow.mmedia) dealing with presentation and user interaction, and classes corresponding to the MPEG-21 functionality (org.iso.mpeg.mpeg21).

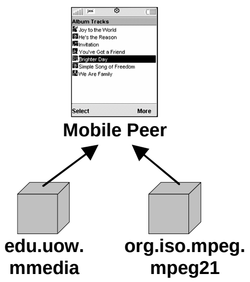


Figure 3: Package structure overview

The MPEG-21 package is examined in specific detail here. It has two key functions: to provide a static representation of a Digital Item in memory, and to realise MPEG-21 processing functionality.

6.1 Digital Item representation

The core of the Digital Item representation is the Mpeg21Object class, an abstract parent for all classes corresponding to a particular MPEG-21 element (*Item*, *Grant*, etc). An implementation of Mpeg21Object is provided for each XML element in every relevant MPEG-21 schema, and is given the following key functionality:

- Instantiation methods based on parsed XML data
- Members for storing element-specific data
- References to parent and child elements, allowing hierarchical navigation

As shown in Figure 4, sub-packages exist for each relevant part of the MPEG-21 Framework – DID (including DIDL and IPMP DIDL), DII, a mobile profile of REL, and IPMP Info – and it is within these packages that Mpeg21Objects are provided for each MPEG-21 element.

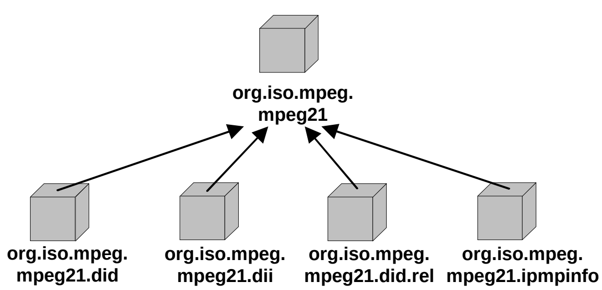


Figure 4: Sub-package structure of MPEG-21 package

6.2 MPEG-21 processing

Within the package structure above, MPEG-21 processing is founded on various Engine classes which realize MPEG-21 functionality. These Engines are designed to interact in a way consistent with the relationships between each relevant part of the MPEG-21 Framework: DID, DII, REL, and IPMP Components. The interaction between these parts is evident in the parsing and verification of a Digital Item represented by a DID, a simplified version of which is expressed in Figure 5.

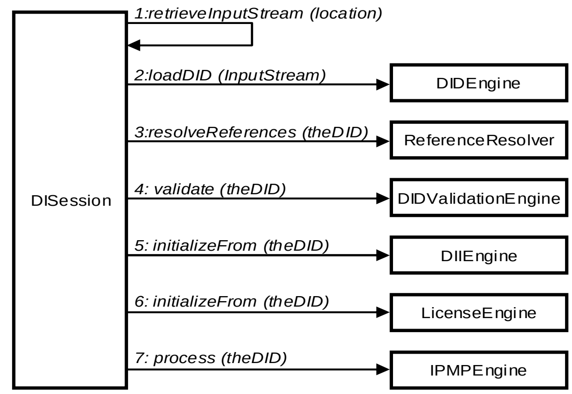


Figure 5: Parsing and verification of a Digital Item

On start-up, the application-level DISession class retrieves an appropriate InputStream (1) according to the location of the DID via HTTP, and instantiates a DIDEngine. DIDEngine loads the DID from this input stream, and constructs and initializes Mpeg21Object instances using the parser (2).

Before the DID is ready for access, internal references are resolved (3), and the DIDValidationEngine checks its validity (4). The DIIEngine scans the newly loaded DID for *Identifiers* (5) and the LicenseEngine scans it for *Grants* (6). Finally, the IPMPEngine uses the *Identifier* and *Grant* tables thus constructed to process protected DID structure represented in IPMP DIDL (7) (see 4.5 above). After this point, the Digital Item is available to the application and user interface.

7. SPEED OF DATA ACCESS

Though an increasing number of MIDP 2.0 devices are 3G capable and provide relatively high-speed connectivity, a significant proportion remains restricted to the low data rates of GPRS. As a Digital Item may reference a range of multimedia content beyond device capabilities in terms of speedy download (and memory storage), the Peer was designed to download assets only after explicit user selection. Consequently some caching mechanism was necessary to avoid multiple downloads when users navigate to assets already presented. The Asset Manager (Figure 6) was developed to act as a proxy between the user session and the web, managing an asset cache according to device free memory, and tracking asset URLs to ensure that images and other content already retrieved into memory are not needlessly reloaded. The Asset Manager and its cached content are also serialized when Digital Items are stored for offline access (see 4.2).

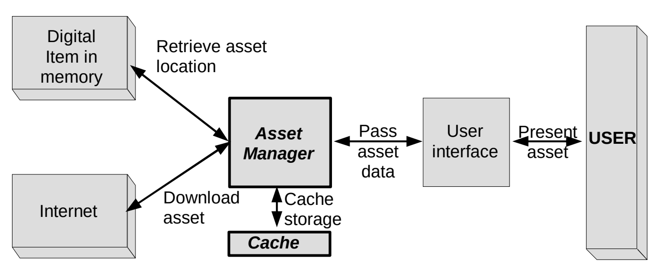


Figure 6: Asset Manager

8. OPTIMIZED ELEMENT PARSING

Given the sheer number of MPEG-21 elements in DIDL, DII, IPMP Components, and REL, an efficient mechanism for the parser to load XML elements is vital to avoid processing delays. MPEG-21 reference software available at time of development generated class instances for XML elements which were identified using large if-then structures, checking each possible element name against a set of String constants. The resulting code was both oversized and inefficient.

The solution makes use of the micro-reflection mechanism available in the CLDC platform, allowing classes to be instantiated dynamically based on their text name. In the optimized Peer, micro-reflection is used to instantiate MPEG-21 Representation classes based on XML namespace and tag.

Table 1: Mapping of XML namespaces to packages

Namespace (urn:mpeg:mpeg21:)	Package (org.iso.mpeg.)
2002:02-DIDL-NS	mpeg21.did.didl
2004:01-IPMP-NS	mpeg21.did.ipmp
2004:01-IPMPINFO-NS	mpeg21.ipmpinfo
2002:01-DII-NS	mpeg21.dii
2003:01-REL-R-NS	mpeg21.rel
2003:01-REL-MX-NS	mpeg21.rel.mx

By mapping XML namespaces to packages, as shown in Table 1, and ensuring that Java classes are named for their corresponding XML tags, the entire hierarchy of if-then constructs in existing implementations was reduced to a single statement:

```
element = Class.forName(packageName +
       tagName).newInstance();
```

If the classes for a specific namespace are not available at runtime, the Parser simply reads and records that section of the DID as plain XML, allowing the integrity of DID parsing to be maintained even on stripped-down platforms.

In addition to improved performance and code efficiency, the micro-reflection approach made a truly modular design for the Peer possible, allowing packages such as REL or IPMP to be plugged in and out simply by adding or removing the .class files from the application JAR. Future parts of MPEG-21 require only an entry in the namespace-package map to facilitate immediate integration into the Peer.

9. DIDL/IPMP DIDL CLASS DESIGN

While a range of existing software implements class structures for elements in DIDL, the Mobile MPEG-21 Peer is the first application to realise an expression of the IPMP DIDL elements. The authors judged that as the IPMP Components specification views DIDL and IPMP DIDL as two Representations of the abstract DID Model, the structure of classes coded to represent elements from these Representations should also reflect the flexibility of this relationship.

The class design selected, unlike other MPEG-21 implementations, represents the DID Model as a set of Java interfaces corresponding to each DID Model entity. These are independent from the implementation classes for elements of each Representation (DIDL, and IPMP DIDL) of the DID Model, which exist in sub-packages (Figure 7).

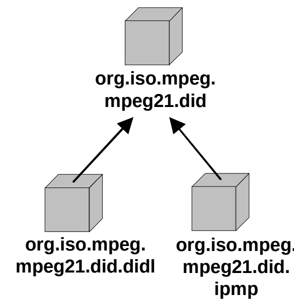


Figure 7: Sub-package structure of DID package

No classes outside the MPEG-21 Representation package access the element classes directly, instead requesting functionality via the interfaces which they implement. Thus, external packages see both IPMPItem and a DIDLItem simply as *Items* and require no special knowledge about the respective implementations. This cohesive, decoupled design pattern is particularly advantageous, as the application package need not be modified whenever revisions and modifications to the IPMP standard lead to changes in the implementation classes.

10. CONCLUSIONS

Based on research into the MPEG-21 Multimedia Framework and Java 2 Micro Edition, an IPMP-enabled mobile implementation of MPEG-21 has been successfully implemented. In the context of rapidly expanding 3G mobile services and ever-increasing demand for mobile content functionality, the Mobile MPEG-21 Peer demonstrates an effective and extensible solution for mobile MPEG-21.

11. REFERENCES

[1] Information Technology – Multimedia Framework (MPEG-21) – Part 1: Visions, Technology and Strategy TR, ISO/IEC JTC1/SC29/WG11/N6388, Munich, March 2004

[2] Van de Walle, R., Burnett, I., Drury, G. (editors), ISO/IEC 21000-2 DID 2nd Edition FDIS, ISO/IEC JTC1/SC29/WG11/N6927, Hong Kong, January 2005

[3] Information Technology – Multimedia Framework – Part 3: Digital Item Identification, ISO/IEC JTC1/SC29/WG11/N4939, Klagenfurt, July 2002

[4] Information Technology – Multimedia Framework – Part 5: Rights Expression Language, ISO/IEC JTC1/SC29/WG11/N5839, Trondheim, July 2003

[5] Watt, S., Lauf, S., Huang, Z., Rodriguez, E. (editors), ISO/IEC 21000-4 IPMP Components FCD, ISO/IEC JTC1/SC29/WG11/N7196, Busan, April 2005

[6] Sun Microsystems, Mobile Information Device Profile (MIDP) JSR 37, JSR 118. http://java.sun.com/products/midp/index.jsp

[7] Minsky, H. *The IAppli/MIDP PicoBrowser.* http://bearlib.sourceforge.net/www/pico.html

[8] W3C, XML Schema, May 2001. http://www.w3.org/TR/2001/