Adding functionality to multimedia content in an MPEG-21 scenario

Frederik De Keukelaere, Saar De Zutter, Rik Van de Walle
Multimedia Lab, ELIS, Ghent University, Belgium.
Interuniversity MicroElectronics Center (IMEC), Leuven, Belgium
frederik.dekeukelaere@ugent.be, saar.dezutter@ugent.be, rik.vandewalle@ugent.be

Abstract

Within the world of multimedia, several standards for content encoding and decoding exist. However, with the current capabilities of multimedia devices only encoding and decoding of multimedia becomes more and more cumbersome. In this paper we explore how interactivity can be added to multimedia content using a new and upcoming standard, ISO/IEC 21000, better known as MPEG-21. The purpose of MPEG-21 is to create an open framework for multimedia delivery and consumption. This framework enables its users to exchange, consume and manipulate a large variety of content in an efficient, transparent and interoperable way. This paper introduces the concept of Digital Item Processing, which allows to include interactivity and functionality in MPEG-21. The feasibility of the MPEG-21 approach is proven by the implementation of a demonstrator on the J2ME platform. Time and memory performance measurements are outlined to demonstrate what the cost of including extra processing information to multimedia content is using MPEG-21.

1. Introduction

Currently the world of multimedia and multimedia devices is confronted with a large set of different multimedia formats. Each of these formats has its own characteristics and capabilities. However, regardless of the fact that they are audio, video, or other coding formats, they all provide a similar high-level functionality, which consists of content encoding and content decoding.

Nowadays, a multimedia experience tends to require more and more interactivity with the end user or even with the terminal and the network on which this content is consumed. This requirement results in an increasing demand to add functionality to multimedia formats.

To add such functionality, it is necessary to go beyond the process of content encoding and decoding and move towards the creation of a framework that allows the inclusion of interactivity.

In MPEG-4, there is the possibility to include basic interactivity in the actual multimedia content using the XMT language [1]. Functional Metadata [2] can be considered as a first step towards a more generic inclusion of interactivity and functionality in multimedia content. In this paper, we describe how functionality and interactivity can be included in a new multimedia framework MPEG-21 [3].

2. MPEG-21

A new upcoming standard, in which we will include functionality and interactivity, is ISO/IEC 21000, better known as the MPEG-21 multimedia framework. It is a broad standard that, besides data encapsulation and data presentation, also addresses the problem of Dynamic Rights Management (DRM), dynamic adaptation of multimedia content, etc.

Contrary to most other multimedia standards, MPEG-21 tries to realize the “big picture”. MPEG-21 describes how various elements of multimedia content fit together throughout the multimedia delivery and consumption chain. It provides its Users, defined as any entity that interacts with the MPEG-21 environment at any point throughout the multimedia delivery and consumption chain, the ability to deliver and consume multimedia data across a variety of terminals, networks and platforms. Because of the generic character of MPEG-21, it is suitable for use by any multimedia related application.

Digital Items (DIs) are the key concept in the framework. Within the framework, they are defined as “structured digital objects, including a standard representation, identification and metadata”. It is the
second part of MPEG-21, the Digital Item Declaration Language (DIDL) [4], that makes it possible to create such a Digital Item composed of multiple multimedia resources. The Digital Item Declaration Language provides the Users the ability to describe the relationship between the different elements of a Digital Item.

Before MPEG-21, a wide set of different technologies existed that each allowed users to express a specific set of relations between the different resources that are part of a Digital Item. Some of them allow the description of temporal or spatial relations between different elements (e.g., SMIL [5]), while others allow the description of the semantic relation between elements of a Digital Item (e.g., RDF [6]).

With MPEG-21, a new technology that allows the ability to express whatever relation between the different parts of a multimedia presentation has been born.

3. Digital Item Processing

Because of the generic character of the Digital Item Declaration Language, it is possible to include both multimedia content and functional content in Digital Item Declarations. The technology that is used for this purpose is called Digital Item Processing [7][8].

Before giving an example of the actual inclusion of functionality in Digital Item Declarations, let us start by introducing a set of terms and definitions that are used in the context of Digital Item Processing.

A Digital Item Declaration (DID) is declared in the DIDL and contains the description of the relations between the different elements of a Digital Item.

A Digital Item Method (DIM) is a method that can be applied to a Digital Item. A DIM is a list of Digital Item Base Operations. A DIM is intended to prescribe the User’s (for example a consumer) interaction with the Digital Item.

The Digital Item Method Language (DIML) is the language in which the Digital Item Methods are expressed.

A Digital Item Base Operation (DIBO) is the smallest possible, elementary action that manipulates a Digital Item in a specific manner. The DIBOs can be considered as somewhat analogous to the standard library of functions of a programming language. The DIMs utilize these operations to accomplish the processing of the Digital Item according to the intentions of the User (for example an author).

An Object Type is a semantic type that can be associated with a DID element.
Table 1. Time measurements

<table>
<thead>
<tr>
<th>DID</th>
<th>parse DID (ms)</th>
<th>build ObjectMap (ms)</th>
<th>DIM with args (ms)</th>
<th>DIM without args (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>B</td>
</tr>
<tr>
<td>1</td>
<td>906</td>
<td>1313</td>
<td>1454</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>953</td>
<td>1281</td>
<td>1500</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>844</td>
<td>1328</td>
<td>1437</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>922</td>
<td>1328</td>
<td>1500</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>906</td>
<td>1422</td>
<td>1672</td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<td>953</td>
<td>1312</td>
<td>1500</td>
<td>16</td>
</tr>
<tr>
<td>7</td>
<td>906</td>
<td>1282</td>
<td>1453</td>
<td>16</td>
</tr>
<tr>
<td>8</td>
<td>922</td>
<td>1344</td>
<td>1453</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>953</td>
<td>1297</td>
<td>1469</td>
<td>15</td>
</tr>
<tr>
<td>10</td>
<td>890</td>
<td>1579</td>
<td>1453</td>
<td>0</td>
</tr>
<tr>
<td>average</td>
<td>915.5</td>
<td>1349</td>
<td>1489</td>
<td>0-16</td>
</tr>
</tbody>
</table>

Table 2. Memory measurements

<table>
<thead>
<tr>
<th>DID</th>
<th>parse DID</th>
<th>build ObjectMap</th>
<th>load DIM with args</th>
<th>load DIM without args</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>B</td>
</tr>
<tr>
<td>bytes</td>
<td>211828</td>
<td>338208</td>
<td>403692</td>
<td>242692</td>
</tr>
</tbody>
</table>

varying time needed to interact with DIs containing different processing information. The results of these tests can be found in table 1 and table 2. In the rest of this section we discuss the setup of the test environment and the results of the tests.

Please note that, since the terminal is implemented in Java 2 Micro Edition, the processing times we measured are only an estimate of the actual processing times. This is the case because execution times were measured using the difference between start time of the measured operation and end time. While measuring the times interference with the garbage collector can occur. Therefore varying results are measured while performing the same operations. However the results of the measurements give us an indication of the relative required time for the measured operations.

For the tests, we used a Pentium 4 emulating a mobile device. The operating system, a Windows XP Professional Edition, runs on top of an Intel Pentium 4 running at 2.8 GHz with Hyper-Threading enabled and with 1 GB of RAM. To emulate the J2ME environment we have used the Wireless Toolkit 2.0_01.

The emulator emulates a color cell phone which enables users to load an MPEG-21 Digital Item and presents a user interface suited for that Digital Item. The application searches for the available DIMs in the Digital Item and presents those methods to the user.

During the tests, we used several Digital Items with different characteristics to illustrate the influence of having processing information present in a Digital Item on the parsing of a DI. Two DIs containing processing information and one DI without any processing information and thus without Digital Item Methods were used.

The average times we measured are 915.5 ms for parsing the DI without any processing information, 1348.6 ms for a DI containing one DIM without arguments and 1489.1 ms for a DI containing two DIMs and arguments for one of those DIMs. From those results it becomes clear that the size of the DID influences the parsing time. Because the parsing is implemented as a one pass operation, the parsing time increases linearly with the size of the DID.

As a second test we measured the time required to build an Object Map. In order to build an Object Map, our demonstrator needs to search the DID for <ObjectType> elements and collect Object Type information about the DID elements. The building of the Object Map returned either 0 ms or 15 to 16 ms. As mentioned earlier, the time measurements are the actual times including the possible interferences, thus it can be assumed that the building of the Object Map on average requires maximum 16 ms. The time to build an Object Map does not seem to be significant compared to the time required to parse DIDs. It is
expected that this amount of time will increase linearly with the amount of objects that need to be added to the Object Map because the search for these objects is a linear operation.

When the user calls a Digital Item Method provided by the Digital Item, the DIM is loaded. The average times we measured for loading the two DIMs containing three DIBOs are 184.4 ms and 164 ms respectively. Loading the DIM containing two DIBOs takes on average 90.5 ms. The difference in time between loading a DIM with one or with several DIBOs, is as expected: if the DIM contains two DIBOs, it takes twice the amount of time as compared to a DIM with just one DIBO, provided the DIBOs are comparable in size.

The different measurements indicate that the largest amount of time is spent parsing the Digital Item. This amount increases linearly with the length of the document. From the measurements described above it becomes clear that the average time needed for processing the MPEG-21 information is only a very small fraction of the time it takes to play most multimedia content. This can therefore be considered as a minimal (generally ignorable) overhead.

The second part of our tests measured the amount of allocated heap memory. As expected, the memory allocated after parsing the DID increases if the size of the DID increases, respectively 211828, 338208 and 403692 bytes. Building the Object Map allocates a comparable amount of memory (242692 and 262904 bytes) for the DIDs C and D, since the added objects are the same. When the first DIM is loaded the amount of allocated memory is increasing to 552960 and 562280 bytes respectively and further increases to 741384 bytes when an additional DIM is loaded. From those results we can conclude that the required amount of heap memory increases rapidly, thus special care must be taken to respect the maximum memory usage for a specific mobile device.

5. Conclusions

In this paper we have described a need for including additional functionality and interactivity in multimedia content. We have demonstrated how this can be done in a new and upcoming standard called MPEG-21. We presented a system for including processing information in MPEG-21 Digital Items. Together with the inclusion of the processing information we introduced the concept of Digital Item Processing. We introduced a set of terms and definitions that are usable in this context.

The feasibility of the introduced concepts is proven by the implementation of a mobile application on the J2ME platform. At the end of this paper we have performed time and memory performance tests to the introduced techniques. For use in the tests, several Digital Items were created, each of them containing different processing information. From the results of the tests, we can conclude that the average time required to handle the additional processing information is in general an ignorable overhead. It is also clearly demonstrated that it is important to take extra care in minimizing the memory constraints of mobile applications.

6. Acknowledgements

The research activities that have been described in this paper were funded by Ghent University, the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT), the Fund for Scientific Research-Flanders (FWO-Flanders), the Belgian Federal Office for Scientific, Technical and Cultural Affairs (OSTC), and the European Union.

7. References