A Scalable Presentation Format for Multichannel Publishing Based on MPEG-21 Digital Items

Davy Van Deursen[1], Frederik De Keukelaere[1],
Lode Nachtergaele[2], Johan Feyaerts[3], and Rik Van de Walle[1]

[1] ELIS, Multimedia Lab, Ghent University - IBBT
Gaston Crommenlaan 8 bus 201, B-9050 Ledeberg-Ghent
{davy.vandeursen, frederik.dekeukelaere, rik.vandewalle}@ugent.be
http://multimedialab.elis.ugent.be
[2] Vlaamse Radio- en Televisieomroep (VRT), Auguste Reyerslaan 52, B-1043 Brussel
lode.nachtergaele@vrt.be
[3] Siemens N.V., Atealaan 34, B-2200 Herentals
johan.feyaerts@siemens.com

Abstract. In order to experience true Universal Multimedia Access, people want to access their multimedia content anytime, anywhere, and on any device. Several solutions exist which allow content providers to offer video, audio, and graphics to as many devices as possible by using scalable coding techniques. In addition, content providers also need a scalable presentation format to be able to create a presentation once and distribute it to all possible target devices. This paper introduces a scalable presentation format combining MPEG-21 technology with the User Interface Markup Language. The introduced presentation format is based on assigning types to MPEG-21 Digital Items and can be used to create a presentation once, whereupon several device-specific versions can be extracted. The reuse of resource and presentation information together with the use of a device-independent presentation language are hereby the key parameters in the development of the scalable presentation format.

1 Introduction

Content providers struggle with the huge heterogeneity in devices consuming multimedia today. They want to distribute their content to as many devices as possible, however due to the differences in device capabilities (e.g., screen size), different versions of the same content have to be created. Accessing multimedia anywhere, anytime, and on any device, is generally known as Universal Multimedia Access (UMA) [1]. To realize this goal, scalable content can be used. Research concerning resources (text, video, audio, and graphics) has been done in order to create scalable resources. However these resources are usually embedded in multimedia presentations and therefore presentations must also be scalable to realize the full UMA experience. This paper introduces a scalable presentation format that can be adapted according to a specific device.

B. Gunsel et al. (Eds.): MRSC 2006, LNCS 4105, pp. 650–657, 2006.

2 Separation of Presentation and Resource Metadata

In order to create a scalable presentation format, (i.e., a presentation format enabling the extraction of different presentations hereby targeting different devices) the separation of resource and presentation metadata is an important parameter to take into account. Resource metadata is information about the used content/resources in the presentation (e.g., an URL to a resource, the resolution of a video fragment, etc.). Presentation metadata is metadata about everything that can be linked to a presentation, except information about resources (e.g., layout and style information about the presentation). To separate resource and presentation metadata, we use two parameters. The first parameter is the location of the metadata: are they located in the same document or in separate documents? The second parameter is the way resource and presentation metadata are linked together. This is realized by using direct or indirect links. An indirect link is a reference to resource metadata, independent of the actual content of these resource metadata. A mapping mechanism has to be provided to translate the indirect links into direct references to resource metadata. The use of indirect links results in an independency between the presentation and resource metadata. With the use of these two parameters we define four different models for the distinction between presentation and resource metadata as illustrated in Fig. 1. In this figure, the dotted arrows represent the indirect links while the other arrows are the direct links.

The first model, shown in Fig. 1(a) has no separation of presentation and resource metadata. They are mixed, and described in the same format. Many presentation formats use this model (e.g., HTML or SMIL).

The second model supports a separation of the presentation and resource metadata in the same document as illustrated in Fig. 1(b). The use of indirect links between presentation and resource metadata implies the usage of a mapping mechanism which results in an independency of presentation and resource metadata. However, there is still a restriction on this independency. It is not possible to reuse the presentation or resource metadata in another document. Only inside the document, there is a flexible approach between presentation and resource metadata. The eXtensible interactive Multimedia Presentation Format (XiMPF) [2] can be mapped to this model. XiMPF uses device-specific presentation languages in combination with indirect referencing to resource metadata, but both resource and presentation metadata are located in the same document.

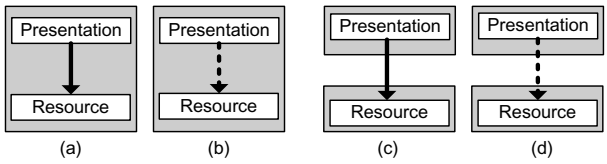

Fig. 1. The four possible models for the separation of presentation and resource metadata. The dotted arrows represent the indirect links, a grey box represents a document.

In model 3, the presentation and resource metadata are separated in two documents. The presentation metadata uses direct references to the resource metadata as shown in Fig. 1(c). With this model, it is possible to reuse the resource metadata in other presentation metadata. However, because of the direct references, it is not possible to reuse the presentation metadata for other resource metadata. This model maps to the proposal of the authors in [3], where a solution for multichannel distribution is introduced by using MPEG-21 Digital Item Declaration (DID) [4] in combination with device-specific presentation languages.

A complete separation of presentation and resource metadata is used in model 4. The presentation metadata uses indirect links to the resource metadata. This implies the usage of a mapping mechanism. With this model, which is shown in Fig. 1(d), it is possible to reuse the resource metadata for other presentation metadata and to reuse the presentation metadata for other resource metadata. Our scalable presentation format maps to model 4 in order to offer the most flexibility in reusing presentation and resource metadata.

3 Technologies for Resource and Presentation Metadata

This section introduces existing technologies for describing resource and presentation metadata. First, a technology for declaring resource metadata is discussed. Second, technologies for device-independent presentation metadata are discussed. By using device-independent presentation languages (in contrast to the work in [3] and [2] where device-specific presentation languages are used), the presentation metadata only needs to be created once. Afterwards, this device-independent presentation is translated into a device-specific presentation.

Declaration of Resource Metadata. It is important that the resource metadata can be organized in a structured manner. In practice, many resources can be split up hierarchically. For example, a news item typically consists of a title, a header, and content. This content is subdivided in paragraphs, pictures, video, and audio fragments. MPEG-21 DID makes it possible to structure the resource metadata by introducing the concept of Digital Items (DIs) which are defined as structured digital objects, with a standard representation, identification, and metadata. Beside the structuring of the resources, a technology for the description of the properties of the resources is needed. Therefore, our scalable presentation format uses the MPEG-7 [5] technology embedded in MPEG-21 DID descriptors.

Device-Independent Presentation Languages. The User Interface Markup Language (UIML) [6] and the Portable Content Format (PCF) [7] both provide solutions to create device-independent presentations. PCF is a presentation format specially designed for TV services and therefore less generic then UIML. For UIML, existing solutions to render or translate UIML documents are available,

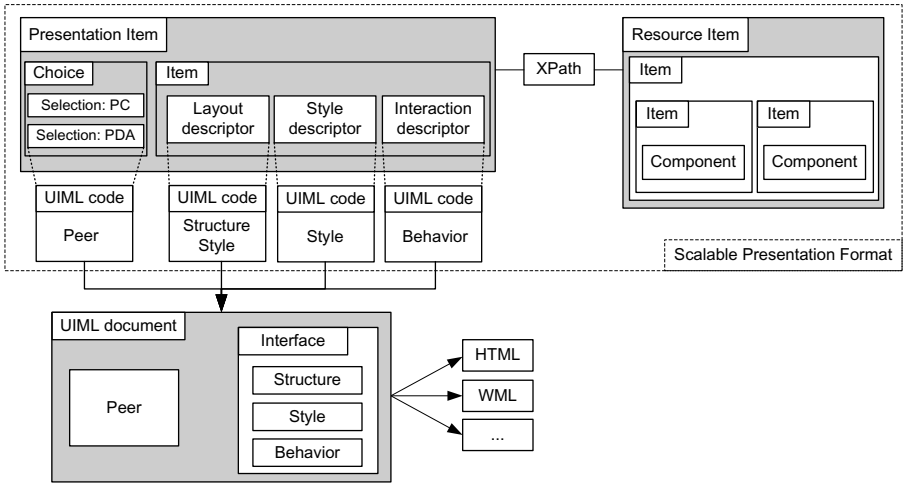

Fig. 2. Overview of the structure of our scalable presentation format

something that is not the case for PCF. Therefore, we have choosen UIML as the technology for the presentation metadata.

UIML is a declarative, XML-compliant meta-language for describing User Interfaces (UIs). UIML is designed to serve as a single language which permits creation of UIs for any device, any target language (e.g., Java or HTML), and any operating system on the device. The UIML document model (which is shown at the bottom of Fig. 2) consists of two major parts: *interface* and *peer*. The *peer* element describes a mapping of classes and names used in the UIML document to device-specific classes and names. The *interface* element describes the parts comprising the UI: structure (contains a list of *part* elements, each describing some abstract part of the UI), style (contains a list of *property* elements with the presentation properties of the parts), and behavior (describes basic interactivity within the UI). We do not make use of the *content* element of UIML, because it is not possible to structure the resource metadata within this *content* element. As discussed above, MPEG-21 DID is used to organize the resource metadata. Note that we also use MPEG-21 DID to organize the presentation metadata (UIML code fragments are included in DIs). This way, a fully MPEG-21 DID compliant presentation format is realized.

4 A Scalable Presentation Format

In this section, our scalable presentation format is discussed. An overview of the structure of the format is given in Fig. 2. The presentation format splits up the presentation and resource metadata, and uses MPEG-21 DID with MPEG-7 and UIML for respectively resource and presentation metadata, as mentioned in Sect. 2 and Sect. 3.

4.1 Assigning Types to Digital Items

In this presentation format, assigning a type to every DI plays a central role. This is realized by using the Digital Item Identification (DII) [4]. An item which belongs to a specific type contains a fixed structure. This structure is the same for all the items of the same type. To assign a type to a DI, the item must contain a *Descriptor* element with a *Statement* element containing a DII *Type* element as illustrated in Fig. 3. Every DI used in the presentation format is assigned a type. DIs used for resource and presentation metadata are further referred to as 'resource items' and 'presentation items' respectively.

4.2 Resource Items

The resource metadata is located in a DI of a specific type (e.g., a news item type). Such a DI can be seen as a tree structure where the leaves of the tree are always *Component* elements containing the actual resource metadata (a graphical representation is shown on the right side of Fig. 2). Note that a resource item is totally independent of any presentation metadata or target device. An example in XML is shown in Fig. 3.

4.3 Presentation Items

A presentation item is a DI which is of the type `presentation` (a graphical representation is shown on the left side of Fig. 2). This means that all the presentation items have the same structure. A *Choice* element is included for the device selection. The most important part of the presentation item is a DI containing the actual presentation description. It can contain three different *Descriptor* elements: a layout descriptor, a style descriptor, and an interaction descriptor. Note that MPEG-21 DII is used to assign a type to a descriptor in addition to assigning a type to a DI as discussed in subsection 4.1. The separation of layout, style, and behavior implies that the reusability of the presentation parts increases (e.g., the style descriptor can be the same for two devices while the layout descriptor is different). The layout, style, and interaction descriptor and the descriptor in the *Selection* element contain UIML code fragments, as elaborated on below.

Device Selection. The presentation item makes it possible to support multiple devices. This is realized by making use of the *Choice* - *Selection* elements of MPEG-21 DID as illustrated in Fig. 2 where two devices (PC and PDA) are supported. The *Selection* element is linked with a specific device and contains a descriptor with a UIML *peer* element pointing to the UIML vocabulary. A UIML vocabulary is device-specific and should therefore be included in a *Selection* element. Different descriptors containing presentation code can belong to one or more devices. This is expressed by making use of the *Condition* element (the condition expresses the specific device the descriptor belongs to). Every descriptor containing presentation information can have a *Condition* element. If the descriptor in question has no *Condition* elements, this descriptor can be used for all the supported devices presented in the *Choice* element.

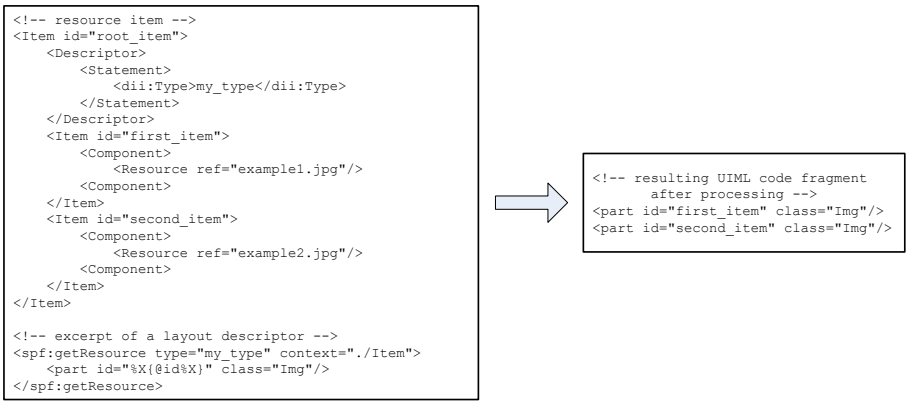

Fig. 3. Use of the *getResource* element in combination with a resource item. The resulting UIML code is shown on the right side.

Layout Descriptor. The layout descriptor describes the layout/structure of the presentation. It contains a UIML *structure* and *style* element. The structure element specifies the layout of the different parts of the presentation. The style element contains the properties which are related to layout (e.g., positioning of parts or id's of parts). Note that for every target device, at least the layout descriptor has to be present (otherwise there will be no UIML *structure* element present and as a consequence, no presentation).

Style Descriptor. The style descriptor consists of a UIML *style* element containing properties about the style aspects of the presentation. Examples are fonts or background colour.

Interaction Descriptor. The interaction descriptor is used to describe some basic functionality for the presentation. An example of such functionality is navigation. Note that only the interaction between the end-user and the presentation is meant here, not any interaction between the presentation and the back-end. This descriptor contains a UIML *behavior* element wherein the behavior of the presentation is described.

4.4 Link Between Presentation and Resource Metadata

We have discussed both the presentation and the resource item of our presentation format. Because they are separated, a mapping mechanism has to be provided in the presentation item to refer to a resource item. In this paper, XPath 1.0 [8] is used to solve this problem. All the resource items of the same type contain a fixed structure. Hence, accessing a resource item of a specific type can be realized in a generic way by using XPath expressions. A new element, *getResource*, is introduced to be able to use these XPath expressions in

the presentation item (an example is shown in Fig. 3). This element enables the selection of specific items within a resource item. Dependent on the number of selected items, the code within the *getResource* element is repeated (i.e., once for every selected item). Within the *getResource* element, the %X{ and %X} delimiters are introduced to access information of the selected item via an XPath expression. This approach makes it possible to make presentations for a specific type of resource items, independent of the actual content of the resource items of this specific type. In essence, every presentation is a template for a specific type of resource items.

4.5 Multiple Presentations

Suppose we created a presentation item using one or more types of resource items. It must be possible to insert this presentation item in another (bigger) presentation. Therefore, we introduce a new element: *usePresentation*. This element can be used in a layout descriptor to insert a presentation item. Together with this presentation item, the resource items needed by this presentation item have to be specified.

5 From Content Provider to End-User

The scalable presentation format, as introduced in this paper, is used by content providers to create presentations suited for multichannel publishing. When a target device is specified, the right descriptors of the presentation item have to be selected. Once this is done, a UIML document is created by combining the descriptor located in the *Selection* element, the layout descriptor, the style descriptor (optionally), and the interaction descriptor (optionally). This is shown in Fig. 2. The last step is to show the UIML document to the end-user. This can be done by translating the UIML code (e.g., translate to HTML) or by rendering the UIML code (e.g., render the code as a Java application).

The Multimedia Content Distribution Platform (MCDP) project [9] is currently investigating how a multimedia distribution system can support a variety of network service platforms and end-user devices, while preventing excessive costs in the production system of the content provider. A prototype implementation for processing our scalable presentation format has been developed in this project. Siemens and the Vlaamse Radio en Televisie (VRT) use the introduced presentation format to create scalable presentations suited for multiple end-user devices.

6 Conclusions

In this paper, we introduced a scalable presentation format for multichannel publishing. This allows content providers to create a presentation once, whereupon they can publish this presentation on every possible target device. We have shown that the use of a structured resource representation format together

with a device-independent presentation language are key parameters in creating a scalable presentation format. To realize this, we used MPEG-21 DID in combination with UIML and made use of assigning types to MPEG-21 DIs. For optimal reusability, a distinction is made between resource and presentation metadata. Two new elements, *getResource* and *usePresentation*, were introduced to access the resource metadata within the presentation metadata and to insert an existing presentation item into a new presentation. Finally, we discussed how to create a device-specific presentation starting from our presentation format.

Acknowledgements

The research activities as described in this paper were funded by Ghent University, the Interdisciplinary Institute for Broadband Technology (IBBT, 50% co-funded by industrial partners), 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 Science Policy Office (BFSPO), and the European Union.

References

1. Vetro A., Christopoulos C., Ebrahimi T.: Universal Multimedia Access, IEEE Signal Processing Magazine vol. 20, no. 2 (2003) 16
2. Van Assche S., Hendrickx F., Oorts N., Nachtergaele L.: Multi-channel Publishing of interactive Multimedia Presentations, Computers & Graphics, vol. 28, no. 2 (2004) 193-206
3. De Keukelaere F., Van Deursen D., Van de Walle R.: Multichannel Distribution for Universal Multimedia Access in Home Media Gateways. In: 5th International Conference on Entertainment Computing, Cambridge (2006) Accepted for publication
4. De Keukelaere, F., Van de Walle, R.: Digital Item Declaration and Identification. In: Burnett, I., Pereira, F., Van de Walle, R., Koenen, R. (eds.): The MPEG-21 Book. John Wiley & Sons Ltd, Chichester (2006) 69-116
5. Martinez J.M., Koenen R., Pereira F.: MPEG-7: The Generic Multimedia Content Description Standard, Part 1, IEEE MultiMedia vol. 9, no. 2 (2002) 78-87
6. Abrams M., Phanouriou C., Batongbacal A., Williams S., Shuster J.: UIML: an appliance-independent XML user interface language, Computer Networks (1999) 1695-1708
7. Digital Video Broadcasting (DVB): Portable Content Format (PCF) Draft Specification 1.0 (2006)
8. W3C: XML Path Language (XPath) 1.0. W3C Recommendation (1999), available on `http://www.w3.org/TR/xpath.html`
9. Interdisciplinary Institute for BroadBand Technology: Multimedia Content Distribution Platform (2006), available on `http://projects.ibbt.be/mcdp`