A Survey on Delivery Context Description Formats – A Comparison and Mapping Model

Christian Timmerer, Johannes Jabornig, Hermann Hellwagner
Department of Information Technology (ITEC), Klagenfurt University
Austria

Journal of Digital
Information Management

ABSTRACT: *Nowadays, mobile devices have implemented several transmission technologies which enable access to the Internet and increase the bit rate for data exchange. Despite modern mobile processors and high-resolution displays, mobile devices will never reach the stage of a powerful notebook or desktop system (for example, due to the fact of battery powered CPUs or just concerning the small-sized displays). Due to these limitations, the deliverable content for these devices should be adapted based on their capabilities including a variety of aspects (e.g., from terminal to network characteristics). These capabilities should be described in an interoperable way. In practice, however, there are many standards available and a common mapping model between these standards is not in place. Therefore, in this paper we describe such a mapping model and its implementation aspects. In particular, we focus on the whole delivery context (i.e., terminal capabilities, network characteristics, user preferences, etc.) and investigated the two most prominent state-of-the-art description schemes, namely User Agent Profile (UAProf) and Usage Environment Description (UED).*

Categories and Subject Descriptors
C.1.4 [Mobile processor]; H.3.4 [Systems and Software]: User profile: **D.2.11 [Software Architectures]:** Languages

General Terms: Mobile devices, User agent profile, MPEG, XML schema

Keywords: CC/PP, DCO, Delivery Context, Metadata, UAProf, UED.t

Received: 11 July 2009; Revised 21 August 2009; Accepted 29 August 2009

1. Introduction

Today, the access to the Internet via mobile phones and other devices which are limited in power capacity and/or rendering functionality increases. Additionally, manufacturers equip their devices with technologies to access various networks, and mobile providers offer services for connecting these devices to the Internet. Thus, using small, mobile devices enables the access to the Internet but often the available content of Web pages is not suitable according to their capabilities. In order to solve this issue, some projects and standards have been released which deal with the description of capabilities and characteristics of all kind of devices. Terminal capabilities and network conditions as well as user characteristics may allow the adaptation of the content for certain purposes.

Amongst others, there are two standards which were designed to meet the requirements of device and user descriptions and are often compared to each other. The first one was released by the WAP Forum (now the Open Mobile Alliance) and is named User Agent Profile (UAProf) [OM06] and the second one is the Usage Environment Description (UED) standard which was standardized within MPEG-21 Digital Item Adaptation (DIA) [VT05]. The aim of this paper is to build a mapping model for these description formats enabling context-aware content delivery independent of the actual description format used. Recently, W3C has started a new work item with the aim to define a delivery context ontology [LF08] but, still, the mapping issue remains.

While the User Agent Profile standard is very popular and has been implemented in a wide range of mobile devices the Usage Environment Descriptions are only limited available and tools which would ease the creation of such descriptions rarely exist. However, a high availability of Usage Environment Descriptions is desired by research projects [Da06][Ax08][En08] and, thus, obtaining information about terminals, networks and users from projects with similar aims to that of Usage Environment Descriptions should be enabled. For example, an implementation compliant to a standardized delivery context description format A requires a mapping module if it receives descriptions compliant to another standard B and vice versa. In order to keep the mapping effort minimal and scalable the proposed method in this paper enable the implementation of a service that performs such kind of mapping.

This paper is organized as follows. Section 2 highlights the main requirements on standards for delivery context descriptions by classifying terminals and their properties. An analysis and comparison of delivery context description formats is presented in Section 3 while Section 4 describes the actual mapping model. Finally, the implementation details are described in Section 5 and the paper is concluded with Section 6 which contains also future work items.

2. Requirements on Standards for Context Descriptions

In this section we have identified different classes of terminals including their hardware and software capabilities. This kind of information should provide us a rough estimation on the requirements for delivery context description standards from the terminal's point of view which also includes the access networks.

When the Internet became popular, the only way to access the Web was through a personal computer (PC) or a workstation. In general, these computers had large color displays with full

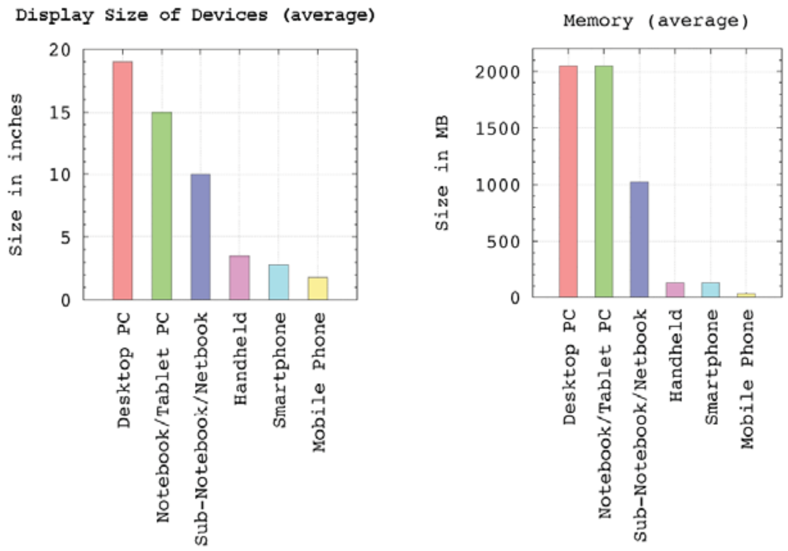


Figure 1. Summary and comparison of terminal's display and memory properties.

graphic capabilities, sufficient computational power without battery issues, and a decent network connection [GLS06]. Nowadays, people tend to access the Web using smaller and mobile devices with various constraints on display capabilities, user input/output facilities, computational power, electric power, and access networks ranging from high-speed Wireless Local Area Network (WLAN) to low-speed General Packet Radio Service (GPRS). In the following we will provide a classification of the various terminals based on their hardware (HW) and software (SW) characteristics

Table 2 in the Appendix provides an overview of HW/SW characteristics of different end user devices (i.e., desktop PC/workstation, notebook/tablet PC, sub-notebook/netbook, handheld, smart phone, and mobile phone) with respect to performance (i.e., CPU), display, permanent storage, memory, network connectivity, electric power, user input/output, extensibility, operating system support, and software in general.

A summary and comparison of terminal's display and memory properties is depicted in Figure 1. As one can see there is still a huge gap between classical mobile devices (i.e., phones) and devices that may have full power supply. Thus, a comprehensive delivery context standard needs to accommodate all these HW/SW properties in an easy-to-understand/use, extensible, and manageable way.

However, not only HW properties like screen size, color capabilities, or user input/output facilities are important, also SW properties like supported operating systems, audio/video/image codecs, etc. become more and more important as diversity of devices increases. In particular, the number of different coding formats a terminal is capable to support – both for encoding and decoding – is of interest for delivery context description formats. As there are so many coding formats available, some may have certain profile/level definitions, and even a class of terminals may define its own constraints, there exists a strong requirement to describe these properties effectively. A key functionality is the possibility to add new coding formats – e.g., through a registration authority – in a convenient and relatively unbureaucratic way as they arise rather rapidly on the market.

3. Analysis and Comparison of Context Description Formats

3.1. Composite Capabilities/Preference Profiles (CC/PP)

The Composite Capabilities/Preference Profiles (CC/PP) [Kl04] comprises descriptions based on the Resource Description Framework (RDF) which cover device capabilities and user preferences by introducing a two-level hierarchy consisting of components and attributes. Components are groups of attributes with related meaning such as the software or the hardware properties of a terminal. A CC/PP document shall contain at least one component each identified by an rdf:type attribute which indicates the type of the component. Attribute values may be simple, i.e., string and integer or rational numbers, or complex, i.e., set (rdf:bag) or sequence (rdf:seq) of simple values. An Example of a CC/PP description can be found in Listing 1. The given example

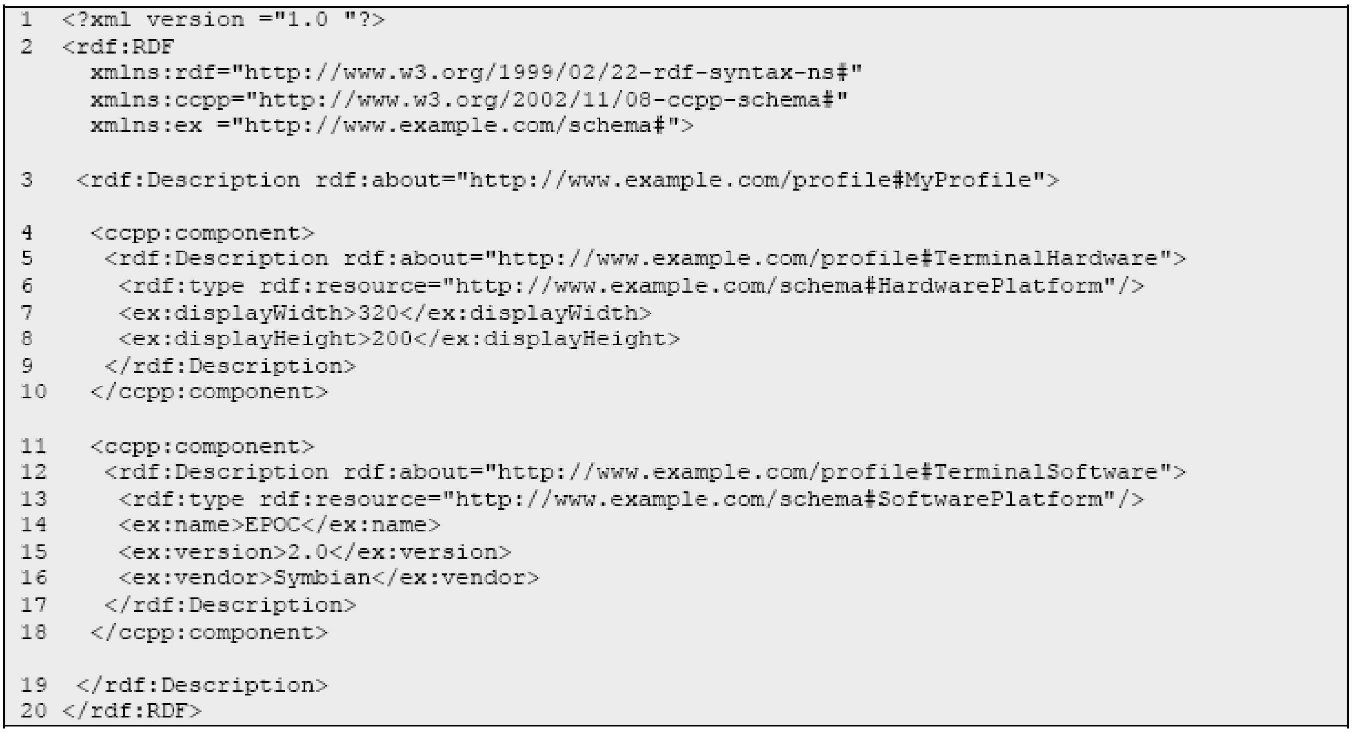
provides a description of the software and hardware platform for a terminal. For the hardware the display width and height is described whereas the software is described with respect to the supported operating system.

As one can see from the example above, CC/PP does not define a vocabulary of terms but provides a common structure for holding any arbitrary vocabulary (i.e., the actual terms for the software and hardware platform are described within a different namespace). Thus, another description format is required which specifies the actual vocabulary, e.g., the User Agent Profile as described next.

3.2. User Agent Profile (UAProf)

The Open Mobile Alliance (OMA) defines the User Agent Profile (UAProf) [OM06] which is based on CC/PP and defines a vocabulary for describing characteristics and capabilities of mainly WAP-enabled mobile devices. The components can be clustered into

—HardwarePlatform;
—SoftwarePlatform;
—BrowserUA;
—NetworkCharacteristics;
—WapCharacteristics; and
—PushCharacteristics.



```
<?xml version ="1.0 "?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ccpp="http://www.w3.org/2002/11/08-ccpp-schema#"
  xmlns:ex ="http://www.example.com/schema#">

 <rdf:Description rdf:about="http://www.example.com/profile#MyProfile">

  <ccpp:component>
   <rdf:Description rdf:about="http://www.example.com/profile#TerminalHardware">
    <rdf:type rdf:resource="http://www.example.com/schema#HardwarePlatform"/>
    <ex:displayWidth>320</ex:displayWidth>
    <ex:displayHeight>200</ex:displayHeight>
   </rdf:Description>
  </ccpp:component>

  <ccpp:component>
   <rdf:Description rdf:about="http://www.example.com/profile#TerminalSoftware">
    <rdf:type rdf:resource="http://www.example.com/schema#SoftwarePlatform"/>
    <ex:name>EPOC</ex:name>
    <ex:version>2.0</ex:version>
    <ex:vendor>Symbian</ex:vendor>
   </rdf:Description>
  </ccpp:component>

 </rdf:Description>
</rdf:RDF>
```

Listing 1. CC/PP example

The component `HardwarePlatform` defines a set of 19 attributes. The attributes provide some general information about the manufacturer and the model of the terminal as well as the included CPU. Detailed information about the display facilities is available which contain the information about the ability of displaying images and displaying color, the resolution in pixels with the possibility to define the ratio of pixel width to pixel height, the resolution in number of characters (columns and lines) and information about fonts. Describing the interaction between human and machine is also done within this section. Therefore, attributes for the keyboard layout and the audio output are specified.

The `SoftwarePlatform` represents the largest component which covers 23 attributes. Attributes which report supported media types are contained as well as preferred languages of the user. Information about the installed operating system and software, audio and video encoders, and support for downloadable software is also considered in this component. Since Java is available for a lot of devices, the definition of the support for Java and for the Common Language Infrastructure (CLI) is facilitated.

The `BrowserUA` is one of the most important components when Web pages are accessed. In the BrowserUA section general information about the browser, HTML and XHTML features, and the abilities to display frames and tables can be given. Furthermore, JavaScript capabilities can be described.

The smallest component, i.e., NetworkCharacteristics, comprises only 4 attributes and provides information about supported and current bearers, security options, and some details about Bluetooth support.

The development of the WAP standard has a long history, hence, a lot of information about WAP is needed to get an overview of the WAP capabilities of a device. The WapCharacteristics include attributes for describing the versions of Wireless Application Protocol (WAP) [WA02], Wireless Markup Language (WML) [WM01] and Wireless Telephony Application (WTA) [WT01] and other relevant information concerning WAP. Also included are attributes which deal with digital rights management concerning standards specified by the OMA.

The PushCharacteristics describe the behavior of data which is pushed to a terminal without a request. Attributes provide information about maximal size, encoding, character set and other elements concerning pushed messages [OM06].

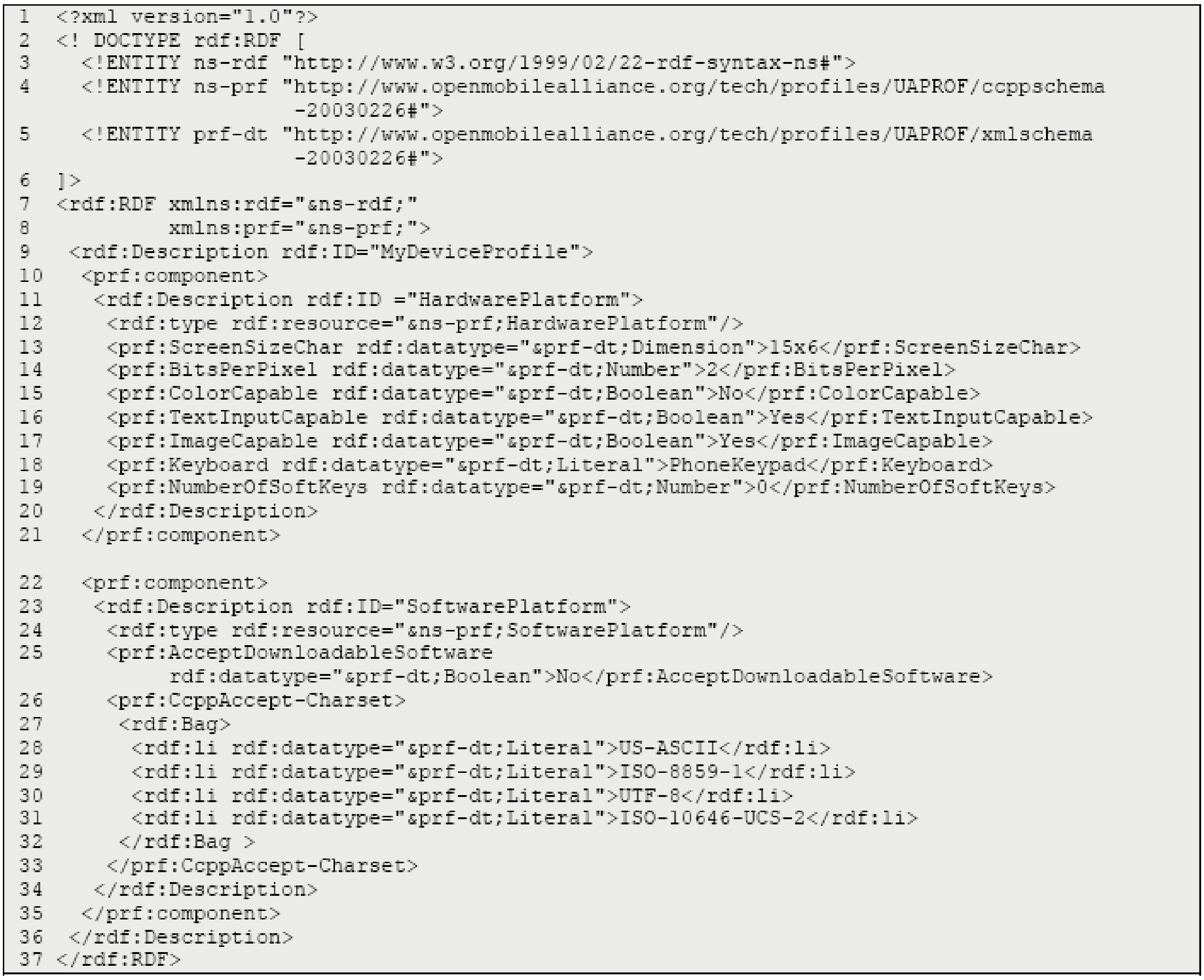
An example of a UAProf description can be found in Listing 2.

The root element of a UAProf must be labeled rdf:RDF and must include xmlns:rdf (line 7) and xmlns:prf(line 8) to refer to the RDF and UAProf namespaces. Furthermore, the root element must contain exactly one child element, namely rdf:Description (line 9), and this element must be identified by the rdf:ID attribute. It is not required that a profile instantiates all the attributes of each component, but for used components, at least one attribute must have a value. Components within a profile are not allowed to have more than once. If a default profile is referred to within the profile, it must contain only one component which must not refer to another URI [OM06]. The example in Listing 2 describes two components. The HardwarePlatform is characterized from line 11 to line 20 and the SoftwarePlatform from line 23 to line 34. Attributes are defined within the rdf:Description tags and have the type defined by rdf:datatype. The multi-valued attribute prf:CcppAccept-Charset can be found at line 26 which uses a rdf:Bag container.

3.3. Usage Environment Description (UED)

The Usage Environment Description (UED) is defined in Part 7 of MPEG-21, i.e., Digital Item Adaptation (DIA) [VT05]. The UED is a very comprehensive vocabulary organized in so-called properties. It is based on XML Schema and its properties can be divided into four categories:

—*User characteristics* provide information pertaining to the user plus his/her usage preferences/history, presentation preferences, accessibility characteristics, and location information including the user’s movement.



```
<?xml version="1.0"?>
<! DOCTYPE rdf:RDF [
  <!ENTITY ns-rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <!ENTITY ns-prf "http://www.openmobilealliance.org/tech/profiles/UAPROF/ccppschema
                  -20030226#">
  <!ENTITY prf-dt "http://www.openmobilealliance.org/tech/profiles/UAPROF/xmlschema
                  -20030226#">
]>
<rdf:RDF xmlns:rdf="&ns-rdf;"
         xmlns:prf="&ns-prf;">
 <rdf:Description rdf:ID="MyDeviceProfile">
  <prf:component>
   <rdf:Description rdf:ID ="HardwarePlatform">
    <rdf:type rdf:resource="&ns-prf;HardwarePlatform"/>
    <prf:ScreenSizeChar rdf:datatype="&prf-dt;Dimension">15x6</prf:ScreenSizeChar>
    <prf:BitsPerPixel rdf:datatype="&prf-dt;Number">2</prf:BitsPerPixel>
    <prf:ColorCapable rdf:datatype="&prf-dt;Boolean">No</prf:ColorCapable>
    <prf:TextInputCapable rdf:datatype="&prf-dt;Boolean">Yes</prf:TextInputCapable>
    <prf:ImageCapable rdf:datatype="&prf-dt;Boolean">Yes</prf:ImageCapable>
    <prf:Keyboard rdf:datatype="&prf-dt;Literal">PhoneKeypad</prf:Keyboard>
    <prf:NumberOfSoftKeys rdf:datatype="&prf-dt;Number">0</prf:NumberOfSoftKeys>
   </rdf:Description>
  </prf:component>

  <prf:component>
   <rdf:Description rdf:ID="SoftwarePlatform">
    <rdf:type rdf:resource="&ns-prf;SoftwarePlatform"/>
    <prf:AcceptDownloadableSoftware
         rdf:datatype="&prf-dt;Boolean">No</prf:AcceptDownloadableSoftware>
    <prf:CcppAccept-Charset>
     <rdf:Bag>
      <rdf:li rdf:datatype="&prf-dt;Literal">US-ASCII</rdf:li>
      <rdf:li rdf:datatype="&prf-dt;Literal">ISO-8859-1</rdf:li>
      <rdf:li rdf:datatype="&prf-dt;Literal">UTF-8</rdf:li>
      <rdf:li rdf:datatype="&prf-dt;Literal">ISO-10646-UCS-2</rdf:li>
     </rdf:Bag >
    </prf:CcppAccept-Charset>
   </rdf:Description>
  </prf:component>
 </rdf:Description>
</rdf:RDF>
```

Listing 2. User Agent Profile with two components [OM06].

—*Terminal capabilities* comprise codec capabilities, input/output characteristics including display/audio output capabilities, and device properties such as device class, power/storage characteristics, data input/output facilities, and CPU capabilities.

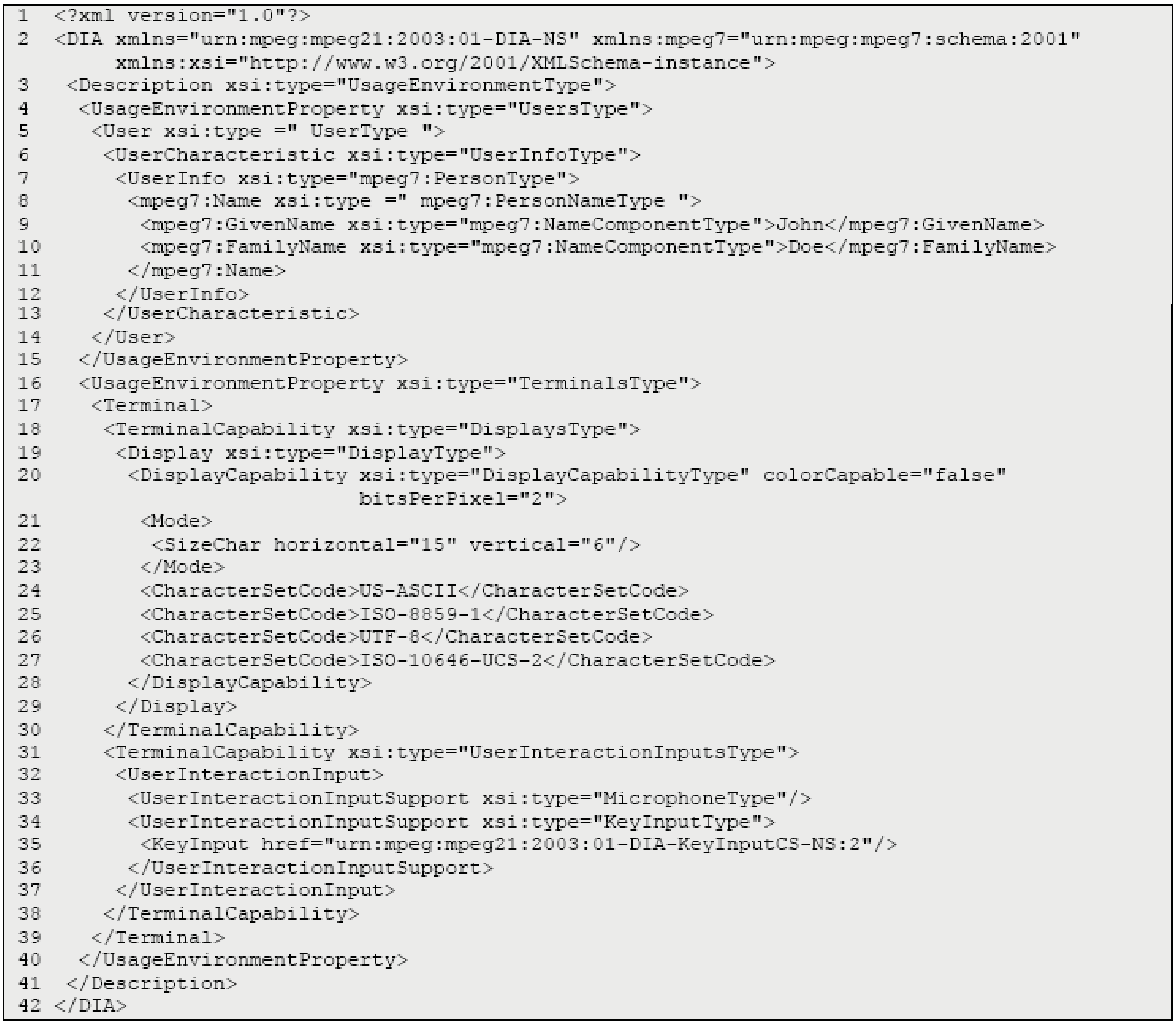
—*Network characteristics* include static and dynamic properties pertaining to the capabilities (e.g., max. bandwidth) and conditions (e.g., available bandwidth) of a network.

—*Natural environment characteristics* provide means for describing lightning conditions, noise level, time, and location.

The *user characteristics* enable a variety of applications including adaptive content selection as well as personalization. Therefore, DIA provides means for describing general information about the user as well as her/his preferences and usage history which has been re-used from the MPEG-7 tool set. Furthermore, the presentation preferences (e.g., format or modality of the multimedia content) belong to the user characteristics. Other important aspects of the user are accessibility (i.e., certain visual or auditory impairments) and location (i.e., mobility and destination) characteristics. The former allows for adaptive delivery of multimedia content according to a user's impairment whereas the latter is important for location-based services.

The concept of terminal in DIA is rather generic and is representative for all kind of devices within the delivery chain including also server and intermediary network nodes. The *terminal capabilities* can be classified into three categories. First, the *codec capabilities* define the encoding and decoding capabilities of a terminal. As such, the supported codecs of the requesting device can be identified which may result in one or more transcoding steps of the original multimedia content. Second, the *input-output characteristics* comprise display capabilities (e.g., resolution or color capability), audio output capabilities (e.g., frequency range or number of output channels), and user interaction inputs (e.g., keyboard or touch screen). These tools control the presentation layout or the user interface of the multimedia content. Third, the *device properties* cover a wide range of tools including power and storage characteristics, and CPU benchmark measures, among others. The power characteristics include information such as remaining battery capacity which may be considered by a sending device in such a way as to adapt its transmission strategy in order to maximize the battery lifetime. Storage characteristics (e.g., transfer rate or size) may influence how the Digital Item may be consumed, e.g., whether it needs to be streamed or locally stored. The benchmark tool enables the description of the CPU performance which could be used to infer a device's capabilities of handling a certain type of media possibly encoded at a certain quality level.

For *network characteristics* two major categories can be identified, namely static capabilities and dynamically varying conditions. The former, the network capabilities, include attributes describing the maximum capacity and the minimum guaranteed bandwidth the network can provide. Additionally, information about in-sequence delivery and how erroneous packets are handled can be signaled using this tool. The latter, i.e., network conditions, provide means for describing the currently available



```
1  <?xml version="1.0"?>
2  <DIA xmlns="urn:mpeg:mpeg21:2003:01-DIA-NS" xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
3   <Description xsi:type="UsageEnvironmentType">
4    <UsageEnvironmentProperty xsi:type="UsersType">
5     <User xsi:type =" UserType ">
6      <UserCharacteristic xsi:type="UserInfoType">
7       <UserInfo xsi:type="mpeg7:PersonType">
8        <mpeg7:Name xsi:type =" mpeg7:PersonNameType ">
9         <mpeg7:GivenName xsi:type="mpeg7:NameComponentType">John</mpeg7:GivenName>
10        <mpeg7:FamilyName xsi:type="mpeg7:NameComponentType">Doe</mpeg7:FamilyName>
11       </mpeg7:Name>
12      </UserInfo>
13     </UserCharacteristic>
14    </User>
15   </UsageEnvironmentProperty>
16   <UsageEnvironmentProperty xsi:type="TerminalsType">
17    <Terminal>
18     <TerminalCapability xsi:type="DisplaysType">
19      <Display xsi:type="DisplayType">
20       <DisplayCapability xsi:type="DisplayCapabilityType" colorCapable="false"
                            bitsPerPixel="2">
21        <Mode>
22         <SizeChar horizontal="15" vertical="6"/>
23        </Mode>
24        <CharacterSetCode>US-ASCII</CharacterSetCode>
25        <CharacterSetCode>ISO-8859-1</CharacterSetCode>
26        <CharacterSetCode>UTF-8</CharacterSetCode>
27        <CharacterSetCode>ISO-10646-UCS-2</CharacterSetCode>
28       </DisplayCapability>
29      </Display>
30     </TerminalCapability>
31     <TerminalCapability xsi:type="UserInteractionInputsType">
32      <UserInteractionInput>
33       <UserInteractionInputSupport xsi:type="MicrophoneType"/>
34       <UserInteractionInputSupport xsi:type="KeyInputType">
35        <KeyInput href="urn:mpeg:mpeg21:2003:01-DIA-KeyInputCS-NS:2"/>
36       </UserInteractionInputSupport>
37      </UserInteractionInput>
38     </TerminalCapability>
39    </Terminal>
40   </UsageEnvironmentProperty>
41  </Description>
42 </DIA>
```

Listing 3. Simple UED example

bandwidth, error, and delay. The main objective of these tools is to enable improved transmission efficiency and media quality optimization with respect to network constraints.

Finally, the *natural environment characteristics* pertain to the physical environmental conditions around a user such as lighting conditions, noise level, or time and location where media resources are consumed or/and processed.

Listing 3 provides an example with information similar to the UAProf in Listing 2.

The `UsersType` of the four derived types of `UsageEnvironmentPropertyBaseType` is used (line 4). To distinguish description tools that are intended to describe multiple entities from those that are intended to describe a single entity, the plural form of that entity is used in the naming of the tool. For instance, the `UsersType` in line 4 can have several instances of `UserType` (line 5). The example in Listing 5.14 describes a user by the given name and the family name using abstract data types of DIA (e.g. `UserInfoType`) and MPEG7 (e.g. `mpeg7:PersonType`). The concept of abstract types is indicated by `xsi:type` and is used in MPEG-7 and MPEG-21 respectively. Further, the presence of MPEG-7 data types can be seen in lines 8 to 11. By using the `xsi:type=”TerminalsType”` the terminal of Listing 2 is described using the UED standard (line 16 to line 40). Listing 3 uses only UED terminology which is similar to that of UAProf and thus the meaning of the elements can easily be derived from Listing 2.

3.4. Delivery Context Ontology (DCO)

The Delivery Context Ontology (DCO) [LF08] is based the Web Ontology Language (OWL) [Mv04] and provides a set of characteristics describing the context in which media resources are consumed, namely:

—Supported audio, video, and image formats.
—Usable display pixels, supported markups, and location provider support.
—Information about the device, environment, and user.

The first category provides means for describing the delivery context with respect to the supported audio, video, and image formats. Therefore, colloquial names, MIME types, name, and version can be given.

Only the usable display pixels are provided as they may be less than the total number available pixels due to the presence of additional items on the display. The markup languages supported by the delivery context are described in a similar way as the audio/video/image formats, i.e., by colloquial names, MIME types, name, and version as well as public and system identifier. Additionally, a so-called markup module can be described. The location provider support gives information about the geo-location of a delivery context entity (e.g., an end-user terminal).

Finally, the information about the device, environment, and user may contain a lot of different characteristics and some of them are briefly described in the following. The *device* properties can be categorized into hardware and software. For hardware, the following properties can be described: battery capacity, current level, and whether it is being charged; Bluetooth support; built-in and extension memory; cameras; input/output character sets; input devices; network support; number of soft keys which are programmable; CPU; audio output support, voice recognition support; text input type. The software properties include information about the user agent; whether the delivery context provides support for Java, MMS, WAP Push, and which operating system; and the capabilities of the browser. The environment covers information about the network and location. Furthermore, it

provides information about the devices' course made good in degrees relative to true north and its speed. Interestingly, this information is indeed comparable with the mobility characteristics as defined in MPEG-21 DIA UED. Finally, the *user* aspects are not (yet) defined as part of the DCO but are foreseen as so called Delivery Context Extensions and particularly using defacto standards such as Friend of a Friend (FOAF) [FOA07]. Another alternative would be the user characteristics as defined in MPEG-21 DIA UED.

3.5. Analysis/Comparison

The previous sections provided an overview of existing standards for delivery context description formats. In this section we will analyse differences and commonalities of these formats. First of all, and most importantly, all standards make use of XML that provides extensibility but UED is based on XML Schema whereas CC/PP and, consequently, UAProf are based on RDF. The most recent standard in this series is DCO which adopted already OWL which is based on RDF. Hence, we observe an incompatibility at the level of technology used for these description formats, mainly between XML Schema and RDF. Although it is possible to provide a high-level mapping between these two technologies, the mapping of concrete schemas/instances itself is a difficult and cumbersome task [HL01].

The second observation we made was that there are only a few characteristics or capabilities that are common across all delivery context description formats in question, e.g., display capabilities and file/coding formats. However, there is sometimes a huge difference in the actual syntax used. For example, display resolution described as `horizontal=1024` and `vertical=768` versus `1024x768` or using MIME types for file/coding formats versus classification schemes (i.e., URNs).

Finally, CC/PP defines only a basic structure (i.e., components and attributes) without specifying a particular vocabulary of terms. UAProf adopts CC/PP and provides a concrete vocabulary mainly targeting WAP-enabled mobile devices. A repository of some specific device profiles is available [W308]. Other industry adoptions of CC/PP are not known but some are envisaged and documented in Annex E of [Ki07]. The UED defines both the structure (i.e., properties) and a comprehensive vocabulary while DCO defines an ontology including not only a vocabulary of delivery context terms but also basic measure units.

In conclusion, there is a need to describe the relationship between commonalities of the different delivery context description formats, i.e., a mapping model which will be described in the next section. To the best of our knowledge, such a mapping model has not been published yet.

4. Mapping Model

UAProf and UED are based on different data models as the former is based on RDF whereas the latter is based on XML Schema with having their pros and cons [HL01]. That is, RDF provides support for rich semantic descriptions but provides limited support for the specification of local usage constraints, e.g., cardinality and datatype constraints. On the other hand, XML Schema provides support for explicit structural, cardinality and datatype constraints but provides little support for the semantic knowledge necessary to enable a flexible mapping between metadata domains.

The main issue is to find a suitable technology for the mapping process which includes the advantages of both standards. Basically, the mapping can be performed by two approaches as discussed in the following and depicted in Figure 2:

Direct mapping model: creating mapping functions that perform direct mapping from one standard to another.
Integration model: integrating both models into a new one with functions to convert between this new model and the initial model.

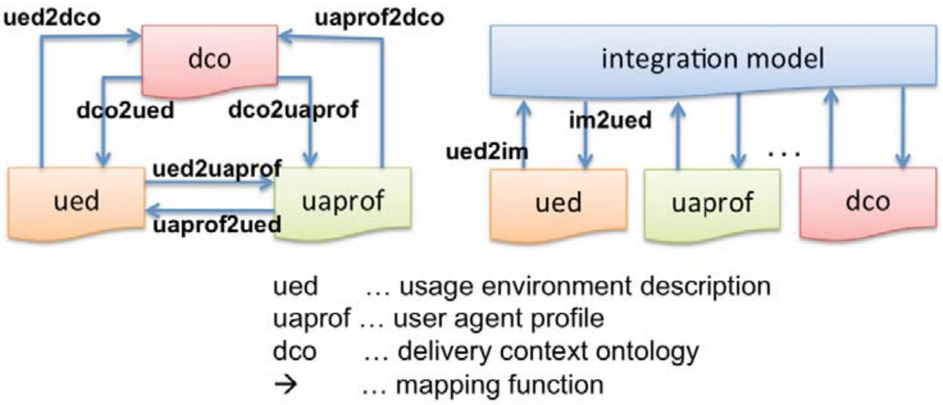


Figure 2. Direct Mapping Model vs. Integration Model

As the *direct mapping model* provides an explicit mapping from one format to another format it lacks of flexibility with respect to the integration of other formats. Thus, it can be only applied for specific solutions whereby the number of explicit mappings increases exponentially with the number of formats between which mappings should be provided.

The *integration model* defines a common interface that allows the provisioning of the individual description formats. For new formats to be added, only the methods for converting to/from this model needs to be implemented without taking into account the existence of the other formats. Thus, the number of mappings increases linearly with the number of formats. However, the integration model needs to be implemented with a certain technology and those in question are XML Schema, RDF, or even OWL:

—One could define an **XML Schema** that covers all components of each standard and well-established XPath/XML processors could be used to extract the required data for a certain standard. Unfortunately, datatype or value range incompatibilities (e.g., UED: `colorCapable={true,false}` vs. UAProf: `ColorCapable={Yes,No}`) cannot be represented with XML Schema which requires external knowledge to be provided. Thus, it would be better to use tools which are able to express the relations between, e.g., datatypes or attribute values.

—**OWL** is based on RDF and provides means to describe the relationship between classes and properties, e.g., classes can be declared distinct or equal, restrictions on properties can be defined as transitive or functional, or the use of cardinality restricts the number of values which can be associated to properties.

4.1. Mapping Levels

The relationships between different delivery context description formats can be found at different levels within the entities of their respective schemas (i.e., XML Schema of OWL). In this paper we introduce four different mapping levels, namely *component*, *datatype*, *element*, and *value*. An example thereof is shown in Table 1.

The *component* mapping level tries to map a predefined group of elements/attributes (e.g. `prf:NetworkCharacteristics`) to similar group of the other description format (e.g. `dia:NetworkType`). Difficulties may arise in case the structure is different, e.g., one has only attributes defined whereas the other includes also elements within a nested structure. Thus, one needs to dig a level deeper and the *element* mapping level tries to map attributes/elements with equal semantics but possibly different syntax, i.e., different tag names. Note that a mapping at the component level is not always sufficient as indicated above which requires describing relationships between individual elements/attributes or even beyond.

The *datatype* mapping level tries to map datatypes with equal domains but different syntax whereas the *value* mapping level tries to map datatypes with different domains but equal semantics.

An example of mapping attributes with the equal semantics and Boolean values is described in Listing 4 which maps the `prf:ColorCapable` to the `map:colorCapable` (assuming `map:colorCapable` is the RDF/XML representation of the `colorCapable` attribute of the `dia:DisplayCapabilityType`).

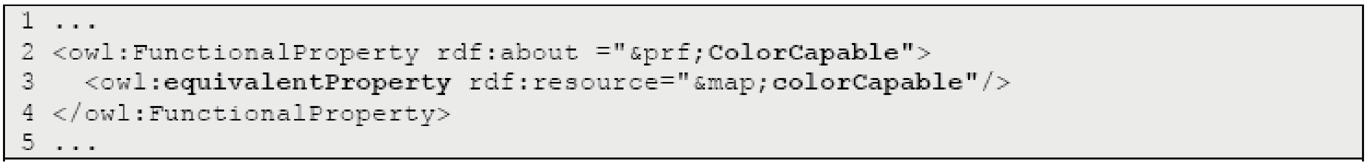
A problem arises as both datatypes are Boolean types but with different syntax: while `xsd:boolean` (as used within UED) accepts true and false, `prf-dt:Boolean` (as used within UAProf) accepts only `Yes` and `No` which requires an appropriate mapping. Listing 5 shows one possibility for such a mapping of these Boolean datatypes. Lines 2 to 6 and lines 8 to 12 describe properties to create a relation between a new defined resource for a Boolean value (prefixed by `btd`) and the Boolean values used by UAProf and UED. The rest provides the mapping between the values and the `bdt` Boolean datatypes.

Another example is shown in Listing 6 which maps a Uniform Resource Name (URN) identifying a certain key input type to the equivalent string representation of UAProf. The usage of URNs to uniquely identify predefined terms is heavily used within UED.

Datatypes such as `prf-dt:Number` and `xsd:nonNegativeInteger` can be mapped directly to each other because both cover the same range of values. However, integer values also raise problems when a mapping from `dia:bitsPerPixel` to `prf:BitsPerPixel` is provided because the UED standard defines an `xsd:integer` datatype and the UAProf standard uses the `prf-dt:Number` datatype which is equal to `xsd:nonNegativeInteger`. Of course, it is unlikely to describe the number of bits per pixel or the horizontal and vertical resolution with a negative number but there is the possibility to do that. OWL lacks to offer appropriate tools to describe what to do if such problems arise.

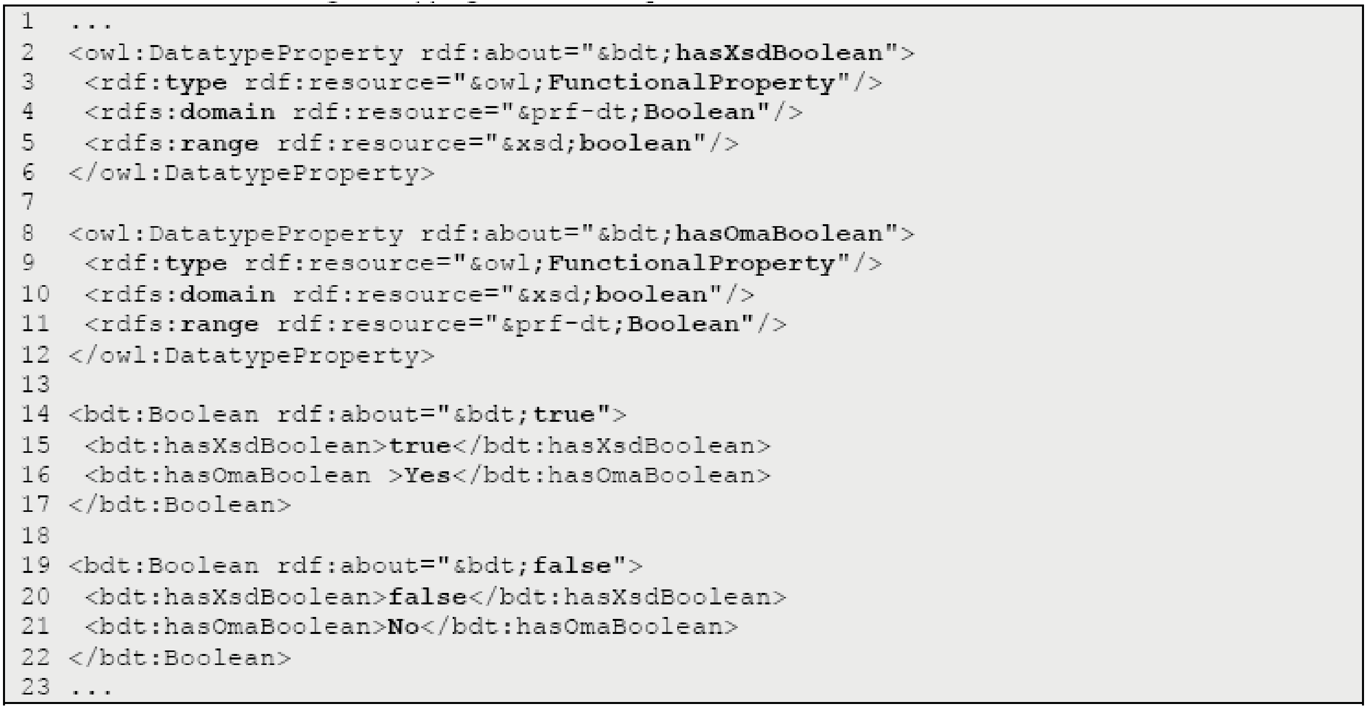
Level	UAProf Example	UED Example
Component	`prf:NetworkCharacteristics`	`dia:NetworkType`
Element	`prf:InputCharSet`	`dia:CharacterSetCode`
Datatype	`prf-dt:Boolean`	`xsd:Boolean`
Value	`Yes`	`true`

Table 1. Example of Mapping Levels for Network Characteristics



```
...
<owl:FunctionalProperty rdf:about ="&prf;ColorCapable">
  <owl:equivalentProperty rdf:resource="&map;colorCapable"/>
</owl:FunctionalProperty>
...
```

Listing 4. Mapping `prf:ColorCapable` to `map:colorCapable`



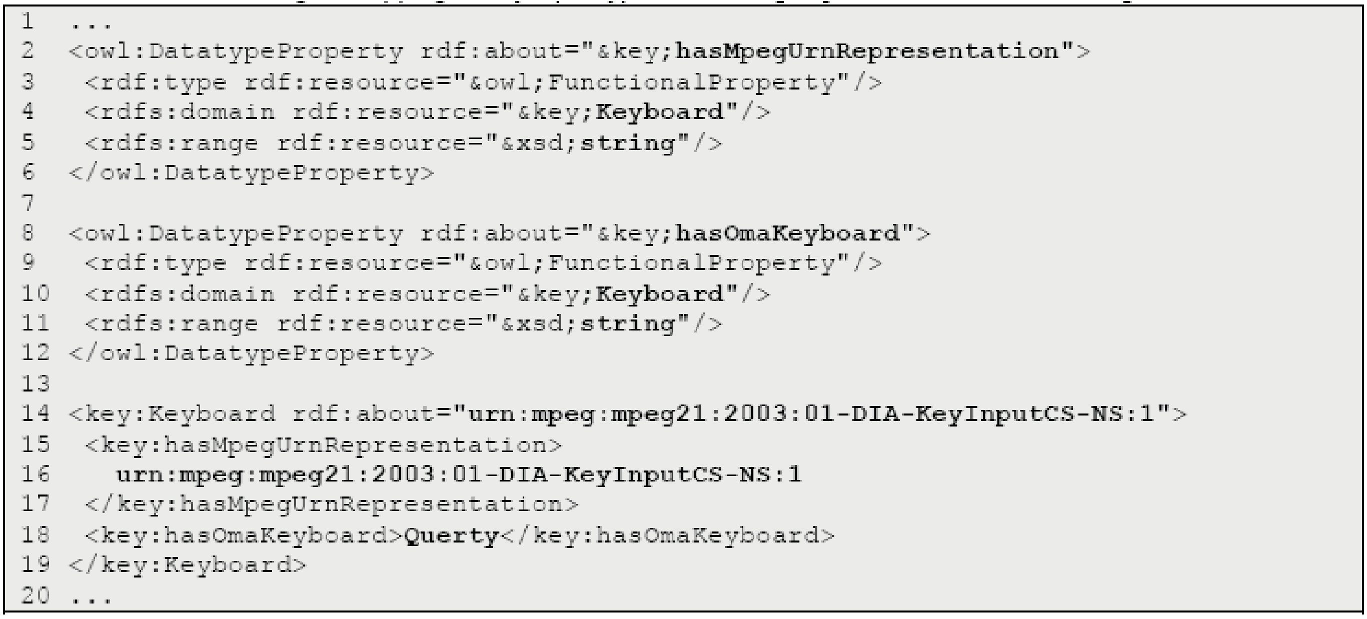
```
...
<owl:DatatypeProperty rdf:about="&bdt;hasXsdBoolean">
 <rdf:type rdf:resource="&owl;FunctionalProperty"/>
 <rdfs:domain rdf:resource="&prf-dt;Boolean"/>
 <rdfs:range rdf:resource="&xsd;boolean"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&bdt;hasOmaBoolean">
 <rdf:type rdf:resource="&owl;FunctionalProperty"/>
 <rdfs:domain rdf:resource="&xsd;boolean"/>
 <rdfs:range rdf:resource="&prf-dt;Boolean"/>
</owl:DatatypeProperty>

<bdt:Boolean rdf:about="&bdt;true">
 <bdt:hasXsdBoolean>true</bdt:hasXsdBoolean>
 <bdt:hasOmaBoolean >Yes</bdt:hasOmaBoolean>
</bdt:Boolean>

<bdt:Boolean rdf:about="&bdt;false">
 <bdt:hasXsdBoolean>false</bdt:hasXsdBoolean>
 <bdt:hasOmaBoolean>No</bdt:hasOmaBoolean>
</bdt:Boolean>
...
```

Listing 5. Mapping the values of `prf-dt:Boolean` and `xsd:Boolean`



```
...
<owl:DatatypeProperty rdf:about="&key;hasMpegUrnRepresentation">
 <rdf:type rdf:resource="&owl;FunctionalProperty"/>
 <rdfs:domain rdf:resource="&key;Keyboard"/>
 <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&key;hasOmaKeyboard">
 <rdf:type rdf:resource="&owl;FunctionalProperty"/>
 <rdfs:domain rdf:resource="&key;Keyboard"/>
 <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>

<key:Keyboard rdf:about="urn:mpeg:mpeg21:2003:01-DIA-KeyInputCS-NS:1">
 <key:hasMpegUrnRepresentation>
   urn:mpeg:mpeg21:2003:01-DIA-KeyInputCS-NS:1
 </key:hasMpegUrnRepresentation>
 <key:hasOmaKeyboard>Querty</key:hasOmaKeyboard>
</key:Keyboard>
...
```

Listing 6. Mapping of key input types: `DIA-KeyInputCS-NS:1` and `Querty`

Another issue is raised with the datatype `prf-dt:Dimension` which is used to describe the resolution of a terminal's display within one string of the form "`HorizontalResolution`"x"`VerticalResolution`" (e.g., 1024x768). This means that two values of the UED (i.e., horizontal and vertical attributes of the Resolution element) are combined to represent one value within UAProf. Thus, external tools are needed to enable such mappings.

4.2. Mapping Classes

In practice, the tag names and datatypes of different description formats can be clustered into four classes which are described in the following.

Direct. Elements falling into this class have equal semantics and compatible datatypes with equal domains but may differ in their syntax (i.e., tag name). For example, `dia:bitsPerPixel` of type `xsd:integer` and `prf:BitsPerPixel` of type `prf-dt:Number` where these datatypes are compatible. Another example is the `dia:CharacterSetCode` of type `mpeg7:characterSetCode` (equal to `xsd:string`) and `prf:CcppAccept-Charset` of type `prfdt:Literal` (equal to `xsd:anySimpleType`) which are used to store string representations of the supported character sets.

Advance. The class advance comprises elements describing the same concept (i.e., equal semantics) but with different, non-compatible datatypes and/or domains. Thus, the actual format is that much different and requires major changes if mapped from one format to the other. For example, the `dia:Resolution` includes two attributes (horizontal and vertical) for describing the resolution of a screen whereas `prf:SreenSize` uses only one value (e.g., 480x320). Another example is the usage of classification schemes versus MIME types as detailed in Section 4.3. Thus, an advanced mapping mechanism is required.

Derive. This class includes mappings where element values can be derived from one or more elements of the respective other description format. The difference to the advance class is that for the derive class the semantic equality is not necessarily a requirement. For example, `prf:SoundOutputCapable` indicates whether a terminal is able to output sound which could be derived from only the presence of a `dia:AudioOutputCapability` element.

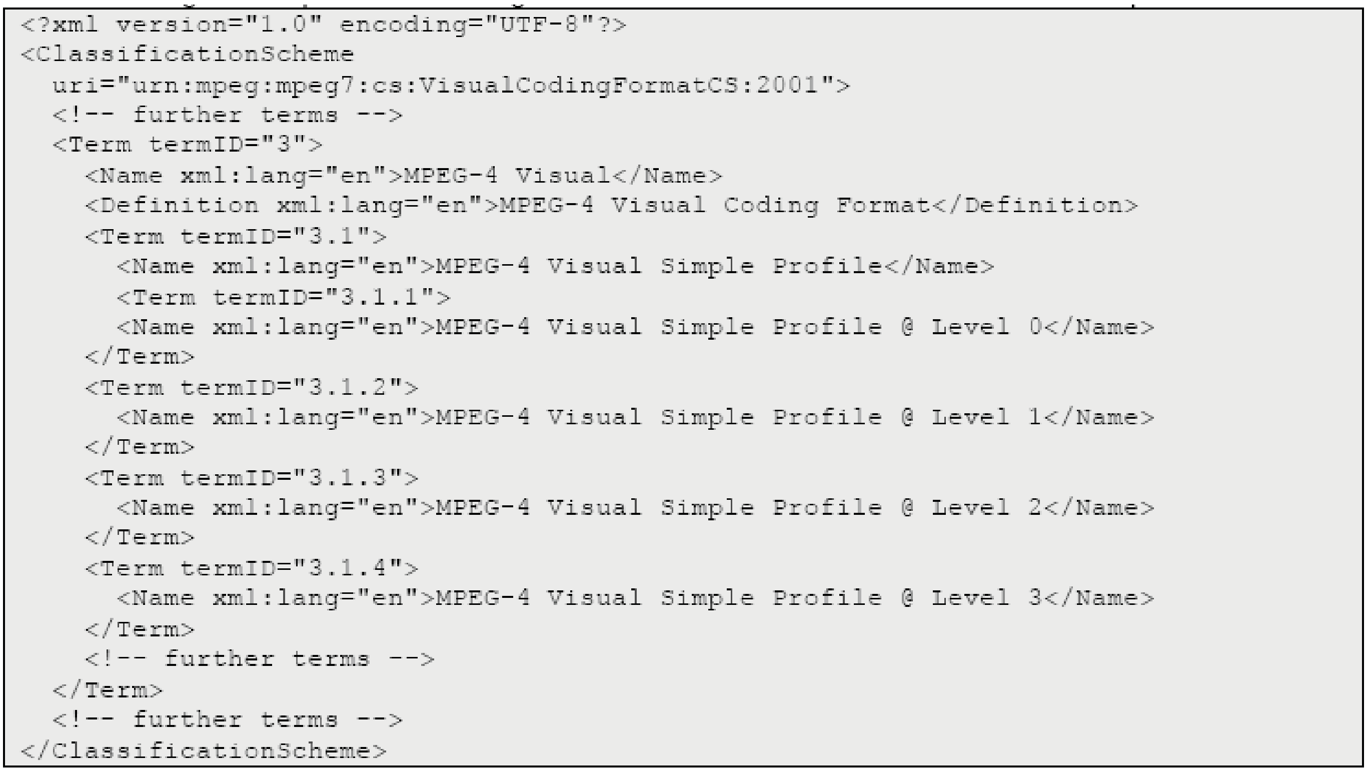
Extend. Elements that cannot be mapped directly, in an advanced way through additional mapping rules, or derived from other elements require proprietary extensions of the respective other description format. For example, properties defined within UAProf but not defined in UED require an extension of the UED schema by adding additional elements and datatypes representing these UAProf properties. In our example, the UAProf standard defines six components and 77 elements which have been mapped – with respect to UED – to the classes described above (quantities in brackets): direct (4), advance (7), derive (4), and extend (62). The specific support for mobile phones within UAProf causes the high number within the class extend whereas UED does not provide means for describing WAP or push characteristics. One could now argue that such a mapping is not required. Please note that for most of the application scenarios – in particular, multimedia content adaptation – the required elements/attributes/tags fall into direct, advance, and derive classes, e.g., adaptation to screen size, codec, bitrate which are covered in all delivery context description formats. Therefore, the class ‘extend’ can be ignored for this kind of applications.

4.3. Additional Mapping Rules for Coding Formats

This section specifically discusses means for describing supported coding formats as this seem to be an inherent part of each delivery context description standard. Unfortunately, the standards in question adopt different technologies for describing this property. In particular, the CC/PP and, thus, UAProf adopts an approach which is based on MIME media types [FB96] whereas MPEG-21 UED relies on classification schemes introduced within MPEG-7 [MSS02].

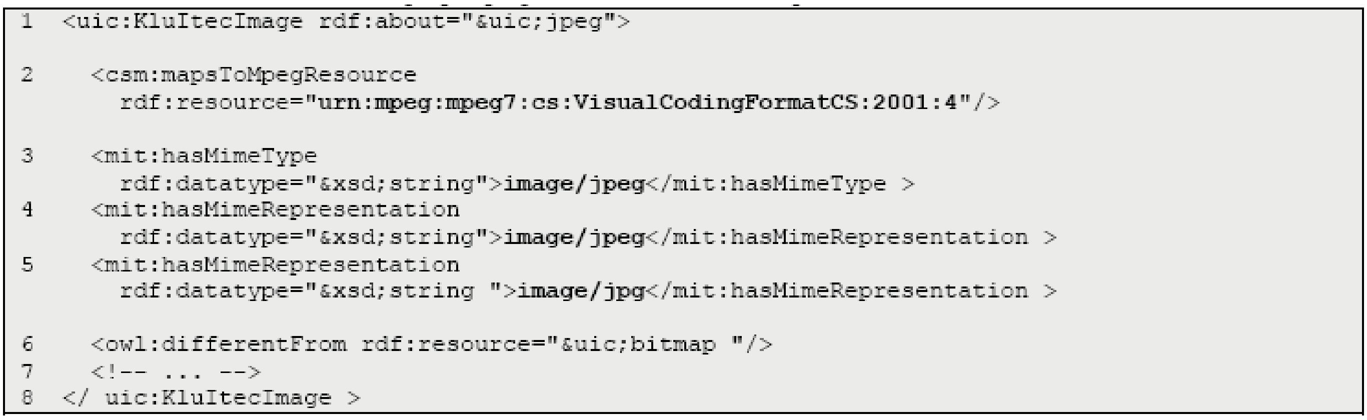
MIME media types are well known within the Internet – thanks to its adoption for HTTP, etc. – and comprises five discrete top-level media types, i.e., text, image, audio, video, and application, as well as two composite top-level media types, i.e., multipart and message. These top-level media types are referred to as content types and the actual coding format is identified through the content sub-type (e.g., `video/mp4`). It is also possible to associate an arbitrary number of parameters in form of key-value pairs to media types which could be used to describe specifics usually defined within profiles/levels. However, most of the audio/video/image MIME type definition does not make use of this possibility. Thus, it is up to the application to identify the exact data format by other means. For example, `video/mp4` may contain a bitstream compliant to MPEG-4 Part 2 (Visual) or MPEG-4 Part 10 (Advanced Video Coding), not mentioning all the available profile/level combinations.

An MPEG classification scheme is an XML document that may contain terms – identifiable by URN – and corresponding definitions of arbitrary semantics in a hierarchically fashion. Thus, it is also possible to include profile/levels of a certain coding format as shown in Listing 7. Although classification schemes are extensible as they are based on XML there is a lack of an approved registration authority to accommodate future coding formats. However, the European Broadcasting Union (EBU) maintains a set of classification schemes used within their specifications (including TV-Anytime) [EBU09].



```
<?xml version="1.0" encoding="UTF-8"?>
<ClassificationScheme
  uri="urn:mpeg:mpeg7:cs:VisualCodingFormatCS:2001">
  <!-- further terms -->
  <Term termID="3">
    <Name xml:lang="en">MPEG-4 Visual</Name>
    <Definition xml:lang="en">MPEG-4 Visual Coding Format</Definition>
    <Term termID="3.1">
      <Name xml:lang="en">MPEG-4 Visual Simple Profile</Name>
      <Term termID="3.1.1">
      <Name xml:lang="en">MPEG-4 Visual Simple Profile @ Level 0</Name>
    </Term>
    <Term termID="3.1.2">
      <Name xml:lang="en">MPEG-4 Visual Simple Profile @ Level 1</Name>
    </Term>
    <Term termID="3.1.3">
      <Name xml:lang="en">MPEG-4 Visual Simple Profile @ Level 2</Name>
    </Term>
    <Term termID="3.1.4">
      <Name xml:lang="en">MPEG-4 Visual Simple Profile @ Level 3</Name>
    </Term>
    <!-- further terms -->
  </Term>
  <!-- further terms -->
</ClassificationScheme>
```

Listing 7. Excerpt of Visual Coding Format Classification Scheme MPEG-4 Visual Simple Profile



```
1  <uic:KluItecImage rdf:about="&uic;jpeg">

2    <csm:mapsToMpegResource
       rdf:resource="urn:mpeg:mpeg7:cs:VisualCodingFormatCS:2001:4"/>

3    <mit:hasMimeType
       rdf:datatype="&xsd;string">image/jpeg</mit:hasMimeType >
4    <mit:hasMimeRepresentation
       rdf:datatype="&xsd;string">image/jpeg</mit:hasMimeRepresentation >
5    <mit:hasMimeRepresentation
       rdf:datatype="&xsd;string ">image/jpg</mit:hasMimeRepresentation >

6    <owl:differentFrom rdf:resource="&uic;bitmap "/>
7    <!-- ... -->
8  </ uic:KluItecImage >
```

Listing 8. Mapping MIME media type `image/jpeg` to `urn:mpeg:mpeg7:cs:VisualCodingFormatCS:2001:4`

Listing 8 describes the mapping from the MIME type `image/jpeg` to an equivalent URN representation which is used in UED. Line 1 of Listing 8 defines a resource for the MIME type image/jpeg. The other prefixes csm, mit and owl are shortcuts for the resources where the used vocabularies are defined (e.g., owl as shortcut for http://www.w3.org/2002/07/owl#). Line 2 defines the mapping from the resource `&uic;jpeg` to the resource `urn:mpeg:mpeg7:cs:VisualCodingFormatCS:2001:4` which represents JPEG as a reference to a classification scheme term. Line 3 defines which string representation for JPEG should be used in UAProf descriptions. Lines 4 and 5 define all representations for the image/jpeg MIME media type. Line 6 uses standard OWL syntax to define that the resource `&uic;jpeg` is different from the resource `&uic;bitmap`.

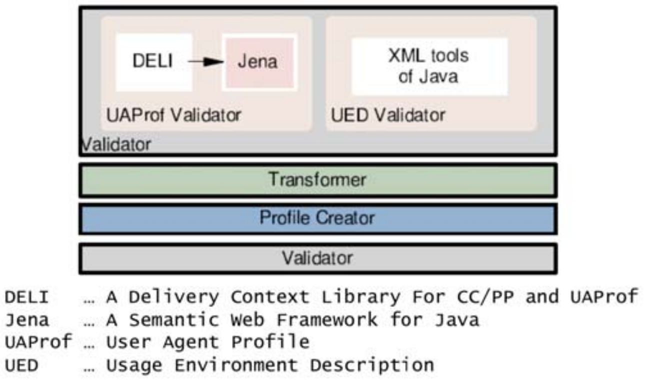

DELI
Jena
XML tools of Java
UAProf Validator
UED Validator
Validator
Transformer
Profile Creator
Validator
DELI ... A Delivery Context Library For CC/PP and UAProf
Jena ... A Semantic Web Framework for Java
UAProf ... User Agent Profile
UED ... Usage Environment Description

Figure 4. Architecture of the Validator

5. Implementation Details

The aim of this section is to provide an overview of our implementation that currently performs a mapping between UAProf descriptions and UEDs (and vice-versa). The high-level architecture is depicted in Figure 3 and comprises three components:

—Validator.
—Transformer.
—Profile Creator.

The *Validator* is responsible for validating incoming and outgoing UAProfs and UEDs. If the received profile is a UED profile, a transformation to an RDF/XML document is needed for further processing which the *Transformer* accomplishes. UAProfs need not be translated because they are already written in RDF/XML syntax which is a requirement of the profile creator. The *Profile Creator* queries data from the profile data which is available in a consistent syntax and creates the desired profile as output which is again checked by the validator before it is delivered. These three components are further detailed in the following.

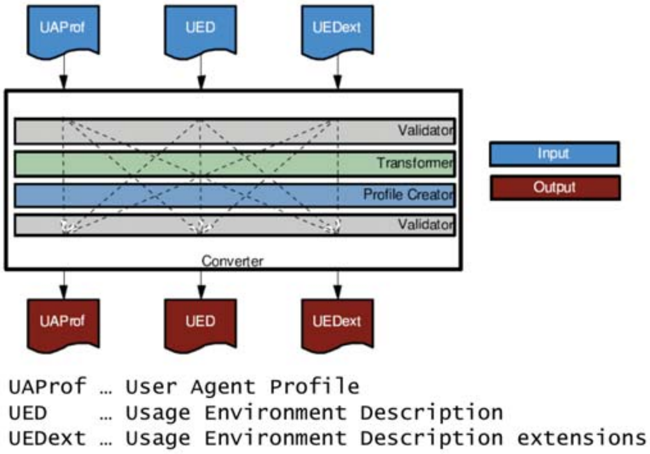

UAProf
UED
UEDext
Validator
Transformer
Profile Creator
Validator
Converter
Input
Output
UAProf
UED
UEDext
UAProf ... User Agent Profile
UED ... Usage Environment Description
UEDext ... Usage Environment Description extensions

Figure 3. High-Level Architecture of the UAProf/UED Mapping Implementation

Validator (cf. Figure 4). The purpose of this component is to validate instances against its specification. This is performed both for inputs and outputs of our implementation. For UAProf we have integrated the DElivery context LIbrary (DELI) [Bu08] which is one of some rarely available tools that are able to validate UAProfs and to extract data from these documents. As the UED schema is an XML schema we have used standard XML schema validation tools such as the Java built-in XML validation Application Programming Interface (API).

Transformer (cf. Figure 5). The transformer is responsible for translating the input instances into an integration model based on RDF as already introduced in Section 4. Therefore, we have implemented style sheets based on the Extensible Stylesheet Language Transformation (XSLT) [Cl99], i.e., only one style sheet is required for each delivery context description language keeping the overall approach scalable.

Profile Creator (cf. Figure 6). Finally, this component generates the designated target delivery context description based on the integration model. In order to query the RDF-based integration model we have used SPARQL Protocol And RDF Query Language (SPARQL) [PS08] and A SPARQL Processor for Jena (ARQ) [HP08] as the actual query engine. The implementation adopts predefined templates and queries to generate desired output format based on the integration model.

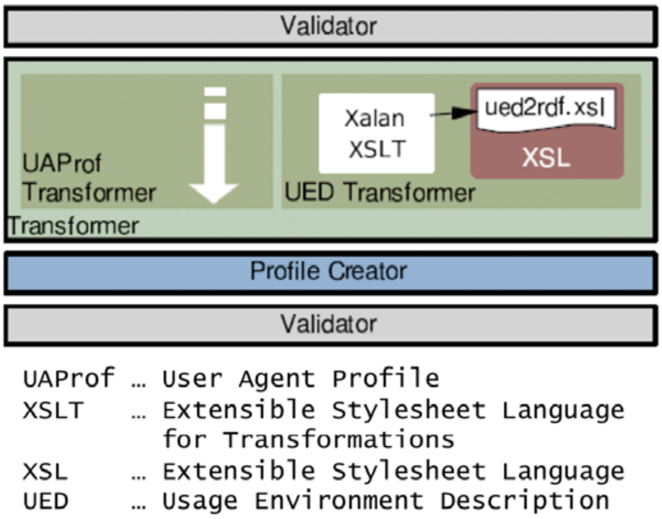

Validator
UAProf Transformer
Xalan XSLT
ued2rdf.xsl
XSL
UED Transformer
Transformer
Profile Creator
Validator
UAProf ... User Agent Profile
XSLT ... Extensible Stylesheet Language for Transformations
XSL ... Extensible Stylesheet Language
UED ... Usage Environment Description

Figure 5. Architecture of the Transformer

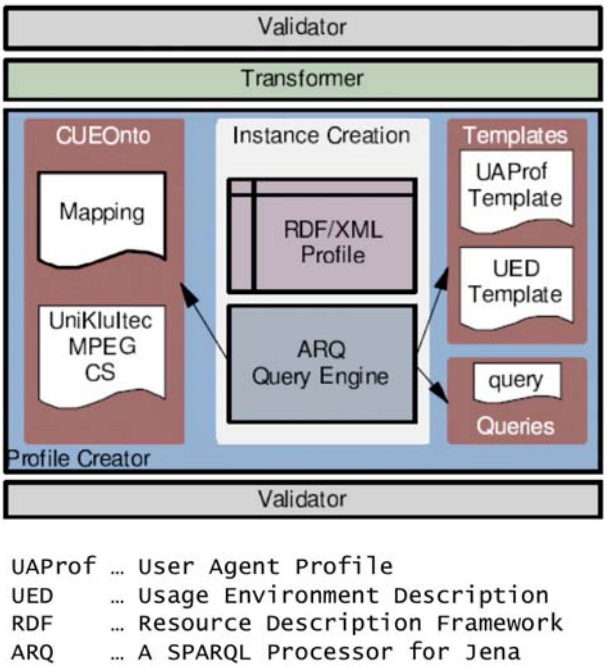


Figure 6. Architecture of the Profile Creator

6. Conclusions and Future Work

In this paper we have presented a model that allows one to map context delivery descriptions between different formats (e.g., OMA UAProf and MPEG-21 UED) that are based on different technologies (i.e., XML Schema and RDF/OWL). For this model we have investigated state-of-the-art terminals in terms hardware and software capabilities as well as analyzed and compared existing delivery context description formats. Based on this analysis and comparison we concluded that there is a need for describing the commonalities and relationships between these description formats using a common model, i.e., following the integration model approach introduced earlier. The mapping model clusters the properties of the individual description formats based on their levels into four classes, namely *direct*, *advance*, *derive*, and *extend*. Based on these classes we have defined the integration model and formulated templates (i.e., using SPARQL and OWL) to query information from the integration model in order to generate the target context delivery format. The feasibility of the approach has been validated through a prototype and implementation details have been described in this paper.

The major findings can be summarized as follows:

—The overlap between different context delivery description formats is not that huge as expected but is clustered around those properties which are considered by the majority of applications areas (e.g., screen size, coding formats, etc.).

—Hence, the classes *direct*, *advance*, and *derive* are sufficient for most of the application areas.

—The relationship between different delivery context description formats needs to be described manually with respect to an integration model (i.e., the mapping function) and requires a thorough analysis of these formats which is sometimes cumbersome (cf. also [HL01]). For each format the mapping functions need to be defined only once with respect to an integration model.

—However, in this paper we have demonstrated that it is feasible – in principle – but requires the integration of many XML-based technologies ranging from XML Schema and RDF to SPARQL and OWL.

The following items are to be considered for future work. The integration of an OWL reasoner may be used to automatically recognize related data and extract specific information by using inference (e.g., mapping between different versions or slight syntax variations). Another future work item is a more detailed investigation of W3C's Delivery Context Ontology (DCO) and whether it can be used as the basis for the integration model for both UED and UAProf. Finally, as the newest description format under development (i.e, DCO) is based on OWL it confirms our decision to use OWL as underlying technology for the integration model.

References

[1] IST-FP6: AXMEDIS (Automating Production of Cross Media Content for Multi-channel Distribution), http://www.axmedis.org/ (last accessed: December 2008).

[2] Butler, M. (2006). DELI: A Delivery Context Library For CC/PP and UAProf, http://delicon.sourceforge.net/ (last accessed: December 2008).

[3] Clark, J., ed. (1999). XSL Transformations (XSLT) Version 1.0, W3C Recommendation, November 1999. http://www.w3.org/TR/1999/ REC-xslt-19991116 (last accessed: December 2008).

[4] IST-FP6 (2008). DANAE (Dynamic and Distributed Adaptation of scalable multimedia coNtent in a context-Aware Environment), http://danae.rd.francetelecom.com/ (last accessed: December 2008).

[5] EBU Metadata Classification Scheme Mainenance Form (2009). http://www.ebu.ch/metadata/maintenanceforms/index_cs.html (last accessed 2009).

[6] IST-FP6 (2008). ENTHRONE (End-to-End QoS through Integrated Management of Content, Networks and Terminals), http://www.ist-enthrone.org/ (last accessed 2008).

[7] Freed, N. 91996). Borenstein, N; (eds.): Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types, RFC 2064, November 1996.

[8] Brickley, D., Miller, L. (2007). FOAF Vocabulary Specification 0.91, Namespace Document, November 2007. http://xmlns.com/foaf/spec/20071002.html (last accessed April 2009).

[9] Gimson, R., Lewis, R., Sathish, S. (eds.) (2006). Delivery Context Overview for Device Independence, W3C Working

Group Note, March 2006. http://www.w3.org/TR/di-dco/ (Last accessed: December 2008).

[10] Hunter, J. (2001). Adding Multimedia to the Semantic Web-Building an MPEG-7 Ontology, In: Proceedings of the International Semantic Web Working Symposium (SWWS), Stanford, August. p. 261-281.

[11] Hunter, J (2001). Lagoze, C.: Combining RDF and XML Schemas to Enhance Interoperability Between Metadata Application Profiles, In: Proceedings of the 10th World Wide Web Conference, Hong Kong, May.

[12] HP Laboratories (2008). ARQ -A SPARQL Processor for Jena, http://jena.sourceforge.net/ARQ/ (last accessed: December 2008).

[13] Kiss, C. (ed.) (2007). Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 2.0, W3C Working Draft, April 2007. http://www.w3.org/TR/CCPP-struct-vocab2/ (last accessed: December 2008).

[14] Klyne G. et.al. (eds.) (2004). Composite Capability/ Preference Profiles (CC/PP): Structure and Vocabularies 1.0, W3C Recommendation, January 2004. http://www.w3.org/TR/CCPPstruct-vocab/ (last accessed: December 2008).

[15] Lewis, R., Fonseca, J.M.C. (eds.) (2008). Delivery Context Ontology, W3C Working Draft, April 2008. http://www.w3.org/TR/dcontology/ (last accessed: December 2008).

[16] Manjunath, B.S., Salembier, P., Sikora, T. (eds.) (2002). Introduction to MPEG-7: Multimedia Content Description Interface, Wiley & Sons, April.

[17] McGuinness, D.L., van Harmelen, F.; (2004). (eds.): OWL Web Ontology Language Overview, W3C Recommendation, February 2004. http://www.w3.org/TR/owl-features/ (last accessed: December 2008).

[18] Open Mobile Alliance Ltd (2006). User Agent Profile, Approved Version 2.0. Technical report, OMA, February.

[19] Prud'hommeaux, E., Seaborne, A (2008). SPARQL Query Language for RDF, W3C Recommendation, January 2008. http://www.w3.org/TR/rdf-sparql-query/ (last accessed: December 2008).

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

[21] w3development.de (2008). UAProf profile repository. http://w3development.de/rdf/uaprof repository/ (last accessed: December 2008).

[22] Wireless Application Protocol (2002). WAP 2.0, Technical White Paper. Technical Report, WAP Forum Ltd.

[23] Wireless Markup Language (2001). Technical Report WAP-238-WML-20010911-a, WAP Forum Ltd., September.

[24] WAP Wireless Telephony Application (2001). Technical Report WAP-266-WTA-20010908-a, WAP Forum Ltd., September.

Authors bibliography

Christian Timmerer an assistant professor in the Department of Information Technology (ITEC), Multimedia Communication Group (MMC), Klagenfurt University, Austria. His research interests include the transport of multimedia content, multimedia adaptation in constrained and streaming environments, distributed multimedia adaptation, and Quality of Service / Quality of Experience. He was the general chair of WIAMIS 2008 and EUMOB2009 and has participated in several EC-funded projects, notably DANAE, ENTHRONE, P2P-Next, and ALICANTE. He also participated in ISO/ MPEG work for several years, notably in the area of MPEG-21, MPEG-M, and MPEG-V. He received his PhD in 2006 from the Klagenfurt University. Publications and MPEG contributions can be found under http://research.timmerer.com, follow him on http://www.twitter.com/timse7, and subscribe to his blog http://blog.timmerer.com.

Johannes Jabornig received his M.Sc. (Dipl.-Ing.) in January 2007 for his research on heterogeneous delivery context models and its comparison and mapping respectively. He is currently self-employed where he is working on CAD/CAM process automation.

Hermann Hellwagner received the M.S. and Ph.D. degrees in Informatics from the University of Linz, Austria, in 1983 and 1988, respectively. He has been a full professor of Informatics in the Institute of Information Technology (ITEC), Klagenfurt University, Austria, for more than ten years. His current research areas are distributed multimedia systems, multimedia communications, and quality of service. He has received many research grants from national (Austria, Germany) and European funding agencies as well as from industry. Dr. Hellwagner is the editor of several books and has published more than 100 scientific papers on parallel computer architecture and parallel programming and, more recently, on multimedia communications and adaptation. He has organised several international conferences and workshops. He is a member of the IEEE, ACM, GI (German Informatics Society) and OCG (Austrian Computer Society), and a member of the Scientific Board of the Austrian Science Fund (FWF).