APPLYING DOMAIN KNOWLEDGE TO MULTI-STEP MEDIA ADAPTATION BASED ON SEMANTIC WEB SERVICES

Peter Soetens and Matthias De Geyter

VRT
Brussels, Belgium
{peter.soetens,matthias.degeyter}@vrt.be

ABSTRACT

Publication of adapted media content on devices involves the description of three parties : the target platform's capabilities and preferences, the media content, and a survey of the available adaptation processes. Existing approaches define various standards to describe these issues but do not offer the flexibility and expressiveness to cope with new emerging devices and media formats. We provide this by defining a single domain ontology on media adaptation specified in OWL, that can describe these three parties using OWL-S and the CC/PP framework. In this document, we outline how such a domain ontology helps to develop an adaptation strategy. We present the implementation of an architecture based on Semantic Web (adaptation) Services. Service composition techniques allow us to automatically create a chain of adaptation services, which will alter the original document to the needs of the target device.

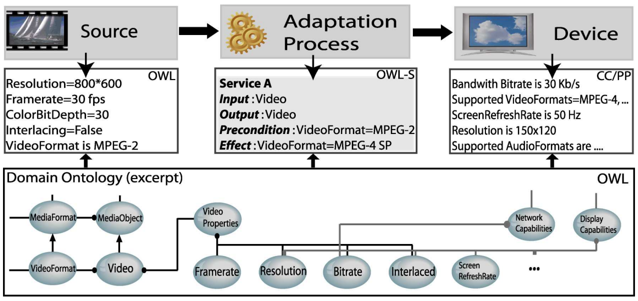
1. INTRODUCTION

A fast increase in the use and development of distributed multimedia applications and a growing popularity and diversity of devices to access the Internet demands a new approach towards media adaptation. This broad range of devices which includes mobile phones, personal digital assistants and set-top boxes, creates a wide spectrum of contexts (network availability, processing power, operating system, etc.). Traditional distribution mechanisms for multimedia are either completely unaware of these different specifications or in best case, may adapt along a limited set of adaptation axes for a limited set of devices. So what is stopping an evolution towards a publication engine that produces fine-tuned multimedia presentations for each specific context? Sufficient mechanisms to modify multimedia seem to exist, providing programs to scale and adapt presentations or transform them to a more suitable document model. The main problem when creating a publication engine however, is the co-ordination of these adaptation steps. This co-ordination is based on the description of the target platform and the media document. Efforts like the MPEG-21 framework [1] and CC/PP [2] try to propose new standards that improve the current use of a limited set of HTTP headers to describe the target platform. For the media document description, various description languages are being proposed like MPEG-7 or our own XiMPF [3]. To solve the adaptation problem however, there is more needed than descriptions of platforms and documents. First of all, there is a third party that requires description : the adaptation steps. A publication engine needs to be able to select the appropriate available adaptation steps based on the descriptions of the media document and the target platforms [4]. Secondly, when different description languages are used the co-ordination of the adaptation steps requires the mapping of the descriptive properties of each party onto each other. Another description format for adaptation steps would lead us to having three different description formats for the 3 parties, when in essence, all three parties use concepts from the media domain. This is why in [5] we proposed to use a single domain ontology on media to annotate all parties involved. A last problem is the decision-making itself, the adaptation strategy that selects the most appropriate adaptation steps and forms the adaptation chain. In this paper, we describe in detail how we can co-ordinate the adaptation process using the knowledge in the domain ontology. We describe how the domain knowledge is applied to describe the three parties involved using Semantic Web technologies like CC/PP, OWL and OWL-S. We demonstrate how a correct chain of Web Services can be composed, each modifying the source document in such a way that the result satisfies the constraints imposed by the target platform.

This work is a result of the Advanced Media project, a joint collaboration between the VRT, IMEC, Universiteit Gent and Vrije Universiteit Brussel.

2. SEMANTICS FOR MEDIA ADAPTATION

The transparent use of multimedia resources in heterogeneous and dynamically changing environments demands for an adaptation before or during distribution, taking into account the client capabilities (specific device characteristics), network bandwidth and user preferences. The description of such characteristics can be done nowadays by using an integrated framework like MPEG-21 or CC/PP. Within the MPEG-21 multimedia framework, DIA (Digital Item Adaptation) aims to provide a standard that describes the metadata needed when performing adaptation operations, such as device capabilities. Unfortunately, this vocabulary for describing adaptation related metadata is difficult to extend when a new context needs to be described and also provides no formal definition of the semantics of MPEG-21 terms. This lack of semantic interoperability, combined with unclear requirements for tools on the DIA part of MPEG-21 makes it difficult to produce a generic adaptation engine. In [6] a multi-step media adaptation engine based on MPEG-21 and a planning methodology is presented. The lack of shared semantics between MPEG-21 and MPEG-7, as described in [7], was creatively bridged by the authors by combining both MPEG-7 and MPEG-21 terms in adaptation operation descriptions. The absence of a rich semantical model however, cannot bridge the gap between the descriptions of the adaptation




```
<owl:ObjectProperty rdf:ID="hasFramerate">
  <rdfs:domain rdf:resource="#VideoProperties" />
  <rdfs:range rdf:resource="#FrameRate" />
  <rdf:type rdf:resource="
  http://www.w3.org/2002/07/owl#FunctionalProperty" />
  <rdfs:subPropertyOf rdf:resource="#hasRefreshRate" />
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="isInterlaced">
 <rdfs:domain>
  <owl:Class>
   <owl:unionOf rdf:parseType="Collection">
     <owl:Class rdf:about="#VideoProperties" />
     <owl:Class rdf:about="#DisplayCapabilities" />
   </owl:unionOf>
  </owl:Class>
 </rdfs:domain>
 <rdfs:range rdf:resource="#Interlaced" />
 <rdf:type rdf:resource="
 http://www.w3.org/2002/07/owl#FunctionalProperty" />
</owl:ObjectProperty>
```

Fig. 1. A domain ontology captures the knowledge necessary to describe documents, adaptation services and devices. A small excerpt shows the OWL description of the domain ontology.

operations and their impact on the context of the client. Furthermore, a richer semantical model also extends reasoning possibilities, preventing problems with planning heuristics as described in [6]. CC/PP [2], created by the W3C, tries to steer clear of these obstacles by using RDF [8] for the specification of platform descriptions and user capabilities. CC/PP itself only defines a framework for such descriptions but it supports the use of vocabularies like UAProf [9].

3. SEMANTICS AND SERVICES

The Semantic Web was an idea thought up by Berners-Lee and evolves about making the contents of the web 'understandable' for machines so that they can work intelligently with this information. For this, information has to be structured and semantically annotated. The RDF Framework [8] was conceived to fulfill this need and it allows for the description of web resources, identified by URI's, by writing triples to make statements. Thus, the means for semantic annotation and interoperability on the web were established. However, for the processing and sharing of knowledge on the web by programs, an additional layer was needed. This layer is now embodied by OWL[10], the Web Ontology Language, which builds on top of RDF Schema. OWL allows for richer expressions, and can be used to represent ontologies, which are generally defined as a "representation of a shared conceptualization of a particular domain" [11].
In parallel to this evolution towards the Semantic Web, the Web is also evolving towards being a provider of services: web sites that do not merely provide static information but allow one to effect some action or change in the world such as flight booking services. In order to employ their full potential, appropriate descriptions for Web services need to be developed. Current technologies such as UDDI, WSDL and SOAP provide limited support for important tasks like service discovery, composition and monitoring [12]. To enable a reliable, large-scale interoperation of web services, these services have to be computer interpretable. This is were the Semantic Web and Web Services come together : The Semantic Web promises services whose properties, capabilities, interfaces and effects are encoded in an unambiguous, machine understandable form, based on the RDF standard and more specifically on OWL. OWL-S is an ontology that builds on top of OWL that can describe web services using a Service profile and model that specifies the transformation produced by the service in terms of *inputs, outputs, preconditions and effects* (IOPEs) . The preconditions and effects denote external conditions required by the service and effects resulting from its execution, respectively. Composition of several web services to achieve one final goal can be performed by matching their respective IOPEs. Moreover, Semantic Web Services can be composed dynamically based on domain knowledge by using more complex logics such as rules that can be defined about the domain. In most of the relevant research on this topic [13] [14] [15], a technique is used which couples the description of a service with a description of a planning or task-methodology. This allows for standard planning or state-space algorithms to be used on the problem of chaining services.

4. OUR APPROACH TO MEDIA ADAPTATION

The basis for our approach [5] is an ontology which describes the concepts and the relations of the media adaptation domain. This shared vocabulary allows us to describe the three entities involved : the target platforms, the adaptation process and the source document. It also enables us to reason about this domain and write down rules and constraints for adaptation. We constructed this media ontology using OWL and we modelled it using elements from both MPEG-21 and UAProf. Fig. 1 shows a graphical representation of a small excerpt and how concepts from this ontology are used to express device, service and document specifications. The CC/PP framework, for client capability descriptions, is integrated in the ontology and describes the client's hardware, software and browser concepts. Hardware concepts include network details, display capabilities -such as resolution and framerate- and benchmarking details about the device. Software and browser concepts describe which document types the client can process and prefers. This framework is instantiated when a CC/PP profile needs to be built that describes client capabilities and preferences. Similar concepts from the domain ontology are also used to describe source documents, specifying the type of the media object (video, audio,. . .) and its specific properties (media format, bitrate, resolution,. . .). These OWL descriptions can be integrated with formats like MPEG-7 or XiMPF. Finally, the adaptation process is annotated through the use of Semantic Web Services that can adapt media items. These Semantic Web Services refer to concepts from the media domain ontology (see fig. 2) in the services' input and output parameters, preconditions and effects for the declaration of their OWL-S service profiles.
For our work on a news distribution system for Belgian broadcaster VRT, we have modelled three platforms (PDA, PC and TV

```
<process:ProcessModel
rdf:ID="DecodeProcessModel">
  <service:describes rdf:resource="#DecodeService"/>
  <process:hasProcess rdf:resource="#DecodeProcess"/>
</process:ProcessModel> <process:AtomicProcess
rdf:ID="DecodeProcess">
 <process:hasInput rdf:resource="#InputVideo"/>
 <process:hasOutput rdf:resource="#OutputVideo"/>
 <process:hasPrecondition>
  <xmt:XMTExpression rdf:ID="InputFormatExpression">
   <xmt:hasExpressionFormula>
    <xmt:Formula rdf:ID="InputFormatFormula">
     <xmt:hasSubject rdf:resource="#InputVideo"/>
     <xmt:hasProperty rdf:resource="
     http://lt.xmt.be/wp2/domainontology.owl#hasVideoFormat"/>
     <xmt:hasDescription rdf:datatype="
     http://www.w3.org/2001/XMLSchema#string">
      http://lt.xmt.be/wp2/domainontology.owl#MPEG-2
     </xmt:hasDescription>
    </xmt:Formula>
   </xmt:hasExpressionFormula>
  </xmt:XMTExpression>
 </process:hasPrecondition>
 <process:hasResult>
  <process:Result rdf:ID="DecodeResults">
   <process:hasEffect>
    <xmt:XMTExpression rdf:ID="OutputFormatExpression">
     <xmt:hasExpressionFormula>
      <xmt:Formula rdf:ID="OutputFormatFormula">
       <xmt:hasSubject rdf:resource="#OutputVideo" />
       <xmt:hasProperty rdf:resource="
       http://lt.xmt.be/wp2/domainontology.owl#hasVideoFormat"/>
       <xmt:hasDescription rdf:datatype="
         http://www.w3.org/2001/XMLSchema#string">
         http://lt.xmt.be/wp2/domainontology.owl#YUV
       </xmt:hasDescription>
      </xmt:Formula>
     </xmt:hasExpressionFormula>
    </xmt:XMTExpression>
   </process:hasEffect>
  </process:Result>
 </process:hasResult>
</process:AtomicProcess>
<process:Input rdf:ID="InputVideo">
 <process:parameterType
rdf:resource="http://lt.xmt.be/wp2/domainontology.owl#Video"/>
 <rdfs:label>Input Video</rdfs:label>
</process:Input>
<process:Output rdf:ID="OutputVideo">
 <process:parameterType rdf:resource="
    http://lt.xmt.be/wp2/domainontology.owl#Video"/>
 <rdfs:label>Output Video</rdfs:label>
</process:Output>
```

Fig. 2. An extract of the OWL-S ProcessModel description for an MPEG-2 Video decoding-service. Other descriptions of effects of this service (like an increase in bitrate) were left out for clarity.

with set-top box), source documents containing video and adaptation services for video material (scale, decode, encode, deinterlace, convert to audio,. . .). We built a system that automatically adapts video for the three different platforms, taking into account their specified context.

5. SERVICE COMPOSITION FOR MEDIA ADAPTATION

Gathering the necessary descriptions and then composing a chain based on the differences between the client capabilities and the source document proofs to be very complex. Our first approach was based on signature matching as described in the matchmaking algorithm of [16]. However, searching the available web service space each time to find a set of successively matching services is a problem that, with an increasing number of services, displays exponential growth. To cope with this problem, we introduced a service registry that categorizes the services according to their signature and the effects they have on an input document. Each service belongs to one or more categories in the tree. A service category describes in essence a general operation on media which is specified, like semantic web services, in terms of input and output-concepts, preconditions and effects. The description of input and output is quite straightforward, specifying the type of media and other necessary input data types (e.g. desired resolution outcome) using the domain ontology. For the annotation of preconditions and effects however, we need to establish a formalism that expresses requirements to use this operation and the possible changes that can occur in both the state of the target platform and the source document. We use a formalism similar to SWRL [17] to specify these preconditions and effects which can either define a change of value for an OWL ObjectProperty or a DataTypeProperty. The description of these categories based on our domain ontology has 2 benefits : it describes the impact of media operations (bridging the gap between document properties and platform properties) and in a hierarchical tree, it allows us to build a service registry that reduces the complexity of the service composition problem.
Our formalism for expressing preconditions and effects is of course also used for actual available services registered for a specific category. In contrast to categories, which will usually only specify which properties are influenced in their effects, services will express specific changes in their effects statements. Fig. 2 shows the ProcessModel description of a decoding-service that converts the VideoFormat of a file from MPEG-2 to YUV (note that this is an effect of the service, input and output are both video-objects). The full description of this service also specifies the effect on the bitrate of the video when converting from MPEG-2 to YUV.

Fig. 3 gives a schematic overview of our approach. We first match the document format and it's descriptive properties against the description of the target platform. There are several possible strategies when performing this match, where some might be based on relations describing the impact of document properties on the device in the domain ontology. We have implemented a relatively simple strategy that checks if there is either a possible equivalence or subsumption relation between classes describing the source document and the target platform. The source document properties that have a match, are then used as a basic set of adaptation axes for the adaptation strategy.
Based on category descriptions and a set of adaptation axes (properties) our simple adaptation strategy will try to find a category of services that adapts the source document along one of the adaptation axes (effects of this category). Detailed descriptions of the services' effects help us to estimate the value for this property after execution and select the appropriate service in this category. Execution order is controlled by signature matching and constraints matching (comparing the services' IOPE-descriptions) but can (in future work) be influenced and refined by defining rules on the domain. Service Categories allow for the definition of rules independent of available or implemented service descriptions. This

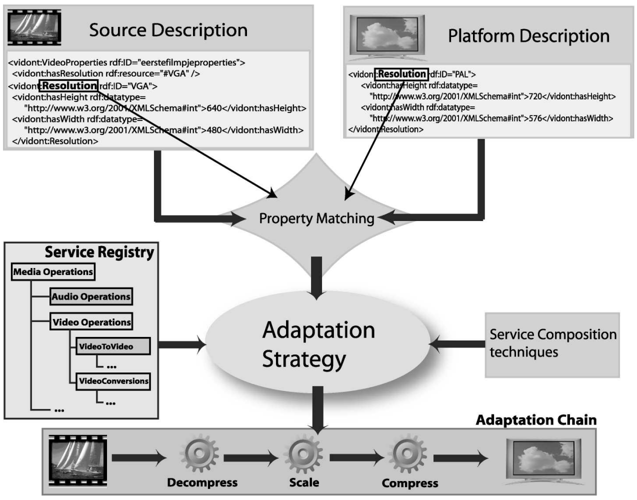


Fig. 3. An overall view of the architecture to support adaptation strategies.

strategy is still relatively simple, using no planning methodology but only the available information on adaptation axes and available services but it already is far more flexible than many of the existing approaches. Our adaptation strategy may even decide to convert video content to audio or a set of keyframes if bandwidth constraints are too tight. As far as complexity for the source document goes, there are little boundaries, provided that future algorithms can rely on extensive domain knowledge specified in the ontology.

6. CONCLUSION

Efficient media adaptation strategies require flexible and semantically rich descriptions of source documents, available adaptation steps and target platforms. This can be accomplished when a shared vocabulary exists through the specification of a domain ontology. We demonstrate in this paper how such an ontology can be constructed and used to support the adaptation process. Domain knowledge enables us to organize the available adaptation services according to their semantical descriptions and thus select more effectively the necessary adaptation services for the composition of an adaptation chain. The knowledge captured by the domain ontology also assists in the matching process between the descriptions of our target platform and the source document to establish the adaptation axes on the source document. These steps were taken to support the implementation of a simple adaptation strategy and clearly demonstrate the need for a rich shared vocabulary in a future flexible adaptation engine.

7. REFERENCES

[1] ISO/IEC, *Information technology – Multimedia framework (MPEG-21) – ISO/IEC 21000*, 2003.

[2] W3C, *Composite Capability/Preference Profile – W3C Recommendation*, Jan. 2004, http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115/.

[3] Steven Van Assche, F. Hendrickx, N. Oorts, and L. Nachtergaele, “Multi-channel publishing of interactive multimedia presentations,” *Elsevier Computer and Graphics*, pp. 193–206, Apr. 2004.

[4] S. Decneut, F. Hendrickx, S. Van Assche, and L. Nachtergaele, “Targeting heterogeneous multimedia environments with web services,” in *Proceedings of the IEEE International Conference on Web Services (ICWS’04)*, 2004.

[5] P. Soetens, M. De Geyter, and S. Decneut, “Multi-step media adaptation with semantic web services,” in *International Semantic Web Conference (ISWC 2004)*, 2004.

[6] D. Jannach, K. Leopold, H. Hellwagner, and C. Timmerer, “A knowledge based approach for multi-step media adaptation,” in *Proceedings of 5th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, 2004.

[7] J. Hunter, “Reconciling MPEG-7 and MPEG-21 semantics through a common event-aware metadata model,” in *CoRR cs.MM/0210021*, 2002.

[8] W3C, *RDF Vocabulary Description Language 1.0: RDF Schema – W3C Recommendation*, Feb. 2004, http://www.w3.org/TR/2004/REC-rdf-schema-20040210/.

[9] WAP, Wireless Application Protocol Forum, *User Agent Profile Specification (UAProf)*, Nov. 1999, http://www.wapforum.org/what/technical/SPEC-UAProf-19991110.pdf/.

[10] W3C, *OWL Web Ontology Language Overview – W3C Recommendation*, Feb. 2004, http://www.w3.org/TR/owl-features/.

[11] T. Gruber, “A translation approach to portable ontology specifications,” *Knowledge Acquisition*, vol. 5, no. 2, pp. 199–220, 1993.

[12] S.P. McIlraith, T. Cao Son, and H. Zeng, “Semantic web services,” *IEEE Intelligent Systems*, vol. 16, no. 2, pp. 46–53, March/April 2001.

[13] J. Kim and Y. Gil, “Towards interactive composition of semantic web services,” in *Semantic Web Services - 2004 AAAI Spring Symposium*, 2004.

[14] A. Gómez-Pérez and R. González-Cabero, “A framework for design and composition of semantic web services,” in *Semantic Web Services - 2004 AAAI Spring Symposium*, 2004.

[15] E. Sirin, B. Parsia, and J. Hendler, “Composition-driven filtering and selection of semantic web services,” in *AAAI Spring Symposium on Semantic Web Services*, 2004.

[16] M. Paolucci, T. Kawamura, T.R. Payne, and Katia Sycara, “Semantic matching of web services capabilities,” in *Proceedings of the First International Semantic Web Conference*, 2002.

[17] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, *SWRL : A Semantic Web Rule Language Combining OWL and RuleML*, http://www.daml.org/2003/11/swrl/.