Knowledge-based Multimedia Adaptation for Ubiquitous Multimedia Consumption

Dietmar Jannach, Klaus Leopold

Abstract

Intelligent, *server-side adaptation of multimedia resources* is becoming increasingly important and challenging for two reasons. First, the market continuously brings up new mobile end-user devices to which the content has to be adapted as these devices support different display formats and operate on various types of networks. On the other hand, with the help of metadata annotations which are now available in the MPEG-7 and MPEG-21 standard, advanced forms of resource adaptations *on the content level* become possible.

As none of the existing multimedia transformation tools and libraries can support all these different forms of basic and advanced adaptation operations, an intelligent multimedia adaptation server has to integrate such external tools and algorithms and perform an adequate sequence of adaptation operations on the original resource before sending it to the client.

In this paper we present the results of the *ISO/IEC MPEG Core Experiment* on using Semantic Web Services technology as a tool for declaratively describing the semantics of adaptation services and constructing multi-step adaptation sequences in an open and extensible multimedia adaptation framework. We show how the semantics of adaptation operations can be captured in the form of input, output, precondition, and effects, how the problem of finding adequate adaptation sequences can be viewed as an Artificial Intelligence planning problem, and finally, how the existing MPEG standards are technically integrated into the service descriptions and how they serve as the shared ontology of the domain.

Our approach both introduces declarative, knowledge-based technology into the involved multimedia communities and on the other hand broadens the application scope of Semantic Web Service technology in the area of general semantic service descriptions and automated program construction.

1 Introduction

Today, a lot of different end user devices are connected to the Internet. One can find desktop PCs, notebooks, workstations, set-top boxes, TV sets, but

Preprint submitted to Elsevier Science

also mobile devices such as PDAs, cell phones, and hand held devices using Internet services. All those devices feature different capabilities in terms of computational power, memory size, display size, or network capabilities. Displaying regular web content (HTML) on these devices is a task of manageable complexity. However, rendering multimedia content in such an environment remains challenging because the content available on the multimedia server can be heterogeneous, e.g., in terms of encoding: A video, for instance, can be encoded in different formats such as MPEG-1, -2, -4, H264, or WMV, using different encoder settings such as spatial and temporal resolution, color depth, or bit rate. Due to this heterogeneity, today's end user devices are in general not able to display all kind of multimedia data. This however, does not conform to the long-term vision of Universal Multimedia Access (UMA) expressed in [1], where any user (respectively device) can consume any multimedia content, anytime and anywhere.

MPEG-21 is the ISO/IEC standard that aims at addressing these new challenges by defining a normative open framework for multimedia delivery and consumption involving all parties in the delivery and consumption chain. A major part of these standardization efforts deals with the definition of an interoperable framework for Digital Item Adaptation (DIA) [2]. Server-side resource *adaptation* is one of the key means for solving some of the problems caused by the above-mentioned heterogeneity: In order to maximize the user experience of the end consumer, the given multimedia resources are transformed adequately such that they fit the consumer's preferences and usage environment before they are sent to the client.

Currently, the MPEG-21 Digital Item Adaptation framework comprises basic mechanisms for describing the adaptation problem in a vendor-independent, interoperable way and defines normative description schemes for capturing terminal capabilities, user preferences, or the format of the given resource itself. The implementation of an intelligent adaptation engine capable of determining the required transformations is intentionally left to the tool vendors (see Figure 1).

This means, however, that tool vendors have to implement their own *Adaptation Engines* that interpret the given consumer requirements and try to figure out somehow what kind of adaptations are required to satisfy these requirements. From a practical perspective, the main challenge when designing such a component is that it has to be extremely flexible with respect to extensibility: Whenever a new adaptation operation has to be integrated in the server, e.g., because a new encoding format should be supported, the logic of the Adaptation Engine has to be adapted accordingly, i.e., it has to be encoded under which circumstances the newly available transformation operation should be applied.

Furthermore, we argue that extensibility with respect to *external transformation tools* will become a key issue, in particular if we look at new standards like MPEG-7 [3] which allow us to enrich media content with semantic content annotations, which in turn facilitates new forms of multimedia experience, like search on specific topics or semantic-based content selection and filtering. Consequently, an extensible adaptation server will have to provide adequate extensibility features for the integration of such specialized third-party tools.

In this paper we describe an extension to the existing *MPEG-21 Digital Item Adaptation* standard which is based on declarative descriptions of tool capabilities, standardized MPEG metadata, as well as Semantic Web Service technology [4] for capturing the semantics of adaptation operations. Based on these extensions, open and extensible adaptation servers can be built that integrate arbitrary third-party tools in a standardized way and in addition are capable of intelligently constructing adequate adaptation sequences to meet the adaptation goals. We propose a knowledge-based approach to construct these adaptation sequences in order to guarantee that no changes in the general mechanism are required when new forms of adaptation are possible as the standards evolve or new tools become available.

The paper is organized as follows. First, we give an overview of the *multi-step digital item adaptation* problem and of the involved MPEG-standards. Next, we show how the problem of constructing adequate adaptation sequences can be transformed to a state-space planning problem which can in turn be solved with standard Artificial Intelligence (AI) planning algorithms. In the following sections we then discuss the details of our proposal of exploiting Semantic Web Services technology as representation mechanism for expressing tool capabilities in a declarative way. We summarize the results of two official *ISO/IEC MPEG core experiments* that were performed in order to show how the proposed approach can be integrated into the existing MPEG-21 standard, how interoperability with other standards (MPEG-7) can be achieved, and how standard Web Service technologies (like the Web Service Description Language WSDL [1]) can be used to dynamically invoke the external tools in a standardized way.

This paper significantly extends the approach presented in [5] and [6], in particular with the detailed results of the ISO/IEC MPEG Core Experiments.

[1] World Wide Web Consortium, Web Service Description Language 1.1, see http://wwww.w3.org.

2 Knowledge-based Multi-step Digital Item Adaptation.

An intelligent adaptation server has to be provided with at least two different pieces of information in order to be able to adapt a multimedia resource adequately upon a client request.

- First, the *user's constraints* (e.g., the terminal capabilities, network bandwidth, or even content preferences) have to be known.
- In addition, there has to be information about the requested resource, for instance about the actual encoding format, in order to determine which sort of adaptations are required or possible.

These two aspects are addressed by the *ISO MPEG* consortium in their recent standards. On the one hand, multimedia resources can be annotated with the help of MPEG-7 metadata documents. The documents for instance contain information about the format but can also contain semantic information about the contents, e.g., a classification of the different scenes of a movie. On the other hand, Part 7 of the new MPEG-21 standard - *Digital Item Adaptation (DIA)* - is providing tools for expressing additional information required for flexible resource adaptation. In particular, it defines normative description schemes [2] for specifying the client's preferences or the *"usage environment"*.

However, at the moment, MPEG-21 DIA does not specify how a multimedia adaptation server actually determines and performs the required transformations and leaves this task up to the tool vendors. As such, there is only limited support for interoperability in situations, where multiple (subsequent) transformations have to be performed on the original resource in order to meet the client's needs. We see this limitation to be too restricting, because specialized adaptation tools will be available on the market when the standards are more and more established in industrial environments and none of these tools will be able to provide all different sorts of required adaptations. Figure 1 sketches the relation of terms in MPEG-21 Digital Item Adaptation and the current scope of standardization [7].

We therefore propose to extend MPEG-21 DIA with a mechanism that provides such interoperability between third-party tools and identified the following major requirements for such a mechanism:

- First, we need a representation mechanism for describing arbitrary tool capabilities (i.e., the semantics of performing an adaptation operation) in a precise form, such that automatic construction of adaptation chains becomes possible.

[2] These specifications are given in the form of XML-Schema documents.

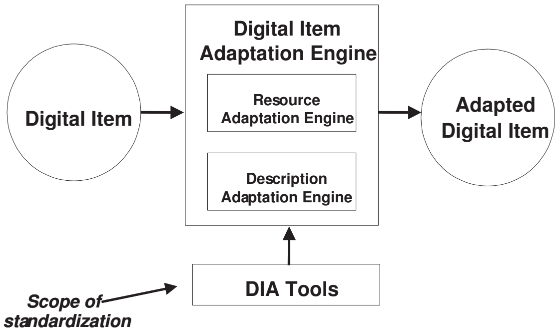

Fig. 1. Current scope of standardization in MPEG-21 DIA.

- Second, we need to find an approach that also allows us to actually invoke the tools and algorithms in a standardized form once the adaptation sequence is determined.
- Finally, the mechanism has to be fully integrated into the MPEG-21 framework for open multimedia delivery and consumption.

2.1 "Planning" adaptation sequences.

In order to address the above-mentioned requirements, we propose adopting a declarative, knowledge-based approach. Knowledge-based systems are nowadays already well-established in different domains because of their specific advantages: When using such systems, the needed domain knowledge (expert knowledge) can be expressed in a declarative form in contrast to a procedural representation, where factual knowledge is in many cases intertwined with the processing logic. Moreover, the semantics of the used languages for expressing the domain knowledge is formally defined in a non-ambiguous and well-understood manner, e.g., in form of some variant of First Order Logic. Finally, the clear separation of domain-knowledge and inference knowledge allows us to build easy-to-maintain knowledge bases and to use highly-optimized and domain-independent reasoning engines that flexibly process the contents of the underlying knowledge base.

In our particular problem setting, we view the problem of constructing adequate adaptation sequences as a classical *"state-space planning problem"* (see e.g., [8]). Such a planning problem is defined in a general, domain independent manner: The inputs to the algorithm are on the one hand declarative descriptions of the actual situation (the start state) and the desired situation (the goal state). In AI-planning approaches, these basic descriptions are defined in the form of simple logical facts. Furthermore, a planning problem also comprises a set of *world-altering actions* which are provided to the system. Each action is annotated with a set of a) preconditions that have to hold when the action should be applied, b) a list of effects when applying the action, and c) the list of inputs and outputs (parameters) of the action when it is embedded in an action sequence. Again, this knowledge can be expressed in a modular way using a standard logic language. Given this knowledge, the planning problem consists of finding a valid sequence of parameterized actions that bring us from the start state to the goal state.

The main advantage of such a general problem-solving approach is that the needed reasoning engine can be implemented in a fully domain-independent manner, i.e., the engine operates solely on arbitrary symbols and does not have to have knowledge about the domain-specific semantic meaning of the actions. In particular in the last decade, significant advances have been made in the corresponding research fields (e.g. starting with [9]), where highly-sophisticated planning algorithms have been developed, such that complex, real-world planning problems can now be efficiently solved.

If we map the problem of finding adaptation sequences in a multimedia sever to such a planning problem, the following correspondences can be found:

- the (description of the) original multimedia resource can be seen as the start state, e.g., the given resolution of the resource.
- the goal state corresponds to the client's user constraints, e.g., the maximum resolution the user's device supports.
- the set of transformations the multimedia server can perform finally corresponds to the actions that alter the state of the resource.

Let us consider the following simplified example for such an adaptation problem where an existing color image is given in some resolution and encoding and should be sent to the client in a different resolution and without color.

The start state, i.e., the description of the resource, can be captured with the help of the following facts, which can be contained in MPEG-7 metadata annotations [3].

$bmpImage(http://path/to/image.bmp), width(640), height(480),$
$color(true).$

The client's constraints and preferences are described in an MPEG-21 *Usage Environment Description*.

$bmpImage(file://path/to/image.bmp), horizontal(320), vertical(240)$
$color(false).$

The following listing shows the description of the spatial scaling operation

[3] Note that we use a simplified logical notation instead of the XML-representation for better readability.

for images where we use the above-mentioned separation of inputs, outputs, preconditions, and effects (IOPE)[4].

$Operation : spatialScale$
$Input : imageIn, oldWidth, oldHeight, newWidth, newHeight$
$Output : imageOut$
$Preconditions : bmpImage(imageIn), width(oldWidth), height(oldHeight)$
$Effects : bmpImage(imageOut), width(newWidth), height(newHeight),$
$horizontal(newWidth), vertical(newHeight)$

The adaptation framework features standard read and write operations used to read images from input sources and write images to the output. These operations could also be more complex; they could for instance be used to read individual frames from a video sequence. The computed adaptation plan of the adaptation decision taking engine could be as follows:

(1) $read(file : //path/to/image.bmp, outImage1)$
(2) $spatialScale(outImage1, 640, 480, 320, 240, outImage2)$
(3) $greyscale(outImage2, outImage3)$
(4) $write(outImage3, file : //path/to/output/image.bmp)$

Note that automatically generated variables *outImage1, outImage2, ...* in the plan also describe how inputs and outputs of subsequent transformation steps are to be treated by the adaptation engine. In the example, having *outImage1* as an output of the *read* operation and the same variable as input to the *spatialScale* operation means that the actual variable's contents have to be forwarded to the next adaptation step.

The main advantages of the described knowledge-based method are quite obvious. First, the approach features a comprehensible mechanism to describe atomic actions which in turn also results in modular and easy-to-maintain knowledge bases. Furthermore, adaptation servers that are developed in that way are robust against changes, i.e., the reasoning algorithm does not have to be changed once there are new adaptation tools available or when the standards evolve and new symbols are used to describe the resource or the usage environment descriptions.

Regarding the expressiveness of a IOPE-style approach of describing tool capabilities, please note that for our purposes we do not need to describe the full functionality of external applications or tools in terms of inputs and outputs but can limit ourselves to the transformation functions. If we look at such functions at a certain abstraction level, all of them share a common structure, i.e., they accept input files of certain formats together with some

[4] We omit the similar grey-scaling specification for sake of brevity.

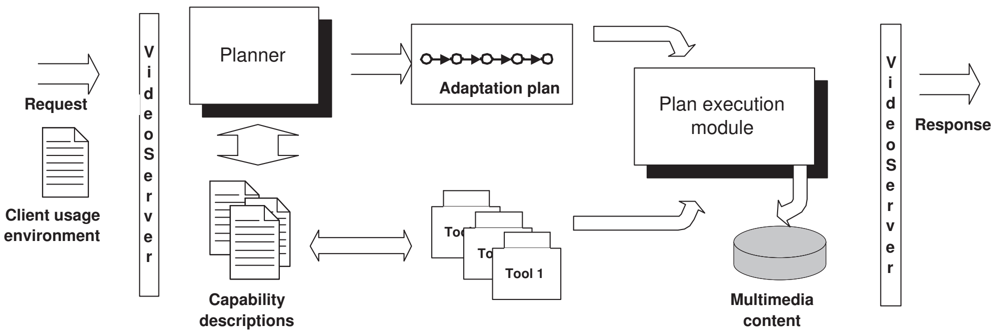

Fig. 2. Overview of planning approach.

(optional) parameters and produce output files (or streams) containing the modified content. Exactly these descriptions of content and format of the input and output files are standardized in the MPEG-standards, both with respect to the declarative, XML-based representation format as well as with respect to the vocabulary to be used. For non-standard cases, the description schemes in the MPEG framework also comprise mechanisms for incorporating application-specific extensions and for using non-standardized terms and annotations. Such non-standardized extensions are also supported in our approach, given the nature of declarative knowledge representation and the fact that the planning algorithm works on general symbols, which can also carry domain-specific information.

The final advantage of the knowledge-based approach lies in the fact that we can rely on highly-optimized, sound and complete planning algorithms that guarantee that a solution will be found if one exists and that we can optimize the produced plans according to different criteria. Figure 2 illustrates the planning process and the involved documents and tools [6].

Still, there are some issues to be addressed in the context of the existing standards such that the potentials of the knowledge-based approach can be fully exploited. As a first step, we need to find a standardized language for describing the adaptation problem syntactically, i.e., how the start and goal states and the actions are textually described. Although PDDL (Planning Domain Description Language) [10] is well-established in the AI planning community as de-facto standard [5], this language is not widely known in other environments and at the moment is not integrated with state-of-the-art XML-based technologies that nowadays are used in Web-based environments.

In addition, in order to guarantee interoperability between different third-party tools, there has to be a common *ontology* with respect to the terms that can be used in the action description. This means for instance that every involved party uses the same term/predicate to describe the *width* of an image

[5] The language is for instance used to describing benchmark problems for comparing the performance of algorithms in planning competitions, see [11].

and also associates the same semantic meaning with it. We will address these two aspects in the following section where we propose the usage of *Semantic Web* technology for our problem. After that we will then show how the remaining technical challenge of a common low-level technical interface for tool invocation can be dealt with and discuss architectural details of a reference implementation afterwards.

2.2 Semantic tool descriptions and the shared ontology.

The World Wide Web for a long time has been solely viewed as a sort of large repository for text, images, or other resources. One of the visions of the *Semantic Web* [6] is that such resources can be searched intelligently by content and not only by keywords and that these resources can also be accessed and processed automatically by software agents.

The first major milestone in this vision was reached by the development of OWL (Web Ontology Language)[13] which has now the status of a W3C (World Wide Web Consortium) recommendation. OWL itself can be viewed as an *upper ontology*, i.e., as a language based on XML and RDF-S [7] that can be used for creating domain-specific ontologies with precise semantics such that further inferences on the concepts can be made with the help of general ontology reasoning tools.

The next important step towards bringing more intelligence to the Web can be seen in the development of *Semantic Web Services* (see e.g., [4]). The main idea of that initiative is to semantically annotate the services that are available on the Web with the goal that they can be automatically searched and accessed by agents. In addition, software agents shall also be enabled to execute more complex tasks by combining the different services in an intelligent manner. One example for such an application from the domain of "travel planning" can be found in [4], where an intelligent agent makes a complex travel arrangement on behalf of the user by exploiting several web services like online hotel reservations or flight bookings.

The additional requirements for this type of service consumption over the Web are currently addressed in the development of OWL-S (OWL-Services)[14], an extension of OWL. Similar to OWL, OWL-S provides an XML-based representation mechanism for describing Web services on a semantic level. OWL-S also exploits the traditional AI-planning approach and relies on using inputs, outputs, preconditions, and effects as basic means for describing and exposing the semantics of a service for a client. As a language for describing especially

[6] see, e.g. [12] for an overview.
[7] Resource Description Framework, http://www.w3.org/TR/rdf-schema/

the preconditions (constraints) for service execution, OWL-S also comprises the general purpose, well-defined rule language SWRL (Semantic Web Rule Language) [15] [8]. Note that OWL-S only defines the language for specifying the constraints but does not impose restrictions on the inference engine that is to be used for evaluating the constraints or do further reasoning with respect to automatic service composition.

In general, automatic detection of suitable services and composition of service chains can only be done on the basis of a shared ontology, e.g., the involved parties have to have a common understanding what ”hotel” or ”reservation” means. Therefore, it is only reasonable that OWL-S is fully integrated in OWL with which such shared ontologies can be built. In the travel planning example, for instance, one would first build an ontology of the domain and then refer to the defined and agreed terms (like ”reservation”) from within the service descriptions.

```
<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="VideoType">
      <Video>
        <MediaInformation id="news1_media">
          <!-- MediaIdentification ... -->
          <MediaProfile>
            <MediaFormat>
              <!-- Content, Medium, FileFormat, Filesize ... -
              <VisualCoding>
                <Format href="urn:mpeg:mpeg7:cs:VisualCodingFormatCS:2001:3.3.1"
                                colorDomain="color">
                  <Name xml:lang="en">
                       MPEG-4 Visual Advanced Simple Profile @ Level 0</Name>
                </Format>
                <Frame width="640" height="480"/>
              </VisualCoding>
            </MediaFormat>
          </MediaProfile>
        </MediaInformation>
      </Video>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

Color attribute, encoding format

Given resolution

Fig. 3. Fragment of an MPEG-7 video description.

Today, a major obstacle to the successful implementation of Semantic Web Services in many domains lies exactly in the problem of establishing such a shared ontology. Even for well-understood application domains like in e-Commerce environments, many competing and partially incompatible pseudo-standards can be found in the market. Although there is a good chance that there will be a common understanding of terms like ”purchase order” or ”ship-to-address” (in an English-speaking world), it seems to be impossible to have a

[8] SWRL currently has the status of a W3C ”Member submission”, see http://www.w3.org/

common community- or even world-wide understanding of products or services to be purchased, their classification, and their relations.

However, in our application domain of multimedia adaptation services, such a shared ontology is implicitly established by the existing MPEG standards. Although not explicitly mentioned, in particular the normative description schemes of MPEG-7 (and MPEG-21) exactly define the allowed symbols, terms, and relations to describe a multimedia resource. The semantics of the terms are described within the standards in textual form but are agreed among the community. Figure 3 shows a fragment of such an MPEG-7 media description, Figure 4 a part of an MPEG-21 Usage environment description. A detailed example of how we can integrate these descriptions into our semantic adaptation service descriptions can be found in the next section. In fact it would be possible to re-write the parts of the standards in terms of an OWL ontology, see e.g. [16]. Nonetheless, it is not reasonable in our approach to expect the resource or usage environment descriptions to be given in the form of OWL ontologies. Rather, we aim at smoothly integrating the existing XML-based descriptions with the semantic service descriptions required for multi-step adaptation problems.

```
<DIA>
  <Description xsi:type="UsageEnvironmentType">
    <UsageEnvironmentProperty xsi:type="TerminalsType">
      <Terminal>
        <TerminalCapability xsi:type="DisplaysType">
          <Display>
            <DisplayCapability
                    xsi:type="DisplayCapabilityType"
                    activeDisplay="true" colorCapable="false">
              <Mode>
                <Resolution horizontal="320" vertical="240"
                    activeResolution="true"/>
                <Resolution horizontal="160" vertical="120"
                    activeResolution="false"/>
                <Resolution horizontal="80" vertical="60"
                    activeResolution="false"/>
              </Mode>
            </DisplayCapability>
          </Display>
        </TerminalCapability>
      </Terminal>
    </UsageEnvironmentProperty>
  </Description>
</DIA>
```

Specification of current resolution

Fig. 4. MPEG-21 DIA description of a terminal with certain display capabilities.

Major parts of OWL-S are also concerned with providing a standardized infrastructure for service registration, detection, and actual service invocation. OWL-S therefore provides XML-based mechanisms of how service providers can publish their services and expose functionality to their potential clients. A central part of these mechanism can be seen in what is called "*grounding*". Grounding basically deals with the actual invocation of Semantic Web Services on the technical level, where state-of-the-art technologies for platform-independent service invocation like UDDI (Universal Description, Discovery and Integration) and WSDL are used. This aspect is again of particular importance in our application domain as a platform- and tool-independent method invocation mechanism for third-party tools (in contrast to Java Remote Method Invocation, CORBA, or Remote Procedure Call) is essential. This technical integration is discussed in detail in the following section.

3 Extending MPEG-21 DIA with Semantic Adaptation Services.

Within two ISO/IEC MPEG Core Experiments [17,18], we have built an experimental adaptation framework based on Semantic Web Service technology where we used OWL-S together with SWRL to declaratively describe and dynamically execute (third-party implementations of) adaptation operations: OWL-S provides the computer-interpretable description of services and the means by which they can be accessed. It also comprises the mechanism to describe the functionality of a service in terms of a description of the transformation that is caused when a service is invoked. SWRL provides the XML format for describing logical facts and rules and can be used within OWL-S for specifying these preconditions and effects.

In detail, OWL-S descriptions are designed to provide three essential types of knowledge about a service: profile, process, and grounding. The *service profile* specifies what the service requires from the user or agent and tells us what it provides for them. The *process model* specifies how the service works. One can consider the profile as the interface for the service, whereas the process represents its internal structure. Finally, the *service grounding* specifies the details of how an agent can access the service on the technical level. This is accomplished by associating the OWL-S specifications with service descriptions expressed in WSDL.

WSDL itself specifies a protocol- and encoding-independent mechanism for Web Service providers to describe the means of interacting with the offered services. WSDL is an XML vocabulary which describes network-reachable services and maps these to a messaging-capable collection of communication endpoints. WSDL separates the abstract definition of service and messages from their concrete binding to a network port and message format. The current WSDL specification describes concrete bindings for SOAP, HTTP GET/POST, and MIME. However, also bindings for the Java programming language are available [9] which is important in our framework for multimedia adaptation. In Figure 5, an overview is given of how OWL-S, WSDL, and

[9] e.g., in the Apache Web Service project, http://ws.apache.org

MPEG-21 terminal capability descriptions are to be integrated for the provision of semantic adaptation services.

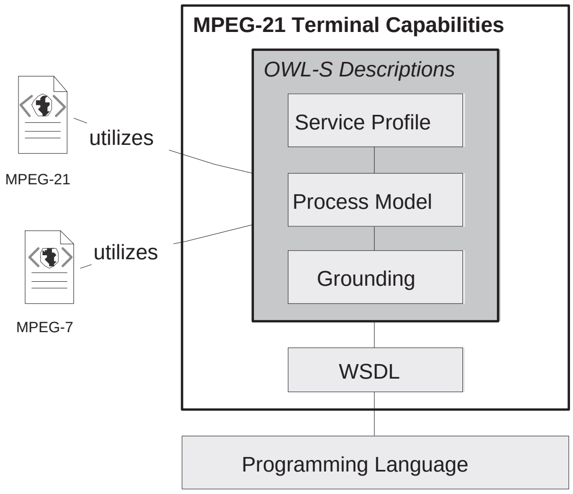

Fig. 5. Hierarchy of Multimedia Web Services.

Figure 6 shows the relevant parts of the OWL-S profile for a ImageGreyscale Service. One can see that the service consists of a single atomic process called *ImageGreyscaler*. The *profile*, i.e., the external view on the Semantic Web Service, also contains a declaration of the relevant input, output, preconditions, and effects, which are further specified in the subsequent *process* definition.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
  ...
  <owl:Ontology rdf:about="">
    <owl:versionInfo>
      $Id: ImageGreyscaler.owl,v 2.3 2005/01/13 klaus Exp $
    </owl:versionInfo>
    <rdfs:comment>OWL-S profile for an image greyscaler</rdfs:comment>
    ...
  </owl:Ontology>

  <profile:AtomicProcess rdf:ID=" ImageGreyscaler ">
    <!-- reference to the process specification -->
    <profile: has_process rdf:resource=" &greyscale_process;#ImageGreyscaler "/>
    <profile:serviceName>ImageGreyscaler</profile:serviceName>
    <!-- references to IOPE specification in the process -->
    <profile: hasInput rdf:resource=" &greyscale_process;#imageIn "/>
    <profile: hasOutput rdf:resource=" &greyscale_process;#imageOut "/>
    <profile: hasPreconditions rdf:resource=" &greyscale_process;#pre_color "/>
    <profile: hasEffect rdf:resource=" &greyscale_process;#eff_color "/>
  </profile:AtomicProcess>
</rdf:RDF>
```

References to the specifications in the process

Fig. 6. OWL-S Profile for a Greyscale Service

The OWL-S description for the process of our example is split into two XML documents, that together build the overall process description. Figure 7 shows fractions of the specification of inputs, outputs, and preconditions. It basically consists of an *atomic process* called *ImageGreyscaler* and a description of its properties. First, the input and output parameters are defined by mapping each parameter to an MPEG term in order to specify its meaning (i.e., the semantics). For instance, the parameter *imageIn* is mapped to the MPEG-7 term *urn:mpeg:mpeg7:cs:FileFormatCS:2001:11* which means that *imageIn* is a bitmap (BMP) image. For the precondition, the parameter *imageIn* is mapped to the MPEG-7 MediaInformation term

urn:mpeg:mpeg21:dia:cs:MediaInformationCS:13

which represents the color domain [10] . Furthermore, the standardized MPEG-7 *colordomain* value *color* is assigned to the precondition. A logic notation of the precondition may look like *colordomain(imageIn, color)* which means that the color-domain of the parameter *imgaeIn* is *color*.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
  ...
  <process:AtomicProcess rdf:ID="ImageGreyscaler">
    <!-- IOPE specification -->
    <process:hasInput>
      <process:Input rdf:ID="imageIn">
        <process:parameterType rdf:datatype="&xsd;#anyURI">
          urn:mpeg:mpeg7:cs:FileFormatCS:2001:11 <!-- BMP image -->
        </process:parameterType>
      </process:Input>
    </process:hasInput>

    <process:hasOutput>
      <process:Output rdf:ID="imageOut">
        <process:parameterType rdf:datatype="&xsd;#anyURI">
          urn:mpeg:mpeg7:cs:FileFormatCS:2001:11 <!-- BMP image -->
        </process:parameterType>
      </process:Output>
    </process:hasOutput>

    <process:hasPrecondition>
      <process:Precondition rdf:ID="pre_color">
        <expr:SWRL-Expression>
            <expr:expressionBody rdf:parseType="Literal">
              <swrlx:AtomList>
                <rdf:first>
                  <swrlx:datavaluedPropertyAtom>
                    <!-- imageIn is a colored BMP image -->
                    <owlx:Class owlx:name=" urn:mpeg:mpeg21:dia:cs:MediaInformationCS:13 "/>
                    <owlx:Individual owlx:name=" #imageIn"/>
                    <owlx:DataValue owlx:datatype="&xsd;string"> color</owlx:DataValue>
                  </swrlx:datavaluedPropertyAtom>
                </rdf:first>
               <rdf:rest rdf:resource="&rdf;#nil"/>
              </swrlx:AtomList>
            </expr:expressionBody>
          </expr:SWRL-Expression>
      </process:Precondition>
    </process:hasPrecondition>
```

Input and output are BMP images

colordomain(imageIn, color)

Fig. 7. Input, Output, and Preconditions for the OWL-S Process for a Greyscale Service

Figure 8 shows the effects of the greyscale process, represented as a list of SWRL atoms. In our example, the parameter *imageOut* is again mapped to the MPEG-7 color domain term but the value is `greylevel` which means that the output image is a greyscaled image. Furthermore, the MPEG-21 terminal constraint - no color display - is met by the greyscale service be-

[10] The MPEG-7 term *color domain* is defined in an MPEG-21 classification scheme.

cause of the second effect which defines the MPEG-21 color capable term *urn:mpeg:mpeg21:2003:01-DIA-AdaptationQoSCS-NS:6.5.9.26* to be *false*.

```
    <process:hasResult>
      <process:Result rdf:ID="eff_color">
        <process:hasEffect>
          <expr:SWRL-Expression>
            <expr:expressionBody rdf:parseType="Literal">
              <swrlx:AtomList>
                <rdf:first>
                  <!-- imageOut is a black and white BMP image -->
                  <swrlx:datavaluedPropertyAtom>
                    <owlx:Class owlx:name=" urn:mpeg:mpeg21:dia:cs:MediaInformationCS:13 "/>
                    <owlx:Individual owlx:name=" #imageOut"/>
                    <owlx:DataValue owlx:datatype="&xsd;string"> greylevel</owlx:DataValue>
                  </swrlx:datavaluedPropertyAtom>
                </rdf:first>
                <rdf:rest>
                  <swrlx:AtomList>
                    <rdf:first>
                      <!-- MPEG-21 constraint 'colorcapable=false' met -->
                      <swrlx:datavaluedPropertyAtom>
                        <owlx:Class owlx:name=" urn:mpeg:mpeg21:2003:01-DIA-AdaptationQoSCS-NS:6.5.9.26 "/>
                        <owlx:DataValue owlx:datatype="&xsd;boolean"> false</owlx:DataValue>
                      </swrlx:datavaluedPropertyAtom>
                    </rdf:first>
                  </swrlx:AtomList>
                </rdf:rest>
              </swrlx:AtomList>
            </expr:expressionBody>
          </expr:SWRL-Expression>
        </process:hasEffect>
      </process:Result>
    </process:AtomicProcess>
  </rdf:RDF>
```

colordomain(imageOut, greylevel)

colorcapable(false)

Fig. 8. Effects of the OWL-S Process for a Greyscale Service

The service profile and process of OWL-S are high-level semantic descriptions of a service. In order to invoke a service that is implemented in a programming language like Java one has to know more low-level, syntactic details. This is where WSDL comes into play. A WSDL document defines services as collections of network endpoints, or ports. In WSDL, the abstract definition of endpoints and messages is separated from their concrete network deployment or data format bindings, which allows us to reuse the abstract definitions: messages, which are abstract descriptions of the data being exchanged, and port types which are abstract collections of operations. The concrete protocol and data format specifications for a particular port type constitute a reusable binding. A port is defined by associating a network address with a reusable binding, and a collection of ports define a service. Hence, a WSDL document uses the following elements in the definition of network services:

- *Types*: a container for data type definitions using some type system (such as an XML Schema Definition).
- *Message*: an abstract, typed definition of the data being communicated.
- *Operation*: an abstract description of an action supported by the service.
- *Port Type*: an abstract set of operations supported by one or more endpoints.
- *Binding*: a concrete protocol and data format specification for a particular port type.
- *Port*: a single endpoint defined as a combination of a binding and a network address.

- *Service*: a collection of related endpoints.

In addition, WSDL also defines a *binding* mechanism, which is used to attach a specific protocol or data format or structure to an abstract message, operation, or endpoint and thus, allows for reuse of abstract definitions. In our example, we use Java bindings for WSDL as our reference implementation is based on this programming language. Figure 9 illustrates the relevant parts of the WSDL description for the *greyscale* service. The WSDL description consists basically of three parts which are *messages*, *port types*, and *bindings*. In the Java-WSDL binding, messages correspond to a set of method arguments. The port type specifies both input and output messages that belong to a method. Bindings are responsible for the mapping of abstract data types to concrete Java types. Furthermore, the binding part of the WSDL description binds the messages to parameters of a concrete Java method.

The crucial elements of the WSDL descriptions can be described as follows:

- *java:binding*: This indicates that the binding is a Java binding.
- *format:typeMapping*: Maps the WSDL types to Java types.
- *java:operation*: This element maps an abstract WSDL operation to a Java method. The *methodName* attribute specifies the name of the Java method corresponding to the abstract operation. The *returnPart* is that part of the abstract output message which corresponds to the return value of the Java method.

Once we have the semantic description of a service and the syntactical binding to the concrete implementation, the *OWL-S grounding* glues both parts together to offer a complete semantic multimedia web service to the clients. The grounding part is straight forward: First, the OWL-S atomic process is mapped to a WSDL operation and next, each OWL-S parameter is mapped to the corresponding WSDL message. Figure 10 shows the relevant parts of the OWL-S grounding description.

4 Reference implementation and evaluation.

In this section we will describe some details of a prototype implementation of the proposed knowledge-based adaptation server which was developed during the Core Experiments. Figure 11 gives an overview of the involved components and introduces the needed (MPEG-21 compliant) terminology.

Our adaptation server basically consists of three major components.

- The *Adaptation Decision Taking Engine (ADTE)*, which implements the

```
<?xml version="1.0"?>
  ...
  <!-- message declarations -->
  <message name="GreyscaleInput">
    <part name="imageIn" type="inputstream"/>
  </message>
  <message name="GreyscaleOutput">
    <part name="GreyscaleReturn" type="outputstream"/>
  </message>

  <!-- port type declarations -->
  <portType name="Greyscaler_port">
    <operation name="Greyscale">
      <input name="GreyscaleInput" message="GreyscaleInput"/>
      <output name="GreyscaleOutput" message="GreyscaleOutput"/>
    </operation>
  </portType>

  <!-- binding declarations -->
  <binding name="JavaBinding" type="Greyscaler">
    <java:binding/>  <!-- indicates that the binding is a Java binding.-->
    <format:typeMapping encoding="Java" style="Java">
      <format:typeMap typeName="inputstream" formatType="java.io.InputStream"/>
      <format:typeMap typeName="outputstream"  formatType="java.io.OutputStream"/>
    </format:typeMapping>
    <operation name="Greyscale_operation">
      <input/>
    </operation>
  </binding>

  <!-- service declaration -->
  <service name="Greyscaler">
    <port name="JavaPort" binding="JavaBinding">
      <java:address className="itec.adapt.ImageGreyscaler"/>
    </port>
  </service>
</definitions>
```

Mapping to concrete Java types

The Java implementation

Fig. 9. WSDL Description for a Spatial Scale Service

intelligence for constructing the needed adaptation sequences,
- the core resource and description *adaptation engine (AE)* that actually executes the transformations upon a client request, and
- a set of graphical *editing tools* that were developed for manually creating the semantic tool descriptions.

The *resources* needed in the framework are the actual multimedia content as well as the (external, third-party) adaptation tools and packages available locally on the server. In addition, our reference implementation also includes a set of *utilities*, that are e.g., used to parse and process the XML-based MPEG documents and to dispatch the client requests.

The **decision-taking component** of our framework is based on a lightweight backward-chaining planning algorithm implemented in Prolog which is capable of solving state-space planning problems expressed in a PDDL-compliant logic notation. As such, the currently un-optimized implementation can be easily exchanged by other planning tools that implement more recent, high-performance planning algorithms. The planning problem in general

```
<?xml version="1.0" encoding="ISO-8859-1"?>
  ...
  <!-- Grounding Instance for the Service -->
  <grounding:WsdlGrounding rdf:ID="GreyscalerServiceGrounding">
    <grounding:hasAtomicProcessGrounding rdf:resource="#GreyscalerGrounding"/>
  </grounding:WsdlGrounding>

  <!-- Grounding for Atomic Process ImageGreyscaler -->
  <grounding:WsdlAtomicProcessGrounding rdf:ID="GreyscalerGrounding">
    <grounding:owlsProcess rdf:resource="&greyscaleer_process;#ImageGreyscaler"/>
    <grounding:wsdlOperation>
      <grounding:WsdlOperationRef>
        <grounding:portType rdf:datatype="&xsd;#anyURI">
          &greyscaler_wsdl;#Greyscaler_port
        </grounding:portType>
        <grounding:operation rdf:datatype="&xsd;#anyURI">
          &greyscaler_wsdl;#Greyscale_operation
        </grounding:operation>
      </grounding:WsdlOperationRef>
    </grounding:wsdlOperation>

    <!-- grounding for input message -->
    <grounding:wsdlInputMessage rdf:datatype="&xsd;#anyURI">
      &greyscaler_wsdl;#Greyscale_Input
    </grounding:wsdlInputMessage>

    <!-- map OWL-S imageIn to WSDL imageIn -->
    <grounding:wsdlInput>
      <grounding:WsdlInputMessageMap>
        <grounding:owlsParameter rdf:resource="&greyscaler_process;#imageIn"/>
        <grounding:wsdlMessagePart rdf:datatype="&xsd;#anyURI">
          &greyscaler_wsdl;#imageIn
        </grounding:wsdlMessagePart>
      </grounding:WsdlInputMessageMap>
    </grounding:wsdlInput>

    <!-- grounding for output message -->
    <grounding:wsdlOutputMessage rdf:datatype="&xsd;#anyURI">
      &greyscaler_wsdl;#GreyscaleOutput
    </grounding:wsdlOutputMessage>

    <!-- map OWL-S imageOut to WSDL GreyscaleReturn -->
    <grounding:wsdlOutput>
      <grounding:WsdlOutputMessageMap>
        <grounding:owlsParameter
          rdf:resource="&greyscaler_process;#imageOut"/>
        <grounding:wsdlMessagePart rdf:datatype="&xsd;#anyURI">
          &greyscale_wsdl;#GreyscaleReturn
        </grounding:wsdlMessagePart>
      </grounding:WsdlOutputMessageMap>
    </grounding:wsdlOutput>
    ...
  </grounding:WsdlAtomicProcessGrounding>
</rdf:RDF>
```

OWL-S to WSDL parameter mapping

Fig. 10. OWL-S Grounding for a Greyscale Service

is known to be an NP-complete problem. In a first evaluation phase, we performed various experiments where we varied the number and types of available transformation operations as well as the complexity of the needed adaptation sequences. Our experiments indicate that the complexity of the plans (in particular the plan length, i.e., the number of steps) remains manageable: The needed plans are typically not longer than five or six steps.

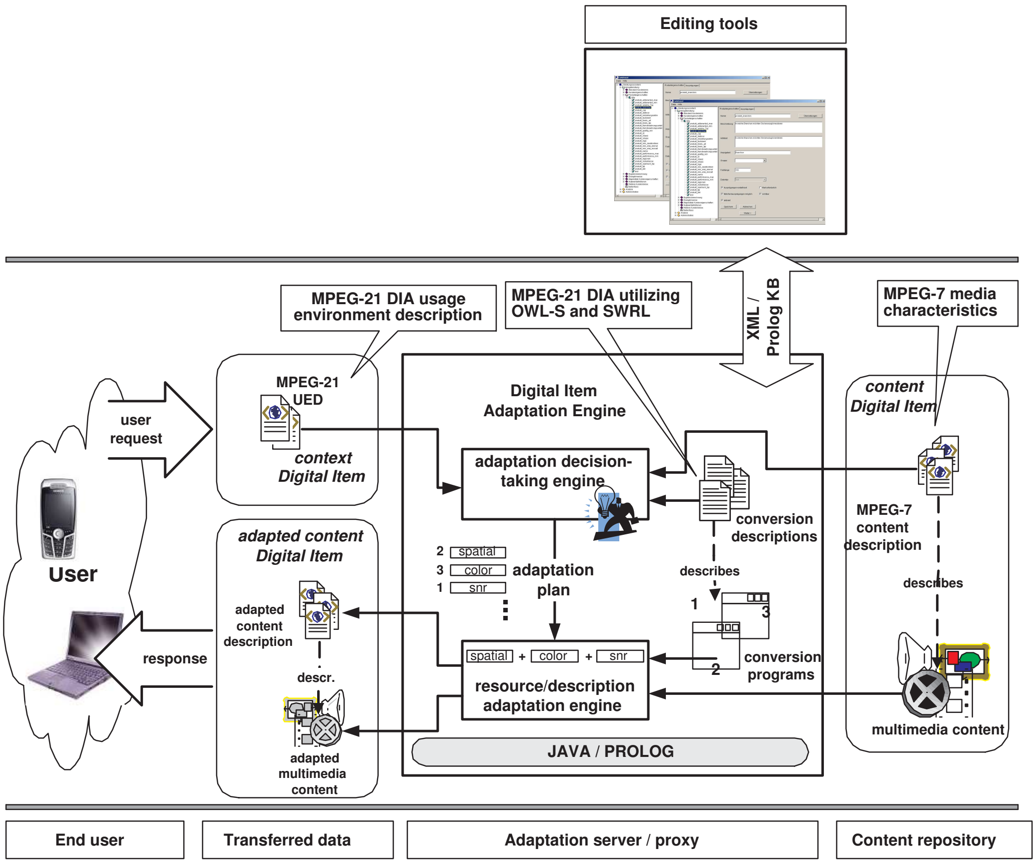

Fig. 11. Architecture overview.

We performed our first experiments on a standard desktop PC (Pentium M processor running at 2 GHz with 512 megabytes of RAM), i.e., no high-performance hardware was used which will be typically available an a multimedia server. Our measurements showed that all of the planning problems were solved in less than one second in such an environment. This time frame includes the time needed for computing the adaptation sequence alone and not the time for actually performing the adaptation on the resource.

When evaluating these running times, one has to have in mind that this computation has to be done only once at the beginning of a multimedia consumption session. The other operations which are required for instance for providing a video in a live stream like establishment of the session, buffering, or performing the actual transformation operations on the resource are typically far more costly. As such, the achieved results are promising as they suggest that the dynamic computation of adaptation sequences only takes a small amount of the overall time needed for providing the content to the consumer.

On the other hand we also evaluated the algorithm for situations where actually no adequate transformation could be made, e.g., because a particular encoding requested by the client is not supported by the multimedia server. In such situations, a multimedia server will have to implement some sort of fallback strategy, which is out of scope of this paper. Such a fallback strategy could be for instance to adapt the content only partially based on some heuristic or simply send the resource as it is and leave the adaptation to the client. Our evaluation showed that also the time needed for detecting such a situation where no adaptation plan can be produced is limited to the above-mentioned time frame of around one second.

Although the first results are already promising, our future work includes the integration of optimizations for further reducing the running times and increasing *plan quality*: Beside the usage of a high-performance planning engine, we are also working on finding good *heuristics* with respect to the optimal ordering of exchangeable steps in the plan. For instance, even in a very simple example, it will be better to reduce the size of an image in a first step before we do other operations like greyscaling on the image. In our current implementation, we use a priority-based annotation mechanism to steer the planning algorithm.

Overall, at the moment a comparison of the running times of our system with other approaches for computing adequate adaptation sequences in a declarative manner is problematic, since - due to the novelty of our approach - no benchmarks exist in the domain. A detailed comparison both with respect to performance as well as to extensibility with the basically hand-coded approach in [19] described in a later section will be part of our future work.

The next part of our framework is the **adaptation engine**, which is a central, Java-based component responsible for processing the client requests. Beside standard tasks like content-streaming, this component has the following specific responsibilities:

(1) Upon a client request, the accompanying MPEG-21 usage descriptions have to be parsed and interpreted.
(2) In a second step, the component checks the format of the requested resource and starts the plan-generation process, if required.
(3) Finally, the component retrieves the computed plan from the Decision Taking Engine and actually invokes the (locally available) transformation tools and individual methods, respectively.

In step (2), the existing MPEG-7 metadata annotations are analyzed in order to check if a transformation is required. If we have to adapt the resource, the next step is to *set up the parameters for the planning problem* for the decision taking engine. Technically, the module converts the XML-based metadata descriptions of the start and goal state into Prolog facts. This transformation is done with a proprietary, internal name conversion scheme. Note that the OWL-S descriptions of the tool capabilities have to be transformed into the internal representation only once at startup, as the set of available operations remains the same for all requests.

For improved performance, we implemented a first simple *plan cache* in this component which stores the different requests together with the computed plans, such that we can reuse the already existing plans by lookup once a similar request arrives. A part of our future work will be to find out improved algorithms to detect that requests are "similar", i.e., where we can reuse an existing plan and only adapt certain parameters without recalculating the whole sequence.

After a valid plan was computed or we found a cached plan, the actual transformations have to be applied. At the moment, our framework supports different types of invocation mechanisms. First, tools and individual methods can be called by using the described XML-based and platform-independent Web Service interface. In principle, these calls could also be to a remote service on a distant host.

For performance and experimental reasons, however, our prototype implementation also features programming language dependent method calls. These calls can be issued either via a Java-Wrapper interface or via a low-level native invocation mechanism[6], such that fast multimedia transformation libraries implemented in the C programming language like FFMPEG or ImageMagick can be easily included.

As a last action, the component can optionally produce an updated MPEG-7 metadata description for the adapted resource. The needed contents for the generation of the correct description (a variation of the description of the original resource) can come as a side result from the planning approach: We can model the changes that are relevant for the metadata descriptions as "effects" of transformation actions, i.e., produce additional facts that can be finally assembled to the new MPEG-7 facts.

Finally, during the experiments we have also developed different *editing and management tools*. These graphical tools for instance include an editor for creating the declarative tool descriptions, i.e., the OWL-S and SWRL files containing the IOPE descriptions as well as the references to the needed MPEG terms.

5 Related work

The work presented in this paper proposes a framework for building next-generation multimedia adaptation servers with the help of semantically annotated adaptation services based on Semantic Web technology. In the following, we will summarize the relations of our work to the different involved research areas. [11]

5.1 Ontologies in multimedia and Semantic Web technology

At the moment, machine-processable *ontologies* like used in Artificial Intelligence and Semantic Web communities only play a limited role in the context of multimedia research.

There exist some first approaches like reported in [20–23] that aim at exploiting the general knowledge-representation and reasoning mechanisms from the *Semantic Web* and OWL in order to open new possibilities for multimedia consumption.

Automated content-analysis for multimedia resources is the goal of the work described in [21]. In particular, they deal with the problem of (semi-) automatically extracting individual concepts and objects from spatio-temporal features or images. In their approach, they use Semantic Web technology for building domain-specific ontologies which shall support the concept identification process such that no human intervention is required. Their choice of using the now-standardized domain-independent knowledge representation mechanism is also driven by the idea that already existing ontologies that were built for other domains can be easily re-used for the extraction process.

A quite different use of ontologies for multimedia services can be found in [20]. In the SCULPTEUR project, a multimedia information system in a digital museum was developed, in which OWL-based ontologies are used as enabling technology for advanced concept-, metadata-, and content browsing and -retrieval functionality. The domain-specific ontology is integrated in a semantic layer that contains references to the actual objects of the digital collection, typically 2-D or 3-D images. However, in their approach, they mainly used Semantic Web technology as a common representation mechanism for integrating and harmonizing the different internal and external systems that actually hold the different pieces of the existing information of the digital repository like textual metadata of the collection's items.

[11] We will deliberately not focus on the literature on AI planning in this paper because we see our work mainly as an application of these techniques.

Another quite recent project where Semantic Web technology is used in the context of multimedia applications is OREL (Ontology-based Rights Expression Language) [23]. The goal of OREL is to enable fine granular access control and permission management for multimedia resources *on the semantic level.* With the help of OREL (which is built on top of the OWL Web Ontology Language), it shall for instance be possible to express rights like "Alice is granted to play the audio clip sample.mp3 for five times free of charge before Christmas 2005". However, MPEG-21 already contains features for developing fine-granular access control with the Rights Expression Language (REL) [24,25] and the Rights Data Dictionary (RDD) [26,27]. These existing MPEG-21 features are designed in a very flexible manner such that they could in principle be used to model such "semantic" access rights. Nonetheless, the usage of a standardized ontology language for implementing such special access rights can have the advantage of improved cross-application interoperability in the context of permission management. Still, the question of how OREL can be integrated in an interoperable way into the existing rights management of MPEG-21 is left open.

Overall, the approaches described so far basically use Semantic Web technology and ontologies as a tool for additional semantic metadata annotation that goes beyond the capabilities of MPEG-7. In contrast to these approaches, we view the MPEG standards as ontologies by themselves that we can subsequently use to semantically annotate possible *content transformations* and not annotate the original content itself.

With respect to Semantic Web Services, no previous work in the area of multimedia adaptation is known to the authors. In fact, the research field is still in its early phases and OWL-S as well as SWRL do not yet have the status of an W3C recommendation (like OWL). As such, current research efforts in the field are still focused on the development of a sound technical basis for Semantic Web Services in terms of integration into standard Web technology as well as on advanced algorithms that shall support automated Web Service identification, error recovery or service composition.

However, even in other domains, only very few first examples of real-world Semantic Web Service applications can be found at the moment. In [28], an application from the financial domain is described, where an intelligent *Notification Agent* manages alerts when critical financial situations arise by automatically discovering and selecting customer notification services. Comparable to our approach, they base parts of their framework on a commonly-agreed ontology (i.e., standard) in the domain, in their case the IFX financial standard. [12] In contrast to our work, where we use OWL-S and SWRL for modelling the possible actions they use *Description Logics* as the language for describing services

[12] see http://www.ifxforum.org/.

and a proprietary algorithm to describe and execute composite, multi-step services.

[29] describes a partially multimedia-related Semantic Web Services application from the academic environment. The *my*Campus environment aims at extending the nowadays relatively simple infotainment services available for *mobile* users by providing higher levels of automation and the development of services that also respect the *context* in which users operate. The main similarity compared with our approach lies in the fact that the authors aim at personalized delivery of (multimedia) content for different user environments, both with respect to hardware as well as software; in addition, they also use AI-based planning technology for the construction of composite services. In contrast to our work, however, they rely on proprietary, domain-specific ontologies for service annotations whereas we use the commonly-agreed multimedia standards as shared domain ontology. In addition, they do not aim at transforming the multimedia content by itself but rather at selecting and distributing existing resources in a personalized manner.

5.2 Multimedia adaptation servers and proxies.

Building distributed multimedia systems with the goal of performing server-side resource adaptation is a research area with a quite long tradition, see e.g., [30] for an overview. The main efforts in that area for a long time were spent with the problem of coping with the hard real-time constraints when delivering continuous media. These challenges were in many cases mainly tackled with the development of (proprietary) low-level stream adaptation algorithms. Only few projects are known at the moment that try to exploit the extended metadata annotation possibilities available with the new MPEG standards; examples are the *ViTooKi* [13] *Video Tool Kit* project [19,31] or the work described in [32].

ViTooKi provides an architecture for an adaptive proxy for audio/visual streams which is capable of adapting multimedia resources according to terminal capabilities (display size and color capabilities) and network characteristics (available bandwidth). For the task of determining adequate adaptation sequences, the concept of *adaptor chains* is introduced which enables them to concatenate several adaptation steps [19] in a predefined way. The "adaptor chains" are dynamically instantiated at runtime according to the usage environment. On the technical level, they also use the MPEG-7 *VariationSet Description Scheme* which can be used to provide hints as to when to apply an adaptation and which algorithm to use.

[13] http://vitooki.sourceforge.net/

In contrast to our knowledge-based approach, the construction process for adequate adaptation chains in *ViTooKi* is not automated in the sense that these chains are intelligently assembled from semantically-enriched tool descriptions. Rather, they rely on a set of pre-defined, manually engineered adaptation chains, which are looked up dynamically at run-time. When the adaptation chain is executed, the corresponding external tools like those contained in the *FFMPEG* library are parameterized and executed. Again, the set of such external tools and algorithms is manually engineered and the existing chains possibly have to be revised when a new tool is introduced to the framework, which is not necessary in our approach.

Nonetheless, our current work includes a detailed analysis of how our knowledge based approach can be integrated into the architecture of the *ViTooKi* project, in particular because this framework implements a solid, basic infrastructure and additional features like intelligent video caching mechanisms.

5.3 Adaptation QoS and further quality aspects

Bitstream Syntax Description (BSD) as a part of the current MPEG-21 DIA standard has some relation to our work. BSD is an additional, orthogonal metadata layer which can be used to describe the structure of the media's bitstream. With the help of a BSD, manipulations can be directly performed on the encoded bitstream without requiring costly decoding or encoding operations. BSD is mainly designed to be used together with *Adaptation QoS* [33,2], another tool defined within the MPEG-21 framework which enables users (primarily content producers) to describe quality of service (QoS) tradeoffs. Adaptation QoS specifies the relationship between environmental (e.g., terminal and network) constraints, media quality, and feasible adaptation operations. For example, a constraint could be the limitation of the network throughput on the path to the client. Adaptation of a video in this case could be accomplished by performing operations like, e.g., frame dropping, coefficient dropping, or wavelet-based reduction, which would lead to changes in the quality of the multimedia stream. Consequently, Adaptation QoS is useful to hint adaptation nodes to meet given constraints with regard to the quality of the adaptation result.

We see our work as more general and on a higher layer than BSD and Adaptation QoS. BSD could be one adaptation step in the presented architecture, providing a fast adaptation mechanism for compressed media streams. In addition, similar to BSD-based media stream adaptation, *transcoding* techniques and systems could be employed as building blocks (individual adaptation steps) in our higher-level architecture. Such a technique for videos is for instance described in [34].

Overall, with regard to *quality aspects* we have to discriminate between two different aspects in the context of adaptation in particular with respect to quality degradation that can come with such adaptations. In the multimedia consumption scenarios which are addressed within our framework, we mainly focus on improving the *quality of the user experience*: Potential quality degradations that result from adaptations are solely steered by the *usage environment* of the consumer, which may also contain specific user preferences. For instance, a consumer might *prefer* to consume the multimedia content in lower quality, e.g., view the video in lower resolution and color depth, being aware of the transfer costs of viewing the video in the best possible quality. All of such quality requirements of the end user have to be contained in the usage environment descriptions which correspond to the adaptation target. Due to the soundness of our knowledge-based approach, the search algorithm, and the precise semantic descriptions of the transformations, it can be guaranteed for such quality preferences that no adaptation sequences are computed by the planner that do not meet or exceed these requirements. Still - as already mentioned previously - we are currently working on the development of adequate algorithm extensions and heuristics such that we can also compute *good* solutions (adaptation chains) fulfilling the constraints as good as possible. The question of *acceptable* quality degradations will then be one of the major issues.

Finally, further investigations on the integration and the interaction of the Adaptation QoS tools with our framework will also be part of our future work. Currently, we view the functionality provided by these tools merely as additional means for further increasing the quality of service on the bitstream level *after* the other required higher-level transformations have been done. Quality degradations done in this phase are thus steered by additional heuristics and are currently not taken into account by the planner, i.e., particular care has to be taken when designing these heuristics.

6 Conclusions

The increasing heterogeneity of end-user devices and networks as well as the vision of ubiquitous multimedia consumption and Universal Multimedia Access create new and challenging requirements for intelligent, server-side multimedia adaptation servers.

In this paper we proposed a novel, knowledge-based framework for building next-generation multimedia adaptation servers, capable of constructing complex adaptation chains from semantically annotated transformation operations. We aim at reaching interoperability by using Semantic Web technology as a common knowledge representation mechanism and flexibility and extensibility through well-established Artificial Intelligence planning algorithms.

Within two ISO/IEC MPEG Core Experiments, we showed how our approach can be technically integrated into existing MPEG standards. The described proposal is now already contained in an (non-normative) amendment of the MPEG-21 DIA standard.

Our work also establishes a link between two up-to-then rather isolated standardization bodies (W3C, ISO/IEC) and advances both related research fields. On the one hand, we introduce knowledge-based approaches to the MPEG-communities and thus broaden the application scope of Artificial Intelligence technology. On the other hand, our problem setting showed that Semantic Web Service technology reaches a level of maturity such that complex, real-world service composition problems can be modeled and solved. Although adaptation services in our context are not necessarily distributed over several network nodes, we also view our work as a contribution in the field of Semantic Web Services in a more broader sense, as we showed how we can utilize this technology as a means of automated software construction.

Acknowledgements. We would like to thank the reviewers for their valuable comments that helped us improving the paper.

References

[1] F. Pereira, I. Burnett, Universal multimedia experiences for tomorrow, IEEE Signal Processing Magazine 20 (2) (2003) 63–73.

[2] A. Vetro, C. Timmerer, Digital item adaptation: Overview of standardization and research activities, IEEE Transactions on Multimedia 7 (2) (2005) 418–426.

[3] J. M. Martinez, R. Koenen, F. Pereira, MPEG-7 - the generic multimedia content description standard, part 1, IEEE MultiMedia 9 (2) (2002) 78–87.

[4] S. McIlraith, T. Son, H. Zeng, Semantic web services, IEEE Intelligent Systems, Special Issue on the Semantic Web 16 (2001) 46–53.

[5] D. Jannach, K. Leopold, C. Timmerer, H. Hellwagner, Toward Semantic Web Services for Multimedia Adaptation, in: Proceedings of 5th Intl. Conference on Web Information Systems Engineering (WISE), Lecture Notes in Computer Science 3306, Springer Verlag, 2004, pp. 641–652.

[6] D. Jannach, K. Leopold, H. Hellwagner, An extensible framework for knowledge-based multimedia adaptation, in: Proceedings of 14th Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE), Lecture Notes in Artificial Intelligence 2070, Springer Verlag, 2004, pp. 434–444.

[7] A. Vetro, C. Timmerer, ISO/IEC JTC 1/SC 29/WG 11 N5845, Text of ISO/IEC 21000-7 FCD - Part 7: Digital Item Adaptation, Trondheim, Norway, 2004.

[8] I. Bratko, Prolog Programming for Artificial Intelligence, 3rd Edition, Addison-Wesley, 2000.

[9] A. Blum, M. Furst, Fast planning through planning graph analysis, Artificial Intelligence 90 (1997) 281–300.

[10] M. Fox, D. Long, PDDL2.1: An extension to PDDL for expressing temporal planning domains, Journal on Artificial Intelligence Research 20 (2003) 61–124.

[11] D. Smith (ed.), Special issue on the 3rd international planning competition, Journal of Artificial Intelligence Research 20.

[12] J. Davis, D. Fensel, F. van Harmelen, Towards the Semantic Web: Ontology-driven knowledge management, John Wiley & Sons, 2003.

[13] I. Horrocks, P. F. Patel-Schneider, F. van Harmelen, From SHIQ and RDF to OWL: The making of a web ontology language, Journal of Web Semantics 1 (1) (2003) 7–26.

[14] D. Martin et al., Bringing semantics to web services: The OWL-S approach, in: Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004), San Diego, California, USA, 2004, pp. 6–9.

[15] I. Horrocks, P. F. Patel-Schneider, A proposal for an OWL rules language, in: Proceedings of the 13th International World Wide Web Conference (WWW'2004), New York, USA, 2004, pp. 723–731.

[16] J. Hunter, Adding adding multimedia to the semantic web - building an MPEG-7 ontology, in: Proceedings of the International Semantic Web Working Symposium (SWWS), August 2001, pp. 261–281.

[17] K. Leopold, C. Timmerer, "CE Report on Conversions", ISO/IEC JTC 1/SC 29/WG 11 M11742, Hong Kong, China, 2005.

[18] C. Timmerer, K. Leopold, D. Jannach, H. Hellwagner, M. Kimiaei, Report of CE on Conversion Parameters, ISO/IEC JTC 1/SC 29/WG 11 M11262, Palma De Mallorca, Spain, 2004.

[19] P. Schojer, L. Boeszoermenyi, H. Hellwagner, B. Penz, S. Podlipnig, Architecture of a Quality Based Intelligent Proxy (QBIX) for MPEG-4 Videos, in: Proceedings of the 2003 ACM World Wide Web Conference (WWW'2003), 2003, pp. 394–402.

[20] M. Addis, M. Boniface, S. Goodall, P. Grimwood, S. Kim, P. Lewis, K. Martinez, A. Stevenson, SCULPTEUR: Towards a New Paradigm for Multimedia Museum Information Handling, in: Proceedings of the International Semantic Web Conference 2003, FL, USA, Lecture Notes in Computer Science, 2870, Springer Verlag, 2003, pp. 582–596.

[21] I. Kompatsiaris, V. Mezaris, M. G. Strintzis, Multimedia content indexing and retrieval using an object ontology, in: Multimedia Content and Semantic Web Methods, Wiley, 2004, pp. 339–371.

[22] V. Mezaris, I. Kompatsiaris, M. G. Strintzis, Region-based Image Retrieval using an Object Ontology and Relevance Feedback, EURASIP Journal on Applied Signal Processing, Special Issue on Object-Based and Semantic Image and Video Analysis (2004) 886–901.

[23] Y. Qu, X. Zhang, H. Li, OREL: AN Ontology-based Rights Expression Language, in: Proceedings of the 2004 ACM World Wide Web Conference (WWW'2004) Alternate Track Papers and Posters, 2004, pp. 324–425.

[24] ISO/IEC 21000-5:2003, Information Technology - Multimedia Framework - Part 5: Rights Expression Language, 2003.

[25] X. Wang, T. DeMartini, B. Wragg, M. Paramasivam, The MPEG-21 rights expression language, IEEE Transactions on Multimedia 7 (3) (2005) 408–417.

[26] ISO/IEC 21000-6:2003, Information Technology - Multimedia Framework - Part 6: Rights Data Dictionary, 2003.

[27] X. Wang, T. DeMartini, B. Wragg, M. M. Paramasivam, C. Barlas, The MPEG-21 rights expression language and rights data dictionary, IEEE Transactions on Multimedia 7 (3) (2005) 408–414.

[28] J. M. López Cobo, S. Losada, Ó. Corcho, V. R. Benjamins, M. Niño, J. Contreras, SWS for financial overdrawn alerting, in: International Semantic Web Conference, 2004, pp. 782–796.

[29] M. Sheshagiri, N. Sadeh, F. Gandon, Using semantic web services for context-aware mobile applications, in: Proceedings of MobiSys2004 Workshop on Context Awareness, Boston, 2004, pp. 782–796.

[30] J. Gecsei, Adaptation in distributed multimedia systems, IEEE MultiMedia 4 (2) (1997) 58–66.

[31] L. Boeszoermeny, H. Hellwagner, H. Kosch, M. Libsie, S. Podlipnig, Metadata driven adaptation in the ADMITS project, EURASIP Signal Processing: Image Communication, Special Issue on Multimedia Adaptation 18 (2003) 749–766.

[32] O. Steiger, D. M. Sanjuán, T. Ebrahimi, MPEG-based personalized content delivery, in: Proc. of the IEEE Int. Conf. on Image Processing, ICIP 2003, Barcelona, Spain, 2003, pp. 14–16.

[33] ISO/IEC 21000-7:2004, Information Technology - Multimedia Framework - Part 7: Digital Item Adaptation, 2004.

[34] A. Vetro, C. Christopoulos, H. Sun, Video Transcoding Architectures and Techniques: An Overview, IEEE Signal Processing Magazine 20 (2) (2003) 18–29.

Author biographies

Dietmar Jannach works as research assistant at the Institute for Business Informatics and Application Systems, Klagenfurt University, Austria, where he received his MS and PhD degree in Computer Science. Dietmar Jannach's current research interests are mainly focused on applications of knowledge-based systems, in particular in the fields of intelligent product configuration, model-based diagnosis, interactive selling technology, and product recommendation.

Klaus Leopold is a research assistant at the Department of Information Technology at the Klagenfurt University, Austria, where he received his MS degree in Computer Science. Klaus Leopold's current research interests are distributed multimedia systems in general and multimedia adaptation, adaptation decision taking, as well as knowledge-based systems and the Semantic Web in specific.