ADVANCED MULTIMEDIA MANAGEMENT – CONTROL MODEL AND CONTENT ADAPTATION

Teodora Guenkova-Luy[1], Andreas Schorr[1], Franz Hauck[1], Miguel Gómez[2], Christian Timmerer[3], Ingo Wolf[4], and Andreas Kassler[5]

[1] Distributed Systems Department, University of Ulm, Ulm, Germany
{teodora.guenkova-luy, andreas.schorr, franz.hauck}@uni-ulm.de

[2] Agora Systems, Madrid, Spain
miguel.gomez@agora-2000.com

[3] Department of Information Technology, Klagenfurt University, Klagenfurt, Austria
christian.timmerer@itec.uni-klu.ac.at

[4] T-Systems International GmbH, Berlin, Germany
wolfi@t-systems.com

[5] Computer Science Department, Karlstad University, Karlstad, Sweden
kassler@ieee.org

ALPEN-ADRIA UNIVERSITÄT KLAGENFURT

Department of Information Technology (ITEC)
Klagenfurt University
Technical Report No. TR/ITEC/06/1.02
February 2006

Proceedings of the 24th IASTED International Multi-Conference
INTERNET AND MULTIMEDIA SYSTEMS AND APPLICATIONS
February 13-15, 2006, Innsbruck, Austria

ADVANCED MULTIMEDIA MANAGEMENT – CONTROL MODEL AND CONTENT ADAPTATION

Teodora Guenkova-Luy, Andreas Schorr and Franz Hauck
Distributed Systems Department
University of Ulm
Ulm, Germany
Email:{teodora.guenkova-luy, andreas.schorr, franz.hauck}@uni-ulm.de

Miguel Gómez
Agora Systems,
Madrid, Spain
Email: miguel.gomez@agora-2000.com

Christian Timmerer
Information Technology Department,
Klagenfurt University,
Klagenfurt, Austria
Email: christian.timmerer@Itec.uni-klu.ac.at

Ingo Wolf
T-Systems International GmbH,
Berlin, Germany
Email: WolfI@t-systems.com

Andreas Kassler
Computer Science Department
Karlstad University
Karlstad, Sweden
Email: kassler@ieee.org

ABSTRACT

The delivery and adaptation of multimedia content in distributed and heterogeneous environments requires flexible control and management mechanisms in terminals and in control entities inside the network. In the near future, it is important to reach interoperability between the IETF approaches on multimedia session establishment and control and the MPEG-21 efforts for multimedia streaming and adaptation to bring advanced multimedia service provisioning and adaptation services towards the customer. MPEG-21 Digital Item Adaptation (DIA) provides normative descriptions for supporting adaptation of multimedia content, but does not define interactions with transport and control mechanisms. On the other hand, the IETF standardization efforts on multimedia session control provide the necessary transport (e.g. RTP) and control mechanisms (SDP/SDPng). We thus bridge the gap between those approaches by creating a converged XML model that enables the integration of session management and negotiation protocols (e.g. SIP or Megaco) inspired by the XML formats of MPEG-21 DIA and SDPng. We also present preliminary implementation results of the converged model along with concepts and implementation of network-based content adaptation mechanisms through media gateways that enable flexible multimedia management for heterogeneous consumer terminals.

KEY WORDS

MPEG-21 DIA, SDP, SDPng, QoS, content adaptation, media gateways

1. Introduction

In the future, multimedia systems need to be flexible and adapt to the environment (e.g. wireless networks) in order to provide high quality communication in heterogeneous environment. The deployment of novel adaptive multimedia systems and services requires advanced description and control models for media. MPEG-21 DIA [1] provides normative XML descriptions (named tools) for applications that handle multimedia content (referred to as Digital Items) by shielding users from network and terminal management and implementation issues. A Digital Item is specified as a bundle of resources and (associated) metadata combined within a standardized structure that is used for various purposes within a generalized adaptation framework. However, MPEG-21 does not specify relations to existing technologies for transport mechanisms, in order to be independent of other specifications and open for future developments. On the other hand, Session Description Protocol next generation (SDPng) [2] defines a language for describing multimedia sessions with respect to technology-specific configuration parameters, terminals, conference services, media gateways, etc.

This contribution presents a practical approach for combining SDPng with MPEG-21 DIA descriptions. Both MPEG-21 DIA and SDPng formats adopt the eXtensible Markup Language (XML) [3], thus, a smooth integration of the two technologies is proposed. In this work, we provide a generalized session control mechanism applying an integrated version of SDPng and MPEG-21 DIA components. This mechanism is reusable also for backward compatibility with the Session Description Protocol (SDP) [4]. Furthermore, we present an adaptation management mechanism for multimedia that is able to fulfil the requirements of heterogeneous clients. It incorporates distributed content delivery and adaptation based on the generalized session control model. We discuss how existing mechanisms could be extended to provide an optimised management model for adaptive multimedia delivery.

2. Background

SDP [4] and SDPng [2] are used to describe multimedia services and applications. Usually, the respective descriptions are carried inside session layer protocols (like SIP [5], Megaco [6], etc.) to exchange configuration information for RTP-based multimedia streams [7]. The main reason for the transition from SDP to SDPng is the adoption of the XML format for session descriptions that shall enable extensible representation mechanism and higher flexibility at introducing new functions. In addition to already available mechanisms for codec descriptions and RTP configurations, SDPng shall enable negotiations of various Quality-of-Service (QoS) parameters associated with the content consumers and providers [8].

The MPEG-21 multimedia framework enables augmented and transparent use of multimedia resources across a wide range of terminals and networks [9]. The MPEG-21 framework is based on Digital Items (DIs) that contain information concerning their own adaptability. Thus, it is possible to customise the DIs on the fly within end-terminals (e.g. on a video server) as well as on network nodes if certain conditions are fulfilled, e.g. QoS aspects, network properties or user preferences. The adaptation process in DIA follows a generic concept for multimedia adaptation utilizing Bitstream Syntax Descriptions (BSDs) in order to remain codec agnostic [10].

As interoperability between those standards is strongly desired, an integrated model is necessary that combines MPEG-21 DIA and SDPng (denoted here as SDPng++) and ensures backward compatibility with applications using SDP. The XML specification of SDPng++ is proposed in [11] and adopted in [12]. SDPng++ reuses and extends the basic SDPng XML namespaces [11] and integrates MPEG-21 DIA Usage Environment (UED) and Bitstream Syntax Description (BSD) namespaces into the SDPng XML schema to provide mechanisms for QoS constraint descriptions at network and application level, which is not available through the basic SDPng. It also accommodates MPEG specific bitstream adaptation for multimedia payloads in RTP streams [11].

3. Description and Control Model for the Multimedia

This section gives an overview on the description and control model for multimedia sessions using SDP/SDPng and MPEG-21 DIA.

3.1 Session Control with SDP and SDPng++

SDP uses an ABNF-like format [13] that has a restricted depth of the definition hierarchy. SDPng applies XML [3] and non-limited depth of the definition hierarchy. It also distinguishes between simple variables (i.e. attributes) and object variables (i.e. elements). Hence, this may lead to structural mismatches and lack of interpretation expressiveness in situations where the applications' internal object-model specifies leaves of the description hierarchy. As an example, if SDP defines a mapping from a single key to a simple value, SDPng will require a key-to-object mapping for objects with a specific structure, multiple contained values and multiple identifiers for these values. In order to cope with this definition problem, we propose a format-independent model for session descriptions (Figure 1). The model represents the natural hierarchy of an adaptive multimedia session.

Figure 1: General definition and control model

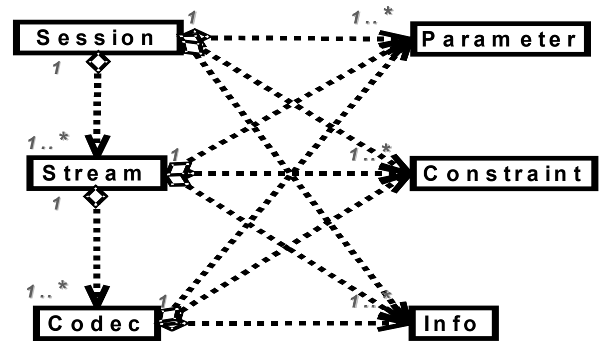


Table 1: Example of SDP/SDPng++ mapping to the general session description model

		ID	Parameter	Constraint	Info
SDP	Session	"**s=**" line or ID from the carrier Protocol (e.g. SIP Call-ID)	"**t=**" line	"**b=**" line	"**i=**" line
	Stream	"**m=**" line	"**c=**" line	"**b=**" line	"**i=**" line
	Codec	"**a=**"line	"**a=rtpmap**" line	"**b=**" line	"**a= recvonly**" line
SDPng++	Session	**Application/ sdpng** MIME-type	<**sdpng**> header element	<**m21-dia: Available Bandwidth**> element	<**sdpng-dia: BSDLive Link**> element
	Stream	<**component**> Element	<**rtp:udp**> Element	<**m21-dia: Available Bandwidth**> element	<**sdpng-dia: BSDLive Link**> element
	Codec	<**alt**> Element	<**audio:codec href="urn:mpeg: mpeg7:cs: AudioCoding FormatCS:2001: 4.4"**/> element	<**m21-dia: Available Bandwidth**> element	<**sdpng-dia: BSDLive Link**> element

Figure 2: SDPng++ XML processing

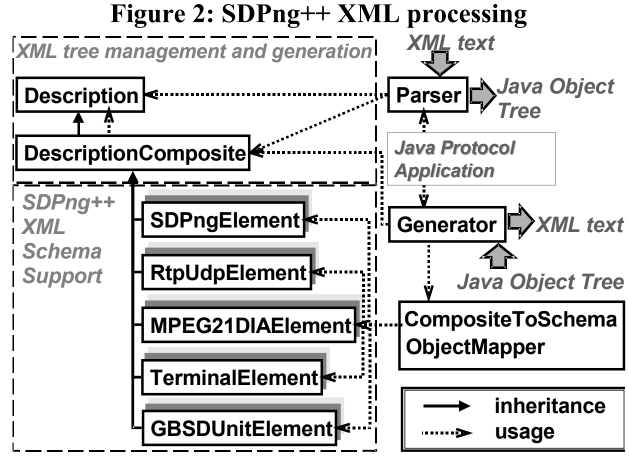


Figure 3: SDPng++ GUI Screen

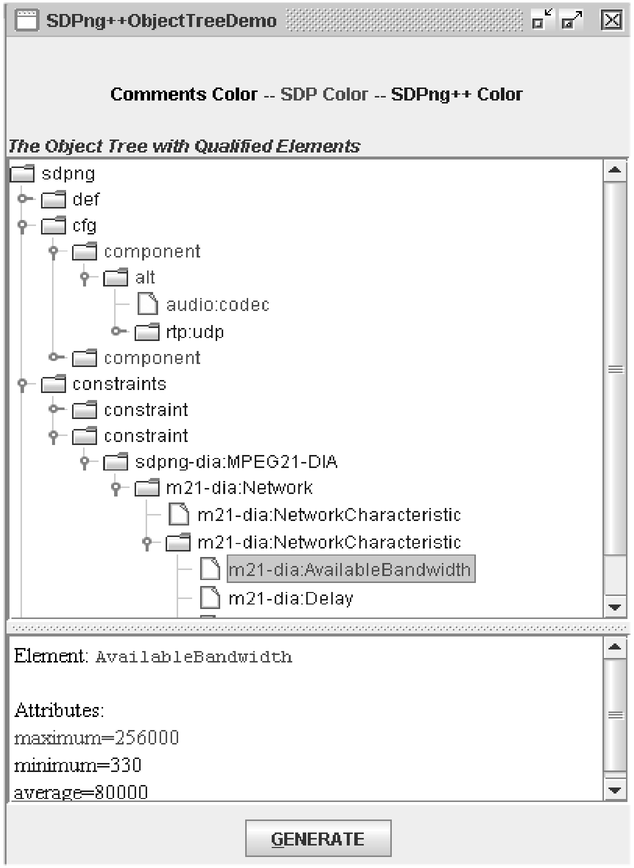


Figure 4: Network Content Adaptation Service [19]

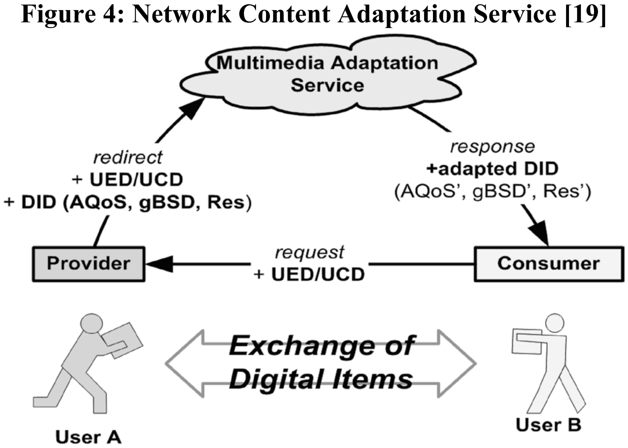


Table 2: Parser and Generator scaling behaviour

Input size [XML lines]	Average Processing Time [ms]	
	Parser	Generator
100	17.33	6.20
500	51.42	27.64
1000	105.76	44.37

In our model, a multimedia session contains one or multiple streams and each stream can be associated with one or multiple media codecs. Each session, stream and codec object contains an identifier (ID)(e.g. session type, number of the stream and codec name) and is associated with one or multiple Info, Constraint and/or Parameter objects. These objects contain an identifier and a general object (e.g. java.lang.Object [14]). The application of different value types within the general object makes the representation usable for alternative description models. It is up to the application to process the object value appropriately, e.g. upon the MIME-type of the description format specified in the description carrier (see [5], [6]). Table 1 shows an example of a mapping between SDP/SDPng++ elements and objects according to the general definition and control model. Due to the flat format of the SDP description model [4], there is no explicit separation of the input data into parameters, constraints and info objects. However, the semantics of the SDP lines enable such specific interpretation. Within the SDPng [2] and SDPng++ [11] models, there is a specific association between the XML elements and the three logical objects of the general model. The differentiation is expressed through the XML top-level elements as defined in SDPng (i.e. <cap>, <def>, <cfg>, <constraints> and <info>). In the general model, the information contained in the <cap>, <def> and <cfg> elements is interpreted as Parameters. The remaining two SDPng elements correspond one-to-one to the objects in the logical description model. The ID information for the SDP and SDPng sessions, streams and codecs corresponds either to an implicit entry or to explicit naming of the respective session, stream and codec. For instance the position of any "m=" line in SDP document interpreted as number is an implicit ID and the SDPng *name* attribute within an element is an explicit one.

3.2 SDPng++ Implementation

The current implementation of the SDPng++ includes core XML management using Java 1.4.x [14]. The XML processing of SDPng++ applies SAX parsing with the SAX interface for Java [15] and Apache SAX parser Xerxes-J [16]. The parsing of SDPng++ requires SAX in order to speed up the application processing as the SDPng++ code is used together with session control protocols that imply stringent time limitations. Currently, the SDPng++ is based on non-optimised XML conversion between SDPng and MPEG-21 DIA. Hence, resulting DOM XML object trees will be inefficient as the schema contains descriptions that are currently not used for the protocol purposes (e.g. some MPEG-21 UED definitions characterize local applications but are useless in distributed environments thus such definitions do not need to be negotiated within the scope of the distributed service management [11]).

The major elements of the SDPng core XML management are (Figure 2):

- The *Parser* implements SAX API for Java and re-uses XERCES-J. It generates non-specific data objects named *DescriptionComposite* that can represent any XML element with a value and attributes. The class *DescriptionComposite* inherits the abstract class *Description* that serves as an accumulator for the information within a single XML element (i.e. name, value as well as its attribute names and values). The *Description* provides abstract methods for the management of the information contained in an XML element. These methods are fully implemented within the *DescriptionComposite*. The Parser builds an object tree out of the generated Composite objects that corresponds to the XML tree in an SDPng++ message. The linking of the XML objects is provided through the *DescriptionComposite* mechanisms for accessing its direct parent and children objects (i.e. a *DescriptionComposite* object has only one parent that is also a Composite and one or many children Composites). For the linking purposes, a *DescriptionComposite* is able to accumulate one or multiple *Description* objects.

- The *Generator* is a customized generator for XML designed to produce any XML output document. It uses specific Java representations for the XML elements that correspond to the XML Schema definitions of SDPng++ (see [11] and Figure 2 that shows an example of some of the elements applied in the SDPng++ schema). Each of the specific objects is implemented as a single class that extends the *DescriptionComposite* in order to inherit the object-tree building mechanisms provided by the *DescriptionComposite*. Unlike the *DescriptionComposite*, every specific element-mapping object corresponds to just one element of the SDPng XML schema and provides specific management per defined attributes of the element and per defined children of the element using *Facade Design Pattern* (see [17]). The specific mapping is expressed the following way, e.g. for accessing the maximum bandwidth information the specific element-mapping object *AvailableBandwidthElement* provides a method *getMaxBandwidth()* whereas the information in the *DescriptionComposite* is accessed through the application of combinations of classes and their methods, like *composite.getAttributes()* that returns a *java.util.Hashtable* and the Hashtable has to be queried then about containing a key named *maximum* (see also the example in Figure 3). The specific element-mapping objects provide structures that are closer in their interpretation to the general definition and control model shown in Section 3.1. These structures can be directly used by the adaptive application. The specific objects define also object-to-XML serialization for each XML element represented though the object. The Generator applies this serialization mechanism to produce a valid SDPng XML document, but as the Generator does not understand the XML schema, it uses the *CompositeToSchemaObjectMapper* in order to provide respective validation for the resulting XML document.

The same XML processing model is used in [18] and some preliminary measurements thereof are shown in Table 2. A typical session description for a unicast session is 6-8kB (~100 lines XML code). The XML lines used for the measurements represent the text between two XML indentations (i.e. "<" and ">" separator symbols). The measurements were conducted on a PC equipped with a 700 MHz AMD Duron processor and 256 MB of memory. The relatively low processing power of the PC is intentionally chosen to emulate mobile terminal processing behaviour. Nevertheless, the processing procedures display good scalability.

4. Advanced Multimedia Content Adaptation

The MPEG-21 DIA enables device- and coding-format independence of media content [19]. We assume that multiple adaptation services sharing metadata will apply standards provided by the different standardisation bodies including MPEG, as it is not realistic to expect that a single adaptation engine is capable to handle all kinds of usage environments and coding formats that will emerge on the market. An example of such an adaptation service is depicted in Figure 4. In this example, User B (the consumer) requests a Digital Item from User A (the provider) by including her/his usage environment description. When content characteristics and usage environment conditions do not match, User B redirects the request including the content Digital Item (multimedia resource, content-related metadata) and the context Digital Item (usage environment description) to the adaptation service that performs the adaptation accordingly. Thereafter, the adaptation service provides the adapted content Digital Item to User B who is now able to consume the adapted Digital Item.

Examples of such adaptation service are media gateways or Content Adaptation Nodes (CANs). Within the scope of the IST-DAIDALOS project [20], we implemented such Content Adaptation Nodes utilizing different IETF and MPEG control and description technologies. The CAN operates as a network service and supports a certain set of adaptation operations. Hence, its usage is associated with terminals and CANs registration in the provider domain to assure trust relationship establishment between the components in the network. The CANs register also in a Service Discovery Server (SDS) within a domain and can be located via the Service Location Protocol (SLP) [21]. Each registration comprises the capabilities of the respective CAN, since different adaptation nodes may offer completely different content adaptation services (e.g. supporting different media codecs, different control and transport protocols, etc.). Therefore, a model for describing content adaptation services is developed. The service was designed in a flexible and modular fashion so that it can be deployed over a diverse set of network environments and tailored to the specifics of the desired usage scenario. However, in the scope of the DAIDALOS project, the service was deployed on a multimedia provisioning platform using SIP [5] for session negotiation, Megaco [6] for loosely-coupled CAN control and using a combination of SLP [21] and OWL [22] for Service Discovery.

Within each CAN, a decision-taking engine monitors the adaptation process and the feedback about the currently perceived network QoS (e.g. obtained via RTCP [7] receiver reports). If necessary, the decision-taking engine may decide to adapt the current parameters of the multimedia session and thus its quality. This allows to support environments with highly variable network QoS and nomadic terminals that may handover to different access network types having a great variation of available bandwidth and QoS support. The decision-taking engine is informed about the variation limits of the adaptation parameters using MPEG-21 DIA descriptions (e.g. of usage environments and preferences) as part of our SDPng++ model, which improves the utility of the CANs. The descriptions with SDPng++ are attached to SIP and Megaco messages (see below). Regarding usage scenarios, two main approaches are considered:

- **Implicit Adaptation Services** are triggered and configured user-transparently via session control elements (e.g. SIP Proxies) located in the network. This type of adaptation service is intended for users equipped with legacy terminals that are not capable of detecting environment changes and of configuring the adaptation process alone. Figure 5 shows a deployment example of such a service. When the SIP proxy receives the request from the consumer (1) it checks his/her user profile as well as the associated environmental constraints (e.g. network status, policy, provider capabilities, etc.) and decides whether an adaptation service should be inserted in the multimedia delivery chain or not (2). Several alternatives were evaluated to make the session control elements aware of the provider capabilities, such as including them in the user profile, proxy-driven polling after registration, splitting the transactions in several different dialogs using a SIP Application Server behaving as a Back-to-Back User Agent [23], etc. In case an adaptation is required, the proxy queries the service discovery server about CANs that can perform the adaptation process and finally selects the most appropriate CAN or a combination thereof (3-4). The session is configured within the selected CAN(s) using Megaco (5), and the information on the adaptation settings returned in the CAN responses (6) is used by the proxy to modify the initial multimedia session description offer provided by the consumer (7) prior to delivering it to the producer (8). The respective Megaco transactions include the creation of a new context at the CAN, associating to the CAN a send-only termination for the consumer and a receive-only termination for the producer. The producer includes in the SIP response a multimedia session description containing an answer to the received offer (9). The answer is modified (10) by the proxy using the information obtained in (6) prior to forwarding it to the consumer (11). The response acknowledgement (12-13) is followed by the media start (14-15).
- **Explicit Adaptation Services** (Figure 6) are triggered and configured directly by user terminals with no aid from the network logic apart from the basic media setup capabilities. This type of adaptation service is intended for users equipped with advanced terminals allowing them to setup and control the adaptation service without further assistance from proxies. After an unsuccessful session establishment attempt (see Figure 6) due to capabilities mismatch (1-6), the consumer decides to explicitly invoke a content adaptation service in order to be able to receive the desired media (7). After searching and selecting a suitable CAN applying Service Discovery (8-9), the consumer composes the adaptation service description (10) and configures it in the CAN using SIP third-party call control (11-16) [24]. The CAN behaves in this case as a SIP terminating User Agent in contrast to the previous scenario where Megaco is used, in order to allow media gateway configuration through a protocol suitable for end-terminals. The information returned with the CAN's response is used from the consumer to modify its initial offer to the producer (17), leading to a successful session establishment (18-23) allowing media exchange through the CAN (24-25). The adaptation process can also be triggered by the producer applying the inverse third-party call control procedure to the shown one.

Both implicit and explicit adaptation services can safely coexist thanks to the application of different user profiles and network policies that enable dynamic selection of adaptation services to be used within the session context. However, the application of the two scenarios simultaneously implies the usage of both SIP and Megaco.

Figure 5: Implicit Adaptation Service Setup

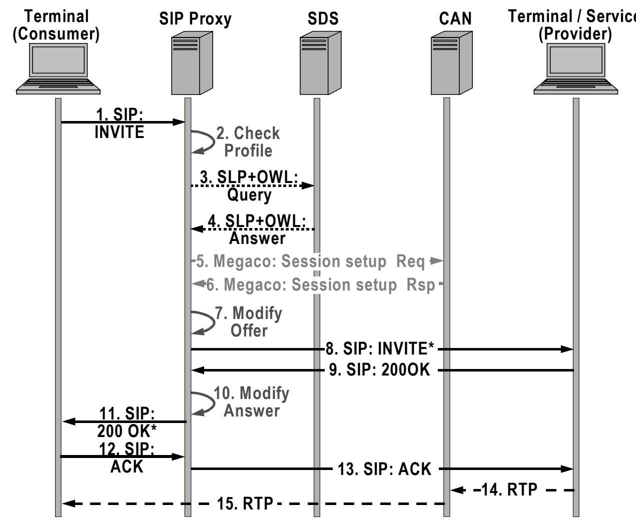
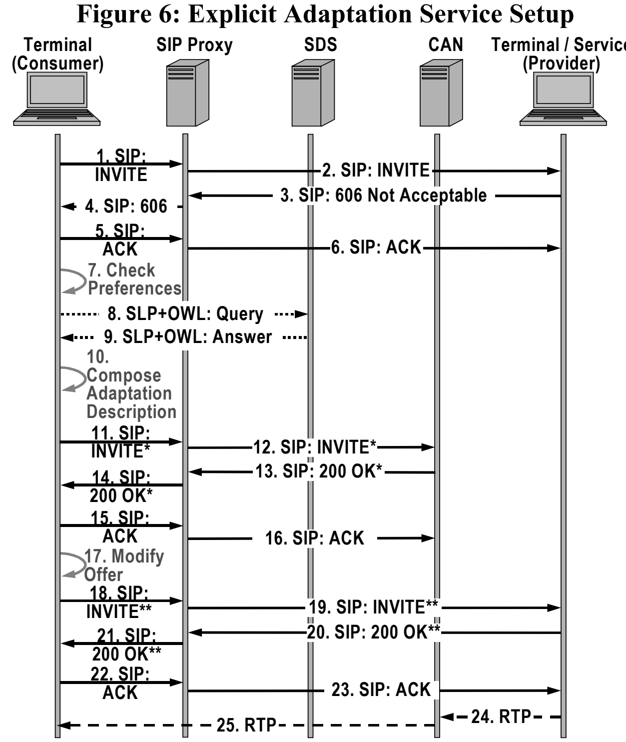


Figure 6: Explicit Adaptation Service Setup



5. Conclusion

This paper presented a novel approach that allows convergence between high-level, protocol-independent multimedia description formats and low-level, multimedia-enabling network protocols. The former (defined by MPEG) provide means for the construction of device and coding format independent adaptation services and the latter (defined by IETF) are used for transport, negotiation and exchange of these descriptions. Consequently, we introduced a general definition and control model and implementation thereof satisfying the requirements of already existing protocols (i.e. SDP) as well as protocols under development (i.e. SDPng). Seamless integration of media description formats as defined within MPEG-21 DIA is also provided. The scalability of the implementation is shown through tests and the usefulness of our SDPng++ model is demonstrated through the realization of two common usage scenarios within the IST-DAIDALOS project for configuring Content Adaptation Nodes. We believe that in order to support heterogeneous networks and environments of the future it is important to integrate such or similar proposals into upcoming versions of SDPng.

6. Acknowledgements

The work described in this paper is based on results of IST FP6 Integrated Project DAIDALOS, which receives research funding from the European Community's Sixth Framework Programme. Apart from this, the European Commission has no responsibility for the content of this paper. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability. Additional support is provided by the DFG within the AKOM framework.

References

[1] A. Vetro et al., Information Technology - Multimedia Framework (MPEG-21) - Part 7: Digital Item Adaptation, *ISO/IEC 21000-7:2004*, Oct. 2004

[2] D. Kutscher et al., Session description and capability negotiation, *IETF Work-in-progress, draft-ietf-mmusic-sdpng-08*, Feb. 2005

[3] W3C, Extensible Markup Language (XML) 1.0 (Third Edition), *W3C Recommendation*, Feb. 2004

[4] M. Handley et al., SDP: Session Description Protocol, *IETF RFC 2327*, April 1998 and *IETF Work-in-progress, draft-ietf-mmusic-sdp-new-25*, July 2005

[5] J. Rosenberg et al., SIP: Session Initiation Protocol, *IETF RFC 3261*, June 2002

[6] C. Groves et al., Gateway Control Protocol Version 1, *IETF RFC 3525*, June 2003

[7] H. Schulzrinne et al., RTP: A Transport Protocol for Real-Time Applications, *IETF RFC 3550*, July 2003

[8] J. Ott, C. Perkins, SDPng Transition, *IETF Work-in-progress, draft-ietf-mmusic-sdpng-trans-04*, May 2003

[9] F. Pereira, J. R. Smith, A. Vetro, Introduction to the Special Section on MPEG-21, *IEEE Trans. on Multimedia*, *7(*3*)*, June 2005, 397-399

[10] G. Panis et al., Bitstream Syntax Description: A Tool for Multimedia Resource Adaptation within MPEG-21, *EURASIP Signal Processing: Image Communication Journal*, *18(*8*)*, Sept. 2003, 721-747

[11] T. Guenkova-Luy et al., Harmonization of Session and Capability Descriptions between SDPng and MPEG-21 Digital Item Adaptation, *IETF Work-in-progress, draft-guenkova-mmusic-mpeg21-sdpng-00*, Feb. 2005

[12] S. Devillers, Information Technology - Multimedia Framework (MPEG-21) - Part 7: Digital Item Adaptation, AMENDMENT 2: Dynamic and Distributed Adaptation *ISO/IEC 21000-7:2004/PDAM 2*, Feb. 2006 (to appear)

[13] D. Crocker, P. Overell, "Augmented BNF for Syntax Specifications: ABNF", *IETF RFC 2234*, Nov. 1997

[14] Sun Microsystems, JavaTM 2 Platform, *Standard Edition, v 1.4.2*, Copyright 2003 Sun Microsystems, <http://java.sun.com/j2se/1.4.2/docs/api/>

[15] SourceForge.net, SAX, *SAX 2.0.2*, 27-April 2004, <http://www.saxproject.org/>

[16] Apache, Xerces Java Parser, *1.4.4 Release*, Copyright 1999, 2000 The Apache Software Foundation, <http://xml.apache.org/xerces-j/>

[17] E. Gamma et al., *Design Patterns – Elements of Reusable Object-Oriented Software* (Addison-Wesley, Reading-Massachusetts (USA), 1994)

[18] T. Guenkova-Luy, A. Kassler, D. Mandato, "End-to-End Quality of Service Coordination for Mobile Multimedia Applications", *IEEE Journal on Selected Areas in Communications, 22(*5*)*, June 2004, 889-903

[19] C. Timmerer, H. Hellwagner, Interoperable Adaptive Multimedia Communication, *IEEE Multimedia Magazine, 12(*1*)*, Jan. 2005, 74-79

[20] DAIDALOS, Designing Advanced network Interfaces for the Delivery and Administration of Location independent, Optimised personal Services, *IST-2002-506997*, <http://www.ist-daidalos.org/>

[21] E. Guttman et al., Service Location Protocol, Version 2, *IETF RFC 2608*, June 1999.

[22] Eric Miller et al., Web Ontology Language (OWL), *W3C Specification,* Feb. 2004

[23] 3GPP, Technical Specification Group Core Network, IP Multimedia (IM) session handling, IM call model, Stage 2 (Release 6), *3GPP TS 23.218 v6.3.0*, Mar. 2005

[24] G. Camarillo et al., Transcoding Services Invocation in the Session Initiation Protocol (SIP) Using Third Party Call Control (3pcc), *IETF RFC 4117*, June 2005