Adaptation de Contenu Multimédia avec MPEG-21: Conversion de Ressources et Adaptation Sémantique de Scènes

Mariam Kimiaei Asadi

TELECOM
PARIS
école nationale
supérieure des
télécommunications

Thèse

présentée pour obtenir le grade de docteur

de l’Ecole Nationale Supérieure des Télécommunications

Spécialité : **Informatique et Réseaux**

Mariam Kimiaei Asadi

Adaptation de Contenu Multimédia avec MPEG-21 :
Conversion de Ressources et Adaptation Sémantique de Scènes

Soutenue le 30 juin 2005 devant le jury composé de:

Cécile Roisin	**Rapporteurs**
Fernando Pereira	
Yves Mathieu	**Examinateurs**
Vincent Charvillat	
Nabil Layaïda	
Alexandre Cotarmanac’h	**Invité**
Jean-Claude Dufourd	**Directeur de thèse**

نبسته ام به کس دل، نبسته کس به من دل
چو تخته پاره بر موج رها رها، رها من

ز من هر آنکه او دور، چو دل به سینه نزدیک
به من هر آنکه نزدیک، ازو جدا، جدا من

نه چشم دل به سويي، نه باده در سبويي
که تر کنم گلويي، به ياد آشنا من

ستاره ها نهفته در آسمان ابري
دلم گرفته اي دوست، هواي گريه با من

شاعر: سیمین بهبهاني

Acknowledgments

I have very limited space to thank all the people who contributed in a way or another to the accomplishment of this thesis. This acknowledgement is therefore far from being perfect.

I am very grateful to my advisor Jean-Claude Dufourd who accepted me in the Multimedia team of ENST and prepared the conditions for me to be able to bring this thesis to its end. His complete expertise of the domain, as well as his constructive and well-placed judgements and technical ideas were profoundly valuable to me. Jean-Claude, I sincerely would like to thank you for the time you always reserved to me and my thesis, specially during these last months where you were rather occupied by some other issues.

I wish to thank Olivier Avaro and France Telecom R&D for providing the financial support for this thesis.

I would also like to thank Fernando Pereira and Cécile Roisin for having accepted to review this thesis as well as for their valuable comments that helped the improvement of the dissertation. My special thanks go also to other members of my jury: Yves Mathieu, Nabil Layaïda, Vincent Charvillat and Alexandre Cotarmanac'h for having accepted to be a part of the thesis committee.

I am indebted to the "unforgettable" ENST ComElec Multimedia gang: Cyril Concolato, Benoît Pellan, Philippe de Cuetos, Souhila Boughoufalah, Jean Lefeuvre and Ginaluca Di Cagno for their valuable contributions to this thesis. Thanks guys, you were there whenever I needed you…

I would also like to thank my friends: Gossass, Mammad, Paul, Joan, Yani, Dora, ti Lion, Maria, Clara, Greg, Sebastien, Farzaneh, Yorgo, Marina, Ideh, Sarah (from LA;)), Nora (the Newyorker;)), Mehdi Roghani, Alireza Babmdad, Maryam Kamarei, Bita and Elhaami.

A big thanks to Seb. Toujours bronzé!! Mais comment tu fais??! :))).

And those without whom this thesis would have never come to its end: Alireza (will never know how to thank you Alireza… you were and are a perfect friend to me…), ma belle blonde Sab (merci pour tout, pour ta présence, tes conseils, pour m'avoir écoutée et consolée quand ca n'allait pas. Tu le sais bien, tu es ma française préférée ;p).

And my precious Yol… Wow, already five years !! Cinq ans que tu as été tout simplement parfaite. You taught me a lot Yol. I will never find words good enough for thanking you, so I just say: don't leave us… stay in Paris Yol… please…

Now your turn, my absolute admirable Shermi! You astonish me every day. You are the best, you know it! Pour moi tu as toujours été là… pendent les hauts et les bas. Merci pour tout Shermi… pour toutes nos discussions KT !! Pour m'avoir supportée pendant toutes mes crises :) And God knows there were a lot of them… I most sincerely thank you…

And finally, my very special thanks to my parents, merci Mâmân, merci Bâbâ, without you I would have never been where I am today… And the rest of my family: Ahmad (you were always there for me, thanks…), Mohammad Hossain (so kind, soooo kind…), my lovely Mehdi (I love you more than every thing, every one… you are my reason…), my dearest sweetest Shirin (no more to say, your name explains every thing…) Nader, Helia (loos, boos:)), Shohreh, Reza, Mohammad, Mahboobeh, and my very dear Ali and Hamid. Thanks to all of you for having loved and supported me since ever...

Abstract

The objective of the Ph.D. thesis presented in this dissertation is to propose new, simple and efficient techniques and methodologies for support of multimedia content adaptation to constrained contexts. The work is based on parts of the on-going MPEG-21 standard that aims at defining different components of a multimedia distribution framework.

The thesis is divided into two main parts: single media adaptation and semantic adaptation of multimedia composed documents.

In single media adaptation, the media is adapted to the context constraints, such as terminal capabilities, user preferences, network capacities, author recommendations and etc. In this type of adaptation, the media is considered solely, i.e. as mono media, and without any multimedia structured presentation, or independently of the multimedia composition (scene) in which it exists. We have defined description tools extending the MPEG-21 DIA schema, for description of hints and suggestions on different media adaptations (also called Resource Conversions) and their corresponding parameters. We have consequently implemented a media adaptation engine that, based on these direct hints and suggestions, as well as context constraints, applies the most appropriate form of media adaptation with optimal values of adaptation parameters, in order to provide the end-user with the best quality-of-experience. Throughout this part of the work, we made several contributions to MPEG-21 DIA.

In semantic adaptation of structured multimedia documents, we addressed the question of adaptation based on temporal, spatial and semantic relationships between the media objects. When adapting a multimedia presentation, in order to preserve the consistency and meaningfulness of the adapted scene, the adaptation process needs to have access to the semantic information of the presentation. We have defined a language as a set of descriptors, for the expression of semantic information of composed multimedia content. These descriptors contain information provided by the author of the multimedia scene, or any other entity in the multimedia delivery chain, that helps the adaptation engine decide on the optimal type and nature of the adaptation(s) that are to be applied to the multimedia document. The information included in these descriptors cover the independent semantic information of each media

object of the scene, the semantic dependencies between media objects of the scene, and the semantic preferences on scene fragmentation.

In our implementation, we used SMIL 2.0 for describing multimedia scenes; however, the methodology is independent of this choice and can be applied to other types of multimedia documents such as MPEG-4 XMT. We implemented a proof-of-concept semantic adaptation engine that manipulates and adapts SMIL 2.0 documents, based on the semantic and physical information of the content, as well as context constraints.

Résumé

L'objectif de la thèse de doctorat présentée dans ce mémoire est de proposer des techniques et des méthodologies nouvelles, simples et efficaces pour l'adaptation de contenu multimédia à diverses contraintes de contexte d'utilisation. Le travail est basé sur des parties de la norme MPEG-21 en cours de définition, qui vise à définir les différents composants d'un système de distribution de contenus multimédia.

Le travail de cette thèse est divisé en deux parties principales : l'adaptation de médias uniques, et l'adaptation sémantique de documents multimédia composés.

Dans l'adaptation de médias uniques, le média est adapté aux contraintes du contexte de consommation, telles que les capacités du terminal, les préférences de l'utilisateur, les capacités du réseau, les recommandations de l'auteur, etc... Dans cette forme d'adaptation, le média est considéré hors de tout contexte de présentation multimédia structurée, ou indépendamment de la composition multimédia (scène) dans laquelle il est utilisé. Nous avons défini des outils et descripteurs, étendant les outils et descripteurs MPEG-21 DIA, pour la description des suggestions d'adaptation de médias (également appelée Conversion de Ressource), et la description des paramètres correspondants. Nous avons réalisé un moteur d'adaptation de médias qui fonctionne selon ces suggestions ainsi que selon les contraintes du contexte, et qui applique au media, la forme la plus appropriée d'adaptation avec des valeurs optimales des paramètres d'adaptation, afin d'obtenir la meilleure qualité d'utilisation. Durant cette partie du travail, nous avons apporté plusieurs contributions à la norme MPEG-21 DIA.

Dans l'adaptation sémantique de documents multimédia structurés, nous avons considéré l'adaptation selon les relations temporelles, spatiales et sémantiques entre les objets média de la scène. En adaptant une présentation multimédia afin de préserver l'uniformité et la logique de la scène adaptée, le processus d'adaptation doit avoir accès à l'information sémantique de la présentation. Nous avons défini un langage d'extension de la description de scène pour l'expression de cette information sémantique, à base de descripteurs. Ces descripteurs contiennent des informations fournies par l'auteur de la scène multimédia, ou par n'importe quelle autre entité dans la chaîne de livraison multimédia. L'information incluse dans ces

descripteurs aide le moteur d'adaptation à décider de la forme et de la nature optimales des adaptations qui doivent être appliquées au document. Cette information consiste en une information sémantique indépendante de chaque objet média, les dépendances sémantiques entre les objets média de la scène et les préférences sémantiques sur la fragmentation de scène.

Pour la réalisation d’un tel système d’adaptation, nous avons utilisé SMIL 2.0 pour décrire nos scènes multimédia. Cependant, la méthodologie est indépendante de ce choix et peut être appliquée à d'autres types de documents multimédia, tels que MPEG-4 XMT. Nous avons implémenté un moteur d'adaptation sémantique expérimental, qui manipule et adapte des documents SMIL, en utilisant les informations sémantiques et physiques du contenu, ainsi que des contraintes de contexte.

Table of Content

List of Figures

Chapter 1

INTRODUCTION

1.1 Introduction

A major part of existing multimedia content is originally authored for being transmitted over high-capacity networks and consumed on full-resource devices. However recent technical advances in the field of multimedia communications have led to letting users consume multimedia content over low-capacity, high-latency networks and limited devices, such as wireless networks and mobile terminals.

Appropriate delivery of such a large diversity of multimedia content to these different types of user devices and environments is a major challenge of a multimedia delivery chain. To do so, the content delivery chains require enough information on the context of the usage environment – network, device, user preferences, etc. – as well as the multimedia content itself, in order to be able to provide the end user with the most suitable form of the content.

To help the content adaptation process, the content creators also, should take into account at the authoring level, the adaptation-needed features of the multimedia content by creating adaptable content and by providing the necessary metadata for adaptation.

A knowledge-based and context-aware multimedia adaptation infrastructure is then needed to satisfy these requirements. Such an infrastructure will propose solutions for support of access, protection, adaptation and delivery of different content types. It will also propose methods to express context constraints as well as content-related information. MPEG (*Moving Picture Experts Group*) and W3C (*World Wide Web Consortium*) have provided recommendations and standards, which support and define frameworks for a multimedia adaptation system.

The work presented in this dissertation is based on the on-going MPEG-21 standard framework [1] and proposes methodologies to deal with problems of multimedia content adaptation to limited devices and for different usage contexts. Limited devices are multimedia terminals with reduced capacities in terms of display size, player capabilities, buffer size, power, etc.

In the field of multimedia content adaptation for constrained contexts, we have distinguished two main parts: single media adaptation and semantic adaptation of rich multimedia-composed documents. In single media adaptation, the media is considered solely, i.e. as mono media, and out of any multimedia structured presentation, or independently of the multimedia composition (scene) in which it exists. Therefore, this kind of adaptation is not a complete solution. In the second part of the work, we address the semantic adaptation of multimedia-composed documents based on temporal, spatial and semantic relationships between their media objects. Single media adaptation constitutes a part of semantic adaptation of multimedia-composed scenes. In semantic adaptation of multimedia scenes, each media object is adapted to the constraints of the usage context (device display size, device player capabilities, user preferences, etc.), based on the whole presentation scenario. Apart from single media resources, in semantic adaptation of multimedia-composed documents, the scene itself, i.e. the spatial, temporal and logical structure of the document is adapted to context constraints as well as to presentation semantic constraints.

This Ph.D. thesis has been performed in the MER (Multimédia Et Réseaux) group at ENST-Paris (Ecole Nationale Supérieure des Télécommunications de Paris). The work was performed for and mainly funded by France Telecom R&D. Parts of this work have been carried out in the context of two European IST projects: ISIS [2] and DANAE [3] and were partially funded by them.

1.2 Motivations and objectives

A complete knowledge-based multimedia content adaptation infrastructure is built upon a number of components. MPEG-21 is an on-going standard that aims at defining a multimedia framework that will enable transparent and augmented use of multimedia content across a wide range of networks and devices used by different communities. The framework is intended to cover the entire multimedia content delivery chain encompassing content production, protection, adaptation and delivery.

Apart from MPEG-21, there exist other standards that have been dealing with one or some of these mentioned multimedia communication components. However, MPEG-21 is the first “big picture” that can describe how all of these elements, either in existence or under development, relate to each other. That is why we have decided to base our work on MPEG-21 and make use of some of its well-defined solutions in the area of multimedia content adaptation.

MPEG-21 in its current form of the time this work started (2002-2003) was missing a number of features in the area of support of conversion-type adaptations (as defined later in Chapter 4). Along this research work, we tried to use advantageous solutions of MPEG-21 for realization of an experimental multimedia content adaptation framework, while proposing new solutions in the areas where MPEG-21 did not seem to propose a complete solution.

As such, MPEG-21 originally provided a complete support for a particular type of resource adaptation that implicitly targets scalable resources, and works on the basis of XML [4] description of the media bitstream and subsequent direct modification of the bitstream. However, at the time, MPEG-21 did not provide a complete support for the integration of adaptation of non-scalable media resources. This type pf media adaptation usually consists in decoding, manipulating and re-encoding the media, and is also called *Resource Conversion*. This is where our work on media adaptation contributed to MPEG-21. Our methodology for support of media resource conversion in MPEG-21 is based on the description of conversion-related information. This metadata helps the adaptation engine decide on the optimal form and parameters values of the adaptation that is to be applied to the media. Through our proposed descriptors, the content author or any other involved entity, can express his preferences and hints on certain conversions for a particular media.

MPEG-21 part 7 only proposes solutions for support of single media adaptation. The support of semantic adaptation of rich multimedia-composed presentations is not yet considered in MPEG-21. In the second part of this work, we defined new solutions for support of semantic adaptation of rich multimedia content, while remaining in the framework of MPEG-21 and using its advantageous solutions.

1.3 Main contributions of this work

This thesis makes the following contributions:

➢ The first part of the work proposes new simple and efficient solutions for support of single media adaptation (Resource Conversion) under the framework of the MPEG-21 standard:

- We participated in a contribution that resulted in adopting descriptors for expression of media conversion preferences to MPEG-21 part 7 (DIA).
- An amendment to MPEG-21 part 7 was established on December 2003, to consider the support of media resource conversion. We have participated in the establishment of this amendment and contributed to it during its evolution. At this time, this amendment specifies solutions for the description of conversion-related information in MPEG-21 part 7. We continue to contribute to this amendment, on

the specification of a new tool that uses DIA *AdaptationQoS* description tool for dynamic expression of conversion parameter values, and supports a more efficient adaptation decision-making.

➢ The second part of our work in the area of adaptation of rich multimedia content, proposes solutions for semantic adaptation of multimedia-composed scenes. We used these solutions under the framework of MPEG-21, and showed that the adaptation engine needs to have access to the semantic information of multimedia content, in order to perform a consistent, meaningful and appropriate adaptation to the multimedia scene. Based on this theory we implemented a semantic adaptation engine for SMIL multimedia documents.

1.3.1 List of contributions to MPEG standardization body

Here is the list the contributions of this work to MPEG-21 standard:

Mariam Kimiaei Asadi, Souhila Boughoufalah and Jean-Claude Dufourd, “Proposed additions to DIA WD 3.0”, ISO/IEC JTC1/SC29/WG11 M9257, December 2002, Awaji, Japan.

Souhila Boughoufalah, Mariam Kimiaei Asadi, Jean-Claude Dufourd, “Proposal for adding User Interaction support to terminal capabilities in DIA”, ISO/IEC JTC1/SC29/WG11/MPEG2002/M9259, December 2002, Japan, Awaji.

Truong Cong Thang, Yong Ju Jung, Yong Man Ro (ICU), Jeho Nam (ETRI), Mariam Kimiaei-Asadi, Jean-Claude Dufourd (ENST), “Report of CE on Modality Conversion Preference (Part 1)”, ISO/IEC JTC1/SC29/WG11/M9495, March 2003, Pattaya, Thailand.

Truong Cong Thang, Yong Ju Jung, Yong Man Ro (ICU), Jeho Nam (ETRI), Mariam Kimiaei-Asadi, Jean-Claude Dufourd (ENST), “Report of CE on Modality Conversion Preference, Part 2: Syntax Refinement of Presentation Priority”, ISO/IEC JTC1/SC29/WG11/M9496, March 2003, Pattaya, Thailand.

Mariam Kimiaei-Asadi, Jean-Claude Dufourd, Truong Cong Thang, “Report of CE on Transmoding”, ISO/IEC JTC1/SC29/WG11/M9745, July 2003, Trondheim, Norway.

Mariam Kimiaei-Asadi, Jean-Claude Dufourd, Truong Cong Thang, Yong Man Ro, Jeho Nam, “Report of CE on Transmoding”, ISO/IEC JTC1/SC29/WG11/M10111, October 2003, Brisbane, Australia.

Mariam Kimiaei-Asadi, Jean-Claude Dufourd, “Proposal to Digital Item Adaptation Amendment 1 on Conversion Descriptors and Transmoding”, ISO/IEC JTC 1/SC 29/WG 11/M10697, March 2004, Munich, Germany.

Mariam Kimiaei-Asadi, Jean-Claude Dufourd, "Contribution to Digital Item Adaptation Amendment 1 on Transcoding and Transforming Conversions", ISO/IEC JTC 1/SC 29/WG 11/M10698, March 2004, Munich, Germany.

Christian Timmerer, Klaus Leopold, Dietmar Jannach, Hermann Hellwagner, Mariam Kimiaei, "Report of CE on Conversion Parameters", ISO/IEC JTC1/SC29/WG11, MPEG2004/M11262, October 2004, Palma de Mallorca, Spain.

Cyril Concolato, Philippe de Cuetos, Mariam Kimiaei-Asadi, Benoît Pellan, "Proposition of ConversionLink tool to DIA Amd 1", ISO/IEC TC1/SC29/WG11/M11670, January 2005, Hong-Kong, China.

Cyril Concolato, Philippe de Cuetos, Mariam Kimiaei-Asadi, Benoît Pellan, Truong Cong Thang, Yong Ju Jung, Yong Man Ro, Jeho Nam, Jae-Gon Kim, Eric Delfosse, "Report of CE on the use of AdaptationQoS for Conversions", ISO/IEC JTC1/SC29/WG11/M11884, April 2005, Busan, Korea.

1.3.2 List of publications

Here is the list of our publications on this work:

Mariam Kimiaei-Asadi, Jean-Claude Dufourd, "Adaptation de Contenus Multimédia par Transmodage dans MPEG-21", Proceedings of Première Conférence Nationale sur le Multimédia Mobile: Mcube 2004, Montbéliard, France.

Alexandre Cotarmanac'h, Kaveh Kamyab, Gabriel Panis, Cyril Concolato, Mariam Kimiaei Asadi et al. "ISIS: Intelligent Scalability for Interoperable Services", Proceedings of first IEE European Conference on Visual Media Production: CVMP 2004, London, UK.

Mariam Kimiaei-Asadi, Jean-Claude Dufourd, "Multimedia Adaptation By Transmoding in MPEG-21", Proceedings of 5th International Workshop on Image Analysis for Multimedia Interactive Services: WIAMIS 2004, Lisbon, Portugal.

Mariam Kimiaei-Asadi, Jean-Claude Dufourd, "Knowledge-based and Semantic Adaptation of Multimedia Content", Proceedings of European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology: EWIMT 2004, London, UK.

Mariam Kimiaei-Asadi, Jean-Claude Dufourd, "Context-aware Semantic Adaptation of Multimedia Presentations", accepted in IEEE International Conference on Multimedia & Expo: ICME 2005, Amsterdam, Netherlands.

Mariam Kimiaei-Asadi, Jean-Claude Dufourd, “Media Resource Conversion in MPEG-21”, accepted in 8th Joint Conference on Information Sciences - 4th International Conference on Intelligent Multimedia Computing and Networking: JCIS - IMMCN 2005, Salt Lake City, Utah, USA.

Mariam Kimiaei-Asadi, Jean-Claude Dufourd, “Support de Transmodage de Contenus Multimédia dans MPEG-21”, accepted paper in Journal of Techniques et Sciences Informatiques, Hermes Publication, expected to be published in spring 2005.

1.4 Outline

The remainder of this dissertation is organized as follows:

Chapter 2. This chapter provides a brief introduction on the basic concepts and objectives of the MPEG-21 standard by emphasizing on the parts that are related to the presented work.

Chapter 3. The objective of this chapter is, firstly, to provide the basic notions, definitions and main components of a single media adaptation framework, and secondly to provide a state-of-the-art and discuss the related work in this area.

Chapter 4. This chapter describes our proposed solutions and the theory behind our methodology in the area of single media adaptation.

Chapter 5. This chapter presents the architecture of our experimental implementation of a single media adaptation engine, based on the methodology described in previous chapter.

Chapter 6. This chapter provides a description of the principal concepts, definitions and components of a framework for multimedia-composed content adaptation, and to, subsequently discuss the state-of the-art on this subject.

Chapter 7. This chapter describes the elements of our proposed solution and the theory of our methodology in the area of multimedia-composed content adaptation.

Chapter 8. This chapter presents the architecture of our proof-of-concept implementation of a semantic multimedia-composed content adaptation engine.

Chapter 9. This chapter concludes the presented work, describes its achievements and discusses envisaged perspectives.

Chapter 2

MPEG-21 STANDARD: CONCEPTS AND OBJECTIVES

Summary

This chapter gives a brief introduction on the concepts and objectives of the on-going MPEG-21 standard. The emphasis is on the parts of MPEG-21, which are more related to my Ph.D. thesis work.

Table of Content

Table of Illustrations

2.1 Introduction

The demand, creation and then consumption of multimedia content are rapidly increasing. A large variety of devices with different capabilities, are used to access this multimedia content. Additionally, these access devices may be used in different contexts, i.e. in different environments, and by different users, each having particular consumption preferences. A complete multimedia adaptation infrastructure is needed to deal efficiently with all the issues of this new multimedia usage context.

All content providers share the very same concerns: management of content, adaptation of content to consumer and device capabilities, protection of rights, protection from unauthorized access/modification, protection of privacy of providers and consumers, etc.

Currently, multimedia technology provides content creators and consumers with a myriad of coding, access and distribution possibilities. At the same time, communication infrastructure is being put into place to enable access to information and multimedia services from almost anywhere and at anytime. Still, no global end-to-end solutions exist allowing all different user communities to interact in an interoperable way. This lack of interoperable – and thus standardized – solutions is stalling the deployment of advanced multimedia packaging and distribution applications, although most of the individual technologies are indeed already present.

New solutions are required for the access, delivery, management, and protection of different content types in an integrated and harmonized way. This should be implemented in a manner that is entirely transparent to the many different users of multimedia services.

A framework is therefore needed which would define such solutions in order to enable the transparent and augmented use of multimedia resources across a wide range of networks and devices. This motivated MPEG group (ISO/IEC JTC1 SC29 WG11) in June 2000 to start to work on the definition of an enabling normative technology for the multimedia applications of the 21st century: MPEG-21 “Multimedia Framework”. MPEG-21 or ISO/IEC 21000 is a standard of ISO (International Organization for Standardization) family. ISO is a worldwide federation of national standards bodies.

2.2 MPEG-21 Multimedia Framework: ISO/IEC 21000

MPEG-21 Multimedia Framework (ISO/IEC 21000) aims at defining a normative open framework for multimedia delivery and consumption for use by all the players in the delivery and consumption chain [5]. This open framework will provide content creators, producers, distributors and service providers with equal opportunities in the MPEG-21 enabled open market. Another purpose is the benefit of the content consumers, providing them access to a large variety of content in an interoperable manner.

The goal of MPEG-21 can thus be rephrased as defining the technology needed to support users to exchange, access, consume, trade and otherwise manipulate multimedia content in an efficient, transparent and interoperable way.

MPEG-21 identifies and defines the mechanisms and elements needed to support the multimedia delivery chain – as described above, as well as the relationships between and the operations supported by them. Within the parts of MPEG-21, these elements are elaborated by defining the syntax and semantics of their characteristics, such as interfaces to the elements.

The MPEG-21 Multimedia Framework is based on two essential concepts: the definition of a fundamental unit of distribution and transaction (the Digital Item) and the concept of Users interacting with Digital Items. The Digital Items can be considered the "what" of the Multimedia Framework (e.g., a video collection, a music album) and the Users can be considered the "who" of the Multimedia Framework.

2.2.1 User Model

A User is any entity that interacts in the MPEG-21 environment or makes use of a Digital Item. Such Users include individuals, consumers, communities, organizations, corporations, consortia, governments and other standards bodies and initiatives around the world. Users are identified specifically by their relationship to another User for a certain interaction. From a purely technical perspective, MPEG-21 makes no distinction between a "content author" (or content provider) and an "end user" (consumer) – both are Users. A single entity may use content in many ways (publish, deliver, consume, etc.), and so all parties interacting within MPEG-21 are categorized as Users equally. A User may assume specific or even unique rights and responsibilities according to his interaction with other Users within MPEG-21 [5].

At its most basic level, MPEG-21 provides a framework in which one User interacts with another User and the object of that interaction is a Digital Item, commonly called content. Some such interactions are creating content, providing content, archiving content, rating content, enhancing and adapting content, aggregating content, delivering content, syndicating content, retail selling of content, consuming content, subscribing to content, regulating content, facilitating transactions that occur from any of the above, and regulating transactions that occur from any of the above. Any of these are “uses” of MPEG-21, and the parties involved are Users [5]. Among “MPEG-21 people”, another term that is sometimes used for User, is MPEG-21 Peer.

2.2.2 Digital Item

Since the “Digital Item” is the very basic concept of MPEG-21, there is a need for a very precise description of what constitutes such an “item”. A Digital Item is a structured digital object with a standard representation, identification and metadata within the MPEG-21 framework. This entity is also the fundamental unit of distribution and transaction within this framework. Clearly there are many kinds of content, and probably just as many possible ways of describing it to reflect its context of use. This presents a strong challenge to layout a powerful and flexible model for Digital Items, which can accommodate the myriad forms that content can take (and the new forms it will assume in the future). Such a model is only truly useful if it yields a format that can be used to represent unambiguously Digital Items defined within the model, to communicate them, and to describe them successfully. The Digital Item Declaration specification (part 2 of ISO/IEC 21000) provides such flexibility for representing Digital Items. (See section 2.3.2). ISO/IEC 21000 consists of several parts. Sixteen parts of MPEG-21 standardization within the Multimedia Framework have started (Part 13 is not attributed):

- Part 1: Vision, Technologies and Strategy
- Part 2: Digital Item Declaration (DID)
- Part 3: Digital Item Identification (DII)
- Part 4: Intellectual Property Management and Protection (IPMP)
- Part 5: Rights Expression Language (REL)
- Part 6: Rights Data Dictionary (RDD)
- Part 7: Digital Item Adaptation (DIA)
- Part 8: Reference Software
- Part 9: File Format
- Part 10: Digital Item Processing (DIP)
- Part 11: Evaluation Methods for Persistent Association Technologies
- Part 12: Test Bed for MPEG-21 Resource Delivery
- Part 14: Conformance

- Part 15: Event Reporting
- Part 16: Binarization
- Part 17: Fragment Identification for MPEG Media Types

2.3 MPEG-21 parts

In the following sections, an introduction is given to some parts of MPEG-21, with special attention to those parts most related to this work: DID (section 2.3.2) and DIA (section 2.3.7).

2.3.1 MPEG-21 Part 1: Vision, Technologies and Strategy

Part 1 of MPEG-21 (ISO/IEC 21000-1 [5]) is a technical report that provides:

- A *vision* for a multimedia framework to enable transparent and augmented use of multimedia resources across a wide range of networks and devices to meet the needs of all Users
- A framework to facilitate the integration of components and standards in order to harmonize *technologies* for the creation, management, manipulation, transport, distribution and consumption of content;
- A *strategy* for achieving a multimedia framework by the development of specifications and standards based on well-defined functional requirements through collaboration with other bodies.

The use case example of Figure 2.1, given in this part of MPEG-21, considers a distributed multimedia system comprising a certain number of Users exchanging Digital Items across a wide range of networks such as the Internet, mobile phone connections, etc., to a variety of terminals. In order to guarantee a smooth delivery of such Digital Items over heterogeneous networks “adaptations” of media resources may be required, for example to overcome network congestion or to allow DIs to be routed at the same time to Users connected via mobile devices and to Users on fixed lines. A large number of adaptation possibilities exist, e.g., videos can be adapted by simple frame dropping or by modifying the quantization coefficients. Such adaptations can happen anywhere in the delivery chain from the DI provider to the DI consumer and can be governed by rights expressions.

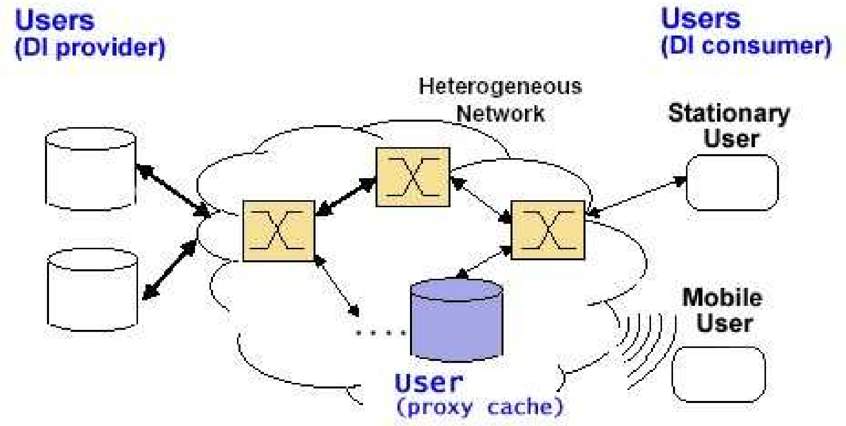


Figure 2.1: Example: Distributed Multimedia System [5]

2.3.2 MPEG-21 Part 2: Digital Item Declaration

The purpose of the Digital Item Declaration (DID) specification is to provide a model and a language for describing Digital Items. Within this model, as described earlier, a Digital Item is the digital representation of "a work", and as such, it is the thing that is acted upon (managed, described, exchanged, collected, etc.). The goal of this model is to be as flexible and general as possible, while providing for the "hooks" that enable higher-level functionality. This, in turn, allows the model to serve as a key foundation in the building of higher-level models in other MPEG-21 elements (such as Identification, Description or IPMP). This model specifically does not define a language in and of itself. Instead, the model helps to provide a common set of abstract concepts and terms that can be used to define such a scheme, or to perform mappings between existing schemes capable of Digital Item Declaration, for comparison purposes.

The DID technology is described in three normative sections:

➢ Model: The Digital Item Declaration Model describes a set of abstract terms and concepts to form a useful model for defining Digital Items.

➢ Representation: Normative description of the syntax and semantics of each of the Digital Item Declaration elements, as represented in XML. This section also contains some non-normative examples for illustrative purposes.

➢ Schema: Normative XML schema comprising the entire grammar of the Digital Item Declaration representation in XML.

Consider a simple "web page" as a Digital Item. A web page typically consists of an HTML document with embedded "links" to (or dependencies on) various image files (e.g., JPEGs and GIFs), and possibly some layout information (e.g., Style Sheets). In this simple case, it is a straightforward exercise to inspect the HTML document and deduce that this Digital Item consists of the HTML document itself, plus all of the other resources upon which it depends.

Now let us modify the example to assume that the “web page” contains some custom scripted logic (e.g., JavaScript, etc.) to determine the preferred language of the viewer (among some predefined set of choices) and to either build/display the page in that language, or to revert to a default choice if the preferred translation is not available. The key point in this modified example is that the presence of the language logic clouds the question of exactly what constitutes this Digital Item now and how this can be unambiguously determined.

The first problem is one of actually determining all of the dependencies. The addition of the scripting code changes the declarative “links” of the simple web page into links that can be (in the general case) determined only by running the embedded script on a specific platform. This could still work as a method of deducing the structure of the Digital Item, *assuming* that the author intended each translated “version” of the web page to be a separate and distinct Digital Item. This assumption highlights the second problem: it is ambiguous whether the author actually intends for each translation of the page to be a standalone Digital Item, or whether the intention is for the Digital Item to consist of the page with the language choice left unresolved. If the latter is the case, it makes it impossible to deduce the *exact* set of resources that this Digital Item consists of, which leads back to the first problem.

The above-stated problems are addressed by the Digital Item Declaration. A Digital Item Declaration (DID) is a document that specifies the makeup, structure and organization of a Digital Item. Part 2 of MPEG-21 contains the DID Specification.

The following sections describe the semantic “meaning” of some of the principal elements of the Digital Item Declaration Model. We do not describe all elements and it should be noted that in the descriptions below, the defined entities in *italics* are intended to be unambiguous terms within this model.

Container. A *container* is a structure that allows *items* and/or *containers* to be grouped. These groupings of *items* and/or *containers* can be used to form logical packages (for transport or exchange) or logical shelves (for organization). *Descriptors* allow for the “labeling” of *containers* with information that is appropriate for the purpose of the grouping (e.g. delivery instructions for a package, category information for a shelf, or as used in this work and described later, the semantic information of a multimedia scene). A *container* itself is not an item; *containers* are groupings of *items* and/or *containers*.

Item. An *item* is a grouping of sub-*items* and/or *components* that are bound to relevant *descriptors*. *Descriptors* contain information about the *item*, as a representation of a work. *Items* may contain *choices*, which allow them to be customized or configured. *Items* may be conditional (on *predicates* asserted by *selections* defined in the *choices*). An *item* that contains no sub-*item*s can be considered an entity -- a logically indivisible work. An *item* that does contain sub-*items* can be considered a compilation -- a work composed of potentially independent sub-parts. *Items* may also contain *annotations* to their sub-parts.

The relationship between *items* and Digital Items (as defined in [6]) could be stated as follows: *items* are declarative representations of Digital Items.

Component. A *component* is the binding of a *resource* to all of its relevant *descriptors*. These *descriptors* are information related to all or part of the specific *resource* instance. Such *descriptors* will typically contain control or structural information about the *resource* (such as bit rate, character set, start points, encryption information, modality, format, encoding, author's hints or other parameters e.g. the key frames for a video resource) but not information describing the "content" within. *Components* are building blocks of *items*.

Descriptor. A *descriptor* associates information with the enclosing element. This information may be a *component* (such as a thumbnail of an image, or a text component), a textual, or an XML *statement*.

Condition. A *condition* describes the enclosing element as being optional, and links it to the *selection*(s) that affect its inclusion. Multiple *predicates* within a *condition* are combined as a conjunction (an AND relationship). Any *predicate* can be negated within a *condition*. Multiple *conditions* associated with a given element are combined as an OR relationship, when determining whether to include the element.

Choice. A *choice* describes a set of related *selections* that can affect the configuration of an *item*. The *selections* within a *choice* are either exclusive (choose exactly one) or inclusive (choose any number, including all or none).

Selection. A *selection* describes a specific decision that will affect one or more *conditions* somewhere within an *item*. If the *selection* is chosen, its *predicate* becomes true; if it is not chosen, its *predicate* becomes false; if it is left unresolved, its *predicate* is undecided.

Resource. A *resource* is an individually identifiable asset such as a video or audio clip, an image, or a textual asset. A *resource* may also potentially be a physical object. All *resources* must be locatable via an unambiguous address.

Statement. A *statement* is a literal textual value that contains information, but not an asset. Examples of likely *statements* include descriptive, control, revision tracking or identifying information.

Figure 2.2 is an example showing the most important elements within this model, how they are related, and the hierarchical structure of the Digital Item Declaration Model. An XML schema is designed for declaring Digital Items, which is meant to be as flexible and general as possible.

DIDL Overview

DIDL documents are XML 1.0 documents. In addition, DIDL syntax is based on an abstract structure defined in the Digital Item Declaration Model. The following abstract entities defined in the Model: *container*, *item*, *component*, *anchor*, *fragment*, *descriptor*, *choice*, *selection*, *condition*, *annotation*, *assertion*, *resource*, *statement*, are each represented in DIDL by a like-named DIDL element: <Container>, <Item>, <Component>, <Anchor>, <Fragment>, <Descriptor>, <Choice>, <Selection>, <Condition>, <Annotation>, <Assertion>, <Resource>, <Statement>. For example, the abstract *descriptor* entity in the Model is represented in DIDL by the <Descriptor> element.

A DIDL document consists of a DIDL root element with a single <Item> child element or <Container> child element. Thus, a DIDL document can represent either an *item* or a *container*. In addition, DIDL defines the following special element types that do not correspond to any of the Model entities:

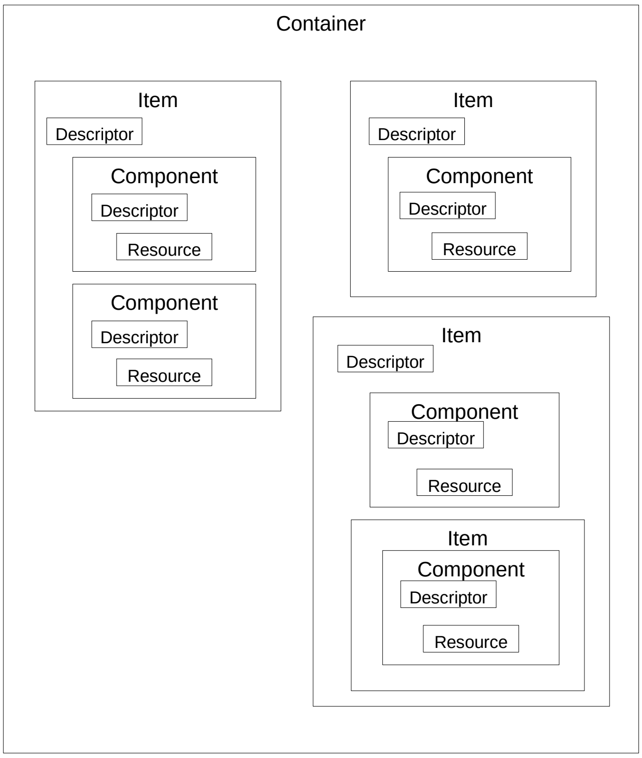


Figure 2.2: Relationship of the principal elements within the DID Model

<Reference>, and <Declarations>. These special elements are used for specific purposes within DIDL. The <Reference> element is used to link the contents of an element inside another element. The <Declarations> element is used to define a set of DIDL elements in a document without actually instantiating them. A declared element (i.e. a child element of a <Declarations> element) is not considered to be instantiated unless it is referenced (by a <Reference> element).

DIDL makes broad use of XML's ID attribute type. Generally, attributes of this type are used to make an internal association between one DIDL element and another. For example, many DIDL elements have an ID attribute, which makes them available as targets of internal references by <Reference> elements, and, in limited cases, available for annotation by <Annotation> elements. In addition, other attributes of type ID that are not named 'id' are used to make specific kinds of associations between specific elements. For example, the "select_id" attribute of the <Selection> element allows <Condition> elements to be associated with specific <Selection>s. It is strongly recommended that attributes of type ID be assigned globally unique values, in order to avoid collisions when DIDL documents are merged. This is especially important when there are external dependencies on the ID values. Finally, it is noted that the use of the ID attribute should not be confused with normative identification mechanisms defined in any other part of ISO/IEC 21000.

The DIDL element is the root element of a DIDL instance document. It may contain an optional DECLARATIONS element, followed by exactly one <Container> or <Item>. The DIDL element must include a namespace declaration that declares the DIDL namespace for the root DIDL element and its contents. This is required so that applications will recognize the document as a DIDL document, which is covered by DIDL specification.

The DIDL namespace URI is "urn:mpeg:mpeg21:2002:02-DIDL-NS". The "02" represents a serial number that is expected to change as the DIDL schema evolves along with this part of ISO/IEC 21000. Figure 2.3 shows a simple DIDL instance.

In our work, we have used DID model and DIDL language for describing our content and context Digital Items.

2.3.3 MPEG-21 Part 3: Digital Item Identification

The third part of MPEG-21 (ISO/IEC 21000-3), entitled Digital Item Identification (DII), mainly specifies how to uniquely identify Digital Items and parts thereof. In this work we have not used DII. More information on this is available in [7].

2.3.4 MPEG-21 Part 4: Intellectual Property Management and Protection

This part of MPEG-21 defines an interoperable framework for Intellectual Property Management and Protection (IPMP) [8]. Fairly soon after MPEG-4, with its IPMP hooks, became an International Standard, concerns were voiced within MPEG that many similar devices and players might be built by different manufacturers, all MPEG-4, but many of them not inter-working.

```
<DIDL xmlns="urn:mpeg:mpeg21:2002:02-DIDL-NS">
  <Declarations>
    <Descriptor id="PHOTO_INFO">
      <Statement mimeType="text/plain">
        Taken with my new SnazzyCam
      </Statement>
    </Descriptor>
  </Declarations>
  <Item>
    <Descriptor>
      <Statement mimeType="text/plain">Photo Album #1</Statement>
    </Descriptor>
    <Item>
      <Descriptor><Reference target="#PHOTO_INFO"/></Descriptor>
      <Component>
        <Resource ref="myFirstPicture.jpg" mimeType="image/jpeg"/>
      </Component>
    </Item>
    <Item>
      <Descriptor><Reference target="#PHOTO_INFO"/></Descriptor>
      <Component>
        <Resource ref="mySecondPic.bmp" mimeType="image/x-ms-bmp" />
      </Component>
    </Item>
  </Item>
</DIDL>
```

Figure 2.3: A simple example for a DIDL instance

This is why MPEG decided to start a new project on more interoperable IPMP systems and tools. The project includes standardized ways of retrieving IPMP tools from remote locations, exchanging messages between IPMP tools and between these tools and the terminal. It also addresses authentication of IPMP tools, and has provisions for integrating Rights Expressions according to the Rights Data Dictionary and the Rights Expression Language. Efforts are currently on-going to define the requirements for the management and protection of intellectual property in the various parts of the MPEG-21 standard currently under development. We have dealt with intellectual property management, and have therefore not used IPMP in this work.

2.3.5 MPEG-21 Part 5: Rights Expression Language (REL)

This part of ISO/IEC 21000 [9] specifies the syntax and semantics of a Rights Expression Language. It does not give any permission, including permissions about who is legally or technically allowed to create Rights Expressions. It does not specify the security measures of trusted systems, propose specific applications, or describe the details of the systems required for accounting (monetary transactions, state transactions, and so on), either. It also does not specify if or when Rights Expressions shall be consulted. However, this part of ISO/IEC 21000 does define an authorization model to specify whether the semantics of a set of Rights Expressions permit a given Principal to perform a given Right upon a given optional Resource during a given time interval based on a given authorization context and a given trust root. A Rights Expression Language is seen as a machine-readable language that can declare rights and permissions using the terms as defined in the Rights Data Dictionary. We have not dealt with Digital Right Management (DRM) issues, and have therefore not used REL in this work.

2.3.6 MPEG-21 Part 6: Rights Data Dictionary (RDD)

This part of ISO/IEC 21000 [10] describes a Rights Data Dictionary which comprises a set of clear, consistent, structured, integrated and uniquely identified Terms to support the MPEG-21 Rights Expression Language (REL), ISO/IEC 21000-5.

Use of the RDD System will facilitate the accurate exchange and processing of information between interested parties involved in the administration of rights in, and use of, Digital Items, and in particular it is intended to support ISO/IEC 21000-5 (REL). As well as providing definitions of Terms for use in ISO/IEC 21000-5, the RDD System is designed to support the mapping of Terms from different namespaces. Such mapping will enable the transformation of metadata from the terminology of one namespace (or Authority) into that of another namespace (or Authority). The dictionary is based on a logical model (the Context Model), which is the basis of the dictionary ontology. The model is described in detail in a normative annex to the specification. It is based on the use of verbs, which are contextualized so that a dictionary created using the model, can be as extensible and granular as required. An annex explaining how the model can be applied to generate new Terms is also provided.

We have not dealt with Digital Right Management (DRM) issues, and have therefore not used RDD.

2.3.7 MPEG-21 Part 7: Digital Item Adaptation (DIA)

A great part of the work presented in this dissertation is constructed upon the basis of this part of MPEG-21. Hence, comparing to our brief introductions on other parts of MPEG-21, in this section we look at DIA in more details. However, where necessary in other chapters, more in-depth explanations and examples will be given on different parts of DIA.

One of the principal goals of MPEG-21 is to achieve interoperable transparent access to (distributed) advanced multimedia content by shielding users from network and terminal installation, management and implementation issues. This will primarily enable the provision of network and terminal resources on demand so that multimedia content can be created and ubiquitously shared, always with the agreed/contracted quality, reliability and flexibility. Towards this goal, the adaptation of Digital Items is required.

As shown in Figure 2.4, Digital Items are subject to a Resource Adaptation Engine, as well as a Description Adaptation Engine, which together produce the adapted Digital Items. The aim of this part of the standard is to specify tools that provide input to the adaptation engine, so that any constraints on the delivery and consumption of resources can be satisfied, and the user experience quality can be guaranteed. It is important to emphasize that the adaptation engines themselves are non-normative tools of Digital Item Adaptation. However, descriptions and format-independent mechanisms that provide support for Digital Item Adaptation in terms of resource adaptation, descriptor adaptation, and/or Quality of Service management are within the scope of this part of the standard.

ISO/IEC 21000-7 [11] specifies the syntax and semantics of tools that may be used to assist the adaptation of Digital Items, i.e., the Digital Item Declaration and resources referenced by the declaration. Users can use the tools to satisfy transmission, storage and consumption constraints, as well as Quality of Service management. DIA does not specify the adaptation engines themselves. The DIA tools in this specification are clustered into eight major categories as illustrated in Figure 2.5.

The categories are clustered according to their functionality and use for DIA around the *Schema Tools* and *Low-Level Data Types*. The schema tools provide uniform root elements for all DIA descriptions as well as some low-level and basic data types, which can be used by several DIA tools independently. The detailed syntax and semantics of the schema tools and low-level data types are specified in [11].

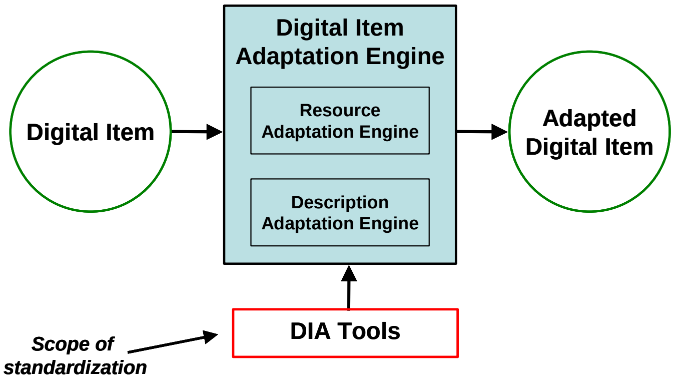

Digital Item Adaptation Engine
Resource Adaptation Engine
Description Adaptation Engine
Digital Item
Adapted Digital Item
DIA Tools
Scope of standardization

Figure 2.4: Illustration of Digital Item Adaptation [11]

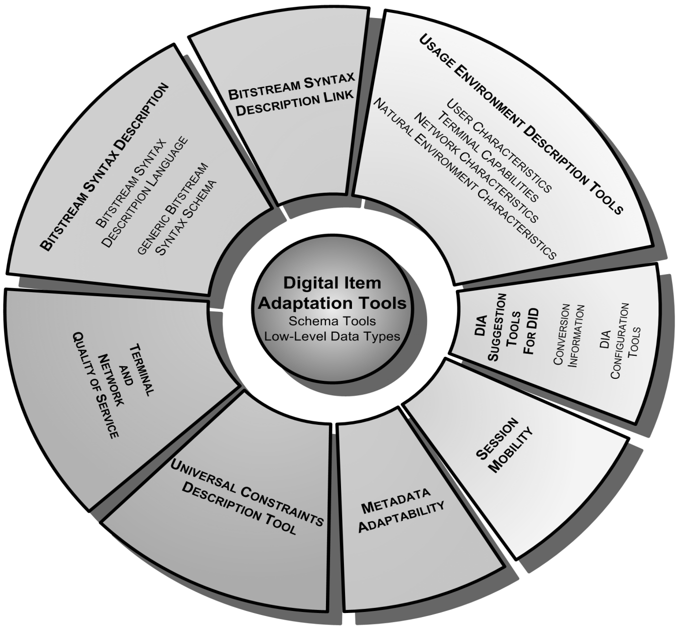

Bitstream Syntax Description
Bitstream Syntax Descritpion Language
Generic Bitstream Syntax Schema
Bitstream Syntax Description Link
Usage Environment Description Tools
User Characteristics
Terminal Capabilities
Network Characteristics
Natural Environment Characteristics
Digital Item Adaptation Tools
Schema Tools
Low-Level Data Types
DIA Suggestion Tools For DID
Conversion Information
DIA Configuration Tools
Terminal and Network Quality of Service
Session Mobility
Universal Constraints Description Tool
Metadata Adaptability

Figure 2.5: Organization of Digital Item Adaptation tools [11]

Usage Environment Description: The first major category is the *Usage Environment Description Tools*, which includes user characteristics, terminal capabilities, network characteristics and natural environment characteristics. These tools provide descriptive information about the various properties of the usage environment, which originate from Users, to accommodate, for example, the adaptation of Digital Items for transmission, storage and consumption. This thesis has contributed to this part of DIA on ConversionPreferences. Figure 2.6 shows a simple example for a DIA instance expressing some Usage Environment constraints concerning the terminal capabilities.

BSDLink: The second category is referred to as *BSDLink*. It provides the facilities to create a rich variety of adaptation architectures based on tools specified within this part of ISO/IEC 21000, ISO/IEC 21000-2, and ISO/IEC 15398 (MPEG-7 [12]) among others. This tool provides the facilities to link so-called steering description tools and *BSD* tools in a flexible and extensible way.

```
<DIA>
  <Description xsi:type="UsageEnvironmentType">
    <UsageEnvironmentProperty xsi:type="TerminalsType">
      <Terminal>
       <TerminalCapability xsi:type="DisplaysType">
         <Display id="primary_display">
           <DisplayCapability xsi:type="DisplayCapabilityType">
             <Mode>
               <Resolution horizontal="720" vertical="480"/>
             </Mode>
            </DisplayCapability>
         </Display>
         <Display id="secondary_display">
           <DisplayCapability xsi:type="DisplayCapabilityType">
             <Mode>
               <Resolution horizontal="176" vertical="144"/>
             </Mode>
           </DisplayCapability>
          </Display>
       </TerminalCapability>
       <TerminalCapability xsi:type="AudioOutputsType">
         <AudioOutput xsi:type="AudioOutputType">
           <AudioOutputCapability xsi:type="AudioOutputCapabilitiesType"
               lowFrequency="30" highFrequency="8000" numChannels="2"/>
         </AudioOutput>
       </TerminalCapability>
       <TerminalCapability xsi:type="UserInteractionInputsType">
         <UserInteractionInput>
           <UserInteractionInputSupport xsi:type="MicrophoneType"/>
         </UserInteractionInput>
         <UserInteractionInput>
           <UserInteractionInputSupport xsi:type="KeyInputType">
             <KeyInput href="urn:mpeg:mpeg21:2003:01-DIA-KeyInputCS-
                              NS:1">
                <mpeg7:Name xml:lang="en">PCKeyboard</mpeg7:Name>
             </KeyInput>
            </UserInteractionInputSupport>
           </UserInteractionInput>
           <UserInteractionInput>
             <UserInteractionInputSupport xsi:type="MouseType">
                 <Mouse buttons="2" scrollwheel="true"/>
</UserInteractionInputSupport>
           </UserInteractionInput>
        </TerminalCapability>
      </Terminal>
    </UsageEnvironmentProperty>
  </Description>
</DIA>
```

Figure 2.6: A DIA instance expressing the terminal capabilities

The extensible linking mechanism allows designing a rich variety of adaptation architectures, e.g., steered by resource adaptation tools such as *Terminal and Network Quality of Service* (explained later), *Usage Environment Description* tools, ISO/IEC 15398 tools, or steered by the User using the *Choice/Selection* mechanism provided by ISO/IEC 21000-2.

The *BSDLink* tool eases the referencing of information assets that can be used for this kind of Digital Item Adaptation, i.e., references to these assets are stored in the *BSDLink*. This description contains at least a reference to the *Bitstream Syntax Description* (*BSD*, see next paragraph) and a reference to the *BSD* transformation sheet. The *BSD* transformation sheet could be parameterized according to the desired adaptation. Additionally, the *BSDLink* may contain a reference to the steering description that governs the whole adaptation process and a reference to the actual resource, which is described by the *BSD*. For a complete walkthrough of an example use case, please refer to Annex B of ISO/IEC 21000-2.

Bitstream Syntax Description: These tools comprise the third major category of Digital Item Adaptation tools. A *BSD* describes the syntax – in most cases, the high level structure – of a binary media resource. Using such a description, a Digital Item resource adaptation engine can transform the bitstream and the corresponding description using editing-style operations such as data truncation and simple modifications. *BSD*-based resource adaptations are described in more details in next chapter.

Terminal and Network Quality of Service: The fourth category of tools is referred to as *Terminal and Network Quality of Service* (QoS). The tools specified in this category describe the relationship between QoS constraints (e.g., on network bandwidth or a terminal's computational capabilities), feasible adaptation operations satisfying these constraints and associated media resource qualities that result from adaptation. This set of tools therefore provides the means to trade-off these parameters with respect to quality so that an adaptation strategy can be formulated and optimal adaptation decisions can be made in constrained environments. In other words, *Terminal and Network Quality of Service* addresses the problem of selecting optimal parameter settings for media resource adaptation to satisfy constraints imposed by terminals and/or networks while maximizing the quality of service. Therefore the *AdaptationQoS* tool specifies the relationship between constraints, feasible adaptation operations satisfying these constraints, and possibly associated utilities (qualities). In this way, terminal and network QoS management is efficiently achieved by adaptation of media resources to imposed constraints.

Since a part of our work is based on the usage of *AdaptationQoS* tool, here, we provide a brief explanation on this tool. The *AdaptationQoS* tool provides the required information allowing the selection of optimal adaptation parameters. The tool has been designed in a modular way, i.e., the data and their relationships are structured in modules, which are the basic structural units for grouping the data. These modules can have one of the following representation formats: a) the *UtilityFunction*, which provides a restricted set of adaptation operation points in a list format to choose from, b) the *LookUpTable*, which is a matrix representation format, enabling selection by interpolation of an adaptation operation point and allowing extra information to be represented, and c) the *StackFunction*, which is a functional representation format. *IOPins* provide the interface of the modules. Each *IOPin* is a uniquely identifiable variable globally declared and referenced from within a module. The value of an *IOPin* may be any value, possibly constrained to a (continuous or discrete) value range specified by the *Axis* element in the *IOPin* declaration or to an externally retrieved parameter specified by the *GetValue* element in the *IOPin* declaration. Two or more modules referencing the same *IOPin* are linked, meaning that the referenced *IOPin*'s value should be the same for all these modules. Furthermore, an *IOPin* can also be referenced from outside *AdaptationQoS*, allowing the external retrieval of its attributed value, as is the case in, e.g., the *BSDLink* tool. The example of Figure 2.7 illustrates the top-level structure of the *AdaptationQoS* description. It contains one or more *AdaptationQoS* modules and one or more *IOPins* used for internal linking and external referencing.

```
<DIA>
        <DescriptionMetadata>
                <!-- optional description of Classification Scheme aliases -->
        </DescriptionMetadata>
        <Description xsi:type="AdaptationQoSType">
                <Module xsi:type="LookUpTableType">
                        <!-- description of chosen data representation -->
                </Module>
                <!-- other possible AdaptationQoS modules -->
                <IOPin id="QUALITY"
                        semantics="urn:mpeg:mpeg21:2003:01-DIA-AdaptationQoSCS-NS:3.1"/>
                <!-- other possible IOPins -->
        </Description>
</DIA>
```

Figure 2.7: AdaptationQoS example

The description generation entity can choose the most appropriate representation format for a module: *UtilityFunction*, *LookUpTable*, *StackFunction*. Finally, the (continuous or discrete) value range of the *IOPin*s can be further constrained by the *Universal Constraints Description* tool as specified in the next paragraph.

Universal Constraints Description: The *Universal Constraints Description* (UCD) tools form the fifth category of tools that enables the possibility to describe limitation and optimization constraints on adaptations. The *UCD* tool is based on establishing a mathematical abstraction where constraints are specified on variables representing resource and environment characteristics using values obtained either externally by XPath expressions from *Usage Environment Descriptions* or by a direct specification of numeric constants. When used in conjunction with the *AdaptationQoS*, these constraints are applied to *IOPins* in *AdaptationQoS*. Otherwise, they are applied to resource or environment characteristics indicated by their semantics. The UCD constraints can be provided by any DI User, i.e. content providers or the content consumers. The adaptation constraints can be specified not only on the resource as a whole but also differentiated with respect to individual units of the resource corresponding to logical partitionings such as GOPs, ROIs, Tiles, Frames etc. Such units are referred to as adaptation units. For each adaptation unit, the description is comprised by a set of limit constraints that specify Boolean expressions which must evaluate to true, along with optional optimization constraints that convey numeric expressions which are to be maximized or minimized within the feasible space of decisions satisfying the limit constraints. Together, the limit and optimization constraints specify a generic single or multi-objective optimization problem.

Metadata Adaptability: The sixth category is referred to as *Metadata Adaptability*. This tool specifies hint information that can be used to reduce the complexity of adapting the metadata contained in a Digital Item.

Session Mobility: For *Session Mobility*, the seventh category of tools, the configuration state information that pertains to the consumption of a Digital Item on one device is transferred to a second device. This enables the Digital Item to be consumed on the second device in an adapted way.

DIA Suggestion Tools for DID: Finally, the eighth category is referred to as *DIA Suggestion Tools for DID*, which provides recommendations – i.e. which kind of conversions are suggested for the described item or component – and provides information required for the configuration of a Digital Item Adaptation Engine.

Through this thesis, we have contributed to two parts of DIA: on conversion preferences and user interaction support in the first category, and on conversion description in the eighth category. We have also contributed to MPEG-21 DIA reference software.

2.3.8 MPEG-21 Part 8: Reference Software

This part of MPEG-21 describes reference software implementing the normative clauses of the other parts of MPEG-21 [13]. The provided information is applicable for determining the reference software modules available for parts of MPEG-21, understanding the functionality of the available reference software modules, and using the available reference software modules.

In addition to the reference software, available utility software that uses the reference software is also described. The utility software can assist in understanding how to use the reference software. Reference software will form the first of what is envisaged to be a number of systems-related specifications in MPEG-21.

2.4 Conclusions

The aim of this chapter was to briefly introduce the multimedia standard on which we based our work, i.e. MPEG-21 standard.

We saw that the objective of MPEG-21 is to define a complete set of standardized tools for different dimensions of a multimedia content adaptation framework. It was also explained that MPEG-21 standard is still under evolution.

A summary was then given on some parts of MPEG-21. Special attention was paid to the parts on which the presented work has been directly founded, and to which it has contributed. We also highlighted our contributions to MPEG-21 DIA.

A good understanding of MPEG-21 DIA and DID is necessary for the comprehension of this dissertation.

Chapter 3

SINGLE MEDIA ADAPTATION

Summary

The objective of this chapter is to generally familiarize the reader to the concept of multimedia content adaptation and to particularly explain the notion of single media adaptation and discuss the related issues and the existing approaches in this area.

Table of Content

Table of Illustrations

3.1 Introduction

Over the past several years, the development of information technology and growth of multimedia popularity as well as user demands have led to the creation of a vast variety of multimedia content and devices. Delivery of such a large diversity of multimedia content to different types of user devices and environments is one of the major challenges of a multimedia delivery chain. The content delivery chains need to transform the original content to an understandable and desired form, in order to satisfy the characteristics of the usage environment (user, device, network and etc.) of the multimedia, so that it would provide the end user with the optimum form of the content. This is the concept of "Multimedia Content Adaptation". We divide the concept of multimedia content adaptation into two major categories: adaptation of a single media and adaptation of a multimedia composition. This chapter deals with the former, i.e. single media adaptation.

In this chapter, we first provide some basic definitions. The principal concepts and elements of "Single Media Adaptation" are then dealt with. Next, we state the problems in this area and explain why we have chosen to build our work upon the framework of MPEG-21. A brief state of the art is then provided by discussing different existing approaches, methods and strategies.

3.2 Basic Definitions

To be able to define Single Media Adaptation, we first need to have a common understanding of "Multimedia Content Adaptation" and therefore an exact definition for it. "Multimedia Content Adaptation" is defined as the transformation of digital multimedia *content* from its initial state to a final state, in order to satisfy a set of constraints, named *context* of the usage. The final content could be obtained either directly from the original content, or by the usage of other existing alternatives. Satisfying a given constraint will result in changing the state of the original content, so that the usage (consumption, transmission, etc.) of the content in its final state does not cause any problem to the presence of that constraint. The mentioned set of given constraints are generally referred to as *context* of the usage. Therefore, a "Multimedia Content Adaptation" is in fact, the customization of a given *content* for a given *context*. The *context* includes different types of constraints such as user preferences, terminal capabilities and characteristics, network characteristics, author recommendations, cost functions, semantic considerations, etc. The set of context constraints is generally referred to as *usage environment characteristics* in MPEG-21.

We now define "Single Media Adaptation" as a multimedia content adaptation where the media is considered solely, i.e. as a mono media, out of any presentation scenario, and independently of the multimedia composition (scene) in which it exists, e.g. a video track, a paragraph of text, etc.

The term *multimedia content* could refer to either a single media or a multi-media composition, while another term, usually used for single media content is *media resource* (or *resource* in short). This shall not be confused with device resources that are understood as device capabilities in terms of display size, player capabilities, buffer size, etc.

To avoid any confusion, in this dissertation, we will always use the term "device resources" for device capacities, the terms "single media", "media" or "resource" for single media content, and the term "multimedia composition" or "multimedia scene" for rich-composed multimedia content.

3.3 Principal Elements

A multimedia content adaptation framework is constructed upon several entities and functions. The principal elements of a content adaptation framework are: original content, description of content, description of context, adaptation core which could be divided into two separate parts: decision making and resource adapting process, and finally the adapted content. Figure 3.1 shows how these elements are related to each other in the structure of a multimedia adaptation system.

In the following sections, we explain the concept of each of these entities, apart from original and adapted content, whose concepts are evident.

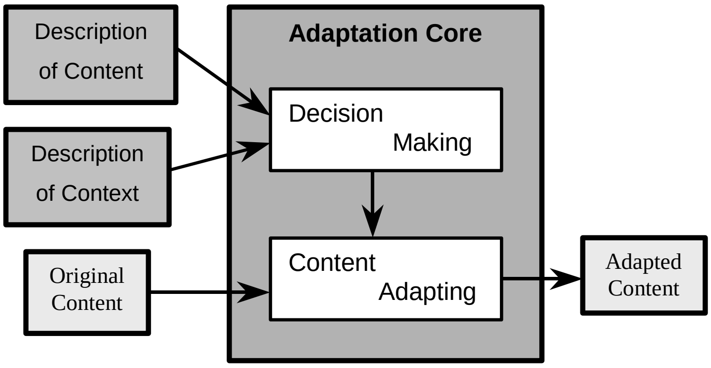


Figure 3.1: Principal elements of a multimedia content adaptation system

3.3.1 Description of content

The explicit description of the content is of very high importance in a multimedia content adaptation framework. To do its job the most properly, a content adaptation engine needs to have an exact and complete description of the original content to be adapted. It is true that some of the characteristics of the content could sometimes be directly extracted from the content itself, such as its modality or format. This however, may sometimes prove to be difficult. Also there are some characteristics of the content that must or might better be given explicitly. Among these, is semantic information such as key frames of a video, encoding parameters or other parameters such as the minimum resolution of a visual media, with which it is still logically visible.

Among W3C (World Wide Web Consortium) recommendations, RDF (see section 3.3.1.2) is designed for content description. Within MPEG (Pictures Experts Group) standards, it is MPEG-7 (see section 3.3.1.1) that provides complete description tools for content. MPEG-21 also defines descriptors for this means, which are based on MPEG-7 descriptors and are useful for adaptation process. In next paragraphs, a short introduction on MPEG-7 and RDF is provided.

3.3.1.1 MPEG-7

MPEG-7 is an ISO/IEC standard developed by MPEG (Moving Picture Experts Group). The MPEG-7 standard, formally named “Multimedia Content Description Interface”, provides a rich set of standardized tools to describe multimedia content [14]. Both human users and automatic systems that process audiovisual information are within the scope of MPEG-7.

MPEG-7 offers a comprehensive set of audiovisual Description Tools. Description Tools are the metadata elements and their structure and relationships that are defined by the standard in the form of Descriptors and Description Schemes to create descriptions. Descriptions are sets of instantiated Description Schemes and their corresponding Descriptors at the users will, which form the basis for applications enabling the needed effective and efficient access (search, filtering adapting and browsing) to multimedia content. MPEG-7 descriptions do not depend on the ways the described content is coded or stored. It is possible to create an MPEG-7 description of an analogue movie or of a picture that is printed on paper, in the same way as of digitized content. MPEG-7 allows different granularity in its descriptions, offering the possibility to have different levels of discrimination. The level of abstraction is related to the way the features can be extracted: many low-level features can be extracted in fully automatic ways, whereas high level features need (much) more human interaction.

The main elements of the MPEG-7 standard are:

- Description Tools: Descriptors that define the syntax and the semantics of each feature (metadata element); and Description Schemes, that specify the structure and semantics of the relationships between their components, that may be both Descriptors and Description Schemes

- A Description Definition Language that defines the syntax of the MPEG-7 Description Tools and allows the creation of new Description Schemes and, possibly, Descriptors that allow the extension and modification of existing Description Schemes;

- System tools that support binary coded representation for efficient storage and transmission, transmission mechanisms (both for textual and binary formats), multiplexing of descriptions, synchronization of descriptions with content, management and protection of intellectual property in MPEG-7 descriptions, etc.

The MPEG-7 descriptions of content may also include:

- Information describing the creation and production processes of the content.

- Information related to the usage of the content (copyright pointers, usage history, broadcast schedule).

- Information of the storage features of the content (storage format, encoding).

- Structural information on spatial, temporal or spatio-temporal components of the content (scene cuts, segmentation in regions, region motion tracking).

- Information about low level features in the content (colors, textures, sound timbres, melody description).

- Conceptual information of the reality captured by the content (objects and events, interactions among objects).

- Information about how to browse the content in an efficient way (summaries, variations, spatial and frequency sub-bands, ...).

```
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001"
      xsi:schemaLocation="urn:mpeg:mpeg7:schema:2001 Mpeg7-2001.xsd">
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="ImageType">
      <Image>
        <MediaLocator>
          <MediaUri>image.jpg</MediaUri>
          </MediaLocator>
          <TextAnnotation>
            <FreeTextAnnotation> Sunset scene </FreeTextAnnotation>
          </TextAnnotation>
          <VisualDescriptor xsi:type="ScalableColorType" numOfCoeff="16"
                             numOfBitplanesDiscarded="0">
            <Coeff> 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 </Coeff>
          </VisualDescriptor>
      </Image>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

Figure 3.2: Describing an image with MPEG-7

```
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001" xsi:schemaLocation="urn:mpeg:mpeg7:schema:2001 Mpeg7-
2001.xsd">
    <Description xsi:type="CreationDescriptionType">
        <CreationInformation>
            <Creation>
                <Title>an example web page</Title>
                <CreationCoordinates>
                    <Date>
                        <TimePoint>1999-08-16</TimePoint>
                        <Duration>P7D</Duration>
                    </Date>
                </CreationCoordinates>
            </Creation>
            <RelatedMaterial>
                <MediaLocator>
                   <MediaUri>http://www.example.org/index.html</MediaUri>
                </MediaLocator>
            </RelatedMaterial>
        </CreationInformation>
     </Description>
</Mpeg7>
```

Figure 3.3: Expressing the creation date of a document in MPEG-7

- Information about collections of objects.
- Information about the interaction of the user with the content (user preferences, usage history).

Figure 3.2 presents an MPEG-7 example describing some parameters of an image resource. Figure 3.3, shows how the creation date of a web document could be given in MPEG-7.

3.3.1.2 RDF

The Resource Description Framework (RDF) is a language for representing information about “resources” in the World Wide Web [15]. It is particularly designed for representing metadata about Web resources, such as the title, author, and modification date of a Web page, copyright and licensing information about a Web document, or the availability schedule for some shared resource. By generalizing the concept of a "Web resource", RDF can also be used to represent information about things that can be identified on the Web, even when they cannot be directly retrieved on the Web. Examples include information about items available from online shopping facilities (e.g., information about specifications, prices, and availability), or the description of a Web user's preferences.

Unlike MPEG-7, RDF does not define any semantics. RDF is based on the idea of describing things in terms of simple properties and property values. This enables RDF to represent simple statements as a *graph* of nodes and arcs representing the resources, and their properties and values. Consider the same example that we gave for MPEG-7, i.e. the example on creation date of a document. The RDF graph for the statement: *http://www.example.org/index.html* *`has a creation-date whose value is`* *August 16, 1999*, after assigning an URIref to the `creation-date` property is as shown in Figure 3.4. Figure 3.5 shows the RDF/XML syntax corresponding to the graph in Figure 3.4.

3.3.2 Description of context

The description of the context plays also an extremely important role in a multimedia content adaptation framework. A knowledge-based multimedia content adaptation framework needs to have exact information on the context (network, device, user, etc.) of the usage of the multimedia content in order to be able to provide the end user with the optimum form of the content.

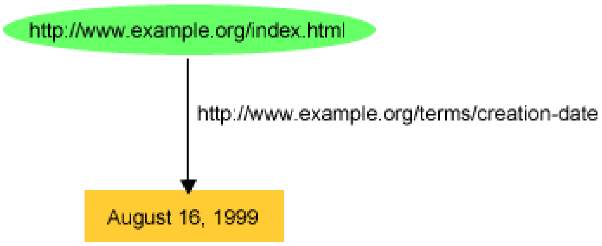

http://www.example.org/index.html
http://www.example.org/terms/creation-date
August 16, 1999

Figure 3.4: A graph describing a Web page's creation date

```
<?xml version="1.0"?>
 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
             xmlns:exterms="http://www.example.org/terms/">

   <rdf:Description rdf:about="http://www.example.org/index.html">
       <exterms:creation-date>August 16, 1999</exterms:creation-date>
   </rdf:Description>

 </rdf:RDF>
```

Figure 3.5: RDF/XML for the Web page's creation date

Different languages for context description have been used by different approaches. Each approach may evidently define its own context description language. Nevertheless, day-by-day, the interoperability between different multimedia adaptation systems is more desired. Therefore, definition and usage of a common context description language is highly valuable. That is why standardization bodies, namely, MPEG and W3C have defined description languages, such as MPEG-7, MPEG-21, RDF and CC/PP (see section 3.3.2.2). A summary on different metadata description standards is provided in [16].

3.3.2.1 MPEG-21 DIA

As briefly described in previous chapter, MPEG-21 DIA defines a complete set of description tools for describing the usage context constraints, known as Usage Environment Description (UED) tools within MPEG-21. The usage environment includes the description of User characteristics, terminal capabilities, network characteristics and natural environment characteristics. These various properties of the usage environment can be used for Digital Item Adaptation.

Users characteristics include general User information, usage preferences and history, presentation preferences, accessibility characteristics, mobility characteristics and destination. These descriptions could help achieve an appropriate and efficient personalization of multimedia content. Terminal capabilities are defined by a wide variety of attributes. Among them are codec capabilities, which include encoding and decoding capabilities, device properties, which include power, storage and data I/O characteristics, and input-output characteristics, which include display and audio output capabilities. The description of a terminal's capabilities is primarily required to satisfy consumption and processing constraints of a particular device. Network characteristics descriptors describe networks in terms of network capabilities and conditions, including available bandwidth, delay and error characteristics. These descriptions could be used for efficient and robust transmission of resources. DIA UED specifies descriptors for describing natural environment characteristics including location and time of usage of a Digital Item, as well as characteristics that pertain to audio-visual aspects. For the visual aspects, illumination characteristics that may affect the perceived display of visual information are specified. For the audio aspects, the description of the noise levels and a noise frequency spectrum are specified. This information can help the Digital Item Adaptation engine perform a better adaptation.

In this work, we have used DIA UED description tools for expression of the usage context constraints.

3.3.2.2 CC/PP

CC/PP, defined by W3C, stands for Composite Capabilities/Preference Profiles [17]. Pros and cons of using CC/PP as a basis for a context model and a context management system are discussed in [18]. CC/PP is based on RDF.

A CC/PP profile is a description of device capabilities and user preferences that can be used to guide the adaptation of content presented to that device and that user. Here "profile" does not refer to a subset of a particular specification, for example the CSS Mobile profile [19], but refers to the document(s) exchanged between devices that describe the capabilities of a device.

A CC/PP profile contains a number of CC/PP attribute names and associated values that are used by a server to determine the most appropriate form of a resource to deliver to a client. It is structured to allow a client to describe its capabilities. A set of CC/PP attribute names, permissible values and associated meanings constitute a CC/PP vocabulary.

It is anticipated that different applications will use different vocabularies; indeed this is needed if application-specific properties are to be represented within the CC/PP framework. But for different applications to work together, some common vocabulary, or a method to convert between different vocabularies, is needed. XML namespaces can ensure that different applications' names do not clash, but does not provide a common basis for exchanging information between different applications. Any vocabulary that relates to the structure of a CC/PP profile must follow the CC/PP specification.

CC/PP is designed to be broadly compatible with the earlier UAProf specification [20] from the WAP Forum. CC/PP is compatible with IETF media feature sets (CONNEG [21]) in the sense that all media feature tags and values can be expressed in CC/PP. However, not all CC/PP profiles can be expressed as media feature tags and values, and CC/PP does not attempt to express relationships between attributes. Figure 3.6 shows an example of CC/PP profile with three components.

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

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

    <ccpp:component>
      <rdf:Description
          rdf:about="http://www.example.com/profile#TerminalHardware">
        <!--  TerminalHardware properties here  -->
      </rdf:Description>
    </ccpp:component>

    <ccpp:component>
      <rdf:Description
          rdf:about="http://www.example.com/profile#TerminalSoftware">
        <!--  TerminalSoftware properties here  -->
      </rdf:Description>
    </ccpp:component>

    <ccpp:component>
      <rdf:Description
          rdf:about="http://www.example.com/profile#TerminalBrowser">
        <!--  TerminalBrowser properties here  -->
      </rdf:Description>
    </ccpp:component>

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

Figure 3.6: An example for CC/PP profile components in XML

One of the weak points of CC/PP is that it is implicitly designed to describe terminal capabilities and end user preferences that are only two parts of the context. CC/PP is therefore limited for description of other parameters of the context, such as network characteristics, natural environmental parameters, location parameters, etc. Another weak point of CC/PP is that the vocabulary for the description of the parameters of the context is not standardized and each application may define its own vocabulary. This avoids the interoperability at the level of the used vocabulary for the description of context parameters.

3.3.3 Resource adaptation core

Within a multimedia content adaptation system, adaptation core is defined to be the entity, which is responsible for deciding on the type of adaptation and then applying that adaptation to the content. Therefore, it could be said that the adaptation process is considered in two parts: decision making and resource adapting, which is done based on the result of the decision making.

3.3.3.1 Decision making

The decision making is defined to be the entity that makes the decisions on the type of the adaptation to apply and also the value of the needed parameters for that particular adaptation. The decision making needs to have access to the description of the context and the content. It also needs some knowledge on resource adaptors, i.e. it needs to know what the existing available resource adaptors are and what they need as parameters.

As an example, consider the case of a video resource which is to be adapted for a target device with image support and no support for video modality. Based on context and content description, the decision maker should decide on the type of conversion. To do so, the decision maker also needs to know whether or not the needed parameters for a particular conversion are given. Let us assume that based on the information that decision maker has on the content (type of content is video), and based on a part of information on context (terminal has image support) it finds the available resource adaptors as follows:

- Knowledge-based video-to-slideshow conversion. This conversion needs as parameter, the key frames to extract from the video.

- Random video-to-slideshow conversion. This conversion has no parameter.

- Knowledge-based video-to-image conversion. This conversion needs as parameter, the identity of the key frame, which is to be extracted from the video.

- Random video-to-image conversion. This conversion has no parameter.

If, because of bandwidth limitations (another part of context information), the video-to-slideshow conversions are not preferred, the decision maker will then have to choose between two different types of video-to-image conversions. If no key frame is given, then the choice of decision maker will be the random video-to-image conversion.

3.3.3.2 Resource adaptation tools

Resource adaptation tools are the single media adaptors. Based on how a resource adaptor changes a media, we define two different main categories for resource adaptation tools: resource adaptors which work on the basis of Direct Bitstream Modification and resource adaptors which work on the basis of Resource Conversion.

The former concerns resource adaptations tools that directly perform the modifications on the bitstream, and this, mostly by removing packages of data, while the later concerns adaptation tools, which completely convert the media. In other words, in Direct Bitstream Modification, the structure of the bitstream remains unchanged. This kind of resource adaptation is implicitly defined for *scalable* media. Resource adaptors under the category of Resource Conversion may (not necessarily) change the structure of adapted bitstream, by for example changing the format or the modality of the resource.

This Ph.D. thesis, in its first part, i.e. the work on single media adaptation, proposes a framework for adaptation by Resource Conversion in MPEG-21 (see Chapter 4). In the following sections, the basis and principle of these two main categories of resource adaptation are described in more details.

Direct Bitstream Modification

For scalable media resources, a variety of adapted versions can be retrieved from the original bitstream by performing simple editing-style operations such as data truncation and simple modifications. Media scaling of scalable resources is categorized under this kind of resource adaptation. It can, therefore, be said that the structure media (bitstream) has not been really *changed*.

In order to provide coding-format independency and therefore interoperability, it is desirable that a processor that is not aware of the specific bitstream coding format can be used for this task. For this, MPEG-21 DIA [11] has defined a generic approach by providing a method based on XML for manipulating bitstreams. The following paragraphs of this section provide a short introduction on this solution.

A binary media resource consists of a structured sequence of binary symbols, this structure being specific to the coding format. A bitstream is defined as the sequence of binary symbols representing this resource. XML is used to describe the high-level structure of a bitstream; the resulting XML document is called a

Bitstream Syntax Description (BS Description, BSD). This description, in most cases, will not describe the bitstream on a bit-per-bit basis, but rather address its high-level structure, e.g., how the bitstream is organized in layers or packets of data. With such a description, it is then possible for a resource adaptation engine to transform the BS Description, for example with an XSLT style sheet, and then generate back an adapted bitstream.

Figure 3.7 depicts the architecture of such a resource adaptation step. The architecture comprises the original *Bitstream* and its *Bitstream Syntax Description*, one (or more) *Bitstream Syntax Description Transformation*(s), the resulting *Transformed Bitstream Syntax Description*, the *Adapted Bitstream* and two processors, a BS Description generator and a bitstream generator.

The output produced in one adaptation step is an adapted bitstream and possibly an updated BS Description (not appearing on the figure) that correctly references the new bitstream.

The BS Description generator parses a bitstream and generates its BS Description. The bitstream and its BS Description are subject to the adaptation. An adaptation engine is assumed to determine the optimal adaptation for the media resource given the constraints as provided by the DIA descriptions. Based on that decision, if the resource is not pre-stored but needs to be derived by adapting an existing resource, then one (or several) BS Description transformation(s) is (are) selected to be applied to the input description. The result of these transformations is a transformed BS Description, which is the base for the generation of the adapted bitstream.

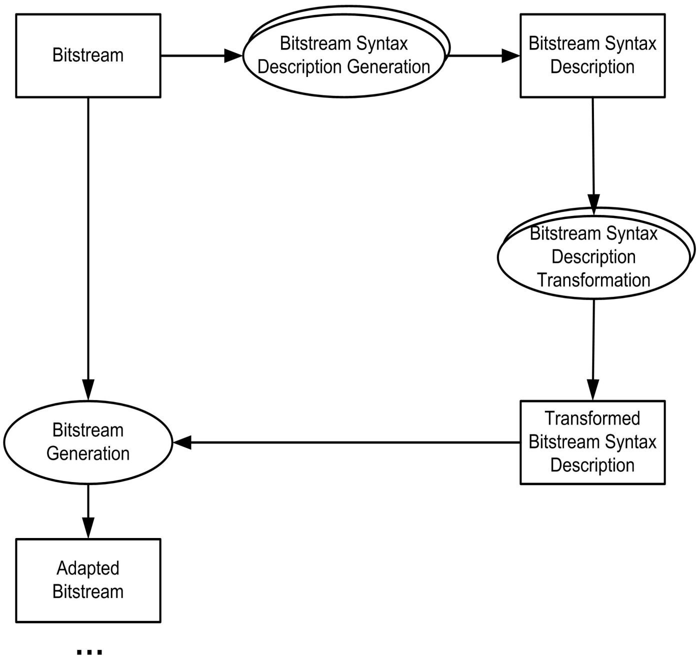

Bitstream
Bitstream Syntax Description Generation
Bitstream Syntax Description
Bitstream Syntax Description Transformation
Bitstream Generation
Transformed Bitstream Syntax Description
Adapted Bitstream
...

Figure 3.7: Architecture of a BSD-based resource adaptation

In order to provide full interoperability, it is necessary that a processor that is not aware of the specific coding format can nevertheless be used to produce a BS Description, and/or generate a bitstream from its description. For this, a new language, based on XML Schema, called Bitstream Syntax Description Language (BSDL), is specified by MPEG-21 DIA. With this language, it is then possible to design specific Bitstream Syntax Schemas (BS Schemas) describing the syntax of a particular coding format. These schemes can then be used by a generic processor to automatically parse a bitstream and generate its description, and vice-versa.

BSDL provides means for describing a bitstream syntax with a codec specific BS Schema. This requires an adaptation engine to know the specific schema. In some use cases this is not desired, for instance if the adaptation takes place on devices with constrained resources, e.g., in gateways and proxies. In these cases, a codec independent schema is more appropriate. Therefore, a generic Bitstream Syntax Schema (gBS Schema) is also specified. The normative gBS Schema introduces means to describe hierarchies of syntactical units and addressing means for efficient bitstream access.

The gBS Description provides an abstract view on the structure of the bitstream that can be used in particular when the availability of a specific BS Schema is not ensured. However, for transformations on gBS Descriptions, coding format specific information has to be included in attributes of the gBS Description.

For the BSDL case on the other hand, all coding format specific information can be provided by the BS Schema, which is common to all BS Descriptions following this schema. As a result, smaller descriptions can be obtained with BSDL. Furthermore, the flexibility provided by BSDL for designing BS schemas allows including additional application specific information into the BS Description.

Depending on the application requirements one or the other technology may be the most appropriate. For more information on this topic please refer to MPEG-21 DIA specification.

Resource Conversion

We define Resource Conversion to be a kind of digital resource adaptation, which is not of Direct Bitstream Modification Type. Adaptations of Resource Conversion type mainly change the structure of the adapted Bitstream. For example decoding, manipulating and re-encoding a media is considered to be an adaptation of Resource Conversion type. There exist quite numerous resource adaptors, which do not need the Bitstream Description Syntax (as defined in previous section). These resource adaptors are defined to be working on the basis of Resource Conversion.

We consider three main types of Resource Conversion: Transmoding, Transcoding and Transforming. These are detailed in the following sections. A Resource Conversion is conceptually defined to be an atomic process. For example a transmoding conversion is a pure modality change, i.e., changes only the modality parameter and is not supposed to change any other parameter.

Resource transmoding

We have invented the term *transmoding* for a particular type of resource conversion. Before giving the precise meaning of *transmoding*, we first need to come up with an exact definition for “modality” and a list of modalities. Modality has at least two meanings: on the perceptual level, modalities are tied to the five human senses, thus there is only “one” visual modality; on the structural level, there may be many modalities within one (perceptual) modality. For example, visual modalities include bitmap video, bitmap image, and two flavors of vector images: graphics2D and graphics3D. Within MPEG-21, the right meaning is the structural one, i.e. modality from the point of view of adaptation, since it dictates the types of algorithms, which can be applied to the resources. We consider a hierarchical schema for the Modality Classification Schema (*ModalityCS*), which we have proposed to MPEG-21 through a *Core Experiment*. In this classification schema, we have five principal modalities and some sub-modalities for some of them:

- Video
- Audio (audio2D, audio3D, speech)
- Image
- Graphics (graphics2D, graphics3D)
- Text

Here, graphics modality refers to vector graphics, while image modality refers to non-vector (bitmap) image. Slideshow is considered to be a particular type of graphics2D sub-modality.

MPEG-21 lately decided to use MPEG7 MDS *ContentCS* for this purpose, and since actual MPEG7 MDS *ContentCS* did not provide all proposed modalities, it was decided to propose an extension to this CS. Our proposed *ModalityCS* is in fact a subset of MPEG-7 *ContentCS*, which considers only single modalities and not multi-modalities, as considered in MPEG-7 *ContentCS*, such as audiovisual modality. More details on this are given in 4.2.1.5.

We now define *transmoding* as a type of modality-conversion adaptation, i.e. a digital resource adaptation that changes the modality of the original resource. When talking about adaptation by modality conversion, based on the point of interest, there exist two types of modality conversion: modality conversion on demand and modality conversion by alternative resource subsitution (in some documents these are also referred to as *online* and *offline* modality conversion). Transmoding refers to on-demand modality conversion of a multimedia resource where no alternative version of the resource in the desired modality is provided. Conceptually, a pure transmoding is not supposed to change other parameters of the original resource. For example a video-to-slideshow transmoding will result into a sequence of bitmap images. Any image format or resolution change of the transmoding-resulted images is not included in the transmoding. Therefore in order to obtain a slideshow of a smaller resolution and in JPG format, a BMP-to-JPG transcoding and then an image resizing transforming should be cascaded to this transmoding. This is of course a conceptual definition, i.e. there may be resource adaptation tools that implement such a whole adaptation process in one step.

Resource transcoding

We define *transcoding* as any format change of a resource while staying in the same modality, in such a manner that the content (essence) of the original resource is not changed. An example of Transcoding is the conversion of a GIF image to a BMP image representing the same picture. Conceptually, a pure transcoding is not supposed to change other parameters of a resource, such as the resolution of a visual media. Therefore, for example in order to convert a GIF image to a smaller BMP image, we consider that a GIF-to-BMP transcoding is first done and then the image is resized. This definition is conceptual and does not intend to impose implementation-related issues. There exist several free transcoding tools such as ImageMagick [22] or FFMPEG [23]. A great number of resource adaptation engines use such already existing transcoding tools.

Resource transforming

We define *transforming* as any parameter change of the original resource such as an image resizing or a text translation, while staying in the same modality and format. We consider three kinds of transforming:

➢ Changing the encoding parameters of an original media, for example re-encoding an MPEG-4 video at a lower bit-rate.

➢ Changing the presentation parameters of an original media, for example resizing or cropping a JPEG image to another smaller JPEG image, or translation of a text media to another text media in another language.

➢ Other parameter changes such as summarizing a video to another shorter video.

3.4 Problem statement

Numerous approaches have been adopted in the area of multimedia content adaptation. Most of these approaches are focused on the adaptation of single media (resource), while the adaptation of multimedia-composed scenes (presentations) has not been of great interest.

In this chapter we focus our interest on single media adaptation. None of the existing approaches in this area, proposes a complete solution that takes into account all the difficulties of the problem (see section 3.5). In this section we describe the issues of single media adaptation with special emphasis on MPEG-21-based solutions.

Different issues of building a general context-based media adaptation framework are discussed in [24]. The issues of multimedia scene adaptation are discussed in chapter 6.

In the area of single media adaptation, so far, adopted methods have been rather limited to certain particular aspects of the question. Often, the proposed solutions are incompatible and therefore cannot be integrated into a same system. Therefore it is very advantageous to work under an interoperable framework and that is the objective of standards such as MPEG-21.

In this section, first are explained the general problems of single media adaptation. Then the missing features of MPEG-21 are discussed.

3.4.1 General issues

The question of resource adaptation covers several aspects. Are listed here some of these aspects which have principal roles:

The diversity of client applications: terminals with different characteristics are part of a heterogeneous environment. Different terminals process the multimedia content differently and have different ways of accessing it. Therefore the different hardware and software specifications of the terminals should be considered in a resource adaptation system.

The diversity of the context and usage environment: a same terminal may be used to access, receive and play a multimedia resource under different contexts such as for example under different bandwidth limitations. This issue should be taken into account in a resource adaptation system.

The diversity of content: the multimedia content, in a resource adaptation system, may be very heterogeneous too. A complex content may be demanded by a simple terminal. In order to assure a suitable and meaningful content delivery, the adaptation system should be able to transform the content into an understandable form for the target device.

Adaptation tools: an efficient resource adaptation system should support the usage of different resource adaptation tools to use in different cases. It does not have to be limited to adaptation by switching between different already existing alternatives of the content. It happens quite often that either the author of a multimedia content have just produced one version of it, or existing alternatives of the content do not satisfy the underlying context.

3.4.2 Issues with MPEG-21

MPEG-21 DIA has a complete set of context describing tools, which covers the description of different parameters of the terminal and usage environment. As mentioned earlier in this document, MPEG-21 DIA has also description tools for describing content, which are based on MPEG-7 descriptors.

Concerning the support for different resource adaptation tools, MPEG-21 provides a complete support for a particular resource adaptation methodology based on direct bitstream modification. However, before our work, MPEG-21 did not provide a complete support for the usage of any other resource adaptation (Resource Conversion) in terms of the description of the recommended values for conversion parameters. This is where our work on Resource Conversion is situated within MPEG-21 DIA. As the result of our contributions DIA, an amendment on MPEG-21 DIA was established on December 2003, to consider the support of Resource Conversion within DIA [25]. We have actively participated in the establishment of this amendment and contributed to it during its evolution. Our approach for support of Resource Conversion in MPEG-21 is described in Chapter 4.

3.5 Existing approaches

In this section a brief state of the art on adopted approaches, methodologies and strategies in the area of single media adaptation is given. First are described the non-MPEG21 approaches. MPEG21-based approaches are then reviewed.

3.5.1 Non MPEG-21 approaches

In the area of media resource adaptation, numerous methodologies have been proposed as solutions to problems in the field. The best known of these are discussed here:

3.5.1.1 InfoPyramid framework

Smith et al. define an extensible multimedia content transmission system [26]. The system uses a new data model named InfoPyramid. The proposed methodology present a framework for managing and manipulating multimedia content composed of image, video, text and audio object. The model proposes at the same time strategies for single media adaptation, as well as adaptation of multimedia composed documents [27]. In this section, we limit our discussion on single media adaptation in InfoPyramid. The rich-multimedia adaptation in InfoPyramid is very limited and will be discussed in Chapter 6.

InfoPyramid manages the different variations of the media objects with different modalities and fidelities. The adaptation is then done either by selecting the existing variations of media objects (offline adaptation), or by on-the-fly manipulating of the media objects. InfoPyramid is used to represent content at multiple modalities and resolutions (fidelities) so that it can be rendered on a variety of devices. Figure 3.8 shows a simplified InfoPyramid for a video. Each media object is represented by a cell in the InfoPyramid. For example the third cell in the lower row corresponds to a high-resolution video. The cells above this one in the video column correspond to lower-resolution or compressed alternatives (lower fidelity) of the video. The cells to the left of video column correspond to image alternatives, and so forth for audio and text columns (different modalities). The cell in the bottom of the text column corresponds to a full-detailed body of text. The cells above it in the text column correspond to the summarized and compressed alternatives (lower fidelity) of the text body. The cells in the audio column correspond to different variations of the text rendered as audio (different modality), such as by text-to-speech conversation.

Resource adaptations in the InfoPyramid are limited to *translation* and *summarization* and some other limited conversions. The *translation* methods cover a few number of modality conversions such as text-to-audio or video-to-image. The *summarization* methods convert the resource within the same modality, but different *fidelity*, for example image compressing, text summarizing, and video abstraction generating. These methods can be cascaded.

Resource adaptation in InfoPyramid is done in two ways:

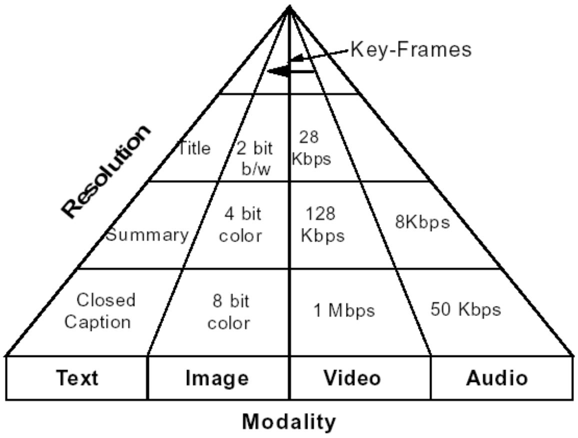


Figure 3.8: An InfoPyramid model for a video object

- Adaptive Delivery (offline Adaptation): in this type of adaptation, the customization is done by selecting the optimum already-existing variations of media objects.

In order to optimize the selection, InfoPyramid defines and uses *content values* for each media object. The content value scores can be based on automatic measures such as entropy, loss of fidelity or distortion. They can also be manually set in a subjective way. The content selection process selects the media object variations, which either maximizes the total content value, or minimizes the total data size. In addition to these two kinds of optimization, some limited device constraints could also be taken into account, for example display size of the target device.

- Online adaptation: in this type of adaptation, InfoPyramid is used as a transient structure in conversion of the media objects to the most appropriate modalities and fidelities. A dynamic routing system that exercises the trade-off between delay and distortion in selecting the conversion paths. In order to do this, for each conversion path L_k, the model defines an input-output signature: $L_k = (L_k^{in}, L_k^{out})$, where

$L_k^{in} = (M_k^{in}, F_k^{in}, D_k^{in}, S_k^{in})$,

$L_k^{out} = (M_k^{out}, F_k^{out}, D_k^{out}, S_k^{out})$, and

$M_k^{in, out}$ = input-output modality,

$F_k^{in, out}$ = input-output fidelity

$D_k^{in, out}$ = input-output data size

$S_k^{in, out}$ = input-output spatial-temporal size.

For example, a 50% text summarizer defines a signature of

$L_k^{in} = (M_k^{in}, F_k^{in}, D_k^{in}, S_k^{in})$,

and $L_k^{out} = (M_k^{in}, 0.5\ F_k^{in}, 0.5\ D_k^{in}, 0.5\ S_k^{in})$,

which indicates that the converter produces an output with the same modality as input, but with a 50% reduction in fidelity, data size and spatial size.

The system uses these descriptions in order to select the set of conversions to apply to the resource. The approach of conversion methods selection is very similar to the selection of resources. The principle is to suppose the existence of certain parameters and to define functions, which are to be minimized based on the values of these parameters. For example, for each conversion method, a *delay rate* parameter is associated. The value of the parameter gives an estimation of the time that the method needs to convert a resource of a given size.

The system then finds the set of conversion methods (paths) that minimize the total *delay rate* of all conversions. The methodology is very interesting, even though not standard-based.

3.5.1.2 CC/PP and RDF-based approaches

UAProf is an earlier implementation of RDF, which was specifically designed to describe capabilities of wireless devices. Unfortunately the model remains very limited to only one category of terminals and to WAP architecture, which suffers from numerous limitations of content accessing. In this section some approaches that are based on usage of CC/PP and RDF are overviewed:

In the area of single media adaptation, NAC (Negotiation and Adaptation Core) [28] proposes the solutions of on-line and off-line resource adaptation. It uses Universal Profiling Schemas [29] that defines extensions to CC/PP. NAC also uses the XSLT style sheets for adaptation of multimedia presentation; this is detailed Chapter 6.

There exist numerous other adaptation techniques using CC/PP and RDF, all of which, as discussed earlier, suffer from the limitations of CC/PP, unless extensions are used. Among these works we may recall [30] which provides only offline resource adaptation, [31] which proposes an approach for QoS-sensitive resource adaptation or [32] which proposes an annotation-based web content adaptation framework that also deals with resource adaptation (for more details on this, please refer to chapter 6). Some others works based on CC/PP are [18][33][34].

3.5.1.3 Adaptation based on augmentation

Susanne Boll et al. in [35] present their work on a cross-media (modality conversion) adaptation framework, which is based on offline adaptation. The essence of the methodology is to *augment* the content by some defined *augmentation models* and therefore to build several alternatives. It then chooses the alternative which best satisfies the constraints and substitute the current media with this alternative. The objective of the approach is the offline adaptation of multimedia presentations. To achieve this, media resources and also document fragments are first adapted on an augmentation-substitution basis. We will discuss this approach in more details in Chapter 6.

3.5.2 Approaches based on MPEG-21

During the last couple of years, as the on-going MPEG-21 standards progressed, several resource adaptation techniques were developed based on its framework. This section gives a brief review on some of these approaches.

3.5.2.1 BSDL and gBSD-based solutions

As described in details, in section 3.3.3.2, the basic idea of BSD is to define a generic method to allow the adaptation of different multimedia resources by a single, media resource-agnostic processor. This solution uses XML to describe the high-level structure of a binary media bitstream, to transform the description (e.g., by means of XSLT), and to construct the adapted media bitstream from the transformed description. Based on this concept, two complementary technologies, BSDL and gBS Schema, are defined within MPEG-21 DIA. The two technologies provide solutions for parsing a Bitstream to generate its XML description, for the generic structuring and marking of this description, and the generation of an adapted bitstream using its transformed description. The two technologies can be used as stand-alone tools.

S. Devillers et al. in [36] present how this basic BSD framework, initially developed for non-streamed content and suffering from inherent limitations and high memory consumption of XML-related technologies such as XSLT, can be advanced and efficiently implemented in a streaming environment and on constrained devices. Two different attempts to solve the inherent problems are described. The first approach proposes an architecture based on the streamed processing of SAX (Simple API for XML) [37] events and adopts STX (Streaming Transformations for XML) [38][39], a streaming transformation language for XML as an alternative to XSLT, whereas the second approach breaks a BSD up into well-formed fragments called Process Units (PUs) that can be processed individually by a standard XSLT processor. Gabriel Panis et al., in [40], focus on the gBS Schema and the joint BSDL/gBS Schema harmonized approach.

The European project ISIS provides also resource adaptations solutions based on BSD for adaptation of *scalable* Video, Audio and Graphics content [41]. As a part of ISIS project, Concolato proposes usage of MPEG-21 for adaptation of BIFS Graphics content [42]. The approach realizes a BIFS encoder that permits to encode a 2D Graphics content in a *scalable* way. The concept of the work is that the BIFS stream representing the animation is encoded in layers. The basic layer contains the BIFS commands permitting to visualize the most low-quality version of the content. The improving layers contain BIFS commands of element addition or insertion, permitting to improve the quality. These commands provide the possibility of adaptation of the content but need higher bit rates. The approach then uses the gBSD solution for describing and then adapting the bitstream of the Graphics content for different bit rates. Since the presented work has been partly done under the framework of this project, ISIS objectives and achievements will be further discussed in next chapter.

Martinez et al. present an MPEG-21-based architecture for context-aware multimedia content adaptation [43]. They implement scalable media adaptation tools by customizing BSD, as well as real-time media modification tools for non-scalable content.

3.5.2.2 Other MPEG-21-based approaches

Research works, contributing to MPEG-21, have been done to find necessary support for offline adaptation by modality change, called offline modality conversion [44]. The proposed solutions are limited to the support of static decision taking and adaptation and do not permit to take into account the metadata that are needed for online adaptation.

D. Jannach et al., in [45], propose an approach for multi-step adaptation of multimedia resources. The methodology is based on semantic descriptions of transformation steps, which are exploited by a classical state-space manner. The proposed framework relies on descriptions of the resource itself (MPEG-7), the usage environment of the resource (MPEG-21), as well as declarative descriptions of the transformation tool. The implemented prototype that is a simple video resource adaptation engine employs a knowledge-based engine for finding and executing the needed adaptation sequences.

The adaptation module of [43] uses also transcoding, transmoding and real-time media modification tools. These adaptations are performed based on content and context descriptions, respectively given by MPEG-7 and MPEG-21 DIA descriptors. The approach also uses some transcoding hints, expressed in MPEG-7.

3.6 Conclusion

This chapter described the basic notion of single media adaptation, which is also called resource adaptation. Principal elements of a resource adaptation system, as well as the issues of such system were discussed. We categorized resource adaptation into two main categories: Direct Bitstream Modification, and Resource Conversion. We also discussed the standardized frameworks for media adaptation and the different existing approaches based on them.

In the area of resource adaptation, we have based our work on Resource Conversion in the MPEG-21 framework; the rationale behind this choice and the theory of work is described in next chapter.

Chapter 4

RESOURCE CONVERSION: OUR METHODOLOGY

Summary

This chapter introduces the theory of our methodology for a Resource Conversion framework in MPEG-21 DIA. We will describe our approach in two phases.

Table of Content

Table of Illustrations

4.1 Introduction

In the area of resource adaptation, we have based our work on Resource Conversion in MPEG-21 framework; the rationale behind this choice is explained in the following paragraphs.

4.1.1 Why MPEG-21?

Chapter 2being dedicated to the MPEG-21 standard, we will not go through the advantages of MPEG-21 over other frameworks in this section. In summary, in comparison to other standardization frameworks (such as CC/PP & RDF), MPEG-21 has proven to be a more complete framework, by providing answers to questions on different aspects of a multimedia delivery chain such as usage environment description, media adaptation tools, digital right management, etc.

4.1.2 Why Resource Conversion?

We previously explained (section 3.4.2) that MPEG-21 DIA, at this time, does not have a complete support for Resource Conversions. That is why, on the subject of single media adaptation, we have chosen to work in the area of Resource Conversion in MPEG-21. In this chapter we describe a framework for Resource Conversion in MPEG-21.

We first describe the theory of the first phase of the work that was developed based on “simple”, “static” and “hard” description of conversion-related information. This phase of the work was performed in two steps; the first step dealt with only Resource Conversions of transmoding type and was performed under the framework of ISIS project. The second step was based on the same methodology, but brought an enhancement to the first step by considering other Resource Conversion types.

Next, the approach of the second phase of the work is described. The idea is to use the DIA *AdaptationQoS* tool for providing “non-static” or “soft” description of conversion parameters. This helps achieving a more intelligent decision-making. The research and development of this phase was shared between the author and other colleagues at ENST. Along both two phases of the work, several contributions were made to MPEG-21 DIA.

4.2 First phase: Static description of conversion-related information

4.2.1 Hinted transmoding

Transmoding has an important role in the content adaptation process of a Universal Multimedia Access system. This is because adaptation by transmoding may be the only solution for many frequent multimedia content adaptation use cases. Nevertheless, so far, the focus of most research works on content adaptation has been limited to either adaptations that do not change the modality of the media (such as BSD-based adaptation, transcoding and transforming), or adaptations that substitute a media with an already-existing alternative in another modality and the needed support for them. As a result, support for transmoding has not been sufficiently investigated.

In this section, we introduce, our proposed descriptors for the expression of modality conversion preferences, as well as our proposed description tool for online transmoding within MPEG-21 DIA. This latter is the very first step of our work on the subject of Resource Conversion. The usefulness and necessity of this description tool is then proven through several use cases. Also the objectives of the ISIS European project to which this approach has been integrated, will be described.

4.2.1.1 Conversion preferences

On the subject of modality conversion preferences within DIA, in a joint research work with ICU, we defined a set of descriptors for the expression of the preferences (of author, user, etc.) regarding the conversion of modality. These descriptors have been proposed to MPEG-21 during a *Core Experiment*, and have been finally promoted to DIA under the name of *ConversionPreference*. The original objective of the work was the expression of preferences on different modality conversions (or transmoding), but it was then extended to cover the expression of preferences on all types of conversion. When several conversions or transmodings are possible for a resource, these descriptors help to make a choice between them.

As described in the previous chapter, in the resource adaptation process, various types of conversions may be carried out when a terminal or network cannot support the consumption or transport of a particular modality or format. For each resource, there may exist many conversion possibilities. Given that a User will have preference for certain modalities or formats over others, the role of the *ConversionPreference* tool is to enable Users to specify these preferences to guide the conversion of resources. The User has two ways for identifying resources. The general way is applied to all resources of a certain original modality or format, and the specific way is applied to the specific resources in which the User is interested.

User preference for a conversion is divided into two levels, qualitative and quantitative. First, a User can specify the relative *orders* for possible conversions of each original modality or format. The orders help an adaptation engine find the destination modality or format when the original one needs to be converted under a given constraint. Second, a User can further specify the numeric *weights* for conversions, which can be considered as a User’s QoS preferences on the conversion of one modality or format to another.

In the example of Figure 4.1, the User wants to apply generally some conversion rules to video resources, where it is most desired that the videos be retained if possible (i.e. order of video-to-video is 1). However, if videos must be converted, they should be converted to audios first (order of video-to-audio is 2). And if not possible, they may be converted to image or text. By means of *SpecificResourceConversions*, the User can express his/her preferences on the conversion of a specific resource of video.

Note— In our work on hinted transmoding, we use *ConversionPreference***,** only for the expression of modality conversion preferences of the end user. The preferences of the author of a particular content are then given in a *Transmoding* descriptor (section 4.2.1.4), associated to that content.

4.2.1.2 Transmoding table

All online transmodings for the defined modalities in Chapter 3are illustrated in Figure 4.2. The white boxes are the transmodings, which seem to make sense, be useful and have generic parameters.

```
<DIA>
   <Description xsi:type="UsageEnvironmentType">
     <UsageEnvironmentProperty xsi:type="UsersType">
        <User>
          <UserCharacteristic xsi:type="ConversionPreferenceType">
            <GeneralResourceConversions>
               <Conversion order="1" weight="1.0">
                  <From href="urn:mpeg:mpeg7:cs:ContentCS:2001:4.2">
                    <mpeg7:Name>Video</mpeg7:Name>
                  </From>
                  <To href="urn:mpeg:mpeg7:cs:ContentCS:2001:4.2">
                    <mpeg7:Name>Video</mpeg7:Name>
                  </To>
               </Conversion>
               <Conversion order="3" weight="1.0">
                  <From href="urn:mpeg:mpeg7:cs:ContentCS:2001:4.2">
                    <mpeg7:Name>Video</mpeg7:Name>
                  </From>
                  <To href="urn:mpeg:mpeg7:cs:ContentCS:2001:4.1">
                    <mpeg7:Name>Image</mpeg7:Name>
                  </To>
               </Conversion>
               <Conversion order="2" weight="1.0">
                  <From href="urn:mpeg:mpeg7:cs:ContentCS:2001:4.2">
                    <mpeg7:Name>Video</mpeg7:Name>
                  </From>
                  <To href="urn:mpeg:mpeg7:cs:ContentCS:2001:1">
                    <mpeg7:Name>Audio</mpeg7:Name>
                  </To>
               </Conversion>
               <Conversion order="4" weight="1.0">
                  <From href="urn:mpeg:mpeg7:cs:ContentCS:2001:4.2">
                    <mpeg7:Name>Video</mpeg7:Name>
                  </From>
                  <To href="urn:mpeg:mpeg7:cs:ContentCS:2001:5">
                    <mpeg7:Name>Text</mpeg7:Name>
                  </To>
               </Conversion>
             </GeneralResourceConversions>
          </UserCharacteristic>
        </User>
      </UsageEnvironmentProperty>
   </Description>
</DIA>
```

Figure 4.1: Expression of general conversion preferences

IN / OUT		Audio			Image	Video	Graphics		Text
		Speech	Audio2D	Audio3D			Graphics2D	Graphics3D	
Audio	Speech	▨	?	?					?
	Audio2D	?	▨	?					
	Audio3D	?	?	▨					
Image		√*			▨	√	√	√	√
Video		√				▨		√	√
Graphics	Graphics2D	√				√	▨	√	√
	Graphics3D	?						▨	?
Text		√	**	**	**	**	**	**	▨

*This transmoding could be done in different ways, one way would be Speech to Text + Text to Image, the other way would be Speech to sign language.

**This transmoding does not make sense as an online conversion.

Figure 4.2: Transmoding table

Generic parameters are not algorithm-dependent. The grey boxes are other transmodings. There are some transmodings, which have not been investigated in this work and need to be investigated by related modality experts; the related boxes contain a question mark.

4.2.1.3 Some use case scenarios

A use case scenario for video-to-image and video-to-slideshow (graphics2D) transmoding is the case of an original video resource for which the provider has not provided the image or slideshow alternatives but has given his intentions concerning the transmoding to image and slideshow through the transmoding descriptor. For the video-to-slideshow conversion, the provider has determined which frames to use as slides, the slides transitions and durations.

A concrete example of this scenario is as follows. The original video resource is a movie trailer, which is designed for ADSL but is not suitable to be transmitted over cellular networks. In the corresponding REL expression, video-to-slideshow transmoding is authorized. The optimal adaptation decision, which is taken on the basis of satisfying all related constraints, is transmoding to slideshow. Once this decision is taken, to perform the action of transmoding, the needed metadata is fetched from the DIA descriptor. For example, the display resolution and the format of the image are fetched from *TerminalCapabilities* descriptor. Any constraint on modality or modality conversion priorities is fetched either from the *ConversionPreference* descriptor or *PresentationPriorityPreferences* descriptor. The available bandwidth needed to determine the compression level of the image sequence, is fetched from the *NetworkCharacteristics* descriptor. And finally the list of the key frames and the relative durations is fetched from the transmoding descriptor.

A use case scenario for text-to-image transmoding could be as follows. The resource to be adapted is a text resource, for example, a message in Persian language (or with any special font, family and style). The device is simple and does not have the Persian font support, but supports image content. In this case, if authorized in REL expression, a solution is to transmode the text to an image. The transmoding process requires font characteristics (to render the text into an image), colour information, etc. The same use case may be given for a text-to-video transmoding, when the text does not fit into the screen of the terminal (whose size is extracted from *TerminalCapabilities* descriptor), and thus needs to be scrolled. This transmoding process additionally requires the author preferences on the type of scrolling.

4.2.1.4 Transmoding description tool

We present a description tool for producing the metadata on the adaptation of a resource by transmoding. We are interested in the point of view of the author (provider) of the resource. Nevertheless, through this descriptor, any peer in the adaptation chain can express his preferences on the parameters of a specific conversion. The author, based on his knowledge on the resources, may wish to express his preferences on transmoding(s) of a particular content and provide some transmoding hints to facilitate, guide or enable the adaptation; this can be done by means of the proposed descriptors.

Through the proposed transmoding descriptor, are expressed the transmoding preferences and hints. Transmoding preferences are the preferred priorities of different transmodings of the same content, and the preferred or suggested quality of each transmoding. The transmoding hints include the descriptions of the most general transformation parameters of a certain transmoding and for a particular resource. The considered parameters are generic, in other words they are based on no particular underlying algorithm. A transmoding descriptor is not supposed to contain any transmoding-related right or permission statement. The corresponding right and authorization statements will be given through REL.

For the description of generic transmodings parameters, we have considered some of the meaningful transmodings in the above table. For each of these, we give the related and generic parameters on which the provider may wish to impose or express his preferences or recommendations. As extensions to DIA schema, XML schemas have been defined for these parameters; Annex A provides the syntax and semantics of these schemas. During several contributions and *Core Experiments*, we proposed these schemas to MPEG-21 DIA. They were not promoted to the standard under the form described here, but gave birth to an activity (in which we participated) on Resource Conversion descriptors in DIA that opened an amendment to DIA [25]. Other transmodings should be investigated by related modality experts. The structure of the proposed transmoding descriptor is extensible, meaning that other transmodings and their generic parameters may at any time be added to it. Our considered set of transmodings is as follows:

- Summarization (video-to-image, video-to-slideshow (graphics2D) and graphics-to-image transmoding)
    - Parameters: resolution, and slide parameters: time code, duration.
- Text Visualization (text-to-video, text-to-image and text-to-graphics transmoding)
    - Parameters: resolution, text color information, background color, text color, and font characteristics: (type, style, size), presentation (motion) style (scrolling, subtitle).
- Speech Visualization (speech-to-text, speech-to-image, speech-to-video and speech-to-graphics transmoding)
    - Parameters: language, resolution, font characteristics (type, style, size), background color, text color.

For the expression of transmoding preferences, we define two notions:

- Quality: The quality attribute gives a hint to the underlying algorithm about the quality of destination/transmoded resource. For a particular transmoding, if the algorithm is simple and with few clearly required and generic parameters, i.e. all those generic parameters could be mentioned in the descriptor of this particular transmoding, then the quality attribute will not be used. But usually, for a particular transmoding, this is not the case, i.e. there are different possible algorithms, with different sets of parameters that cannot be summarized into a generic descriptor. In such cases, as all the parameters cannot be expressed in the descriptor of this particular transmoding, then the provider can use the quality attribute. This attribute gives a hint to the algorithm about what quality is wanted. It can be mapped to a quality-related parameter of the algorithm. We believe that although the quality attribute is not really reproducible from one chosen algorithm to another, it is better to mention it since it gives a useful indication to the adapting peer. For instance, for all the transmodings of-which the transmoded resource is video, the quality attribute may express the quality of video in terms of the number of frames per second.

- Priority: This attribute is used to set priorities between different transmodings of the same content. For example an author, who has expressed his preferred parameters for video-to-image and video-to-slideshow transmodings of an original video, may wish to express that he prefers his video resource to be transmoded into slideshow rather than image.

Figure 4.3 and Figure 4.4 present two simple examples for video-to-image and image-to-text transmodings. The preferred (or suggested) priority, quality and parameters of each transmoding are expressed.

```
<Transmoding quality="1.0" priority="1">
     <TransmodingParameters xsi:type="VideoSummarizationParametersType">
          <To href="urn:mpeg:mpeg7:cs:ContentCS:2001">
             <mpeg7:Name>Image</mpeg7:Name>
          </To>
          <Slide importance="hight">
             <mpeg7:MediaTimePoint>T01:14:30:12F24</mpeg7:MediaTimePoint>
          </Slide>
     </TransmodingParameters>
</Transmoding>
```

Figure 4.3: An example of transmoding descriptor for video-to-image transmoding

```
<Transmoding quality="1.0" priority="1">
 <TransmodingParameters xsi:type="TextVisualizationParametersType">
   <To href="urn:mpeg:mpeg7:cs:ContentCS:2001">
        <mpeg7:Name>Image</mpeg7:Name>
   </To>
   <FontParameters fontSize="20">
        <FontStyle>BOLD</FontStyle>
        <FontFamily>ARIAL</FontFamily>
   </FontParameters>
   <Resolution horizontal="300" vertical="400"/>
   <TextMotionStyle href="urn:mpeg:mpeg21:cs:TextMotionStyleCS:2003">
        <mpeg7:Name>TickerTape</mpeg7:Name>
   </TextMotionStyle>
   <TextColorInformation foregroundColor="0.2 0.5 0.6" backgroundColor="0 0 0"/>
  </TransmodingParameters>
</Transmoding>
```

Figure 4.4: An example of transmoding descriptor for image-to-text transmoding

4.2.1.5 Description of the resource

The description of the original resource is needed to help the process of adaptation decision-making. As only transmoding-type adaptations are considered here, it is sufficient to express only the modality of the original resource. In other words, to choose which of the available transmodings should be applied to the resource, the decision-making engine, in addition to the modality conversion preferences and some times the transmoding parameters, needs to know the modality of the resource as well.

As an example, we consider an original video resource. The target device has no video support, but supports image, graphics and text resources. An alternative version of this resource in text modality is provided in the DID. The author has also given the transmoding parameters for a video-to-slideshow conversion; key frames, durations, transitions and so on. To decide on the type and parameters of the transmoding, the decision-making algorithm first needs to know the original modality of the resource, it then verifies what transmoders are available for this original modality. We consider that two transmoders are available: text-to-image and text-to-slideshow. In the modality conversion preferences, the latter has a higher priority than the former and video-to-text conversion (i.e. the choice of the already-given text alternative), it is therefore chosen as a first option. The decision-making then verifies if the necessary parameters for this transmoding are given. If so, it sends the transmoding type and its parameters to the resource adaptor engine.

Even thought the modality of a resource may be obtained from the MIME type attribute, we have decided to use explicit descriptors for the expression of the modality. We first used our defined *ModalityCS* (as defined in 3.3.3.2) for this purpose, which is in fact a subset of the related MPEG-7 *ContentCS*. However, when MPEG-21 DIA decided to use the MPEG-7 *ContentCS* for this, we decided to stay harmonized with it. Figure 4.5 shows how *ContentCS* is used to describe the modality of a resource.

```
<mpeg7:Content href="urn:mpeg:mpeg7:cs:ContentCS:2001">
       <mpeg7:Name>Image</mpeg7:Name>
</mpeg7:Content>
```

Figure 4.5: Usage of MPEG-7 ContentCS for describing resource modality

4.2.1.6 Integrating transmoding-related descriptors into DID

In this part we intend to show how the transmoding-related descriptors, i.e. modality conversion preferences, transmoding parameters and resource modality description, could be placed in a DID instance. We have envisaged two different ways for this, but before going through these two expression methods, let us define two basic concepts: CDI and XDI.

XDI: conteXt Digital Item. An XDI is a DI or a fragment of a DI that contains the context information. It is expressed in DIDL and contains the expression of context, commonly given in DIA UED (Usage Environment Description) descriptors. It does not contain or does not reference any media resource.

CDI: Content Digital Item. A CDI is a DI or a fragment of a DI that contains the content and the content-related information. It is expressed in DIDL and contains (references to) the media resources and provides their related information, such as their description.

Having defined these two concepts, we now describe two ways for placing transmoding-related descriptors within a DID instance.

Expression of virtual resources: The suggested parameters of a certain type of transmoding for a particular resource are expressed as a *Selection* element within a DID *Choice* element in a CDI. A *Component* element related to the above *Selection* element will give the provider the possibility to express his transmoding (recommended) parameters. For this purpose, the transmoding descriptors will be given within a *Statement* element of the *Descriptor* element of this *Component* element. The corresponding transmoding *Component* element is a virtual component since it does not include any resource of the destination modality. It contains, in its *Resource* element, the original resource and in its associated *Descriptor*/*Statement* element, the description of the original resource and also the *Transmoding* descriptor. The *Transmoding* descriptor provides the author with a way for expressing his/her suggested parameters and preferences on a transmoding. An example of the DID instance of such CDI is given in Figure 4.6.

Note— We have defined *Transmoding* descriptor as an extension to DIA and that is why it is appeared as a DIA element. In Figure 4.6, *Transmoding* descriptor is put directly in the DIDL *Statement* element, the reason is that this phase of the work was developed before the opening of DIA Amendment 1, which defined a placeholder within DIA for expression of conversion-related information.

Although *ConversionPreference* descriptor is meant to specifically or generally express the conversion preferences of any MPEG-21 User, here we use it only for expressing the general preferences of the end user on modality conversions. The author preferences for each particular Resource Conversion, is given in CDI by *Transmoding* descriptor. The *ConversionPreference* will be given in a XDI. Figure 4.7 shows how general conversion preferences of the end user are expressed in an XDI.

Simple *Descriptor/Statement*-based expression: This way of integration of transmoding-related information into a DID instance is quite similar to the one based on the concept of virtual resources. The only difference is that the *Transmoding* descriptor is directly attached to the corresponding *Resource* element containing the original resource. No virtual resource is then required. Figure 4.8 represents this idea.

```
<DIDL>
 <Item>
   <Choice >
     <!-- video -->
     <Selection select_id="1"/>
     <!-- audio -->
     <Selection select_id="2"/>
     <!–transmoding to image -->
     <Selection select_id="3"/>
   </Choice>

<!–the provided video resource -->
   <Component>
     <Condition require="1"/>
     <Resource ref="video" />
   </Component>
<!–the provided text version -->
   <Component>
     <Condition require="2"/>
     <Resource ref="text" />
   </Component>
<!–no image version of the resource is provided -->
   <Component>
     <Condition require="3"/>
     <Descriptor>
               <Statement>
                      <dia:Transmoding >
<!– here goes the transmoding descriptors expressing the recommendations and hints of the author
concerning the video-to-image transmoding -->
                      </dia:Transmoding >
               </Statement>
     </Descriptor>
     <Descriptor>
               <Statement>
 <!– here goes the description of the original resource modality -->
                      <mpeg7:Content href="urn:mpeg:cs:ContentCS:2001">
                              <mpeg7:Name>video</mpeg7:Name>
                      </mpeg7:Content>
               </Statement>
     </Descriptor>

     <!–here is indicated the resource of the initial modality -->
     <Resource ref="video" />
    </Component>
  </Item>
</DIDL>
```

Figure 4.6: A CDI showing the concept of virtual resources and usage of transmoding descriptors in DID

```
<DIDL>
  <Container>
    <Descriptor>
      <Statement>
<!-- context information given by UED containing conversion preferences, for example: -->
          <dia:DIA>
            <dia:Description xsi:type="UsageEnvironmentType">
              <dia:UsageEnvironmentProperty xsi:type="UsersType">
                <dia:User>
                  <dia:UserCharacteristic xsi:type="ConversionPreferenceType">
                    <dia:GeneralResourceConversions  .../>
                  </dia:UserCharacteristic>
                </dia:User>
              </dia:UsageEnvironmentProperty>
            </dia:Description>
          </dia:DIA>
      </Statement>
    </Descriptor>
       .  .  .
  </Container>
</DIDL>
```

Figure 4.7: An example of an XDI containing conversion preferences

```
<DIDL>
 <Item>
    <Component>
       <Descriptor>
          <Statement>
              <dia>
                    <!– here goes the transmoding descriptors expressing the recommendations
                    and hints of the author concerning the video-to-image transmoding -->
              </dia>
          </Statement>
       </Descriptor>
       <Descriptor>
          <Statement>
              <!– here goes the description of the original resource modality -->
              <mpeg7:Content href="urn:mpeg:cs:ContentCS:2001">
                     <mpeg7:Name>video</mpeg7:Name>
             </mpeg7:Content>
          </Statement>
       </Descriptor>

      <!–here is indicated the resource -->
      <Resource ref="video" />
    </Component>
  </Item>
</DIDL>
```

Figure 4.8: Association of *Transmoding* descriptor to corresponding *Resource*

We have used and implemented both of these expression ways; nevertheless the latter is preferable since the concept of virtual resource is not formally defined within MPEG-21 DIA.

4.2.1.7 ISIS

ISIS, Intelligent Scalability for Interoperable Services, is an IST European project. The goal of ISIS is to design, implement and validate a multimedia framework that allows for audio-visual content to be created once and adapted to a wide range of service scenarios or contexts by customization to different transport characteristics and end device capabilities as well as personalization to end-user preferences.

The envisaged transport mechanisms are those of fixed, wireless and mobile Internets. The end-devices include the variety of personal computers and handheld terminals connected to these networks. To this end, the following objectives were pursued: 1- Development of tools for coding, managing, delivering and playing back scalable audio, video and graphics contents; 2- Integration in a consistent end-to-end production, delivery and consumption chain; 3- Validation and demonstration of scalability functionality through a prototype application.

The innovation of the ISIS project is a multimedia representation format and management tool that allow for audio-visual content to be created once and adapted to a wide range of service scenarios by customization to different transport characteristics, end-device capabilities, natural environment conditions as well as personalization to end-user preferences. This new technology will:

➢ Enable content creators to provide their content for many different usage environments without the need to create or store many different versions of the same content;

- Enable service providers to provide their clients with the optimum quality and with customized content with respect to the users' current usage environment.

These objectives represent a significant progress compared to today's situation where:

- Content and service providers have to develop, store and maintain multiple versions of the same content. It is then not always possible to choose the correct version to address all relevant user communities and relevant usage scenarios;

- End users have limited access to the content or very reduced QoS while accessing the content due to a mismatch between their available terminal and network resources and the content formats stored on the servers.

To achieve its objectives, the project has analyzed, specified and developed a complete end-to-end production chain, which constitutes significant innovation over the state-of-the art in major technological areas of work:

1. Development of innovative content representation formats with an emphasis on scalable content management techniques,

2. Design of an integrated customization and personalization framework able to deal with complex multimedia content formats and metadata,

3. Demonstration of the end-to-end platform accessing the content over various networks including fixed and wireless networks as well as fixed and mobile terminals.

ISIS integrates different types of media resource adaptation, among-which are gBSD-based adaptation of BIFS, scalable audio and scalable video content, as well as a set of resource transmodings. Based on our methodology, i.e. by usage of the transmoding-related descriptors as defined in this section, we have developed the media transmoding part. The implementation of the ISIS transmoding module is described in more details in next chapter.

4.2.2 A complete Resource Conversion methodology

The last section described a methodology for hinted-transmoding within the framework of MPEG-21, this section provides a description on how to apply this methodology to cover other types of Resource Conversion, i.e. transcoding and transforming. The principal elements of the approach are the description of the modality, format and form of the original resource, the description of the parameters of conversions, the expression of the author and end user conversion preferences.

4.2.2.1 Description of the original resource

Since the presented methodology in this section provides a framework for description of the information on all types of Resource Conversion and not only transmoding, the description of the resource should be extended to provide information on all aspects of content, which could be of use to decide and perform a conversion. These aspects are modality, format, coding and form parameters. As in the first phase of the work, we chose to provide the description of the resource in MPEG-7 descriptors. Figure 4.9 provides an MPEG-7 resource description for a video encoded in MPEG-4 – and stored in an mp4 file, with 24 frames per second and with dimension of 240*144.

The MPEG-7 resource description can be provided in a separate XML file. Our choice was to take advantage of the DIDL structure by giving the resource description within the *Statement* of (one of) the *Descriptor(s)* of the *Component* element that contains the corresponding *Resource* element. Figure 4.10 demonstrates the idea.

```
<mpeg7:MediaFormat>
        <mpeg7:Content href="urn:mpeg:cs:ContentCS:2001">
                <mpeg7:Name>video</mpeg7:Name>
        </mpeg7:Content>
        <mpeg7:FileFormat href="urn:mpeg:cs:FileFormatCS:2001">
                <mpeg7:Name>mp4</mpeg7:Name>
        </mpeg7:FileFormat>
        <mpeg7:VisualCoding>
           <mpeg7:Format href="urn:mpeg:cs:VisualCodingFormatCS:2001">
                <mpeg7:Name>MPEG-4 Visual</mpeg7:Name>
           </mpeg7:Format>
           <mpeg7:Frame width="240" height="144" rate="24" />
        </mpeg7:VisualCoding>
</mpeg7:MediaFormat>
```

Figure 4.9: MPEG-7 Resource description of an MPEG-4 video

```
<DIDL>
             .
             .
             .
        <Component >
                <Descriptor>
                        <Statement >
                        <!-- the description of the resource in MPEG-7 !-->
                        </Statement>
                </Descriptor>
                     .
                     .
                     .
                <Resource />
        </Component>
             .
             .
             .
</DIDL>
```

Figure 4.10: Integration of resource description in a DID instance

4.2.2.2 Description of conversion-related information

We have defined XML schemas for the expression of conversion information. The descriptors contain the description of conversion preferences and suggested/recommended generic parameters of a set of transmoding, transcoding and transforming conversions. The syntax and semantic of the XML schemas are provided in Annex B.

Note— At the time, DIA Amendment 1 proposed an open and extensible conversion description. The syntax and semantic of our proposed schemas are based on the *ConversionInformation* and *ConversionDescription* defined in the Proposed Draft AMD1 [25]. There are slight differences in syntax and semantic with final DIA Amendment 1.

Conversion parameters

The considered transmoding parameters are the same as described in the last section. Other generic conversion parameters that were considered in this work are as follows:

- Transcoding parameters: Each transcoding conversion descriptor includes the description of the resulting format and coding parameters.
- Transforming parameters:
  - Text translation. Parameters: language (resulting language), font parameters that are optional and are provided in case of any change comparing to original rich information; for example, the conversion of a rich text in English to a rich text in Japanese, or a rich text in Thai to a rich text in Indian. The resulting rich information may vary from the original rich information.
  - Image resizing and cropping. Parameters: starting point, which is optional and used in case of cropping, and size, which is also optional and used to express the resulting size.
  - Speech translation. Parameters: language (resulting language)
  - Text summarization. Parameters: a set of starting and ending character points.

Conversion preferences

As for the transmoding descriptor, we have considered two attributes for conversion preferences: quality and priority. Following the same policy as for the transmoding descriptor, these attributes describe the preferences of the author for a certain conversion of a particular resource, and are expressed within the corresponding conversion descriptor in the CDI. The general preferences of the user are provided in *ConversionPreference* in the XDI. Figure 4.11 presents an example of a CDI containing two original resources: one text and one image. For the text resource, the author has expressed his preferences and hints for two conversions: text-to-image (TextVisualization) and text translation. For the image resource one conversion has been described: image reformatting. This conversion description may have been generated for example from the UED/UCD. In this figure "rcd:" is the prefix for the Resource Conversion Description namespace. *ti* refers to text-to-image transmoding, *tt* refers to text-to-text (text translation) transforming and *ii* refers to image-to-image transcoding.

```
<DIDL>
 <Item>
  <Component>
   <Descriptor>
    <Statement mimeType="text/xml">
     <dia:ConversionInformation>
      <dia:ConversionDescription xsi:type="rcd:TransmodingConversionType">
       <dia:ConversionUri>http://www.example.com/ti</dia:ConversionUri>
       <rcd:Transmoding>
        <rcd:Parameters xsi:type="rcd:TextVisualizationParametersType">
         <rcd:To href="urn:mpeg:mpeg7:cs:ContentCS:2001">
          <mpeg7:Name>Image</mpeg7:Name>
         </rcd:To>
         <rcd:FontParameters fontSize="20">
          <rcd:FontStyle>BOLD</rcd:FontStyle>
          <rcd:FontFamily> ARIAL</rcd:FontFamily>
         </rcd:FontParameters>
         <rcd:Resolution horizontal="300" vertical="400"/>
          <rcd:TextMotionStyle href="urn:mpeg:mpeg21:dia:cs:TextMotionStyleCS:2003">
           <mpeg7:Name>TickerTape</mpeg7:Name>
          </rcd:TextMotionStyle>
         <rcd:TextColorInformation foregroundColor="0.2 0.5 0.6" backgroundColor="0 0 0"/>
        </rcd:Parameters>
       </rcd:Transmoding>
      </dia:ConversionDescription>
      <dia:ConversionDescription xsi:type="rcd:TransformingConversionType">
       <dia:ConversionUri>http://www.example.com/tt</dia:ConversionUri>
       <rcd:Transforming>
        <rcd:Parameters xsi:type="rcd:TextTranslationType">
         <rcd:Language href="urn:mpeg:mpeg21:cs:LanguageCS:2001">
          <mpeg7:Name>French</mpeg7:Name>
         </rcd:Language>
         <rcd:FontParameters fontSize="20">
          <rcd:FontStyle>BOLD</rcd:FontStyle>
          <rcd:FontFamily>ARIAL</rcd:FontFamily>
         </rcd:FontParameters>
        </rcd:Parameters>
       </rcd:Transforming>
      </dia:ConversionDescription>
     </dia:ConversionInformation>
    </Statement>
   </Descriptor>
   <Resource mimeType="text/plain" ref="mytext.txt"/>
  </Component>
  <Component>
   <Descriptor>
    <Statement mimeType="text/xml">
     <dia:ConversionInformation>
      <dia:ConversionDescription xsi:type="rcd:TranscodingConversionType">
       <dia:ConversionUri>http://www.example.com/ii</dia:ConversionUri>
        <rcd:Transcoding>
         <rcd:Parameters xsi:type="rcd:FinalFormatType">
          <rcd:TargetFormat href="urn:mpeg:mpeg21:cs:FormatCS:2001">
           <mpeg7:Name>JPG</mpeg7:Name>
          </rcd:TargetFormat>
         </rcd:Parameters>
        </rcd:Transcoding>
       </dia:ConversionDescription>
      </dia:ConversionInformation>
     </Statement>
    </Descriptor>
    <Resource mimeType="image/png" ref="myimage.png"/>
   </Component>
 </Item>
</DIDL>
```

Figure 4.11: An example of a complete CDI containing conversion descriptors

4.2.2.3 Description of composite conversions

To realize a composed conversion, the atomic conversions can be cascaded (AND) and/or paralleled (OR). The expression method of the conversion descriptors in a DID instance, should, therefore allow for such a feature. Figure 4.12 shows how this can be done within our model.

Figure 4.13 shows a concrete example of composite conversions. For a text resource, two conversions described in two *ConversionDescription*s are cascaded in one *ConversionInformation*. This composed conversion (text-to-image plus image transcoding) is then paralleled to another *ConversionInformation* containing a text translation conversion, by means of a *Conversion* element.

Note— It is assumed that the atomic text-to-image transmoding results in a bitmap image.

4.2.3 Limitations of the static description of conversion parameters

Static description of conversion parameters values helps the adaptation engine make a correct decision on the form of the adaptation and a correct selection of the conversion parameters values. However, in cases where different values are provided for a conversion parameter, this approach is either not the best way for conducting the decision-making, or proves to impose limitations.

Consider the following use case example. The original resource is a non-scalable color JPEG image of size 640*480. The DI provider wishes to make this content accessible to terminals with various display resolutions and color capabilities. He also wishes to suggest a tool for the conversion, identified by an URI. To achieve the conversion, this tool needs the following parameters: a) scaling factor for image resizing (from 0 to 3), and b) number of colors of the output image (from 1 to 3). The values of these parameters are selected based on the some values given in the UED: terminal display size and color capabilities.

```
<Statement >
     <dia:Conversion>

       <dia:ConversionInformation>
             <rcd:ConversionDescription ... />
                          OR
             <rcd:ConversionDescription ... />
      </dia:ConversionInformation>

                  AND

      <dia:ConversionInformation>
             <rcd:ConversionDescription ... />
      </dia:ConversionInformation>

     </dia:Conversion>
</Statement>
```

Figure 4.12: ANDing and ORing conversions in DID

```
<DIDL>
 <Item>
  <Component id="R1">
   <Descriptor>
    <Statement mimeType="text/xml">
     <dia:Conversion>
      <dia:ConversionInformation>
       <rcd:ConversionDescription xsi:type="rcd:TransmodingConversionType">
        <dia:ConversionUri>http://www.example.com/TI</dia:ConversionUri>
        <rcd:Transmoding>
         <rcd:Parameters xsi:type="rcd:TextVisualizationParametersType">
          <rcd:To href="urn:mpeg:mpeg7:cs:ContentCS:2001">
           <mpeg7:Name>Image</mpeg7:Name>
          </rcd:To>
          <rcd:FontParameters fontSize="20">
          <rcd:FontStyle>BOLD</rcd:FontStyle>
          <rcd:FontFamily>ARIAL</rcd:FontFamily>
         </rcd:FontParameters>
         <rcd:Resolution horizontal="300" vertical="400"/>
          <rcd:TextMotionStyle href="urn:mpeg:mpeg21:dia:cs:TextMotionStyleCS:2003">
           <mpeg7:Name>TickerTape</mpeg7:Name>
          </rcd:TextMotionStyle>
          <rcd:TextColorInformation foregroundColor="0.2 0.5 0.6" backgroundColor="0 0 0"/>
         </rcd:Parameters>
        </rcd:Transmoding>
       </rcd:ConversionDescription>
       <rcd:ConversionDescription xsi:type="rcd:TranscodingConversionType">
        <dia:ConversionUri>http://www.example.com/II</dia:ConversionUri>
        <rcd:Transcoding>
         <rcd:Parameters xsi:type="rcd:FinalFormatType">
          <rcd:TargetFormat href="urn:mpeg:mpeg21:cs:FormatCS:2001">
           <mpeg7:Name>JPG</mpeg7:Name>
          </rcd:TargetFormat>
         </rcd:Parameters>
        </rcd:Transcoding>
       </rcd:ConversionDescription>
      </dia:ConversionInformation>
      <dia:ConversionInformation>
       <rcd:ConversionDescription xsi:type="rcd:TransformingConversionType">
        <dia:ConversionUri>http://www.myURL.com/TT</dia:ConversionUri>
        <rcd:Transforming>
         <rcd:Parameters xsi:type="rcd:TextTranslationType">
          <rcd:Language href="urn:mpeg:mpeg21:cs:LanguageCS:2001">
           <mpeg7:Name>French</mpeg7:Name>
          </rcd:Language>
          <rcd:FontParameters fontSize="20">
           <rcd:FontStyle>BOLD</rcd:FontStyle>
           <rcd:FontFamily> ARIAL</rcd:FontFamily>
          </rcd:FontParameters>
         </rcd:Parameters>
        </rcd:Transforming>
       </rcd:ConversionDescription>
      </dia:ConversionInformation>
     </dia:Conversion>
    </Statement>
   </Descriptor>
    <Resource mimeType="text/plain" ref="mytext.txt"/>
  </Component>
 </Item>
</DIDL>
```

Figure 4.13: Expressing a composite conversion

For such use cases, static description of conversion parameters values is not a good idea, since the conversion parameters, i.e. the scaling factor and the number of colors, do not have only one possible value, but rather a value space. Also the selection of final values should be optimized. In other words, the static description of conversion parameters values gives direct hints on the final values of conversion parameters. Therefore, once an adaptation is chosen, the adaptation engine retrieves these directly-given values. No optimized selection is hence done on the values of parameters. However, in some use cases, such as the above one, we needs to give several values for a parameter and let the decision-making engine decide on the optimum value of the parameter based on the context constraint.

In next section, we present another description tool that uses DIA *AdaptationQoS* in order to solve such limitations. In section 4.3.3, we will show how this tool can deal with the above use case.

4.3 Second phase: ConversionLink

This section explains our design of a more efficient description tool, called ConversionLink that allows “soft” and non-static expression of the parameters values for a particular conversion. This helps achieving a more intelligent decision-making on the optimal values of the conversion parameters. The development of this tool is a joint work with Cyril Concolato, Philippe de Cuetos and Benoît Pellan at ENST under the framework of DANAE (IST-2004-507113) and TIRAMISU (IST-2003-506983) [46] European IST projects.

ConversionLink takes a different route from the methodology presented in previous sections that proposed “static” and “hard” expressing methods for conversion parameters and preferences. The strategy behind the idea of ConversionLink is the “soft” expression of conversion information, as could be done by DIA *AdaptationQoS*.

The need for concrete descriptions of conversions was recognized through our contributions to MPEG, which resulted in the creation of DIA Amendment 1. The latest version of DIA Amendment 1 specifies a *ConversionDescription* element, which is a placeholder for providing suggestions to a DIA engine on how to convert a resource (or a DID *Item* or *Component*). *ConversionDescription* is considered as a Digital Item Adaptation description tool, which falls into the 8th category of DIA tools, referred to as *DIA Suggestion Tools for DID*. DIA Amendment 1 proposes to identify conversions by using a URI and optional parameters, or using an XML syntax, as we did in the first step of our work. However, based on the implementations of DIA that were developed in our group under the framework of IST projects, we are convinced that the DIA Amendment 1 specification of the *ConversionDescription* element is not sufficient, specific and “soft” enough to describe an interoperable and flexible conversion, notably compared to (g)BSD-based adaptation by usage of the *BSDLink* and *AdaptationQoS*.

4.3.1 Methodology

Rather than defining a way for expressing sets of static conversion parameters, as can be done by *ConversionDescription*, the goal of ConversionLink is to provide a framework that enables smart selection of the values of conversion parameters (output values) within several given values, based on given values for some parameters or conditions (input values). This is done in an interoperable way, even if the Resource Conversion tools used are non-standard. We proposed a tool to extend to conversions, the possibilities given by *BSDLink* for (g)BSD-based adaptation. In order to facilitate future adoption, the structure and semantics of the ConversionLink adaptation tool is intentionally very close to that of *BSDLink*.

As *BSDLink* for (g)BSD-based adaptation, ConversionLink enables linking *AdaptationQoS* steering description tools and conversion tools. It provides a way to describe the parameters that will be needed by the conversion tool, and to give suggested values for them. ConversionLink reuses the exact syntax and semantics of most of the schema types defined for *BSDLink*: *SteeringDescriptionType*, *BaseParameterType*, *ConstantType* and *IOPinRefType*. Annex C provides only the syntax and semantics of new types and elements.

4.3.2 Using ConversionLink in a DIA engine

Figure 4.14 recalls the general adaptation architecture given in figure B.1 of ISO/IEC 21000-7. We can see that D_1 represents a CDI, while D_2 represents an XDI. The usage of *ConversionLink* in a DIA engine is based on *BSDLink* usage principles and follows the same processing steps. ISO/IEC 21000-7 explains:

"*The adaptation engine consists of the Description Processing Engine (DPE) and the Resource Processing Engine (RPE). The DPE takes as an input the descriptions (D_1) contained within DI_1 and descriptions (D_2) contained within DI_2. The output of the DPE is on the one hand D_3, e.g., a possible transformed description, which governs the actual resource adaptation (input to RPE) and on the other hand D_4 – which will become part of the adapted Digital Item (DI_3).*

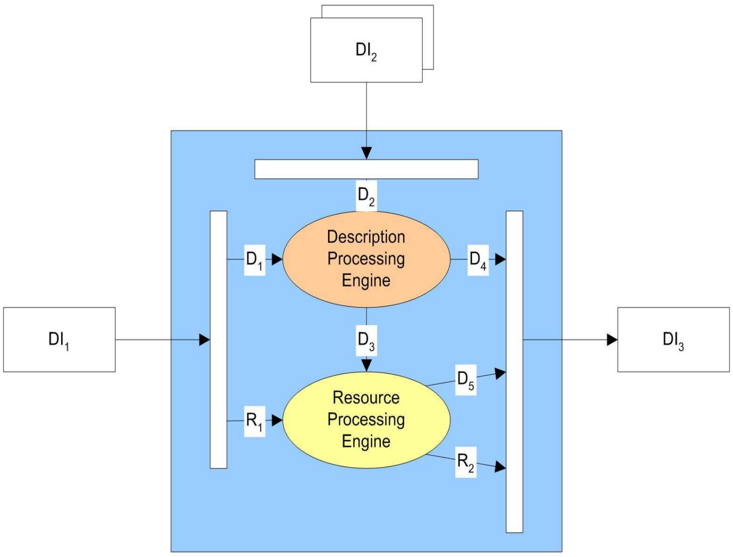


Figure 4.14: High-level architecture of a Digital Item Adaptation engine

The RPE adapts the resource (R_1) governed by means of the output of the DPE (D_3). The output is the adapted resource (R_2) and possibly an updated (transformed) description (D_5), which reflects the modifications in the resource accordingly. Finally, the transformed Descriptions (D_3 and D_5) and the resource are composed to the adapted Digital Item (DI_3). The adapted Digital Item could be also validated in the same way as the input Digital Item."

Here in order to illustrate the use of ConversionLink tool, we recall the first example of ISO/IEC 21000-7 Annex B: "Usage of *BSDLink* tool" for BSD-based adaptation. In this example, instead of a scalable JPEG2000 image resource we consider a non-scalable JPEG image. Figure 4.15 presents example descriptions and resources for the DIA Engine in the case of conversion-based adaptation. Figure 4.16 depicts the architecture of the Description Processing Engine (DPE). The processing of D_1 (*ConversionLink*), $D_{2.1}$(UED), and $D_{2.2}$(UCD) is as follows:

Short term	Example instantiation
D_1	ConversionLink with an AdaptationQoS description as steering description, a description of the input resource, and a reference on the tool suggested for the conversion.
D_2	Usage Environment Description (UED) and/or Universal Constraints Description (UCD)
D_3	ConversionDescription leading the conversion
D_4	Transformed ConversionLink and resource description
R_1	Resource, e.g., JPEG image
D_5	Updated resource descriptions
R_2	Adapted resource, e.g., scaled JPEG image or grayscale JPEG image

Figure 4.15: Example descriptions and resources

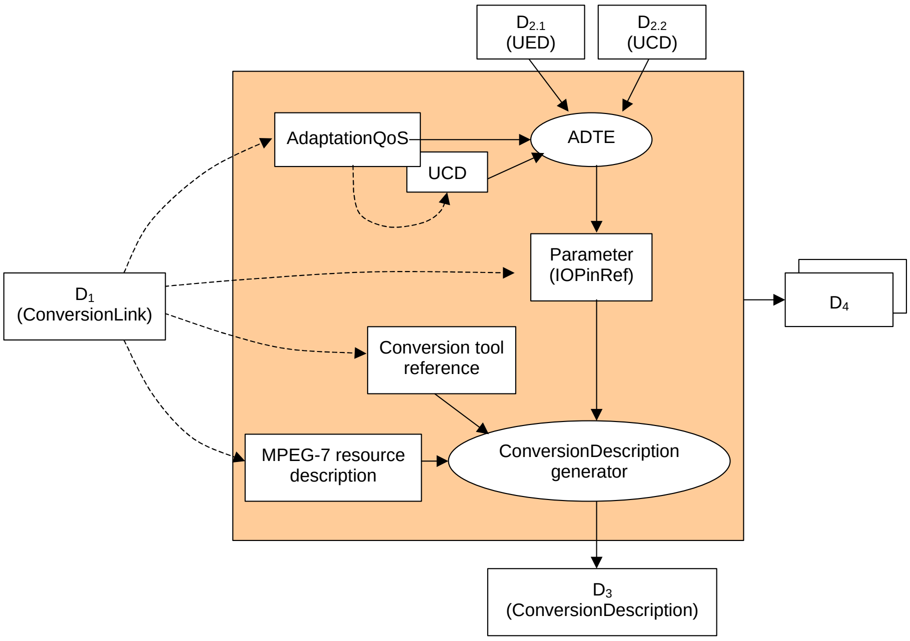

D2.1 (UED)
D2.2 (UCD)
AdaptationQoS
ADTE
UCD
Parameter (IOPinRef)
D1 (ConversionLink)
D4
Conversion tool reference
MPEG-7 resource description
ConversionDescription generator
D3 (ConversionDescription)

Figure 4.16: Architecture of Description Processing Engine

1. The steering description, i.e., *AdaptationQoS*, forms the input for the Adaptation Decision Taking Engine (ADTE) together with the UED.

The *AdaptationQoS* description as well as the UED may be further restricted by two separate UCDs.

2. The ADTE retrieves the actual values for the variable parameters (*IOPins*) defined in the *ConversionLink*, from the UED. Additionally, it resolves the constraints as defined in the UCDs.

3. The ConversionDescription generator produces conversion descriptions using the MPEG-7 resource description, the *IOPinRef* parameters and the conversion tool reference.

4. The ADTE together with the whole DPE might also produce a transformed MPEG-7 description.

Subsequently, the conversion description is forwarded to the Resource Processing Engine (RPE), i.e., the conversion tool, as depicted in Figure 4.17. The RPE generates the converted resource and possibly updates the addressing information according to the converted resource.

Note— By comparing the ConversionLink solution with simple static conversion description, we see that the ConversionDescription (D3) in above figures is in fact the equivalent (or the same as) our defined static conversion descriptions as done in first and second phases of this work. Therefore the first solution (static conversion description) can be reused within a ConversionLink solution.

4.3.3 ConversionLink example

Consider the use case of section 0. Figure 4.18 shows how *ConversionLink* solution can be used for this use case. The Description Processing Engine (DPE) will use it jointly with the UED and/or UCD to obtain values for *IOPins* SCALE and NR_COLORS. These *IOPin*s are used to retrieve the values for the parameters of conversion tool *urn:enst:ImageProcessing* (namely rescale and colors). The Reference to a description of the input resource (akiyo.xml#ImageDesc) may be useful for providing information specific to the resource, that can be used as additional parameters for the conversion (e.g., the key frames of a video for video summarization, or the region of interest of an image for image cropping). This would require that the Resource Processing Engine (RPE) is aware of the characteristics of the conversion tool used for this conversion.

The UED does not differ from the example given in ISO/IEC 21000-7. It describes a display resolution of 176x220 pixels and no color capability. For simplicity of presentation, in this example we do not further constrain the usage of the DI by UCDs. Concerning the generation of ConversionDescription (D3), Figure 4.19 shows the output of the Description Processing Engine to the Resource Processing Engine as a description of type *ConversionInformationType*.

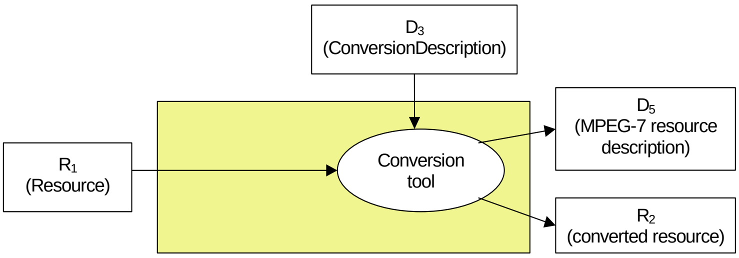


Figure 4.17: Architecture of Resource Processing Engine

Note— The syntax of this example (Figure 4.19) is conformed to the latest version of DIA Amendment 1 at this time.

This description provides the following suggestions of conversion to the Resource Processing Engine: the conversion tool to use (*urn:enst:ImageProcessing*) and the parameters to use with this tool. The values of parameters “rescale” and “colors” are obtained from the results of the Adaptation Decision-Taking Engine. The Resource Processing Engine may now execute the conversion of the resource and generate the adapted resource as well as updated resource descriptions.

In a *ConversionLink*-based approach, the syntax of the metadata is generated by the Description Processing Engine and should govern the adaptation of the resource. In the case of *BSDLink*, it is a (g)BSD description of the final resource. In ConversionLink approach this should be a *ConversionDescription* element with constant parameters as in Figure 4.19. However, because of the ConversionDescription/*ConversionParameters* element in DIA is currently not specific enough, the interoperability is not insured when the DPE and RPE are implemented by different vendors.

Figure 4.20 gives an example to illustrate the advantages of using *ConversionLink* when describing alternative conversions on a resource. We consider a JPEG2000 image resource. Possible adaptations are described by different description tools, such as a *BSDLink* or a *ConversionLink* in a CDI. Depending on the UED and UCD, issued by the provider or consumer of the DI, the Adaptation Decision-Taking Engine will select the adaptation that meets the expressed constraints based on AQoS description and resource description. The fourth conversion suggests a default Resource Conversion tool by giving direct static values of conversion parameters. In this example we can see how the syntax of *ConversionLink* can be used for either static description of conversion parameters values and or dynamic description of conversion parameters values. We could also express the static description of conversion parameters (fourth conversion) via static *ConversionDescription* as in Figure 4.19. In such case, we could see, how *ConversionLink*, *BSDLink* and static *ConversionDescription* could be used together in one CDI for expression of alternative adaptations. However, *ConversionLink* is after all a more complete solution, since it provides the opportunity to express both static and dynamic values of conversion parameters.

```
<DIA>
 <DescriptionMetadata>
  <ClassificationSchemeAlias alias="AQoS" href="urn:mpeg:mpeg21:2003:01-DIA-AdaptationQoSCS-NS"/>
  <ClassificationSchemeAlias alias="MEI" href="urn:mpeg:mpeg21:2003:01-DIA-MediaInformationCS-
NS"/>
  </DescriptionMetadata>

  <Description xsi:type="AdaptationQoSType" id="akiyo_AQoS2">
  <!-- Provides the scale factor wrt display resolution -->
    <Module xsi:type="LookUpTableType">
      <Axis iOPinRef="DISP_WIDTH">
        <AxisValues xsi:type="IntegerVectorType">
          <Vector>80 160 320 640</Vector>
        </AxisValues>
      </Axis>
      <Axis iOPinRef="DISP_HEIGHT">
        <AxisValues xsi:type="IntegerVectorType">
          <Vector>60 120 240 480</Vector>
        </AxisValues>
      </Axis>
      <Content iOPinRef="SCALE">
        <ContentValues xsi:type="IntegerMatrixType" mpeg7:dim="4 4">
          <Matrix>
            3 3 3 3
            3 2 2 2
            3 2 1 1
            3 2 1 0
          </Matrix>
        </ContentValues>
      </Content>
    </Module>

    <!-- Provides the number of color components wrt color capability of the display -->
    <Module xsi:type="LookUpTableType">
      <Axis iOPinRef="COLOR_CAPABLE">
        <AxisValues xsi:type="BooleanVectorType">
          <Vector>false true</Vector>
        </AxisValues>
      </Axis>
      <Content iOPinRef="NR_COLORS">
        <ContentValues xsi:type="IntegerMatrixType" mpeg7:dim="2">
          <Matrix>1 3</Matrix>
        </ContentValues>
      </Content>
    </Module>

    <IOPin id="DISP_WIDTH"><!-- Frame width of the resource -->
      <!-- Horizontal resolution of a display -->
      <GetValue xsi:type="SemanticalDataRefType" semantics=":AQoS:6.5.9.1"/>
    </IOPin>

    <IOPin id="DISP_HEIGHT"><!-- Frame height of the resource -->
    <!-- Vertical resolution of a display -->
      <GetValue xsi:type="SemanticalDataRefType" semantics=":AQoS:6.5.9.2"/>
    </IOPin>

    <IOPin id="COLOR_CAPABLE"><!-- Color capability of a display -->
      <GetValue xsi:type="SemanticalDataRefType" semantics=":AQoS:6.5.9.26"/>
    </IOPin>

    <IOPin id="SCALE"/><!-- Scale factor for the conversion tool -->
    <!-- Number of output color components for the conversion tool -->
    <IOPin id="NR_COLORS"/>
  </Description>

  <Description xsi:type="ConversionLinkType">
    <SteeringDescriptionRef uri="#akiyo_AQoS2"/>
    <ResourceDescriptionRef uri="akiyo.xml#ImageDesc"/>
    <BitstreamRef uri="akiyo.jpg"/>
    <ResourceConversionToolRef uri="urn:enst:ImageProcessing123"/>
    <Parameter xsi:type="IOPinRefType" name="rescale"><Value>SCALE</Value></Parameter>
    <Parameter xsi:type="IOPinRefType" name="colors"><Value>NR_COLORS</Value></Parameter>
  </Description>
</DIA>
```

Figure 4.18: ConversionLink usage example

```
<did:DIDL>
  <did:Item>
    <did:Descriptor id="ConversionDescription">
      <did:Statement mimeType="text/xml">
        <DIA>
          <Description xsi:type="ConversionInformationType">
            <ConversionDescription xsi:type="ConversionUriType">
              <sx:rightUri definition="urn:enst:ImageProcessing"/>
              <ConversionParameter>
                <Parameter xsi:type="ConstantType" name="rescale">
                  <Value>3</Value>
                </Parameter>
                <Parameter xsi:type="ConstantType" name="colors">
                  <Value>1</Value>
                </Parameter>
              </ConversionParameter>
            </ConversionDescription>
          </Description>
        </DIA>
      </did:Statement>
    </did:Descriptor>
  </did:Item>
</did:DIDL>
```

Figure 4.19: An example of generated ConversionDescription by DPE

4.4 Conclusion

In this chapter we described the theory of our methodology for defining tools for Resource Conversion within MPEG-21. We described how the two principal phases of our approach, led to the adoption of some of our proposed tools by MPEG-21 DIA.

The first phase of the approach considered “direct” and “static” description of the conversion parameters and preferences. It defined a set of generic conversion parameters for each Resource Conversion, that are not dependent of any particular algorithm. It also defines attributes for the expression of (author) preferences on a certain conversion for a particular resource. These attributes express the preferred quality and priority of conversion(s). We argued that providing the values of the conversion parameters and preferences helps the adaptation decision-making engine to make a more intelligent decision on the optimal type and parameters values of adaptation.

The second phase of the work considered the usage of DIA *AdaptationQoS* tool for non-static description of conversion parameters and optimization of the decision-making process. The proposed *ConversionLink* provides the Description Processing Engine with a link to an AQoS description and a mapping between the *IOPins* used in the *AdaptationQoS* description and the parameters that will be used by the conversion tool for this conversion. Although the Resource Conversion tools can be proprietary, *ConversionLink* allows describing how to obtain the corresponding (generic or specific) conversion parameters in an interoperable way, which would facilitate the adoption of MPEG-21 for conversion of media with a variety of tailored tools from different vendors. The approach is similar to that of DIA *BSDLink* tool for (g)BSD-based adaptation. Next chapter describes our implementation of an MPEG-21-based Resource Conversion engine.

```
<did:DIDL>
  <did:Item>
    <did:Descriptor id="ImageDesc">
      <did:Statement mimeType="text/xml">
        <DIA>
          <!-- BSD-based adaptation alternative -->
          <Description xsi:type="BSDLinkType" id="BsdAdaptation">
           <SteeringDescriptionRef uri="AQoS_BABY.xml"/>
           <BSDRef uri="BABY_BSD.xml"/>
           <BitstreamRef uri="baby.jp2"/>
           <BSDTransformationRef uri="jp2.xsl"
            type="http://www.w3.org/1999/XSL/Transform"/>
           <BSDTransformationRef uri="jp2.stx"
            type="http://stx.sourceforge.net/2002/ns"/>
           <Parameter xsi:type="IOPinRefType" name="scale">
            <Value>SCALE</Value>
           </Parameter>
           <Parameter xsi:type="IOPinRefType" name="CsizIn">
             <Value>NR_COLORS</Value>
           </Parameter>
          </Description>
          <!-- Down-scaling JPEG conversion alternative -->
          <Description xsi:type="ConversionLinkType" id="ResourceConversion1">
            <SteeringDescriptionRef uri="AQoS_BABY.xml"/>
            <ResourceDescriptionRef uri="#ImageDesc"/>
            <BitstreamRef uri="baby.jp2"/>
            <ResourceConversionToolRef uri="urn:enst:JpegTranscoding"/>
            <Parameter xsi:type="IOPinRefType" name="rescale">
              <Value>SCALE</Value>
            </Parameter>
            <Parameter xsi:type="IOPinRefType" name="colors">
              <Value>NR_COLORS</Value>
            </Parameter>
          </Description>
          <!-- Cropping JPEG conversion alternative -->
          <Description xsi:type="ConversionLinkType" id="ResourceConversion2">
            <SteeringDescriptionRef uri="AQoS_BABY2.xml"/>
            <ResourceDescriptionRef uri="#ImageDesc"/>
            <BitstreamRef uri="baby.jp2"/>
            <ResourceConversionToolRef uri="urn:enst:JpegTranscoding2"/>
            <Parameter xsi:type="IOPinRefType" name="crop_x">
              <Value>CROP_X</Value>
            </Parameter>
            <Parameter xsi:type="IOPinRefType" name="crop_y">
              <Value>CROP_Y</Value>
            </Parameter>
          </Description>
          <!-- Static default JPEG conversion alternative -->
          <Description xsi:type="ConversionLinkType" id="DefaultConversion">
            <ResourceDescriptionRef uri="#ImageDesc"/>
            <BitstreamRef uri="baby.jp2"/>
            <ResourceConversionToolRef uri="urn:enst:JpegTranscoding"/>
            <Parameter xsi:type="ConstantType" name="rescale">
             <Value>3</Value>
            </Parameter>
            <Parameter xsi:type="ConstantType" name="colors">
             <Value>1</Value>
            </Parameter>
          </Description>
        </DIA>
      </did:Statement>
    </did:Descriptor>
    <did:Component>
      <did:Resource ref="baby.jp2" mimeType="image/jp2"/>
    </did:Component>
  </did:Item>
</did:DIDL>
```

Figure 4.20: A DIDL example with alternative conversions

Chapter 5

RESOURCE CONVERSION: THE IMPLEMENTATION

Summary

In this chapter we describe the architecture of our implementations of an MPEG-21 DIA-based Resource Conversion Engine. This will include the description of our proof-of-concept implementations of a statically-hinted transmoding engine and its enhancement into a full Resource Conversion engine including transcoding and transforming conversions.

Table of Content

Table of Illustrations

5.1 Introduction

In this chapter, we describe the architecture of our implementation of an MPEG-21-based Resource Conversion framework. The theory of the approach was described in the previous chapter. At the time of writing this dissertation, the implementation of an *AdaptationQoS*-based Resource Conversion engine – using *ConversionLink* tool – is not yet complete and the work is being continued under the framework of DANAE project.

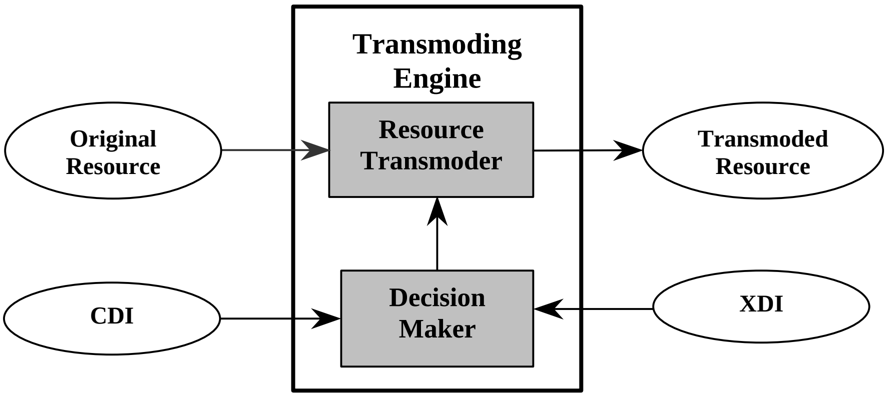
This chapter describes, in a first step, the implementation of a hinted transmoding engine within the framework of ISIS project, and in a second step, evolving this engine to a Resource Conversion engine which performs other conversion types: transcoding, and transforming.

5.2 Statically-hinted transmoding

In this section we first present the architecture of our implementation of a hinted transmoding engine, based upon our proposed transmoding description tool. We then describe how such engine was exploited to provide solutions for some use case scenarios of ISIS project.

5.2.1 Architecture of Transmoding Engine

In this section, we describe the architecture of our implementation of a transmoding engine. In the first step, we developed a parsing and validating software within the framework of the MPEG-21 DIA reference software, that parses and validates the UED and the transmoding descriptor, and stores the usage environment and transmoding metadata in an internal structure. In the second step, we developed a decision-making software and integrated it, together with a set of transmoding tools, to the parser software. The whole software was contributed to MPEG-21 DIA in a *Core Experiment*.



CDI (Content Digital Item): Resource description (MPEG-7), transmoding parameters (*Transmoding* descriptor), and reference to original resource.
XDI (conteXt Digital Item): Context constraints (UED)

Figure 5.1: Architecture of Transmoding Engine

The Transmoding Engine is shown in Figure 5.1, and is composed of two main blocks: the Decision Maker and Resource Transmoder. Its inputs are the original resource, its description together with the transmoding parameters and preferences, included in a CDI, and the description of the context in an XDI. The Decision Maker is the decision-making algorithm, which determines the optimal form of transmoding for the resources with certain constraints. The Resource Transmoder is an implementation of a set of transmodings tools developed at ENST, and produces the transmoded resource. More details on the functionality of these two blocks are given in 5.2.2.2.

5.2.2 Transmoding in ISIS

As mentioned earlier, this implementation was exploited in the ISIS IST project. In this section, we first give a brief description of the overall architecture of ISIS, and detail the architecture of transmoding module and its integration into ISIS project. We then describe transmoding use case scenarios of ISIS, which necessitated and were answered by the integration of our transmoding engine into ISIS architecture.

5.2.2.1 ISIS overall architecture

The Personalization and Customization Framework of ISIS is depicted in Figure 5.2. In this figure the blocks that are within the scope of personalization and customization, are shown with solid frame line. These blocks are: Personalization Server, Media Customizer, Streaming Server, Content Description Database, Network & Device Description Database and the Temporary Cache Database. All other modules are shown in dashed line.

The Personalization Server is the unit that receives the request from the user and creates from the information in the Content Database a list of content items that is relevant for the given user. The Media Customizer is the unit that receives the client's selection of content, determines the optimal customization steps to be applied and performs the adaptation. The Streaming Server is the unit that provides the streaming functionalities of the server and communicates with the end user's terminal. The Content Description Database stores the XML metadata for the available content in the form of DIDs. The Network & Device Description Database stores the XML descriptions of certain types of networks and terminal devices, also in the form of DIDs, using MPGE-21 DIA UED descriptors. The third database is used as a temporary cache to store the user's constraints description for the duration of a session. The summarized walkthrough of a single session, as shown in Figure 5.2, is as follows:

1: The Client sends a request including the Terminal Capabilities, Network Capabilities, User Characteristics and Natural Environment Characteristics (an XDI). In order to conserve bandwidth, the Terminal and/or Network Capabilities may be referenced in the DID sent by the client, using a DeviceID and NetworkID. The DIDs are sent using an HTTP request.

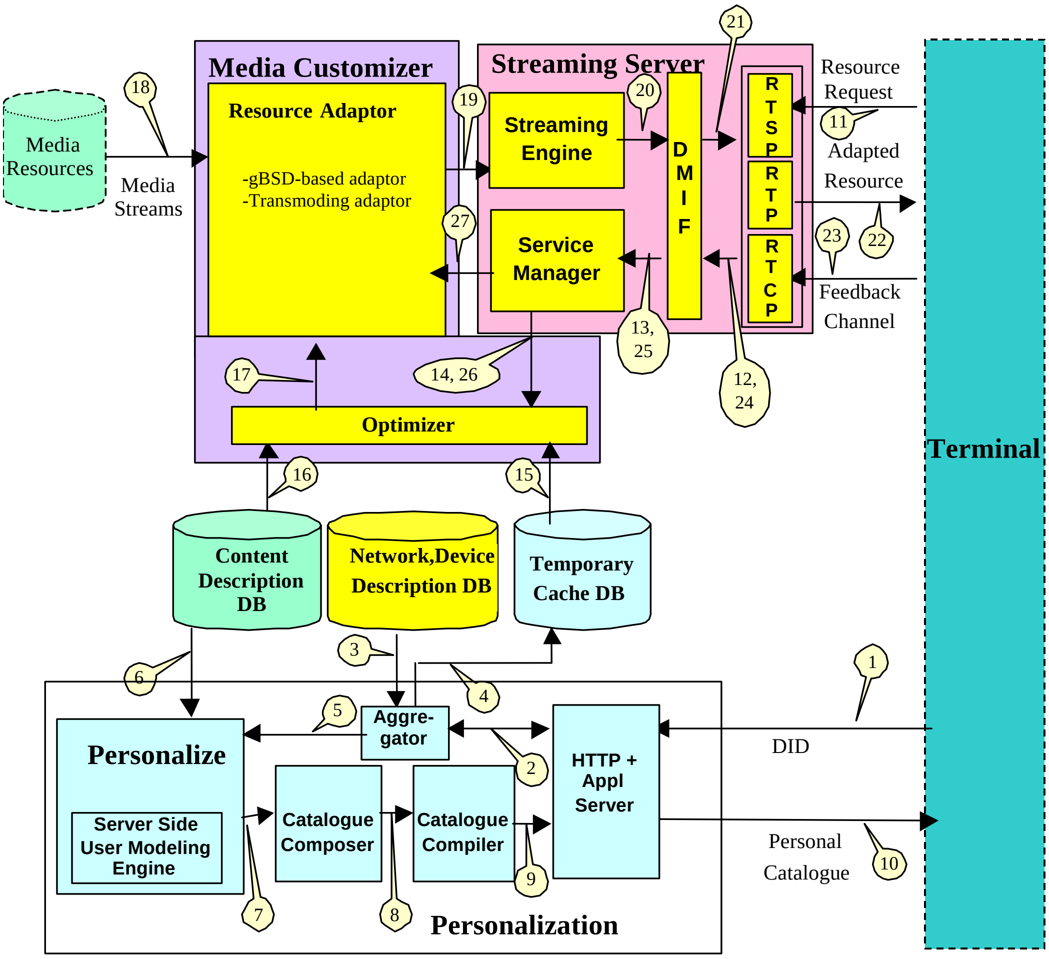


Figure 5.2: ISIS Personalization & Customization architecture

2, 3, 4, 5, 6, 7, 8, 9: The HTTP/Application Server extracts the DID and passes it to the Aggregator. This latter first retrieves the Network and Device descriptions from the corresponding database, using the DeviceID and NetworkID. The retrieved descriptions are aggregated into the DID. The Aggregator also generates and inserts in the DID a Session ID. This is an ID to identify the Session and will be unique throughout the session. The SessionID replaces the existing DII of the DID. The Aggregator stores the aggregated DID in the Temporary Cache Database for later retrieval by the Optimizer. The DID and its DII are passed to the database. The Aggregator then sends the aggregated DID to the Personalizer. The Personalizer processes the DID and issues a Query to the Content Database. This query is requesting content DIDs that are relevant to the user. Technical boundaries set by the Device, Network and Natural Environment are not considered here assuming that the adaptation process can customize any resource such that the technical constraints can be met. If occasionally that should not be the case the optimizer will generate an error message. The query returns a list of DIIs relevant to the user request along with a short description of the DID content. This short description will be included in the DID, at the highest level, and will be extracted by the Personalizer. The list of DIIs is passed to the Catalogue Composer, which composes a catalogue using a default presentation template, the list of DIIs and the short description of each DID. The catalogue is an XML-based description of the scene to be presented to the user and is passed to the Catalogue Compiler to be compiled into a BIFS scene and put into an MP4 file. The Catalogue Compiler passes the produced mp4 file to the HTTP Server.

10: The HTTP Server transmits to the end user’s terminal the mp4 file containing the catalogue.

11: The user selects the preferred content and the Terminal requests the corresponding resources. Session ID and DII (that are associated with the selected content) are included in the RTSP request.

12, 13, 14, 15, 16: The request reaches the Service Manager. The Streaming Server extracts from the RTSP request the file name, the SessionID and the DII and passes them to the Optimizer as parameters. Using the Session ID II, the Optimizer retrieves the XDI of the given user from the Temporary Cache Database. Using the DII, the Optimizer retrieves the requested DID from the Content Description Database.

17: The Optimizer processes the adaptation options listed in the retrieved DID, and selects the option that satisfies the constraints, as they are described in the constraints DID, and results the maximum quality. Once the optimal adaptation option is selected, the Optimizer calls the Resource Adaptor. If the adaptation option is a (g)BSD based bitstream adaptation, the parameters to be passed to the Resource Adaptor are a reference to the Bitstream Syntax Description ((g)BSD), a reference to the XSLT stylesheet for the description transformation and the set of parameters to be passed to the stylesheet. If the adaptation option is a transmoding, the parameters to be passed to the Resource Adaptor are the transmoding form, and the required parameters for this transmoding, which are to be passed to the corresponding transmoder. If no adaptation option matches the constraints, an error message will be generated and passed, through the Resource Adaptor and Streaming Engine, to the end-user's terminal.

18: The Resource Adaptor retrieves the specified resource. In case of a BIFS scene, this data exchange is not needed; therefore no action is performed here.

19: The Resource Adaptor performs the transformation on the gBSD, using the XSLT stylesheet and parameters it received as input from the Optimizer. Then, the transformed gBSD is used to adapt the resource. In a case of a transmoding, the Resource Adaptor performs the decided transmoding and In the case of BIFS scene, the B4 description is compiled into an MPEG-4 stream. The stream of the adapted/transmoded resource or scene is then input to the Streaming Engine.

20, 21, 22: The stream of the adapted resource is passed from the Streaming Engine to the DMIF and then to the RTP stack. The resource is then streamed to the end user's terminal.

If the first requested resource was a scene, then upon receipt of the scene, the client will begin requesting the individual resources included in the scene. For each resource, steps 11-20 are repeated.

23, 24, 25, 26, 27: For 3D mesh, feedback information on the current viewing location and cached data is passed from the RTCP stack to the DMIF and then to the Service Manager. It is then input to the Optimizer, which then re-evaluates the constraints using the new information and selects, if necessary, another adaptation option. It then inputs the new adaptation parameters to the Resource Adaptor repeating steps 18-22. Then feedback data is directly routed to the resource adaptor, where it is parsed and interpreted by a standalone module, which proceeds to the specific bitstream adaptation, using the gBSD.

5.2.2.2 Transmoding module in ISIS architecture

The Transmoder plug-in is in a part of Media Customizer Module in the overall architecture of ISIS. Its inputs are user's selection of content from the Streaming Server (DID), the description of the content and the transmoding parameters from the Content Description Database and the user's constraints description from the Temporary Cache Database. As shown in Figure 5.3, the transmoder plug-in consists of two main blocks: Transmoding Optimizer and Transmoding Resource Adaptor. Same colors are used for same modules in Figure 5.2 and Figure 5.3 in order to provide better understanding.

As ISIS treats other types of adaptations (than transmoding), the Transmoding Optimizer block is a part of the ISIS principal Optimizer module in Figure 5.2. It, based on the given constraints decides on the optimal transmoding. The Transmoding Resource Adaptor block performs the corresponding transmoding adaptation based on the transmoding parameters given in the content's DID. The output of the Transmoding Resource Adaptor is the transmoded resource, which is input to the Streaming Manager.

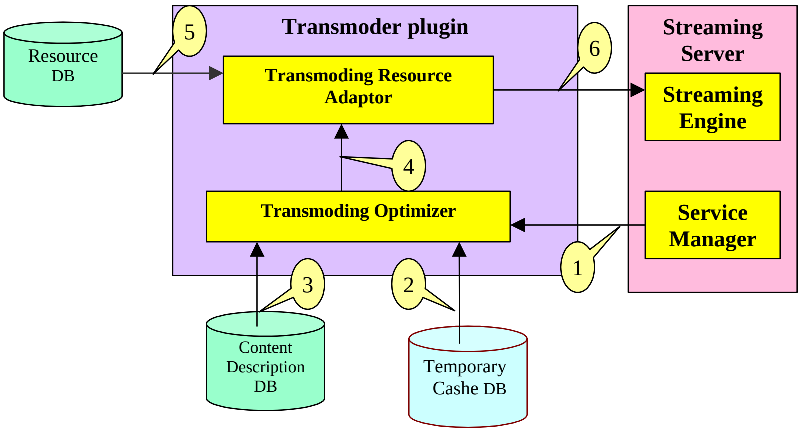


Figure 5.3: Architecture of ISIS Transmoding Module

Session walkthrough is as follows:

1. The Streaming Server extracts the file name from the RTSP request, the SessionID and the DII and passes them to the Decision Maker as parameters.

2. Using the SessionID, the Transmoding Optimizer retrieves the XDI of the given user from the Temporary Cache Database.

3. Using the DII, the Transmoding Optimizer retrieves the requested DID from the Content Description Database.

4. The Transmoding Optimizer processes the adaptation options listed in the retrieved DID, and selects the transmoding solution that satisfies the constraints, as they are described in the constraints DID, and results in the maximum quality. Once the optimal transmoding adaptation option is selected, the Transmoding Optimizer calls the Transmoding Resource Adaptor. If no adaptation option matches the constraints, an error message will be generated and passed, through the Transmoding Resource Adaptor and Streaming Engine, to the end-user's terminal.

5. The Transmoding Resource Adaptor performs the transformation on the resource using the transmoding parameters extracted from DID from the Transmoding Optimizer.

6. The stream of the transmoded resource is input to the Streaming Engine.

Transmoding Optimizer sub-module

The Transmoding Optimizer sub-module is depicted in Figure 5.4. It has been entirely developed in JAVA, and comprises three main modules:

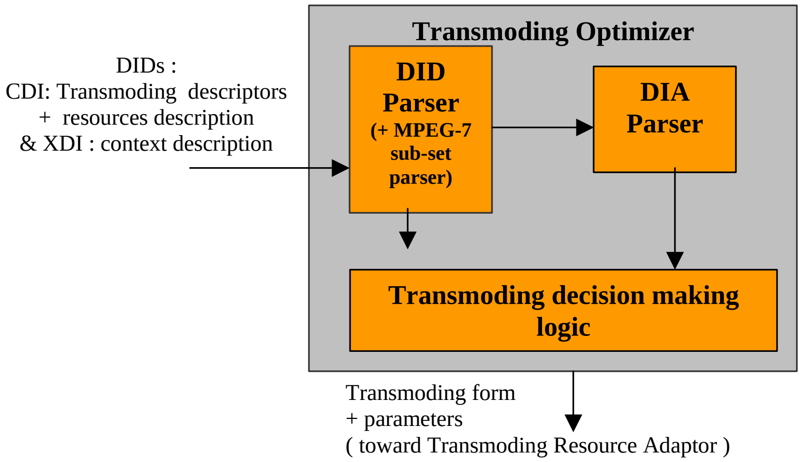


Figure 5.4: Architecture of Transmoding Optimizer sub-module in ISIS

The DID Parser

It takes as input the DIDs: CDI and XDI, and analyses them. It stores the reference(s) to the resource(s) locations. It then forwards the DIA descriptors of the XDI (context) and CDI (transmoding descriptor) to the DIA Parser. We have used the DID parser of the MPEG-21 reference software developed by Ghent University [47]. An MPEG-7 parser (which parses only a subset of MPEG-7 descriptors) has been integrated to this DID parser. It analyzes the resource descriptions (CDI: resource description) in order to retrieve and store the parameters of resource(s) description.

The DIA Parser

We implemented a DIA UED parser that parses and analyses the MPEG-21 DIA including Transmoding descriptors. It receives, form the DID Parser, DIA UED and Transmoding descriptors, analyzes them and stores the metadata in an internal structure.

The Transmoding Decision Making Logic

It is the decision making engine of the Transmoding Optimizer. It takes as input the constraints metadata and the resource references, stored in an internal structure. It works on a rule-based approach:

Static Priorities have been associated to different constraints, for example terminal characteristics are more important than user preferences. First are verified constraints of higher priority. These are the one needed for making any transmoding decision, for example modality supports of terminal. These constraints will be used to filter out all transmoding options that are not supported by the current terminal and network. In the second step, from the remaining options that are technically feasible then an optimal decision is taken using the transmoding preferences of the author, as well as other constraints. It then extracts the given values of the related transmoding parameters (coming from Transmoding descriptor). The Transmoding Optimizer module, then, returns the optimal transmoding form and the values of related parameters, to the Transmoding Resource Adaptor the decision.

Transmoding Resource Adaptor sub-module

The Transmoding Resource Adaptor is an implementation of a small set of transmodings: text-to-image, graphics2D-to-video and graphics2D-to-image (all these transmoding modules have been developed at ENST). The Transmoding Resource Adaptor sub-module is depicted in Figure 5.5. Inputs are the transmoding form, the required parameters and the reference of the resource to be transmoded.

Image-to-text plug-in: We developed a simple java implementation that simply draws a text to an image using the transmoding-related parameters, such as display size, font parameters and color parameters.

Graphics2D to video plug-in: This plug-in has been developed under the framework of GPAC project [48] and at ENST and is a tool that can dump a pure MPEG-4 2D BIFS animation (no audio, no image, no video) to an MPEG-4 video with a given frame rate as the transmoding parameter. This tool has been developed in C.

Graphics2D-to-image plug-in: This plug-in is the same as the previous one, outputs a bitmap using the frame number (time point) parameter, given as the transmoding parameter.

5.2.3 Exploitation of the Transmoding engine

5.2.3.1 Examples for usage of Transmoding engine

This section intends to explain the results of the use of our transmoding engine, by describing concrete transmoding usage examples.

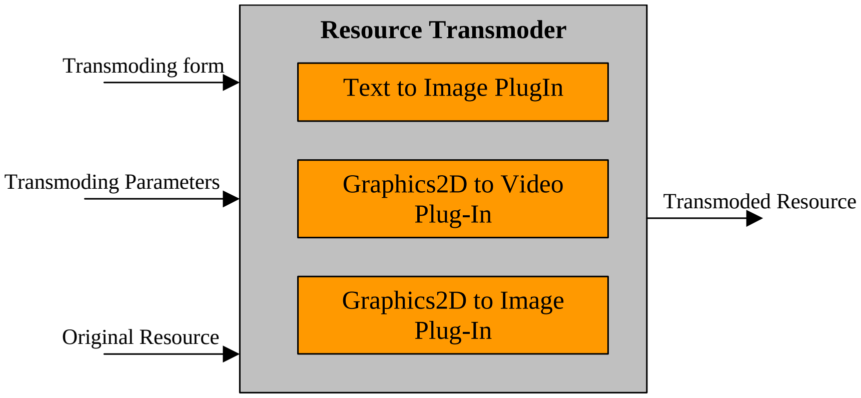
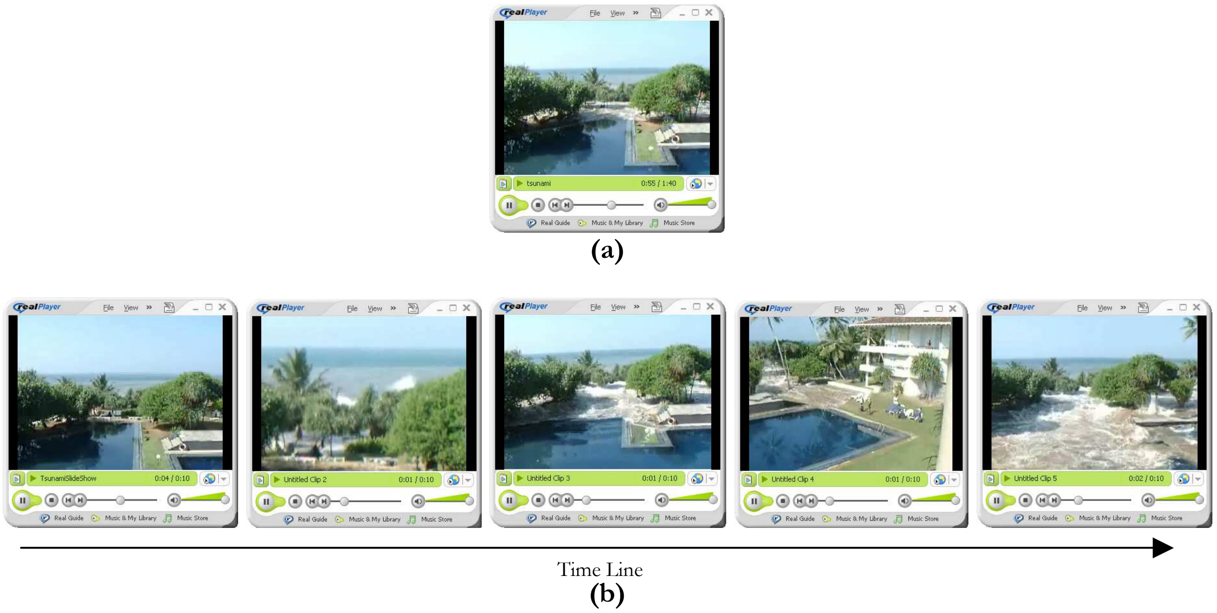


Figure 5.5: Architecture of Transmoding Resource Adaptor sub-module in ISIS

Example A: The original resource is a video content and the context constraints are as follows:

- Terminal capabilities (decoding): Audio: MP3. Image: JPEG. No video. Graphics-2D: slideshow. Display resolution: 320*240. Bits per pixel: 8.
- User preferences:
  - Modality Conversion preferences: video to image, video to slide show, no video to audio
  - Modality Priority: graphics, audio, image
- Provider-side constraints: transmoding parameters
  - Video-to-image, method (tool): frame selection, parameters: Time-code=01:30:01.24, priority=1, quality = 1.
  - Video-to-slideshow, method (tool): video summarization, parameters: Time-code= 00:00:04.24 duration=5, Time-code=00:10:14.24 duration=5, Time-code=00:20:04.24 duration=5, Time-code=00:50:12.24 duration=5, Time-code=01:30:01.24 duration=5. Priority=1, quality = 1.
- Cost function constraints: Adaptation operations based on transmoding are less expensive than adaptation operations based on direct bitstream modifications.

To best satisfy these constraints, the optimal adaptation is a video-to-slideshow transmoding. The parameters are retrieved from the corresponding transmoding descriptor. The video is then transmoded to a slideshow. Figure 5.6 shows screenshots of a video and the five slides of the resulting slideshow.


tsunami 0:55 / 1:40
TsunamiSlideShow 0:04 / 0:10
Untitled Clip 2 0:01 / 0:10
Untitled Clip 3 0:01 / 0:10
Untitled Clip 4 0:01 / 0:10
Untitled Clip 5 0:02 / 0:10
Time Line

(a)

(b)

Figure 5.6: Screenshots of a) original video media, and b) five slides of the adapted slideshow

Example B: The service provider provides multi-language text-image content. The block diagram of such multimedia service is depicted in Figure 5.7. We consider two different multimedia content and two different terminals: one with and one without Persian character set support. When the original content contains an image and an English text, it is sent to both terminals and no transmoding is required. We now consider an original Persian multimedia content containing one image and two text resources (title and body) in Persian language. In a non-transmoding scenario, this content is sent to a terminal supporting Persian character set, without any transmoding. In a transmoding scenario, the decoding capabilities of the second terminal are audio (MP3), image (JPEG), and no support for Persian character set. Available display sizes, i.e. the suggested resolutions for the texts that are to be transmoded to image, are 109*43 for the title text, and 268*125 for the body text. This information is given within the corresponding transmoding descriptor. No English version of the Persian text is provided. The adaptation decision taken by the optimizer is then two text-to-image transmodings. The original Persian texts are then transmoded to two images using the transmoding parameters given in the corresponding transmoding descriptors. In Figure 5.7, this transmoding case is outlined via a dashed-line box.

Now consider the multimedia document of Figure 5.8 as the original Persian multimedia document available on the server. Our focus is on the Persian text resources. For the non-transmoding case, as described above, the Persian texts are rendered as text resources as shown in part (a) of Figure 5.9. Depending on the display size of the terminal, some text resizings may be required. The Persian texts are rendered using the available Persian font of the terminal, which is not necessarily the same as in the original document, as can be seen in part (a) of Figure 5.9. For the transmoding case, the Persian texts are transmoded into JPEG images, using the desired – here the original – font (which is available in the server), as shown in part (b) of the same figure. The resolution and text font parameters of the resulting images are extracted form the transmoding descriptors associated to each original text media in the CDI.

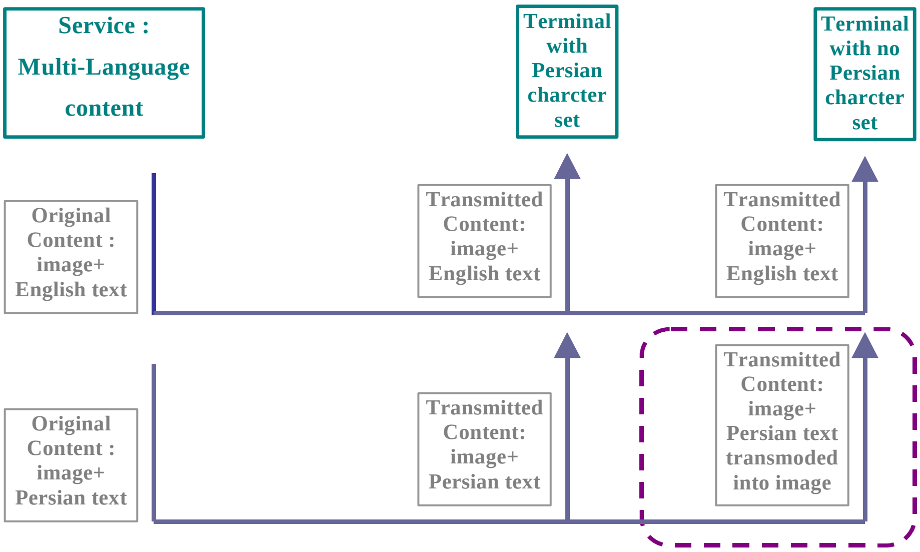
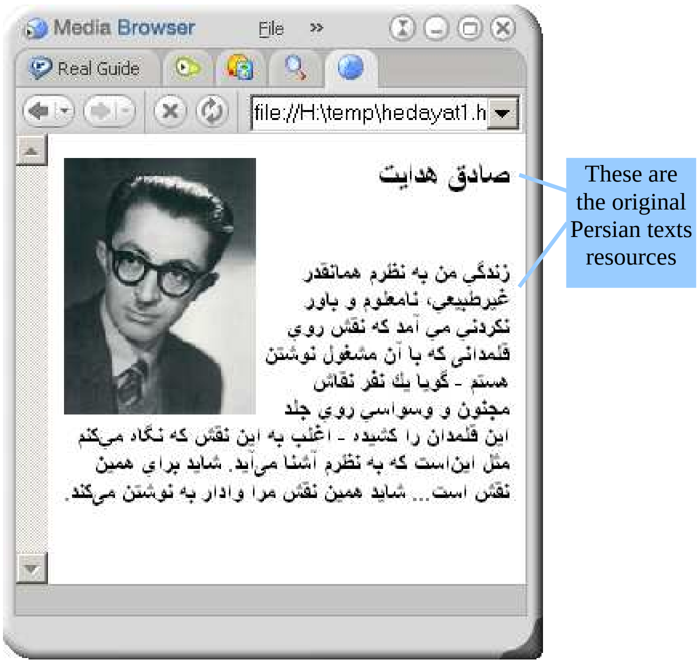

Service :
Multi-Language
content
Terminal with Persian charcter set
Terminal with no Persian charcter set
Original Content : image+ English text
Transmitted Content: image+ English text
Transmitted Content: image+ English text
Original Content : image+ Persian text
Transmitted Content: image+ Persian text
Transmitted Content: image+ Persian text transmoded into image

Figure 5.7: Block diagram of a Multi-language multimedia service that uses transmoding.

Once, the text-to-image transmodings is done – using the recommended resolutions given in the corresponding transmoding descriptors, depending on the device display size, image resizings may be required (see our MSSA solutions in Chapter 7and Chapter 8for this).

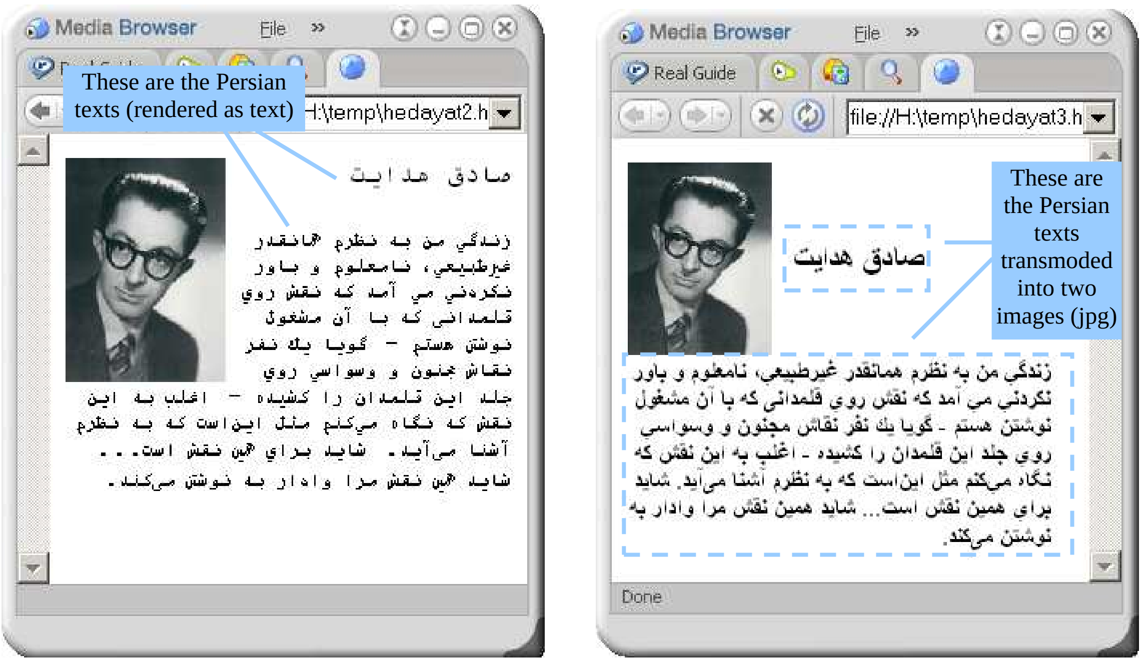


صادق هدايت

زندگي من به نظرم همانقدر
غيرطبيعي، نامعلوم و باور
نكردني مي آمد كه نقش روي
قلمداني كه با آن مشغول نوشتن
هستم - گويا يك نفر نقاش
مجنون و وسواسي روي جلد
اين قلمدان را كشيده - اغلب به اين نقش كه نگاه ميكنم
مثل اين‌است كه به نظرم آشنا مي‌آيد. شايد براي همين
نقش است... شايد همين نقش مرا وادار به نوشتن مي‌كند.

These are the original Persian texts resources

Figure 5.8: Screenshot of an original Persian multimedia content in the server.



These are the Persian texts (rendered as text)

صادق هدايت

زندگي من به نظرم همانقدر
غيرطبيعي، نامعلوم و باور
نكردني مي آمد كه نقش روي
قلمداني كه با آن مشغول
نوشتن هستم - گويا يك نفر
نقاش مجنون و وسواسي روي
جلد اين قلمدان را كشيده - اغلب به اين
نقش كه نگاه مي‌كنم مثل اين‌است كه به نظرم
آشنا مي‌آيد. شايد براي همين نقش است...
شايد همين نقش مرا وادار به نوشتن مي‌كند.

These are the Persian texts transmoded into two images (jpg)

صادق هدايت

زندگي من به نظرم همانقدر غيرطبيعي، نامعلوم و باور
نكردني مي آمد كه نقش روي قلمداني كه با آن مشغول
نوشتن هستم - گويا يك نفر نقاش مجنون و وسواسي
روي جلد اين قلمدان را كشيده - اغلب به اين نقش كه
نگاه مي‌كنم مثل اين‌است كه به نظرم آشنا مي‌آيد. شايد
براي همين نقش است... شايد همين نقش مرا وادار به
نوشتن مي‌كند.

(a) Persian texts (title and body) are rendered as text resources

(b) Persian texts (title and body) are rendered as image resources

Figure 5.9: Screenshot of the adapted content of Figure 5.8 for a) a terminal supporting Persian character set, and b) a terminal with no Persian character set support.

5.2.4 Evaluation of the Transmoding engine

As seen in these transmoding examples, when the optimum or the only feasible form of adaptation is a modality conversion for a resource for which no alternative in the desired modality is provided, the Transmoding module serves as a resource adaptation module that performs the on-demand modality conversion. Transmoding is an “enabling” technology that allows adaptations that are otherwise infeasible. However, transmoding is not necessarily an often-used technology as transcoding is less disruptive of the presentation. In the context of an adaptation engine for multimedia structured documents, transmoding may be very interesting. We have used transmoding in our Multimedia Scene Semantic Adaptation framework (see Chapter 7 and Chapter 8).

Advantages and weak points: Using an on-demand real-time transmoding engine in an adaptation system, as in ISIS, removes the need for a heavy authoring task. This is due to the fact that, to be sure of the adaptability of his content, the author is not forced to provide different modalities of his content any more. However, a client-server adaptation system such as ISIS that uses on-demand resource transmoding (or adaptation in general) becomes problematic when many simultaneous transmodings or adaptations are necessary to satisfy user requests. This will cause an increase in response time and possibly the saturation of the server. A distributed architecture, using proxy servers, is obviously a good solution that can improve the functionality of the system face to such problems. The continuation of ISIS, DANAE European IST project – in which we participate – aims at developing a distributed adaptation framework.

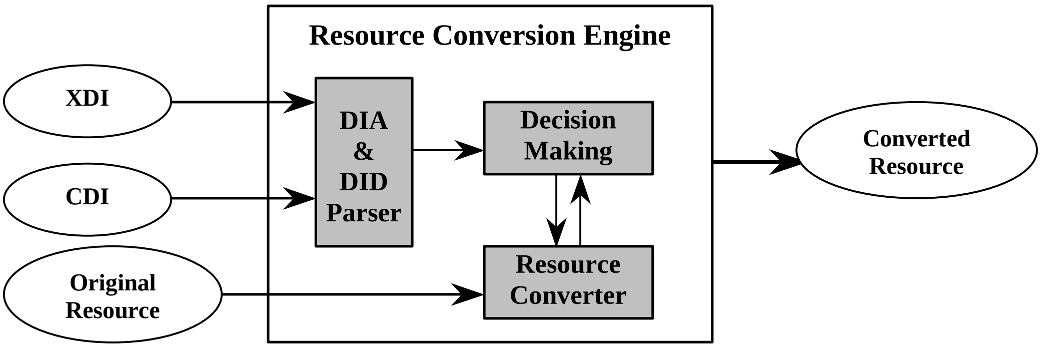
Some indicative measurements: We have done some measurements for three transmoding tools. The OS is Windows™ 2000, the processor is an AMD Athlon™ XP 2000. The target display size is assumed to be 176x144. The average execution times are: 2800 ms for Graphics (BIFS 213Kb, 15s, 12fps) to Video (MPEG-4 Video), 420 ms for Text to Image (mostly PNG encoding) and 15 ms for Graphics to Image conversion. Our transmoding tools are not optimized, these measures are therefore only indicative.

5.3 Statically-hinted Resource Conversion engine

In this section we describe our implementation of a Resource Conversion engine, which is in fact the completion of our transmoding engine, in order to integrate other conversion types.

5.3.1 Architecture

Figure 5.10 depicts the architecture of our MPEG-21 Resource Conversion Engine. It is very similar to the architecture of the Transmoding Engine as described in 5.2.1. The inputs are the CDI, the XDI and the original resource.



CDI (Content Digital Item): Resource description (MPEG-7), conversion parameters (*RCD* descriptors), and reference to original resource.
XDI (conteXt Digital Item): Context constraints (UED): user, terminal, network characteristics

Figure 5.10: Architecture of Resource Conversion Engine

The CDI contains the description of content (in MPEG-7), description of conversion parameters and preferences, and references to original resources. The XDI contains the description of the context (user preferences, terminal and network characteristics) in DIA UED. The DIA parser does also the parsing and validating of the Resource Conversion Descriptors (RCD) as described in previous chapter. The DID&DIA parsers parse the CDI and XDI and store the metadata in an internal structure. The basis of the decision-making algorithm is almost the same as for transmoding engine. The Decision Maker uses this metadata to decide on the optimal type of conversion: it first verifies the description of the content and high-priority constraints such as terminal modality/format support and constructs a list of possible conversions. Then based on the preferences of end user and author it chooses a conversion, for which it fetches the required parameters values in the description of conversion parameters. It then sends the chosen type of conversion, together with the values of necessary parameters, to the Resource Converter, which will convert the resource. The resource converter includes a set of free resource converters such as ImageMagick and FFMPEG, as well as a set of converters developed at ENST as described earlier.

5.4 Conclusion

In this chapter we presented our experimental implementations of MPEGE-21-based statically-hinted transmoding and Resource Conversion engines. These implementations use “static” description of conversion parameters and preferences in order to perform a correct adaptation. Under the framework of two European projects, we are currently working on the implementation of a more efficient Resource Conversion engine based on the second phase of our methodology as described in the previous chapter. Such a Resource Conversion engine will take advantage of usage of MPEG-21 *AdaptationQoS* tool for non-static expression of conversion parameters.

Chapter 6

ADAPTATION OF MULTIMEDIA-COMPOSED PRESENTATIONS

Summary

In this chapter we discuss the notion of adaptation and customization of rich multimedia-composed presentations. We also analyze the principal elements of such framework. We then provide a state-of-the-art on this subject and discuss existing approaches in this area.

Table of Content

Table of Illustrations

6.1 Introduction

When the user's preferences, network capabilities or the device characteristics change, the original presentation may become un-playable, inconvenient or unsatisfactory. A multimedia presentation adaptation system is then required to customize both media objects and the presentation structure – i.e. its spatial and temporal for layout – for a particular context of usage. Providing structural semantic metadata on multimedia scenes helps and in some cases enables such a system to adapt multimedia documents to different usage contexts in a more appropriate manner.

With current document mark-up languages such as HTML [49], the layout of a Web page is relatively static – no temporal dimension, and fixed [50]. Such mark-up languages do not provide any mechanism for description of the metadata that is required for adaptation. SMIL ([51]) provides more flexible mark-ups for multiple alternative layouts, nevertheless, semantic metadata is still missing.

As the first chapter of the second part of this dissertation, this chapter discusses the notion of adaptation and customization of rich multimedia-composed presentations. It then provides a brief state-of-the-art on this subject. We will end by situating our work on semantic adaptation of multimedia presentations, among other existing approaches in this area.

In this dissertation, the terms “multimedia presentation”, “multimedia document” and “multimedia scene”, refer all to multimedia-composed content that contains multiple synchronized (temporally and spatially) media resources, together with the integrating spatial, temporal and logical structure. Adaptation of multimedia scenes, then addresses the adaptation of the scene media objects, as well as the adaptation of the document structure, based on temporal, spatial and semantic relationships between its media objects, and based on the context constraints.

6.2 Basic definitions

6.2.1 Multimedia Scene

A multimedia scene is a multimedia presentation that integrates multiple static or continuous temporal and spatial synchronized media resources. It also specifies how they should be combined together and, based on spatial, temporal, interaction and animation factors, be presented to the user. We refer to the single media resources of a multimedia scene as media objects.

Note— The term “scene” has been adopted from MPEG-4 standard.

There exist several languages for describing multimedia scenes. The MPEG group has developed XMT and BIFS (*BInary Format for Scenes*), which are description languages for MPEG-4 scenes [52]. SMIL (*Synchronized Multimedia Integration Language*), a W3C recommendation, is a specification language with temporal functionalities. $\mathrm{Z_{Y}X}$ is another model for describing multimedia documents [53][54]. Section 6.4 describes some of the most common, accomplished and important multimedia document description models and languages.

6.2.2 Multimedia scene adaptation

Adaptation of a multimedia scene consists of: a) adaptation of its every single media resource, and b) adaptation of the presentation structure i.e. the spatial and temporal relationships between the media objects, in order to: a) satisfy the context constraints, b) satisfy the presentation constraints. In short a multimedia scene adaptation solution should answer the question of single media adaptation with regards to not only the context constraint but also the structure of the presentation. It also should deal with the question of structure adaptation with regards to context constraints.

6.2.3 Semantic adaptation of multimedia scenes

Multimedia scene adaptation, as defined above is not a "complete" multimedia scene adaptation, as it does not guaranty the consistency, coherence and meaningfulness of the resulting scene. A "complete" multimedia scene adaptation should satisfy two types of constraints:

- Physical context constraints, such as terminal capabilities, network characteristics and user/author preferences (on single media adaptation) and etc.

- Content structural semantic constraints, i.e. the semantic constraints that would assure the production of a consistent, meaningful and coherent scene. For instance, consider one image media and its text caption within a multimedia presentation. If, within the adaptation process, the image is eliminated because of a bandwidth limitation, or a lack of image modality support by the terminal, the adaptation engine should also remove the caption of the image. This is a structural semantic constraint, i.e. a semantic constraint related to the structure of the scene. The term "semantic" refers in general to meaning, but in this dissertation, we use the term "semantic" to refer to "structural semantic", although this is a very limited use.

We call "*semantic multimedia scene adaptation*", a multimedia scene adaptation that takes into account both these constraints categories. This is where our work in the area of multimedia presentations adaptation is situated.

6.3 Principal elements

In this section, we describe the principal and basic elements of a multimedia presentation adaptation system. Please notice that, in this section, we do not intend to describe any particular architecture for such system. We only aim to describe the required elements of a multimedia presentation adaptation framework.

Some of the principal elements of a semantic multimedia scene adaptation system are related to media objects adaptation part of the scene adaptation, and are, therefore, shared with a single media adaptation system, as described in Chapter 3. These are:

Resource description, which is in fact the description of the media objects of the scene (we sometimes refer to this as "*physical* content description"), and

Context description. We have previously described this element in section 3.3.2.

Here we describe the required elements specific to a multimedia scene adaptation system.

6.3.1 Semantic information of a scene

An adaptation peer needs to have access to the semantic information of the presentation, which includes the information on semantic relationship between the two media objects (image and text caption) as well as semantic information on each existing media object in the scene.

A simple example is a multimedia document with two images and two texts, each text giving explanation on one of the two images. We assume that even after maximum downscaling of the images, the display size of the user device is still too small for the whole scene. A fragmentation of the scene then becomes necessary. Now, in order to keep the related image and text together in the same scene fragment, and to temporally sort the fragmentations in the correct order, the adaptation engine needs some semantic information on the scene.

As illustrated by this simple example, a complete multimedia content adaptation requires a good understanding of the original document. If the adaptation process fails to analyze semantic structure of a document, then the adaptation result may not be accurate and may cause user misunderstanding or non-comprehension. We, therefore, observe that in a multimedia scene adaptation system, the adaptation core should be provided by an accurate semantic description of the scene. As opposed to *physical* content description, the semantic information of the scene is also called *semantic* description of content. Several solutions have been proposed for the semantic description of multimedia presentations. We will discuss them in section 6.5.

6.3.2 Scene semantic adaptation core

Within a multimedia scene adaptation system, scene semantic adaptation core, is in fact the equivalent of "Resource adaptation core" in a single media adaptation system, and is the entity, which is responsible for

- deciding on the changes (adaptations) to be applied to the structure of the presentation,
- deciding on the type of adaptations to be applied to media objects,
- applying the chosen changes to the structure of the presentation, and

- applying the chosen adaptations to the media objects.

Therefore, it could be said that the adaptation process is considered in two parts: scene optimization that is equivalent of decision-making in single media adaptation, and scene adaptation, which will be done based on the result of the optimization.

The required sub-entities responsible for scene optimization and scene adaptation are explained here.

6.3.2.1 Scene optimization

Scene optimization is responsible to find the optimal form of the scene based on the context and content constraints. So, in fact the scene optimizer element is a decision-making algorithm. The context constraints are, as in the case of Single Media Adaptation, the usage environment parameters such as terminal capabilities, network characteristics, user preferences, etc. The content constraints are the physical and semantic information of the content, as described above.

6.3.2.2 Scene adaptation tool

In a multimedia scene adaptation system, the scene adaptation tool element is responsible for applying adaptation techniques to the whole structure of the document and to the media objects. The adaptation techniques and the necessary parameters for them are decided by the scene optimizer element.

Presentation adaptation tool:

The presentation adaptation element is defined to be the entity responsible for applying the adaptation techniques to the whole structure of the scene, i.e. the temporal organization and the presentation layout of the scene.

The presentation adaptation tool customizes the structure of the scene based on the output of scene optimization. Such a tool could be implemented in XSLT ([55]) for XML-based multimedia documents.

Resource adaptation tools:

This element is defined to be the entity responsible for applying resource adaptation techniques to media objects of a scene. It adapts the resources based on the decision made by the scene optimization element about the type of adaptation and the required parameters. It is the same as defined in section 3.3.3.2.

6.4 Scene description models and languages

In this section we provide brief summaries on some important models and languages for multimedia documents: these are the SMIL 2.0 language, the ZYX model, MPEG-4 BIFS (XMT-A) and Madeus.

Multimedia applications need data models for the representation of the composition of media objects; these are called multimedia document models, on which multimedia presentations description languages are based. They are employed to model the relationships between the media objects participating in a multimedia presentation. The initial requirements to multimedia documents are the modeling of the temporal and spatial course of a multimedia presentation and also the user interaction. However, as authoring of multimedia documents is a very time consuming and costly task, attention has been drawn to reuse multimedia scenes for efficiency and economical reasons. This would ease the task of personalization of multimedia presentations for particular contexts and over heterogeneous environments.

For more detailed information on adaptive multimedia document models, the reader can refer to Ph.D. thesis of Lionel Villard who investigated different document models for editing and adapting multimedia presentations [56].

6.4.1 SMIL 2.0 language

6.4.1.1 Overview

SMIL (pronounced "smile"), Synchronized Multimedia Integration Language, was approved by the World Wide Web Consortium in 1998 as the standard markup language for multimedia presentations [57]. SMIL is similar to HTML, which is the language used for the layout of web pages with text and graphics. SMIL is an XML-based language used to layout multimedia presentations, and adds powerful multimedia and timing capabilities to basic layout and formatting. SMIL allows for creation of sophisticated-looking and interactive rich-media presentations. The last version of SMIL is SMIL 2.1 [58].

Using SMIL, an author can describe the temporal behavior of a multimedia presentation, associate hyperlinks with media objects and describe the layout of the presentation on a screen. SMIL also allows reusing of its syntax and semantics in other XML-based languages, in particular those who need to represent timing and synchronization. For example, SMIL 2.0 components are used for integrating timing into XHTML [59] and into SVG [60]. SMIL supports a number of media file formats and various streaming media file types including video, audio, animation, photos, graphics and text.

6.4.1.2 SMIL documents

SMIL documents are XML 1.0 documents. The root element of SMIL documents is the *smil* element. The *smil* element can contain the following children: *body*, *head*. The *head* element contains information that is not related to the temporal behavior of the presentation. Three types of information may be contained in *head*. These are meta-information, layout information, and author-defined content control. The *body* element contains information that is related to the temporal and linking behavior of the document. It acts as the root element of the timing tree. Figure 6.1 shows a simple SMIL document. The presentation contains an image and an audio, which will be played in parallel.

```
<smil xmlns="http://www.w3.org/2000/SMIL20/CR/Language">
        <head>
            <layout>
               <region id="imageRegion" left="10" .../>
            </layout>
        </head>
        <body>
            <par dur ="120s">
                <img id="photo" src="myPhoto.jpg" region="imageRegion"/>
                <audio id="song" src="mySong.mp3" />
            </par>
        </body>
</smil>
```

Figure 6.1: A SMIL document example

The media content of a SMIL document is structured by using composite elements so-called time containers. A time container (or operator) carries a particular temporal semantic that allows for definition of temporal placement of media objects. These operators are *seq*, *par* and *excl* elements. A *seq* container, short for "sequence" defines a sequence of elements in which elements play one after the other. A *par* container, short for "parallel", defines a simple time grouping in which multiple elements can play back at the same time. *excl* is a time container with semantics based upon *par*, but with the additional constraint that only one child element may play at any given time.

6.4.1.3 SMIL modules and profiles

A *Module* is a collection of semantically-related XML elements, attributes, and attribute values, that represents a unit of functionality. Modules are defined in coherent sets. This coherency is expressed in that the elements of these modules are associated with the same namespace.

A *Language Profile* is a combination of modules. Modules are *atomic*, i.e. they cannot be subset when included in a language profile. Furthermore, a module specification may include a set of integration requirements, to which language profiles that include the module must comply.

Commonly, there is a main language profile that incorporates nearly all the modules associated with a single namespace. For example, the SMIL 2.0 language profile uses most of the SMIL 2.0 modules.

Usually, the same name is used to loosely reference both -"SMIL 2.0" in the example. Also, the name "profile" is used to mean "language profile".

SMIL 2.0 provides a scalability framework, where a family of scalable SMIL profiles can be defined using subsets of the SMIL 2.0 language profile. A SMIL document can be authored conforming to a scalable SMIL profile such that it provides limited functionality on a resource-constrained device while allowing richer capabilities on a more capable device. SMIL 2.0 Basic (or SMIL Basic) is a profile that meets the needs of resource-constrained devices such as mobile phones and portable disc players. The SMIL Basic profile provides the basis for defining scalable SMIL profiles. A SMIL profile allows a SMIL user agent to implement only the subset of the SMIL 2.0 standard it needs, while maintaining document interoperability between devices profiles built for different needs. A scalable profile enables user agents of a wide range of complexity to render from a single, scalable, SMIL document intelligent presentations tailored to the capabilities of the target devices. Conformance to a SMIL Basic profile provides a basis for interoperability guarantees. The advantages of scalable profiles are:

- Authors can re-purpose SMIL content targeting a wide range of devices that implement SMIL semantics.
- The rendering of the same content can be improved automatically as devices get more powerful.
- All SMIL 2.0 documents can share a document type, a schema, and a set of defined namespaces, and the required default xmlns declaration.
- Any future SMIL 2.0 extensions can easily be incorporated into SMIL documents and user agents.

A scalable profile is defined by extending the SMIL Basic profile, using the content control facilities to support application/device specific features via a namespace mechanism.

6.4.1.4 Adaptation support in SMIL

The structure of SMIL documents is defined in a way that helps the adaptation of SMIL documents to different contexts. The SMIL specification also defines several elements that can help the process of adaptation. In this section we describe these mechanisms.

Spatial organization of SMIL documents: In SMIL 2.0, the spatial information of the media objects are given using *layout*, *root-layout* and *region* elements. SMIL 2.0 separates the description of the presentation layout and the description of the content (media object). Therefore, layout modifications (adaptations) can be done without changing the content. Likewise, the content could be modified or adapted, without changing the spatial organization of a document. In SMIL, only the *region* identifiers link the content to the spatial organization.

Interaction and hyper linking in SMIL: SMIL specification defines several mechanisms of interaction, particularly hypermedia links. Figure 6.2 shows a simple example of a SMIL document containing a link. When the user clicks on the video object “myVideo2.mpeg”, a new presentation window will be started. The usage of hypermedia links helps adapting the SMIL documents by scene decomposition. This is done by providing links to media objects that, due to limited display size, cannot be displayed on the display together with other media objects.

```
...
<seq>
    <video src="myVideo1.mpeg"/>
    <par>
        <a href=http://www.example.org/aSMILdocumnet show="new">
            <video src="myVideo2.mpeg" ... />
        </a>
        <text src="myText.txt" ... />
    </par>
</seq>
...
```

Figure 6.2: An example for linking in a SMIL document

Content control in SMIL: SMIL 2.0 specification defines content control modules, that contain elements and attributes, which provide for runtime content choices and optimized content delivery. SMIL content control functionality is partitioned across four modules:

- BasicContentControl, containing content selection elements and predefined system test attributes;

CustomTestAttributes, containing author-defined custom test elements and attributes;

- PrefetchControl, containing presentation optimization elements and attributes; and
- SkipContentControl, containing attributes that support selective attribute evaluation.

SMIL 1.0 provides a "test-attribute" mechanism to process an element only when certain conditions are true, for example when the language preference specified by the user matches that of a media object. One or more test attributes may appear on media object elements or timing structure elements; if the attribute evaluates to *true*, the containing element is played, and if the attribute evaluates to *false* the containing element is ignored.

SMIL 1.0 also provides the *switch* element for expressing that some document parts are alternatives, and that the first one fulfilling certain conditions should be chosen. This is useful to express that different language versions of an audio file are available, and that the client may select one of them. The SMIL 2.0 BasicContent module includes the test attribute functionality from SMIL 1.0 and extends it by supporting new system test attributes. In example of Figure 6.3, one text object should be selected to accompany the video object. If the system language is English, *subtitle.txt* is selected. If the system language is French, *soustitre.txt* is selected. This example shows how adaptation by alternative substitution is supported within SMIL.

```
...
  <par>
    <video src="myVideo.mpg" ... />
    <switch>
      <text src="subtitle.txt" systemLanguage="fr" ... />
      <text src="soustitre.txt" systemLanguage="en" ... />
    </switch>
  </par>
...
```

Figure 6.3: An example for use of switch element in SMIL documents

6.4.2 Z$_Y$X model

Susanne Boll and Wolfgang Klas propose an approach for modeling adaptable and reusable multimedia documents [53] [54]. The model is called Z$_Y$X and offers primitives that provide a support for reuse of structure and layout of document fragments and for the adaptation of the content and its presentation, to the context. The model design aims to fulfill the three principal requirements: *reusability*, *adaptability* and *presentation-neutrality*, as defined in the following paragraphs. Here, we provide a brief summary on Z$_Y$X concepts and structure, and describe how the model is designed to achieve these three objectives.

6.4.2.1 Concepts and structure of Z$_Y$X model

In this section, we present the terminology and the basic concepts of the Z$_Y$X model. The Z$_Y$X model describes a multimedia document by means of a tree. The nodes of the tree are the *presentation elements* and the edges of the tree *bind* the presentation elements together in a hierarchical fashion. Each presentation element has one *binding point* with which it can be bound to another presentation element. It also has one or more *variables* with which it can bind other presentation elements. Additionally, each presentation element can bind *projector variables* to specify the element's layout. Figure 6.4 depicts the graphical representation of these basic elements.

Presentation elements can be media objects or elements that represent the temporal, spatial, layout, and interactive relationships between the media objects. Figure 6.5 represents a document tree model. The root element the sequential element *seq*, and binds the media objects *Image* and *Text* to its variables v_2 and v_4, as well as a parallel element *par* to its variable v_3. The *par* element synchronizes a *Video* and an *Audio*. The fragment starts with the presentation of the root element, i.e. the sequential element, whose binding point can bind the fragment to another presentation element in a more complex multimedia document tree. Unbound variables can later be used, e.g., a title at the beginning and a summary at the end of the sequence, later.

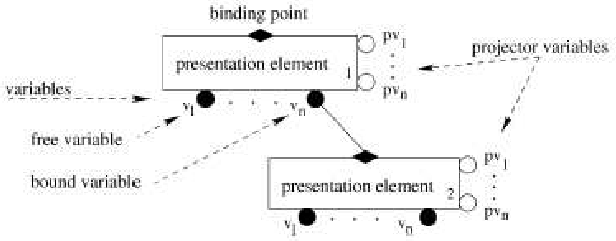


Figure 6.4: Graphical representation of the basic document elements

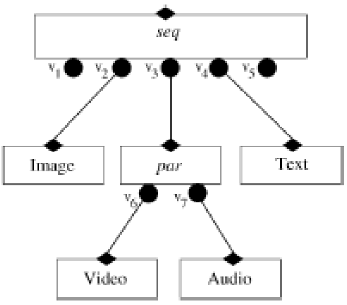


Figure 6.5: Simple document tree for a Z_YX fragment

6.4.2.2 Reusability

The model provides features that allow for reusing parts of medias or document fragments. With regards to the granularity of reusability, the model supports two levels of reusability: reusability at the media level, and reusability at fragment level. Reusability at media level is assured by means of selector element: *temporal-s* and *spatial-s* elements. The former helps the selection of a temporal part of a continuous media, while the latter supports the selection of a spatial area of visual media objects. Reusability at fragment level is provided by the presence of free (unbound) variables, the encapsulation of a fragment into a complex media element and the usage of external media elements (fragments composed in other formats). With regards to the reusability type, the model supports identical and structural reuse.

The identical reuse can be realized by usage of selector elements, while for realization of structural reusability; the model provides the projector elements that influence the visual and audible layout. They define how a media object or fragment is presented, for example the presentation speed of a video or the spatial position of an image. They permit the separation of layout and presentation. Therefore by means of changing and or adding projector elements, one can change the layout of a document. This allows for reusability of the same structure with different presentation layouts.

6.4.2.3 Adaptability

The model defines adaptability as the ability of the document to best match the context of the user who requested the document. To support this, both descriptions of the context and multimedia content that can be adapted to this context are needed. The model captures the context in a so-called User *profile*, i.e. the metadata that describes the user's topics of interest, presentation system environment, and network connection characteristics.

The model provides two presentation elements for an adaptation of the document to a user profile: the *switch* element and the *query* element. The *switch* element allows us to specify different alternatives for a specific part of the document. With each of the alternatives under a *switch* element, there is associated metadata that describes the context in which this specific alternative is the best choice for presentation. This metadata is specified as a set of discriminating attribute-value pairs for each alternative. During presentation, the user profile is evaluated against the metadata of the *switch* and that alternative is selected for presentation of which the discriminating attributes best match the current user profile. A *switch* element can be used only if all alternatives can be modeled at authoring time, in advance to the presentation. Hence, the *switch* element implements the requirement for static adaptability of the model. Non-static adaptability can be specified with a *query* element. By means of metadata, the query represents the fragment that is expected at this point in the presentation. When the document is selected for presentation the *query* element is evaluated and the element is replaced by the fragment best matching the metadata given by *query* element. The *query* element provides for the *dynamic* adaptability of the model as the evaluation of the query and the selection of the fragment takes place just before presentation.

6.4.2.4 Presentation-neutrality

S. Boll et al., in [53] define that a multimedia content is presentation-neutral when the multimedia material is independent of the actual realization of a presentation for a particular client and under a certain context. The requirement of presentation-neutrality is strongly interrelated with the structural reusability. The explicit separation of structure and layout allows for presentation-neutral representation. As outlined before, the variables of a presentation element need not to be bound in the first place, this also applies for the projector variables. It is possible to specify the presentation-neutral course of the presentation and, later, bind the presentation-dependent layout just when the document is selected for presentation. Then, the presentation-neutral structure of the document is bound via projector variables to the presentation-dependent layout defined by a set of projectors.

6.4.3 XMT-A Language (BIFS)

MPEG-4 scene description language BIFS (BInary Format for Scenes) is based on the Virtual Reality Modeling Language (VRML) [61]. In VRML, a scene consists of three different tools: the scene graph, an event routing mechanism, and prototypes. The scene graph is constituted of nodes, grouped in a hierarchical structure, which describes objects on screen and their properties. Nodes can be roughly divided in *grouping* nodes (that are used to logically combine objects and compose them spatially, and *leaf* nodes that provide graphic primitives (circle, rectangular, line, cube, cylinder etc.), text, video, audio and sensors to interact with objects on the screen (i.e. double click, drag-n-drop functionalities). Event routing provides the author with a mechanism through which events generated by nodes can be propagated to other nodes. This processing can change the state of the nodes, create additional events or change the structure of the scene graph. Finally prototypes allow for extension of pre-build sets of nodes.

BIFS is a binary format, which means that a compiler is needed to translate the scene from text to binary. Of course the authoring will be done through text. The textual format used is the XMT-A format, the official XML low-level MPEG-4 scene description language standardized by MPEG. XMT-A is the exact transcription of the BIFS bitstream to text and is a very low-level representation of the scene. However there is another standard textual format for MPEG-4 called XMT-O, which is a high-level representation of the scene more or less compatible with the SMIL standard: authoring is simpler but you have less controls on objects you're creating, unless using XMT-A in XMT-O.

Simply said, a BIFS scene is described as a tree in which leaves are media object (text, 2D/3D objects, video, audio) and intermediate nodes define groups of objects, sub-tree transformation rules (2D space, 3D space, color etc.). The root of this tree defines the graphical context of the presentation, either a 2D world or a 3D world. For instance the *OrderedGroup* node allows grouping and ordered rendering of child nodes in the 2D plane. By default children appear in their declaration order: the first child is drawn in the background, the last on top of all others. The *OrderedGroup.order* field is used to modify this behavior. The XMT syntax is as shown in Figure 6.6. Nodes in the scene may have properties, called *field***s** that can be modified at run time. In XMT-A, a node is described by an XML element, and a field is described by an XML element or attribute, depending on the type of the field (elements for fields that can contain nodes, attributes otherwise).

```
<OrderedGroup order="0 1 2 3">
       <children> ... </children>
</OrderedGroup>
```

Figure 6.6: Ordered rendering of child nodes in the 2D plane in BIFS

BIFS has added several new concepts to VRML standard. BIFS is a compressed binary format; as a consequence, scenes are optimized in size. BIFS elementary streams are composed of access units that include BIFS commands to add new nodes, replace field values, and replace the whole scene tree. These commands enable single changes to the scene. BIFS-Anims streams enable structured changes to a scene. Together, these constructs add dynamic scenes. In other words, scenes are not static but can change over time. This allows for streaming of MPEG-4 scenes. Using this tool, an MPEG-4 terminal connected to a network can receive BIFS updates that modify completely the scene running on the terminal. This feature can be advantageous for adaptation purposes.

6.4.4 Madeus

As part of the Opera project [62], Nabil Layaïda, Muriel Jourdan, Cécile Roisin, Vincent Quint et al., define a multimedia document model called Madeus for describing multimedia scenarios [63] [64]. In Madeus, the description of a multimedia document is organized upon four dimensions: logical, temporal, spatial and hypermedia. In accordance with the idea of separating document information into different dimensions, in order to make it adaptive, the general structure of each document is decomposed in four main parts: MediaContent and MediaUse dimensions that describe the logical structure of the document, a Temporal dimension for synchronization between document parts, and a spatial dimension for layout. Since hypermedia information is closely related to interactivity, it is described in either the Temporal or MediaUse parts. An example of a Madeus XML source document is shown in Figure 6.7.

The *MediaContent* element contains raw media data, for instance, pixels of a picture, characters of a text, etc., and the intrinsic properties of the media, like the duration of a video or its size. The *MediaUse* element indicates a particular use of the content with specific style properties, for instance a line border color, a font size, etc. The content part may also be used to refine the media description. For instance, the content of a video can be structured in sequences, scenes, shots, etc. The logical model allows hierarchically organizing of media objects. As shown in Figure 6.8, a group element of type C-Group or U-Group plays an aggregate role. Its semantics depend on the document type and not on its presentation.

For instance, for this slideshow document, media content can be gathered by media types, and media uses can be gathered by the slide structure of the slideshow. A group element can define default values for some attributes of its children elements (for instance the color, the character size, etc.). The temporal model allows the organization of media objects over time.

The underlying model of this language is interval-based, meaning that each media object has a corresponding time interval characterized by a *begin*, a *duration* and an *end* attribute. Every MediaUse element is associated with a temporal *interval* element that carries all its temporal attributes required.

The spatial model allows spatial organization of media objects. Madeus defines a set of spatial vocabulary (left_align, bottom_spacing, etc.) for the two spatial dimensions. A spatial attribute cannot have indefinite value. More precisely, the spatial model organizes the document space as a 2D box hierarchy.

```
<Madeus>
        <MediaContent> ... </MediaContent>
        <MediaUse> ... </MediaUse>
        <Temporal> ... </Temporal>
        <Spatial> ... </Spatial>
</Madeus>
```

Figure 6.7: Overall structure of a Madeus document

```
<MediaContent> <!-- Content specification part -->
       <C-Group>
          <C-Group ID="Text" MIMEType="text/plain">
               <DefContent ID="ST1">Introduction</DefContent>
               <DefContent ID="ST2">Multimedia</DefContent>
          </C-Group>
          <C-Group ID="Video" MIMEType="video/mpg">
               <DefContent ID="Film" src="http://./Film.mpg">
                      <Scene StartFrame="0" EndFrame="20">
                             <Shot StartFrame="0" EndFrame="5"/>
                             <Shot StartFrame="5" EndFrame="20"/>
                      </Scene>
               </DefContent>
          </C-Group>
       </C-Group>
</MediaContent>
<MediaUse> <!-- Objects specification part -->
       <U-Group>
               <DefUse ID="U-Film" Content="Film" BorderWidth="1" BorderColor="black"/>
               <U-Group ID="TOC_entries" FontColor="black">
                      <DefUse ID="U-title1_toc" Content="SlideTitle1" FontSize="12"/>
                      <DefUse ID="U-title2_toc" Content="SlideTitle2" FontSize="12"/>
               </U-Group>
               <U-Group ID="Slide1" FontColor="blue">
                      <DefUse ID="U-title1" Content="SlideTitle1" FontSize="32"/>
                      <U-Group ID="U-Body1"> ... </U-Group>
               </U-Group>
               <U-Group ID="Slide2"> ... </U-Group>
       </U-Group>
</MediaUse>
```

Figure 6.8: A slideshow expressed in Madeus logical model

The hypermedia model allows description of links between elements or different parts of elements. It is based on XLink standard [65] and enhanced with temporal and spatial behavior.

Lionel Villard, Cécile Roisin et al. present a general framework for document production, which is based on Madeus model, and satisfies generic adaptation needs [66]. They use the Madeus model for generation of adaptable presentations. This presentation process is based on XSLT transformation techniques and constraint technologies for document formatting. The system suggests splitting the presentation generation into three steps. After encoding the content independently from its presentation using Madeus XML DTD, a transformation step is performed that allows obtaining a new representation, reflecting all the dimensions needed for presentation. This step allows adapting the content to the presentation device and to user preferences. A second simple transformation (a decoration) is then applied in order to adapt the media to the end user. Finally, the formatting process produces a representation playable by the presentation engine.

6.4.5 Other languages

Other than the above-described languages and models, there exist other multimedia scene description languages, among which, we can recall, HTML/XHTML, SVG and the new MPEG ongoing standard; LASeR (Lightweight Application Scene Representation) [67].

HTML is an SGML application conforming to International Standard ISO 8879 [68]. XHTML is a reformulation of HTML in XML. We do not really consider HTML/XHTML as "multimedia" presentation description languages, since HTML/XHTML do not specify temporal synchronizations and are rather defined for describing simple text-image documents.

SVG (Scalable Vector Graphics) is a modularized language, originally defined for describing two-dimensional vector and mixed vector/raster graphics in XML. SVG 1.1 does not define the description of media objects such as audio or video. SVG 1.2 will support (already under definition) media elements similar to the SMIL media elements. SVG 1.2 media elements are video, audio and animation.

LASeR is an emerging standard of MPEG group, which is dedicated to *lightweight* scene description of rich-media interactive and streamable services over mobile devices such as cell phones. The scene description of LASeR is based on SVG.

6.5 Existing approaches for scene adaptation

While numerous approaches have been adopted in the area of single media adaptation, less work has been done on the adaptation of multimedia scenes. MPEG-21 is dedicated to support resource adaptation and do not propose solutions to support structure adaptation of multimedia documents. In this section we describe some existing approaches in the area of adaptation of multimedia documents.

6.5.1 InfoPyramid Solution

Mohan et al. present very limited solutions for semantic adaptation of multimedia presentations, based on some incomplete semantic information, mainly on the *purpose* of image media objects [27]. The semantic information is not explicit and is derived from the original image object [69]. As discussed in Chapter 3, InfoPyramid rather aims at proposing solutions for resource adaptation and has not really solved the question of document adaptation.

6.5.2 Alternative-based scene adaptation

F. Rousseau et al., also propose solutions for the adaptation of synchronized multimedia presentations [70]. They use temporal extensions to HTML as a base for adding adaptation support. Through defining temporal extensions to HTML, and new elements and attributes, they propose means for specifying:

Alternate content: Elements that allow defining alternate media objects, temporal composition, and layout.

Content descriptions: A way of associating (semantic) metadata information with elements of a presentation (e.g. importance, keywords, duration).

Content predicates: A condition that selects alternate content or elements that match content descriptors.

One of the weak points of the approach is that it relies on the usage of media object alternatives and not media on-line adaptation. Also, it remains incomplete from the semantic point of view, i.e. the semantic relationships between media objects of the scene are not taken into account.

6.5.3 Temporal layout adaptation

J. Euzenat et al., present solutions for adaptation of multimedia documents only along the temporal dimension [71]. They do not discuss adaptation of the spatial layout and semantic dimension of the scene.

6.5.4 Semantic Web

In the area of *Semantic Web* [72], several research activities have been done on the ontology-based semantic description of Web documents based on RDF. As described in [72], the Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries. The Semantic Web activity is led by W3C. The concept is to have data on the web defined and linked in a way that it can be used by machines not just for display purposes, but for automation, integration and reuse of data across various applications.

Nagao propose external semantic annotation in order to make Web documents adaptable [73]. The proposed semantic information remains incomplete concerning dependencies between media objects.

6.5.5 Constraint-based layout for web documents

Borning et al. present a system architecture in which both the author and the viewer can impose page layout constraints, some required and some preferential [50]. The definition of these constraints allows the presentation designer to specify what are the desired spatial properties of the scene, rather than how these properties are to be maintained. Two system architectures, based on different ways of dividing the work of constraint solving between Web server and Web client, are identified. An authoring tool is proposed that provides linear arithmetic constraints for specifying the layout of the document as well as finite-domain constraints for specifying font size relationships.

The constraints capture the semantic of the design, those aspects that must hold true for the layout to be appealing. The constraints are given strengths. The strength label *required* means that the constraint must be satisfied. Other strength values denote the levels of preference. The model allows the designer to provide multiple *constraint style sheets*. Each style sheet includes constraints that define (a particular form of) the layout of the design and that dictate when the design is appropriate. The constraint-based document layout model has three main components: the *document authoring tool*, *the viewing tool, and the constraint solver*.

The principal contributions of this work are:

- A description of how constraints provide important functionality for web page layout and a constraint-based negotiation model for determining the web page appearance.
- An analysis of the requirements on the constraint solver, and of different ways to the constraint solving responsibilities can be partitioned between client and server.
- An implemented prototype that demonstrates the feasibility of the idea.

The objective of the work is to only adapt the layout to author and end-user preferences and constraints. The approach considers neither adaptations to usage and network context constraints, nor adaptation of media objects.

6.5.6 External annotation-based adaptation of HTML scenes

Hori et al, propose semantic annotation and adaptation of HTML documents [74]. The approach proposes external annotations to characterize ways of content adaptation rather than to describe individual content. External annotation files contain hints associated with elements in an original document. RDF is used as the syntax of annotation files. User profiles and context constraints are described using CC/PP. XPath [75] and Xpointer [76] are used for associating annotated portions of a document with annotations. The proposed annotation vocabulary includes three types of annotation: *alternatives*, *splitting hints* and *selection criteria* as described below. Further details on this vocabulary can be found in [77].

Alternatives: Alternative representations of a document or any set of its elements can be provided.

Splitting hints: An HTML file, which can be shown as a single page on a normal desktop computer, may be divided into multiple pages on clients with smaller display screens. The vocabulary provides a tag that specifies a set of elements to be considered as a logical unit.

Selection criteria: An annotation may also contain information to help an adaptation engine select from several alternative representations, the one that best suits the client device.

The execution sequence of the page-splitting module of the approach is depicted in Figure 6.9 can be briefly described as follows:

1. Upon reception of the request from a client browser, an original page is retrieved for the first time from a content server.

2. The editor component of the plug-in tries to find the locations of annotation files, if it does not succeed the original page is returned as it is and the session is terminated.

3. Taking in to account the client capabilities included in an HTML request header, the generator extracts a portion of a document object tree and returns a sub-tree to the client.

The authors show the results of applying their adaptation method to a particular form of web page (a news page from a corporate Web site). The news page consists of three tables stacked from top to bottom. The top and middle tables correspond respectively to a header menu and a search form. The bottom table, however, is used for laying out the content.

The *header* role annotation, which is assigned to a header element means that the page-splitting plug-in manages to keep the header element in every fragmented page.

In addition, if a high importance value is assigned to this element, this element should not be omitted in any case. For the Sidebar element, we can imagine that the role is annotated as *auxiliary*; therefore, this portion of the news page will be presented as a separate page upon receipt of a request from a small-screen device (activated by hyper linking).

The role of the bottom table in the news page can be annotated as *layouter*. This means that the bottom table will not be retained in the display for small-screen devices. If the search form table is annotated as a *low* importance element, it will be completely omitted.

Figure 6.10 illustrates how a news page is fragmented.

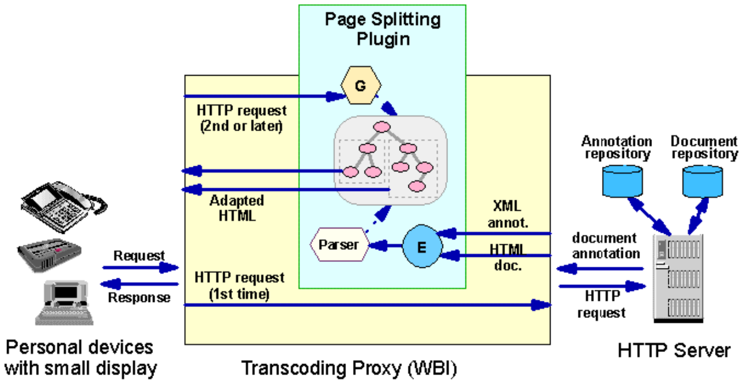


Figure 6.9: Hori's annotation-based adaptation by a page-splitting plug-in

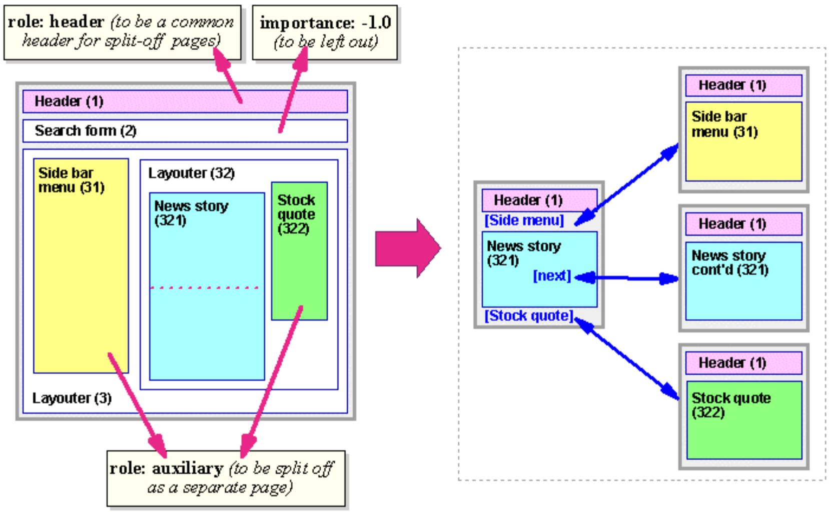


Figure 6.10: Hori's annotation-based fragmentation of an actual news page

This approach is interesting. However, compared to other types of synchronized multimedia documents, such as SMIL documents, HTML documents are by definition easier to adapt, since no complex timing synchronization is possible in HTML. Also another weakness of this work is that the adaptation strategy seems to be dependent on the spatial structure – news page form – of the document.

6.5.7 NAC solution for SMIL scene adaptation

Lemlouma T. and Layaïda N. propose a framework for SMIL content adaptation for embedded devices such as PDA, cellular phones [28] [78]. In order to adapt the SMIL content to these devices, the content access passes through an intermediate entity called adaptation proxy where the adaptation occurs. They propose to add semantic metadata related to the content. This additional information is directly related to the adaptation mechanisms and includes: the SMIL internal relationships defined between different objects of the content and additional metadata that can relate SMIL objects with other external objects available at the server side. The proposed adaptation process includes: version selections, document transformation and media adaptation as described below.

6.5.7.1 Versions selections

This kind of adaptation consists to choose the best variant of the multimedia content or object on behalf of the user agent. Generally the selection is applied on the available variant set and based on variants description and the user requirements. Selection criteria may include the language, the media type, the char-set, etc. The adaptation proxy processes the SMIL switch element and chooses, when possible, the first acceptable element. An element (a timing structure or media object) with no test attribute is always acceptable. The content selection can be expressed using the SMIL system test. Adaptation proxy evaluates the test using the information extracted from the different user profiles.

The variants selection is applied and the SMIL document is restructured by eliminating the switch elements and the non-selected alternatives. They also use SMIL in-line test attributes, to apply elements selection. In this situation, the adaptation proxy acts in the same manner as with the switch element.

6.5.7.2 Documents transformation

This kind of adaptation concerns transformations that are applied on the global document organization or tree and can modify its structure. The SMIL structural transformation applied by the proxy can either keep the same media resource used by the original SMIL document, filter it or require an external media resource transformation to adapt the media for the end user context. Document manipulations are generally applied using XSLT.

6.5.7.3 Media resources adaptation

In order to make media objects available to the client and adapted to the environment characteristics such as the mobility and low available bandwidth, media resources shall be substituted, removed or transformed to an acceptable format using available adaptation techniques and methods. In this approach a limited set of semantic vocabulary is introduced that expresses two limited media relationships: the *equivalent* and *adapted to* relations. An *equivalent* relation means simply that the two related objects have the same role in the multimedia presentation. An *adapted to* relation means that the target object represents a constrained version of the content. The approach proposes usage of a semantic parameter indicating the level of the adaptation can be associated with the *adapted to* relation. This can be exploited in the content negotiation strategy if a media resource can be adapted to more than one object. The adaptation of media objects referenced by the SMIL content includes video and image adaptation. Video adaptation considers the characteristics of the wireless network and client device screen dimensions and color resolution. Image adaptations include image compression and resizing.

The approach is interesting but remains incomplete, limited and unclear in the area of semantic adaptation of the temporal and spatial layout of SMIL documents. The semantic, temporal and spatial dependencies between media objects are not completely taken into account.

6.5.8 SMIL adaptable documents for mobile devices

R. Steele et al. present their work on a SMIL-based system for mobile device for dynamic adaptation of presentations in response to changing mobile computing-related factors [79]. The methodology used in this research considers two alternatives for delivering a multimedia presentation to a mobile device: as a monolithic multimedia file or as a SMIL-based presentation. They identify the assumed mobile computing architecture (e.g. WiFi [80] and GPRS [81] connectivity), as well as aspects that effect the access of multimedia presentations from mobile devices and devise techniques that make use of SMIL to improve such access to multimedia presentations.

Each mobile device has an agent that will manage the retrieval and display of the SMIL-based multimedia presentation. An agent dynamically manages aspects of multimedia resource retrieval such as what files to retrieve and when, by following the statements of the SMIL file but also making some decisions based on its knowledge of the current state of the mobile device and wireless connection.

As a mobile device passes through different access networks, with different access parameters, the system tries to find the location of the closest copy of the requested multimedia file, which will then be returned to the requesting mobile device. So the proposed solution is to retrieve the requested file from the closest/ fastest source.

If the presentation provider has prepared the presentation in such a way that its elements are decomposed into smaller parts referenced separately in the presentation description, the chance that the download of a particular file would be interrupted mid-stream by a change in connectivity is decreased. Accessing the source for which there is the “best” network link decreases possible transmission disturbance and as a result, improves the presentation quality.

In addition if the presentation provider has provided different alternative multimedia files, these alternative are then used depending on the current network connectivity at the mobile device’s location e.g. WiFi, GPRS. This could is done by using the SMIL *switch* element. As a mobile device moves, the sequence of definitions of multimedia resources in the SMIL file dictates what new multimedia files will be requested. The requests are performed by the device-based agent, which contacts the proxy server to request the file from the most appropriate server – this server may differ from the server previously used.

The approach aims at optimizing the access of mobile devices to media files specified in a SMIL document. It does not concretely treat the question of document structure and media object adaptation.

6.5.9 RIML

In the context of devices independent delivery, some existing languages offer many advantages and adaptability facilities. Among these languages are: XHTML that combines the advantages of both XML and HTML, SMIL with a whole module (content control module) dedicated to support device heterogeneity and the adaptation for different delivery contexts, XEvents for user interactions [82], XForms for processing forms [83], etc. Research works have been and are being done to define a language that would be completely device independent and that integrates all the aspects of content adaptability [84]. The goal is to specify the description of the application user interface in an XML-based mark-up language to enable a rendering engine to adapt the complexity of the application input and output according to established devices classes. This language is called Renderer-Independent Markup Language (RIML) [85][86].

The main objective of the CONSENSUS project is to develop technology that enhances the mobile device usage of enterprise applications and provides a high level of usability [87]. One of the primary project objectives is the specification of a Renderer-Independent Markup Language (RIML) for defining such application user interfaces. Documents written in this language can be transformed into target languages, creating multiple versions of the original that are suitable for a wide range of visual browsers on mobile devices and voice browsers. Inclusion of application-level clues enables this efficient and automated transformation to take place, ensuring that the final output on targeted devices and browsers delivers a highly usable product.

RIML is an XML-based language that combines features of several existing markup languages. RIML is an XHTML language profile; XHTML serves as the host language of the new language profile. The RIML profile specification defines which parts of XHTML are used and how parts of XForms, XFrames, SMIL BasicContentControl and other new functionality is integrated into the new language [88]. XForms 1.0 Candidate Release is almost the final version of XForms, the new technology for forms developed by the W3C. XFrames is a specification on frames that addresses many of the shortcomings of the currently used mechanisms. SMIL BasicContentControl is a small part of the SMIL 2.0 Recommendation that is suitable for managing conditional content in any XML document. According to [89], some of the main features of RIML are:

➢ RIML Focus on device-independent visual and voice user interface description. RIML can be used for defining the user interface of an enterprise application in a device-independent way. By including application-level hints, RIML enables automated adaptation to specific devices and visual and voice browsers to generate highly usable results. RIML does not define business logic, but tries to be compatible with the business logic of already existing applications for desktop clients.

➢ Multiple layouts: The author can define multiple overall page layouts in a single RIML document. The most suitable layout for a specific device and browser can be selected during the transformation of the RIML document into the target language.

➢ Pagination and navigation: When a RIML document is too large to fit on a single page, it will be paginated. The division into multiple pages and means for navigating among these pages are generated automatically from application-level hints provided by the author.

➢ Alternative and optional content: RIML allows for alternative variants of the same content and also optional content through RIML Extended Content Control. This functionality, which is an extension of SMIL BasicContentControl mechanisms, defines additional so-called test attributes, which the author can use for annotating optional and alternative content. By defining the conditions that refer to the delivery context, the author can provide guidance to the automated adaptation.

➢ Content Control test attributes of the technical device level and the usability level: Some of the Extended Content Control test attributes refer to the technical capabilities of devices and browsers. Inclusion of these technical attributes addresses the syntactic adaptation issues in the final step of the target language generation. Other attributes refer to usability characteristics of a device, especially those related to the user's interaction with browser-rendered content. Such attributes could potentially influence the semantic adaptation process, introducing a new method of impacting the usability level.

Gabriel Dermler et al. completed the first specification of RIML and the design of the adaptation system needed to transform RIML into HTML, WML and XHTML MP markup [90].

RIML stresses the separation of content definition (i.e. what is to be presented) from the description of dynamic adaptations, which can be performed on the content in order to match varying capabilities of devices. Special row and column structures are used in RIML to specify content adaptation. Their semantics is enhanced to cover pagination and layout directives in case pagination needs to be done. Automated pagination support was a main design goal for RIML.

The overall layout of a RIML document is defined by using elements defined by the RIML layout modules. The layout modules define a hierarchy of containers in which the content should be placed. These containers are row, column, grid and frame. The adaptation system will use suitable mechanisms of the target language to accomplish this layout whenever and wherever possible. The layout definition of RIML is described in [89].

6.6 Conclusion

In this chapter we reviewed the basic concepts and notions of multimedia scene adaptation. Principal requirements and elements of a multimedia document adaptation system were described as well. We then reviewed and discussed different approaches that have been adopted in this area. We saw that none of these approaches has succeeded to propose a complete framework for adaptation of rich structure multimedia document and that most of them suffer from lack of semantic adaptation techniques.

In our approach, we propose a semantic and context-aware multimedia adaptation framework based on MPEG-21. The methodology guarantees consistence, coherence and meaningfulness of the adapted document layout.

Next chapter describes, in detail, the methodology of our approach, while the architecture of our evaluation implementation of this approach is explained in Chapter 8.

Chapter 7

MULTIMEDIA SCENE SEMANTIC ADAPTATION (MSSA)

Summary

This chapter describes our methodology for the realization of a multimedia scene semantic adaptation framework. We describe our solutions for each element of such framework as defined in previous chapter.

Table of Content

Table of Illustrations

7.1 Introduction

Mobile devices such as PDAs (Personal Digital Assistant) represent an increasing proportion of today online content access. These devices have reduced display capabilities, therefore, we argue that an entity should be inserted into a multimedia delivery chain to take care of providing the needed metadata and attaching it to the content, for presentation adaptation to the context constraints such as device display capabilities. Of course, as these access devices are used by users, each having different preferences, and over networks having different characteristics, the layout adaptation is further more constrained. Within a multimedia content delivery chain, the above-mentioned entity should be placed after the multimedia-authoring step, or integrated to the authoring task. In other words, from an adaptation point of view, it is preferable and sometimes required that the designer of a multimedia system specifies how the structure of presentation should evolve based on the change of environments (e.g., from a desktop screen to a mobile display panel).

We propose and present a semantic and context-aware multimedia adaptation framework based on MPEG-21. The methodology guaranties consistence, coherence and meaningfulness of the adapted document layout. We introduce and use description of semantic information of a multimedia presentation, in order to make it adaptable for different contexts. The process of adaptation of each resource and of the whole multimedia presentation takes into account the constraints of the context as well as the semantic metadata of the multimedia content for both single media objects and the whole presentation. Our Multimedia Scene Semantic Adaptation (MSSA) framework addresses the adaptation of multimedia-structured documents based on temporal, spatial and semantic relationships between the media objects.

Please notice that we do not mean to deal with the architectural aspect of a semantic multimedia adaptation system. In other words, we neither mean to discuss the architecture of the underlying network (client, server and proxy-related issues), nor to address the underlying messaging and transport protocol. We consider that these issues are out of the scope of the presented work, which rather aims to propose approaches for different issues of a multimedia adaptation *engine* – by investigating new solutions such as MPEG-21 – and to, then, develop a framework for an experimental or proof-of-concept implementation.

We have chosen SMIL for description of multimedia documents. However the methodology is independent from this choice and the approach can be applied to other multimedia description languages.

As analyzed in Chapter 6, principal elements of a multimedia presentation adaptation system are:

- Content physical description,
- Context description,
- Content semantic description,
- Scene semantic adaptation core:
    a. Scene optimization
    b. Scene adaptation:
        i. Presentation structure adaptation tools
        ii. Resource adaptation tools

This chapter provides a detailed description of the methodology of our approach for realization of a framework for semantic adaptation of multimedia presentations. We first argue our choice of SMIL for the description of multimedia documents. Then, for each principal element of such framework as recalled above, we describe our approach and propose solutions. We start by describing our solution for the expression of metadata, namely semantic information of multimedia scenes. Through some examples, we also describe how we use MPEG-21 DIDL for embedding the semantic metadata of scenes. We then present the basic concepts of our scene semantic adaptation core and provide a brief description on the algorithm of this module. Our implementation of MSSA framework is detailed in next chapter.

7.2 Scene description: why SMIL?

As different multimedia description languages have different adaptation-related features, the choice of scene description language is an important issue in the context of a multimedia scene adaptation framework. In our approach, we use SMIL 2.0 for describing scenes; nevertheless, the methodology of the work is independent from this and can be applied to other multimedia description languages. We chose SMIL 2.0 since:

- SMIL is a “high level” scene description language. This facilitates performing adaptations on a SMIL scene, compared to, say, performing an adaptation on an XMT-A scene.

- SMIL allows complete separation of the spatial layout of the scene from its temporal and structural layout and also from the content itself (i.e. media objects). This facilitates independent manipulation of spatial layout, as well as independent manipulation of media objects.

- By defining concepts such as switch element and test attributes, SMIL allows adaptability by means of alternative replacing. Of course, pre-providing media alternatives (which could be used under different context constraints) is in general neither a good idea, nor a desired solution for the question of multimedia adaptation. This is due to the fact that it is not logic to expect the author to provide different versions and variants of his/her resources for a large number of contexts. Furthermore, an alternative-based adaptation system may face a context for which no variant of the resource is provided. A better solution for multimedia content adaptation would certainly be to create only one version of the content, and to then develop on-line adaptation structures that would adapt the content on-the-fly, for each particular context. Nevertheless, alternative-based off-line adaptation could some times be of interest. This is useful when on-line creation of a variant is not possible. For instance, online conversion of a video (with no subtitle and no audio) to text is not possible unless a text alternative is present.

- SMIL provides flexibility thanks to modularization of the language profiles. SMIL Basic profile is ideal for limited devices.

- SMIL can be employed by Web site creators to specify how media elements (video, sound, still images) can be presented and played in sequence or parallel as part of a Web presentation.

Although SMIL 2.0 has rather a very well-defined specification, working with SMIL 2.0 has some disadvantages, namely:

- There exist no clean, fully open and fully customizable SMIL player that can be used as a common research and development resource. RealNetworks RealPlayer is available for nearly all desktop environments [91]. Apple uses SMIL 1.0 in QuickTime players [92]. Microsoft's implementation of XHTML+SMIL in Internet Explorer 6 also has a wide reach [93]. There are many other SMIL players, such as Oratrix's GRiNS SMIL 2.0 player [94] and Ambulant [95]. And yet, none of the existing commercial players provides a complete and correct SMIL 2.0 implementation. The commercial players that exist are geared to the presentation of proprietary media.

➢ The existing SMIL players are not interoperable, meaning that, playing the same SMIL content by two different players, e.g. Ambulant and RealPlayer, will not always result in a same layout. List of existing SMIL players is available on the W3C SMIL site [96].

Among SMIL players who run on embedded devices such as Pocket PCs or PDAs, there is SMIL Player by InterObject that supports SMIL 2.0 Basic Profile [97]. This player is claimed to run on PC with Windows NT/2000/XP and handheld devices with Pocket PC, such as Compaq iPAQ. PocketSMIL 2.0 is another SMIL 2.0 Player for the Pocket PC [98]. It supports SMIL2.0 Basic profile.

A problem that happens quite often when working with SMIL players is that they do not play correctly – or do not play at all – some SMIL content, which are however conformant to SMIL 2.0 specification. This is because most of SMIL players do not support the whole W3C SMIL 2.0 specification.

Currently, the most used SMIL 2.0 Player around the world is RealPlayer (or RealOnePlayer). We have chosen to mainly test our SMIL content with RealPlayer, as it is the most complete SMIL player available for free. RealPlayer plays rich SMIL 2.0 content as well as SMIL 2.0 Basic Profile content, which is adequate for devices with reduced resources. Besides, RealNetworks provides a relatively complete SMIL content production guide that describes SMIL content and elements that can be played in RealPlayer.

Our adapted SMIL documents are conformant to SMIL 2.0 Basic Profile and 3GPP SMIL Language Profile [99], and are therefore, expected to be played by any SMIL player for embedded devices. In addition, the capabilities of the player in terms of supported types and formats of media are taken into account by means of DIA UED descriptors that describe terminal capabilities.

7.3 Expression and interpretation of metadata

This section describes how physical content description, context description, and content semantic are expressed and interpreted within MSSA framework.

7.3.1 Content physical description

The same as for our resource conversion framework, we use MPEG-7 descriptors for explicit description of the content (physical content description) in the MSSA framework.

7.3.2 Context description

The description of context is given by MPEG-21 UED descriptors. We do not use MPEG-21 UCD for further constraining; but instead we attribute a static priority to each context constraint. This way the constraints are satisfied in their order of priority.

7.3.3 Semantic Information Declaration (SID)

Our semantic adaptation system requires an in-depth understanding of the document. It, hence, needs human intervention. The semantic information could be either provided by the author of the document or by any other editor or entity. We have defined XML schemas for the expression of semantic information of a multimedia scene. In the corresponding CDI, these descriptors are given in a *Statement* element. The SID (Semantic Information Declaration) descriptors are used by the adaptation engine to decide on the type, nature and parameters of the adaptation(s) to be applied to the scene.

Figure 7.1 shows the structure of a complete CDI containing a reference to the considered SMIL scene, its SID descriptors, references to each media object of the scene and their alternatives (if present) and or RCD (Resource Conversion Description) descriptors (if present).

```
<DIDL >
    <Item> <!-- item containing scene and SID!-->
        <Descriptor><!-- SID descriptors of the scene !- ->
            ...
        </Descriptor>
        <Component ><!--reference of the smil scene !-- >
            ...
        </Component>
    </Item>
    <Item> <!-- media objects and RCD and alternatives!-->
        <Item > <!–one media different alternatives and required conditions !-->
            <Choice >
                ...
            </Choice>
            <Component>
                <Condition .../>
                <Resource .../>
            </Component>
            <Component>
                <Condition .../>
                <Resource .../>
            </Component>
        </Item>
        <Item> <!–one media object !-->
            <Component>
                ...
            </Component>
        </Item>
        <Item> <!–one media object!-->
            <Component>
                ...
            </Component>
        </Item>
          . . .
    </Item>
</DIDL>
```

Figure 7.1: General structure of a complete CDI

The information included in SID descriptors is categorized into three main parts: independent semantic information of each media object, semantic dependencies between media objects of the scene, and semantic preferences on scene fragmentations.

In order to better understand these semantic metadata, we consider the SMIL scene described in Figure 7.2. This SMIL content is originally designed to be played on a desktop device. If played by RealPlayer 10, the layout will be as shown in Figure 7.3. Here we have reduced the overall size of the window in order to fit it into the document.

We now proceed to explain our semantic information descriptors.

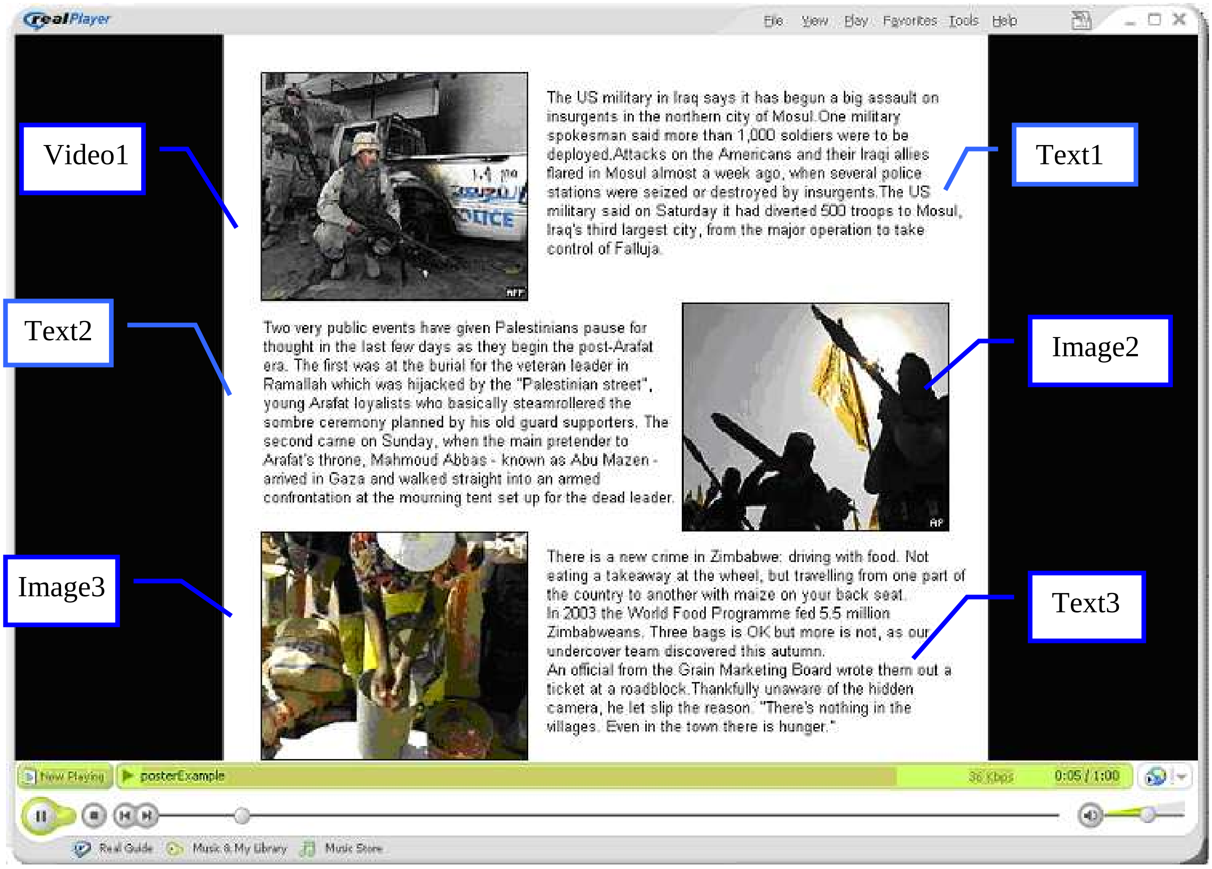
7.3.3.1 Independent semantic information

This category of semantic information describes, for each media object, its independent semantic information, in the context of the scene, such as its importance, role and maximum authorized resolution reduction. These parameters are provided as XML attributes in the CDI that contains the SID descriptors.

The *importance* attribute describes the semantic importance of the corresponding media object. It can have one of *high*, *medium* and *low* values. The attribution of usage environments resources such as display size or network bandwidth can be defined to be proportional to the value of media importance. While adaptation of *high* and *medium* importance media objects are relatively privileged, *low* importance media objects may be omitted from the scene in case of limited usage environments resources.

```
<smil>
 <head>
  <layout>
    <root-layout width="1000" height="950" background-color="#ffffff"/>
      <region id="video1" fit="fill" width="350" height="300" left="50" top="50"
                                                                   background-color="#ffffff"/>
      <region id="image2" fit="fill" width="350" height="300" left="600" top="350"
                                                                   background-color="#ffffff"/>
      <region id="image3" fit="fill" width="350" height="300" left="50" top="650"
                                                                   background-color="#ffffff"/>
      <region id="text1" fit="fill" width="550" height="280" left="420" top="70"
                                                                   background-color="#ffffff"/>
      <region id="text2" fit="fill" width="550" height="280" left="50" top="370"
                                                                   background-color="#ffffff"/>
      <region id="text3" fit="fill" width="550" height="280" left="420" top="670"
                                                                   background-color="#ffffff"/>
  </layout>
 </head>
 <body>
  <par>
    <video id="v1" src="v1.jpg" dur="60s" region="video1"/>
    <text id="t1" src="t1.txt" dur="60s" region="text1"/>
    <img id="i2" src="i2.jpg" dur="60s" region="image2"/>
    <text id="t2" src="t2.txt" dur="60s" region="text2"/>
    <img id="i3" src="i3.jpg" dur="60s" region="image3"/>
    <text id="t3" src="t3.txt" dur="60s" region="text3"/>
  </par>
 </body>
</smil>
```

Figure 7.2: An example of a SMIL 2.0 content



The US military in Iraq says it has begun a big assault on insurgents in the northern city of Mosul.One military spokesman said more than 1,000 soldiers were to be deployed.Attacks on the Americans and their Iraqi allies flared in Mosul almost a week ago, when several police stations were seized or destroyed by insurgents.The US military said on Saturday it had diverted 500 troops to Mosul, Iraq's third largest city, from the major operation to take control of Falluja.

Two very public events have given Palestinians pause for thought in the last few days as they begin the post-Arafat era. The first was at the burial for the veteran leader in Ramallah which was hijacked by the "Palestinian street", young Arafat loyalists who basically steamrollered the sombre ceremony planned by his old guard supporters. The second came on Sunday, when the main pretender to Arafat's throne, Mahmoud Abbas - known as Abu Mazen - arrived in Gaza and walked straight into an armed confrontation at the mourning tent set up for the dead leader.

There is a new crime in Zimbabwe: driving with food. Not eating a takeaway at the wheel, but travelling from one part of the country to another with maize on your back seat.
In 2003 the World Food Programme fed 5.5 million Zimbabweans. Three bags is OK but more is not, as our undercover team discovered this autumn.
An official from the Grain Marketing Board wrote them out a ticket at a roadblock.Thankfully unaware of the hidden camera, he let slip the reason. "There's nothing in the villages. Even in the town there is hunger."

Figure 7.3: Layout of the SMIL content of Figure 7.2 in RealPlayer 10

The *role* attribute describes the semantic role of the corresponding media object. It can have one of *key*, *key-convertible*, *redundant* and *decorative* values. For instance if a media object has a *key* role, it means that under no circumstance, it should be either converted (i.e. adapted, downscaled, etc.) or removed from the scene. Hence if any adaptation is needed and if the deletion of this object is necessary for the adaptation, the adaptation will not be possible. A media object of *key-convertible* role should under any circumstances stay in the scene, either under its original or converted form. A media object of *redundant* role is meant to be redundant for another media object present in the scene. In this latter case the media object, for which, this media object is a redundant, should be indicated within the descriptor that contains media objects semantic dependencies (*redundantFor* element). A media object of *decorative* role is understood as a decorative media that is not a part of the essence of, or information included in the communicated content. Advertisements are of such media role type.

For the visual media objects such as image, graphics and video, another attribute describing the preferred or authorized maximum spatial downscaling, called *maxRRF* (maximum Resolution Reduction Factor) is also categorized under this category of semantic information.

Figure 7.4 shows the CDI *Item* containing these SID attributes for the SMIL scene of Figure 7.2 and Figure 7.3. The media objects are mapped to media objects of the SMIL scene – which is referenced through this CDI – via *ref* attributes, however this could also be done by using XPath expressions. The corresponding SMIL scene of this example has six media objects: one video, two images, and three texts.

7.3.3.2 Media objects semantic dependencies

This category of semantic information includes absolute semantic dependencies, spatial semantic dependencies, and synchronization semantic dependencies between media objects of a scene.

The absolute semantic dependencies describe the presence of which media objects are a precondition for the presence of another media object and which text media objects are closed captions for another media.

The synchronization semantic dependencies provide the preferred semantic synchronization information between media objects if a temporal fragmentation is needed. The spatial semantic dependencies describe which media objects should be kept close together in case of a spatial rearrangement of the scene. Also if within a scene, a visual media object has a caption text, this dependency can be expressed within spatial semantic dependencies. Figure 7.5 shows the complete CDI of Figure 7.4. As can be seen in the *Item* containing SID descriptors, the presence of the media object t1 (text1) is a precondition for the presence of media object v1 (video1) in the presentation.

```
<Item id="posterExample">
    <Descriptor>
      <Statement mimeType="text/plain">
        this is a digital item declaration for a smil multimedia presentation:
        posterExample.smil, which contains 4 media. This did contains also semantic
        metadata of the presentation
      </Statement>
    </Descriptor>
    <Descriptor>
      <Statement mimeType="text/xml">
       <sid:SID>
         <sid:Object id="video1" ref="v1" importance="high" maxRRF="2" role="key">
              <!--  semantic information of this media object through the scene!-->
         </sid:Object>
         <sid:Object id="text1" ref="t1" importance="medium"  maxRRF="3"
                                                                  role="key-convertible">
              <!--  semantic information of this media object through the scene!-->
         </sid:Object>
         <sid:Object id="image2" ref="i2" importance="low" maxRRF="2" role="redundant" >
              <!--  semantic information of this media object through the scene!-->
         </sid:Object>
         <sid:Object id="text2" ref="t2" importance="medium" role="key-convertible">
              <!--  semantic information of this media object through the scene!-->
         </sid:Object>
         <sid:Object id="image3" ref="i3" importance="low" maxRRF="2" role="redundant" >
               <!--  semantic information of this media object through the scene!-->
         </sid:Object>
         <sid:Object id="text3" ref="t3" importance="low" role="key-convertible">
               <!--  semantic information of this media object through the scene!-->
         </sid:Object>
       </sid:SID>
      </Statement>
    </Descriptor>
    <Component id="smilScene">
       <Resource mimeType="application/smil" ref="posterExample.smil"/>
    </Component>
</Item>
```

Figure 7.4: Independent semantic information attributes for SMIL scene of Figure 7.2

We can also see that the media object v1 should be spatially kept close to, and, temporally synchronized with t1. As can be seen in the *Item* containing media-related information (second *Item*), for the video source, parameters of video-to-image transmoding are given in RCD descriptors. Also for text1 media, whose *maxRRF* is 3, the alternative summarized text is given.

In this example, we can see that some metadata expressions are redundant. For example, semantic dependency between v1 and t1 is expressed once within the corresponding *Object* element of v1 and another time within the *Object* element of t1. This redundancy could be eliminated by expressing the metadata in only one *Object*. However, this redundancy will not cause any problem at the level of implementation unless the expressed metadata do not match to each other.

Note— Exactly like *PreconditionMedia* element in *AbsoluteSemanticDependencies*, closed captions media objects are indicated in an XML element *CaptionMedia*.

7.3.3.3 Semantic preferences on scene fragmentation

This third category of semantic information describes semantic preferences and priorities on eventual spatial and temporal fragmentation of scene.

The example of Figure 7.5 expresses that if a scene fragmentation is necessary, when arranging the fragments timings, the media object v1 is preferred to be placed in the first fragment (shiftingPriority="1"), and in arranging the spatial layout of fragments, v1 shall be kept in the same fragment as t1.

It should be noted that general spatial semantic preferences are different from fragmentation spatial preferences. In example of Figure 7.5, the media object v1 should in general be kept close to t1. This is a general remark, meaning that in rearranging the spatial layout of the scene, even if no fragmentation is needed for adaptation, v1 should be kept close to t1. On the contrary, the semantic spatial fragmentation preferences, express that in case of scene fragmentation, v1 shall be kept in the same fragment as t1.

The schema, syntax and semantics of SID descriptors are provided in Annex C.

7.3.4 Interpretation and usage of SID metadata

As mentioned earlier in this chapter, the MSSA engine uses the SID descriptors in order to find (decide), and then perform the optimal adaptation form for the considered scene. The scene optimizer entity is in charge of finding (deciding on) the type and nature of the adaptation(s) as well as the values of its (theirs) required parameters. The scene adaptor entity then performs this (these) adaptation(s) and outputs the adapted scene.

```
<?xml version="1.0" encoding="UTF-8"?>
<DIDL xmlns="urn:mpeg:mpeg21:2002:01-DIDL-NS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:mpeg:mpeg21:2003:01-DIDL-NS didl.xsd"
xmlns:sid="urn:mpeg:mpeg21:2004:01-SID-NS">
 <Item id="smilscene1">
   <Descriptor>
     <Statement mimeType="text/xml">
       <sid:SID>
         <sid:Object id="video1" importance="medium" ref="v1" maxRRF="2">
           <sid:SemanticDependencies>
             <sid:AbsoluteSemanticDependencies>
               <sid:PreConditionMedia>
                 <sid:MediaObject ref="t1"/>
               </sid:PreConditionMedia>
             </sid:AbsoluteSemanticDependencies>
             <sid:SynchronizationSemanticDependencies synchronizedTo="t1"/>
             <sid:SpatialSemanticDependencies>
               <sid:KeepCloseTo>
                  <sid:MediaObject ref="t1"/>
               </sid:KeepCloseTo>
             </sid:SpatialSemanticDependencies>
           </sid:SemanticDependencies>
           <sid:SemanticFragmentationPreferences>
             <sid:TimeFragmentationPreferences shiftingPriority="1"/>
             <sid:SpatialFragmentationPreferences keepWith="t1" position="left"/>
           </sid:SemanticFragmentationPreferences>
         </sid:Object>
         <sid:Object id="text1" ref="t1" importance="medium" maxRRF="3" role="key-convertible">
           <sid:SemanticDependencies>
             <sid:SpatialSemanticDependencies>
               <sid:KeepCloseTo>
                 <sid:MediaObject ref="v1"/>
               </sid:KeepCloseTo>
             </sid:SpatialSemanticDependencies>
           </sid:SemanticDependencies>
         </sid:Object>
         <sid:Object id="image2" ref="i2" importance="low" maxRRF="2" role="redundant">
           <sid:SemanticDependencies>
             <sid:AbsoluteSemanticDependencies>
               <sid:RedundantFor>
                 <sid:MediaObject ref="t2"/>
               </sid:RedundantFor>
             </sid:AbsoluteSemanticDependencies>
             <sid:SpatialSemanticDependencies>
               <sid:KeepCloseTo>
                 <sid:MediaObject ref="t2"/>
               </sid:KeepCloseTo>
             </sid:SpatialSemanticDependencies>
           </sid:SemanticDependencies>
         </sid:Object>
         <sid:Object id="text2" ref="t2" importance="medium" role="key-convertible">
           <sid:SemanticDependencies>
             <sid:SpatialSemanticDependencies>
               <sid:KeepCloseTo>
                 <sid:MediaObject ref="i1"/>
               </sid:KeepCloseTo>
             </sid:SpatialSemanticDependencies>
           </sid:SemanticDependencies>
          </sid:Object>
          <sid:Object id="image3" ref="i3" importance="low" maxRRF="2" role="redundant">
            <sid:SemanticDependencies>
              <sid:AbsoluteSemanticDependencies>
                 <sid:RedundantFor>
                   <sid:MediaObject ref="t3"/>
                 </sid:RedundantFor>
              </sid:AbsoluteSemanticDependencies>
              <sid:SpatialSemanticDependencies>
                <sid:KeepCloseTo>
                  <sid:MediaObject ref="t3"/>
                </sid:KeepCloseTo>
              </sid:SpatialSemanticDependencies>
            </sid:SemanticDependencies>
           </sid:Object>
           <sid:Object id="text3" ref="t3" importance="low" role="key-convertible">
             <sid:SemanticDependencies>
               <sid:SpatialSemanticDependencies>
                 <sid:KeepCloseTo>
```

```
                  <sid:MediaObject ref="i3"/>
                </sid:KeepCloseTo>
              </sid:SpatialSemanticDependencies>
            </sid:SemanticDependencies>
          </sid:Object>
        </sid:SID>
      </Statement>
    </Descriptor>
    <Component id="smilScene">
       <Resource mimeType="application/smil" ref="posterExample.smil"/>
    </Component>
  </Item>
  <Item>
    <Component id="v1">
      <Descriptor>
        <Statement mimeType="text/xml">
         <dia:ConversionInformation>
           <dia:ConversionDescription xsi:type="rcd:TransmodingConversionType">
             <dia:ConversionUri>http://www.example.fr/vi</dia:ConversionUri>
             <rcd:Transmoding quality="1.0" >
               <rcd:TransmodingParameters xsi:type="VideoSummarizationParametersType">
                 <rcd:To href="urn:mpeg:mpeg7:cs:ContentCS:2001">
                    <mpeg7:Name>Image</mpeg7:Name>
                 </rcd:To>
                 <rcd:Slide importance="hight">
                   <mpeg7:MediaTimePoint>T01:14:30:12F24</mpeg7:MediaTimePoint>
                 </rcd:Slide>
               </rcd:TransmodingParameters>
             </rcd:Transmoding>
           </dia:ConversionDescription>
         </dia:ConversionInformation>
        </Statement>
      </Descriptor>
      <Resource mimeType="video/mpeg" ref="v1.gif"/>
    </Component>
  </Item>
  <Item>
    <Item id="t1">
      <Choice minSelections="1" maxSelections="1">
        <Descriptor>
          <Statement mimeType="text/plain"> What resolution? </Statement
        </Descriptor>
        <Selection select_id="originalSize">
          <Descriptor>
            <Statement mimeType="text/plain">original text</Statement>
          </Descriptor>
        </Selection>
        <Selection select_id="maxRR">
          <Descriptor>
            <Statement mimeType="text/plain">maximum downscaled text</Statement>
          </Descriptor>
        </Selection>
      </Choice>
      <Component>
        <Condition require="originalSize"/>
        <Resource ref="t1.txt" mimeType="text/plain"/>
      </Component>
      <Component>
        <Condition require="maxRR"/>
        <Resource ref="summarizedT1.txt" mimeType="text/plain"/>
      </Component>
    <Item>
      <Component id="i2">
        <Resource mimeType="image/gif" ref="i2.gif"/>
      </Component>
    </Item>
    <Item>
      <Component id="t2">
        <Resource mimeType="text/plain" ref="t2.txt"/>
      </Component>
    </Item>
    <Item>
      <Component id="i3">
        <Resource mimeType="image/gif" ref="i3.gif"/>
      </Component>
    </Item>
    <Item>
```

```
       <Component id="t3">
         <Resource mimeType="text/plain" ref="t3.txt"/>
       </Component>
     </Item>
   </Item>
</DIDL>
```

Figure 7.5: Expressing semantic dependencies for the SMIL scene of Figure 7.2

In this section we describe the rules we set for interpretation of SID metadata by our MSSA scene optimizer. These interpretation rules shall be respected by the MSSA scene optimizer.

7.3.4.1 Media object presence rules

Media objects having a *PreConditionMedia*, may be present in the scene, only if their *PreConditionMedia* object is present in the scene.

7.3.4.2 Media object adaptation rules

Only non-*key role* media objects can be subject to adaptation (conversion).

7.3.4.3 Media object omitting rules

Not all media objects could be subject to elimination from the presentation. Media objects of *decorative role* may be omitted from the scene, independently of their *importance* value. Between several *redundant role* media objects of different *importance* values, that are redundant for each other, those with lower *importance* may be omitted. *Low importance* media objects of non-specified *role* could also be subject to deletion. A *CaptionMedia* media should be omitted when its corresponding visual media is omitted. A media object should be omitted when its corresponding *PreConditionMedia* object is eliminated.

7.3.4.4 General semantic spatial and temporal organization rules of media objects

A media object should be temporally synchronized to its corresponding *synchronizedTo* object. A media object must be kept close to its corresponding *KeepCloseTo* media object. A visual media should be spatially kept close to, and, temporally synchronized to its *CaptionMedia* media object.

7.3.4.5 Scene fragmentation rules

Media objects related by *keepWith* should be kept in the same scene fragment. The *position* should be respected if possible. Media objects of smaller *shiftingPriority* values shall be proportionally placed in first fragments. Within the media objects of a fragment, the one who has the maximum *duration* value imposes the duration of that fragment.

7.4 Scene semantic adaptation core

Considering the hierarchy that we developed for elements of a multimedia adaptation system, and as recalled earlier in this chapter, the scene semantic adaptation core typically consists of two parts: scene optimizer and scene adaptor.

Within our MSSA framework, we decided to integrate the presentation structure adaptor part of the scene adaptor entity into the scene optimizer part. This is due to the fact that in our implementation, the scene optimization part (scene rearrangement calculations) and presentation structure adaptation (rearrangements of the spatial, temporal and logical structure of the scene) were implemented close together and approximately in one step. Consequently the elements hierarchy of the scene semantic adaptation core of MSSA framework changes as follows:

- Scene optimization
    a. Presentation optimization, i.e. decision-making and calculations of structure rearrangements
    b. Presentation structure adaptation, i.e. applying structure rearrangements
- Media adaptation process, i.e. media resource adaptation tools

In fact the intelligent part of adaptation core, is the scene optimizer, for, it contains the decision-making algorithm of the adaptation core. The scene adaptor part is just a set of adaptation tools that perform the adaptation(s) decided by optimizer algorithm.

Two types of SMIL scenes are considered in this work: simple and sequential scenes. We define a simple SMIL scene as a SMIL document that within its *body* element, the outermost enclosing element that wraps up the media objects is a *par* element. Figure 7.6 shows an example for such type of SMIL scene.

```
<smil>
   <head>
      <layout>
         <root-layout width="650" height="550" background-color="#ffffff" />
         <region id="video" width="150" height="150" left="50" top="50" />
         <region id="text1" width="300" height="200" left="300" top="50" />
         <region id="image" width="90" height="180" left="400" top="300" />
         <region id="text2" width="300" height="150" left="50" top="300" />
      </layout>
   </head>
   <body>
      <par>
         <video src="video.ram" dur="15s" region="video" />
         <text src="text1.txt" dur="15s" region="text1"/>
         <img src="image.gif" dur="15s" region="image"/>
         <text src="text2.txt" dur="15s" region="text2"/>
      </par>
   </body>
</smil>
```

Figure 7.6: A simple SMIL scene

```
<smil>
    <head>
        <layout>
            <root-layout width="1000" height="950" background-color="#ffffff"/>
            <region id="video" width="150" height="150" left="50" top="50"/>
            <region id="text" width="300" height="200" left="300" top="50"/>
            <region id="image" width="90" height="180" left="400" top="300"/>
            <region id="textt" width="300" height="150" left="50" top="300"/>
        </layout>
    </head>
    <body>
        <seq>
            <par dur="15s">
                <video id="v1" src="v1.ram" region="video"/>
                <text id="t1" src="t1.txt" region="text"/>
                <img id="i1" src="i1.jpg" region="image"/>
                <text id="tt1" src="tt1.txt" region="textt"/>
            </par>
            <par dur="15s">
                <video id="v2" src="v2.ram"  region="video"/>
                <text id="t2" src="t2.txt"  region="text"/>
                <img id="i2" src="i2.jpg"  region="image"/>
                <text id="tt2" src="tt2.txt" region="textt"/>
            </par>
        </seq>
    </body>
</smil>
```

Figure 7.7: A sequential SMIL scene

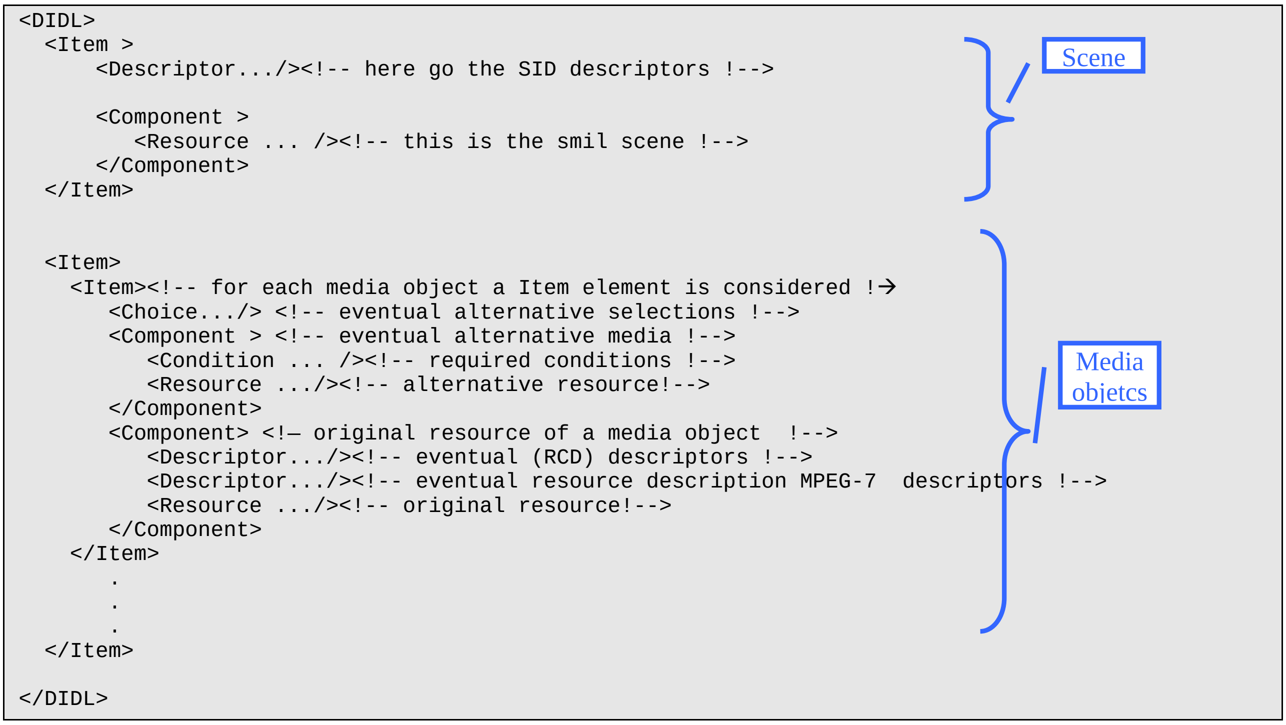
We define a sequential SMIL scene to be a SMIL document in which the outermost enclosing element of the *body* element is a *seq* element. Figure 7.7 shows an example for such type of SMIL scene.

7.4.1 Scene optimization

The information provided in XDI and the corresponding CDI of the scene are used for scene optimization. The XDI contains usage context information. The CDI contains semantic information of the scene, provided by SID descriptors, and also static resource conversion information, provided by RCD (Resource Conversion Description) descriptors as described in chapter 4.

Figure 7.8 shows an example of such CDI. As can be observed, a *Component* element provides the URI of the scene, and its associated *Descriptor* element contains the SID information of the considered scene. The media objects as well as their corresponding static RCD and MPEG-7 (physical content description of media objects) descriptors are placed in an *Item* element. For each media object a *Component* element is provided; the associated *Descriptor* elements provide RCD and MPEG-7 metadata. The whole is wrapped up in an enclosing *Item* (or *Container*) element.

In the following, we briefly describe our optimization algorithm, i.e. the optimization calculations for "presentation optimization" and scene rearrangement or "presentation layout adaptation".



```
<DIDL>
  <Item >
      <Descriptor.../><!-- here go the SID descriptors !-->

      <Component >
         <Resource ... /><!-- this is the smil scene !-->
      </Component>
  </Item>


  <Item>
    <Item><!-- for each media object a Item element is considered !→
       <Choice.../> <!-- eventual alternative selections !-->
       <Component > <!-- eventual alternative media !-->
          <Condition ... /><!-- required conditions !-->
          <Resource .../><!-- alternative resource!-->
       </Component>
       <Component> <!– original resource of a media object  !-->
          <Descriptor.../><!-- eventual (RCD) descriptors !-->
          <Descriptor.../><!-- eventual resource description MPEG-7  descriptors !-->
          <Resource .../><!-- original resource!-->
       </Component>
    </Item>
       .
       .
       .
  </Item>

</DIDL>
```

Figure 7.8: An example of a CDI with SID and RCD descriptors

7.4.1.1 Optimization calculations algorithm

This section describes the algorithm that we developed for our scene optimization. Figure 7.9 shows an overall and rough flowchart of our scene optimization algorithm, while Figure 7.10 demonstrates the flowchart of the adaptation layout box as highlighted in Figure 7.9. In the following, we explain these algorithms in more details.

1- The modalities of all the media objects present in the scene are first checked. For each media object, whose modality is either not supported by the target device or is not wished by the user (UED descriptors), the corresponding SMIL media element (*img*, *video* or *text*) is either eliminated -if necessary- or decided to be converted to another SMIL media element; this later is in fact a decision for performing a transmoding.

The formats of all the media objects of the scene are subsequently checked. For each media object, whose format is not supported by the target device, either it is decided to do a resource transcoding, or the corresponding SMIL element is eliminated.

In such cases of transmoding or transcoding, the attributes of the corresponding *region* element remain intact as we consider that any media transmoding or transcoding is a pure modality or format conversion and dose not change the spatial size of the media.

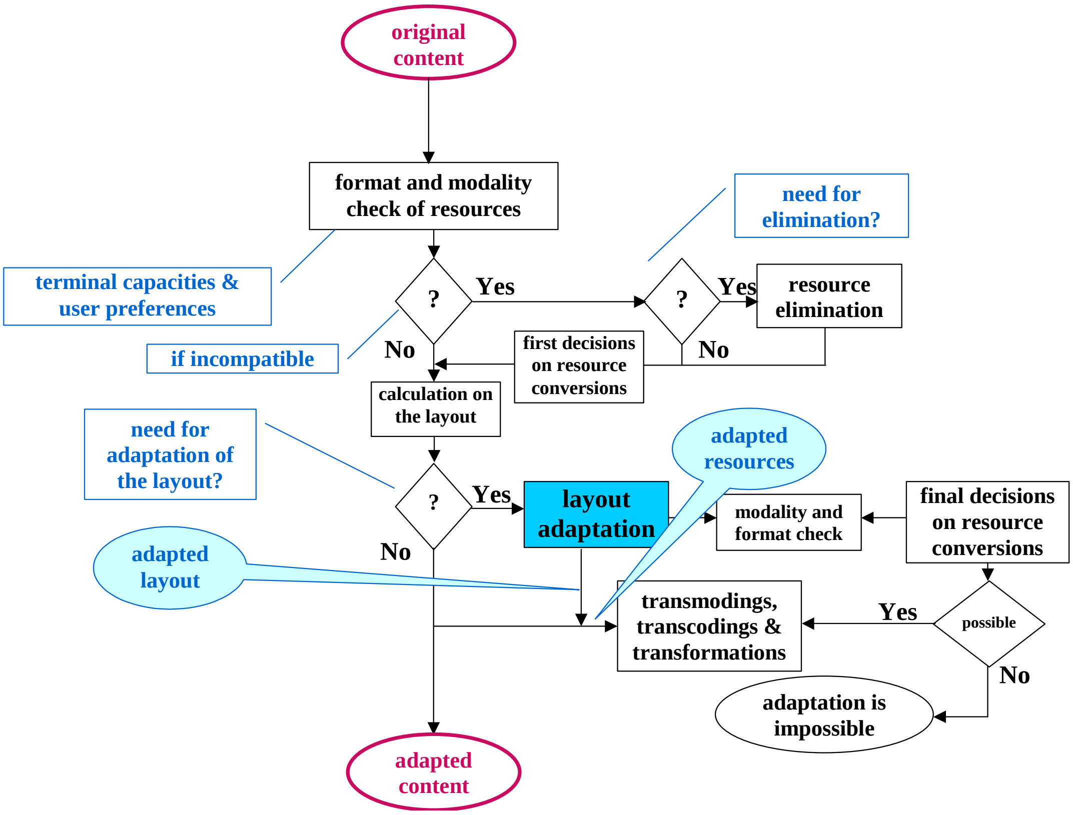

original content
format and modality check of resources
terminal capacities & user preferences
need for elimination?
?
Yes
?
Yes
resource elimination
if incompatible
No
first decisions on resource conversions
No
calculation on the layout
need for adaptation of the layout?
adapted resources
?
Yes
layout adaptation
modality and format check
final decisions on resource conversions
adapted layout
No
transmodings, transcodings & transformations
Yes
possible
No
adaptation is impossible
adapted content

Figure 7.9: Flowchart of our scene optimization algorithm

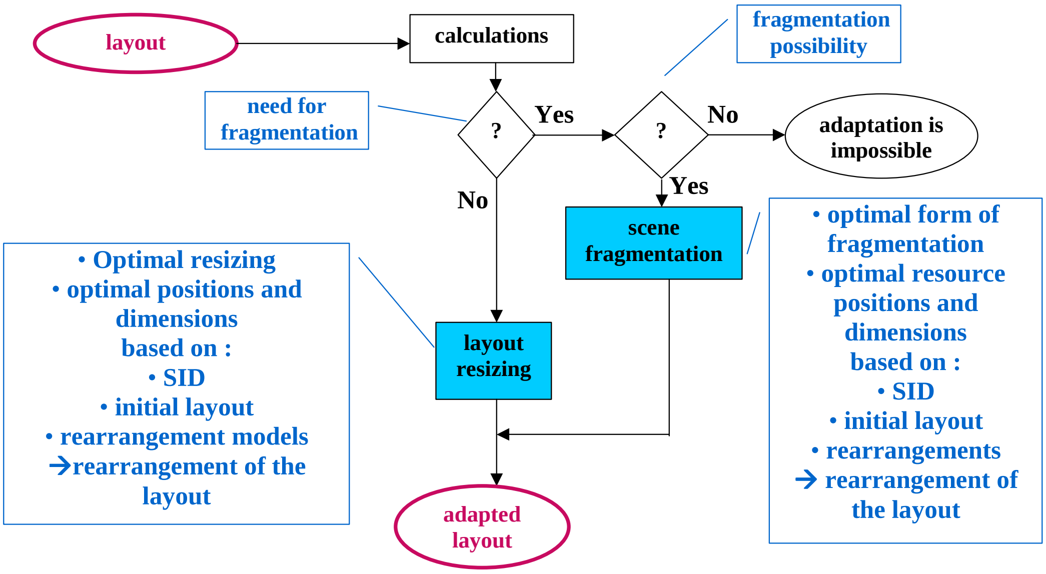


Figure 7.10: Flowchart of our layout adaptation algorithm

The value of the *src* attribute of the new SMIL media element points to an URI that does not exist for the moment, for, the unsupported media will be transmoded or transcoded later by the media resource adaptation module.

In case, based on the device-supported and user-preferred modalities, no transmoding is possible, considering the media objects omitting rules, described in section 7.3.4, either it is decided to remove the corresponding media object from the scene or the scene adaptation is reported to be impossible and the process is terminated.

Also if the media object which is to be removed is a precondition for another media object (*PreConditionMedia* element in SID), based on media object omitting rules, it is verified if the deletion of this latter media object is permitted or not. If it is permitted, the decision is to remove both media objects. And if not, scene adaptation is reported to be impossible and the process terminates. In case, a media object removing is finally decided to take place, the relative *region* element, if not referenced by another media object is removed of the document.

2- Some calculations are done on the spatial layout of the scene in order to find the best form of it. We roughly explain these calculations here :

Consider that the SMIL document has M screens. For a *simple* SMIL document, we define "*par screen*" as the outermost enclosing *par* element within the *body* element, while for a *sequential* SMIL document, a *par screen* is defined as the *par* element, contained in the outermost enclosing *seq* element of the body element. Sometimes, in order to be better understood, we also use the term "*par element*" for *par screen*, but this should not be misleading and it should be noted that this is not "any" *par* element, but only those defined above.

Now, let us consider a (SMIL) generic spatial layout with M *par screen*s, each containing N media objects (or grouping elements, such as a *seq* element that represents a slideshow). To better understand this, let us consider the Figure 7.11 that shows a generic layout for the mth *par screen* of a SMIL scene (where $1 \leq m \leq M$). As can be observed the spatial layout is completely generic, i.e. we do not limit our scenes layout to any specific type of multimedia documents (e.g. news pages as considered in [74]). A, B, C, and D are the $N=4$ media objects (or media grouping elements) of this *par screen*. For each mth *par screen* of the SMIL document:

a. We define that several media objects are *horizontally-parallel*, when they share the width of the display. Groups of media objects that are *horizontally-parallel* are constructed. If we consider than in the mth *par screen*, there exist P such groups, then the pth such group is denoted by Gh_p for $1 \leq p \leq P$.

For instance, as shown by horizontal dashed lines in Figure 7.11, the groups of *horizontally-parallel* media objects are: $\{A, B\}$, $\{A, D\}$ and $\{C, D\}$. Hence we can write:

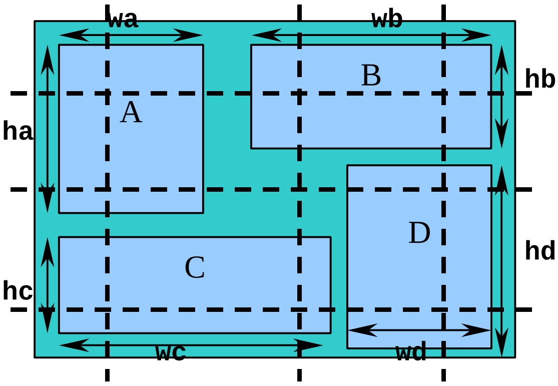


Figure 7.11: Generic *m*th *par screen* of a SMIL scene with media objects A, B, C and D

$\mathtt{Gh}_1 = \{A, B\}$, $\mathtt{Gh}_2 = \{A, D\}$, $\mathtt{Gh}_3 = \{C, D\}$, and $P = 3$

The optimizer then constructs a group of these *horizontally-parallel* media objects groups. In the following, $\mathtt{GGh}_m$ denotes the group of *horizontally-parallel* media objects groups for an mth *par* of a SMIL document. (GG stands for Group of Groups)

$$\mathtt{GGh}_m = \sum_{p=1}^{P} \mathtt{Gh}_p \quad \text{for } 1 \le m \le M$$

For the *par screen* of Figure 7.11, this will be as follows:

$$\mathtt{GGh}_m = \{\{A, B\}, \{A, D\}, \{C, D\}\}$$

b. Then for each mth *par*, the groups of *vertically-parallel* media objects are constructed. If we consider that in the mth *par screen*, there exist Q such groups, then the qth such group is denoted by $\mathtt{Gv}_q$ for $1 \le q \le Q$.
Observing the vertical dashed lines of the example of Figure 7.11, it can be seen that these *vertically-parallel* media objects groups are:

$\mathrm{Gv}_1 = \{A, C\}$, $\mathrm{Gv}_2 = \{B, C\}$, $\mathrm{Gv}_3 = \{B, D\}$, and $Q = 3$

In the following, $\mathtt{GGv}_m$ denotes the group of *vertically-parallel* media objects groups for an mth *par* of a SMIL document:

$$\mathtt{GGv}_m = \sum_{q=1}^{Q} \mathtt{Gh}_q \qquad \text{for } 1 \le m \le M$$

$$\mathtt{GGv}_m = \{\{A, C\}, \{B, C\}, \{B, D\}\}$$

c. For each $\texttt{Gh}_p$ of each $\texttt{GGh}_m$, the sum of region widths of its media objects (obtained from the *region* element referenced by the corresponding media object) is then calculated. Subsequently within the $\texttt{GGh}_m$, the $\texttt{Gh}_p$ group, which has the maximum sum of region widths, is found; this happens for $p = p_{max}$). For each mth *par screen* we denote this maximum sum of region widths by $\texttt{maxGroupWidthSUM}_m$. The same will be done for $\texttt{GGv}_{m,}$ and the maximum sum of region heights, $\texttt{maxGroupHeightSUM}_m$ is calculated (this happens for $\texttt{Gv}_{q_{max}}$).
For instance, if we consider the following assumptions for the region dimensions of visual media objects of the *par screen* of example Figure 7.11:

$\texttt{wa<wd<wb<wc}$, and $\texttt{hc<hb<ha<hd}$, then

$\texttt{maxGroupWidthSUM}_m = \texttt{wd+wc}$, that happens for $\texttt{Gh}_3$, meaning that $p_{max} = 3$, and

$\texttt{maxGroupHeightSUM}_m = \texttt{hd+hb}$, that happens for $\texttt{Gv}_3$, meaning that $q_{max} = 3$.

d. Using the values of $\texttt{maxGroupWidthSUM}_m$ and $\texttt{maxGroupHeightSUM}_m$, and also the values of *maxRRF* (maximum Resolution Reduction Factor, extracted from the corresponding media object SID descriptors in the CDI) for the corresponding media objects, the minimum possible region widths sum for the objects of $\texttt{Gh}_{p_m}$ and $\texttt{Gv}_{q_{max}}$, i.e. the $\texttt{Gh}$ and $\texttt{Gv}$ groups having the maximum media objects width and height sum, is then calculated. For each mth *par screen*, the calculated values will be denoted $\texttt{maxMinGroupWidthSUM}_m$ and $\texttt{maxMinGroupHeightSUM}_m$. For instance, if the *maxRRF* of all media objects of Figure 7.11 is 2, then:

$$\texttt{maxMinGroupWidthSUM}_m = \frac{\mathrm{wd}+\mathrm{wc}}{2}, \text{ and}$$

$$\texttt{maxMinGroupHeightSUM}_m = \frac{\mathrm{hd}+\mathrm{hb}}{2}$$

3- The above calculations are repeated for each *par screen* of the SMIL document. Subsequently the largest values of $\texttt{maxGroupWidthSUM}_m$, $\texttt{maxGroupHeightSUM}_m$, $\texttt{maxMinGroupWidthSUM}_m$ and $\texttt{maxMinGroupHeightSUM}_m$, among all *par screen*s are calculated. These values are respectively denoted by $\texttt{maxGroupWidthSUM}_{m_{max}}$, $\texttt{maxGroupHeightSUM}_{m_{max}}$, $\texttt{maxMinGroupWidthSUM}_{m_{max}}$ and $\texttt{maxMinGroupHeightSUM}_{m_{max}}$.

4- The dimension of the layout of the SMIL document is then checked out. If one of the layout width or height size is respectively bigger than target device display width and height size, then it is decided that a layout adaptation is required, meaning that the scene shall be resized (downscaled) and/or fragmented. Otherwise the layout size remains intact. Then a second modality and format check is performed on the resources that are decided to be present in the final scene. Based on this check, if any media conversion is decided to be done, the related conversions – if possible – will be performed by the media adaptation modules, and the scene will be out put. However if the decided conversions are impossible – for example if the needed converter is not available, the adaptation is reported to be impossible.

Layout adaptation algorithm:

In case where a layout adaptation is required, following equation is considered:

$$\frac{\mathtt{maxGroupWidthSUM}_{m\ max}}{\mathtt{layoutWidth}} = \frac{\mathtt{maxMinGrou\,pWidthSUM}_{m\ max}}{\mathtt{minDisplay\,Width}}$$

The above equation means that the ratio of $\mathtt{maxGroupWidthSUM}_{m_{max}}$ to $\mathtt{layoutDisplay}$ is considered to be equal to the ratio of $\mathtt{maxMinGroupWidthSUM}_{m_{max}}$ to the minimum required display width. The same is considered for height dimension:

$$\frac{\mathtt{maxGroupHeightSUM}_{m\ max}}{\mathtt{layoutHeight}} = \frac{\mathtt{maxMinGrou\,pHeightSUM}_{m\ max}}{\mathtt{minDisplay\,Height}}$$

From the above equations, the values of $\mathtt{minDisplayWidth}$ and $\mathtt{minDisplayHeight}$ are calculated. We then define that a scene fragmentation is necessary if

$\mathtt{minDisplayWidth} > \mathtt{displayWidth}$, or

$\mathtt{minDisplayHeight} > \mathtt{displayHeight}$.

In this case, the scene fragmentation sub-module is called to handle the case. Otherwise, i.e., if no scene fragmentation is necessary, the case will be handed to the scene downscaling sub-module.

In any of these cases, the *height* and *width* attributes of the *root-layout* element of the adapted SMIL scene are respectively set to the device display height and width. If scene fragmentation or scene resizing sub-modules decide on any media resizing, the aspect ratio is preserved for all visual media objects, except for the text media objects.

A simple example for scene optimization calculations

To better understand all above defined calculations, let us consider a very simple layout adaptation example. The layout of the SMIL scene, which is to be adapted, is shown in Figure 7.12. The SMIL scene is of *simple* type, i.e. it has only one *par* element in *body* element (see Figure 8.1). It contains four visual media objects, for each of which, the value of *maxRRF* is 3.

In this example, in order to avoid complexity, the adaptation is supposed to be done for satisfying only one criterion: display size of target device. No other usage environment constraint is considered. Hence, the corresponding XDI is very simple and not provided here. The corresponding semantic metadata CDI is given in Figure 7.13. Again for simplicity reasons, the CDI provides only SID metadata and no RCD (Resource Conversion Description) and media resource description is provided. We will then have:

$$\mathtt{maxGroupWidthSUM}_{m_{max}} = 750,$$

$$\mathtt{maxGroupHeightSUM}_{m_{max}} = 550,$$

$$\mathtt{maxMinGroupWidthSUM}_{m_{max}} = \frac{750}{3} = 250$$

, and

$$\mathtt{maxMinGroupHeightSUM}_{m_{max}} = \frac{550}{3} \cong 183.3$$

, consequently:

$$\mathtt{minDisplayWidth} = \frac{900 \times 250}{750} = 300$$

$$\mathtt{minDisplayHeight} = \frac{700 \times 183}{550} \cong 130$$

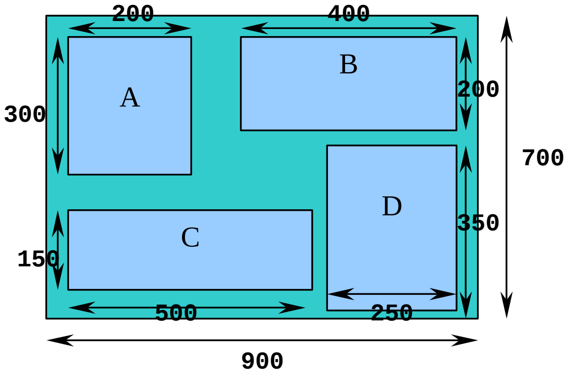


Figure 7.12: An example layout with four visual media objects

```
<DIDL>
  <Item id="smilscene1">
    <Descriptor>
      <Statement mimeType="text/xml">
        <sid:SID>
          <sid:Object id="A" role="key-convertible" ref="A" maxRRF="3">
            <sid:SemanticDependencies>
              <sid:AbsoluteSemanticDependencies>
                <sid:PreConditionMedia>
                  <sid:MediaObject ref="C"/>
                </sid:PreConditionMedia>
              </sid:AbsoluteSemanticDependencies>
              <sid:SynchronizationSemanticDependencies synchronizedTo="C"/>
              <sid:SpatialSemanticDependencies>
                <sid:KeepCloseTo><sid:MediaObject ref="C"/></sid:KeepCloseTo>
               </sid:SpatialSemanticDependencies>
            </sid:SemanticDependencies>
            <sid:SemanticFragmentationPreferences>
              <sid:TimeFragmentationPreferences shiftingPriority="1"/>
              <sid:SpatialFragmentationPreferences keepWith="C"/>
            </sid:SemanticFragmentationPreferences>
          </sid:Object>
          <sid:Object id="B" ref="B" role="key-convertible" maxRRF="3">
            <sid:SemanticDependencies>
             <sid:SynchronizationSemanticDependencies synchronizedTo="C"/>
             <sid:SpatialSemanticDependencies>
               <sid:KeepCloseTo><sid:MediaObject ref="D"/></sid:KeepCloseTo>
             </sid:SpatialSemanticDependencies>
            </sid:SemanticDependencies>
            <sid:SemanticFragmentationPreferences>
               <sid:TimeFragmentationPreferences shiftingPriority="2"/>
               <sid:SpatialFragmentationPreferences keepWith="D"/>
            </sid:SemanticFragmentationPreferences>
          </sid:Object>
          <sid:Object id="C" ref="C" role="key-convertible" maxRRF="3">
             <sid:SemanticDependencies>
              <sid:SpatialSemanticDependencies>
                <sid:KeepCloseTo><sid:MediaObject ref="A"/></sid:KeepCloseTo>
              </sid:SpatialSemanticDependencies>
             </sid:SemanticDependencies>
          </sid:Object>
          <sid:Object id="D" ref="D" role="key-convertible" maxRRF="3">
            <sid:SemanticDependencies>
              <sid:SpatialSemanticDependencies>
                <sid:KeepCloseTo><sid:MediaObject ref="B"/></sid:KeepCloseTo>
              </sid:SpatialSemanticDependencies>
            </sid:SemanticDependencies>
          </sid:Object>
        </sid:SID>
      </Statement>
    </Descriptor>
    <Component id="ABCDsmilScene">
        <Resource mimeType="scene/smil" ref="ABCDsmilScen.smil"/>
    </Component>
  </Item>
</DIDL>
```

Figure 7.13: Example semantic metadata CDI for layout of Figure 7.12

We now consider two adaptation cases: one with a target device display size of 400*300 and one with a target device display size of 144*176 (usual PDA display size).

A. Device display size is 400*300

We have:

`MinDisplayWidth = 300 < displayWidth = 400`, and

```
MinDisplayHeight = 130 < displayHeight = 300
```

No scene fragmentation is hence necessary, and a layout downscaling is sufficient for adaptation.

B. Device display size is 144*176

We have:

```
MinDisplayWidth = 300 > displayWidth = 176
```

A scene layout fragmentation is therefore unavoidable for adaptation.

7.4.1.2 Scene rearrangement algorithm

Scene rearrangement is in fact the presentation layout adaptation and consists of two types of layout adaptation: layout downscaling (i.e. spatial scene resizing) and scene fragmentation (i.e. scene page splitting).

Please note that, since “optimal” does not have an absolute signification, based on how one defines optimization, scene downscaling optimization, or scene fragmentation optimization, the optimization algorithm may be completely different according to which constraints are deemed most important. Our choice was to privilege maximum use of the available display size based on our defined interpretation rules of semantic information of the scene, as described in 7.3.4.

Through the process of scene downscaling and fragmentation, a media downscaling is decided, if the value of the *role* attribute within corresponding SID descriptor is not *key*. Otherwise the media object will remain in the scene with its original region dimensions and the scene downscaling or fragmentation calculations will be redone, this time considering the unchangeable region dimensions of this media.

In the following, we describe scene downscaling and fragmentation calculations. In order to avoid the complexity, we only explain the calculations process in it simplest case, i.e. for the case where concerned media objects do not have a *key role*. Calculations for other cases are not described, however dealt with in our algorithm.

7.4.1.2.1 Spatial scene layout downscaling

In case, where a layout downscaling is decided to be necessary by the optimization calculations, then the scene-downscaling sub-module downscales the layout of the scene. In this section we give a rough description of the optimizing algorithm that we have developed for scene layout downscaling.

We consider the following equation:

$$\frac{\texttt{maxGroupWidthSUM}_{m\ max}}{\texttt{layoutWidth}} = \frac{\texttt{adaptedMax GroupWidth SUM}_{m\ max}}{\texttt{displayWid th}}$$

This equation means that we consider the ratio of the $\texttt{maxGroupWidthSUM}_{m_{max}}$ (i.e. the sum of the widths of the media objects of the *horizontally-parallel* group that has the maximum width sum through all the original SMIL scene) to the `layoutWidth` (i.e. the width size of the layout of the original SMIL scene) to be equal to the ratio of the $\texttt{maxGroupWidthSUM}_{m_{max}}$ of the adapted SMIL scene, denoted by $\texttt{adaptedMaxGroupWidthSUM}_{m_{max}}$ to the `displayWidth` (i.e. the width size of the target device display).

This, of course implies that the size of the layout of the adapted SMIL scene is the same as the device display size.

The value of $\texttt{adaptedMaxGroupWidthSUM}_{m_{max}}$ represents the maximum limit of width sum of horizontally parallel media object groups in the adapted scene, and is calculated from the above equation. The same is done to calculate the value of $\texttt{adaptedMaxGroupHeightSUM}_{m_{max}}$.

Then in order to calculate the resizing factor of each media object, two values, called optimal width and height downscaling factors, are calculated as follows:

$$\texttt{wf}=\frac{\texttt{adaptedMaxGroupWidthSUM}_{m\ max}}{\texttt{maxMinGroupWidthSUM}_{m\ max}}$$ (optimal width downscaling factor), and

$$\texttt{hf}=\frac{\texttt{adaptedMaxGroupHeightSUM}_{m\ max}}{\texttt{maxMinGroupHeightSUM}_{m\ max}}$$ (optimal height downscaling factor).

The optimizer then calculates which of the above computed values; `wf` or `hf`, is the overall optimal downscaling factor. To avoid too many details, we do not explain how this is calculated, nevertheless we consider one only case; for example if

$\texttt{wf} \times \texttt{maxMinGroupHeightSUM}_{m_{max}} \leq \texttt{displayHeight}$, and

$\texttt{wf} \times \texttt{maxMinGroupWidthSUM}_{m_{max}} \leq \texttt{displayWidth}$,

then the optimal downscaling factor is the `wf` if `hf` < `wf`. This optimal downscaling factor will be denoted by `optimalMediaDownscalingFactor`.

After computing `optimalMediaDownscalingFactor`, for each media object of the scene, it is verified if a *simple resizing* is possible. A *simple resizing* is possible when, for the concerned media object:

$\frac{\texttt{displayWidth}}{\texttt{layoutWidth}} \geq \frac{1}{maxRRF}$ and

$\frac{\texttt{displayHeight}}{\texttt{layoutHeight}} \geq \frac{1}{maxRRF}$.

Then the optimizer calculates another value denoted by `simpleResizingFactor`, which is either: (based on which one gives an optimal adapted scene)

$\frac{\texttt{displayWidth}}{\texttt{layoutWidth}}$, or

$\frac{\texttt{displayHeight}}{\texttt{layoutHeight}}$.

Then if for a media object a *simple resizing* is possible, and if

$$\texttt{simpleResizingFactor} \geq \frac{\texttt{optimalMediaDownscalingFactor}}{maxRRF},$$

then the corresponding visual media object shall be downscaled by this `simpleResizingFactor`. Therefore, the values of *width* and *height* attributes of *region* element will be changed – adapted – by the optimizer, and the media resource downscalings will be done by resource transmoders (media adaptation module).

Otherwise, the considered visual media object shall be downscaled by its corresponding $\frac{\texttt{optimalMediaDownscalingFactor}}{maxRRF}$ factor. Hence, again the values of *width* and *height* attributes of *region* elements will be changed and the corresponding media resizings will be done by resource transformers.

In an alternative solution, we took into account the value of *importance* attribute (of SID descriptors) in the calculation of downscaling factor of each media object.

This results in a smaller downscaling for media objects of higher importance.

Calculation of the values of *width* and *height* attributes of *region* elements of corresponding media objects is rather straightforward and simpler than calculating the correct values of the corresponding *top* and *left* attributes. The latter depends on more parameters. Based on the old and new *width* and *high* attributes of *region* elements and *root-layout* element and based on the old *top* and *left* values, and certainly based on the general spatial semantic dependencies and preferences (expressed in SID descriptors), and also taking into account the media objects that were removed from the layout, the optimizer calculates the appropriate new values of *top* and *left* attributes of each visual media object.

A simple example for scene layout downscaling

Let us now calculate these values for case A of our considered example. We will have:

$$\mathtt{adaptedMaxGroupWidthSUM}_{m_{max}} = \frac{750\times 400}{900} \cong 333,$$

$$\mathtt{adaptedMaxGroupHeightSUM}_{m_{max}} = \frac{550\times 300}{700} \cong 236,$$

$\mathtt{wf} = \frac{330}{250} = 1.332$, and $\mathtt{hf} = \frac{236}{183} = 1.29$

Consequently the optimal downscaling factor will be:

$$\mathtt{OptimalMediaDownscalingFactor} = 1.332$$

For all the media objects, a *simple resizing* is possible, and

$$\mathtt{SimpleResizingFactor} \cong \frac{\mathtt{optimalMediaDownscalingFactor}}{\mathtt{maxRRF}} = 0.44$$

Therefore, the values of *height* and *width* attributes of corresponding *region* element, in the SMIL document, will be divided by a factor of 0.44. The media resource adaptors will then downscale the concerned visual media object by this factor. The adapted layout is showed in Figure 7.14.

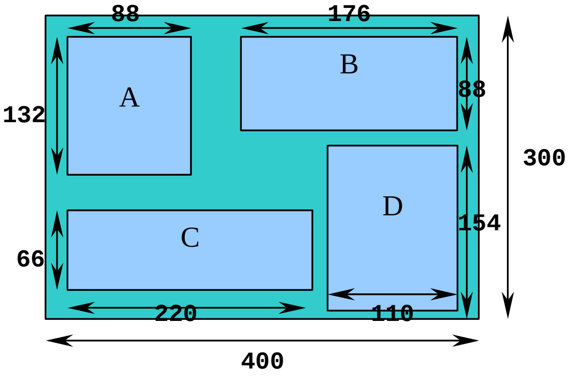


Figure 7.14: Adapted layout of Figure 7.12 for a 300*400 display size

For this particular example, as the SMIL document was of *simple* type and the *maxRRF* values of all visual media objects were the same, and since we did not take into account the device capabilities and user preferences (therefore no transmoding and transcoding were decided), the optimization and downscaling calculations were unusually very light.

7.4.1.2.2 Scene fragmentation

Whenever the optimization calculations result in deciding a scene fragmentation, the scene is fragmented.

For each *par* element (*par screen*) of the original SMIL scene, the followings are done:

a. Based on the semantic information of the scene, I groups of semantically-dependent media objects (for each *par screen*) are constructed. The ith such group is denoted by $\mathtt{SG}_i$.

b. Then for each $\mathtt{SG}_i$, which is going to be a fragment (sub-scene) of the adapted scene, the sub-scene will be adapted as explained in the following. The following will be performed for all groups. If possible, two temporally consequent groups are integrated in one fragment.

- For each ith semantically-related media object group ($1<i<I$) , the maximum height and width, among the heights and widths of all objects of the considered group are calculated. These are respectively denoted by $\mathtt{maxGroupHeight}_i$ and $\mathtt{maxGroupWidth}_i$. Then, based on the *maxRRF* values of them media objects, for each group, the minimum possible $\mathtt{maxGroupHeight}_i$ and $\mathtt{maxGroupWidth}_i$, are calculated. These are respectively denoted by $\mathtt{maxMinGroupHeight}_i$ and $\mathtt{maxMinGroupWidth}_i$.

  Then the sum of the heights and widths of each semantically-related object group are calculated, these are respectively denoted by $\mathtt{hightSum}_i$ and $\mathtt{widthSum}_i$. Again, taking into account he *maxRRF* values of media objects, the minimum $\mathtt{hightSum}_i$ and $\mathtt{widthSum}_i$ are calculated. These will be respectively denoted by $\mathtt{minHightSum}_i$ and $\mathtt{minWidthSum}_i$.

- We then define that for a semantically-related objects group, a *simple fragmentation* is possible if:

  $\mathtt{heightSum}_i < \mathtt{displayHeight}$, and $\mathtt{widthSum}_i < \mathtt{displayWidth}$,

  or

  $\mathtt{heightSum}_i < \mathtt{displayHeight}$, and $\mathtt{maxGroupWidth}_i < \mathtt{displayWidth}$,

  or

  $\mathtt{widthSum}_i < \mathtt{displayWidth}$, and $\mathtt{maxGroupHeight}_i < \mathtt{displayHeight}$.

This means that the objects of this group can be placed in a scene fragment without being downscaled. Then based on the above inequalities and comparison of the following values:

$\texttt{dh}_i = \texttt{displayHeight} - \texttt{heightSum}_i$, and

$\texttt{dw}_i = \texttt{displayWidth} - \texttt{widthSum}_i$,

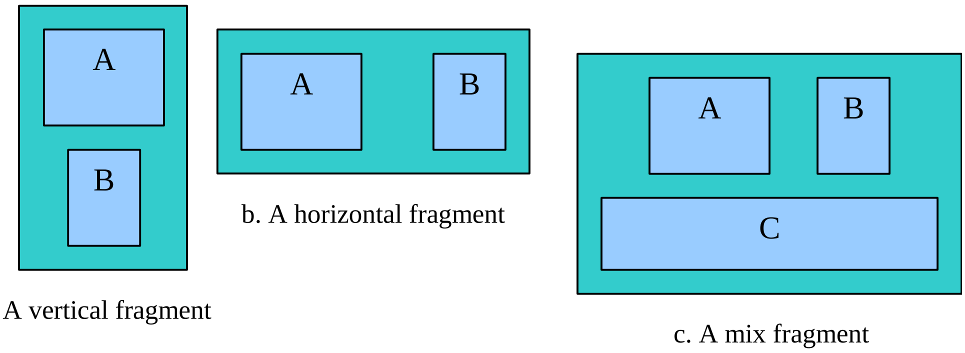
it will be decided whether a vertical, horizontal or mixed fragmentation is optimal.

A vertical fragmentation results in a vertical layout, meaning that the media objects of the group will be placed in a *vertically-parallel* way (see Figure 7.15.a). A horizontal fragmentation results in a horizontal layout, meaning that the media objects will be placed in a *horizontally-parallel* way (see Figure 7.15.b). If a group contains more than three visual media objects, we consider that a mixed fragmentation could be envisaged (see Figure 7.15.c). The position calculations are more complex for mixed fragmentation.

In order to avoid going through too many details, we do not explain our algorithm of fragmentation type decision-making. Once the optimal type of fragmentation (vertical, horizontal or mixed) is decided, it will be applied to the scene as described in the following.

- If for the considered semantically-related objects group, a *simple fragmentation* is not possible, i.e. if media downscaling is required, the media downscaling factor for each object of the group is then calculated.

  Based on the *maxRRF* value of each visual media object, the values of *height* and *width* attributes of the corresponding *region* element, the values of `maxGroupHeight` and `maxGroupWidth` of the corresponding semantically related object group, and also based on the display height and width, the downscaling factor of each media object of each semantically related object groups are calculated. The *importance* value can be alternatively taken in to account.



a. A vertical fragment

b. A horizontal fragment

c. A mix fragment

Figure 7.15: Scene fragments resulting from scene (a) vertical, (b) horizontal and (c) mixed fragmentations

The calculations are similar to the scene downscaling calculations, and are not therefore described here. We just remind that if

`minWidthSum`$_i$ `> displayWidth`, and `minHeightSum`$_i$ `> displayHeight`,

no fragmentation is possible without any media object omitting. We, therefore, have to see if, based on media object omitting rules (section 7.3.4.3), there are any media objects which could be dropped off from this group or not. If possible, after removing those media objects, we redo the calculations for this new group.

- Once the new values are calculated for *height* and *width* attributes of the corresponding *region* elements, they replace the old ones.

  The new *top* and *left* values will be calculated based on decided type of fragmentation (vertical, horizontal or mixed), the new already-calculated dimensions of the layout and each media object, and also spatial fragmentation preferences, expressed through SID descriptors. Synchronization dependencies are also respected through fragment constructions.

c. After constructing the scene fragments, the structure of the scene will also be changed as follows. The scene fragments, which are in fact the semantically related object groups, will each be represented by a *par* element. If it is desired to result in an un-interactive sequential scene, scene fragments will then be simply sequenced by means of a *seq* enclosing element. In case an interactive adapted scene is desired, the scene fragments will be sequenced by means of "next" buttons placed in each scene fragment. "Back" buttons can be also used for play back. For each scene fragment, in the course of calculation and optimization of media objects spatial dimensions and positions, the spatial dimension of "next" buttons – except for the last fragment – are evidently taken into account.

All temporal and spatial dependencies and preferences, expressed through the SID metadata, are also respected within each fragment and through the whole scene.

If spatially possible, two consequent semantically related object groups are placed in a same *par screen*.

d. After finishing the adaptation of the SMIL XML document, the new (adapted) calculated spatial dimensions of each media object is provided to the media adaptation tools that will then downscale the media. Also if there is any transmoding or transcoding to be done, the related adaptation tools are called and the necessary parameters are provided to them.

For downscaling text resources, as we do not use any on-the-fly text summarization tool, we assume that the scene author provides one or several summarized alternative texts in the CDI. We assume that the required (for rendering) regions dimensions of these summarized versions are pre-calculated and implicitly hinted by giving a corresponding *RRF* (Resolution Reduction Factor) value in the CDI. In other words, for each of these pre-provided alternatives a RRF – comparing to original text – is pre-calculated and provided. Therefore, if through the scene optimization calculations, the region of a text resource is downscaled, based on the downscaling factor, one of these alternatives will replace the original text. Please refer to Figure 7.8 see how in a CDI these alternatives can be expressed. An example was given in Figure 7.5.

An example for scene layout fragmentation

Let us consider the case B of example of section **Erreur ! Source du renvoi introuvable.**.

We will have:

$I = 2$, $\mathtt{SG}_{1} = \{\mathtt{A},\ \mathtt{C}\}$, $\mathtt{SG}_{1} = \{\mathtt{B},\ \mathtt{D}\}$,

$\mathtt{maxGroupHeight}_{1} = 300$, $\mathtt{maxMinGroupHeight}_{1} = 100$, $\mathtt{maxGroupWidth}_{1} = 500$, $\mathtt{maxMinGroupWidth}_{1} \cong 167$, $\mathtt{hightSum}_{1} = 450$, $\mathtt{minHeightSum} = 150$, $\mathtt{widthSum}_{1} = 700$, $\mathtt{minWidthSum}_{1} = 233$, and

$\mathtt{maxGroupHeight2}_{1} = 350$, $\mathtt{maxMinGroupHeight}_{1} \cong 117$, $\mathtt{maxGroupWidth}_{2} = 400$, $\mathtt{maxMinGroupWidth}_{1} \cong 133$, $\mathtt{hightSum}_{2} = 550$, $\mathtt{minHeightSum} \cong 183$, $\mathtt{widthSum}_{2} = 650$, $\mathtt{minWidthSum}_{1} \cong 217$.

Since for both groups, $\mathtt{hightSum}_{i} > \mathtt{displayHeight}$, $\mathtt{widthSum}_{i} > \mathtt{displayWidth}$, $\mathtt{maxGroupHeight}_{i} > \mathtt{displayHeight}$, and $\mathtt{maxGroupWidth}_{i} > \mathtt{displayWidth}$, no *simple fragmentation* is possible.

For the first group, $\mathtt{minHeightSum}_{i} < \mathtt{displayHeight}$, therefore a vertical fragmentation is possible. While for the second group, $\mathtt{minWidthSum}_{i} > \mathtt{displayWidth}$, and $\mathtt{minHeightSum}_{i} > \mathtt{displayHeight}$. No fragmentation is hence possible for this group without any media object removing. And based on media object removing rules, no media object can be removed from the scene (absence of *importance* attribute means *importance* = high). No adaptation is hence possible at all.

If the media objects *maxRRF* values were 4 (instead of 3), the adaptation would have been possible, in that case, we would have:

$\mathtt{maxGroupHeight}_{1} = 300$, $\mathtt{maxMinGroupHeight}_{1} = 75$, $\mathtt{maxGroupWidth}_{1} = 500$, $\mathtt{maxMinGroupWidth}_{1} = 125$, $\mathtt{hightSum}_{1} = 450$, $\mathtt{minHeightSum} \cong 113$, $\mathtt{widthSum}_{1} = 700$, $\mathtt{minWidthSum}_{1} = 175$, and

$\texttt{maxGroupHeight2}_1 = 350$, $\texttt{maxMinGroupHeight}_1 \cong 88$, $\texttt{maxGroupWidth}_2 = 400$, $\texttt{maxMinGroupWidth}_1 = 100$, $\texttt{hightSum}_2 = 550$, $\texttt{minHeightSum} \cong 138$, $\texttt{widthSum}_2 = 650$, $\texttt{minWidthSum}_1 \cong 163$.

And for both groups, we have $\texttt{minHeightSum} < \texttt{dispalyHieght}$, therefore a vertical scene fragmentation is possible for both fragments. For this vertical fragmentation we consider the following equation:

$$\texttt{adaptedMaxWidth} = 0.95 \times \texttt{displayWidth}$$

This equation means that the optimal adapted width of the media object of this group who has the maximum width among this group is considered to be equal to 95% of the display width. Therefore:

$\texttt{adaptedWidth}_C \cong 137$, and consequently we'll have:

$\texttt{adaptedHeigh}_C \cong 41$, $\texttt{adaptedHeight}_A \cong 82$, and $\texttt{adaptedWidth}_A \cong 54$

Doing the same calculations for the second group, we will end up by an adapted fragmented scene as shown in Figure 7.16.

The *top* and *left* values could be now calculated for each media object, however, here, we do not explain this calculation.

7.4.2 Scene media objects adaptation

The media adaptation tools that we use in our MSSA framework are the same as used in MRC (Media Resource Conversion) framework.

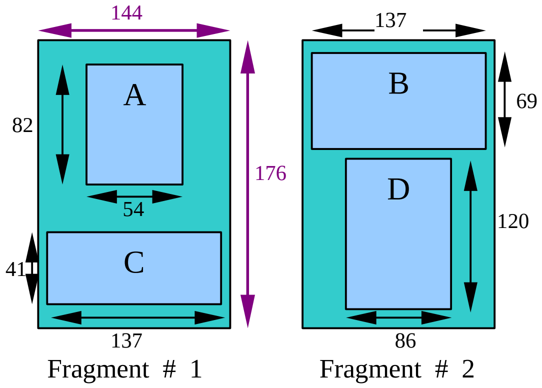


Figure 7.16: Adapted fragmented scene for example of section 7.4.1.2.2

7.5 Conclusion

In this chapter we described our methodology for realization of a framework for semantic customization of multimedia presentations to usage context constraints. We reasoned our choice of SMIL 2.0 language for description of our multimedia scenes. We then explained our solutions for each element of such a framework as defined in previous chapter.

We described how we choose to express the semantic information of a multimedia scene on the basis of MPEG-21. Our proposed descriptors for semantic information description, as well as their interpretation within the system were also detailed.

We subsequently provided a brief description of the algorithm of our scene semantic adaptation core. The architecture of an experimental implementation of this framework is detailed in next chapter.

Chapter 8

MSSA IMPLEMENTATION

Summary

In this chapter we describe the overall architecture of our Multimedia Scene Semantic Adaptation engine. The functionalities of optimization and adaptation modules are also described.

Table of Content

Table of Illustrations

8.1 Introduction

One of the important issues in implementing a multimedia presentation adaptation system is the type of multimedia documents on which the adaptation process will be performed. Depending on the type of used multimedia documents, the "front-end" module of the system varies. For example for XML-based multimedia documents, an XML parser and validator is needed right at the beginning of the process. Then, based on what language, the multimedia document is written in, a language-specific parser and validator is necessary if documents have not been subject to validation before.

One other important issue that should be taken into account is the capabilities and characteristics of multimedia players that exist for the chosen type of multimedia documents, and which run under the different usage contexts in the considered scenarios.

The capabilities of the considered target devices in terms of adaptation are also of great importance. Based on these capabilities, some of the adaptation tasks may be left to the end user device (or player).

In this chapter we present our proof-of-concept implementation for our MSSA framework. This experimental adaptation engine of SMIL 2.0 multimedia documents is implemented based on the methodology described in previous chapter. We first describe the overall architecture of the system and then discuss each module in more detail. A number of examples are discussed through the chapter in order to facilitate the comprehension of the mechanism of the system and show the achievements of the system.

It shall be noted that through this experimental implementation, we do not aim to deal with the questions of a complete end-to-end multimedia distribution system. We consider that transport protocol-related or client-server architectural questions are out of scope of this work. In particular, we solely search to prove how semantic metadata are important and vital for correct performance of an adaptation system.

8.2 Architecture of MSSA system

Before going through the details of the architecture of our MSSA engine, let us recall the elements hierarchy that we considered for our scene semantic adaptation core in previous chapter:

- Scene optimization
  - Presentation optimization; this is in fact the part of scene optimization that is in charge of decision-making and computation of the optimal layout. It calculates the optimal form of the presentation and the media objects.
  - Presentation structure adaptation (layout rearrangement); this is in fact the adaptation of the SMIL XML document. It is in charge of adapting the SMIL XML document, i.e. the presentation.
- Media resource adaptation: this part is in charge of adapting the resources based on the decisions made by the scene optimization part.

Figure 8.1 depicts the overall architecture of our multimedia scene semantic adaptation engine. The part enclosed by (red) dashed line is the semantic scene adaptation core. Its direct inputs are the CDI and XDI. The CDI is a DIDL XML document containing the resources conversion-related and scene semantic information. The SMIL scene and its media resources are also referenced through the CDI and will be fetched by corresponding modules along the adaptation process (input dashed line arrows). Alternatively the SMIL scene may be directly input to the engine. An example of such CDI was given in previous chapter. The XDI is also a DIDL XML document containing UED (Usage Environment Description) metadata.

As shown by numbered steps in Figure 8.1, the overall walkthrough of an adaptation session is as follows:

1- The CDI and XDI are input to the adaptation engine.

2- The SMIL scene is fetched from its URI.

3- According to the optimization algorithm described in previous chapter and based on the metadata provided in the CDI and XDI the optimizer calculates the optimal form of the layout of the presentation as well as the optimal form of the media objects. The scene structure is then converted to the decided optimal form.

4- The media objects are fetched from their URI and are then converted to their optimal decided forms.

5- The media objects and the scene are output and saved.

We now proceed to explain the detailed architecture and function of each module.

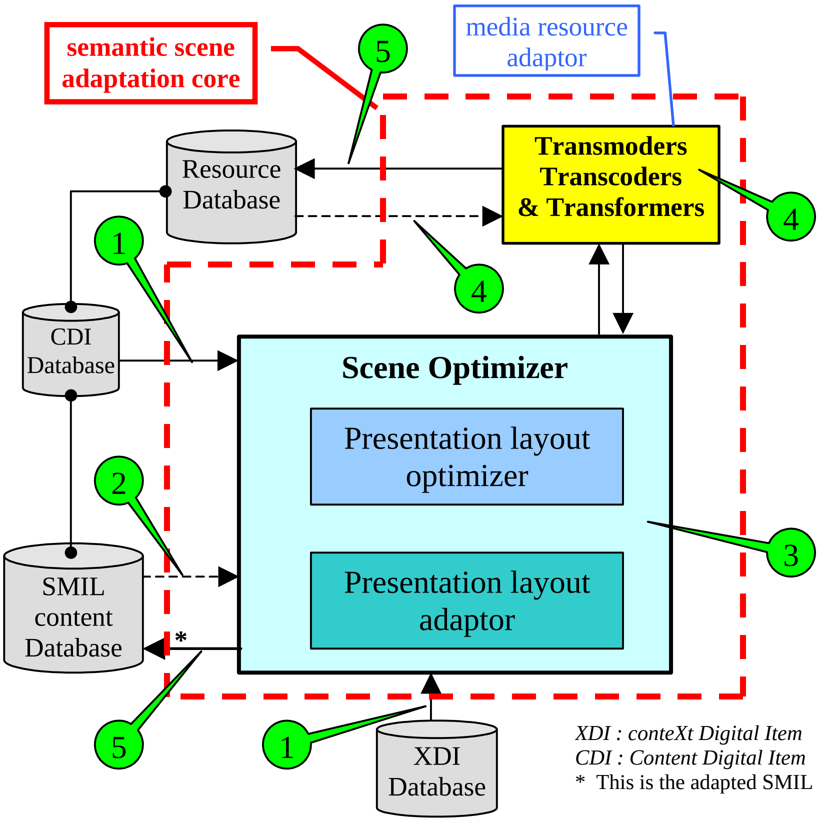

semantic scene adaptation core
media resource adaptor
Resource Database
Transmoders Transcoders & Transformers
CDI Database
Scene Optimizer
Presentation layout optimizer
Presentation layout adaptor
SMIL content Database
XDI Database
*
1
2
3
4
4
5
5
1
XDI : conteXt Digital Item
CDI : Content Digital Item
* This is the adapted SMIL

Figure 8.1: Architecture of multimedia scene semantic adaptation engine

8.2.1 Scene optimizer

The scene optimizer function is to first try to find an optimal form for the temporal, spatial and logical structure of the layout of the presentation that satisfies the context constraints and semantic dependencies of the scene and to, then, convert the scene layout structure to this adapted. It works on the basis of the optimization algorithm described in previous chapter.

The optimizer module is totally implemented in Java language. Its inputs are the CDI of the SMIL scene and the corresponding XDI. Figure 8.2 demonstrates the general architecture of our scene optimizer module. The scene optimization is done for a number of context constraints such as target device display size, device capabilities (from the point of view of resource modality and format support), as well as user and author preferences. We have not implemented a scene optimization for bandwidth criteria; however, this can be done by considering the *importance* value of each media object of the scene. Meaning that the available bandwidth would be shared among media objects of the scene, proportionally to their *importance* value.

The optimization is done along a number of scene optimizing sub-modules. In the following we provide a rough description of the algorithm of scene optimizer. The optimization process walkthrough is analyzed through the description of each sub-module function.

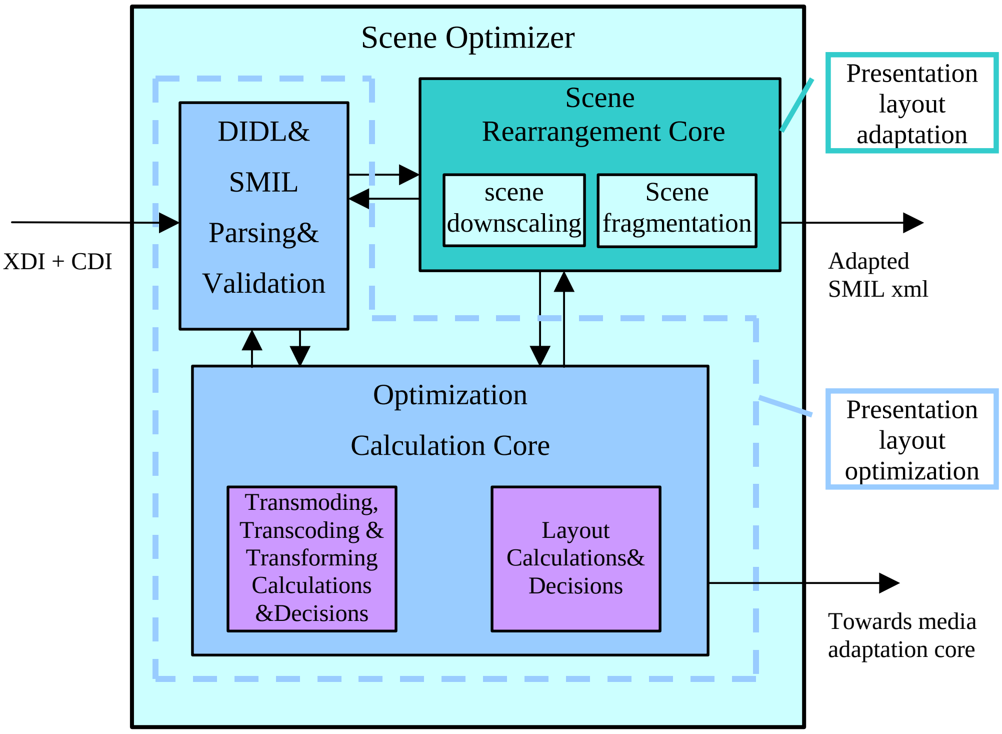

Scene Optimizer
DIDL& SMIL Parsing& Validation
Scene Rearrangement Core
scene downscaling
Scene fragmentation
Optimization Calculation Core
Transmoding, Transcoding & Transforming Calculations &Decisions
Layout Calculations& Decisions
XDI + CDI
Presentation layout adaptation
Adapted SMIL xml
Presentation layout optimization
Towards media adaptation core

Figure 8.2: Architecture of scene optimizer module

8.2.1.1 DIDL and SMIL parsing and validating

i. The optimizer takes the XDI and CDI as input.

ii. The XDI is parsed and validated through a DID+DIA parser. The UED metadata is saved into an internal structure. We used the DID parser of the MPEG-21 reference software, provided by Ghent University, and developed a DIA parser and validator that parses, validates and then saves the DIA metadata in an internal structure.

iii. The CDI file containing the SID and RCD metadata as well as the references to SMIL scene URI and its media objects URIs, is parsed and validated through a DID+RCD+SID parser, the metadata is then saved in an internal structure.

iv. The SMIL scene is retrieved form its URI. It is then parsed and validated through a SMIL parser. We implemented a SMIL parser that verifies, parses and validates our input SMIL scenes. The SMIL parser also extracts the type of the SMIL document (*simple* or *sequential*).

8.2.1.2 Optimization calculation Core

This sub-module is in fact the implementation of the first step of optimization, i.e. the part of scene optimization that is in charge of decision-making and computation of the optimal layout. It calculates the optimal form of the presentation and the media objects based on the algorithm described in section **Erreur ! Source du renvoi introuvable.**.

8.2.1.3 Scene rearrangement core

This sub-module is implemented in JAVA. It is in charge of performing the second step of optimization, i.e. the presentation layout adaptation. It applies the calculated rearrangements to the structure of the scene. It consists of two types of layout adaptation: layout downscaling and scene fragmentation and works on the basis of the algorithm described in section 7.4.1.2.

8.2.2 Media resource adaptor

The media adaptation tools, that are a set of transmoders, transcoders and transformers, will adapt the media resources following the adaptation parameters (i.e. downscaling, transmoding or transcoding parameters) that are provided by the scene optimizer.

The media adaptation tools that we used in this work are the same that we used for resource conversion framework, as described in chapter 5.

8.2.3 Exploitation and performance of MSSA engine

In this section we describe how our MSSA engine can be exploited in an adaptation framework. To do this, we provide some adaptation cases for different context constraints and describe how MSSA engine adapts the content for these adaptation cases. We also discuss the MSSA engine's performance by reviewing its advantages and weak points.

8.2.3.1 Adaptation examples for SMIL scene of Figure 7.2

Let us recall the SMIL scene of Figure 7.2. A screenshot of its layout is given in Figure 7.3. We consider this SMIL scene as the original full version of the content that should be adapted for different constraints. We consider the following adaptation cases:

Case A:

Constraints: The available display size for rendering the content is 200 * 300. This could be for example a percentage of the device display size. The player is capable of playing all media modalities: text, image (JEPG, JIF), graphics and video (MPEG). The corresponding CDI containing content-related information is provided in Figure 8.3.

```
<?xml version="1.0" encoding="UTF-8"?>
<DIDL xmlns="urn:mpeg:mpeg21:2002:01-DIDL-NS" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:mpeg:mpeg21:2003:01-DIDL-NS didl.xsd"
xmlns:sid="urn:mpeg:mpeg21:2004:01-SID-NS" xmlns:rcd="urn:mpeg:mpeg21:2003:01-DIA-NS">
 <Container>
  <Item id="smilscene1">
   <Descriptor>
    <Statement mimeType="text/xml">
     <SID>
       <FragmentationType>interactive</FragmentationType>
       <Object importance="medium" id="v1" ref="video1" maxRRF="2">
        <SemanticDependencies>
         <AbsoluteSemanticDependencies>
          <PreConditionMedia>
           <MediaObject ref="t1"/>
          </PreConditionMedia>
         </AbsoluteSemanticDependencies>
         <SpatialSemanticDependencies>
          <KeepCloseTo>
           <MediaObject ref="t1"/>
          </KeepCloseTo>
         </SpatialSemanticDependencies>
        </SemanticDependencies>
        <SemanticFragmentationPreferences>
         <TimeFragmentationPreferences shiftingPriority="2"/>
         <SpatialFragmentationPreferences keepWith="t1" position="left"/>
        </SemanticFragmentationPreferences>
       </Object>
       <Object importance="low" id="t1" ref="text1" maxRRF="3">
        <SemanticDependencies>
         <SpatialSemanticDependencies>
          <KeepCloseTo>
           <MediaObject ref="i1"/>
          </KeepCloseTo>
         </SpatialSemanticDependencies>
        </SemanticDependencies>
        <SemanticFragmentationPreferences>
         <TimeFragmentationPreferences shiftingPriority="1"/>
         <SpatialFragmentationPreferences keepWith="v1" position="right-below"/>
        </SemanticFragmentationPreferences>
       </Object>
       <Object importance="medium" id="i2" ref="image2" maxRRF="2">
        <SemanticDependencies>
         <SpatialSemanticDependencies>
          <KeepCloseTo>
           <MediaObject ref="t2"/>
          </KeepCloseTo>
         </SpatialSemanticDependencies>
         </SemanticDependencies>
         <SemanticFragmentationPreferences>
          <TimeFragmentationPreferences shiftingPriority="4"/>
          <SpatialFragmentationPreferences keepWith="t2" position="right-up"/>
         </SemanticFragmentationPreferences>
        </Object>
        <Object importance="low" id="t2" ref="text2" maxRRF="3" role="redundant">
         <SemanticDependencies>
          <SpatialSemanticDependencies>
           <KeepCloseTo>
            <MediaObject ref="i2"/>
           </KeepCloseTo>
          </SpatialSemanticDependencies>
          <AbsoluteSemanticDependencies>
           <RedundantFor>
             <MediaObject ref="i2"/>
           </RedundantFor>
          </AbsoluteSemanticDependencies>
         </SemanticDependencies>
         <SemanticFragmentationPreferences>
          <TimeFragmentationPreferences shiftingPriority="3"/>
          <SpatialFragmentationPreferences keepWith="i2" position="left-below"/>
         </SemanticFragmentationPreferences>
        </Object>
        <Object importance="medium" id="i3" ref="image3" maxRRF="2">
         <SemanticDependencies>
          <AbsoluteSemanticDependencies>
           <PreConditionMedia>
            <MediaObject ref="t3"/>
```

```
      </PreConditionMedia>
     </AbsoluteSemanticDependencies>
     <SpatialSemanticDependencies>
      <KeepCloseTo>
       <MediaObject ref="t3"/>
      </KeepCloseTo>
     </SpatialSemanticDependencies>
    </SemanticDependencies>
    <SemanticFragmentationPreferences>
     <TimeFragmentationPreferences shiftingPriority="2"/>
     <SpatialFragmentationPreferences keepWith="t3" position="left"/>
    </SemanticFragmentationPreferences>
    </Object>
    <Object importance="low" id="t3" ref="text3" maxRRF="3" >
     <SemanticDependencies>
      <SpatialSemanticDependencies>
       <KeepCloseTo>
        <MediaObject ref="i3"/>
       </KeepCloseTo>
      </SpatialSemanticDependencies>
     </SemanticDependencies>
     <SemanticFragmentationPreferences>
      <TimeFragmentationPreferences shiftingPriority="1"/>
      <SpatialFragmentationPreferences keepWith="i3" position="right-below"/>
     </SemanticFragmentationPreferences>
    </Object>
   </SID>
  </Statement>
 </Descriptor>
 <Component id="smilScene">
  <Descriptor>
   <Statement mimeType="text/plain">this is the smil scene</Statement>
  </Descriptor>
  <Resource mimeType="application/smil" ref="posterExample.smil"/>
 </Component>
</Item>
<Item>
 <Descriptor>
  <Statement mimeType="text/plain">item containing medias and related info</Statement>
 </Descriptor>
 <Choice minSelections="1" maxSelections="1">
  <Descriptor>
   <Statement mimeType="text/plain">What resolution?</Statement>
  </Descriptor>
  <Selection select_id="originalSize">
   <Descriptor><Statement mimeType="text/plain">original</Statement></Descriptor>
  </Selection>
  <Selection select_id="RRFis2">
   <Descriptor>
    <Statement mimeType="text/plain">downscaled media with RRF=2</Statement>
   </Descriptor>
  </Selection>
  <Selection select_id="RRFis3">
   <Descriptor>
    <Statement mimeType="text/plain">maximum-downscaled media with RRF=3</Statement>
   </Descriptor>
  </Selection>
 </Choice>
 <Item id="video1">
  <Component >
   <Condition require="originalSize"/>
   <Resource mimeType="video/mpeg" ref="v1.mpg"/>
  </Component>
 </Item>
  <Item id="text1">
   <Component>
    <Condition require="originalSize"/>
    <Resource ref="t1.txt" mimeType="text/plain"/>
   </Component>
   <Component>
    <Condition require="RRFis3"/>
     <Resource ref="maximumSummarizedT1.txt" mimeType="text/plain"/>
   </Component>
   <Component>
    <Condition require="RRFis2"/>
     <Resource ref="summarizedT1.txt" mimeType="text/plain"/>
   </Component>
```

```
      </Item>
      <Item id="image2">
       <Component >
        <Condition require="originalSize"/>
        <Resource mimeType="image/jpg" ref="i2.jpg"/>
       </Component>
      </Item>
      <Item id="text2">
       <Component>
        <Condition require="originalSize"/>
        <Resource ref="t2.txt" mimeType="text/plain"/>
       </Component>
       <Component>
        <Condition require="RRFis3"/>
        <Resource ref="maximumSummarizedT2.txt" mimeType="text/plain"/>
       </Component>
       <Component>
        <Condition require="RRFis2"/>
        <Resource ref="summarizedT2.txt" mimeType="text/plain"/>
       </Component>
      </Item>
      <Item id="image3">
       <Component >
        <Resource mimeType="image/jpg" ref="i3.jpg"/>
       </Component>
      </Item>
      <Item id="text3">
       <Component>
        <Condition require="originalSize"/>
        <Resource ref="t3.txt" mimeType="text/plain"/>
       </Component>
       <Component>
        <Condition require="RRFis3"/>
        <Resource ref="maximumSummarizedT3.txt" mimeType="text/plain"/>
       </Component>
       <Component>
        <Condition require="RRFis2"/>
        <Resource ref="summarizedT3.txt" mimeType="text/plain"/>
       </Component>
      </Item>
     </Item>
   </Container>
</DIDL>
```

Figure 8.3: CDI for example of case A

As expressed within the CDI, for each text media, two summarized versions, corresponding to Resolution Reduction Factors (RRF) of 2 and 3 are provided. This is to help reducing the occupied space of text media objects when needed. If no summarized versions of text media objects are provided, it can be envisaged to reduce the text font size or apply online text summarizing algorithms, in order to downscale the text media objects.

Adaptation solution by MSSA: As the original modality and format of the media objects are supported, no transmoding and transcoding is done. The adapted dimensions of visual media objects are calculated and corresponding transformings (media resizings) are done. Based on spatial calculations, the original texts cannot be put in the adapted scene, and the maximum-summarized versions of texts are selected. The adapted scene is an interactive vertically fragmented scene, as shown in Figure 8.4. The adapted values of media positions and dimensions can be seen in this figure. Three screenshots of the three fragments of the adapted scene, played with RealPlayer 10, are provided in Figure 8.5. Please note that in this figure, in order to use less room, the screenshots are downscaled with a factor of 60%.

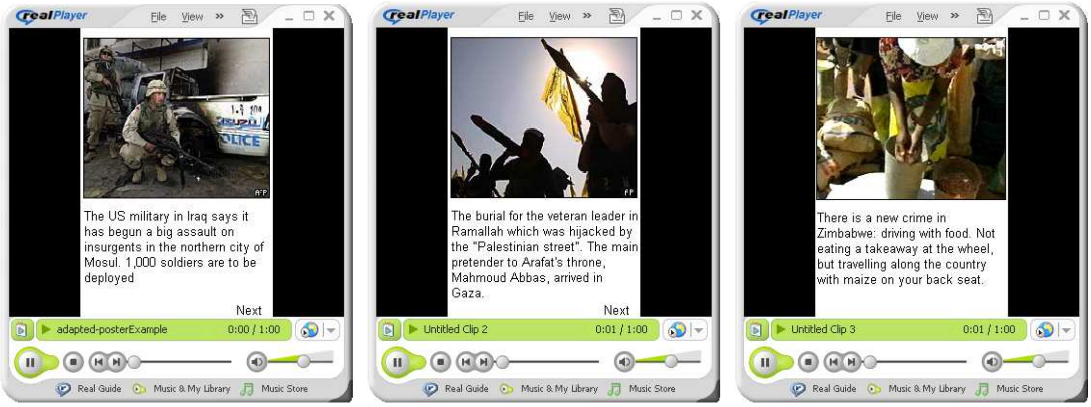
It can be seen that the available display size has been optimally used. In the last fragment no “Next” button is needed, a bit more space is hence available to render the image and text.

Case B:

Constraints: The available display size is the same as in case A. Based on device player capabilities and user preferences, the accepted media modalities are: text and image (JEPG, JIF). The corresponding CDI containing content-related information is almost the same as in Figure 8.3, except for the *maxRRF* value of the t2, which is, this time, set to 1. Also, in order to express the transmoding parameters of a video-to-image conversion, the corresponding *Item* element of video media is changed as given in Figure 8.6.

```
<smil>
    <head>
        <layout>
            <root-layout background-color="#ffffff" height="300" width="200"/>
            <region fit="fill" height="167" id="video1" left="3" top="10" width="194"/>
            <region fit="fill" height="167" id="image2" left="3" top="10" width="194"/>
            <region fit="fill" height="169" id="image3" left="1" top="10" width="197"/>
            <region fit="fill" height="99" id="text1" left="2" top="187" width="196"/>
            <region fit="fill" height="99" id="text2" left="2" top="187" width="196"/>
            <region fit="fill" height="101" id="text3" left="0" top="189" width="200"/>
            <region height="15" id="nextButtonRegion" left="160" top="285" width="30"/>
        </layout>
    </head>
    <body>
        <par>
            <video dur="60s" id="v1" region="video1" src="v1.mpg"/>
            <text dur="60s" id="t1" region="text1" src="maximumSummarizedT1.txt"/>
            <a href="#page1"><text region="nextButtonRegion" src="next.txt"/></a>
        </par>
        <par id="page1">
            <img dur="60s" id="i2" region="image2" src="i2.jpg"/>
            <text dur="60s" id="t2" region="text2" src="maximumSummarizedT2.txt"/>
            <a href="#page2"><text region="nextButtonRegion" src="next.txt"/></a>
        </par>
        <par id="page2">
            <img dur="60s" id="i3" region="image3" src="t3.jpg"/>
            <text dur="60s" id="t3" region="text3" src="maximumSummarizedT3.txt"/>
        </par>
    </body>
</smil>
```

Figure 8.4: Adapted SMIL scene for adaptation case A



The US military in Iraq says it has begun a big assault on insurgents in the northern city of Mosul. 1,000 soldiers are to be deployed

The burial for the veteran leader in Ramallah which was hijacked by the "Palestinian street". The main pretender to Arafat's throne, Mahmoud Abbas, arrived in Gaza.

There is a new crime in Zimbabwe: driving with food. Not eating a takeaway at the wheel, but travelling along the country with maize on your back seat.

Figure 8.5: Screenshots of fragments of the adapted scene for adaptation case A

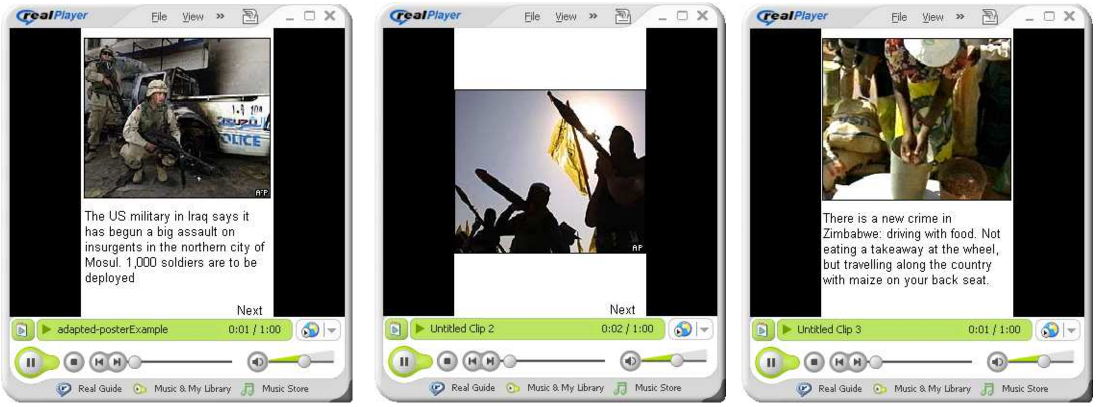
```
<Item id="video1">
 <Component >
  <Condition require="originalSize"/>
  <Descriptor><Statement mimeType="text/plain">rcd metadta</Statement></Descriptor>
  <Descriptor>
   <Statement mimeType="text/xml">
    <dia:ConversionInformation >
     <rcd:ConversionDescription xsi:type="rcd:TransmodingConversionType">
      <dia:ConversionUri>http://www.ensttransmodingTool.com/videotoimage</dia:ConversionUri>
      <rcd:Transmoding>
       <rcd:Parameters xsi:type="rcd:VideoSummarizationParametersType">
        <rcd:To href="urn:mpeg:mpeg7:cs:ContentCS:2001">
         <mpeg7:Name>Image</mpeg7:Name>
        </rcd:To>
        <rcd:Slide importance="hight">
         <mpeg7:MediaTimePoint>T00:00:00:01F24</mpeg7:MediaTimePoint>
        </rcd:Slide>
       </rcd:Parameters>
      </rcd:Transmoding>
     </rcd:ConversionDescription>
    </dia:ConversionInformation>
   </Statement>
  </Descriptor>
  <Resource mimeType="video/mpeg" ref="v1.mpg"/>
 </Component>
</Item>
```

Figure 8.6: DIDL *Item* element for video media object of adaptation case B

Adaptation solution by MSSA: As a modality mismatch was found between the present modalities in the original scene and allowed modalities, a video-to-image transmoding is decided, calculated and performed. The conversion parameters are extracted form the CDI. The adapted dimensions of visual media objects are calculated and corresponding transformings (media resizings) and substitutions are done. Based on t1 and t3 *maxRRF* values and spatial calculations, the optimum alternatives are selected for t1 and t3. t1 and t3 are substituted by these selected summarized alternatives.

```
<smil>
    <head>
        <layout>
            <root-layout background-color="#ffffff" height="300" width="200"/>
            <region fit="fill" height="167" id="video1" left="3" top="10" width="194"/>
            <region fit="fill" height="171" id="image2" left="0" top="64" width="200"/>
            <region fit="fill" height="169" id="image3" left="1" top="10" width="197"/>
            <region fit="fill" height="99" id="text1" left="2" top="187" width="196"/>
            <region fit="fill" height="101" id="text3" left="0" top="189" width="200"/>
            <region height="15" id="nextButtonRegion" left="160" top="285" width="30"/>
        </layout>
    </head>
    <body>
        <par>
            <img dur="60s" id="transmodedv1" region="video1" src="transmodedv1.jpg"/>
            <text dur="60s" id="t1" region="text1" src="maximumSummarizedT1.txt"/>
            <a href="#page1"><text region="nextButtonRegion" src="next.txt"/></a>
        </par>
        <par id="page1">
            <img dur="60s" id="i2" region="image2" src="i2.jpg"/>
            <a href="#page2"><text region="nextButtonRegion" src="next.txt"/></a>
        </par>
        <par id="page2">
            <img dur="60s" id="i3" region="image3" src="i3.jpg"/>
            <text dur="60s" id="t3" region="text3" src="maximumSummarizedT3.txt"/>
        </par>
    </body>
</smil>
```

Figure 8.7: Adapted SMIL scene for adaptation case B


The US military in Iraq says it has begun a big assault on insurgents in the northern city of Mosul. 1,000 soldiers are to be deployed

Next

adapted-posterExample 0:01 / 1:00

Next

Untitled Clip 2 0:02 / 1:00

There is a new crime in Zimbabwe: driving with food. Not eating a takeaway at the wheel, but travelling along the country with maize on your back seat.

Untitled Clip 3 0:01 / 1:00

Figure 8.8: Screenshots of the fragments of the adapted scene for adaptation case B

This is not the case for t2. As its *maxRRF* value is 1, no downscaling is allowed. Because of space limitation t2 was removed from the scene as it was of *low importance*. The adapted scene is an interactive vertically fragmented scene as shown in Figure 8.7. Screenshots of the three fragments of the adapted scene are provided in Figure 8.8. The screenshots are downscaled with a factor of 60%.

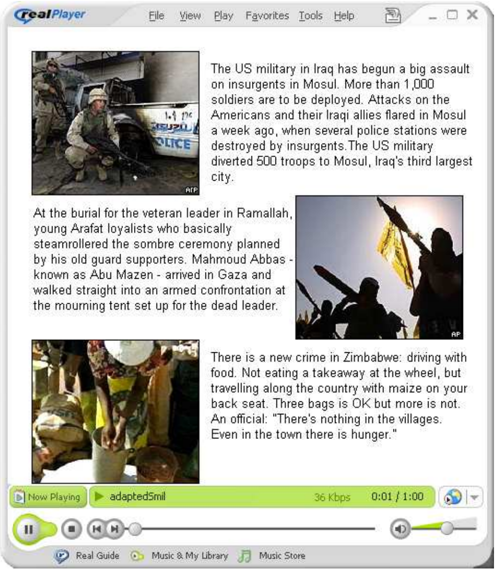
Case C:

Constraints: The available display size is 500*475. Other constraints are the same as for case A.

Adaptation solution by MSSA: Scene adaptation is done without any scene fragmentation. The video and images are resized. Optimum summarized versions are selected to replace text media objects. The adapted SMIL scene is as in Figure 8.9. Figure 8.10 shows a screenshot of the adapted scene.

```
<smil>
    <head>
        <layout>
            <root-layout background-color="#ffffff" height="475" width="500"/>
            <region background-color="#ffffff" fit="fill" height="150" id="video1" left="25"
top="25" width="175"/>
            <region background-color="#ffffff" fit="fill" height="150" id="image2" left="300"
top="175" width="175"/>
            <region background-color="#ffffff" fit="fill" height="150" id="image3" left="25"
top="325" width="175"/>
            <region background-color="#ffffff" fit="fill" height="140" id="text1" left="210"
top="35" width="275"/>
            <region background-color="#ffffff" fit="fill" height="140" id="text2" left="25"
top="185" width="275"/>
            <region background-color="#ffffff" height="140" id="text3" left="210" top="335"
width="275"/>
        </layout>
    </head>
    <body>
        <par>
            <video dur="60s" id="v1" region="video1" src="v1.mpg"/>
            <text dur="60s" id="t1" region="text1" src="summarizedT1.txt"/>
            <img dur="60s" id="i2" region="image2" src="i2.jpg"/>
            <text dur="60s" id="t2" region="text2" src="summarizedT2.txt"/>
            <img dur="60s" id="i3" region="image3" src="i3.jpg"/>
            <text dur="60s" id="t3" region="text3" src="summarizedT3.txt"/>
        </par>
    </body>
</smil>
```

Figure 8.9: Adapted SMIL scene for adaptation case C



The US military in Iraq has begun a big assault on insurgents in Mosul. More than 1,000 soldiers are to be deployed. Attacks on the Americans and their Iraqi allies flared in Mosul a week ago, when several police stations were destroyed by insurgents.The US military diverted 500 troops to Mosul, Iraq's third largest city.

At the burial for the veteran leader in Ramallah, young Arafat loyalists who basically steamrollered the sombre ceremony planned by his old guard supporters. Mahmoud Abbas - known as Abu Mazen - arrived in Gaza and walked straight into an armed confrontation at the mourning tent set up for the dead leader.

There is a new crime in Zimbabwe: driving with food. Not eating a takeaway at the wheel, but travelling along the country with maize on your back seat. Three bags is OK but more is not. An official: "There's nothing in the villages. Even in the town there is hunger."

Figure 8.10: Screenshot of the adapted layout for adaptation case C

Case D:

Constraints: The available display size is 144*176. Other constraints are the same as in case A.

Adaptation solution by MSSA: A vertical scene fragmentation is done. Due to lack of space, text media objects are all removed from the scene since they are of *low importance*. The adapted SMIL scene is as shown in Figure 8.11. Screenshots of the three fragments of the adapted scene are shown in Figure 8.12.

In this example we can imagine that if the images and the video were of low *importance* and the text media objects were of medium or high *importance*, the images and the video would have been removed from the scene and the text media objects would have remained in the adapted scene.

```
<smil>
    <head>
        <layout>
            <root-layout background-color="#ffffff" height="176" width="144"/>
            <region fit="fill" height="123" id="video1" left="0" top="13" width="144"/>
            <region fit="fill" height="123" id="image2" left="0" top="13" width="144"/>
            <region fit="fill" height="123" id="image3" left="0" top="13" width="144"/>
            <region height="15" id="nextButtonRegion" left="104" top="161" width="30"/>
        </layout>
    </head>
    <body>
        <par>
            <video dur="60s" id="i1" region="video1" src="v1.jpg"/>
            <a href="#page1"><text region="nextButtonRegion" src="next.txt"/></a>
        </par>
        <par id="page1">
            <img dur="60s" id="i2" region="image2" src="i2.jpg"/>
            <a href="#page2"><text region="nextButtonRegion" src="next.txt"/></a>
        </par>
        <par id="page2">
            <img dur="60s" id="i3" region="image3" src="i3.jpg"/>
        </par>
    </body>
</smil>
```

Figure 8.11: Adapted SMIL content for adaptation case D

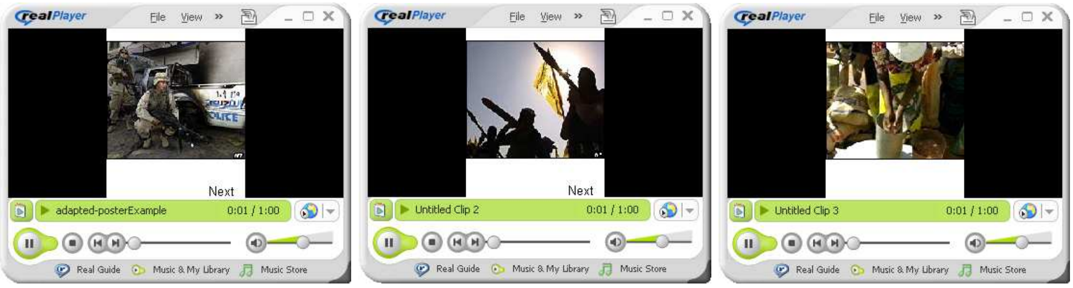
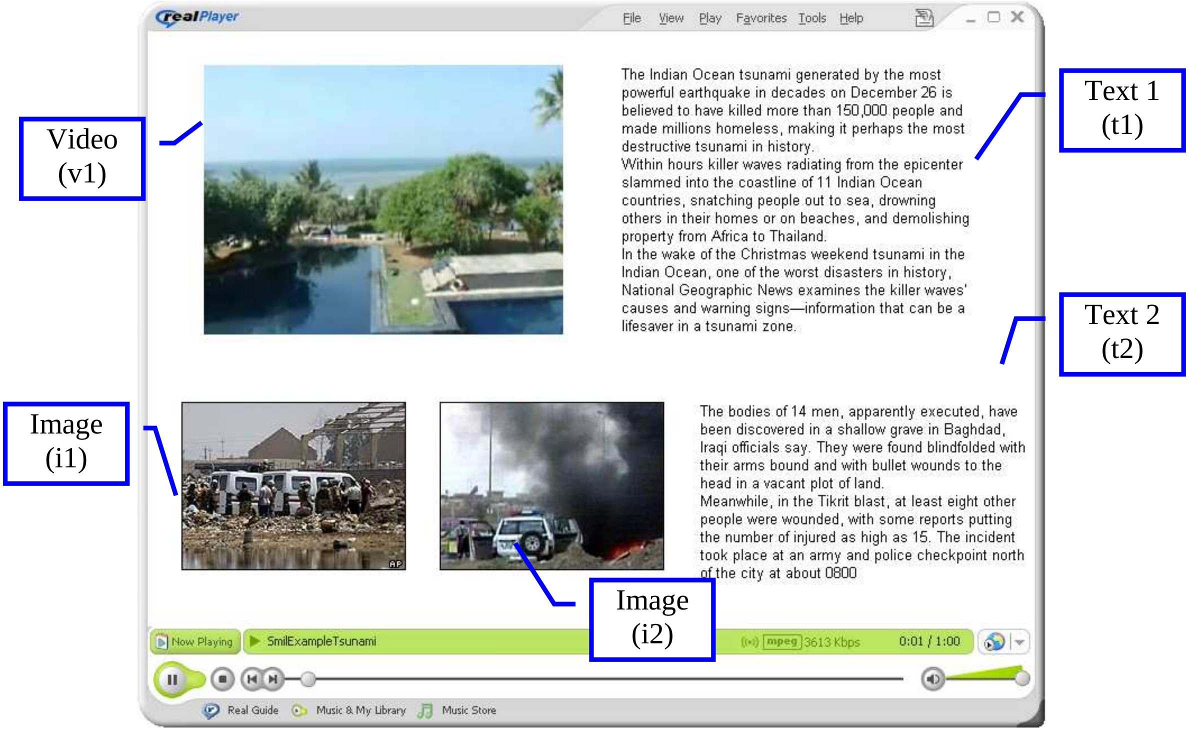

realPlayer
File View »
Next
adapted-posterExample 0:01 / 1:00
Untitled Clip 2 0:01 / 1:00
Untitled Clip 3 0:01 / 1:00
Real Guide
Music & My Library
Music Store

Figure 8.12: Screenshots of the fragments of the adapted scene for adaptation case D

8.2.3.2 Another adaptation example

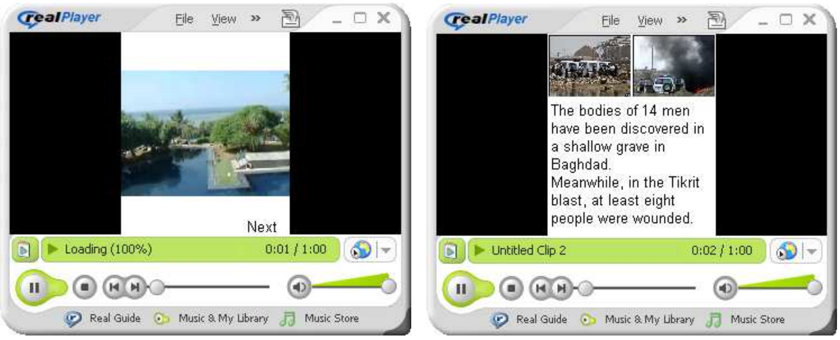
Let us very shortly go through another adaptation example. The original SMIL document is shown in Figure 8.13 (70% downscaled); it contains five media objects: one video resource and one text resource that are semantically related, and two image resources that are semantically related to another text resource. The available display size of the target device is 150*180 pixels. The first text (t1) is of low importance, with *maxRRF* value of 1. The other text resource (t2), the video and the images are of high importance and their *maxRRF* values are 3. Summarized version of t2 is available. The target device supports video, image and text modalities. The adapted SMIL scene is a fragmented scene as shown in Figure 8.14 (70% downscaled). The first fragment contains only the video resource, as t1 text resource is eliminated from the scene. The other three semantically-related media objects, i.e. i1, i2 and t2 are put together in a second fragment. Summarized version of t2 has replaced it.



The Indian Ocean tsunami generated by the most powerful earthquake in decades on December 26 is believed to have killed more than 150,000 people and made millions homeless, making it perhaps the most destructive tsunami in history.
Within hours killer waves radiating from the epicenter slammed into the coastline of 11 Indian Ocean countries, snatching people out to sea, drowning others in their homes or on beaches, and demolishing property from Africa to Thailand.
In the wake of the Christmas weekend tsunami in the Indian Ocean, one of the worst disasters in history, National Geographic News examines the killer waves' causes and warning signs—information that can be a lifesaver in a tsunami zone.

The bodies of 14 men, apparently executed, have been discovered in a shallow grave in Baghdad, Iraqi officials say. They were found blindfolded with their arms bound and with bullet wounds to the head in a vacant plot of land.
Meanwhile, in the Tikrit blast, at least eight other people were wounded, with some reports putting the number of injured as high as 15. The incident took place at an army and police checkpoint north of the city at about 0800

Figure 8.13: A SMIL document containing one video, two text and two image resources



The bodies of 14 men have been discovered in a shallow grave in Baghdad. Meanwhile, in the Tikrit blast, at least eight people were wounded.

Figure 8.14: Adapted SMIL scene of Figure 8.13 for example of section 8.2.3.2

8.2.3.3 Advantages and weak points of MSSA engine

As we saw in above adaptation examples, the MSSA engine can be exploited in a multimedia content adaptation framework, to serve adaptation of SMIL 2.0 documents for limited devices. The methodology, however, is independent of the choice of SMIL and can be applied to other types of multimedia documents.

The most valuable advantage of the MSSA engine is that it reduces the workload of the authors. Instead of designing several presentation layouts for different terminal player capabilities and display sizes, the author provides only the full version of his multimedia document and indicates its semantic information by means of simple XML descriptors. The MSSA engine then takes care of the rest by providing a personalized version of the original document for each user request.

We experienced that position and dimension calculations proved to be complicated for scenes with high number of media objects. Also it is true that for complex SMIL contents with numerous hyper-linking and timing dependencies, the programming task may become quite complicated, but the methodology remains valid.

If such engine is integrated into a client-server architecture, based on the capacities of the server, important response delays may be expected in case of numerous simultaneous adaptation requests. Distributed architectures are more appropriate in such cases.

One important missing point of our MSSA engine is that, in its current from, it does not deal with the question of adaptation for bandwidth limitations. Besides, the optimization of bandwidth sharing between media objects is not dealt with. Theses features can be added to the implementation by using *AdaptationQoS* tool and taking into account the *importance* of each media object for bandwidth sharing optimization.

We have chosen to implement our MSSA engine for SMIL 2.0 documents, however, through the implementation of MSSA, we learned that multimedia documents, whose spatial layouts are pre-defined in an absolute manner, are not best adaptable compared to device independent multimedia documents. Multimedia scene description languages that are rendering-independent, allow describing spatial constraints rather than exact media positions and dimensions. This makes the spatial layout optimization process less heavy. RIML is one of such languages (see section 6.5.9). Its weak point is that it only allows describing spatial constraints; semantic dependencies are not completely considered within RIML model. Another possibility is the addition of constraints to languages with absolute layout by default. Cameron McCormack et al. propose such a constraint extension for SVG, resolved through the use of scripting [100]. Semantic dependencies are not dealt with in this proposal, either.

8.3 Conclusion

In this chapter, we described the architecture of our implementation of a multimedia scene semantic adaptation engine, which was realized based on the methodology described in previous chapter. We also described the algorithm of our scene optimizing modules.

We used a set of descriptors – as defined in previous chapter – for describing semantic information of a multimedia scene such as inter-media semantic, spatial and temporal dependencies, fragmentation preferences, and also each media independent semantic information. Through this proof-of-concept implementation, we demonstrated that in order to perform a correct adaptation that preserves the consistency and the meaningfulness of the adapted scene, the adaptation process needs to have access to a number of semantic information of the presentation.

The scene layout adaptation is done for a generic layout model and for satisfying context constraints such as target device capabilities and user preferences, as well as satisfying content constraints, such as scene semantic and media conversion information.

Semantic adaptation of structured multimedia documents is a complex issue and needs to be addressed more completely. The order of complexity grows more significant as complex temporal dependencies are introduced between media objects of a scene. A perspective of this work is enhancing the set of resource adaptors and evaluating bandwidth limitations and the usage of MPEG-21 AQoS for optimization of media resources bandwidth sharing.

Chapter 9

CONCLUSIONS AND PERSPECTIVES

In this dissertation, we formulated new solutions under the framework of the on-going MPEG-21 standard, for adaptation of single media content and rich synchronized multimedia-composed content, to constrained contexts, i.e. limited devices, user preferences and author constraints.

9.1 Summary and achievements

After giving an overview of MPEG-21 standard in Chapter 2, in Chapter 3we described the concept of single media adaptation and analyzed its principal components. In the same chapter we discussed the existing approaches in this area.

Chapter 4and Chapter 5describe the first part of the presented work, that is in the area of resource adaptation. In Chapter 4 we introduced our solutions for an MPEG-21 Resource Conversion framework. We presented description tools for expression of conversion-related preferences as well as conversion parameters. We started by presenting a description tool for transmoding conversions, which we used in the implementation of a media transmoding module in ISIS IST project. We then described the enhancement and generalization of this description tool to a complete conversion description tool that covers other conversion types: transcoding and transforming. We also presented another tool for dynamic description of conversion parameter values that uses the DIA *AdaptationQoS*. This description tool, called *ConversionLink*, was developed under the framework of DANAE and TIRAMISU IST projects. In Chapter 5, we presented the architecture of our transmoding and Resource Conversion implementations.

Our work on the subject of Resource Conversion was contributed to the MPEG-21 standard. We contributed descriptors for the expression of media conversion preferences to DIA (*ConversionPreference*). An amendment to DIA was established on December 2003 and is still on-going. This amendment aims at adding the support of resource conversion to DIA. The current amendment promotes a generic descriptor for the description of conversion-related information in DIA. Currently

ENST is leading a *Core Experiment* for specifying a new description tool that uses the DIA *AdaptationQoS* tool for dynamic expression of conversion parameter values, and helps support a more intelligent adaptation decision-making process.

In Chapter 6 we described the basic concepts and principal requirements of a multimedia-composed adaptation framework, and discussed the existing approaches in this area. Chapter 7described our methodology for support of a scene semantic adaptation framework. We presented description tools for expression of semantic information of multimedia document. We also described how these tools could be used under the framework of MPEG-21 DIA. Lastly, Chapter 8 presented the architecture of our implementation of a proof-of-concept semantic adaptation engine for SMIL content.

Our work in the area of adaptation of rich multimedia content proposes solutions for semantic adaptation of multimedia synchronized scenes. We proved that the adaptation engine requires semantic information of multimedia content in order to perform a correct, consistent and meaningful adaptation on it, and we implemented our proposed semantic adaptation solutions.

9.2 Areas of future work

This work could be continued in several directions, both in the areas of single media content adaptation and multimedia-composed content semantic adaptation.

9.2.1 Single media content adaptation

As we explained earlier in this document, an activity, led by ENST has begun in the MPEG-21 subgroup on the usage of DIA *AdaptationQoS* for the description and selection of conversion-related parameters values. This tool (*ConversionLink*) is a new tool similar to *BSDLink* for (g)BSD-adaptation, which describes how to retrieve parameters for a conversion as a function of UED and UCD constraints by using *AdaptationQoS*. We believe that this direction will bring interesting solutions for support of any kind of media conversion in MPEG-21 DIA. However, this tool has not yet been completely implemented, and the harmonization of ConversionLink with other DIA tools needs to be further investigated.

9.2.2 Multimedia-composed content semantic adaptation

In the area of semantic adaptation of multimedia-composed documents, several directions can be envisaged for the continuation of our work.

One immediate continuation of the work on MSSA is to take into account the question of the adaptation of the scene against bandwidth limitations of the connection. It will be interesting to investigate the possibilities of optimization of bandwidth sharing between the different media objects of the scene based on their respective *importance*.

Our experience with scene description languages like SMIL proves that high-level scene description languages (like XMT-O) are more easy to use for semantic adaptation compared to low-level description languages (like XMT-A). However since both these types of languages rather define exact positionings and occupied dimensions of media objects in the layout, even by having access to full semantic information of the scene, the implementation of correct-calculation of visual media objects positions and resolutions in the adapted layout is very complex, especially for scenes with high number of media objects. We believe that the best way toward an efficient, consistent, and at the same time simple multimedia semantic adaptation framework is to develop adaptable multimedia description languages that are device-independent. RIML is an example of such a description language, but it only addresses the device-independent media positioning issues and does not deal with the between-media semantic dependencies and media independent semantic information. Hence, research on the subject of an adaptable device-independent multimedia scene description language that addresses the problem of semantic adaptation of multimedia content is an appropriate direction for future work in this area.

RESUME LONG FRANÇAIS

Introduction

Aujourd'hui, afin de pouvoir fournir et consommer des contenus multimédia de manière transparente, il est important de disposer d’une infrastructure d'adaptation. Sans une telle infrastructure, les créateurs de contenus et les fournisseurs de services sont confrontés à plusieurs problèmes pour fournir un contenu multimédia à leurs consommateurs. Entre autres, l'augmentation, souvent non réaliste, du nombre des versions de contenus à produire, à stocker et à distribuer, ainsi que l’impossibilité de servir à chaque client une version optimale correspondant à sa configuration sont autant de limitations qui interdisent un accès universel aux contenus multimédia.

Une infrastructure d’adaptation nécessite une description des contenus multimédia et du contexte de consommation de l’utilisateur ainsi qu’un ensemble d’informations guidant le processus d’adaptation. C’est en partie aux créateurs de contenu de prendre en compte les dispositifs nécessaires à l’adaptation, et ceci, en créant des contenus adaptables et en fournissant les méta-données nécessaires à l'adaptation.

Plusieurs groupes internationaux de standardisation comme le W3C (*World Wide Web Consortium*) et MPEG (*Moving Picture Experts Group*) proposent des solutions pour aider et guider des infrastructures d’adaptation de contenus multimédia. Par exemple, le cadre de travail de CC/PP (*Composite Capabilities/ Preferences Profile*) [?], défini par le W3C, définit un modèle de description des caractéristiques de terminaux et de préférences utilisateurs pour guider le processus d’adaptation de contenu. D’autre part, MPEG-21 [?] normalise dans sa partie 7 (Digital Item Adaptation) des techniques d'adaptation de contenus multimédia.

Le travail présenté dans ce manuscrit est basé sur la norme MPEG-21 (en cours de spécification [?]) et propose des méthodologies pour résoudre les problèmes d'adaptation de contenus multimédia aux terminaux limités et aux différents contextes d'utilisation. On entend par terminaux limités des terminaux de capacités réduites en terme de résolution d'écran, de support des codecs, de taille de buffer, de puissance, etc.

Dans notre étude de l'adaptation de contenus multimédia aux contextes contraints, nous avons délibérément distingué deux types d’adaptation: l’adaptation du média lui-même et l’adaptation sémantique des documents multimédia composés. Dans l'adaptation "mono-média", les médias sont considérés comme des entités indépendantes, hors de tout contexte de présentation structurée, ainsi qu’indépendamment de la composition multimédia (scène) dans laquelle ils sont utilisés. L'adaptation sémantique des documents multimédia composés, quant à elle, est basée sur les rapports temporels, spatiaux et sémantiques entre leurs objets médias. Cependant, l'adaptation de contenu mono-média constitue une partie inextricable de l'adaptation sémantique des scènes multimédia composées. En effet, dans l'adaptation sémantique des scènes multimédia, chaque objet média est adapté aux contraintes du contexte d'utilisation (taille d'affichage du terminal, capacités de décodage, préférences utilisateurs, etc.), tout en prenant en compte le scénario et la logique de la présentation. D’autre part, la scène, c'est-à-dire la structure spatiale, temporelle et logique du document, est également adaptée aux contraintes de contexte aussi bien qu'à des contraintes sémantiques de la présentation, et ce, indépendamment des ressources médias.

Cette thèse de doctorat a été effectuée dans le groupe MER (Multimédia et Réseaux) à l’ENST Paris (Ecole Nationale Supérieure des Télécommunications) et propose des solutions innovantes pour traiter les deux types d’adaptation décrits ci-dessus. Le travail a été principalement financé par France Télécom R&D, et en partie effectué dans le cadre de deux projets européens IST: ISIS [?] et DANAE [?].

Adaptation de contenu multimédia

La plupart des contenus multimédia actuels sont construits pour être consommé sur des réseaux haut-débits et par des terminaux puissants. Pourtant, les progrès techniques de ces dernières années ont permis la consommation de contenus multimédia sur les réseaux bas-débits comme les réseaux sans fil, et par des terminaux aux ressources réduites comme les téléphones mobiles et les agendas de poche électroniques. Avec la montée en puissance du nouveau marché du multimédia sur téléphone mobile, le besoin de systèmes d’adaptation de contenus à un contexte est de plus en plus fort.

Définissions l’adaptation de contenu multimédia en tant que transformation de l’état original d’un contenu à un état final, afin de satisfaire un ensemble de contraintes liées au contexte d’utilisation, de sorte que l’état final soit compatible avec ces contraintes. Le contenu adapté peut être obtenu directement à partir du contenu original ou par l’utilisation de variantes existantes de ce dernier. L'adaptation est parfois également appelée personnalisation.

Les principaux besoins d’un système d’adaptation de contenu sont : le contenu adaptable, les informations sur le contenu (caractéristiques physiques, informations sémantiques), les informations sur le contexte d’utilisation (capacités du terminal et réseau, préférences de l’utilisateur, recommandations de l’auteur, etc.), et les outils d’adaptation.

Les normes multimédia concernées par l'adaptation

Une chaîne de distribution de contenus multimédia comprend la création, la production, l’adaptation, la livraison et la consommation du contenu. Pour atteindre ces objectifs, le contenu doit être identifié, décrit, contrôlé et protégé. Le transport et la livraison du contenu multimédia s’opèrent sur un ensemble hétérogène de terminaux et de réseaux pouvant donner lieu au report d’événements variés (*event reporting*). Un tel mécanisme de distribution inclura une livraison fiable, une gestion des caractéristiques et des préférences des utilisateurs, tout en respectant la sécurité de ces données, et une gestion des transactions financières.

Une infrastructure particulière est requise pour ce nouveau type d'utilisation de contenus multimédia, permettant d’assurer l’interopérabilité de systèmes fournissant des contenus multimédia, ainsi que la simplification, et si possible l’automatisation, des transactions. Ceci exige une vision partagée par tous les participants afin d'intégrer les technologies assurant la sécurité de la livraison des contenus, la sécurité de paiement ainsi que la gestion de droits.

Certaines normes du W3C et de MPEG définissent des solutions pour le support d’une infrastructure d’adaptation de contenus multimédia. Dans les paragraphes suivants nous donnons une courte introduction sur quelques solutions existantes.

CC/PP et RDF

CC/PP est basé sur XML [?] ainsi que sur RDF (*Resource Description Framework*) [?]. CC/PP se limite à la description des contextes et permet uniquement de décrire de manière statique les caractéristiques logicielles et matérielles d’un terminal ainsi que les préférences de l’utilisateur. RDF est une autre norme du W3C, qui permet l’expression et l’utilisation intelligente des méta-données. Elle associe aux ressources un ensemble de méta-données les décrivant.

CC/PP et RDF restent des solutions incomplètes car elles se limitent à la description statique d'un sous-ensemble restreint des informations nécessaires à l’adaptation. Par exemple, elles ne mettent à disposition aucun moyen pour la description des droits d’utilisation ou des paramètres nécessaires à l’adaptation. Des travaux de recherche basés sur CC/PP et RDF ont été effectués, proposant des extensions à CC/PP [?] [?].

MPEG-21

Le groupe MPEG normalise aussi des outils qui fournissent du support pour l'adaptation de contenus multimédia. MPEG-21 est une norme ISO de la famille MPEG, qui identifie et définit les éléments principaux d'une chaîne de distribution de contenus multimédia et les rapports entre ces éléments. MPEG-21 définit également l'ensemble des interfaces entre ces éléments.

La notion de base de MPEG-21 est le « Digital Item » (DI). Un Digital Item est une représentation numérique d' « un travail » et en tant que tel, c'est la chose sur laquelle on agit: contrôle, décrit, échange, adapte, etc. Un DI contient à la fois des ressources multimédia et les descriptions des multiples facettes de ces ressources.

La figure 1 présente l'infrastructure de MPEG-21 pour une chaîne de distribution de contenu multimédia. Le contenu original est fourni sous la forme d'un DI. Différentes entités MPEG-21 manipulent ce DI, et le DI adapté est finalement reçu par l'utilisateur ayant un terminal conforme à la norme.

Ci-dessous une liste non exhaustive des éléments principaux de MPEG-21:

➢ DID (Digital Item Declaration): un langage XML de description de DI [?].

➢ DIA (Digital Item Adaptation): Un langage XML de description de contraintes et des adaptations possibles [?]. Dans DIA, les descripteurs MPEG-7 peuvent être largement utilisés. MPEG-7 est une norme définie par MPEG pour la description des contextes et des contenus [?].

➢ REL (Right Expression Language) : un langage XML d'expression des droits associés à un contenu [?].

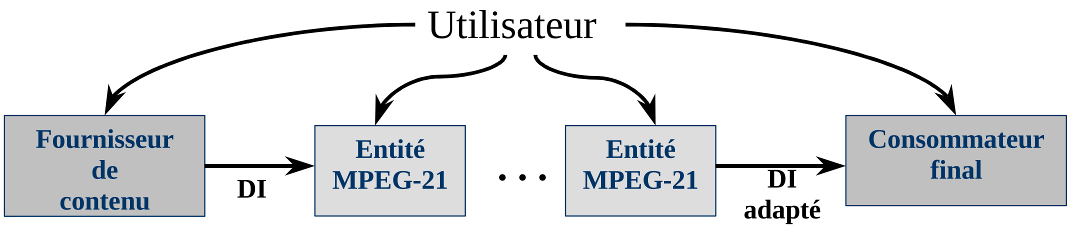


Figure 1. La chaîne de distribution de contenus multimédia de MPEG-21

➢ RDD (Right Data Dictionary) : un langage XML pour la définition des droits, permettant en particulier de créer de nouveaux droits et d'étendre ainsi le domaine géré par REL [?].

➢ IPMP (Intellectual Property Management and Protection) : ensemble de technologies permettant au contenu d'être contrôlé et protégé à travers une grande variété de réseaux et de terminaux.

Un Digital Item est donc représenté par un document XML nommé DID, dans lequel il y a entre autres:

- Des descripteurs de ressources, comme une vidéo ou un flux sonore.
- Des descripteurs DIA rendant possible des adaptations de ressources.
- Des descripteurs REL et RDD de droits d'accès pour le contenu complet et/ou pour chaque ressource.

Les caractéristiques du terminal sur lequel ce Digital Item doit être présenté, du réseau utilisé et de l'utilisateur, avec ses droits et préférences, peuvent être représentés dans un ou plusieurs fragments de DID. L'ensemble du DI et des contraintes est géré par un environnement qui doit obéir aux règles de protection IPMP.

L'infrastructure d'adaptation de MPEG-21 repose sur plusieurs éléments, dont les contraintes et les caractéristiques de l'environnement dans lequel le contenu multimédia est utilisé. Les caractéristiques de l'environnement d'utilisation (en anglais UED: *Usage Environment Description*) contiennent les différentes contraintes telles que les capacités du terminal utilisateur, les caractéristiques du réseau, les préférences de l'utilisateur et de l'auteur ainsi que les contraintes imposées par n'importe quelle autre entité intermédiaire existant dans une chaîne de distribution de contenus multimédia. La décision optimale d'adaptation doit être prise en respectant toutes ces contraintes.

Dans MPEG-21, il existe aussi des outils de description tels que *AQoS* (*Adaptation Quality of Service*) et *BSD* (*Bitstream Syntax Description*) conçu pour faciliter ou guider la tâche de l'adaptation d'une ressource scalable par modification directe de bitstream dans le domaine binaire. Ces outils de description fournissent les paramètres et les méta-données nécessaires au processus d'adaptation d'une ressource multimédia.

Au moment où ce travail a commencé, MPEG-21 ne proposait pas d'outils de description pour guider d'une manière dynamique et intelligente les adaptations des contenus non *scalable*, également appelées conversions de ressource.

Première partie : adaptation mono-média

Les principaux éléments requis pour l'adaptation de contenus mono-média sont la description du média (également appelé ressource), la description du contexte et des techniques d'adaptation. La figure 2 schématise un tel système d'adaptation. Le moteur d'adaptation est basé sur des techniques d'adaptation qui consiste à séparer la phase de prise de décision de la forme optimale de l'adaptation à appliquer à la ressource de la phase d'adaptation proprement dite. La prise de décision est gérée par l'algorithme d'optimisation tandis que l'adaptation de ressource est effectuée par différents outils d'adaptation ou par substitution de ressource par ses variations existantes.

Nous catégorisons l'adaptation mono-média en trois catégories: la modification directe de flux binaire, la conversion de ressource et l'adaptation par substitution.

Modification directe de flux binaire. La modification directe de flux binaire est plutôt destinée aux médias *scalable*s et consiste en la manipulation (suppression) directe de parties du flux binaire, et ne change pas la structure haut niveau du flux.

La norme MPEG-21 met à disposition des moyens complets pour supporter un tel type d'adaptation. Les outils de description et d'adaptation AQoS et BSD aident et guident les processus de prise de décision et d'adaptation. Ces outils de description fournissent les paramètres et les méta-données nécessaires au processus de transcodage d'une ressource multimédia. Le principe de ces outils repose sur la construction d'une description XML de la structure du flux original. Ensuite, selon les contraintes du contexte, cette description XML est transformée. Au final, le flux original est manipulé selon cette description transformée afin d'obtenir le flux adapté.

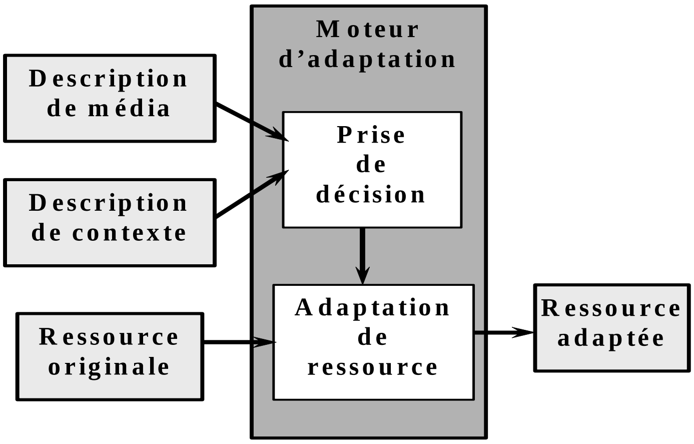


Figure 2 Le schéma d'un système d'adaptation de ressource

Conversion de ressource. La conversion de ressource est une adaptation qui consiste en un changement de la structure du flux binaire. Par exemple, le décodage partiel, la manipulation puis l'encodage d'un media est considéré comme une conversion de ressource. Cette thèse propose des moyens efficaces et nouveaux basé sur la norme MPEG-21, pour le support de ce type d'adaptation dans un système MPEG-21.

Au commencement de cette étude, MPEG-21 ne proposait pas d’outil de description pour guider les conversions de ressources. La première partie du travail de cette thèse a été une étude de ces descriptions qui a abouti à plusieurs contributions à la norme MPEG-21.

Substitution de ressource. La troisième catégorie est l’adaptation hors-ligne, à savoir la substitution de ressource par des alternatives déjà existantes, selon les contraintes du contexte.

Notre approche : conversion de ressource dans MPEG-21

Selon les divers changements que subit le média (changement de modalité, de format, etc.), nous définissons trois types de conversion de ressource : transmodage, transcodage, et transformation.

Transmodage

Pour définir le transmodage, nous devons dans un premier temps adopter une définition exacte de «la modalité » d’une ressource multimédia ainsi qu’une liste de modalités.

La modalité a au moins deux significations. Au niveau perceptuel, des modalités sont attachées aux cinq sens humains, ainsi il y a seulement « une » modalité visuelle. Au niveau structurel, il peut y avoir plusieurs modalités dans une même modalité perceptuelle. Par exemple la modalité perceptuelle « visuelle » peut inclure vidéo « bitmap » (c'est-à-dire décrite point par point), image « bitmap » et deux sortes d’images vectorielles (graphiques 2D et graphiques 3D).

Dans MPEG-21, le niveau de modalité significatif du point de vue de l'adaptation est le niveau structurel, puisque la structure impose les types d'algorithmes qui peuvent être appliqués aux ressources.

Nous proposons une approche hiérarchique pour la classification de modalité pour le transmodage. Dans cette classification, nous avons cinq modalités principales et quelques modalités secondaires pour certaines d'entre elles :

- Vidéo
- Audio (Audio2D, Audio3D, Parole)
- Image
- Graphiques (Graphiques2D,Graphiques3D)
- Texte

Depuis peu, MPEG-21 utilise une version étendue du ContentCS de MPEG-7 pour la classification des modalités. Cette classification est commune à notre approche mais contient aussi des modalités multiples comme la modalité « audio-visuelle ». Notre classification des modalités est un sous-ensemble de cette classification qui ne contient que les mono-modalités.

Note — Dans ce travail, la modalité graphiques se rapporte à des graphiques vectoriels et la modalité image se rapporte à des images bitmap. Nous distinguons les modalités graphiques 2D et 3D car les modèles utilisés dans ces deux types de descriptions et les algorithmes de rendu sont fondamentalement différents.

Le concept de transmodage se rapporte à la conversion d'une ressource multimédia en changeant sa modalité initiale en une autre modalité. Par conséquent, nous définissons le transmodage comme un type d'adaptation de ressource qui change la modalité de la ressource originale.

Dans ce travail, le transmodage fait référence à une adaptation à la volée ou à la demande, et non pas à la substitution d'une ressource multimédia par des versions alternatives pré-existantes sous d'autres modalités.

Quelques scénarios d'utilisation

Le transmodage est nécessaire dans divers scénarios d'adaptation de contenus multimédia. Voici un exemple de transmodage vidéo-vers-image et vidéo-vers-graphiques2D : l'auteur d'une ressource vidéo n'a pas fourni les versions image ou séquence d'images, mais a donné ses intentions de transmodage vers image ou séquence d'images (c'est-à-dire la modalité graphiques2D) par le moyen de descripteurs de transmodage. Pour la conversion vidéo-vers-graphiques2D, l'auteur a déterminé les trames vidéo qui sont à employer en tant qu'images, les transitions entre ces images et les durées de présentation pour chaque image.

Un exemple concret de ce scénario est le suivant: la ressource vidéo originale est une bande annonce d'un film, conçue pour la transmission sur ADSL mais inappropriée à la diffusion sur réseaux cellulaires. Dans l'expression correspondante de REL, le transmodage vidéo-vers-graphiques2D (séquence d'images) est autorisé. La décision optimale d'adaptation est la conversion de cette vidéo en une séquence d'images. Une fois cette décision prise, afin d'effectuer le transmodage, les données nécessaires sont extraites du descripteur DIA. Par exemple, la résolution d'affichage et le format de l'image sont récupérés du descripteur *TerminalCapabilities* [?]. Les contraintes sur la modalité et les priorités des modalités sont récupérées dans le descripteur *ModalityConversionPreference* ou *PresentationPriorityPreferences* [?]. La bande passante disponible, nécessaire pour déterminer le niveau de compression de la séquence d'image, est récupérée dans le descripteur *NetworkCharacteristics* [?] Finalement, la liste des trames à utiliser, leurs importances (à prendre en compte dans les cas de limitations de bande passante) et les durées relatives sont récupérées dans le descripteur *transmoding*.

Voici également un scénario de cas d'utilisation pour le transmodage texte-vers-image : la ressource à adapter est une ressource texte, par exemple, un message en langue Persane (ou avec une police spéciale). Le terminal ne dispose pas de la police persane, mais peut présenter des images. Dans ce cas, et si autorisé dans l'expression de REL, une solution est de transmoder le texte en une image. Le processus de transmodage exige la connaissance des caractéristiques de la police (afin d’afficher le texte dans une image), l'information sur les couleurs, etc. Le même cas d'utilisation peut être considéré pour un transmodage de type texte-vers-vidéo, quand le texte ne rentre pas dans l'écran du dispositif (dont la taille est extraite à partir du descripteur *TerminalCapabilities*), et a ainsi besoin de défiler sur l’écran. Ce processus de transmodage exige en plus la connaissance des préférences d'auteur sur le type de défilement du texte : défilement horizontal sur une ligne, vertical sur plusieurs lignes, présentation par paragraphes successifs, etc.

Outil de description de transmodage

Nous avons développé notre outil de description des paramètres de transmodage comme un langage XML étendant le langage DIA. Cet outil de description permet de décrire les informations de transmodage d’une manière statique. Nous nous sommes plus particulièrement intéressés au point de vue du fournisseur, mais toute autre entité dans la chaîne d'adaptation peut exprimer ces informations sur les paramètres d'un transmodage spécifique par le biais de cet outil. Ces informations comprennent:

- Les préférences de transmodage concernant la qualité et la priorité. Le créateur de contenu peut exprimer un niveau de qualité pour un transmodage spécifique d’une ressource précise. Un niveau de priorité peut être attribué à chaque transmodage pour guider l’algorithme d’optimisation.

- Les paramètres de transmodage. Le fournisseur, à partir de sa connaissance des ressources, peut fournir quelques conseils ou recommandations de transmodage pour guider l'adaptation. L’outil permet la description des paramètres spécifiques au transmodage les plus généraux, c’est-à-dire que les paramètres ne sont liés à aucun algorithme particulier. Pour cela, nous avons considéré un ensemble de transmodages qui ont des paramètres généraux et spécifiques au transmodage.

Les syntaxes et sémantiques de l’outil de description de transmodage sont détaillées dans l’Annexe A.

Moteur de transmodage

Afin de mettre en application l'outil de description de transmodage, dans une première phase nous avons développé un module de gestion des descripteurs de transmodage, et l’avons intégré dans le logiciel de référence de MPEG-21. Dans une deuxième phase, un ensemble de convertisseurs de modalité a été intégré dans ce logiciel. L’ensemble du logiciel a été proposé à MPEG-21 comme résultat d'un « *Core*

Experiment » sur le transmodage. Cette implémentation est intégrée dans le projet européen ISIS [?]. Notre moteur de transmodage est présenté dans la figure 3 et se compose de deux blocs principaux : le Décideur et le Transmodeur de ressources. Les entrées du module Transmodage sont :

- Le choix de contenu de l'utilisateur, transmis par le *Streaming Server*, qui reçoit la requête initiale de l'utilisateur (étape 1).
- La description du contenu venant d'une base de données de descriptions de contenus (étape 2).
- La description des contraintes de l'utilisateur et des caractéristiques de l'environnement d'utilisation (réseau, terminal, etc.) venant d'une base de données provisoire (étape 3).
- Le contenu à adapter, venant d'une base de données de ressources (étape 4).
- La ressource adaptée est envoyée au *Streaming Server* (étape 5).

Transcodage et transformation

Nous définissions le transcodage en tant qu'une conversion de ressource qui change le format de codage de la ressource sans changer sa modalité, tandis que la transformation est définit en tant qu'une conversion de ressource changeant des paramètres de ressource autre que le format et la modalité. A titre d'exemple, ces paramètres peuvent inclure des paramètres d'encodage ou de mise en forme.

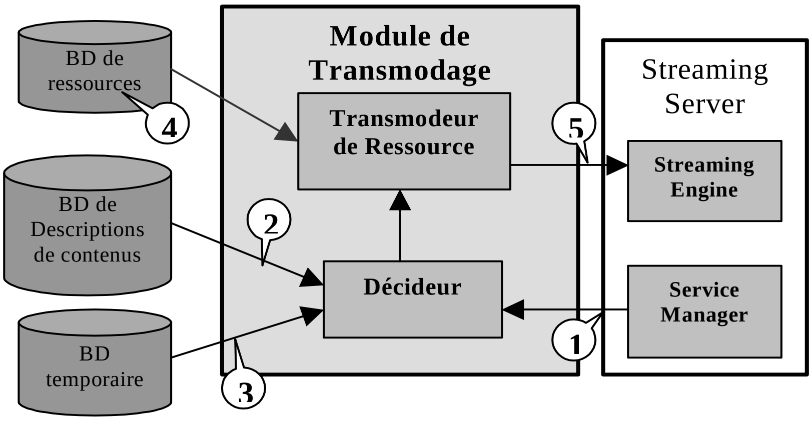


Figure 3. L'architecture du module de transmodage dans ISIS

Note — Les trois types de conversions définis ci-dessus (transmodage, transcodage et transformation) sont des conversions atomiques. A titre d'exemple un transmodage video-vers-image n'est pas sensé changer la résolution de l'image adaptée, car le changement de résolution est en fait une conversion de type transformation. Obtenir une image plus petite en résolution, à partir d'une vidéo, est donc une conversion composite. Des conversions composites peuvent être obtenues par différentes compositions des conversions atomiques.

Dans la deuxième phase de notre travail sur la conversion de ressource, nous avons étendu notre outil de description de transmodage à un outil de description de conversion générale. Pour cela, en plus des paramètres de transmodage, nous avons étudié les paramètres génériques et spécifiques des conversions du type transcodage et transformation. Les syntaxes et sémantiques du langage RCD (*Resource Conversion Declaration*) sont détaillées dans l'Annexe B.

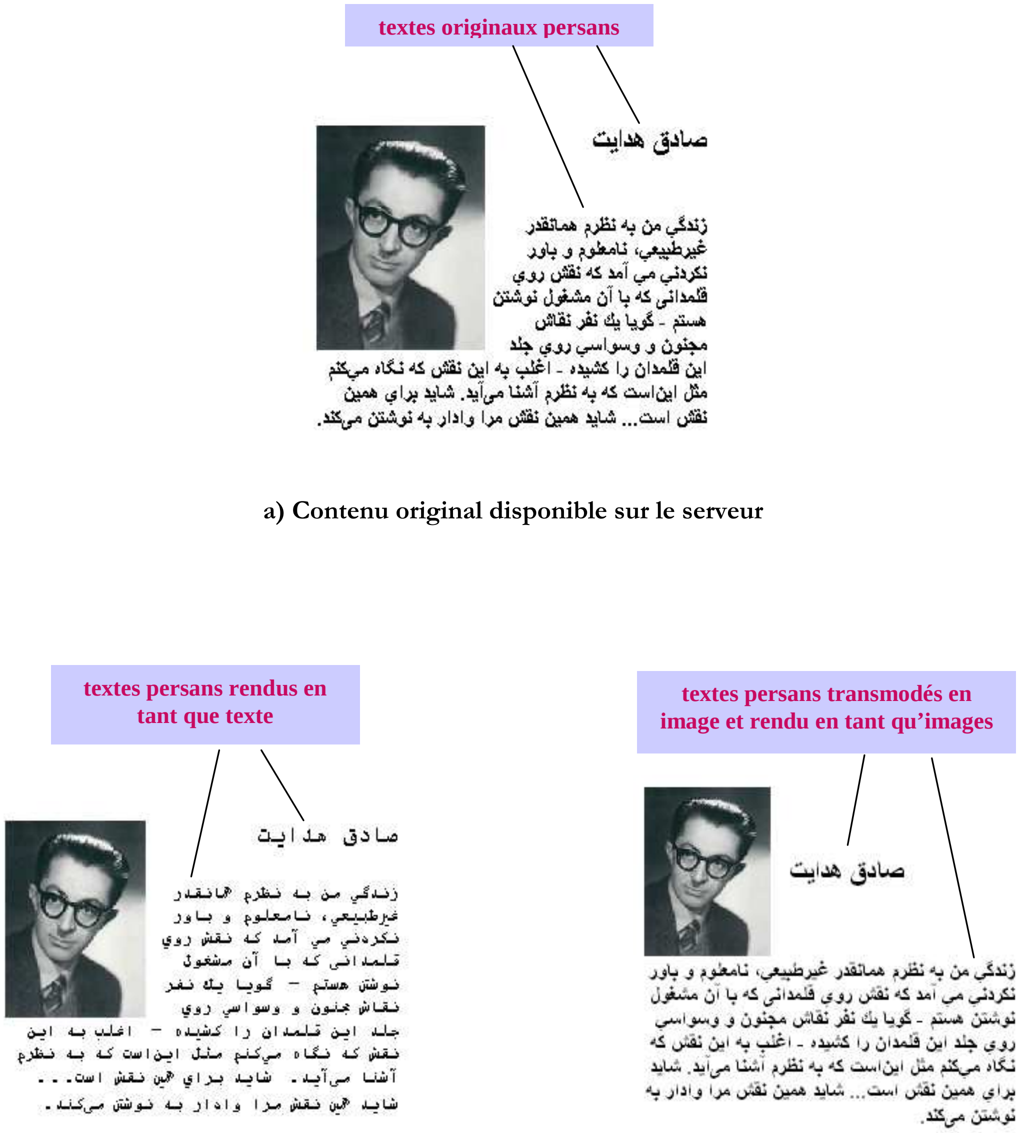
Afin de mettre en application cet outil de description de conversion ou RCD, nous avons ensuite développé un moteur de conversion de ressource, basé sur l'architecture du moteur de transmodage. Ce moteur contient un ensemble de transmodages, transcodages et transformations de média. Figure 4 démontre deux exemples de conversion de média d'un même contenu original pour différents contextes.

Deuxième partie : adaptation sémantique de scènes multimédia

Considérons une scène multimédia en tant qu'une présentation intégrant plusieurs objets médias discrets ou continus, temporellement synchronisés et spatialement organisés entre eux. L'adaptation d'une scène multimédia implique l'adaptation des objets médias présents dans cette scène ainsi que l'adaptation de la présentation elle-même. Dans ce cadre, définissons l'adaptation sémantique de scènes multimédia comme l'adaptation de scènes multimédia prenant en compte également les contraintes sémantiques de la structure de la présentation.

Les éléments requis pour l'adaptation sémantique de scènes multimédia sont la description de la scène, la description du contexte, les informations sur le contenu, et des techniques d'adaptation sémantique. Les informations sur le contenu contiennent la description des caractéristiques physiques (format, modalité, etc.) ainsi que la description des informations sémantiques (importance, etc.). Les techniques d'adaptation sémantique se décomposent en des techniques d'adaptation de la structure de scène et des techniques d'adaptation de ressource.

Plusieurs travaux de recherche ont été effectués dans le domaine de l'adaptation de scène multimédia. Néanmoins, il n'existe pas de cadre complet qui propose des techniques d'adaptation sémantique de présentation.


textes originaux persans
صادق هدايت
زندگي من به نظرم همانقدر
غيرطبيعي، نامعلوم و باور
نكردني مي آمد كه نقش روي
قلمداني كه با آن مشغول نوشتن
هستم - گويا يك نفر نقاش
مجنون و وسواسي روي جلد
اين قلمدان را كشيده - اغلب به اين نقش كه نگاه مي‌كنم
مثل اين‌است كه به نظرم آشنا مي‌آيد. شايد براي همين
نقش است... شايد همين نقش مرا وادار به نوشتن مي‌كند.
textes persans rendus en tant que texte
textes persans transmodés en image et rendu en tant qu'images

a) Contenu original disponible sur le serveur

b) Contenu transmis pour un terminal avec caractères persans. L'image est redimensionnée et la police de texte est

c) Contenu transmis pour un terminal sans caractères persans. Les images

Figure 4. Deux exemples de conversion de ressources

Nous avons donc besoin d'un cadre de travail plus générique qui propose des solutions pour maintenir la cohérence de scène et qui permettent de conserver son sens. Notre approche essaie de trouver des réponses à ces besoins.

Outil de description SID

Nous avons défini un langage XML pour la déclaration des informations structurellement sémantiques de scène (c'est à dire du point de vue de la structure de la présentation). Ce langage est appelé SID (*Semantic*

Information Declaration). Les instances du langage sont contenues dans un CDI (*Content Digital Item*) qui contient également les références de la scène et des ressources, la description des ressources, ainsi que la description des conversions (exprimée par des outils de description de conversion). Le langage SID permet de décrire trois types d'information sémantique:

- Information sémantique propre à chaque objet media présent dans la scène, dans le contexte de la présentation. Cette information est décrite par l'intermédiaire des attributs XML `importance`, `role`, `maxRRF` (*maximum Resolution Reduction Factor*) :

  - L'attribut `importance` représente l'importance de cet objet média dans le contexte de la présentation. Les valeurs possibles sont `low`, `medium`, et `high`.

  - L'attribut `role` représente le rôle de l'objet média concerné dans le contexte de la présentation. Par exemple si l'objet média a un rôle clé -c'est à dire qu'il n'est pas possible de l'éliminer de la scène sous aucune condition, mais qu'il peut être adapté, remplacé ou converti, la valeur de l'attribut `role` sera `key-convertible`. Les valeurs possibles pour cet attribut sont : `key`, `key-convertible`, `redundant` et `decorative`.

  - L'attribut `maxRRF` permet de décrire le seuil maximal autorisé pour la réduction de la taille (résolution) spatiale de l'objet.

- Dépendances sémantiques entre objets médias de la scène. Nous avons considéré trois types de dépendance :

  - Dépendances absolues des objets médias. Par exemple, le fait que la présence d'un média soit une condition préalable pour la présence d'autres médias, peut être déclaré par l'élément XML `PreConditionMedia`, et le fait qu'un texte soit une légende d'un autre média peut être exprimé par l'élément XML `CaptionMedia`.

  - Dépendances spatiales des objets médias. A titre d'exemple, le fait que deux objets médias doivent être gardés côte à côte si un réarrangement spatial de la scène est nécessaire, peut être exprimé par l'élément XML `KeepCloseTo`.

  - Dépendance de synchronisation entre objets médias. Par exemple, le fait que deux objets médias doivent être synchronisés dans la scène adaptée, quand l'adaptation exige un réarrangement temporel de la scène, peut être exprimé par l'élément XML `KeepCloseTo`.

- Préférences et priorités sémantiques sur les éventuels découpages (fragmentations) spatiaux et temporels de la scène.

Les syntaxes et sémantiques du langage SID sont détaillées dans l'Annexe C.

Règles d'interprétation

Afin de pouvoir développer un algorithme d'optimisation, nous avons défini un ensemble de règles d'interprétation pour les informations incluses dans les descripteurs SID. Ces règles d'interprétation sont décrites plus en détails dans le manuscrit.

Moteur d'adaptation sémantique de scène (MSSA : *Multimedia Scene Semantic Adaptation*)

Nous avons développé un moteur d'adaptation sémantique de scène pour les documents multimédia SMIL. Le schéma du moteur est démontré dans la figure 5. Les CDI et XDI (*Contexte Digital Item*) contiennent les références de la scène et de ses médias, les contraintes du contexte ainsi que les informations sur le contenu.

Le moteur se décompose en deux parties principales: l'optimisation de scène et l'adaptation des médias. L'optimisation est effectuée en deux étapes: l'optimisation de la présentation et l'adaptation des objets médias. Dans un premier temps la présentation est optimisée, et ensuite selon la forme optimale trouvée, la structure de la présentation est adaptée.

L'optimisation de la présentation consiste en la prise de décision sur la forme optimale de la scène et des calculs de réarrangement de la structure. L'adaptation des médias est effectuée par plusieurs outils d'adaptation et de conversion de ressource.

Les différents modules de l'optimiseur de scène sont illustrés dans la Figure 6. Leur fonctionnement et l'algorithme d'optimisation sont détaillés dans le manuscrit.

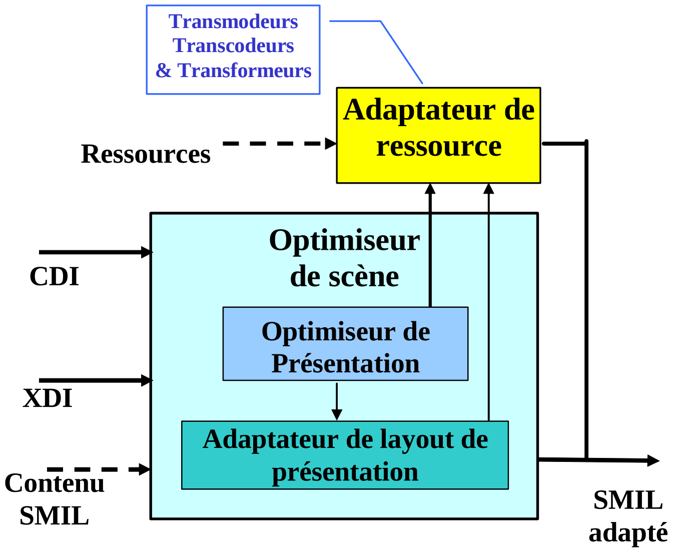


Figure 5. L'architecture du moteur MSSA

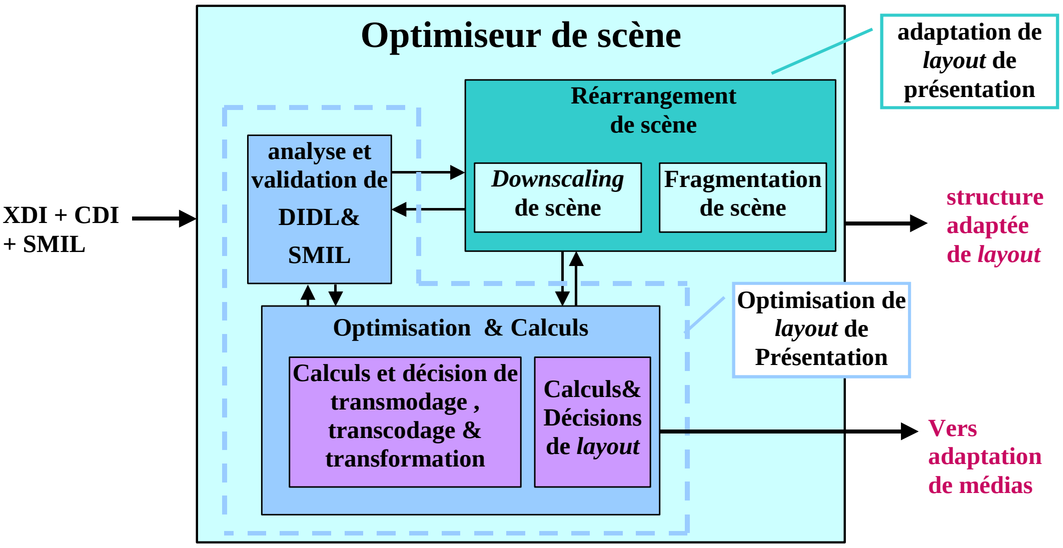

Optimiseur de scène
adaptation de *layout* de présentation
Réarrangement de scène
Downscaling de scène
Fragmentation de scène
analyse et validation de DIDL& SMIL
XDI + CDI + SMIL
structure adaptée de *layout*
Optimisation & Calculs
Calculs et décision de transmodage , transcodage & transformation
Calculs& Décisions de *layout*
Optimisation de *layout* de Présentation
Vers adaptation de médias

Figure 6. L'architecture détaillée du module optimiseur de scène

Exemples d'adaptation sémantique de scène

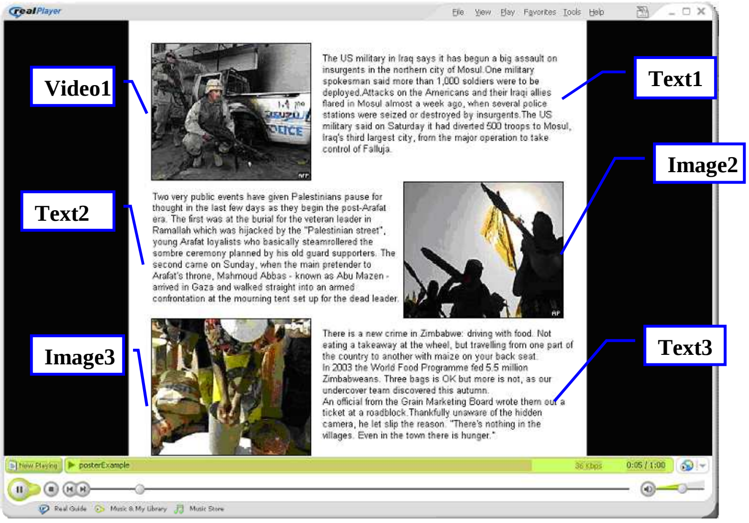
Cette section décrit quelques exemples d'adaptation sémantique de scènes SMIL à différents contextes. Les instances de CDI (comprenant les informations SID) et XDI sont données dans le manuscrit.

Exemple A.

Cet exemple illustre l'adaptation de la scène initiale de la figure 7 dans deux cas différents. Dans la scène de la figure 7, le texte Text1 explique la vidéo Video1, tandis que les images Image2 et Image3 illustrent respectivement les textes Text2 et Text3. Les couplets d'objets (Text1, Video1), (Texte2, Image 2) et (Text3, Image3) sont donc des groupes de média sémantiquement reliés.

Cas A-1.

Récapitulatif de XDI et CDI: La taille d'écran est supposée être de 200 x 300 et les modalités acceptées sont seulement texte et image. L'aspect visuel de la présentation est privilégié en spécifiant une importance basse pour tous les textes (`importance` = `low`). La taille spatiale des textes Text1 et Text3 peut au maximum être réduite par l'utilisation de versions courtes (`maxRRF` = 3) tandis que le texte Text2 ne peut pas être résumé (`maxRRF` = 1). Toutes les images ainsi que la vidéo sont d'une importance moyenne (`importance` = `medium`), et leur taille spatiale peut être au maximum réduite d'un facteur de deux (`maxRRF` = 2). Les paramètres de conversion vidéo-vers-image sont donnés dans l'instance de CDI.

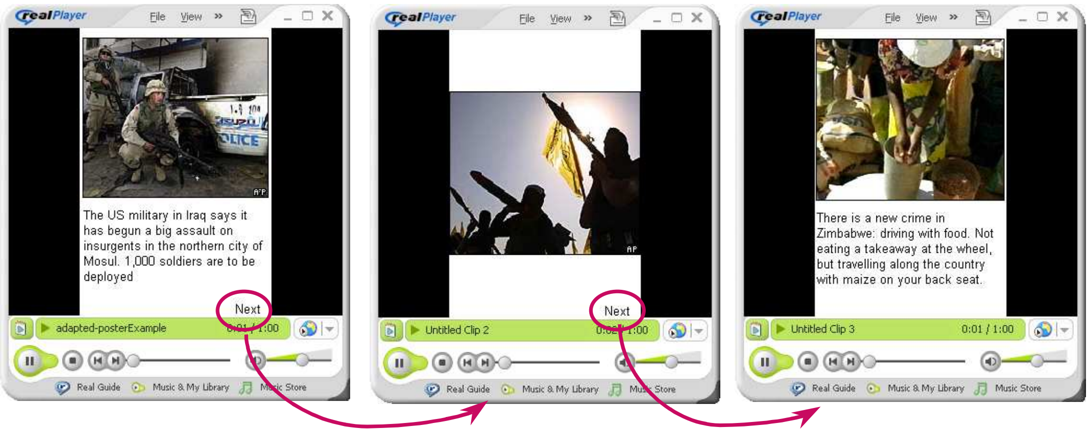


The US military in Iraq says it has begun a big assault on insurgents in the northern city of Mosul.One military spokesman said more than 1,000 soldiers were to be deployed.Attacks on the Americans and their Iraqi allies flared in Mosul almost a week ago, when several police stations were seized or destroyed by insurgents.The US military said on Saturday it had diverted 500 troops to Mosul, Iraq's third largest city, from the major operation to take control of Falluja.

Two very public events have given Palestinians pause for thought in the last few days as they begin the post-Arafat era. The first was at the burial for the veteran leader in Ramallah which was hijacked by the "Palestinian street", young Arafat loyalists who basically steamrollered the sombre ceremony planned by his old guard supporters. The second came on Sunday, when the main pretender to Arafat's throne, Mahmoud Abbas - known as Abu Mazen - arrived in Gaza and walked straight into an armed confrontation at the mourning tent set up for the dead leader.

There is a new crime in Zimbabwe: driving with food. Not eating a takeaway at the wheel, but travelling from one part of the country to another with maize on your back seat.
In 2003 the World Food Programme fed 5.5 million Zimbabweans. Three bags is OK but more is not, as our undercover team discovered this autumn.
An official from the Grain Marketing Board wrote them out a ticket at a roadblock.Thankfully unaware of the hidden camera, he let slip the reason. "There's nothing in the villages. Even in the town there is hunger."

Figure 7. Une scène multimédia SMIL



The US military in Iraq says it has begun a big assault on insurgents in the northern city of Mosul. 1,000 soldiers are to be deployed

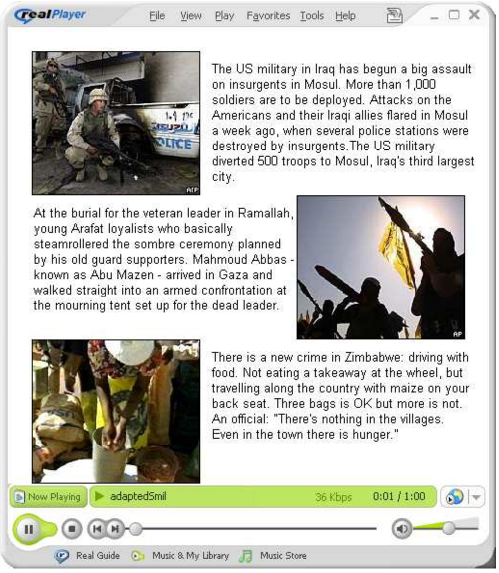
There is a new crime in Zimbabwe: driving with food. Not eating a takeaway at the wheel, but travelling along the country with maize on your back seat.

Figure 8. La scène adaptée pour l'exemple A-1

La solution d'adaptation trouvée et effectuée par moteur MSSA est une fragmentation de scène. Les versions les plus courtes des textes Text1 et Text3, ont remplacé les textes originaux. Le texte Text2 est supprimé et un transmodage vidéo-vers-image a été effectué sur la vidéo. Les images ont été redimensionnées. La scène adaptée est représentée figure 8.

Cas A-2.

Le récapitulatif du XDI et CDI est le même que dans le cas A-1 sauf que toutes les modalités sont jouables sur le terminal et plus particulièrement la vidéo. La taille de l'écran est plus importante (500 x 475) et le texte Text2 peut être résumé (maxRRF = 2).



The US military in Iraq has begun a big assault on insurgents in Mosul. More than 1,000 soldiers are to be deployed. Attacks on the Americans and their Iraqi allies flared in Mosul a week ago, when several police stations were destroyed by insurgents.The US military diverted 500 troops to Mosul, Iraq's third largest city.

At the burial for the veteran leader in Ramallah, young Arafat loyalists who basically steamrollered the sombre ceremony planned by his old guard supporters. Mahmoud Abbas - known as Abu Mazen - arrived in Gaza and walked straight into an armed confrontation at the mourning tent set up for the dead leader.

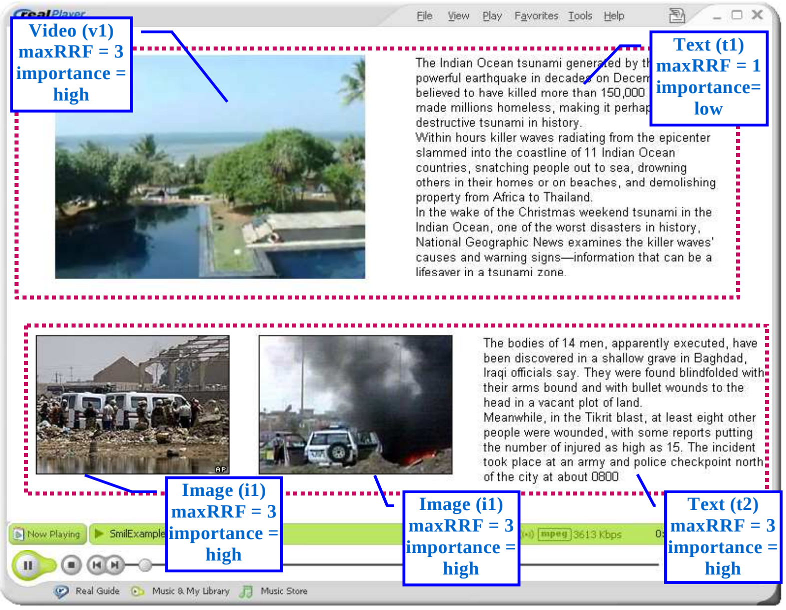
There is a new crime in Zimbabwe: driving with food. Not eating a takeaway at the wheel, but travelling along the country with maize on your back seat. Three bags is OK but more is not. An official: "There's nothing in the villages. Even in the town there is hunger."

Figure 9. La scène adaptée pour l'exemple A-2

La solution d'adaptation trouvée et effectuée par moteur MSSA est un redimensionnement de scène. Les images et la vidéo ont également été retaillées. Les versions courtes optimales des textes ont remplacé les textes originaux. La Figure 9 montre la scène adaptée.

Exemple B.

La scène initiale, illustrée dans la figure 10, contient une vidéo, deux images et deux textes. La vidéo est sémantiquement liée au premier texte, tandis que le deuxième texte explique les deux images.

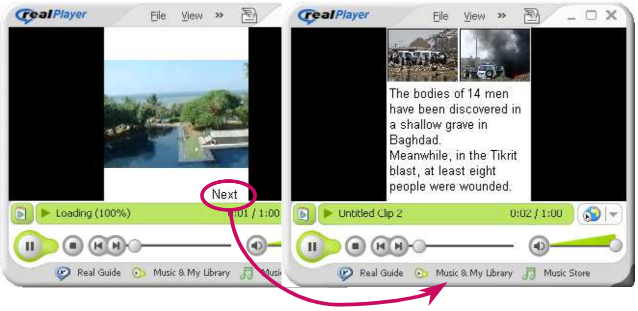


realPlayer
File View Play Favorites Tools Help
Video (v1) maxRRF = 3 importance = high
Text (t1) maxRRF = 1 importance= low
Image (i1) maxRRF = 3 importance = high
Image (i1) maxRRF = 3 importance = high
Text (t2) maxRRF = 3 importance = high
Now Playing
SmilExample
mpeg 3613 Kbps
Real Guide
Music & My Library
Music Store

The Indian Ocean tsunami generated by th powerful earthquake in decades on Decem believed to have killed more than 150,000 made millions homeless, making it perhap destructive tsunami in history.
Within hours killer waves radiating from the epicenter slammed into the coastline of 11 Indian Ocean countries, snatching people out to sea, drowning others in their homes or on beaches, and demolishing property from Africa to Thailand.
In the wake of the Christmas weekend tsunami in the Indian Ocean, one of the worst disasters in history, National Geographic News examines the killer waves' causes and warning signs—information that can be a lifesaver in a tsunami zone.

The bodies of 14 men, apparently executed, have been discovered in a shallow grave in Baghdad, Iraqi officials say. They were found blindfolded with their arms bound and with bullet wounds to the head in a vacant plot of land.
Meanwhile, in the Tikrit blast, at least eight other people were wounded, with some reports putting the number of injured as high as 15. The incident took place at an army and police checkpoint north of the city at about 0800

Figure 10. La scène initiale de l'exemple B



The bodies of 14 men have been discovered in a shallow grave in Baghdad.
Meanwhile, in the Tikrit blast, at least eight people were wounded.

Figure 11. La scène adaptée de l'exemple B

Récapitulatif de XDI et CDI : La taille de l'écran du terminal est extrêmement réduite (150 x 180). Toutes les modalités sont supportées et pour chaque texte, deux versions courtes sont disponibles. Les autres informations sont données sur la figure 10.

La solution d'adaptation trouvée et effectuée par moteur MSSA est la fragmentation de la scène en deux morceaux. Le texte t1 est supprimé. Les deux images et la vidéo ont été redimensionnées. La version courte optimale de t2 l'a remplacé. Un réarrangement de layout a eu également lieu sur le deuxième fragment. La figure 11 montre la scène adaptée.

Conclusions

Adaptation de contenu mono-média et conversion de ressources :

Nous avons développé un outil de description dans le cadre de la norme MPEG-21 DIA, qui permet de décrire d'une manière statique les informations de conversion. Nous avons ensuite développé un moteur de conversion guidée qui permet l'adaptation des ressources médias par conversion, selon les contraintes de contexte, les préférences d'utilisateur ainsi que les recommandations de l'auteur sur les priorités, les qualités et les paramètres de conversion. Ces recommandations sont déclarées par notre outil de description.

Notre travail sur la conversion de ressource dans MPEG-21 DIA a proposé des solutions innovantes pour le support de l'adaptation guidée par l'auteur. Ce travail a également prouvé que les recommandations de l'auteur sur les paramètres, priorités et qualités des conversions, guident l'adaptation et améliorent son résultat.

Adaptation sémantique de scène :

Nous avons défini un langage pour la description de méta-données structurellement sémantique des présentations multimédia. En utilisant ce langage, nous avons ensuite développé un algorithme d'optimisation de scène et un moteur d'adaptation sémantique de documents SMIL.

Notre travail sur l'adaptation structurellement sémantique de documents multimédia a montré que l'information sémantique sur la structure de document joue un rôle essentiel dans l'adaptation de scène. Obtenir une scène adaptée cohérente et logique nécessite la disposition de cette information. Il est donc très important de trouver des solutions qui permettent de décrire d'une manière générale cette information pour tous les documents multimédia qui sont à adapter automatiquement.

Perspectives

Adaptation de contenu mono-média et conversion de ressources :

L'étude plus approfondie des solutions de conversion guidée est une des premières perspectives de notre travail dans le domaine d'adaptation mono-media. Un autre domaine pour de futur travail est l'étude d'un outil de description qui permettrait la description des valeurs contraintes des paramètres d'adaptation d'une manière dynamique. Les valeurs contraintes des paramètres de conversion sont soit les valeurs exactes des contraintes de contexte (décrits en UED) ou bien, peuvent être calculées à partir de ces valeurs. Un tel outil de description permettrait également la sélection plus intelligente des valeurs des paramètres, parmi un ensemble de valeurs données pour chaque paramètre, et ceci en utilisant les outils AdaptationQoS et UCD de MPEG-21 DIA. Actuellement une activité sur un tel type d'outil de description de conversion est en cours au sein du group MPEG-21 DIA, dont l'ENST est le coordinateur. Cet outil de description est appelé ConversionLink.

Adaptation sémantique de scène :

Une suite immédiate de notre travail dans le cadre de MSSA est la prise en compte de critère de bande passante dans l'adaptation sémantique. Ceci concerne en effet l'étude des solutions innovantes d'un algorithme d'optimisation, d'allocation et de partage de bande passante entre les objets médias d'une scène multimédia, en fonction des informations sémantiques de la scène.

L'étude des modèles et des langages de description de scène qui permettront de décrire une scène sans donner la définition absolue de layout, serait également une suite intéressante pour ce travail. Un tel modèle de description de scène doit permettre de décrire un layout en fonction des contraintes sémantiques, temporelles et spatiales.

L'étude des solutions de génération automatique des informations sémantiques, trouvées en fonction des historiques d'utilisation des documents multimédia par de différents utilisateurs, est un autre axe pour le futur travail dans ce domaine.

ANNEX A: SYNTAX AND SEMANTICS OF TRANSMODING DESCRIPTION TOOL

Transmoding Syntax

The following proposed syntax has been validated against the DIA schema.

```
<!--  ################################################     -->
<!--   Definition of Transmoding                            -->
<!--  ################################################     -->
<complexType name="TransmodingType">
 <complexContent>
  <extension base="dia:DIABaseType">
   <sequence>
    <element name="TransmodingParameters" type="dia:TransmodingParametersType"
                                                                         maxOccurs="unbounded"/>
   </sequence>
   <attribute name="quality" type="mpeg7:zeroToOneType" use="optional" />
   <attribute name="priority" type="positiveInteger" use="optional" />
  </extension>
 </complexContent>
</complexType>

<!--  ################################################   -->
<!--   Definition of TransmodingParametersType           -->
<!--  ################################################   -->
<complexType name="TransmodingParametersType" abstract="true">
 <complexContent>
  <extension base="dia:DIABaseType">
   <sequence>
    <element name="To" type="mpeg7:ControlledTermUseType"/>
   </sequence>
   <attribute name="method" type="anyURI" use="optional"/>
  </extension>
 </complexContent>
</complexType>

<!--  ################################################     -->
<!--   Definition of VideoSummarizationParametersType       -->
<!--  ################################################     -->
<complexType name="VideoSummarizationParametersType">
 <complexContent>
  <extension base="dia:TransmodingParametersType">
```

```
   <sequence>
    <element name="Slide" maxOccurs="unbounded">
        <complexType>
            <complexContent>
                <extension base="mpeg7:MediaTimeType">
                    <attribute name="importance" type="string" />
                </extension>
            </complexContent>
        </complexType>
    </element>
   </sequence>
  </extension>
 </complexContent>
</complexType>

<!--  ################################################      -->
<!--   Definition of TextVisualizationParametersType        -->
<!--  ################################################      -->
<complexType name="TextVisualizationParametersType">
 <complexContent>
  <extension base="dia:TransmodingParametersType">
   <sequence>
    <element name="FontParameters" type="dia:FontParametersType" minOccurs="0"/>
    <element name="Resolution" type="dia:ResolutionType" minOccurs="0"/>
    <element name="TextMotionStyle" type="mpeg7:ControlledTermUseType" minOccurs="0"/>
    <element name="TextColorInformation" type="dia:TextColorInformationType" minOccurs="0"/>
   </sequence>
   <attribute name="frameRate" type="float" use="optional" />
  </extension>
 </complexContent>
</complexType>

<complexType name="FontParametersType">
 <sequence >
  <element name="FontStyle" type="string" minOccurs="0"/>
  <element name="FontFamily" type="string" minOccurs="0" />
 </sequence>
 <attribute name="fontSize" type="positiveInteger" use="required"/>
</complexType>

<complexType name="TextColorInformationType">
 <attribute name="backgroundColor" type="dia:RGBColorType" />
 <attribute name="foregroundColor" type="dia:RGBColorType" />
</complexType>

<simpleType name="RGBColorType">
 <restriction base="dia:zeroToOneVectorType">
   <length value="3"/>
 </restriction>
</simpleType>

<simpleType name="zeroToOneVectorType">
  <list itemType="mpeg7:zeroToOneType"/>
</simpleType>

<!--  ################################################      -->
<!--   Definition of SpeechVisualizationParametersType      -->
<!--  ################################################      -->
<complexType name="SpeechVisualizationParametersType">
 <complexContent>
```

```
  <extension base="dia:TransmodingParametersType">
   <sequence>
    <element name="Language" type="string"/>
    <element name="FontParameters" type="dia:FontParametersType" minOccurs="0"/>
    <element name="Resolution" type="dia:ResolutionType" minOccurs="0"/>
    <element name="TextPresentationStyle" type="mpeg7:ControlledTermUseType" minOccurs="0"/>
    <element name="TextColorInformation" type="dia:TextColorInformationType" minOccurs="0"/>
   </sequence>
   <attribute name="frameRate" type="float" use="optional" />
  </extension>
 </complexContent>
</complexType>
```

Transmoding Semantics

Semantics of `TransmodingType`:

Name	*Definition*
`TransmodingType`	Tool for describing suggested transmoding adaptation information
`TransmodingParameters`	Describes the values of the parameters of a transmoding

Semantics of `TransmodingParametersType`:

Name	*Definition*
`TransmodingParametersType`	`TransmodingParametersType` extends `DIABaseType` and provides a base abstract type for various types of transmoding-related parameters.
`To`	Describes the final media modality
`method`	Describes a suggested transmoding method

Semantics of `VideoSummarizationParametersType`:

Name	*Definition*
`VideoSummarizationParametersType`	Tool for describing the transmoding parameters related to video summarization.
`Slide`	Describes the slide parameters.

Semantics of `TextVisualizationParametersType`:

Name	*Definition*
`TextVisualizationParametersType`	Tool for describing transmoding parameters related to text-to-video, text-to-image and text-to-graphics conversion.
`FontParameters`	Describes the parameters of the font of the text.
`Resolution`	Describes the resolution of the image.
`TextMotionStyle`	Describes the motion style of the text in case of text-to-graphics and text-to-video transmoding.
`TextColorInformation`	Describes the color information of the text: backGroundColor and forGroundColor.
`frameRate`	Describes the frame rate.

Semantics of `FontParametersType`

Name	*Definition*
`FontParametersType`	Tool for describing parameters of the text font.
`FontFamily`	Describes the font family.
`fontSize`	Describes the font size.
`FontStyle`	Describes the font style.

Semantics of `TextColorInformationType`:

Name	*Definition*
`TextColorInformationType`	Tool for describing color information of the text.
`backGroundColor`	Describes the color of the background.
`forGroundColor`	Describes the color of the text.

Semantics of `SpeechVisualizationParametersType`:

Name	*Definition*
`SpeechVisualizationParametersType`	Tool for describing transmoding parameters related to speech-to-text, speech-to-image, speech-to-video, and speech-to-graphics transmodings.
`Language`	Describes the final language of the text.
`FontParameters`	Describes the parameters of the font of the text.
`Resolution`	Describes the resolution of the image.
`TextMotionStyle`	Describes the motion style of the text in case of text-to-graphics and text-to-video transmoding.
`TextColorInformation`	Describes the color information of the text.
`frameRate`	Describes the frame rate.

ANNEX B: SYNTAX AND SEMANTICS OF CONVERSION DESCRIPTION TOOL

ConversionDescriptor Schema:

Syntax:

```
<schema targetNamespace="urn:mpeg:mpeg21:2003:01-DIA-NS" xmlns="http://www.w3.org/2001/XMLSchema"
        xmlns:dia="urn:mpeg:mpeg21:2003:01-DIA-NS" xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001"
        elementFormDefault="qualified" attributeFormDefault="unqualified">

    <import namespace="urn:mpeg:mpeg7:schema:2001" schemaLocation="mpeg7-udp-2003-dia.xsd"/>
    <include schemaLocation="DIA.xsd"/>
    <include schemaLocation="UsageEnvironment.xsd"/>

<!-- ################################################       -->
<!--  Definition of ConversionDescriptorType               -->
<!-- ################################################       -->
<complexType name="ConversionDescriptorType">
 <complexContent>
  <extension base="dia:DIADescriptionType">
   <sequence>
    <element name="Conversion" type="dia:ConversionInformationType" minOccurs="0"
                                                                          maxOccurs="unbounded"/>
   </sequence>
  </extension>
 </complexContent>
</complexType>

<!-- ################################################       -->
<!--  Definition of Converion Information                   -->
<!-- ################################################       -->
<complexType name="ConversionInformationType">
 <complexContent>
  <extension base="dia:DIADescriptionType">
   <sequence>
    <element name="ConverionInformation" type="dia:ConversionInformationBaseType" minOccurs="0"
                                                                          maxOccurs="unbounded"/>
   </sequence>
  </extension>
 </complexContent>
</complexType>
<complexType name="ConversionInformationBaseType" abstract="true">
```

```
 <complexContent>
  <extension base="dia:DIADescriptionType"/>
 </complexContent>
</complexType>

<!-- ################################################       -->
<!--  Definition of ConversionDescription                   -->
<!-- ################################################       -->
<complexType name="ConverionDescriptionType">
 <complexContent>
  <extension base="dia:ConversionInformationBaseType">
   <sequence>
    <element name="ConversionDescription" type="dia:ConversionDescriptionBaseType"
                                                            minOccurs="0" maxOccurs="unbounded"/>
   </sequence>
  </extension>
 </complexContent>
</complexType>

 <complexType name="ConversionDescriptionBaseType" abstract="true">
  <complexContent>
   <extension base="dia:DIADescriptionType">
    <sequence>
     <element name="ConversionUri" type="anyURI" minOccurs="0"/>
    </sequence>
   </extension>
  </complexContent>
 </complexType>
</schema>
```

Semantics:

Semantics of the `ConversionDescriptorType`

Name	*Definition*
`ConversionDescriptorType`	Tool for describing conversions
`Conversion`	Describes suggested conversions for a particular resource

Semantics of the `ConversionInformation`:

Name	*Definition*
`ConversionInformationType`	Tool for describing suggested conversions for a particular resource
`ConversionInformation`	Describes a conversion or a set of cascaded (ANDed) suggested conversions for a particular resource

Semantics of the `ConversionDescriptor`:

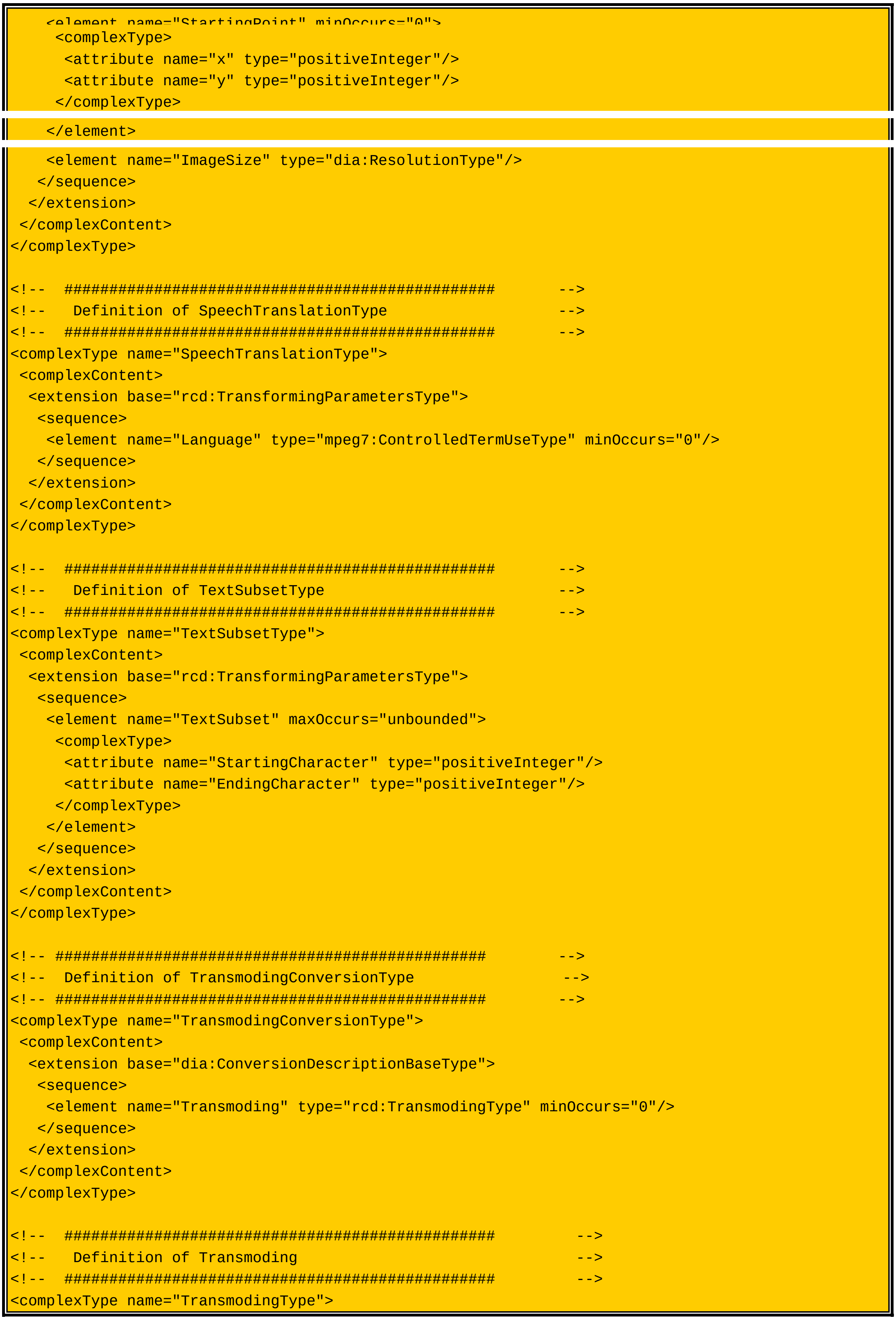
Name	*Definition*
`ConversionDescriptionType`	Tool for describing a suggested atomic conversion for a particular resource
`ConversionDescription`	Describes a suggested atomic conversion for a particular resource
`ConversionUri`	Identifies an atomic conversion using a URI.

RCD schema:

Syntax:

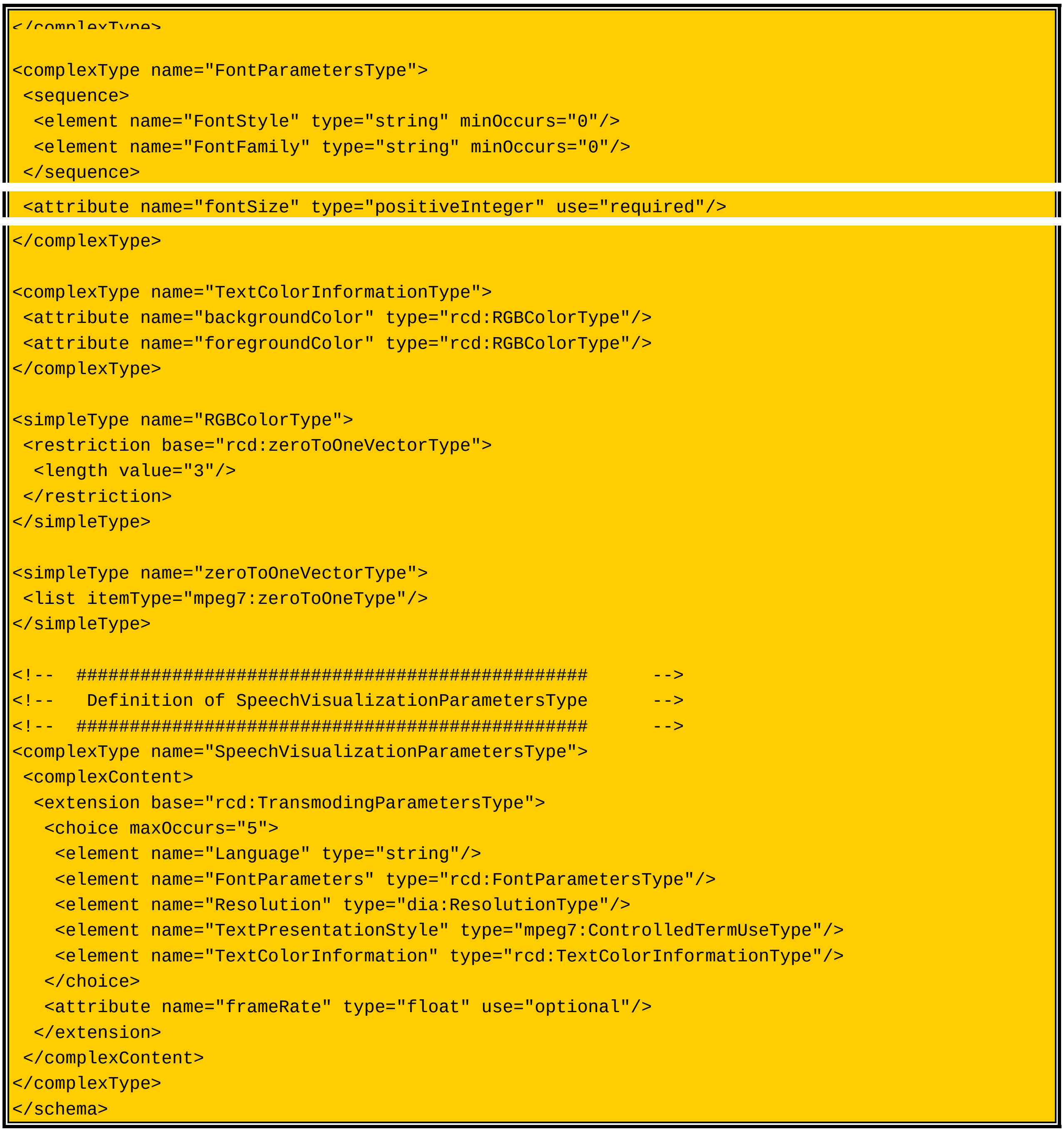
```
<schema   targetNamespace="urn:enst:2005:RCD" xmlns="http://www.w3.org/2001/XMLSchema"
   xmlns:rcd="urn:enst:2005:RCD" xmlns:dia="urn:mpeg:mpeg21:2003:01-DIA-NS"
   xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001" elementFormDefault="qualified"
   attributeFormDefault="unqualified">

   <import namespace="urn:mpeg:mpeg7:schema:2001" schemaLocation="mpeg7-udp-2003-dia.xsd"/>
   <import namespace="urn:mpeg:mpeg21:2003:01-DIA-NS" schemaLocation="ConversionDescriptor.xsd"/>


<!-- ################################################       -->
<!--  Definition of TranscodingConversionType               -->
<!-- ################################################       -->
<complexType name="TranscodingConversionType">
 <complexContent>
  <extension base="dia:ConversionDescriptionBaseType">
   <sequence>
     <element name="Transcoding" type="rcd:TranscodingType" minOccurs="0"/>
   </sequence>
  </extension>
 </complexContent>
</complexType>

<!--  ################################################         -->
<!--   Definition of Transcoding                               -->
<!--  ################################################         -->
<complexType name="TranscodingType">
 <complexContent>
  <extension base="dia:DIABaseType">
   <sequence>
    <element name="Parameters" type="rcd:TranscodingParametersType"/>
   </sequence>
  </extension>
 </complexContent>
</complexType>

<complexType name="TranscodingParametersType" abstract="true">
 <complexContent>
  <extension base="dia:DIABaseType"/>
 </complexContent>
</complexType>

<complexType name="FinalFormatType">
 <complexContent>
  <extension base="rcd:TranscodingParametersType">
   <sequence>
    <element name="TargetFormat" type="mpeg7:ControlledTermUseType" minOccurs="0"/>
   </sequence>
  </extension>
 </complexContent>
```

```
</complexType>


<!-- ################################################       -->
<!--  Definition of TransformingConversionType              -->
<!-- ################################################       -->
<complexType name="TransformingConversionType">
 <complexContent>
  <extension base="dia:ConversionDescriptionBaseType">
   <sequence>
    <element name="Transforming" type="rcd:TransformingType" minOccurs="0"/>
   </sequence>
  </extension>
 </complexContent>
</complexType>

<!--  ################################################      -->
<!--   Definition of TransformingType                       -->
<!--  ################################################      -->
<complexType name="TransformingType">
 <complexContent>
  <extension base="dia:DIABaseType">
   <sequence>
    <element name="Parameters" type="rcd:TransformingParametersType" minOccurs="0"
                                                                          maxOccurs="unbounded"/>
   </sequence>
  </extension>
 </complexContent>
</complexType>

<!--  ################################################      -->
<!--   Definition of ChangingParametersType                 -->
<!--  ################################################      -->
<complexType name="TransformingParametersType" abstract="true">
 <complexContent>
  <extension base="dia:DIABaseType"/>
 </complexContent>
</complexType>

<!--  ################################################       -->
<!--   Definition of TextTranslationType                     -->
<!--  ################################################       -->
<complexType name="TextTranslationType">
 <complexContent>
  <extension base="rcd:TransformingParametersType">
   <sequence>
    <element name="Language" type="mpeg7:ControlledTermUseType" minOccurs="0"/>
    <element name="FontParameters" type="rcd:FontParametersType"/>
   </sequence>
  </extension>
 </complexContent>
</complexType>

<!--  ################################################       -->
<!--   Definition of ImageResizeCroppingType                 -->
<!--  ################################################       -->
<complexType name="ImageResizeCroppingType">
 <complexContent>
  <extension base="rcd:TransformingParametersType">
   <sequence>
```



```
    <element name="StartingPoint" minOccurs="0">
     <complexType>
      <attribute name="x" type="positiveInteger"/>
      <attribute name="y" type="positiveInteger"/>
     </complexType>
    </element>
    <element name="ImageSize" type="dia:ResolutionType"/>
   </sequence>
  </extension>
 </complexContent>
</complexType>

<!--  ################################################       -->
<!--   Definition of SpeechTranslationType                   -->
<!--  ################################################       -->
<complexType name="SpeechTranslationType">
 <complexContent>
  <extension base="rcd:TransformingParametersType">
   <sequence>
    <element name="Language" type="mpeg7:ControlledTermUseType" minOccurs="0"/>
   </sequence>
  </extension>
 </complexContent>
</complexType>

<!--  ################################################       -->
<!--   Definition of TextSubsetType                          -->
<!--  ################################################       -->
<complexType name="TextSubsetType">
 <complexContent>
  <extension base="rcd:TransformingParametersType">
   <sequence>
    <element name="TextSubset" maxOccurs="unbounded">
     <complexType>
      <attribute name="StartingCharacter" type="positiveInteger"/>
      <attribute name="EndingCharacter" type="positiveInteger"/>
     </complexType>
    </element>
   </sequence>
  </extension>
 </complexContent>
</complexType>

<!-- ################################################        -->
<!--  Definition of TransmodingConversionType                 -->
<!-- ################################################        -->
<complexType name="TransmodingConversionType">
 <complexContent>
  <extension base="dia:ConversionDescriptionBaseType">
   <sequence>
    <element name="Transmoding" type="rcd:TransmodingType" minOccurs="0"/>
   </sequence>
  </extension>
 </complexContent>
</complexType>

<!--  ################################################         -->
<!--   Definition of Transmoding                               -->
<!--  ################################################         -->
<complexType name="TransmodingType">
```

```
 <complexContent>
  <extension base="dia:DIABaseType">
   <sequence>
    <element name="Parameters" type="rcd:TransmodingParametersType" minOccurs="0"
                                                                 maxOccurs="unbounded"/>
   </sequence>
  </extension>
 </complexContent>
</complexType>

<!--  ################################################      -->
<!--   Definition of TransmodingParametersType        -->
<!--  ################################################      -->
<complexType name="TransmodingParametersType" abstract="true">
 <complexContent>
  <extension base="dia:DIABaseType">
   <sequence>
    <element name="To" type="mpeg7:ControlledTermUseType"/>
   </sequence>
  </extension>
 </complexContent>
</complexType>


<!--  ################################################      -->
<!--   Definition of VideoSummarizationParametersType       -->
<!--  ################################################      -->
<complexType name="VideoSummarizationParametersType">
 <complexContent>
  <extension base="rcd:TransmodingParametersType">
   <sequence>
    <element name="Slide" maxOccurs="unbounded">
     <complexType>
      <complexContent>
       <extension base="mpeg7:MediaTimeType">
        <attribute name="importance" type="string"/>
       </extension>
      </complexContent>
     </complexType>
    </element>
   </sequence>
  </extension>
 </complexContent>
</complexType>

<!--  ################################################       -->
<!--   Definition of TextVisualizationParametersType         -->
<!--  ################################################       -->
<complexType name="TextVisualizationParametersType">
 <complexContent>
  <extension base="rcd:TransmodingParametersType">
   <choice maxOccurs="4">
    <element name="FontParameters" type="rcd:FontParametersType"/>
    <element name="Resolution" type="dia:ResolutionType"/>
    <element name="TextMotionStyle" type="mpeg7:ControlledTermUseType"/>
    <element name="TextColorInformation" type="rcd:TextColorInformationType"/>
   </choice>
   <attribute name="frameRate" type="float" use="optional"/>
  </extension>
 </complexContent>
```



```
</complexType>

<complexType name="FontParametersType">
 <sequence>
  <element name="FontStyle" type="string" minOccurs="0"/>
  <element name="FontFamily" type="string" minOccurs="0"/>
 </sequence>
 <attribute name="fontSize" type="positiveInteger" use="required"/>
</complexType>

<complexType name="TextColorInformationType">
 <attribute name="backgroundColor" type="rcd:RGBColorType"/>
 <attribute name="foregroundColor" type="rcd:RGBColorType"/>
</complexType>

<simpleType name="RGBColorType">
 <restriction base="rcd:zeroToOneVectorType">
  <length value="3"/>
 </restriction>
</simpleType>

<simpleType name="zeroToOneVectorType">
 <list itemType="mpeg7:zeroToOneType"/>
</simpleType>

<!--  ################################################      -->
<!--   Definition of SpeechVisualizationParametersType      -->
<!--  ################################################      -->
<complexType name="SpeechVisualizationParametersType">
 <complexContent>
  <extension base="rcd:TransmodingParametersType">
   <choice maxOccurs="5">
    <element name="Language" type="string"/>
    <element name="FontParameters" type="rcd:FontParametersType"/>
    <element name="Resolution" type="dia:ResolutionType"/>
    <element name="TextPresentationStyle" type="mpeg7:ControlledTermUseType"/>
    <element name="TextColorInformation" type="rcd:TextColorInformationType"/>
   </choice>
   <attribute name="frameRate" type="float" use="optional"/>
  </extension>
 </complexContent>
</complexType>
</schema>
```

Semantics:

Semantics of `TranscodingType`:

Name	*Definition*
`TranscodingConversionType`	Tool for describing transcoding conversions.
`Transcoding`	Describes a transcoding conversion.
`Parameters`	Describes parameters of a transcoding conversion.
`TargetFormat`	Describes the target format of a transcoding conversion.

Semantics of `TransformingType`:

Name	*Definition*
`TransformingConversionType`	Tool for describing transforming conversions.
`Transforming`	Describes a transcoding conversion.
`Parameters`	Describes parameters of a transforming conversion.

Semantics of `TextTranslationType`:

Name	*Definition*
`TextTranslationType`	Tool for describing the transforming parameters related to text translation.
`Language`	Describes the resulting language.
`FontParameters`	Describes the font parameters of the resulting (rich) text.

Semantics of `ImageResizeCroppingType`:

Name	*Definition*
`ImageResizeCroppingType`	Tool for describing the transforming parameters related to image resizing or cropping.
`StartingPoint`	Describes the starting point of the image.
`ImageSize`	Describes the size of the resulting image

Semantics of `SpeechTranslationType`:

Name	*Definition*
`SpeechTranslationType`	Tool for describing the transforming parameters related to speech translation.
`Language`	Describes the resulting language.

Semantics of `TextSubsetType`:

Name	*Definition*
`ImageResizeCroppingType`	Tool for describing the transforming parameters related to subset selection of a text.
`StartingCharacter`	Describes the character of the original text, which will be the starting character for the subtext.
`EndingCharacter`	Describes the character of the original text, which will be the ending character for the subtext.

Semantics of `TransmodingType`:

Name	*Definition*
`TransmodingConversionType`	Tool for describing transmoding conversions.
`Transmoding`	Describes a transmoding conversion.
`Parameters`	Describes parameters of a transmoding conversion.
`To`	Describes the target modality.

The semantic of the rest of transmoding descriptors is the same as given in Annex A.

ANNEX C: SYNTAX AND SEMANTICS OF SID DESCRIPTION TOOL

SID Syntax:

```
<schema targetNamespace="urn:enst:2004:SID-NS" xmlns:sid="urn:enst:2004:SID-NS"
        xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
         attributeFormDefault="unqualified">


<!-- ################################################       -->
<!--  Definition of Semantic Adaptation Declaration         -->
<!-- ################################################       -->
<element name="SID">
 <complexType>
  <sequence>
   <element name="FragmentationType" type="string" minOccurs="0" maxOccurs="1" />
   <element name="Object" type="sid:ObjectType" minOccurs="1" maxOccurs="unbounded"/>
  </sequence>
 </complexType>
</element>

<!-- ################################################       -->
<!--  Definition of Object                                  -->
<!-- ################################################       -->
<complexType name="ObjectType" >
 <sequence minOccurs="0">
  <element name="SemanticDependencies" type="sid:SemanticDependenciesType" minOccurs="0"/>
  <element name="SemanticFragmentationPreferences"
                                  type="sid:SemanticFragmentationPreferencesType" minOccurs="0"/>
 </sequence>
 <attribute name="id" type="string" use="required"/>
 <attribute name="ref" type="string" use="required"/>
 <attribute name="importance" type="string" use="optional"/>
 <attribute name="role" type="string" use="optional"/>
 <attribute name="maxRRF" type="positiveInteger" use="optional"/>
</complexType>

<!-- ################################################       -->
<!--  Definition of Semantic Dependencies                   -->
<!-- ################################################       -->
<complexType name="SemanticDependenciesType">
 <sequence minOccurs="1" maxOccurs="3">
```

```
  <element name="AbsoluteSemanticDependencies" type="sid:AbsoluteSemanticDependenciesType"
                                                                                 minOccurs="0" />
  <element name="SynchronizationSemanticDependencies"
         type="sid:SynchronizationSemanticDependenciesType" minOccurs="0" maxOccurs="unbounded"/>
  <element name="SpatialSemanticDependencies" type="sid:SpatialSemanticDependenciesType"
                                                                                  minOccurs="0"/>
 </sequence>
</complexType>

<!-- ################################################       -->
<!--  Definition of Absolute Semantic Dependencies          -->
<!-- ################################################       -->
<complexType name="AbsoluteSemanticDependenciesType">
 <sequence>
  <element name="PreConditionMedia" type="sid:MediaType" minOccurs="0" maxOccurs="1"/>
  <element name="RedundantFor" type="sid:MediaType" minOccurs="0" maxOccurs="1"/>
 </sequence>
</complexType>

<!-- ################################################       -->
<!--  Definition of MediaType                               -->
<!-- ################################################       -->
<complexType name="MediaType" >
 <sequence>
  <element name="MediaObject" minOccurs="1" maxOccurs="unbounded">
   <complexType>
    <attribute name="ref" type="string" use="required"/>
   </complexType>
  </element>
 </sequence>
</complexType>

<!-- ################################################       -->
<!--  Definition of Synchronization Semantic Dependencies   -->
<!-- ################################################       -->
<complexType name="SynchronizationSemanticDependenciesType">
 <attribute name="synchronizedTo" type="string" />
</complexType>


<!-- ################################################       -->
<!--  Definition of Spatial Semantic Dependencies           -->
<!-- ################################################       -->

<complexType name="SpatialSemanticDependenciesType">
 <sequence minOccurs="1">
  <element name="KeepCloseTo" type="sid:MediaType" minOccurs="0" maxOccurs="1" />
 <element name="CaptionMedia" type="sid:MediaType" minOccurs="0" maxOccurs="1"/>
</sequence>
</complexType>

<!-- ################################################       -->
<!--  Definition of Semantic Fragmentation Preferences      -->
<!-- ################################################       -->
<complexType name="SemanticFragmentationPreferencesType">
 <sequence minOccurs="1">
  <element name="TimeFragmentationPreferences" type="sid:TimeFragmentationPreferencesType"
                                                                                 minOccurs="0" />
  <element name="SpatialFragmentationPreferences" type="sid:SpacialFragmentationPreferencesType"
                                                           minOccurs="0" maxOccurs="unbounded" />
 </sequence>
</complexType>

<!-- ################################################       -->
<!--  Definition of Time Fragmentation Preferences  -->
<!-- ################################################       -->
<complexType name="TimeFragmentationPreferencesType">
 <attribute name="shiftingPriority" type="positiveInteger" use="optional"/>
 <attribute name="duration" type="positiveInteger" use="optional"/>
</complexType>

<!-- ################################################       -->
<!--  Definition of Spatial Fragmentation Preferences  -->
<!-- ################################################       -->
<complexType name="SpacialFragmentationPreferencesType">
 <attribute name="keepWith" type="string" use="required"/>
 <attribute name="position" type="string" use="optional"/>
</complexType>
</schema>
```

SID semantics:

Semantics of `SID`:

Name	*Definition*
`SID`	Describes semantic information of a multimedia presentation.
`FragmentationType`	Describes the suggested type of fragmentation: sequential or interactive.
`Object`	Describes semantic information of a media object of the presentation.

Semantics of `ObjectType`:

Name	*Definition*
`ObjectType`	Tools for describing semantic information of media objects of the presentation.
`SemanticDependencies`	Describes the semantic dependencies between the corresponding media object and other media objects of the presentation.
`SemanticFragmentationPreferences`	Describes the semantic fragmentation-related preferences for the fragment in which the corresponding media object will be placed.
`id`	Describes the id of the corresponding media object.
`ref`	Describes the reference to the corresponding media object in the presentation. This shall be equivalent with the id of the corresponding media object in the scene.
`importance`	Describes the semantic importance of the corresponding media object.
`role`	Describes the semantic role of the corresponding media object.
`maxRRF`	Describes the authorized maximum spatial resolution

Name	*Definition*
	reduction factor of a visual media object.

Semantics of `SemanticDependenciesType`:

Name	*Definition*
`SemanticDependenciesType`	Tools for describing semantic dependencies between the corresponding media object and other media objects of the presentation.
`AbsoluteSemanticDependencies`	Describes the absolute semantic dependencies between the corresponding media object and other media objects of the presentation.
`SynchronizationSemanticDependencies`	Describes the synchronization-related semantic dependencies between the corresponding media object and other media objects of the presentation.
`SpatialSemanticDependencies`	Describes the spatial semantic dependencies between the corresponding media object and other media objects of the presentation

Semantics of `AbsoluteSemanticDependenciesType`:

Name	*Definition*
`AbsoluteSemanticDependenciesType`	Tools for describing absolute semantic dependencies between the corresponding media object and other media objects of the presentation.
`PreConditionMedia`	Describes for the presence of which media object, the presence of the considered media object is a pre-condition.
`RedundantFor`	Describes that the considered media object is Redundant for which media object of the scene.

Semantics of `MediaType`:

Name	*Definition*
`MediaType`	Tools for referencing a media object.
`MediaObject`	References a media object
`ref`	Describes the id of the referenced media object.

Semantics of `SynchronizationSemanticDependenciesType`:

Name	*Definition*
`SynchronizationSemanticDependenciesType`	Tools for describing synchronization-related semantic dependencies between the corresponding media object and other media objects of the presentation.
`synchronizedTo`	Describes the id of the referenced media object.

Semantics of `SpatialSemanticDependenciesType`:

Name	*Definition*
`SpatialSemanticDependenciesType`	Tools for describing spatial semantic dependencies between the corresponding media object and other media objects of the presentation.
`KeepCloseTo`	Describes to which media object, the considered media object shall be spatially kept close to.

Name	*Definition*
`CaptionMedia`	Describes which media object, is a close caption for the considered media object.

Semantics of `SemanticFragmentationPreferencesType`:

Name	*Definition*
`SemanticFragmentationPreferencesType`	Tools for describing semantic fragmentation-related preferences for the fragment in which the corresponding media object would be placed.
`TimeFragmentationPreferences`	Describes the timing-related preferences of the fragment in which the corresponding media object would be placed.
`SpatialFragmentationPreferences`	Describes the spatial fragmentation-related preferences related to this object.

Semantics of `TimeFragmentationPreferencesType`:

Name	*Definition*
`TimeFragmentationPreferencesType`	Tools for describing timing-related preferences of the fragment in which the corresponding media object would be placed.
`shiftingPriority`	Describes timing priority of the fragment in which the corresponding media object would be placed.
`duration`	Describes the suggested duration of the fragment in which the corresponding media object would be placed.

Semantics of `SpacialFragmentationPreferencesType`:

Name	*Definition*
`SpacialFragmentationPreferencesType`	Tools for describing spatial fragmentation-related preferences for this media object.
`keepWith`	Describes with which media object, this media object should be in the same fragment.
`position`	Describes the position of this media object against the `keepWith` media object in the fragment.

BIBLIOGRAPHY

[1] Jan Bormans, Keith Hill, MPEG-21 Overview v.5, ISO/IEC JTC1/SC29/WG11/N5231, October 2002, Shanghai.

[2] ISIS web site: http://isis.rd.francetelecom.com/

[3] DANAE web site: http://danae.rd.francetelecom.com/

[4] Extensible Markup Language 1.0 (Second Edition), W3C Recommendation, 6 October 2000, http://www.w3.org/TR/2000/REC-xml-20001006.

[5] ISO/IEC 21000-1:2003, Information Technology—Multimedia Framework (MPEG-21)—Part 1: Vision, Technology and Strategy, 2003.

[6] ISO/IEC 21000-2:2004, Study of ISO/IEC FCD 21000-2 DID 2nd Edition, ISO/IEC JTC 1/SC 29/WG 11N6770,October 2004, Spain (Palma de Mallorca).

[7] ISO/IEC 21000-3:2003, Information Technology — Multimedia Framework (MPEG-21) — Part 3: Digital Item Identification, 2003.

[8] Simon Watt, Shane Lauf, Zhongyang Huang, Eva Rodriguez, “Text of ISO/IEC 21000-4 CD”, ISO/IEC JTC 1/SC 29/WG 11/ N6772, October 2004, Spain (Palma de Mallorca).

[9] ISO/IEC 21000-5:2004, Information Technology — Multimedia Framework (MPEG-21) — Part 5: Right Expression Language, 2003.

[10] ISO/IEC 21000-6:2004, Information Technology — Multimedia Framework (MPEG-21) — Part 6: Right Data Dictionary, 2004.

[11] ISO/IEC 21000-7:2004, Information Technology — Multimedia Framework (MPEG-21) — Part 7: Digital Item Adaptation, 2004.

[12] José M. Martínez, “MPEG-7 Overview (version 9)”, ISO/IEC JTC1/SC29/WG11/N5525, March 2003, Pattaya.

[13] Frederik De Keukelaere, Christian Timmerer, Gerrard Drury and Xin Wang, “ISO/IEC 21000-8/FCD MPEG-21 reference software”, ISO/IEC JTC 1/SC 29/WG 11/N6628, July 2004, USA (WA, Redmond).

[14] José M. Martínez, Rob Koenen and Fernando Pereira, “MPEG-7: the generic multimedia content description standard (part 1)”, IEEE Multimedia, 9 (2), April-June 2002, 78-87.

[15] Frank Manola, Eric Miller, “RDF Primer”, World Wide Web Consortium Working Draft 23 January 2003: http://www.w3.org/TR/2003/WD-rdf-primer-20030123.

[16] “Overview and applicability of metadata standards, v1.0, Technical report”, CILab, Department of Cultural Technology and Communication, University of the Aegean, January 2003.

[17] Graham Klyne et al, “Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies”, http ://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115/, W3C Recommendation, 15 January 2004.

[18] Indulska, J., Robinson, R., Rakotonirainy, A., Henricksen, K., “Experiences in using CC/PP in context-aware systems”, Proceedings of the 4th International Conference on Mobile Data Management, 21-24 January, 2003, Melbourne, Australia. Lecture Notes in Computer Science. Springer Verlag, LNCS 2574. pp. 247-261.

[19] http://www.w3.org/TR/css-mobile

[20] WAP-174: “UAProf User Agent Profiling Specification (1999)” as amended by WAP-174_100 User Agent Profiling Specification Information Note (2001) Wireless Application Protocol Forum available at http://www.wapforum.org/what/technical_1_2.htm.

[21] RFC 2533: A Syntax for Describing Media Feature Sets; G. Klyne; IETF Request for Comments: ftp://ftp.isi.edu/in-notes/rfc2533.txt.

[22] http://www.imagemagick.org

[23] http://ffmpeg.sourceforge.net

[24] Z. Lei and N. D. Georganas, “A General Framework for Context-based Media Adaptation” Proceedings of ACM Multimedia 2001 Doctoral Symposium, September 2001, Ottawa, Canada.

[25] Thomas DeMartini, Chris Barlas, Jean-Claude Dufourd, “Text of ISO/IEC 21000-7 P/DAM-1, ISO/IEC JTC 1/SC 29/WG 11/N6647”, July 2004, Redmond, USA.

[26] Smith J. R., Mohan R. and Li C. S., “Scalable Multimedia Delivery for Pervasive Computing”, Proceedings of Seventh ACM International Conference on Multimedia (Part 1), p. 131-140, 30 October- 4 November, 1999, Orlando, Florida, USA.

[27] Rakesh Mohan, John R. Smith, Chung-Sheng Li: “Adapting Multimedia Internet Content for Universal Access”, IEEE Transactions on Multimedia 1(1): pp.104-114, Vol. 1, No. 1, March, 1999.

[28] Tayeb Lemlouma, “Architecture de négociation et d’adaptation de services multimédia dans des environnements hétérogènes”, Ph.D. thesis, 2004, Institut Nationale Polytechnique de Grenoble, France.

[29] Tayeb Lemlouma and Nabil Layaïda, “Universal Profiling for Content Negotiation and Adaptation in Heterogeneous Environments”, W3C Workshop on Delivery Context, W3C/INRIA Sophia-Antipolis, 4-5 March 2002, France.

[30] Minsu Jang, Jaehong Kim, Joochan Sohn, “Web Content Adaptation and Transcoding based on CC/PP and Semantic Templates”, The Twelfth International World Wide Web Conference, 20-24 May 2003, Budapest, Hungary.

[31] Wai Yip Lum , Francis C. M. Lau, A Context-Aware Decision Engine for Content Adaptation, IEEE Pervasive Computing, Vol. 1 No. 3, pp.41-49, July 2002.

[32] M. Hori, G. Kondoh, K. Ono, S. Hirose, and S. Singhal, “Annotation-based Web content transcoding”. The International Journal of Computer and Telecommunications Networking, Vol. 33, No. 1 , pp. 197-211, July 2000.

[33] Peter Soetens, Matthias De Geyter and Stijn Decneut, “Multi-step Media Adaptation with Semantic Web Services”, 3rd International Semantic Web Conference ISWC2004, 2004, Hiroshima, Japan.

[34] Maja Metso and Jaakko Sauvola, “The media wrapper in the adaptation of multimedia content for mobile environments”, Proceedings SPIE Vol. 4209, Multimedia Systems and Applications III, Boston, MA, USA.

[35] Susanne Boll, Wolfgang Klas, and Jochen Wanden, “A Cross-Media Adaptation Strategy for Multimedia Presentation”, Proceedings of the ACM Multimedia'99, October 30 - November 5, 1999, Orlando, Florida, USA.

[36] Sylvain Devillers, Christian Timmerer, Jörg Heuer, and Hermann Hellwagner, Bitstream Syntax Description-Based Adaptation in Streaming and Constrained Environments, to appear in IEEE Transactions on Multimedia, vol. 7, no. 2, April 2005.

[37] Simple API for XML. http://www.saxproject.org/

[38] O. Becker, “Transforming XML on the Fly”, XML Europe 2003, 5-8, May 2003.

[39] Streaming Transformations for XML (STX), Working Draft 5, May 2003. http://stx.sourceforge.net/.

[40] Gabriel Panis, Andreas Hutter, Jörg Heuer, Hermann Hellwagner, Harald Kosch, Christian Timmerer, Sylvain Devillers and Myriam Amielh, Bitstream Syntax Description: A Tool for Multimedia Resource Adaptation within MPEG-21, EURASIP Signal Processing: Image Communication Journal, Vol. 18, No. 8, pp. 721-747, September 2003 (Special Issue on Multimedia Adaptation).

[41] Kamyab K., et al. “ISIS: Intelligent Scalability for Interoperable Service”, Conference on visual media production CVMP 2004, 15-16 March 2004, London, UK.

[42] Cyril Concolato and Jean-Claude Dufourd, “Adaptation de contenu MPEG-4 BIFS suivant la norme MPEG-21”, Mcube Multimédia et Mobilité, 30-31 March 2004, Montbéliard, France.

[43] J. M. Martínez, J. Bescós, V. Váldes and L. Herranz, “ Integrating Metadata-Driven Content Adaptation Approaches”, Proceedings of the European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology, November 2004, London, UK.

[44] Cong Thang T., Ju Jung Y., Wook Lee J., Man Ro Y., “Modality conversion for universal multimedia services”, International Workshop on Image Analysis for Multimedia Interactive Services, 21-23 April 2004, Lisbon, Portugal.

[45] Dietmar Jannach, Klaus Leopold, Hermann Hellwagner, Christian Timmerer, “A Knowledge Based Approach for Multi-step Media Adaptation”, International Workshop on Image Analysis for Multimedia Interactive Services, 21-23 April 2004, Lisbon, Portugal.

[46] http://www.tiramisu-project.org/

[47] Frederik De Keukelaere, Christian Timmerer, Gerrard Drury and Xin Wang, “ISO/IEC JTC 1/SC 29/WG 11/N6628”, July 2004, Redmond, WA, USA.

[48] http://gpac.sourceforge.net

[49] "HTML 4.01 Specification" D. Raggett, A. Le Hors, I. Jacobs. W3C Recommendation, 24 December 1999, Available at http://www.w3.org/TR/html401/

[50] A. Borning, R. Lin and K. Marriott, “Constraint-based document layout for the Web”, ACM/Springer Verlag Multimedia Systems Journal Vol. 8 No. 3, 2000, pp. 177-189, 2000.

[51] "Synchronized Multimedia Integration Language (SMIL) 1.0", P. Hoschka. W3C Recommendation 15 June 1998, Available at http://www.w3.org/TR/REC-smil.

[52] Rob Koenen, “Overview of the MPEG-4 Standard”, http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm, March 2002.

[53] Boll S., Klas W., “ZYX - A Semantic Model For Multimedia Documents and Presentations”, Proceedings of the 8th IFIP Conference on Data Semantics (DS-8), Rotorua, New Zeeland, 5-8 January 1999.

[54] Boll S., “ZYX: Towards Flexible Multimedia Document Models for Reuse and Adaptation”, Ph.D. thesis, University of Vienna, Austria, 2001, available on http://mmit.informatik.uni-oldenburg.de/pubs/dissertation.

[55] XSL Transformations (XSLT) Version 1.0, W3C Recommandation, http://www.w3.org/TR/xslt/, 16 November 1999.

[56] Lionel Villard, “Modèles de document pour l'édition et l'adaptation de présentations multimédia”, Ph.D. thesis, INPG, France, 2002, available on http://www.research.ibm.com/people/v/villard.

[57] Synchronized Multimedia Working Group of W3C, “Synchronized Multimedia Integration Language (SMIL 2.0)”, W3C Recommendation, http://www.w3.org/TR/smil20/, 07 August 2001.

[58] Synchronized Multimedia Working Group of W3C, “Synchronized Multimedia Integration Language (SMIL 2.1)”, W3C Recommendation, http://www.w3.org/TR/2005/WD-SMIL2-20050201/, 01 February 2005.

[59] "XHTML+SMIL Profile" Debbie Newman, Patrick Schmitz, Aaron Patterson. W3C Working Draft, work in progress. Available at http://www.w3.org/TR/XHTMLplusSMIL/

[60] "Scalable Vector Graphics (SVG) 1.1 Specification". W3C Recommendation 14 January 2003, Available at http://www.w3.org/TR/SVG.

[61] VRML 2.0 Specification, ISO/IEC 14772-1:1997, http://www.vrml.org/specifications.

[62] Web site of Opera project, http://opera.inrialpes.fr.

[63] N. Layaïda, “Madeus: système d'édition et de présentation de documents structurés multimedia», Ph.D. thesis, Université Joseph Fourier, Grenoble, France, June 1997. Available at http://opera.inrialpes.fr/people/Nabil.Layaida/these/toc.html.

[64] M. Jourdan, N. Layaïda, L. Sabry-Ismail, “MADEUS: an authoring environment for multimedia documents”, IEEE International Conference on Multimedia Computing and Systems, pp. 644-645, Ottawa (Canada), June 1997.

[65] W3C, XML Link Language (XLink), http://www.w3.org/TR/xlink, 2000.

[66] Lionel Villard, Cécile Roisin, Nabil Layaïda, “An XML-based multimedia document processing model for content adaptation”, In Proceeding of the Eighth International Conference on Digital Documents and Electronic Publishing (DDEP’00), Springer Computer Science, Munich, Germany, September 14, 2000.

[67] Web site of LASeR, http://www.mpeg-laser.org.

[68] “Information Processing -- Text and Office Systems -- Standard Generalized Markup Language (SGML)”, ISO 8879:1986. http://www.iso.ch/cate/d16387.html

[69] S. Peak and J. R. Smith, “Detecting Image Purpose in World-Wide Web Documents”, Symposium on Electronic Imaging: Science and Technology - Document Recognition, IS&T/SPIE, 1998.

[70] F. Rousseau, J.A. García-Macías, J. Valdeni de Lima, A. Duda, “User Adaptable Multimedia Presentations for the WWW”, Computer Networks. Vol. 31, No. 11-16, pp. 1273-1290 , 1999.

[71] J. Euzenat, N. Layaïda, V. Dias, “A semantic framework for multimedia document adaptation”, Proceeding of 18th International Joint Conference on Artificial Intelligence (IJCAI), Acapulco (MX), San-Mateo (CA US), p. 31-36 (2003).

[72] W3C Semantic Web Activity, http://www.w3.org/2001/sw/.

[73] Katashi Nagao et al., “Semantic Annotation and Transcoding: Making Web Content More Accessible”, IEEE MultiMedia, 2001, pp. 69-81.

[74] Masahiro Hori et al., “Annotation-based Web Content Transcoding”, Computer Networks, June 2000, pp. 197-211.

[75] XML Path Language (XPath) Version 1.0. W3C Recommendation, http://www.w3.org/TR/xpath (11/1999).

[76] XML Pointer Language (XPointer). W3C Working Draft, http://www.w3.org/TR/WD-xptr (12/1999).

[77] Annotation of Web Content for Transcoding. W3C Note, http://www.w3.org/TR/annot/ (07/1999).

[78] Lemlouma T. and Layaïda N., “SMIL Content Adaptation for Embedded Devices”, SMIL Europe 2003, Paris, 12-14 February, 2003.

[79] Robert Steele, Marcin Lubonski, Yuri Ventsov, and Elaine Lawrence: “Accessing SMIL-based Dynamically Adaptable Multimedia Presentations from Mobile Devices”, Proceeding of the International Conference on Information Technology: Coding and Computing, ITCC04, 2004.

[80] Wi-Fi Specification. Available at http://grouper.ieee.org/groups/802/11/.

[81] GPRS Platform. Available at http://www.gsmworld.com/technology/gprs/index.shtml.

[82] http://www.w3.org/TR/2001/WD-xhtml-events-20010607/Overview.html

[83] http://www.w3.org/MarkUp/Forms/

[84] Tayeb Lemlouma, Nabil Layaïda, “Device Independent Principles for Adapted Content Delivery”, available at http://opera.inrialpes.fr/people/Tayeb.Lemlouma/publication.html.

[85] http://natalian.org/archives/2004/09/27/riml/

[86] G. Grassel, M. Lau. and A. Spriestersbach. "Definition and Prototyping of Renderer-Independent ML", Nokia, SAP and IBM Germany, W3C Workshop on Device Independent Authoring Techniques, 25-26 September 2002, SAP University, St. Leon-Rot, Germany.

[87] Consensus Web site, http://www.consensus-online.org.

[88] Consensus Project, "RIML Specification Document", 3G Mobile Context Sensitive Adaptability - User Friendly Mobile Work Place for Seamless Enterprise Applications, Consensus member of W3C DI WG, http://www.consensus-online.org/publicdocs/Consensus_32407_D4_ExecSum.pdf

[89] Consensus Project, "RIML Layout Definition", 3G Mobile Context Sensitive Adaptability - User Friendly Mobile Work Place for Seamless Enterprise Applications, Consensus member of W3C DI WG, https://www.consensus-online.org/publicdocs/20031021-RIML-layout.pdf, October 2003.

[90] Gabriel Dermler, Michael Wasmund, Guido Grassel, Axel Spriestersbach, and, Thomas Ziegert, "Flexible pagination and layouting for device independent authoring", WWW2003 Emerging Applications for Wireless and Mobile access Workshop, 2003.

[91] RealPlayer, available at http://www.real.com/realplayer.html.

[92] QuickTime, available at http://www.apple.com/quicktime.

[93] Microsoft Internet Explorer, available at http://www.eu.microsoft.com/windows/ie/default.mspx.

[94] Available at http://www.oratrix.com/Products/G2P.

[95] Ambulant open SMIL player, available at http://www.cwi.nl/projects/Ambulant/distPlayer.html.

[96] Synchronized Multimedia, http://www.w3.org/AudioVideo.

[97] Available at http://www.inobject.com/mmplay.htm.

[98] PocketSmil 2.0, available at http://wam.inrialpes.fr/software/pocketsmil.

[99] 3GPP Specification detail, available at http://www.3gpp.org/ftp/Specs/html-info/26246.htm

[100] C. McCormack, K. Marriott, and B. Meyer, "Adaptive layout using one-way constraints in SVG", Proceedings of third Annual Conference on Scalabe Vector Graphics, SVG Open 2004, Tokyo, Japan, September 2004, available on http://www.svgopen.org/2004/papers/ConstraintSVG.