Helsinki University of Technology
Publications in Telecommunications Software and Multimedia
Teknillisen korkeakoulun tietoliikenneohjelmistojen ja multimedian julkaisuja
Espoo 2003 TML-A6

EXTENSIONS TO THE SMIL MULTIMEDIA LANGUAGE

Kari Pihkala

TEKNILLINEN KORKEAKOULU
TEKNISKA HÖGSKOLAN
HELSINKI UNIVERSITY OF TECHNOLOGY

Helsinki University of Technology
Publications in Telecommunications Software and Multimedia
Teknillisen korkeakoulun tietoliikenneohjelmistojen ja multimedian julkaisuja
Espoo 2003 TML-A6

EXTENSIONS TO THE SMIL MULTIMEDIA LANGUAGE

Kari Pihkala

Dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the Department of Computer Science and Engineering, for public examination and debate in Auditorium T2 at Helsinki University of Technology (Espoo, Finland) on the 28th of November, 2003, at 12 noon.

Helsinki University of Technology
Department of Computer Science and Engineering
Telecommunications Software and Multimedia Laboratory

Teknillinen korkeakoulu
Tietotekniikan osasto
Tietoliikenneohjelmistojen ja multimedian laboratorio

Distribution:
Helsinki University of Technology
Telecommunications Software and Multimedia Laboratory
P.O.Box 5400
FIN-02015 HUT
Tel. +358-9-451 2870
Fax. +358-9-451 5014

ISBN 951-22-6803-5 (print)
ISBN 951-22-6804-3 (pdf)
ISSN 1456-7911

Otamedia Oy
Espoo 2003

Abstract

The goal of this work has been to extend the Synchronized Multimedia Integration Language (SMIL) to study the capabilities and possibilities of declarative multimedia languages for the World Wide Web (Web). The work has involved design and implementation of several extensions to SMIL.

A novel approach to include 3D audio in SMIL was designed and implemented. This involved extending the SMIL 2D spatial model with an extra dimension to support a 3D space. New audio elements and a listening point were positioned in the 3D space. The extension was designed to be modular so that it was possible to use it in conjunction with other XML languages, such as XHTML and Scalable Vector Graphics (SVG) language.

Web forms are one of the key features in the Web, as they offer a way to send user data to a server. A similar feature is therefore desirable in SMIL, which currently lacks forms. The XForms language, due to its modular approach, was used to add this feature to SMIL. An evaluation of this integration was carried out as part of this work.

Furthermore, the SMIL player was designed to play out dynamic SMIL documents, which can be modified at run-time and the result is immediately reflected in the presentation. Dynamic SMIL enables execution of scripts to modify the presentation. XML Events and ECMAScript were chosen to provide the scripting functionality.

In addition, generic methods to extend SMIL were studied based on the previous extensions. These methods include ways to attach new input and output capabilities to SMIL.

To experiment with the extensions, a Synchronized Multimedia Integration Language (SMIL) player was developed. The current final version can play out SMIL 2.0 Basic profile documents with a few additional SMIL modules, such as event timing, basic animations, and brush media modules. The player includes all above-mentioned extensions.

The SMIL player has been designed to work within an XML browser called X-Smiles. X-Smiles is intended for various embedded devices, such as mobile phones, Personal Digital Assistants (PDA), and digital television set-top boxes. Currently, the browser supports XHTML, SMIL, and XForms, which are developed by the current research group. The browser also supports other XML languages developed by 3rd party open-source projects.

The SMIL player can also be run as a standalone player without the browser. The standalone player is portable and has been run on a desktop PC, PDA, and digital television set-top box. The core of the SMIL player is platform-independent, only media renderers require platform-dependent implementation.

Keywords: XML, SMIL, multimedia, player, XForms, scripting, 3D sound

Preface

This work was carried out in the Telecommunications Software and Multimedia Laboratory, Helsinki University of Technology, Finland, during the years 2001-2003.

I would like to thank my thesis supervisor Professor Petri Vuorimaa for an interesting research topic and all the guidance during the work. This thesis would not have been possible without his encouragement and supervision.

Special thanks go to the group working with the X-Smiles browser, Mr. Mikko Honkala, Mr. Juha Vierinen, Mr. Mikko Pohja, and Mr. Alessandro Cogliati. Furthermore, Mr. Pablo Cesar deserves thanks as the digital television expert. They all gave me new ideas and stimulated good debate about multimedia, the art of programming, computer science, and triangular shaped ion-wind lifters. The original student group, who started the X-Smiles browser, deserves thanks for building a great playfield for us. In addition, I am grateful to the co-authors Mr. Tapio Lokki and Mr. Niklas von Knorring. Niklas was the original developer of the SMIL 1.0 player, which gave me a quick start to the research work.

Moreover, I would like to thank the pre-examiners of my thesis, Prof. Helena Ahonen-Myka and Dr. Lloyd Rutledge. They gave good improvement suggestions and feedback about the thesis. Mike Clark deserves thanks for improving the language of this thesis.

I would like to acknowledge the personnel of Telecommunications Software and Multimedia Laboratory for taking care of daily routines. This meant that I was able to concentrate on my research work instead of messy paperwork.

Finally, I would like to thank the Nokia Foundation for financial support during the years 2001-2003.

Otaniemi, Espoo, 8th November 2003

Kari Pihkala

Table of Contents

List of Publications

This thesis summarizes the following publications, referred to as [P1]-[P8]:

[P1] K. Pihkala and P. Vuorimaa, "Nine Methods to Extend SMIL for Multimedia Applications," submitted to *Multimedia Tools and Applications*.

[P2] K. Pihkala, M. Honkala and P. Vuorimaa, "Multimedia Web Forms," in *SMIL Europe 2003 Conference*, Paris, France, February 12-14, 2003.

[P3] K. Pihkala and T. Lokki, "Extending SMIL with 3D Audio," in *Proceedings of the International Conference on Auditory Display*, Boston, USA, July 6-9, 2003, pp. 95-98.

[P4] K. Pihkala and P. Vuorimaa, "Design of a Dynamic SMIL Player," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, Lausanne, Switzerland, August 26-29, 2002, pp. 189-192.

[P5] K. Pihkala, J. Vierinen, and P. Vuorimaa, "Content Customization Using Device Profiles," in *Proceedings of the 2nd Intl. Workshop on Intelligent Multimedia and Networking*, Durham, North Carolina, USA, March 8-12,2002.

[P6] K. Pihkala, N. von Knorring, and P. Vuorimaa, "SMIL in X-Smiles," in *Proceedings of the International Conference on Distributed Multimedia Systems*, Taipei, Taiwan, September 26-28, 2001, pp. 416-422.

[P7] K. Pihkala, P. Cesar, and P. Vuorimaa, "Cross-platform SMIL Player," in *Proceedings of the IASTED International Conference Communications, Internet and Information Technology*, St. Thomas, Virgin Islands, USA, November 18-20, 2002, pp. 48-53.

[P8] K. Pihkala, M. Honkala, and P. Vuorimaa, "A Browser Framework for Hybrid XML Documents," in *Proceedings of the 6th IASTED International Conference: Internet and Multimedia Systems, and Applications*, Kauai, Hawaii, USA, August 12-14, 2002, pp. 164-169.

List of Abbreviations

2.5G	Second and a Half Generation
3G	Third Generation
3GPP	The 3rd Generation Partnership Project
3D	Three-dimensional
ADSL	Asymmetric Digital Subscriber Line
AABIFS	Advanced AudioBIFS
API	Application Programming Interface
ASN.1	Abstract Syntax Notation 1
ATM	Automatic Teller Machine
AWT	Abstract Window Toolkit
BIFS	BInary Format for Scenes
CSiro	Commonwealth Scientific & Industrial Research Organization
CC/PP	Composite Capability / Preference Profile
CSS	Cascading Stylesheet
CPU	Central Processing Unit
CD-ROM	Compact Disk-Read Only Memory
DAVIC	Digital Audio-Visual Industry Consortium
DI	Digital Item
DIA	Digital Item Adaptation
DVD	Digital Versatile Disk
DSSSL	Document Style Semantics and Specification Language
DTD	Document Type Definition
DVB	Digital Video Broadcasting
ECMA	European Computer Manufacturers Association
ftv GUI	Future TV Graphical User Interface
GIF	Graphics Interchage Format
GIS	Geographic Information Systems
GUI	Graphical User Interface
HAVi	Home Audio/Video Interoperability
HPAS	Hypermedia Presentation and Authoring System
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
HyTime	Hypermedia/Time-Based Structuring Language
INRIA	Institut National de Recherche en Informatique et en Automatique
ISDN	Integrated Services Digital Network
IPMP	Intellectual Property Management and Protection
ISO	International Standardization Organization
J2ME	Java 2 Micro Edition
J2SE	Java 2 Standard Edition
JMF	Java Media Framework
JPEG	Joint Photographic Experts Group

LAN	Local Area Network
LOTOS	Language of Temporal Ordering Specification
MHEG	Multimedia and Hypermedia Information Coding Experts Group
MLFC	Markup Language Functional Component
MMAPI	Mobile Multimedia API
MMS	Multimedia Messaging System
MPEG	Motion Pictures Experts Group
NIST	National Institute of Standards and Technology
PC	Personal Computer
PDA	Personal Digital Assistant
POTS	Plain Old Telephone System
RAM	Random Access Memory
RDD	Rights Data Dictionary
RDF	Resource Description Framework
REL	Rights Expression Language
RT-LOTOS	Real-Time LOTOS
RTL	RT-LOTOS Laboratory
SGML	Standard Generalized Markup Language
S2M2	Streaming Synchronized MultiMedia
SMDL	Standard Music Description Language
SMIL	Synchronized Multimedia Integration Language
SMPTE	Society of Motion Picture and Television Engineers
SVG	Scalable Vector Graphics
SYMM	Synchronized Multimedia Working Group
TLA	Time Labeled Automaton
TTY	TeleTYpewriter
TV	Television
UAProf	WAP User Agent Profile
UPS	Universal Profiling Schema
URI	Universal Resource Identifier
VCR	Video Cassette Recorder
VDSL	Very high speed asymmetric Digital Subscriber Line
VoIP	Voice over Internet Protocol
VRML	Virtual Reality Markup Language
W3C	World Wide Web Consortium
WAP	Wireless Application Protocol
WAV	Waveform audio
WML	Wireless Markup Language
WWW	World Wide Web
XHTML	XML version of HTML
X3D	Extensible 3D Graphics
XML	Extensible Markup Language
XMT	Extensible MPEG-4 Textual Format
XSL	Extensible Stylesheet Language
XSLT	XSL Transformations
XSL FO	XSL Formatting Objects

1 Introduction

Since Extensible Markup Language (XML) was introduced in 1998, several XML based languages have evolved. Synchronized Multimedia Integration Language (SMIL) [15] was the first XML based language released in 1998. It is a multimedia language for the World Wide Web (Web) and already has two versions, SMIL 1.0 and SMIL 2.0. Other XML languages have been introduced since, such as XHTML [80] and XForms [24]. XHTML is an XML based version of the popular HTML, while XForms is a new form language expected to replace legacy forms in the Web.

This thesis describes an implementation of an extensible SMIL 2.0 player. The player has been extended with 3D audio, XForms, new input and output capabilities, and it renders dynamic SMIL documents. The last, for instance, enables scripting. These will allow for richer multimedia presentations and also indicate some issues with the integration of XML languages.

1.1 The Rise of Multimedia

Multimedia is here. It has been applied in game technology for decades and it is now taking over television and mobile phones. In the future, other devices will emerge. As they all are connected to the Internet, they will have access to Web based multimedia.

Internet connected devices can be categorized in several ways. Table 1 gives an overview to various categorizations in selected references. Revett et al. [85] have categorized devices based on access to Internet commerce services, while Lewis [68] has categorized Information Appliances, i.e., smart devices connected to the Web. Korolev et al. [64] have categorized devices for Web page adaptation. For a similar reason, the Cascading Stylesheet (CSS) specification [9] has rules to display styles for various media types. All of these point out that there are a wide variety of devices accessing the Internet. Thus, Internet content, including Web based multimedia, will be presented in a wide variety of gadgets.

Figure 1 depicts an estimate of the shipment of multimedia terminals in 2003 [74]. Clearly, mobile phones will be the best selling multimedia terminals. PCs, games consoles, and set-top boxes are forecasted to be the next best sold.

Author and Publication	Categories
Revett et al. [85] (Access devices to Internet commerce services)	• Internet screenphones (i.e., telephones with Internet access and a screen) • Set-top boxes (i.e., cable and digital TV) • High street devices (i.e., ATMs and Internet kiosks) • Network computers • Fixed-line telephones
Lewis et al. [68] (Information Appliances)	• Fixed-line and mobile telephones • Web-enabled set-top boxes • Personal Information Managers (PDAs) • WebMisc (e.g., Internet connected cars, digital cameras, and watches)
Korolev et al. [64] (Device categories for Web page adaptation)	• Desktops with broadband connection • Notebooks with dial-up/wireless connection • Handheld computers (PDAs) • WAP devices
CSS Specification [9] (Various media types)	• Speech synthesizers • Braille tactile feedback devices • Paged braille printers • Handheld devices • Paged opaque materials • Projected displays • Color screens • TTY terminals • Televisions

Table 1: Various categorizations of devices.

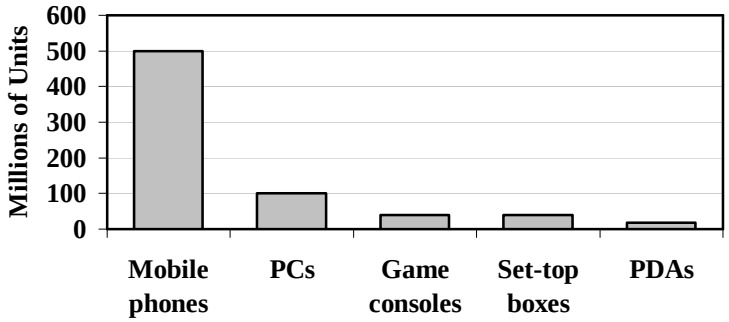

Figure 1: Estimated world-wide shipment of various multimedia terminals in 2003. (Figures are based on various public market research sources) [74].

Mobile phones have become multimedia terminals [34]. Mobile multimedia services are based on extensive standardization work, which provides high quality and wide interoperability. This standardization work is heavily conducted

at the 3rd Generation Partnership Project (3GPP), which brings together a number of telecommunications standards bodies. The emerging 3G mobile networks will utilize an even broader set of multimedia standards, one of them being SMIL.

Game consoles, such as Sony's PlayStation 2, have become very powerful dedicated computers [31]. Games can be optimized for a single platform and there are no compatibility issues between single supplier's units. The future of game consoles will undoubtedly be even more innovative, as they are not constrained by the backward compatibility requirement, which is slowing down the evolution of PCs.

Digital television is bringing the multimedia experience into living rooms. Multimedia applications are distributed along with digital television signals and displayed over a TV screen. Digital television can enrich broadcasted TV programs with interactive content made of text, images, audio, and video [81]. Currently, the digital television broadcast can be received with a set-top box, which can be attached to an analog television. A return channel for interactive applications is provided with a modem or an ADSL connection.

Convergence between game consoles, PCs, digital TVs, and mobile phones is evident [40]. It results from digital transmission and Internet technology. The former provides an abstraction between service and service delivery. The same network can deliver different service types and different networks can deliver the same services. Internet technology creates a common platform for content addressing and interpretation. Thus, services are available on different terminals. For instance, game consoles and digital televisions are used to browse the Web, while PCs can display TV services.

However, divergence is also occurring [40]. As the mentioned systems mature, new, dedicated devices appear. Pervasive computing [107] will drive computational intelligence into daily appliances, such as pens, vending machines, microwave ovens, watches, and toys. Services will be available everywhere at any time.

Convergence makes it an appealing idea to create a common platform for all devices, a so-called device independent platform. The common factor in these devices is the connection to the Internet. However, divergence makes it difficult to guarantee interoperability [40]. As an example, browsing the Web is possible with TV sets, although it is not very popular due to rendering and navigation problems.

1.2 Multimedia Applications

Multimedia has several applications. Žagar et al. [111] have categorized these using two criteria. The first groups multimedia applications in four main categories emphasizing users and markets. The categories are multimedia collaboration enabling real-time audio and video communication, multimedia information services providing access to multimedia databases, audio-visual services on demand, and multimedia messaging. Typical applications in each category are listed in Table 2. The second criterion considers traffic and networks and groups applications into three categories, which are real-time

streaming, real-time applications with block transmission (e.g., Web browsing, interactive games), and non real-time applications (e.g., e-mail).

Category	Typical Applications
Multimedia collaboration	Videotelephony, videoconference, computer supported collaborative work, telemedicine, and interactive games
Multimedia information services	Public and business information, home banking and shopping, news, and magazines
Audio-visual services on demand	Video-on-demand pay TV, distance presentation, and broadcast videography
Multimedia messaging	E-mail, voice mail, and MMS messages

Table 2: Categories of multimedia applications based on user and market criteria [111].

Traditionally, envisioned multimedia applications have been entertainment (i.e., video-on-demand, interactive cinema, and networked games), education, home shopping, healthcare, Geographic Information Systems (GIS), and multimedia communications [63]. More recent applications are infotainment [89] and super teletext [81]. Neuvo et al. [74] have envisioned three modes of interaction in the future mobile terminals. These are multimedia browsing, multimedia messaging, and rich call. A rich call is a combination of audio, image, and video communication.

Multimedia messaging [97] is the latest innovation in 2.5G mobile phones. It enables sending of text, images, audio, and video composed as a presentation to other phones via Multimedia Messaging System (MMS). Composition of these messages is made easy with an in-built camera capable of taking images and videos. In addition to person-to-person messaging, various content services are offered. Figure 2 depicts a map service, which provides a street map for any address in Finland.

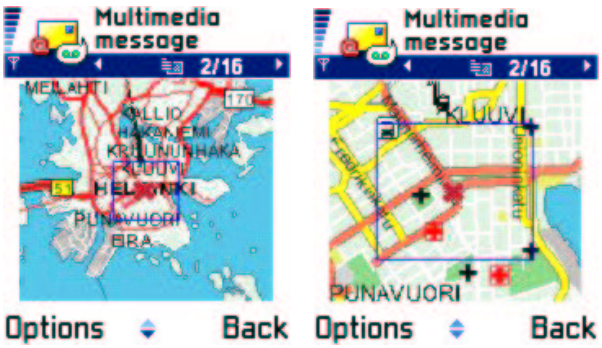

Figure 2: MMS based map service in a mobile phone (with permission, from Mind On Move Oy).

Infotainment [88] is information, which is made entertaining and exciting with the help of audio, video, and increased user interaction. Info kiosks typically utilize infotainment to keep the user interested.

Digital television super teletext is an enhanced version of the teletext in analog televisions [81]. It offers full color images and more information compared to the legacy systems. Pages can be interactive and feedback can be sent via a return channel, which typically uses a telephone line or a cable TV for data transmission.

1.3 Multimedia Systems

A broad range of applications requires different types of multimedia systems. Multimedia systems can be standalone or distributed, interactive or non-interactive, persistent or transient, and predetermined or indeterminate. Several perspectives to such systems are possible, depending on the application. The perspectives are [33]:

- The systems perspective considers enabling hardware technology, streaming servers, high-speed networks, operating systems, schedulers, and communication systems.
- The engineering and scientific perspective uses multimedia to visualize voluminous data sets.
- The arts and education perspective considers multimedia authoring and presentations, multimedia browsers, and digital libraries.
- The database and information retrieval perspective considers graphical/iconic query languages, retrieval effectiveness, and semi-automated tools for semantic content capturing.
- The teleconferencing and collaborative work environments perspective uses distributed interactive real-time multimedia to share data.
- The entertainment, broadcasting, and business services perspective covers, e.g., movies on demand, digital television, telemedicine, and news.

Varying multimedia systems are problematic for content providers and users. The same content does not run in every system. MPEG-21 [10] is a multimedia framework, which offers interoperable exchange, access, consumption, trade, and the manipulation of digital media. It defines Digital Items (DI), which are packages of media resources. Furthermore, it defines Intellectual Property Management and Protection (IPMP) tools, machine interpretable Rights Expression Language (REL), and Rights Data Dictionary (RDD) to associate intellectual properties and copyright information to DIs. MPEG-21 Digital Item Adaptation (DIA) can produce adapted DIs based on the terminal capabilities, network capabilities, delivery capabilities, user characteristics, natural environment characteristics, service capabilities, and relations among users. MPEG-21 is an ongoing standardization effort by ISO, and is expected to be finished by the end of 2003.

1.4 Aim of the Study

The recent trend has been to create multimedia languages as declarative languages, instead of using a procedural approach, such as programming language APIs or scripts. Declarative languages have several benefits over procedural approaches [96]. However, they are not as expressive as procedural languages. In the case of XML based languages, this lack of expressiveness can be compensated with proper extensions.

The aim of this thesis is to evaluate an XML based multimedia language, i.e., SMIL, to study its applicability to describe rich multimedia presentations. This aim can be subdivided into the evaluation of:

- The extensibility of SMIL as an XML language
- The integration of SMIL with other XML languages

The study started with an implementation of a multimedia player for SMIL documents. The player is part of an XML browser, which enables extensibility and integrations. Then, extensions were designed and implemented for SMIL. They were 3D audio capability, integration with an XForms form language, and procedural scripting support with ECMAScript [27].

This thesis has concentrated on declarative multimedia systems from an extensibility and integration point of view. Authoring has not been considered although it is a very important aspect of successful multimedia systems. Fortunately, XML has tools to transform arbitrary authoring formats to presentation formats. This enables creation of semantically high authoring formats, while presentation formats are optimized for playback.

1.5 Organization of the Thesis

This chapter has given an overview to multimedia systems. The next chapter describes multimedia document models to give an idea of the complexity of possible multimedia documents. Chapter 3 introduces and compares XML based multimedia languages and discusses the integration and extension of these languages. Chapter 4 talks about multimedia adaptation, which is important for different devices. Chapter 5 presents the implemented SMIL player and related work. Finally, Chapter 6 concludes the thesis.

2 Multimedia Document Models

This chapter describes multimedia document models. A multimedia document model gives primitives that are used to compose multimedia documents, which are then displayed as multimedia presentations. Thus, a multimedia document model shapes the requirements for a multimedia player. It also specifies extension and integration possibilities.

Being such a young research area, the terminology for multimedia varies. Therefore, the definitions for *media object*, *multimedia document*, *multimedia document model*, and *multimedia presentation* are given. Figure 3 depicts the relationships of these definitions.

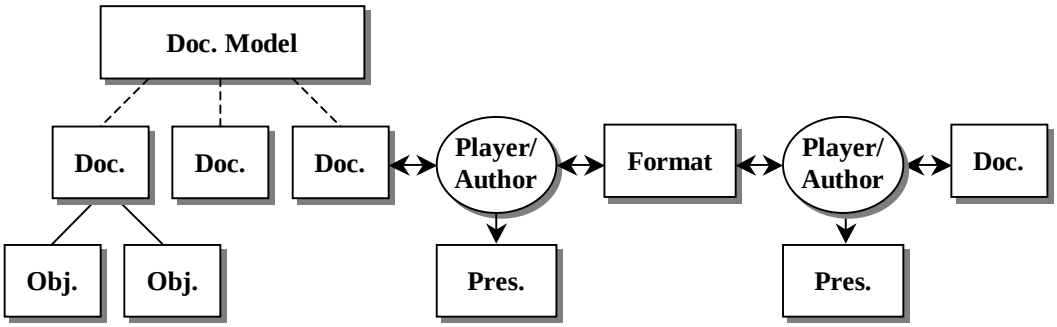

Figure 3: A multimedia document model defines semantics for multimedia documents. Each document combines a set of media objects. An authoring tool creates a multimedia document and saves it in a multimedia format for content exchange. A multimedia player plays out a multimedia document as a multimedia presentation.

A media object is a unit of text, image, graphics, video, audio, or any other data. There are various types of media [62]. *Discrete media* is a single element, while *continuous media* is a time-ordered sequence of discrete media after digitization. Continuous media can be *pre-stored media* (i.e., presentational media-on-demand), *live broadcast media*, or *live interactive media* (e.g., video-conferencing) [78]. The transmission mechanism can be *one-to-one* (i.e., unicast) or *one-to-many* (i.e., multicast).

A multimedia document is defined by Jourdan et al. [57] as "a set of objects from different media (e.g., text, image, video, audio) that are spatially and temporally organized and on which a navigational structure can be set." Thus, a multimedia document is considered to include navigational structure, although Hardman et al. [37] refer to these as hypermedia documents.

A multimedia document model provides primitives to describe multimedia documents. It defines the expressiveness power of multimedia documents with its *spatial*, *temporal*, *interaction*, and *adaptation* model [8]. Multimedia documents are instances of multimedia document models, just as text documents are instances of document style instructions. *A multimedia language* is defined with a multimedia document model.

A multimedia document format is a serialization of a multimedia document for data exchange between applications [8].

A multimedia presentation is the runtime representation of a multimedia document. It includes spatio-temporal rendering with user interaction and adaptation [8]. A multimedia player renders the presentation.

A multimedia document model is composed of *spatial*, *temporal*, *interaction*, *adaptation*, and *metadata models*. These are described next.

2.1 Spatial Models

Various spatial models have evolved since the first drawings were scribbled on a wall of a cave thousands of years ago. Once images were laid out in some specific order, text was born. Text can still be enriched with images, positioned at various locations relative to the text. For about a century, audio and video have been laid out in conjunction with text and images, first in non-interactive movies, and recently in computer based interactive multimedia presentations.

In multimedia documents, spatial models are used to specify where media objects are positioned. Boll [8] has categorized these spatial models as *absolute positioning*, *directional relations*, and *topological relations*. In addition to these, *text flows* [88] are used in conventional text documents and sometimes even in multimedia documents. Figure 4 gives an illustration of these spatial models.

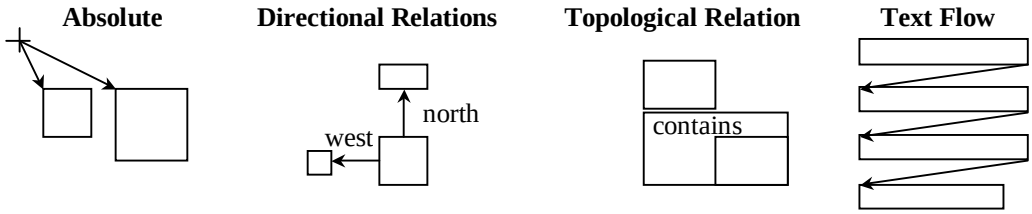

Figure 4: Various spatial models [8][88].

Absolute positioning defines a position for a media object as coordinates relative to an origin. In the simplest case, the origin is the top-left corner of the application window. A commonly used unit of measure is a pixel, however, millimeters, inches and other units are also applicable. Sometimes, the ordering of the areas is defined in case they overlap. This is often named z-indexing. *Relative positioning*, often used in multimedia languages, is a type of absolute positioning. It has several meanings depending on the language.

Directional relations define order in space [79]. These are used to compose presentations, where shape, size, and distance are irrelevant. Good examples are subway maps, where the order and direction (e.g., one station is north of another

station) are important, but the shape of the station is irrelevant, denoted just with a point. There are nine primitive relations: north-east, north, north-west, east, west, south-east, south, south-west, and the same position. In addition to these, the direction can be described more precisely, e.g., strong-north, and strong-bounded-north, which describe overlapping of objects. With these, a total of 169 different directional relations are possible for two-dimensional areas.

Topological relations describe a media object's position relative to other media objects [28]. Eight topological spatial relations can be identified for continuous areas. These are disjoint, touch, equals, inside of, covered by, contains, covers, and overlap. Topological relations are usual in GIS systems, where users ask queries about spatial relations between objects.

Text flow defines the positions of media objects as a one-dimensional flow, which is displayed in a two-dimensional area by using wraparound or scrolling [88]. Typically, *text flow* is used for text, but also images and other media objects can be placed in a flow. However, text flow is rarely used with multimedia documents.

In addition to the spatialization of visual objects, audio-spatialization is possible. Audio can be rendered into a given position by any of the above models. Audio is usually modeled as a point, restricting the use of directional and topological relations (e.g., a point cannot be inside a point). In reality, all audio sources have a width and height and thus all spatial models are applicable. An example of audio rendering with absolute positioning and text flow is given in [P3].

Design effects (i.e., spatial operations) alter the style and look of a media object. These operations include mosaic effects, transformations, mixing, filtering, coloring, and altering volume or pitch [69]. However, these do not affect the position of a media object and thus are not part of a spatial model.

Spatial information can be separated from content with a stylesheet [58][9]. This is achieved by having the content in a lexically ordered structure, while a stylesheet contains information about the spatial model and design effects, such as text position and font colors.

2.2 Temporal Models

Timing, *scheduling*, or *orchestration* of a multimedia document specifies the temporal order of media objects. It also describes the temporal relationships between media objects, which allows the starting and ending of objects based on other objects.

Synchronization defines the occurrence of simultaneous events [62]. Synchronization has several levels; some levels apply to the object level structures, i.e., synchronizing separate video and audio streams, while some apply to physical data streams, to synchronize audio and video that are embedded in the same stream [72]. The temporal model considers the former.

A time can be *resolved (determinate)* or *unresolved (indeterminate)* [96], as specified in the SMIL specification [5]. A resolved time has a calculated time relative to the global presentation time. An unresolved time exists logically in the time model, even though it cannot yet be calculated, and thus is not part of

scheduling. An unresolved time can become resolved by some activation at runtime. Activation may be event-based, link-based, or externally based.

A time can be *definite* or *indefinite* [5]. Definite times are typical finite time values, such as "6 seconds", while an indefinite time denotes an unspecified time. An indefinite time is not an unresolved time. It is resolved, but not specified. Furthermore, an indefinite time is not the same as infinite time. Infinite time happens after forever (in practice, never), while indefinite time occurs at an unspecified time, which can become implicitly specified by a link traversal or other external activation method.

Boll [8] has categorized timing in multimedia documents into *timepoint-based* models, *event-based* models, *interval-based* models, and *script-based* models. These are presented here with a deeper analysis.

Timepoint-based models define the start and end times for timed elements with time instants, which are zero-length moments in time. Time instants have a specific known value relative to the global presentation time and predictable temporal relationships. This means that all temporal relations are resolved and can be calculated before the presentation is played. This kind of document is linear and has a single presentation form [39][38]. Timepoint-based models can only have before (<), after (>), and equal (=) relationships between time instants [42]. These models are useful for synchronizing multiple elements within a single linear presentation, a timeline. However, non-predictable changes, such as user interaction that lead to non-linear presentation, cannot usually be described with these models [37].

Timepoint-based presentations can vary by jumping to another section of a timeline. The document is usually sampled over time, rendering frames at a constant rate. Continuous media can be played out, but the hard synchronization of media objects and Internet based unreliable media delivery is not well handled due to predefined fixed time instants. Generally, these models cannot handle media of unknown duration, such as streamed continuous media [95].

This model was used in the first CD-ROMs, and in audio and video editing. For instance, the Society of Motion Picture and Television Engineers (SMPTE) use time instants to synchronize audio and video by assigning time instants for tracks in a motion picture [63]. Futhermore, Athena Muse [43] uses a similar principle.

Interval-based models define time intervals for media objects. An interval is made of two time instants, and thus this model has similar properties to the timepoint-based model [36]. However, intervals have a higher semantic level, as Allen [3] has pointed out by defining thirteen relationships for them. The relationships are equals, before, meets, overlaps, during, starts, and finishes. All but equals have an inverse relation, which brings the total number of relationships to thirteen. In fact, these relationships correspond to one-dimensional directional relationships [79], which are used in the spatial directional relations model (cf. Section 2.1).

End-point exclusive timing has intervals, where the start time is included but the end time is excluded [96]. This arithmetic follows the commonly used concept of intervals. For instance, a minute is an interval of 60 seconds, but our clocks only display seconds from 0 to 59. End-point exclusive intervals are common in interval-based systems.

Interval-based models enable the use of parallel and sequential relations, which are actually *equals* and *meets* interval relationships, respectively. It is

possible to specify most temporal interactions with these two relations. This has been used by Postel et al. [83] and Duda [25], and in SMIL [5], and ZYX [8]. Parallel and sequential relations create a tree structure, i.e., *hierarchical time*, where the local time of a child object is a function of the respective local time of the parent object [94]. If all times are resolved (determinate), then all local times can be converted to a global time, which is the time in the root object of the tree. *Relative timing* [94] means that the time of an object in the tree is dependent on another object, either because of parallel or sequential relations, or because of *time-arcs*, which define synchronizational relationships between the objects. A time-arc specifies that a media object starts or ends when another object anywhere in the tree starts or ends. A *time graph* [96] is a time structure, where objects use relative timing, and thus break the tree structure, e.g., by having time-arcs across branches of a hierarchical tree.

Interval-based models can include *time transformations* [94], where the pace of time can be accelerated, decelerated, scaled, translated, or reversed. These require a concept of duration, and thus are not possible in other temporal models.

Like timepoint-models, interval-based models cannot represent unpredictable timing, such as user interactions, media of unknown duration, or unreliable media delivery. Therefore, several proposals to enhance the model have been published. Possible solutions are the use of parallel and sequential relations [25] or open time intervals [106]. Layaïda et al. [66] propose flexible durations and resynchronization after delivery delays. Other attempts are *constraint-based systems*, including MET++ [1] and Madeus [56][57]. A constraint-based system has relations that should hold [93]. A non-constrained model processes media objects without any impact from other objects, while a constrained model processes objects so that they influence each other. All these can be considered to be temporal adaptation models, further discussed in Section 2.4.

Event-based models start and end media objects according to events that occur during multimedia presentation runtime. The time for an event is not determined (i.e., is unresolved) until the event occurs. Events are generated by a user interface (e.g., mouse and keyboard events), a timing engine (e.g., repeat, pause, resume events), a presentation engine (e.g., media onload and document mutation events), or are author-defined (e.g., streamed as an event track). Thus, the final presentation is not known until it is played making each presentation potentially different [94].

Purely event-based models are good for interaction, making them a good choice for highly interactive content. These models can handle media of unknown duration and unreliable delivery. However, there generally is no synchronization between objects. Simple implementations often suffer from delay of event propagation, which can accumulate into a notable skew in long presentations. This can be overcome with a simple synchronization with a master clock or by marking events with a time mark (e.g., VRML 97 uses this approach) [95]. Since event-based timing models lack a timeline, time transformations are impossible [94]. Event-based models traditionally cause a piece of script to be executed at the event observation time. However, declarative approaches have emerged, such as SMIL [5].

Script-based models combine scheduling and synchronization into procedural languages. For instance, RT-LOTOS [19] is a temporal extension of the standard

Formal Description Technique LOTOS. It adds features for delays, latency, temporal action observation, and time variables.

These four basic models can be combined. Timepoint-based models and interval-based models are good for storytelling, but lack user interaction. Event-based models are good for user interaction, but have poor scheduling facilities. A combination of interval-based and event-based models provide a powerful approach to the scheduling multimedia presentations [37][95][35]. This approach has been adopted in SMIL 2.0.

Often, timing information appears with the content, usually as attributes [76]. However, this approach is not very beneficial when timing does not follow the content structure. Furthermore, timing of layout is not easy and time templates are impossible. Timesheets declare timing in a separate section, solving the problem of combining content and timing. In addition, timesheets can assign relations between elements rather than properties to them [58][96]. Thus, timesheets would be an ideal solution for scheduling XML languages, as temporal information could be attached to any arbitrary XML language. Unfortunately, timesheets are not currently part of any standard.

2.3 Interaction Models

Boll [8] has categorized interaction into three basic types, *navigational interactions*, *design interactions*, and *movie interactions*.

Navigational interactions affect the spatial or temporal layout of the document. These offer the user a choice to decide which presentation path is to be followed. *Linking from media subparts* enable links from temporal or spatial parts of a media object, while *linking to subparts* allows the specification of link anchors at temporal or spatial parts of a media object [8].

Design interactions influence the design effects of a media object, e.g., color, video transitions, or audio volume [8].

Movie interactions allow the user to control the global time of a presentation with actions similar to a VCR, e.g., play, stop, fast-forward, reverse, and bookmarks. This interaction method is offered by the presentation engine, while the two other, navigational and design interactions, are defined by the author in a multimedia document [8]. In addition to the temporal control mentioned by Boll, spatial controls are offered by most players to affect the layout. These are typically scroll or scaling facilities to view large documents.

Navigational and design interactions can be activated in various ways, depending on the underlying temporal model. Timepoint-based and interval-based models usually include link activation, which allows traversal in a presentation time model. Event-based temporal models use events to activate interaction. A hybrid interval-based and event-based model has link and event activation [37]. Both, link and event activation is used in SMIL.

2.4 Adaptation and Accessibility

Adaptation is required to play a multimedia presentation in various environments. Adaptation should consider the technical infrastructure, e.g., network bandwidth, screen-size, and input mechanisms (cf. Section 1.3). It should also consider user preferences, e.g., language settings, accessibility to the blind and deaf, knowledge level, and the user's interests.

Adaptation can offer alternative spatial, temporal, or design effects for a presentation by reducing the size, expanding the time, or modifying the font, respectively. Adaptation can add or remove media objects all together. Boll [8] has categorized adaptation to the *extent* to which adaptability is modeled and *when* the adaptation is exploited.

Extent of adaptability considers adaptation to technical infrastructure and user preferences. These properties select various presentation paths.

Static or dynamic adaptability considers, whether the adaptation alternatives are known at the authoring time, or whether the adaptation alternatives are resolved during the presentation playback [8]. An example of the former case is hard-coding two alternate sizes of a presentation for two different screen sizes. The latter resolves presentation alternatives on the fly, e.g., with a query to a database to retrieve a correct document fragment. The fragment can be modified after the document has been authored. In addition, constraint-based systems perform dynamic adaptation. A constraint-based system can shrink media objects to fit a screen, still maintaining the proportions of the objects. A constraint-based system can also stretch time to fill in temporal gaps.

In addition to these, adaptation can further be categorized according to the target of adaptation, e.g., spatial, temporal, interaction, and media adaptation.

Spatial adaptation is required to cope with varying spatial conditions. For instance, a constraint-based system will try to fit media objects to the constraints given by the screen size. Inserting a new object will affect the placement of all other objects [11].

Temporal adaptation is required if the user wants to stretch the timeline or if resynchronization is required. Various methods to adapt interval-based timing were discussed in Section 2.2. Furthermore, Kim et al. [60] have proposed elastic time to adapt duration of a presentation to user preferences.

Interaction adaptation is required in presentation environments with varying interaction methods, such as a mouse, a touch screen, or a keyboard. Typically, the presentation engine configures device controls properly, and the presentation author does not need to worry about it.

Media adaptation happens if the presentation engine does not understand the original media format. Usually, the presentation engine cannot perform media adaptation, except by switching between alternative media formats. Solutions are discussed in Section 4.

2.5 Metadata Models

Metadata describes content of a document. Metadata is not presented to the user, but the multimedia system uses it to search, categorize, filter, and process a

document. Metadata usually describes content properties (e.g., style, shape, and color), author, copyright, Digital Rights Management, and billing information. Simple metadata models describe metadata as property-value pairs, while more complex ones use a graph.

Resource Description Framework (RDF) [65][23] is a metadata language defined by the World Wide Web Consortium (W3C) and it is the basic building block for Semantic Web [7]. Semantic Web aims to annotate Web content so that it will be machine-processable. The RDF data model is a graph made of statements. Each statement is made of a subject, a predicate, and an object, as depicted in Figure 5. The subject and the predicate are resources referenced by a URI. The object can be a resource or a literal, which is a plain text string. RDF itself does not define any vocabularies. They can be defined with RDF Schema. Currently, some vocabularies exist, for instance, Dublin Core [41].

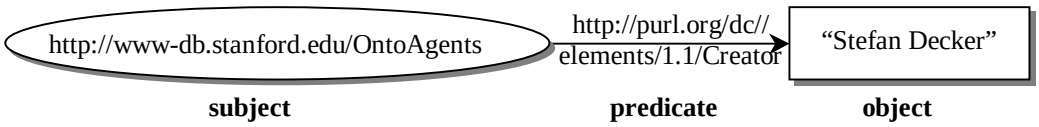

Figure 5: A simple RDF statement [23].

MPEG-7 [70] is a complete system for metadata description and delivery defined by ISO (International Standardization Organization). MPEG-7 Descriptors represent features, which describe the characteristics of data, e.g., the author, shape, or motion. Descriptors cover many application domains, but are still extensible for special purposes and future needs. Description Schemes specify the structure and semantics of the relationships between Descriptors and other Description Schemes. It is possible to relate Descriptors with spatial or temporal relations, e.g., a 'ball' *is close to* a 'player' or a 'kick' *precedes* a 'goal'. Moreover, semantic relations are possible, e.g., a 'win' is a *result of* a 'goal'. The combination of these relations is a graph, as depicted in Figure 6. To define Descriptors and Description Schemes, Description Definition Language is used. In addition to these, MPEG-7 defines several other tools. Binarization and storage tools provide a compact method to represent MPEG-7 metadata. Transport tools offer the possibility to stream metadata, for instance, along a video sequence. Synchronization tools keep multiple copies of descriptions in different physical locations consistent. In addition, tools exist for the management and protection of intellectual properties.

Figure 6: An example of a MPEG-7 metadata graph.

2.6 Summary

The five models must be combined into a single multimedia document model. Combining several models into one hierarchy, in this case into an XML tree, raises problems. Several hierarchical models cannot be represented in one hierarchy. To resolve this, the most important model is usually selected as the main hierarchy. The other models are then attached to the main hierarchy in various ways. Van Ossenbruggen et al. [77] have described three methods to do this. These are embedding new elements and attributes to the main hierarchy, adding information to the head section as a separate hierarchy, and attaching information by using external stylesheets. These are exemplified in Figure 7. Ten Kate et al. [58] also describe three ways to attach a temporal model to a spatial model. These are inline integration in the main hierarchy, attaching time to a spatial stylesheet, and using external timesheets.

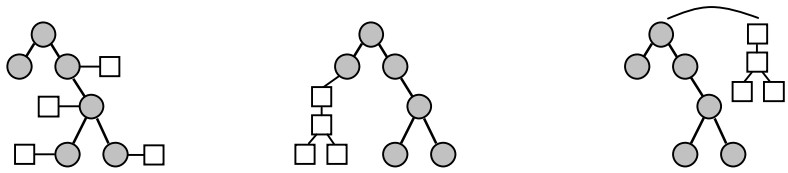

Figure 7: Combination of models by embedding attributes, adding information to a head section, and external stylesheets [77].

As an example, SMIL uses the hierarchical temporal model as the main hierarchy, and represents its hierarchical spatial model in its head section as a separate hierarchy. XHTML+SMIL uses the hierarchical spatial model as the main hierarchy, while the temporal model is embedded in the main hierarchy as a set of attributes.

Usually, interaction and adaptation models are embedded in the main hierarchy, e.g., <a> link elements in SMIL. Bouvin et al. [12] point out that this is problematic, because links cannot reside in an external document. XLink [21] is a solution by W3C for this. In addition to links in an external document, it provides bidirectional links and is more suitable for XML languages, as a link can originate from any element.

Metadata is usually added to the head section as a separate hierarchy or attached as a separate document. Alternatively, pieces of metadata can be embedded in the main hierarchy as attributes.

3 XML Based Multimedia Document Formats

A multimedia document format is required to represent multimedia documents in a form of serialized data. Serialized data, such as a text or a binary file, is needed for multimedia document exchange. An authoring tool writes data out to be read by another application, such as an authoring tool or a multimedia player. This chapter gives an overview to XML based multimedia document formats.

XML [13] is a meta-language, which describes languages. It is an application profile of Standard Generalized Markup Language (SGML) and all XML documents are conforming SGML documents. The philosophy of XML is based on separation of content from format, hierarchical data structures, embedded tagging, and user-definable structures [101]. Separation of content from format is the most important concept behind XML. This means that information should be identified on as abstract level as possible to be displayed and processed in several ways. An XML document consists of a hierarchical data structure, where a root element begins the document. There can be child elements under the root element, and any element can further have more children. Embedded tagging is used to mark up where an element begins and ends. Additional information can be provided with attributes, which are pairs of names and values, attached to a start tag. XML is not a tag set, but it defines a way to create tags. Thus, XML documents have user-defined structures with tags and attributes. Often, the user is a standardizing committee or an organization, which creates document formats for common use.

The syntax of XML can be described with either a Document Type Definition (DTD) [13] or an XML Schema [29]. An XML DTD defines the structure, the allowed tags and attributes in a document. The syntax of an XML DTD is a subset of SGML's DTD syntax. An XML Schema offers the same facility with additional constructs, such as data typing. Furthermore, its syntax is based on XML.

XML based multimedia document formats covered in this chapter are XHTML, SVG, SMIL, Extensible 3D Graphics (X3D), Extensible MPEG-4 Textual Format (XMT), MHEG-8, Madeus, and ZYX. In addition to these, HyTime is covered, as it is closely related to XML. Furthermore, Java is presented as it is currently very commonly used to author and display multimedia presentations. Flash is also mentioned as a proprietary non-XML format.

The multimedia formats have relationships depicted in Figure 8. HyTime and XML are subsets of SGML [101][90], while HTML is a DTD of SGML. SMDL is a DTD of HyTime [75] and XHTML and other XML based formats are DTDs of XML. XHTML has the same semantics as HTML, and is therefore presented close to it.

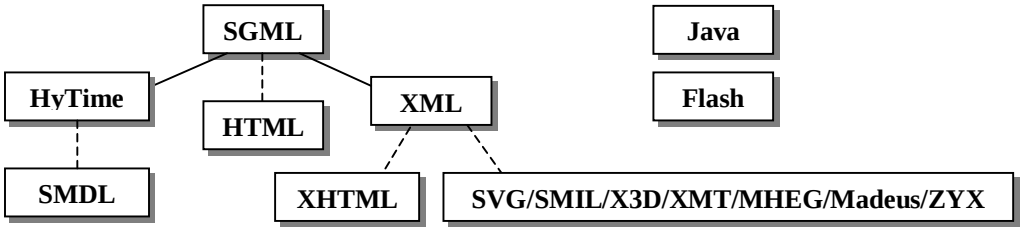

Figure 8: SGML and XML related multimedia formats and their relations. A solid line denotes a subset, while a dashed line means a DTD.

3.1 HyTime

Hypermedia/Time-Based Structuring Language (HyTime) is an ISO standard to represent multimedia documents in a presentation independent format. The first edition was released in 1992 [50], while the second edition followed in 1997 [52]. HyTime is a meta-language built upon SGML. It enables hyperlinking and object coordinate systems for space, time, and any other imaginable dimension. HyTime cannot be used as such to create multimedia documents, as it is a meta-language. An SGML DTD is required to define an actual document format, the same way as an XML DTD defines syntax for an XML document. For instance, Standard Music Description Language (SMDL) is a HyTime-based document format to represent music [75].

The second edition of HyTime contains facilities to transform HyTime documents into presentation formats using Document Style Semantics and Specification Language (DSSSL) [50]. DSSSL can transform any SGML document into an SGML or non-SGML format [90]. Figure 9 clarifies this. For instance, an SMDL document can be transformed into SMIL. Nowadays, DSSSL is mostly replaced by XSL Transformations (XSLT) [18]. XSLT is an XML language for transforming XML documents into other XML documents. Because XML has taken over SGML, XSLT has replaced DSSSL.

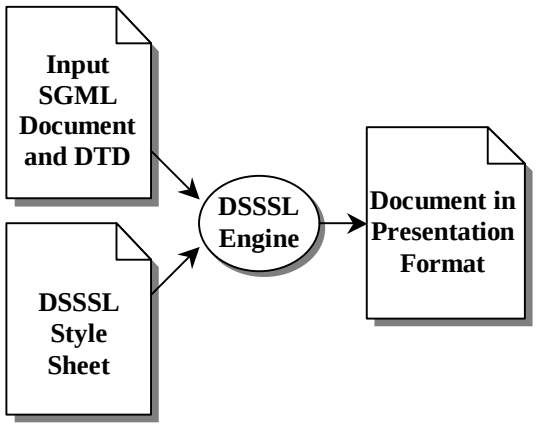

Figure 9: DSSSL Usage [90].

HyTime is a very comprehensive language allowing construction of n-dimensional multimedia document formats. However, HyTime does not provide a model for interaction. Furthermore, it does not include presentation format modeling, requiring transformation into a presentation format with DSSSL. There are not many HyTime-based document formats available. One of the first was the previously mentioned SMDL, which has not gained much popularity. On the player side, there are not that many options. HyOctane [62] can play out HyTime-based multimedia documents. By default, it plays its own DTD, but it is extensible to play other HyTime DTDs.

3.2 MHEG

Multimedia and Hypermedia Information Coding Experts Group (MHEG), a working group in one of the International Organization for Standardization (ISO) subcommittees, has defined a set of multimedia interchange formats, which have been named after the working group itself [26][72][73]. MHEG-1 is the base standard finished in 1996. MHEG-3 extends it with a script extension and MHEG-4 includes registration procedure for identifiers. MHEG-5 is aimed for terminals with restricted resources and thus is a sub-profile of MHEG-1. MHEG-6 extends MHEG-5 with a Java virtual machine to run sophisticated applications [53]. MHEG-7 addresses the conformance and interoperability of MHEG-5 engines and applications. There are several delivery formats for MHEG. Abstract Syntax Notation 1 (ASN.1), a binary format standardized by ISO, is the preferred encoding format for the early MHEG versions. However, a textual format is also available and MHEG-8 provides an XML based encoding for MHEG-5.

MHEG was developed in the 1990's to provide a practical multimedia standard, as such did not yet exist. It was aimed for video-on-demand, interactive CD-ROMs, info kiosks, and interactive TVs. In the United Kingdom, MHEG-5 has been chosen as the base format for interactive television [49]. Despite wide interest in the beginning, MHEG does not receive much attention anymore as new multimedia standards have evolved. One of the drawbacks in the design of MHEG is that the standard only provides an event-based temporal model, disregarding synchronization of media objects. It is good for multimedia document exchange, but not for authoring as it does not have a high semantic level [8].

3.3 XHTML

XHTML [80] is an XML based version of the famous HTML language. XHTML 1.0 was released in 2001 and XHTML 2.0 is under way by W3C. XHTML 1.0 has the same tag set and semantics as HTML 4.01. Thus, it is not really a multimedia language, but still it provides limited interaction and timing with scripting functionality. It is also the basis for the XHTML+SMIL profile (cf. Section 3.5).

XHTML uses absolute and flow positioning. Relative positioning, a variation of absolute positioning, can be used to move objects relative to their original position. Often, a CSS stylesheet defines the layout and style for an XHTML document. A stylesheet is a set of rules that tell how to display a document. It can be defined within an XHTML document or be an external document. A stylesheet can modify the margins, fonts, colors, and borders of a document. Stylesheets can refer to other stylesheets, thus the name Cascading Style Sheets.

In XHTML, interaction happens via links and events. Adaptation is possible for different media types (e.g., for screen, tv, handheld, and print), which each can have its own CSS styles.

XHTML has been split into modules, which each describe a set of functionality. The modules can be grouped to form profiles, which describe full languages. This way different application domains can define a language that best suits them. The XHTML Basic profile is a combination of basic modules including style sheets, forms, and tables. The XHTML Basic profile is already implemented in the latest mobile phones and its intention is to replace the Wireless Markup Language (WML), which is the traditional markup language used in the phones. Although the latest phones support XHTML Basic, the high number of legacy WML browsers will still keep most of the content as WML.

3.4 SVG

Scalable Vector Graphics (SVG) [30] is an XML based vector graphics format specified by W3C in 2001. It uses absolute positioning and interval- and event-based temporal models. The temporal model actually reuses functionality from SMIL. Interaction happens via links or events. Spatial adaptation in SVG is inherent – vector graphics is scalable to any size. For adaptation, SVG also includes a switch functionality, which can be used to select alternative parts of the document to be displayed based on the screen size, bit rate, and user selected language. To enhance SVG, Badros et al. [6] have suggested an extension to SVG for spatial constraints. These make the spatial adaptation even more flexible.

W3C is currently specifying SVG 1.2, which would include a better temporal model and a text flow based spatial model. Integration with other XML languages, such as XForms, has also been planned.

Several SVG players are available for PCs, PDAs, and mobile phones. At its current state, most authoring tools can export and import SVG graphics. However, the use of SVG for Web graphics has been overshadowed by Macromedia's Flash format.

3.5 SMIL

Synchronized Multimedia Integration Language (SMIL) was released in 1998 by the Synchronized Multimedia Working Group (SYMM) at the W3C [47]. Shortly after that, the SMIL standardization was continued, and in August 2001,

the SMIL 2.0 recommendation was released. SMIL is a multimedia language specifically designed for the Web. It is based on XML and fully declarative, removing a need for scripting [87].

SMIL 2.0 has been split into modules, which each declare a set of functionality. A profile combines these modules into a full language. There are several profiles for SMIL. SMIL 2.0 Language Profile is the main profile and includes almost all SMIL modules. SMIL 2.0 Basic profile is a limited version of it, while the XHTML+SMIL profile extends XHTML mainly with SMIL's timing modules [16]. Modules can also be reused in other languages. The SMIL Animation module is part of SVG.

SMIL uses absolute positioning with hierarchical positioning. Hierarchical positioning defines regions inside parent regions, which form a tree structure. In SMIL, relative positioning defines the media position relative to the parent region's dimensions. In some cases, absolute positioning based layout is not sufficient. For instance, XForms repeat feature requires a text flow. Thus, a text flow model has been experimented with in [P2], following the idea presented by Hoschka et al. [46] [48].

The main benefits of SMIL are ease of learning and adaptability. SMIL has been widely adopted in multimedia players [15], such as RealNetworks's RealOne player [84] and Apple's QuickTime [4]. It has been selected by 3GPP as the format for MMS in mobile phones. Millions of people already have a SMIL based MMS authoring tool and player in their mobile phones.

3.6 X3D

The Extensible 3D Graphics (X3D) [109] language is a specification currently being worked on by the Web3D Consortium. It is an XML based version of VRML 97 (Virtual Reality Markup Language) [55]. X3D uses 3D absolute positioning, similarly to VRML. It uses timepoint- and event-based timing, where events carry time stamps to avoid event propagation delay. X3D includes adaptation with Level-Of-Detail alternatives, which enable objects of varying complexity. Moreover, scripts can be used to adapt the document. Interaction happens via events and links. Events can cause objects to be started or stopped. Links can change the viewpoint, i.e., the location where the user is looking at the world. No temporal links are offered.

It remains to be seen how popular X3D will become. After all, it is only an XML based version of the VRML97 language, which was not very popular. MPEG-4 has functionality similar to X3D, and thus may become the preferred format.

3.7 MPEG-4 and XMT

MPEG-4 [54] is a comprehensive multimedia toolkit for multimedia exchange and presentation [82]. The basic idea is to split a scene into objects and to describe relations between these objects. MPEG-4 includes tools to encode

individual image, audio, and video objects. It also defines a multimedia description language, Binary Format for Scenes (BIFS), to compose a whole presentation from media objects. BIFS is a binary format based on VRML 97.

Extensible MPEG-4 Textual Format (XMT) is an XML-based version of the BIFS to ease authoring of MPEG-4 [59]. It has a two-tier architecture made of XMT-Ω and XMT-A. XMT-Ω has a high semantic level and is compatible with SMIL, while XMT-A has a one-to-one mapping to BIFS and is compatible with X3D. These relations can be seen in Figure 10. The idea is that XMT-Ω can be transformed into XMT-A, which can further be transformed into BIFS. This enables conversion of SMIL and X3D into BIFS. Because XMT-Ω has a higher semantic level than XMT-A, XMT-A cannot be transformed back into XMT-Ω. XMT-A is similar to X3D and uses the same spatial, temporal, interaction, and adaptation models. XMT-Ω is similar to SMIL.

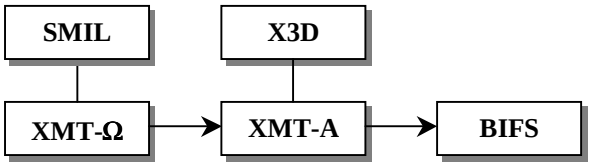

Figure 10: Relations between XMT-Ω, XMT-A, BIFS, SMIL, and X3D.

Being a new standard, MPEG-4 has not yet been adopted widely. Still, some tools of MPEG-4, such as Basic profile level 0 of video compression, are already available in mobile phones.

3.8 Madeus

Madeus [56] is a research multimedia system developed in the OPERA project at Institut National de Recherche en Informatique et en Automatique (INRIA), a French research institute. It is an authoring environment for interactive multimedia, but also includes a presentation tool. Documents for Madeus are described with a declarative multimedia document format based on XML. Madeus researches the creation and authoring process of multimedia and bases its multimedia format on authoring requirements. The document format is made of logical, spatial, temporal, and navigational parts. The spatial and temporal models rely in constraints to ease the authoring process. Madeus is not in wide public use, but has given a lot of feedback to the existing multimedia formats.

3.9 ZYX

ZYX [8] is a research multimedia document model designed at Vienna University of Technology. It emphasizes reuse, adaptation, and presentation-neutrality. It is possible to reuse media objects, document fragments, and entire

documents. Adaptation can be applied dynamically at the presentation time, giving a strong support for personalization. Presentation-neutrality is achieved because ZyX is a semantically high language. It can be transformed into presentation languages, such as MHEG-5, SMIL, or HTML, for presentation. With these features, ZyX provides currently the best facilities for reuse and adaptation. However, Boll [8] considers the future multimedia document models to be more abstract, suited for (semi-)automatically generated multimedia presentations. These models will most likely be template-based, created by domain specific authoring tools. They will also be separate from presentation models.

3.10 Java

Java, developed by Sun Microsystems, can be used to present multimedia presentations in Web browsers in the form of applets. An applet is a piece of code that has access to standard Java 2 Standard Edition (J2SE) classes enabling creation of time-based applications. Java Media Framework (JMF) is an optional package, which eases playback of audio and video [32]. In small clients, such as PDAs and mobile phones, MIDlets can be downloaded and executed. They use Java 2 Micro Edition (J2ME) classes and an optional Mobile Multimedia API (MMAPI), which are more restrictive than J2SE classes. Finally, DVB (Digital Video Broadcasting) based digital television receivers can download Xlets, which have access to core classes of J2SE, but the user interface is defined with HAVi (Home Audio/Video Interoperability) classes designed for the television environment. The benefit of Java is that it follows the "write-once-run-anywhere" concept, and due to its procedural nature almost any kind of behavior can be programmed.

However, procedural languages have several drawbacks compared to declarative languages [96]. One of the drawbacks of Java is the complexity of programming – novice multimedia authors do not necessarily possess such abilities. Furthermore, authoring tools have a hard time interpreting generated code, especially if it has been generated by another authoring tool. Declarative languages tend to be more secure, even though Java exposes all programs to its sand-box model. The browsers or players playing declarative languages can be optimized for the given environment, while the usually ad-hoc created Java programs are hard to optimize, especially during run-time. Declarative languages can be generated automatically with the existing tools, such as XSLT [18]. Finally, Semantic Web [7], introduced by Tim Berners-Lee, requires declarative languages that can be interpreted by search engines, information brokers, and information filters.

Nevertheless, Java is a competitive option to declarative multimedia languages due to its availability and expressive power. In addition, it has a capability to read in XML and process it. Therefore, arbitrary XML encoded presentations are possible.

Concerning various models, Java provides several ready-made layout models. Absolute positioning is the default behavior of the layout engine. Overlay and card layouts add z-indexing to it. Flow layout provides a text flow style of

layout. Border layout provides limited directional relations, i.e., north, west, east, south, and center positioning. Grid, gridbag, and box layout are types of constrained absolute positioning. The constraints control the width and height of the cells in a grid. In addition to these, it is possible to implement custom layout models. Temporal models must be explicitly programmed, while interaction is provided with user interface events. Adaptation is possible by reading the system properties. From a semantic point of view, metadata is hard to describe in a way that is useful for other tools.

3.11 Macromedia Flash

Flash [70], developed by Macromedia, is an animation software for the Web. It uses a binary format, which is not XML based. The format defines shapes, frames in time, animations, and actions. It has a very fast and well-optimized animation player, which runs on Windows, Macintosh, and Linux computers. Flash animations are very similar to SVG, as both are based on vector graphics. Concolato et al. [19] have shown that a Flash animation is half the size of a compressed SVG animation. It also starts playback in less time because the animation is loaded progressively. Therefore, Flash is very well suited for the Web. The latest version enables extending the core functionality with Featured Third-Party Extensions. The drawback of the format is that it is not XML, which means that it cannot be integrated with other XML formats.

Over 90% of Internet-enabled desktops are capable of playing Flash, compared to 18% for SVG. Therefore, Flash is currently the dominant animation and vector graphics format in the Web. Furthermore, recent mobile phones support the Flash format. Thus, the format may become a rival for SMIL in this area.

3.12 Review of Formats

The discussed multimedia document formats have been summarized in Table 3, which is based on the evaluation made by Boll [8]. The table has been supplemented with an evaluation of SVG, X3D, MPEG-4, XMT-Ω, Java, and Flash. The table does not include the metadata model. Usually, simple property-value pairs, RDF, or MPEG-7 can be used with the XML formats.

Format	Temporal Model				Spatial Model			Interaction Model					Adaptation			
	Point-based	Event-based	Interval-based	Script	Absolute	Flow	Spatial Relationships	From Spatial Subparts	From Temporal Subparts	To Spatial Subparts	To Temporal Subparts	Design Interaction	Technical	User	Static	Dynamic
XHTML 1.0				X	X	X		X				(X)	X	X	X	
XHTML+SMIL		X	X	X	X	X		X	X		X		X		X	
SMIL 1.0			X		X			X	X		X		X		X	
SMIL 2.0		X	X		X			X	X		X	(X)	X	X	X	
MHEG-5/6		X		X	X			X	X		X	X	X	X	X	X
HyTime	X				X											
Madeus			X		X		(X)	(X)	(X)	X	X	X				
ZYX			X		X			X	X	X	X		X	X	X	X
SVG 1.1		X	X	X	X			X	X		X	(X)	X	X	X	
X3D	X	X			X			X	X	X						
MPEG-4	X	X			X			X	X	X			X		X	
XMT-Ω		X	X		X			X	X	X		(X)	X	X	X	
Java		X		X	X	X	X						X	X	X	X
Flash	X	X			X										X	

Table 3: Multimedia formats compared [8]. The original table has been extended with SVG, X3D, MPEG-4, XMT-Ω, Java, and Flash.

Boll [8] has evaluated the semantic level and functionality of various multimedia languages, as depicted in Figure 11. A language with a high semantic level describes the meaning of the presentation, while a lower semantic level describes the presentation. High multimedia functionality provides a large set of primitives to model features, while lower functionality gives less expressive power. Again, the figure has been supplemented with an evaluation of SVG, X3D, MPEG-4, Java, and Flash.

As the table depicts, SVG is similar to SMIL 2.0. However, it has more drawing primitives with less temporal functionality. Thus, it has more functionality, but less semantics than SMIL 2.0.

MPEG-4 has only timepoint- and event-based timing, but a rich set of graphics primitives. Thus, it has a low semantic level and a high functionality level. The textual format XMT-Ω is based on SMIL 2.0, but has higher functionality with additional graphics primitives.

X3D is compatible with MPEG-4 and is very similar to it. However, MPEG-4 offers more functionality, as it also describes streaming functionality.

Java is a programming language and has an extremely low semantic level compared to the others. However, it has the most functionality, which is only

restricted by the Java APIs. Flash is a declarative language to describe animations. It is similar to SVG, but has a lower semantic level because it has been optimized for fast rendering, not for authoring.

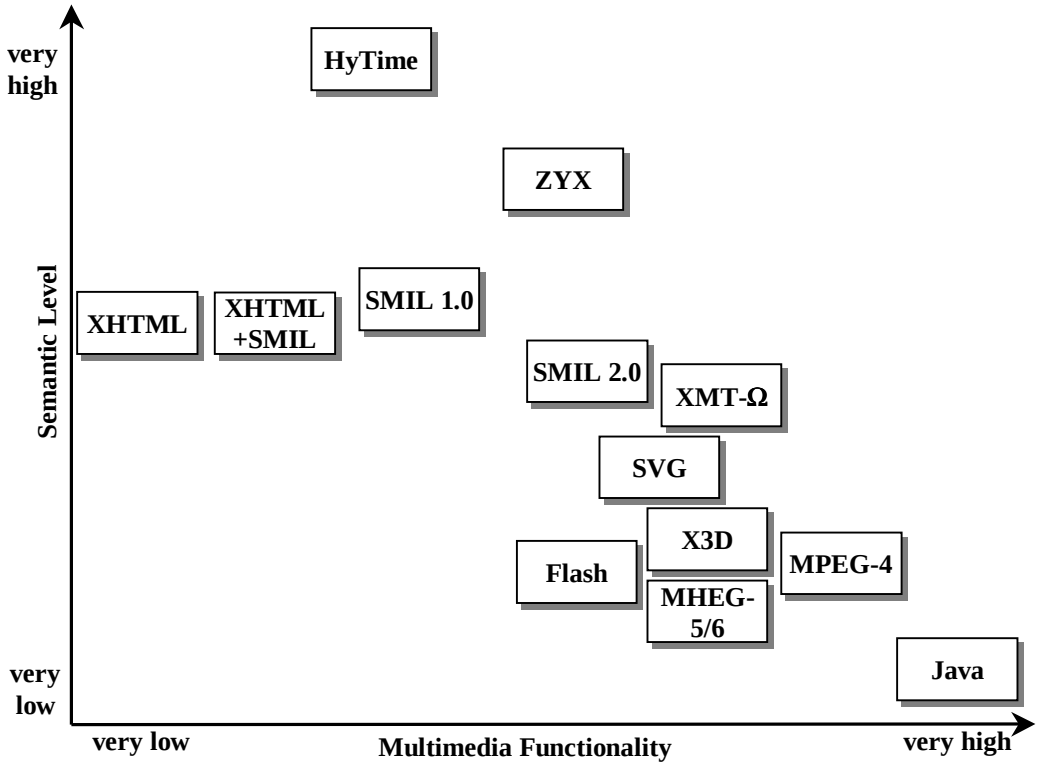

Figure 11: Multimedia functionality and semantic level of different formats. Adopted from Boll [8], supplemented with SVG, X3D, MPEG-4, XMT-Ω, Java, and Flash.

3.13 Integration and Extension of Formats

XML is extensible and has methods to integrate XML languages [14]. This kind of hybrid document contains elements from two or more languages. To avoid misinterpretation of elements with the same name, the languages are identified with a namespace. A namespace is a collection of names, identified uniquely by a URI reference. A qualified name is made of a namespace prefix and a local part. Figure 12 depicts an example of a SMIL document with XForms elements. SMIL elements use a default namespace and thus do not have any prefixes. They are identified by the first row and URI "`http://www.w3.org/2001/SMIL20/Language`". XForms elements have qualified names with the `xfm` prefix and are identified by URI "`http://www.w3.org/2002/xforms`".

```
<smil xmlns="http://www.w3.org/2001/SMIL20/Language"
         xmlns:xfm="http://www.w3.org/2002/xforms">
  <head>
    <xfm:model id="form1">
      <xfm:submission id="s1" method="post"
                                        action="http://..."/>
      <xfm:schema/>
      <xfm:instance id="i1" xmlns="">
        <purchaseOrder>
          <name>Alice Smith</name>
        </purchaseOrder>
      </xfm:instance>
    </xfm:model>
    <layout>
     ...
    </layout>
  </head>
  <body>
    <text region="r1" src="data:,XForms in SMIL" dur="2s"/>
    <xfm:textarea ref="/purchaseOrder/name" region="r2"
                                                dur="2s"/>
    <xfm:submit name="Submit" submission="s1" region="r3"
                                                dur="2s">
      <xfm:label>Submit</xfm:label>
    </xfm:submit>
  </body>
</smil>
```

Figure 12: An example of a SMIL document with XForms.

Integration of XML based multimedia formats is not always straightforward, due to their different document models. This thesis has studied a number of integrations. All of them extend the SMIL language.

[P2] adds Web forms to SMIL. Web forms are one of the key features in the Web, as they offer a way to interact with a server. The XForms language was used in the integration. Figure 12 depicts an example of a SMIL document with XForms. The example uses absolute positioning to place the form elements. The article also discusses issues with the XForms repeat feature, which assumes a text flow spatial model. However, SMIL provides only absolute positioning. As a solution in [P2], a text flow based CSS layout has been experimented with. As mentioned in Section 2.6, combining two hierarchical models can be problematic. The result in [P2] was a hierarchical spatial model (i.e., CSS) in the same structure as the hierarchical temporal model, i.e., in the body section of a SMIL document. When the CSS layout was added, it had to follow the structure of the temporal model. This made authoring of documents more difficult, as layout modifications altered time, and vice versa.

[P3] introduces an approach to include 3D audio in SMIL. First, the SMIL spatial model, which is two dimensional, is extended with an extra dimension to describe a 3D space. However, it is also possible to simply use the SMIL layout attributes without any extensions. In that case, elements are positioned on a two dimensional surface. New audio elements and a listening point are positioned in

the given space. The extension has been designed to be independent from spatial and temporal models. It is also modular. Therefore, it is possible to integrate the 3D audio module with other XML languages, as well. As an experiment, it has been integrated with XHTML and SVG. Figure 13 illustrates an XHTML document with the 3D audio extension. When the user scrolls the document horizontally, the audio sources move along with the images. Moreover, audio sources are attenuated so that those that are outside the screen are not audible.

Figure 13: Screenshot of an XHTML document with spatialized audio sources. It is possible to scroll the document to see the images and hear the sounds on the left and right side.

[P4] introduces dynamic SMIL and a player to render SMIL documents that are changed during run-time. This enables scripting. The rationale has been Web applications, which require more control over the presentation than offered by the standard declarative SMIL. Scripting is achieved with the help of XML Events and ECMAScript. XML Events introduces an element to bind a document event to a script element, which contains a piece of code written in ECMAScript language [27]. This enables execution of scripts when a SMIL event (e.g., beginEvent, endEvent) or a user-initiated event (e.g., activateEvent, focusInEvent) is observed. Scripts can modify the presentation, and the changes are displayed immediately.

Finally, generic methods to extend SMIL have been presented in [P1]. The methods include ways to attach new input sources and output capabilities to SMIL along with extended internal logic, i.e., scripting or other presentation control languages. The article introduces ways to listen to and dispatch events, create timed actions, display new visual components, animate attributes, create new media players, and attach new media parameters to media elements. All of these are independent of the used spatial model. For instance, visual components can be used with absolute positioning and text flow based models. All except timed actions and animated attributes are also independent of the temporal model. Thus, the extensions are mostly generic.

All of the proposed integrations and extensions were implemented with the help of the framework described in [P8].

4 Adaptation

Adaptation is required because multimedia clients have different properties and users have various preferences. The screen size, number of colors, amount of memory, network bandwidth, network connection persistence, input devices, CPU speed, and available media formats are examples of capabilities, which vary from one client to another. A few of them are depicted in Table 4. Used language, font size, and window size are good examples of user preferences. A multimedia document model may take these into account. In that case, a multimedia player at a client will perform the adaptation. In the Web environment, adaptation can occur at a server or proxy.

Property	**Standard PC**	**PDA (Compaq iPaq)**	**Mobile Phone (Nokia 3650)**
Screen size (pixels)	1024x768	240x320	176x208
Colors	16.7 million	65536	4096
Formats	Any	HTML	WML, XHTML
Input method	keyboard, mouse	touch screen	keypad
Network connection	LAN	20 kbps	20 kbps

Table 4: Device property examples.

As mentioned in Section 2.4, adaptation can concern *spatial*, *temporal*, *interaction models*, and *media*. Spatial, temporal, and interaction model adaptations are applied using languages designed for structure modifications, such as XSLT. Media adaptation requires transcoding.

XSL [2] is the style language of XML. It is made of three parts, XSL Transformations (XSLT) [18], XML Path Language (XPath), and XSL Formatting Objects (XSL FO). XSLT is used to transform XML documents, XPath is used by XSLT to refer to parts of an XML document, and XSL FO describes formatting semantics. The idea of XSL is that any arbitrary XML document can be transformed into an XSL FO document with XSLT. The XSL FO document is then rendered. However, XSLT can output any XML language, and is a very useful tool to adapt XML documents. Figure 14 depicts this process, which is similar to the DSSSL processing, cf. Section 3.1.

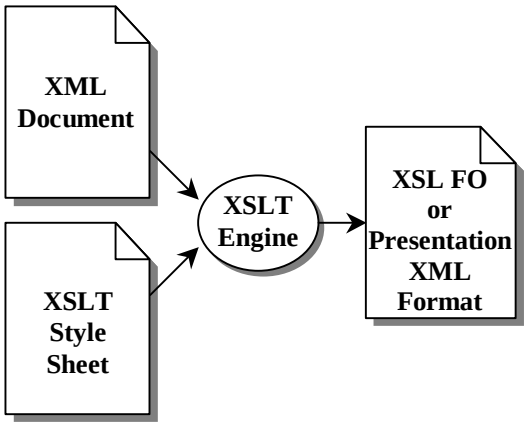

Figure 14: The use of XSLT to transform XML.

XSLT, being able to transform XML, can be used to create documents for different devices. This happens by designing a separate XSLT stylesheet for each device, keeping the source document unchanged. Vuorimaa et al. [105] have used this approach to run multimedia services on different kinds of multimedia devices. [P5] describes a use of XSLT to alter the layout of a document. Figure 15 depicts the resulting transformed document on a desktop screen and a mobile device.

Figure 15: Two versions of a transformed document [P5].

Transcoding alters media objects so that the client can process them. Smith et al. [100] have introduced InfoPyramid to manage and manipulate media objects. InfoPyramid is depicted in Figure 16. The numbers denote content value. A smaller number means a higher value and usually more bandwidth requirements. Values can be based on automatic measures, such as entropy or distortion. They can also be assigned manually. Based on the scores, media can be transcoded to maximize content value or to minimize load time.

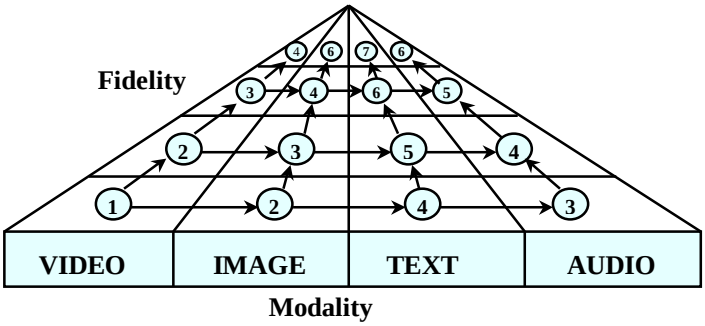

Figure 16: InfoPyramid to manage and manipulate media objects [100].

4.1 Server-side and Proxy-based Adaptation

Server-side and proxy-based adaptations have similar characteristics. They preprocess the document and media objects before delivering them to a client. Figure 17 depicts these processes.

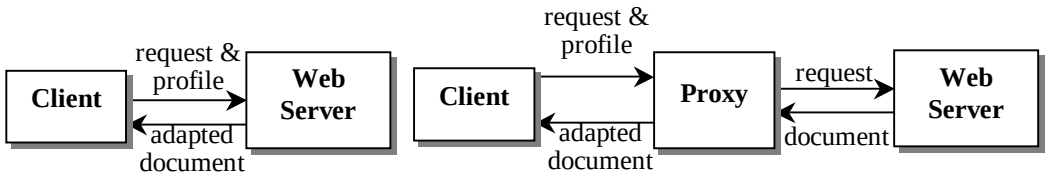

Figure 17: Server-side and proxy-based adaptation.

In proxy-based adaptation, there is an intermediate server between a client and a Web server. The client sends a page request with its profile to the proxy, which sends the page request further to the Web server. The returned document is processed by the proxy, so that it can be displayed at the client. This process involves altering spatial, temporal, and interaction models of the document. Furthermore, media objects can be transcoded. It has been predicted that proxies will not be able to cope with the increased number of mobile devices [64]. Moreover, proxies can only filter information and cannot perform adaptation, for instance, according to language preferences.

In the server-side solution, the client sends its profile straight to the Web server. The Web server can then select an appropriate off-line prepared pretranscoded document to be sent back to the client or adapt the document on-the-fly. Also, media objects can be transcoded. The benefit of server-side adaptation is that the content author can control how the document will be adapted.

Server and proxy-based adaptation requires knowledge about the client's capabilities and user's preferences. There are several ways to obtain these. The most common and applicable is to inspect the HTTP (Hyper Text Transport Protocol) request header. It even works with older browsers, but does not give much information (e.g., only the browser name and version). Other techniques add information to the HTTP request header for better information exchange.

W3C has developed a protocol called Composite Capability / Preference Profile (CC/PP) [61] for capability exchange. In CC/PP, a device profile, i.e., device capabilities and user preferences, is sent in a HTTP request header from a client to a server. The profile is added to the header using the HTTP Extension Framework.

The device profile has been split into a default profile and a difference profile. The default profile is the original profile of the device, which will never change. The difference profile describes the real capabilities and preferences, for instance, the preferred language, amount of added memory, or the version of an updated software. It includes only those values, which are different from the default profile.

The default profile and any differences are described in the RDF language. Table 5 shows examples of RDF statements for a profile.

Subject (Component)	**Predicate (Attribute)**	**Object (Value)**
Browser	BrowserName	X-Smiles
Browser	FramesCapable	No
Hardware	Display	320x200

Table 5: A snippet of a device profile as RDF statements.

It is possible to create complex net graphs with RDF, but fortunately only a simple tree structure has been used in CC/PP. This simplifies parsing and processing of the profiles. CC/PP only defines the structure and transfer format, but not the vocabulary for content negotiation. WAP User Agent Profile (UAProf) [108] and Universal Profiling Schema (UPS) [67] are vocabularies for this purpose.

CC/PP has been criticized [64], because it describes the client capabilities in too much detail, which causes a lot of analysis at a server and results in too many adaptation alternatives. Moreover, describing a default profile for each possible client increases network traffic. As a solution, devices can be classified with a few categories, resulting in less traffic and analysis [64].

Server and proxy-based adaptation can use XSLT stylesheets and transcoding. In case of multimedia documents with a declarative adaptation model, the adaptation alternatives in the document can be preprocessed at a server. This will reduce the document size, which may be essential for mobile devices with limited bandwidth and memory. In case of SMIL, switch cases in Content Control modules can be preprocessed at a server. This causes redundant elements to be removed, therefore reducing the document size. To summarize, CC/PP is a method to transfer the client capabilities to the server, while SMIL has methods to adapt the document. Thus, both are needed to successfully process SMIL switch cases at a server.

SMIL Content Control modules rely on static adaptation, where document alternatives are defined at the authoring time. It is possible to adapt a document to the network bit rate, screen depth, screen size, CPU, and operating system. Also, the available components and supported SMIL modules can be tested. Available user preferences are closed captions, language selection, overdub selection, and audio descriptions.

Spatial adaptation in SMIL can select between alternative layouts, while temporal adaptation selects between alternative temporal paths. Interaction adaptation is automatically performed by the SMIL player. SMIL does not define any methods for media adaptation.

[P5] describes an implementation of a CC/PP capable server with document adaptation made with XSLT. It transforms XML documents, including SMIL documents. The system uses server-side adaptation and the WAP UAProf vocabulary to describe the client capabilities.

It is possible to modify the implemented server to process SMIL switch cases. This requires replacing the XSLT processor with a SMIL content filter. Furthermore, a map between the SMIL attributes and the WAP UAProf attributes is required. Table 6 depicts the mapping. Some of the SMIL attributes do not have a counterpart. On the other hand, the systemComponent attribute can be mapped to any WAP UAProf Attribute.

SMIL Attribute	**WAP UAProf Attribute**
systemBitrate	-
systemCaptions	-
systemLanguage	-
systemOverdubOrCaption	-
systemRequired	-
systemScreenDepth	HardwarePlatform / BitsPerPixel
systemScreenSize	HardwarePlatform / ScreenSize
systemAudioDesc	-
systemCPU	HardwarePlatform / CPU
systemComponent	any
systemOperatingSystem	SoftwarePlatform / OSName

Table 6: Mapping between SMIL test attributes and WAP UAProf attributes.

4.2 Client-side Adaptation

Client-side adaptation can take advantage of all information about the capabilities of the client device. It can use the same adaptation methods as a server. However, the applicability of XSLT and media transcoding in a client is limited by the CPU power and available memory. Fortunately, other methods are available.

Clients can use CSS Media Queries to select the appropriate CSS or XSLT stylesheet. The stylesheet is selected based on the device type (e.g., screen, handheld, tv, and print) and optional tests (e.g., display size, number of colors, and resolution). Processing of a selected XSLT stylesheet may be heavy for a client, but CSS is a light and powerful formatting method for adapting XML documents. In CSS adaptation, the appropriate CSS stylesheet is selected and the document is rendered with that style.

5 Implementation of a SMIL Player

This chapter describes how a multimedia player can be realized for the SMIL language. Also, various available SMIL multimedia players are described.

5.1 Overview of a Multimedia Player

A multimedia player, which composes a presentation from media objects, has a parser to read in a multimedia document, a scheduler to solve timing and synchronization, and other components, such as layout and interaction managers. Figure 18 depicts these main components. The benefit of having separate media players is extensibility. New media players can be added on a plug-in basis, and the player is easy to port to various platforms, as described in [P7].

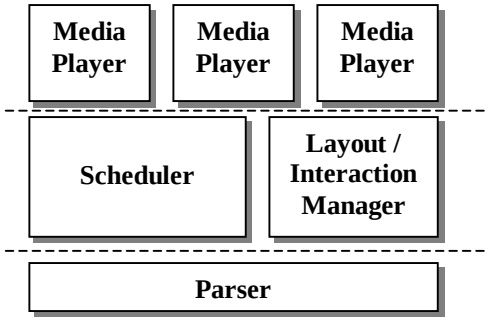

Figure 18: The main components of a multimedia player.

The layout manager takes care of the spatial composition of the presentation. It has the appropriate functionality to fulfill the required spatial model with an applicable constraint system.

The scheduler can be purely reactive or predictive-reactive [56]. Reactive players play out event-based presentations by performing actions as events are received. Predictive-reactive players extend this functionality with an ability to predict the future of the presentation. An MHEG player is a good example of a reactive player, while Madeus is predictive-reactive.

One of the hardest tasks in a scheduler is to keep all the media players synchronized. In documents with an interval-based temporal model, the scheduler must deal with uncertain durations of media objects. Layaïda et al. [66] propose a “best effort” scheduler, which is dynamic enough to deal with these. The algorithms are based on two complementary phases. Reformatting attempts to modify the duration of objects while maintaining the intended

presentation. It shrinks or stretches the duration of objects to be played in the future to meet the presentation schedule. A reparation process is activated when reformatting does not fully succeed. It tries to minimize the period of de-synchronization by postponing the timing of objects that will be played in the future.

The interaction manager responds to navigational interaction by dispatching events to elements or by traversing a hyperlink. The former will start and stop elements, while the latter may cause a jump in space or time. Design interactions modify the parameters of media players, while movie interactions influence the global time of a presentation.

Static adaptation is taken care of by the scheduler. It resolves the final presentation from adaptation alternatives when the presentation is started. This happens by disregarding those parts, which are deselected by adaptation. Dynamic adaptation requires resolving the alternatives during playback.

The following sections describe several SMIL players. It is possible to see an evolution of multimedia players from simple event-based systems to predictive-reactive players.

5.2 Related SMIL 1.0 Players

The SMIL 1.0 specification gave birth to several SMIL players, because it was fairly simple. In SMIL 1.0, the temporal model is interval-based and the spatial model supports flat absolute positioning. Interaction happens via links, while static adaptation controls alternative presentation paths. Thus, implementations are rather straightforward to implement. They are usually written in Java and use Java Media Framework (JMF) to play out audio and video.

Shim et al. [98] have created a Java-based SMIL 1.0 player, which can be embedded into a Web page as an applet. The player has a modular design, where each media player is implemented as a Java Bean. Media players exist at least for text, images, video, and HTML. Scheduling is coordinated with events dispatched by JMF. Thus, the player is reactive. Inspection of their earlier work [110] confirms that the player cannot play out certain SMIL constructs, such as time containers with a positive begin time. This is because they use JMF as a timer to start and stop media. Time containers do not have an associated JMF media player, and thus they cannot be scheduled correctly. Also, some SMIL 2.0 functionalities, such as animations, will not be possible with their JMF dependent event-based approach.

Shin et al. [99] present a Java-based SMIL player. It is a standalone player, which uses JMF for media playback. They create a new thread for every media object to prefetch and render them. It remains unclear how the scheduling is performed in details. They also present a buffering and caching mechanism to cope with unreliable network behavior.

A framework for event-driven multimedia presentation systems has been proposed by Rodrigues et al. [86]. It schedules SMIL presentations with an event-based scheduler. Media players are implemented with JMF players. The scheduler reacts to events dispatched by the JMF players. To cope with intervals, which are not event-based, they use timers. Timers dispatch events, which cause

the presentation to proceed. To cope with variations caused by unpredictable factors such as communication and operational delays, media time in media players is polled and the result is used to adjust the presentation playback.

Several other Java-based SMIL 1.0 players are available, e.g., Soja by Helio, Hypermedia Presentation and Authoring System (HPAS) by Compaq, and Streaming Synchronized MultiMedia (S2M2) developed by the National Institute of Standards and Technology (NIST).

5.3 Related SMIL 2.0 Players

Sampaio et al. [92] present a method to verify the semantics of a SMIL document, which is not achieved with a DTD or XML Schema. They translate the SMIL document into RT-LOTOS [19], and analyze it with a RTL tool, which generates a minimal reachability graph. The graph can be inspected to find inconsistencies in the document. To avoid state space explosion, which happens easily, actions in the graph are combined, if they do not have incompatible synchronization arcs. In addition to verification, they present a method to play out the SMIL document. The reachability graph is converted into a scheduling graph called Time Labeled Automaton (TLA) and into a contextual information file. The TLA describes temporal behavior, while the contextual information file describes spatial positioning, media objects, and exclusive time containers.

In the TLA, each state has a timer, which determines the duration for that state. The timer starts when the automaton enters the state, and freezes when the state is left. Aggregation techniques can be applied to avoid the state space explosion problem. In addition to scheduling SMIL, the same approach can be used to schedule other multimedia document formats [91]. The player has been implemented in Java and uses JMF for media playback.

Other SMIL 2.0 players are Grins [17], RealOne [84], and QuickTime [4]. Microsoft's Internet Explorer can also display the XHTML+SMIL language profile. These are commercial players without details about the implementation, and thus it is not possible to make a deeper analysis about them.

Overall, most of the SMIL players rely on an event-based scheduler, which listens to media players or timers. However, certain SMIL 2.0 functionalities, such as animations and transitions, require continuous sampling of the presentation. Sampling takes snapshots of the presentation as a function of time. The sampling rate used should be small enough so that a user perceives sampled images as a continuous movie. For example, the CSiro SVG viewer [21] has a sampling scheduler to play out SMIL Animations. Even though the temporal model of SVG is much simpler than that of SMIL, the principles are the same. Therefore, it is possible to implement a SMIL player with a similar scheduler.

5.4 The X-Smiles Browser

The SMIL 2.0 player, implemented as part of this thesis, is a component of X-Smiles. X-Smiles is an XML browser intended for embedded devices. The

implementations of various components of the browser are mainly presented in [44][45][104][P4][P6][P7][P8].

The X-Smiles browser was started as a student software project in 1998. It started as a SMIL 1.0 player [P6], but was continued afterwards towards a full-blown XML browser. The task was made easier by several 3rd party components. Currently, the browser can play out XHTML, SMIL, and XForms, which are developed by the current research group. The browser also supports XML parsing, XSLT transformations, XSL FO, SVG, and X3D, developed by 3rd party open-source projects.

The X-Smiles browser is intended for various devices, such as desktop PCs, PDAs, and digital televisions. Figure 19 depicts the architecture of the browser. The browser is made of four layers. At the bottom, the XML parser (*Xerces*, developed by the Apache project), reads in the XML document. The XSLT transformer (*Xalan*, from the Apache project) can transform the XML document into renderable language. The next layer, the browser core, contains an XML broker and data about the document history and browser configuration. The XML broker will check the namespace of the elements in the XML document and forwards the elements to the respective *Markup Language Functional Component* (MLFC). Each MLFC can render one XML language. This process is documented in detail in [P8]. Figure 19 depicts some of the MLFCs available in the browser. On top of the figure, there are the *Graphical User Interfaces* (GUIs). These allow customization of the browser for various platforms. The browser also includes virtual GUIs [103], which simulate target devices. This enables easy prototyping.

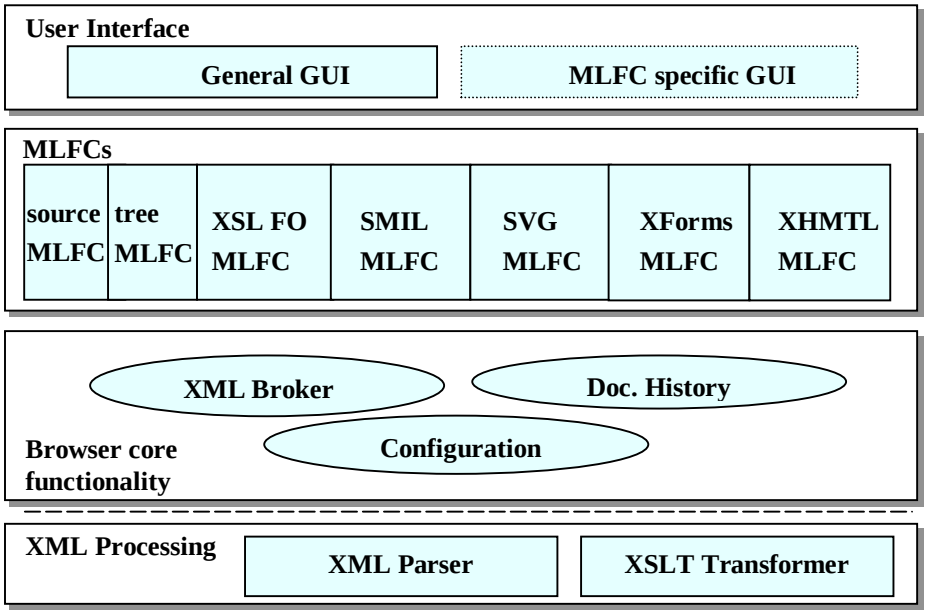

Figure 19: The architecture of the X-Smiles browser [104].

The browser is designed to handle *hybrid documents*. A hybrid document is an XML document, which contains several XML languages, distinguished by a namespace. To handle these, there are two types of MLFCs in the framework: *hosts* and *parasites*. There is one host MLFC per document. The host is

identified by the document's root element and it decides the master layout for the document. The layout model can differ between different host MLFCs. For example, XHTML has a flow type of layout, while SVG uses absolute positioning.

A parasite MLFC always needs a host to live in. A parasite, such as XForms MLFC, has to adapt to the host's layout model. Sometimes the parasite does not have visible components (e.g., XML Events MLFC). The host and parasite MLFCs use common interfaces to communicate with each other. For instance, a SMIL document needs to access the graphical component of an XForms element in order to place it on the screen. A VisualComponentService interface is provided by renderable XForms elements for the SMIL MLFC to achieve this.

One benefit of the modular design is that it is rather easy to add new XML languages to the browser without modifying the existing ones. This enables creation of various small XML languages [P1] and completely new experimental languages [102][P3].

5.5 SMIL Players in X-Smiles

[P6] describes the first version of the SMIL 1.0 player in X-Smiles. It had a similar design to the other SMIL 1.0 players, as it used JMF to play and schedule the presentation. Therefore, it had the same limitations as the other SMIL 1.0 players.

An improved version has been written to play out SMIL 2.0 presentations [P4][P7]. The player can play out SMIL 2.0 Basic profile documents with a few additional SMIL modules, such as event timing, basic animations, and brush media. The design of the new player has been improved in several ways. First, it has been designed to be JMF independent as depicted in Figure 20. Text and image players have been implemented with standard Java classes, while audio and video are played with JMF. These media players implement an interface, which makes adding new media players easy. The scheduler has been designed to orchestrate the presentation with its internal timers. Thus, the JMF media player can be removed all together, and the SMIL player is still functional.

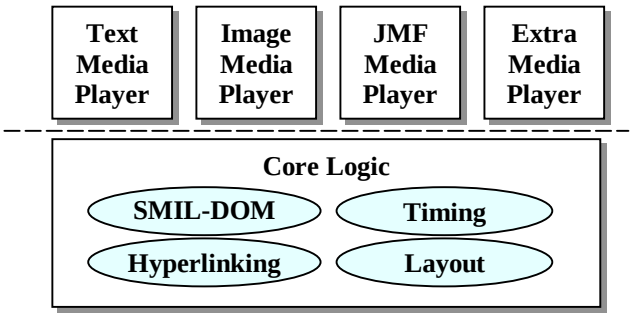

Figure 20: Design of the SMIL 2.0 player [P7].

Scheduling is a combination of timers and listening for events from JMF and the user interface. Timers take care of resolved times, while events take care of

SMIL 2.0 event-based timing. In case of unresolved times, due to streamed media with unknown duration, the player resorts to events. When the streamed media ends, the player will receive an event from JMF, and the time will become resolved.

SMIL Animation is handled by an animation scheduler, which uses a limited sampling approach. It only samples animation elements every 50 ms to calculate animation values and updates animation target attributes accordingly. This requires rerendering the presentation every 50 ms.

The SMIL 2.0 player has been designed to work in the X-Smiles browser, but can also be executed as a standalone player. The standalone player is portable, and has been run in a desktop PC, a PDA, and a digital television set-top box [P7]. Figure 21 depicts a digital television set-top box version of the player.

Figure 21: Digital television set-top box version of the SMIL 2.0 player.

Porting is made easy with modular media players. A new set of players can be implemented for any system. The core system has been written to be platform-independent, and does not require any porting.

6 Conclusions

Web-based multimedia is becoming widespread. This thesis has extended and evaluated the declarative SMIL multimedia language to study the future of Web-based multimedia languages. The main contributions of the work described in this thesis are the following.

Publications [P1-P4] describe several novel extensions to the SMIL 2.0 language:

- An implementation and evaluation of SMIL and XForms language integration [P2]. XForms provides a convenient method to add Web forms to any XML language.
- An extension to SMIL 2.0 to play out 3D audio [P3]. This provides advanced audio capabilities, which are currently missing in the SMIL 2.0 language.
- An extension to integrate scripting into SMIL [P4]. ECMAScript scripts are executed with the help of XML Events.
- Generalized methods to extend the SMIL language [P1]. To some extent, these methods can also be used with other XML languages.

As a conclusion, the SMIL language can be extended with new capabilities. The extensions fit well into SMIL, if they rely on absolute positioning and SMIL's temporal model. 3D audio and scripting fall into these categories. However, if another kind of spatial model is required, e.g., a text flow, there will be problems. An experiment was performed with the XForms language. Its repeat feature requires a text flow, and therefore a CSS layout was experimented with. The CSS formatted the body section of the SMIL document, which also describes the timing. The result was that the same section defined the layout and timing, which made authoring harder. The same conflict can be seen in the XHTML+SMIL profile, which also combines layout and timing in the same section.

Currently, W3C is defining the SVG 1.2 specification, which will include more SMIL timing modules. There have been plans to include a text flow spatial model, as well. It remains to be seen how the planned text flow will be integrated with the SMIL timing modules.

The reusability of XML languages was shown with a 3D audio extension, which was designed to be modular. Therefore, it was possible to use it in SMIL, but also in other XML languages, such as XHTML. Reusable modules require care with their spatial and temporal models. The 3D audio extension was designed to interoperate with absolute and text flow models. Playing out audio

samples requires simple definition of start and end times with an interval- or event-based model.

Adaptation of documents for various devices was studied in publication [P5]. The adaptation happens on a server, which selects the best fitting XSLT and processes it before sending the page back to the client. The publication gives a novel method to select an XSLT stylesheet based on a device profile.

Publication [P6] describes an early implementation of a SMIL 1.0 player, which has been used as a reference and basis for the implemented SMIL 2.0 player. Publication [P7] presents a design of a portable SMIL player. The design makes it easy to port the player to any platform with a support for Java. Finally, publication [P8] gives a browser framework for XML language extensibility and integration. The framework has been a valuable tool to experiment with the above-mentioned extensions.

7 Summary of Publications and Author's Contribution

This chapter of the thesis summarizes the articles and describes the contributions of the author. The work presented here is part of a larger project, where the X-Smiles XML browser has been developed. The basic ideas for the browser were given by the original X-Smiles development team, M.Sc. Jukka Heinonen, M.Sc. Niklas von Knorring, M.Sc. Aki Teppo, Mr. Teemu Ropponen, M.Sc. Oskari Kurki, and M.Sc. Toni Kopra. Also, M.Sc. Mikko Honkala and Mr. Juha Vierinen have given a lot of feedback about the ideas presented in these articles. Specifically, M.Sc. Mikko Honkala has designed and implemented the XForms implementation referred to in the articles [P1], [P2], [P4], and [P6]. The content selection algorithm in [P5] was designed in collaboration with Mr. Juha Vierinen. M.Sc. Pablo Cesar performed the graphical user interface work in [P7].

Publication [P1]
This article presents nine general ways to extend SMIL 2.0. The methods include ways to add new input sources and output capabilities along with extended internal logic. Furthermore, an implementation of an extensible SMIL player is given to play out the presentations. These extensions have proved to be useful in several projects, where SMIL is used as a presentation language. Typically, these are specific custom applications, which require multimedia processing.
The author has developed the methods and the application. He has written most of the text, getting support from Prof. Petri Vuorimaa mostly in terms of comments and proofreading the article.

Publication [P2]
This article integrates XForms with SMIL. SMIL 2.0 enables user interaction with its time-based hyperlinking and event model. However, often more advanced interaction is desired, such as information exchange with a server. Traditionally, Web forms have offered means to send data to a server. XForms is an effort by W3C to create a host language independent Web form standard. However, the advanced XForms repeat feature requires a flow layout, which is not available in SMIL. Issues in the integration of SMIL and XForms are presented with possible solutions.
The author has implemented the experimental CSS layout engine for SMIL and developed two of the applications. The integration of SMIL and XForms was designed with M.Sc. Mikko Honkala, who also completely designed and implemented XForms. The author has written 70 % of the text.

Publication [P3]
This article extends SMIL with 3D audio. The SMIL 2D layout is extended with an extra dimension. New audio elements are positioned in the 3D space, whilst a listener element defines a listening point. Similarly to MPEG-4 AABIFS perceptual modeling approach, an environment element describes environmental parameters for audio elements. These extensions enable interactive 3D audio capabilities in SMIL. In addition, any XML based rendering language, such as XHTML and SVG, can be extended with 3D audio capabilities by using a similar approach.
The author has designed and implemented the 3D audio extension for SMIL and has written most of the text. Dr. Tapio Lokki helped mainly by proofreading the text.

Publication [P4]
This article describes a design and implementation of a SMIL 2.0 player. Web applications frequently require more control over multimedia presentations than offered by the standard SMIL 2.0. The article extends SMIL with scripting. The SMIL 2.0 player is also integrated into the X-Smiles browser, thus enabling playing SMIL with XForms, XSL FO, SVG, and XHTML.
The author has designed and implemented the SMIL 2.0 player on top of the SMIL 1.0 player. XML Events has also been mainly implemented by the author. The author has written 90 % of the text.

Publication [P5]
This article describes an implementation of a CC/PP client and server. It presents a method to select appropriate content on the server based on the client capabilities. This happens by transforming an abstract XML document with XSLT to presentation format. The result is sent to the client.
The author has designed and implemented the CC/PP client and server. The server-side logic was designed in collaboration with Mr. Juha Vierinen. The author has written 90 % of the text.

Publication [P6]
This article presents a design and implementation of a SMIL 1.0 player. The player is part of the X-Smiles browser, and can display text, images, audio, and video. A SMIL document can also include references to SVG and other XML languages supported by the X-Smiles browser. Integration with XForms is also introduced.
The author has implemented the integration of the SMIL player with other XML languages and has written Sections 3.4, 4, and 5.

Publication [P7]
This article presents the design and implementation of a portable SMIL player. The player has been written in Java and can be executed on top of AWT, Swing, and *ftv* GUI frameworks. This allows it to be run on various platforms, e.g., PCs, PDAs, and digital television set-top boxes. New media players can easily be added to the player, therefore complying with the fundamental idea of SMIL, integrating various media formats. The performance of AWT, Swing, and *ftv* versions are evaluated to see how they fit to play SMIL. Finally, requirements for the underlying graphical environment to play SMIL are given.

The author has designed and implemented the portable SMIL player. The author has written 60 % of the text, i.e., half of the Sections 1 and 4 and completely the Sections 3 and 5. M.Sc. Pablo Cesar has designed and implemented the *ftv* GUI framework.

Publication [P8]
This article presents a framework for an XML browser capable of displaying hybrid XML documents. The framework enables the implementation of extensions for existing XML languages. As an example, SMIL, XForms, and XML Events languages are integrated.
The author has partially designed and implemented the proposed framework, and has written 50 % of the text.

Bibliography

[1] P. Ackermann, "Direct Manipulation of Temporal Structures in a Multimedia Application Framework," in *Proc. ACM Multimedia 1994*, San Francisco, CA, USA, Oct. 1994, pp. 51-58.

[2] S. Adler et al., "Extensible stylesheet language (XSL) Version 1.0," *W3C Recommendation*, Oct. 15, 2001.

[3] J. Allen, "Maintaining Knowledge about Temporal Intervals," *Communications of ACM*, vol. 26, no. 11, pp. 832-843, 1983.

[4] Apple Computer Inc., "QuickTime and SMIL," referenced Dec. 19, 2002, http://developer.apple.com/techpubs/quicktime/qtdevdocs/IQ_InteractiveMovies/quicktimeandsmil/

[5] J. Ayars et al., "Synchronized Multimedia Integration Language (SMIL 2.0)," *W3C Recommendation*, Aug. 7, 2001.

[6] G. J. Badros et al., "A Constraint Extension to Scalable Vector Graphics," in *Proc. Tenth International World Wide Web Conference*, Hong Kong, May 1-5, 2001.

[7] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," *Scientific American*, May 2001.

[8] S. Boll, "*ZYX, Towards Flexible Multimedia Document Models for Reuse and Adaptation*," PhD Thesis, University of Vienna, Austria, 2001.

[9] B. Bos et al., "CSS2: Cascading Style Sheets, level 2," *W3C Recommendation*, May 12, 1998.

[10] J. Bormans, J. Gelissen, and A. Perkis, "MPEG-21: The 21st Century Multimedia Framework," *IEEE Signal Processing Magazine*, vol. 20, no. 2, pp. 53-62, 2003.

[11] A. Borning, R. Lin, and K. Marriot, "Constraint-based Document Layout for the Web," *Multimedia Systems*, vol. 8, no. 3, pp. 177-189, 2000.

[12] N. Bouvin and R. Schade, "Integrating Temporal Media and Open Hypermedia on the World Wide Web," in *Proc. Eight International World Wide Web Conference*, Toronto, Canada, May 11-14, 1999, pp. 375-387.

[13] T. Bray et al., "Extensible Markup Language (XML) 1.0 (second edition)," *W3C Recommendation*, Oct. 6, 2000.

[14] T. Bray et al., "Namespaces in XML," *W3C Recommendation*, Jan. 14, 1999.

[15] D. Bulterman, "SMIL 2.0 Part 1: Overview, Concepts, and Structure," *IEEE Multimedia*, vol. 8, no. 4, pp. 82-88, 2001.

[16] D. Bulterman, "SMIL 2.0 Part 2: Examples and Comparisons," *IEEE Multimedia*, vol. 9, no. 1, pp. 74-84, 2002.

[17] D. Bulterman et al., "GRiNS: an Authoring Environment for Web Multimedia," in *Proc. World Conference on Educational Multimedia, Hypermedia and Educational Telecommunications*, Seattle, Washington, USA, June 19-24, 1999.
[18] J. Clark, "XSL Transformations (XSLT) Version 1.0," *W3C Recommendation*, Nov. 16, 1999.
[19] C. Concolato, J.-C. Moissinac and J.-C. Dufourd, "Representing 2D Cartoons using SVG," in *SMIL Europe 2003 Conference*, Paris, France, February 12-14, 2003.
[20] J.-P. Courtiat, R. de Oliveira, and L. Andriantsiferana, "Specification and Validation of Multimedia Protocols using RT-LOTOS," in *Proc. Fifth IEEE Computer Society Workshop on Future Trends of Distributed Computing Systems*, August 28-30, 1995, pp. 354-362.
[21] CSiro, "SVG Toolkit," referenced May 29, 2003, http://sis.cmis.csiro.au/svg/
[22] S. J. DeRose et al., "XML Linking Language (XLink) Version 1.0," *W3C Recommendation*, June 27, 2001.
[23] S. Decker, P. Mitra, and S. Melnik, "Framework for the Semantic Web: an RDF Tutorial," *IEEE Internet Computing*, vol. 4, no. 6, pp. 68-73, 2000.
[24] M. Dubinko et al., "XForms 1.0," *W3C Recommendation*, Oct. 14, 2003.
[25] A. Duda, "Structured Temporal Composition of Multimedia Data," in *Proc. IEEE International Workshop on Multi-Media Data Base Management Systems*, Blue Mountain Lake, NY, USA, Aug. 28-30, 1995, pp. 136-142.
[26] M. Echiffre et al., "MHEG-5 – Aims, Concepts, and Implementation Issues," *IEEE Multimedia*, vol. 5, no. 1, pp. 84-91, Jan-Mar 1998.
[27] Standard ECMA-262, "*ECMAScript Language Specification, 3rd Edition*," December 1999, http://www.ecma-international.org/publications/standards/ECMA-262.htm
[28] M. Egenhofer and R. Franzosa, "Point-Set Topological Spatial Relations," *Int. Journal of Geographical Information Systems*, vol. 5, no. 2, pp. 161-174, 1991.
[29] D. Fallside, "XML Schema Part 0: Primer", *W3C Recommendation*, May 2, 2001.
[30] J. Ferraiolo et al., "Scalable Vector Graphics (SVG) 1.1 Specification," *W3C Recommendation*, Jan. 14, 2003.
[31] M. Froggatt, "The power to play," *IEE Review*, vol. 48, no. 2, pp. 13-19, 2002.
[32] R. Gordon and S. Talley, "*Essential JMF: Java Media Framework*," Prentice Hall, Upper Saddle River, New Jersey, 1998.
[33] V. Gudivada, "Multimedia Systems: An Interdisciplinary Perspective," *ACM Computing Surveys*, vol. 27, no. 4, pp. 545-548, 1995.
[34] P. Haavisto, R. Castagno, and H. Honko, "Multimedia Standardization for 3G Systems," in *Proc. 5th Intl. Conf. on Signal Processing*, Beijing, China, Aug. 21-25, 2000, pp. 32-39.

[35] M. Haindl, "A New Multimedia Synchronization Model," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 1, pp. 73-83, 1996.
[36] C. L. Hamblin, "Instants and Intervals," in *Proc. of the 1st conf of the Intl. Society for the Study of Time*, Springer-Verlag, New York, 1972, pp. 324-331.
[37] L. Hardman et al., "The Link vs. the Event: Activating and Deactivating Elements in Time-based Hypermedia," *the Review of Hypermedia and Multimedia*, vol. 6, pp. 89-109, 2000.
[38] L. Hardman et al., "Hypermedia: the Link with Time," *ACM Computing Surveys*, vol. 31, no 4es, 1999.
[39] L Hardman et al., "Do You Have the Time? Composition and Linking in Time-based Hypermedia," in *Proceedings of the tenth ACM Conference on Hypertext and hypermedia*, Darmstadt, Germany, Feb. 21-25, 1999, pp. 189-196.
[40] S. Hartwig et al., "Mobile Multimedia - Challenges and Opportunities," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, pp. 1167-1178, 2000.
[41] D. Hillman, "Using Dublin Core," April 2001, referenced May 11, 2003, http://purl.org/dc/documents/wd/usageguide-20000716.htm.
[42] N. Hirzalla, B. Falchuk, and Ahmed Karmouch, "A Temporal Model for Interactive Multimedia Systems," *IEEE Multimedia*, vol. 2 no. 3, pp. 24-31, 1995.
[43] M. E. Hodges, R. M. Sasnett, and M. S. Ackerman, "A Construction Set for Multimedia Applications," *IEEE Software*, vol. 6, no. 1, pp. 37-43, 1989.
[44] M. Honkala and P. Vuorimaa, "XForms in X-Smiles," *WWW Journal*, vol. 4, no. 3, 2001, pp. 151-166.
[45] M. Honkala and P. Vuorimaa, "Advanced UI features in XForms", in *Proc. 8th International Conference on Distributed Multimedia Systems*, San Francisco, California, USA, Sept. 25 - 28, 2002, pp. 715-722.
[46] P. Hoschka et al., "Synchronized Multimedia Integration Language (SMIL) 1.0 Specification," *W3C Recommendation*, June 15, 1998.
[47] P. Hoschka, "An Introduction to the Synchronized Multimedia Integration Language," *IEEE Multimedia*, vol. 5, no. 4, pp. 84-88, 1998.
[48] P. Hoschka and C. Lilley, "Displaying SMIL Basic Layout with a CSS2 Rendering Engine," *W3C Note*, July 20, 1998.
[49] J. Hunter, "DTV Data Services – Experiences of the Rollout on UK-DTT," in *IEE Colloquium on Interactive Television*, no. 99/200, 1999.
[50] ISO/IEC 10179:1996, "*Document Style Semantics and Specification Language (DSSSL),*" 1996.
[51] ISO/IEC 10744, "*Hypermedia/Time-based Document Structuring Language (HyTime),*" 1992.
[52] ISO/IEC 10744:1997, "*Hypermedia/Time-based Structuring Language (HyTime). 2nd Edition,*" 1997.
[53] ISO/IEC 13522-5, "*(MHEG-5), Support for Base-Level Interactive Applications,*" 1997.

[54] ISO/IEC 14496:1999, "*Coding of Moving Pictures and Audio*," 1999.
[55] ISO/IEC 14772-1, "*The Virtual Reality Modeling Language*," 1997.
[56] M. Jourdan et al., "Madeus, an Authoring Environment for Interactive Multimedia Documents," in *Proc. Sixth ACM International Conference on Multimedia*, Bristol, UK, Sept. 14-16, 1998, pp. 267-272.
[57] M. Jourdan et al., "Constraints Techniques for Authoring Multimedia Documents," *Constraints Journal*, vol. 6, no. 1, pp. 115-132, 2001.
[58] W. ten Kate, P. Deunhouwer, and R. Clout, "Timesheets - Integrating Timing in XML," in *Proc. WWW9 Workshop: Multimedia on the Web*, Amsterdam, Netherlands, May 15, 2000.
[59] M. Kim, S. Wood, and L. Cheok, "Extensible MPEG-4 Textual Format (XMT)," in *Proc. 2000 ACM Workshops on Multimedia*, Los Angeles, California, USA, Oct. 30 – Nov. 4, 2000, pp. 71-74.
[60] M. Kim and J. Song. "Multimedia Documents with Elastic Time," in *Proc. Third ACM International Conference on Multimedia*, San Francisco, California, Nov. 5-9, 1995, pp. 143-154.
[61] G. Klyne et al., "Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0," *W3C Proposed Recommendation*, October 15, 2003.
[62] J. F. Koegel et al., "HyOctane: A HyTime Engine for an MMIS," in *Proc. First ACM International Conference on Multimedia*, Anaheim, California, USA, Aug. 1-6, 1993, pp. 129-136.
[63] J. F. Koegel Buford, "*Multimedia Systems*," Addison-Wesley, New York, New York, 1994.
[64] V. Korolev and A. Joshi, "An End-End Approach to Wireless Web Access," in *Workshop of 21st IEEE Intl. Conf. Distributed Computing Systems*, Phoenix, Arizona, USA, April 16-19, 2001, pp. 473-478.
[65] O. Lassila, "Resource Description Framework (RDF) Model and Syntax Specification," *W3C Recommendation*, Feb. 22, 1999.
[66] N. Layaïda, L. Sabry-Ismail, and C. Roisin, "Dealing with Uncertain Durations in Synchronized Multimedia Presentations," *Multimedia Tools and Applications*, vol. 18, no. 3, pp 213-231, 2002.
[67] T. Lemlouma and N. Layaïda, "SMIL Content Adaptation for Embedded Devices," in *SMIL Europe 2003 Conference*, Paris, France, February 12-14, 2003.
[68] T. Lewis, "Information appliances: gadget Netopia," *Computer*, vol. 31, no. 1, pp. 59-68, 1998.
[69] T. Little and A. Ghafoor, "Spatio-Temporal Composition of Distributed Multimedia Objects for Value-Added Networks," *Computer*, vol. 24, no. 10, pp. 42-50, 1991.
[70] Macromedia Inc., "Macromedia Flash MX 2004," referenced Oct 25, 2003, http://www.macromedia.com/software/flash/
[71] B. S. Manjunath, P. Salembier, and T. Sikora, "*Introduction to MPEG-7*", John Wiley & Sons Ltd., Chichester, England, 2002.
[72] B. D. Markey, "HyTime and MHEG," in *the 37th IEEE Conf. Of Compcon Spring 1992*, San Francisco, California, USA, Feb. 24-28, 1992, pp. 25-40.

[73] T. Meyer-Boudnik and W. Effelsberg, "MHEG Explained," *IEEE Multimedia*, vol. 2., no 1, pp. 26-38, 1995.

[74] Y. Neuvo and J. Yrjänäinen, "Wireless Meets Multimedia - New Products and Services," in *Proc. Intl. Conf. on Image Processing*, Rochester, New York, USA, Sept. 22-25, 2002, pp. I-1 – I-4.

[75] S. R. Newcomb, "Standard Music Description Language Complies with Hypermedia Standard," *Computer*, vol. 24, no. 7, pp. 76 –79, 1991.

[76] D. Newman et al., "XHTML+SMIL Profile," *W3C Note*, Jan. 31, 2002.

[77] J. van Ossenbruggen, L. Hardman, and L. Rutledge, "Integrating Multimedia Characteristics in Web-based Document Languages," *CWI Technical Report INS-R0024*, Dec. 2000, http://www.cwi.nl/ftp/CWIreports/INS/INS-R0024.ps.Z.

[78] K. Page, D. Cruickshank, and D. De Roure, "Its About Time: Link Streams as Continuous Metadata," in *Proc. Twelfth ACM Conference on Hypertext and Hypermedia*, Århus, Denmark, Aug. 14-18, 2001, pp. 93-102.

[79] D. Papadias and T. Sellis, "Qualitative Representation of Spatial Knowledge in 2D Space", *VLOB Journal*, vol. 3, no. 4, pp. 479-516, 1994.

[80] S. Pemberton et al., "XHTML 1.0: The Extensible HyperText Markup Language," *W3C Recommendation*, Jan. 26, 2000.

[81] C. Peng and P. Vuorimaa, "A Digital Teletext Service," in *Proc. 9th WSCG International Conference on Computer Graphics, Visualization and Computer Vision*, Czech Republic, Feb. 5-9, 2001, pp. 120-125.

[82] F. Pereira and T. Ebrahimi, "*The MPEG-4 Book*," Prentice Hall, Upper Saddle River, New Jersey, USA, 2002.

[83] J. Postel et al., "An Experimental Multimedia Mail System," *ACM Transactions on Office Information Systems*, vol. 6, no. 1, pp. 63-81, 1988.

[84] RealNetworks Inc., "RealSystem Production Guide," referenced Dec. 19, 2002, http://service.real.com/help/library/guides/production8/htmfiles/smilext.htm

[85] M. Revett and G. South, "Consumer Devices for eCommerce Access," *BT Technology Journal*, vol. 17, no. 3, pp. 112-123, 1999.

[86] R. Rodrigues and L. Soares, "A Framework for Event-driven Hypermedia Presentation Systems," in *Proc. 8th International Conference on Multimedia Modeling*, Amsterdam, Netherlands, Nov. 7-9, 2001. pp. 169-185.

[87] L. Rutledge, "SMIL 2.0: XML for Web Multimedia," *IEEE Internet Computing*, vol. 5, no. 5, pp. 78-84, 2002.

[88] L. Rutledge et al., "Anticipating SMIL 2.0: The Developing Cooperative Infrastructure for Multimedia on the Web," in *Proc. Eighth International World Wide Web Conference*, Toronto, Ontario, Canada, May 11-14, 1999.

[89] L. Rutledge, L. Hardman, and J. von Ossenbruggen, "Evaluating SMIL: Three User Case Studies," in *Proc. ACM Multimedia '99*, Orlando, Florida, USA, Oct. 30-Nov. 5, 1999, pp. 171-174, vol. 2.

[90] L. Rutledge et al., "Structural Distinctions Between Hypermedia Storage and Presentation," in *Proc. ACM Multimedia '98*, Bristol, UK, Sept. 12-16, 1998, pp. 145-150.

[91] P. Sampaio and J. Courtiat, "A Formal Approach for the Presentation of Interactive Multimedia Documents," in *Proc. Eighth ACM International Conference on Multimedia*, Marina del Rey, California, USA, Oct. 30 – Nov. 4, 2000, pp. 435-438.

[92] P. Sampaio and J. Courtiat, "Hypermedia and Graphics 2: An Integrated Environment for the Presentation of Consistent SMIL 2.0 Documents," in *Proc. ACM Symposium on Document Engineering*, Atlanta, Georgia, USA, Nov. 9-10, 2001, pp. 115-124.

[93] M. Sannella et al., "Multi-way Versus One-way Constraints in User Interfaces: Experiences with the Deltablue Algorithm", *Software Practice and Experience*, vol. 23, no. 5, pp. 529-566, 1993.

[94] P. Schmitz, "Multimedia Meets Computer Graphics in SMIL 2.0: A Time Model for the Web," in *Proc. Eleventh International World Wide Web Conference*, Honolulu, Hawaii, USA, May 7-11, 2002.

[95] P. Schmitz, "Unifying Scheduled Time Models with Interactive Event-based timing," *Technical Report MSR-TR-2000-114*, Microsoft Corporation, Nov. 2000.

[96] P. Schmitz, "The SMIL 2.0 Timing and Synchronization Model: Using Time in Documents", *Technical Report MSR-TR-2001-01*, Microsoft Corporation, Jan. 2001.

[97] J. Sevanto, "Multimedia Messaging Service for GPRS and UMTS," in *Proc. IEEE Wireless Communications and Networking Conference*, New Orleans, USA, Sept. 21-24, 1999, pp. 1422 –1426.

[98] S. Shim, J. Gao, and Y. Wang, "Multimedia Presentation Components in E-commerce," in *the Second Intl. Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems*, San Jose, California, USA, June 8-9, 2000, pp. 158-165.

[99] D. Shin and D. Shin, "Design and Implementation of a SMIL player," *IEEE Transactions on Consumer Electronics*, vol. 48, no. 3, pp. 168-169, 2002.

[100] J. Smith, R. Mohan, and C.-S. Li, "Scalable Multimedia Delivery for Pervasive Computing," in *Proc. Seventh ACM International Conference on Multimedia (Part 1)*, Orlando, Florida, USA, Oct. 30 – Nov. 4, 1999, pp. 131-140.

[101] T. Usdin and T. Graham, "XML: Not a Silver Bullet, But a Great Pipe Wrench," *StandardView*, vol. 6, no. 3, pp. 125-132, 1998.

[102] J. Vierinen and P. Vuorimaa, "Dynamic Markup Language Based User Interfaces For A Browser," in *Proc. IASTED International Conference on Communications, Internet & Information Technology*, St. Thomas, Virgin Islands, USA, Nov. 18-20, 2002, pp. 54-59.

[103] J. Vierinen, K. Pihkala, and P. Vuorimaa, "XML based Prototypes fpr Future Mobile Services," in *Proc. 6th World Multiconference on Systemics, Cybernetics and* Informatics, Orlando, USA, July 14-18, 2002.

[104] P. Vuorimaa et al., “A Java Based XML Browser for Consumer Devices,” in *the 17th ACM Symposium on Applied Computing*, Madrid, Spain, March 10-13, 2002, pp. 1094-1099.

[105] P. Vuorimaa, J. Teirikangas, and J. Vierinen, “Ubiquitous Multimedia Services with XML,” in *Proc. 1st Int. Conf. Universal Access in Human-Computer Interaction*, New Orleans, Louisiana, USA, Aug. 5-10, 2001, pp. 742-746.

[106] T. Wahl and K. Rothernel, “Representing Time in Multimedia Systems,” in *Proc. Intl. Conference on Multimedia Computing and Systems*, Boston, Massachussetts, USA, May 14-19, 1994, pp. 538-543.

[107] R. Want et al., “Disappearing hardware,” *IEEE Pervasive Computing*, vol. 1, no. 1, pp. 36-47, 2002.

[108] WAP Forum, “WAG UAProf,” Work in Process, May 30, 2001.

[109] Web 3D Consortium, “X3D Draft Specification,” Feb. 24, 2002, http://www.web3d.org/x3d/

[110] J. Xia, S. Shim, and Y. Wang, “Design and Implementation of a SMIL Player,” in *Proc. SPIE*, vol. 3648, San Jose, CA, USA, pp. 382-389, 1999.

[111] D. Žagar and S. Rimac-Drlje, “Applications Classification and QoS Requirements,” in *Proc. 24th Intl. Conf. on Information Technology Interfaces*, Cavtat, Croatia, June 24-27, 2002, pp. 517-522.