Dottorato di Ricerca in Informatica

Università di Bologna, Padova

INF/01 - INFORMATICA

Rich Media Content Adaptation in E-Learning Systems

Silvia Mirri

Department of Computer Science, University of Bologna

Mura Anteo Zamboni, 7

I-40127 Bologna, Italy

Coordinatore:
Prof. Özalp Babaoglu

Tutore:
Prof. Marco Roccetti

Abstract

The wide use of e-technologies represents a great opportunity for underserved segments of the population, especially with the aim of reintegrating excluded individuals back into society through education. This is particularly true for people with different types of disabilities who may have difficulties while attending traditional on-site learning programs that are typically based on printed learning resources. The creation and provision of accessible e-learning contents may therefore become a key factor in enabling people with different access needs to enjoy quality learning experiences and services.

Another e-learning challenge is represented by *m-learning* (which stands for *mobile learning*), which is emerging as a consequence of mobile terminals diffusion and provides the opportunity to browse didactical materials everywhere, outside places that are traditionally devoted to education.

Both such situations share the need to access materials in limited conditions and collide with the growing use of rich media in didactical contents, which are designed to be enjoyed without any restriction.

Nowadays, Web-based teaching makes great use of multimedia technologies, ranging from Flash animations to prerecorded video-lectures. Rich media in e-learning can offer significant potential in enhancing the learning environment, through helping to increase access to education, enhance the learning experience and support multiple learning styles. Moreover, they can often be used to improve the structure

of Web-based courses. These highly variegated and structured contents may significantly improve the quality and the effectiveness of educational activities for learners. For example, rich media contents allow us to describe complex concepts and process flows. Audio and video elements may be utilized to add a “human touch” to distance-learning courses. Finally, real lectures may be recorded and distributed to integrate or enrich on line materials. A confirmation of the advantages of these approaches can be seen in the exponential growth of video-lecture availability on the net, due to the ease of recording and delivering activities which take place in a traditional classroom. Furthermore, the wide use of assistive technologies for learners with disabilities injects new life into e-learning systems. E-learning allows distance and flexible educational activities, thus helping disabled learners to access resources which would otherwise present significant barriers for them. For instance, students with visual impairments have difficulties in reading traditional visual materials, deaf learners have trouble in following traditional (spoken) lectures, people with motion disabilities have problems in attending on-site programs.

As already mentioned, the use of wireless technologies and pervasive computing may really enhance the educational learner experience by offering mobile e-learning services that can be accessed by handheld devices. This new paradigm of educational content distribution maximizes the benefits for learners since it enables users to overcome constraints imposed by the surrounding environment. While certainly helpful for users without disabilities, we believe that the use of new mobile technologies may also become a fundamental tool for impaired learners, since it frees them from sitting in front of a PC. In this way, educational activities can be enjoyed by all the users, without hindrance, thus increasing the social inclusion of non-typical learners. While the provision of fully accessible and portable video-lectures may be extremely useful for students, it is widely recognized that structuring and managing rich media contents for mobile learning services are complex and expensive tasks. Indeed, major difficulties originate from the basic need to provide a textual equivalent for each media resource composing a rich media Learning Object (LO). Moreover, tests need to be carried out to establish whether a given LO is fully accessible to all kinds of learners. Unfortunately, both these tasks are truly time-consuming processes, depending on the type of contents the teacher is writing and on the authoring tool he/she is using. Due to these difficulties, online LOs are often distributed as partially accessible or totally inaccessible content.

Bearing this in mind, this thesis aims to discuss the key issues of a system we have developed to deliver accessible, customized or nomadic learning experiences to learners with different access needs and skills. To reduce the risk of excluding users with particular access capabilities, our system exploits Learning Objects (LOs) which are dynamically adapted and transcoded based on the specific needs of non-typical users and on the barriers that they can encounter in the environment. The basic idea is to dynamically adapt contents, by selecting them from a set of media resources packaged in SCORM-compliant LOs and stored in a self-adapting format. The system schedules and orchestrates a set of transcoding processes based on specific learner needs, so as to produce a customized LO that can be fully enjoyed by any (impaired or mobile) student.

Acknowledgements

I would like to thank Prof. Paola Salomoni for her invaluable support and advice. She was always ready to discuss new ideas; she always gave me insightful and very competent comments. I'm really grateful to her for the important academic guidance, the constant presence, the confidence in my abilities and more over.

Many thanks to Prof. Marco Roccetti for his fundamental and constant sustain and precious suggests. I feel actually honored for his assistance.

Prof. Salomoni and Prof. Roccetti made possible this Ph.D by supporting me from any point of view and by allowing me to learn and to know.

I'm very grateful to Dr. David Sloan and Dr. Gregory R. Gay for their kind willingness to review my thesis and for their important comments and suggestions. Their reviews and their very favorable judgments make me even more proud.

I would also like to thank the whole SWIMM Project; in particular I'm very grateful to Prof. Enrico Lodolo for his careful lead. Without his support the whole project would not have the same results in such deadlines. Many thanks to all SWIMMers, they have been great companions, we had enjoyable collaborations, interactions and coffee breaks too. A special thank goes to Daniele Pantieri for all our interesting and useful collaborations.

I would like to thank Dr. Stefano Ferretti for his daily support, the gratifying collaborations and suggestions.

Many thanks also to Prof. Giorgio Casadei, Dr. Stefano Cacciaguerra and Dr. Vittorio Ghini. Finally, I would also like to thank all my colleagues, above all the XIX course ones.

That was the academic part. However, I can’t help mentioning my parents and friends.

I would like to thank Vico for his constant, fast and helpful support, which he offered to me so many times. Many thanks to Boni Catering for the daily provision of food, coffee and thoughtlessness and to Monica Flowers for her careful assistance. As many thanks to Little Sara for her essential delivery service and for these years of chats. Other thanks to Marina, because she makes each day a spring day. Many thanks also to Erik for his technical support in the last difficult (and unfortunately often) moments and to Gabriele for his kindness. Thanks to Diego and Matilde, for her nice paints and touching cards.

I would also like to thank all my friends and in particular Debora for deep and long standing friendship and the cheerfulness Alessia, Ivan and she always give me.

Many thanks to my mother Lucia and my father Renato. They always bear, drive and encourage me. A special thank to my grandfather Livio for the unconditioned trust in me.

Last but certainly not least, many thanks to Luca, for his unique and unlimited support and for his constant presence during this long and not always easy time (and in particular for the nightly printer service). During all these years, we have been reach together so many finishing lines and as many will waiting for us.

Finally, remembering my grandmother Carolina: "*Bhè, tan'e brisa incora finì ed stuge'?*".

Contents

List of Figures

List of Tables

Chapter 1

1. Introduction

Offering an increasing access to a wider range of learners is usually considered one of the main benefits provided by e-learning systems [44]. However, on-line educational and training services are frequently based on anytime technologies that do not cope with “everyone” and “everywhere” dimensions [75]. Commonly, e-learning materials are designed to be used with a specific hardware device, with a particular software technology and a specific (fixed up) configuration. This is particularly true when e-learning materials are mainly based on rich media contents.

The term “Rich Media” is typically used to describe a broad range of interactive digital media that exhibit dynamic motion, taking advantage of enhanced sensory feature such as video, audio and animation. This motion may occur over time or in direct response to user interaction. Rich media is creating new opportunities in education [29] [125] [127]. For example, University of California provides courses and lectures through Google Video [135]. The integration of audio, video, and graphics within a browser has made possible new interactive forms and experiences for teaching and learning. Educators now have a wide variety of tools and systems to develop and deliver content live as well as on-demand to the students anywhere and anytime. This content can either be created by using a variety of sophisticated multimedia production practices or can simply be captured using VCR-like recording systems of actual classroom events. Either way, students benefit from vastly improved learning experiences or the flexibility to participate and interact like and when needed [135] [170].

As a consequence, learning content results as poorly available to those users who have unconventional access capabilities. Stated simply, technological barriers arise for:

i) students with disabilities, who typically use assistive and adaptive technologies to access to the PC and to the Internet [18] [28] [97], and
ii) students equipped with mobile devices (e.g., smart phones, PDAs) who are constrained by the limited capabilities (e.g., screen dimension, network bandwidth) of their workstations [27] [53] [85] [120] [142] [155].

Nowadays e-learning is one of the most inaccessible Web-based technologies and students with disabilities are frequently ruled out from virtual classrooms [14]. Instead, learners with disabilities may really benefit from e-learning due to their specific needs [73] [130]. For instance, students with visual impairments have difficulties in reading traditional printed materials, deaf learners have troubles in following traditional (spoken) lectures or, finally, people with motion disabilities have problems in attending on-site programs [122].

In order to further encourage the development of accessible e-learning platforms and contents, many countries have compelled accessibility by law, e.g., US [136], UK [134], Canada [132] and Italy [68].

Improving accessibility of learning resources can also result in an enhanced e-learning experience for mobile users [155]. In fact, making e-learning accessible ensures that learning materials are suitable to be enjoyed by all the learners, regardless of environmental or technological constraints. This also allows the accommodation of individual learning styles and preferences. To summarize, new learning paradigms are emerging which will be able to offer more intense and immersive learning experiences to students. Two main remarks drive this analogy: first of all it is obvious that a limited device restricts user capabilities so that a set of alternative strategies are needed to overcome these constrains. Secondly, context awareness is strictly related with device profiling, but it is not limited to it and it is important to consider that in a specific situation any user can be limited by the context. For example, a user need a different rendering of an e-lecture while he/she is carrying out an experimental trial in a laboratory and has sight and hands busy. Voice interaction is a clear example of technology that is used both to implement mobile learning [27] [53] [120] and to enhance e-learning accessibility.

In this context, different formats and transformation mechanisms have been proposed, which consider multimedia contents as simple flows or objects (for example embedded in Web pages). On the other hand, multimedia research is working on content adaptation with a media-centered point of view and new standards are described directly embedding adaptation mechanisms. Main literature on these topics is described in Section 2 of this dissertation.

To integrate rich and interactive multimedia in e-learning applications, different dimensions of the problem are currently missed in both mentioned approaches. First, rich media could not be considered just as interactive not-continuous elements (like hypertextual pages) or flows (like video or audio), but they are complex synchronous objects that combine interactivity with time and space constraints. Secondly in mobile learning applications the interaction between the user and the system must be influenced by different conditions: where you are, who you are and which resources are available to you. Context encompasses more than just the user's location, because other things of interest are also mobile and changing. Context includes lighting, noise level, network connectivity, communication costs, communication bandwidth, and even the social or personal situation of users.

1.1 Problem statement

On the plethora of use cases that e-learning users are typically engaged in, we are particularly interested in considering conditions which are strongly bound to learners' needs and devices capabilities. In such contexts, providing rich didactical materials to learners may cause problems, generating in some cases the loss of content information and of e-learning objectives as a whole too. On the other side, rich media actually improves e-learning experience and didactical materials [29] [127] and their use in teaching environments is continuously and constantly growing.

In order to avoid the loss of didactical information and to provide rich e-learning content to users who e-learn in non-typical bounded circumstances, it is necessary to adapt such rich content. The adaptation activity has to be planed by taking into account both users' needs and devices capabilities, in order to decide which transformations are needed.

With this in view, the aim of this thesis is to point out the main issues which are involved in applying transcoding strategies in order to produce device and user dependant didactical materials and how such a service might be best delivered. As already mentioned, such teaching resources are based on rich media content, which nowadays are widely used to enhance the quality and the effectiveness of e-learning inside a wide range of different situations [126]. We specifically refer to video-lectures that represent a complex rich media, widely diffused and easy to convey from traditional classroom lectures. Video-lectures are examples of rich media that express, in the same time, potentials and difficulty of providing complex multimedia content to who e-learns under limited conditions.

In order to prevent the exclusion of users with non-typical access capabilities, it is necessary to dynamically adapt and transcode Learning Objects (LOs). LOs transcoding and adaptation should be based on users' specific preferences and needs and on the technological barriers they could meet by using non-typical hardware or software platforms (assistive technologies, mobile devices, etc).

In this dissertation we present an approach for the design, the development and the evaluation of a system which is able to face the above mentioned issues, by providing a complete profiling mechanism that takes into account both learners' needs and preferences as well as devices capabilities. According to such a profiling approach, our system manages and transcodes multimedia resources so as to automatically produce multidevice suitably adapted presentations.

1.2 New Achievements of the Thesis

Based on such a context, the main novelties of this dissertation are summarized as follows:

i) both the learner and the device profiling are taken into account,
ii) sensorial overhead avoidance is guaranteed, when it is necessary (in other words, whenever any learner has sensorial disabilities or when the device does not support rich media formats) and
iii) rich media transcoding is done, with synchronicity maintenance or degradation in a feasible and efficient way.

In order to completely profile the learner's context, we have to consider and to combine data regarding any user needs and preferences and her/his device capabilities. Such a dual profiling becomes strategic whenever non-typical situations arise, as for example learners with sensorial and physical disabilities, mobile learners equipped with devices with limited capabilities, and, finally, learners with disabilities who are using limited devices. As a consequence, an effective mechanism to describe any user and device has to be adopted [119]. On one hand, learner description has to take into account his/her preferences and needs in order to tailor learning contents, by distinguishing preferred and required accommodations. A personal user profile has to provide a means to describe learners' interaction with an e-learning environment, in terms of sensorial and physical needs, context conditions and, finally, display, control and content information preferences. On the other hand, devices have to be described in terms of hardware capabilities, supported software and assistive technologies equipment. In literature (which is described in Chapter 2) several standards and solutions have been proposed. Unfortunately no one of them represents a whole and fully supported proposal, although a mechanism which combines such two aspects is needed. In Chapter 4, we describe our proposal in terms of any learner and device profiling, also showing some use cases.

Transcoding rich media content may produce the parallelization of more than one information flow on a specific sensorial channel. This represents a problematic side effect and causes a sensorial overhead in learners with sensorial disabilities. We can consider, for example, a blind user accessing a video-lecture composed of the teacher's talk and of some synchronous slides supporting the lecture. Two (synchronous) audio tracks are technically available:

i) the main audio track reproducing the talk and
ii) the audio track produced by voice synthesis reading slides content.

Usually, assistive technologies do not read textual contents that change dynamically; hence, the second track (and its related information) is lost. Similar cases may take place in several interesting conditions, included mobile contexts.

In order to face such an issue, the presented approach proposes a feasible mechanism (which is described in more detail in Chapter 5). The system checks the presence of a parallel and contemporaneous presentation of different tracks involving a specific human sense and unties colliding tracks, degrading the rich media synchronicity in order to obtain a continue resource or a sequence of discrete resources, without losing any didactical content.

Indeed, the need of degrading synchronicity may be also due to device capabilities and it could occur despite user abilities. Hence, the requirement of a user feasible approach in transcoding synchronous rich media e-learning content is emerging. Some encoding formats include issues inside their primitives in order to automatically offer support to synchronous alternatives. However, when such approaches are not naïf or easily usable, they show limitation on other fronts. In fact, generally only a static set of limited, pre-defined preferences is provided to the user; this hampers the development of sophisticated customization mechanisms able to select among alternative contents or to adequately transcode single media. Main related literature is presented in Chapter 2. In Chapter 5, we illustrate our proposal in terms of rich media adaptation and transcoding.

1.3 Outline of the thesis

In this Section we summarize the overall organization of this thesis:

- Chapter 2 introduces some backgrounds necessary to understand the remainder of the thesis. In particular, the Chapter recalls concepts related to content adaptation and transcoding (presenting literature related to architectural, scheduling processes and multimedia adaptation issues), accessibility, e-learning and content negotiation.
- Chapter 3 presents main issues related to learners and devices profiling, in order to obtain a complete metadata and information to set didactical material use context.
- Chapter 4 illustrates main issues related to rich media adaptation, in order to obtain the most suitable synchronization degradation and to avoid sensorial overhead in learners with disabilities, by transcoding single media and/or the whole presentation.

- Chapter 5 describes the main system architecture issues, in order to transcode LOs, meeting learners' needs and their device capabilities. In particular the Chapter presents how such a system works and an implementation of it.
- Chapter 6 reports experimental results which assesses the performances of the presented system. Due to the peculiarities of such a system, three notable aspects result to be of interest in our investigation: transcoding facilities on single media resources, efficacy of having distributed all transcoding facilities and the efficacy of our caching system.
- Chapter 7 concludes the thesis by summarizing the obtained results and by outlining future researches.

Chapter 2

2. Background

The aim of this Chapter is to point out the fundamental concepts at the basis of the work presented in this thesis and to introduce the main key subjects in literature, which are involved in such a work.

First, Section 2.1 are going to present main content adaptation and transcoding issues, by illustrating typical choices from an architectural point of view, also by considering standards which are devoted to multimedia synchronization and (sometimes basic) adaptation. Second, in Section 2.2, we describe main accessibility topics, regarding rich media and Web resources; then we introduce accessibility standards and international laws. Third, Section 2.3 presents some e-learning issues, describing main e-learning standards and accessibility key issues. Final, in Section 2.4, we discuss standards useful in content negotiation and device identification.

2.1 Content Adaptation and Transcoding

The growing diffusion of devices coupled with the ability to deliver information anywhere at any time has improved the user's flexibility and the quality of services. It has also created a need for the development and deployment of new infrastructures supporting multiple platforms. As a result, new techniques for delivering content according to device features and even specific languages have emerged [10] [57] [76] [103].

In 1991 Weiser announced the era of ubiquitous computing and described a vision of proliferation of computational resources that provide access to information when and wherever desired [145]. This proliferation has indeed occurred, with a wide range of commonly used devices such as mobile phones, personal digital assistants (PDAs), palmtops or laptops.

Different technical criteria of wired and wireless networks and devices require different applications. Developing applications for mobile devices is particularly challenging because of a high network error rate, small usable keypads and screen on the devices, browsers incompatibility, short battery life, limited network bandwidth, etc.

Adapting typical Web content and services for PCs to small devices is one of the content adaptation hot topics [32] [33] [86] [168]. As further wireless networks evolve into their third generation, the number of available devices will grow. Information presentation on mobile devices needs to address the shortcomings of wireless appliances with small display sizes, different features for data input, limited graphics, etc. In order to display the same amount of information, a different number of pages may be needed depending on the device type [128].

The main obstacles to the pages interoperability are as follows: possible application bugs; some devices don't support functions, such as new mobile phones that only support Java and non-standard proprietary markup language extension. The final result is that the same page might have a great variety of appearances and could run in several ways, depending on the platform and device [47].

So content adaptation and transcoding are necessary and should be based on information such as the device capabilities and preferences, the network characteristics and some application-specific parameters; therefore, Web content and applications should be generated or adapted for a better user experience [52]. Device independence principles [165] are independent from any specific markup language, authoring style or adaptation process.

Device independence also offers users other kinds of benefits. For example, accessibility is a fundamental concern, and in some countries a legal requirement [68] [132] [134] [136]. Users must be able to interact with the Web in ways that suit their abilities [15] [95]. Offering options that let users replace images with text, present text as speech, or interact using voice or special input devices can benefit a wide range of users [56] [124] [129]. Different circumstances might also alter the way the users want to interact. A user in a car, for example, might switch from visual to audio-only interaction while driving.

According to the W3C definition [146], *content adaptation* is the transformation and the manipulation of contents (such as images, audio, videos, texts and presentations) to meet desired targets (defined by the terminal capabilities and the application needs) [30] [52]. Such adaptations include: format transcoding (e.g. eXtensible Markup Language - XML [151] to HyperText Markup Language - HTML [153], Scalable Vector Graphics - SVG [158] to GIF), scaling (of images as well as video and audio streams), media conversion (e.g. text-to-speech), resampling, file size compression and document fragmentation [55].

Transcoding is the process of converting a media file or object from one format to another [24]. This process is typically used to convert video, audio and image formats, but it is also used to adapt multimedia presentations and Web pages to the constraints of non-standard devices, e.g. the mobile devices. It is well-known that mobile devices have limited capabilities, such as smaller screen sizes, lower memory and slower bandwidth rates [128]. But most existing multimedia presentations and Web pages are created to be displayed on desktop computers and, usually, Web designers provide complex, detail-rich content, with multimedia experiences. Thus in mobile environments, transcoding must face the diversity of mobile devices. This heterogeneity imposes an intermediate state of content adaptation to ensure a proper presentation on each target device [32] [33] [86] [103] [168].

We can summarize content adaptation and transcoding operations on single media as follows [25] [55] [80] [81] [94] [106] [111]:

- *Transformation*: the conversion of content from its original form to another. Transformations can be performed automatically, depending on the type of conversion e.g., Text to Speech (TTS) or animation to image. Other kinds of transformations, however, need a predefined explicit declaration of content equivalence (made off-line), like in the case of translation from image to text. The conversion can also be done between the encoding formats of the same media type (e.g. audio files from WAV to MP3).

- *Scaling*: recoding and/or compressing specific media content. Scaling has effects in terms of reduction of size, quality and data rate of contents. Examples of scaling are image and video resizing, audio re-coding and compression.
- *Translation* from the original language to a different one, based on the user profile. This operation is only performed for textual and audio speech contents.

2.1.1 Architectural Approaches

Due to different device capabilities, content adaptation and transcoding need to be implemented before the content is presented to the user.

HTML [153] is not a device independent markup language because of its mixture of elements defining content and presentation. A good device independent application allows the content to be specified in a unified, optimized way on many different kinds of devices [17]. One way, according to the device independence principles, is to use any styling languages Cascading StyleSheet (CSS) [149] or the eXtensible Stylesheet Language (XSL) [152] to add style and presentation information to the content written in XML [151]. The Web output will then have a suitable content format for a non-usual browser.

The major technical requirement for access to information systems from various devices is the presentation of information in multiple formats and content tailoring to the capabilities of any particular device types. Mobile and wired devices are equipped with browsers that support various media formats.

An intuitive solution to the problem of device-dependent content delivery could be the appliance of many different views on the same data and apply them according to the formats supported and the presentation features of devices. Data must therefore be delivered in different markup languages such as WML [140], XHTML [163] or HTML [153]. This approach has, however, many shortcomings. It results in rewriting applications for various browsers, markup languages and device types, maintaining large code bases and gathering design expertise at least for the most popular appliances available on the market. In order to avoid creating separate user interfaces for each type of device, alternative techniques have to be considered.

Another approach is to retrieve data from an information system in XML format and to convert it to the appropriate markup language with eXtensible Stylesheet Language Transformations (XSLT) [164].

A transformation expressed in XSLT describes a set of rules for converting the input (source) document tree into a structure called a result tree, consisting of result objects. The conversion is achieved by associating patterns with templates.

Each template matches various sets of elements in the source tree and then describes the contribution that the matched element makes to the result tree. In constructing the result tree, elements from the source tree can be filtered and reordered, furthermore, new elements can be added. Using XML and XSLT in order to generate appropriate markup elements separates content from presentation and allows the same data to be presented in different ways. It enables us to reuse fragments of data, as well as generating multiple output formats and styles tailored to the device types. The most important drawback of this method is the need to maintain numerous stylesheets and to update each stylesheet separately if the view changes [10].

From an architectural point of view, four categories should be mentioned that represent the most significant distributed solutions for content adaptation [30] [80], i.e.:

i) *client-side* approaches,
ii) *server-side* approaches,
iii) *proxy-based* approaches and
iv) *service-oriented* approaches.

2.1.1.1 Client-side approach

In a *client-side* approach, the transcoding process is the responsibility of the client application, as Figure 2.1 shows.

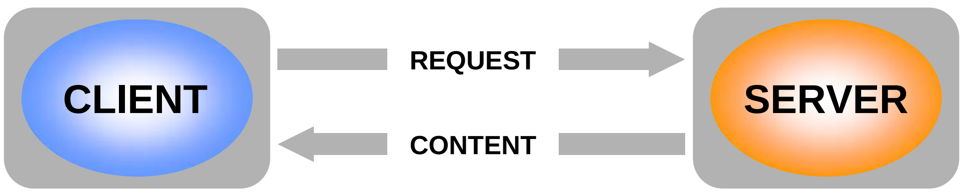

Figure 2.1 Client Based Adaptation

Client-side solutions can be classified into two main categories [30] [80] with different behaviours:

1. the clients receive multiple formats and adapt them by selecting the most appropriate one to play-out, or
2. the clients compute an optimized version from a standard one. This approach suggests a distributed solution for managing heterogeneity, supposing that all the clients can locally decide and employ the most appropriate adaptation to them.

2.1.1.2 Server-side approach

In a *server-side* approach, the server (that provides contents) performs the additional functional of content adaptation [30] [80] (Figure 2.2). In such an approach, content adaptation can be carried out in an *off-line* or *on-the-fly* fashion.

Figure 2.2 Server Based Adaptation

In the former, content transcoding is performed whenever the resource is created (or uploaded on the server) and a human designer is usually involved to hand-tailor the contents to different specific profiles. Multiple formats of the same resources are thus stored on the server and they are dynamically selected to match client specifications. In all the on-the-fly solutions, adapted contents are dynamically produced before delivering them to the clients.

2.1.1.3 Proxy-based approach

In *proxy-based* approaches, the adaptation process is carried out by a node (i.e. the proxy) placed between the server and the client [30] [80] (Figure 2.3). In essence, the proxy captures replies by the server to the clients requests and performs three main actions:

1. It decides whether performance enhancements are needed.
2. It performs content adaptations.
3. It sends the adapted contents to the client.

Figure 2.3 Proxy Based Adaptation

To accomplish this task as a whole, the proxy must know the target device, the user capabilities (this information must be received from the client) and a "full" version of the original contents (this data must be received from the server). As a consequence, the use of network bandwidth could be intensive in the network link between the proxy and the server.

2.1.1.4 Service-based approach

The dynamic nature of adaptation mechanisms together with emerging opportunities offered by the new Web Service technologies, now provide a new approach of *service-oriented* content adaptation [30] [80] (see Figure 2.4).

The philosophy at the basis of these approaches is fundamentally different from those previously discussed, since the transcoding and the adaptation activities are organized according to a service-oriented architecture. Indeed, the number of content adaptation typologies, as well as the set of multiple formats and related conversion schemes is still increasing. This dynamism is one of the reasons that makes it difficult to develop a single adaptation system that can accommodate all the types of adaptations; therefore, third-party adaptation services are important.

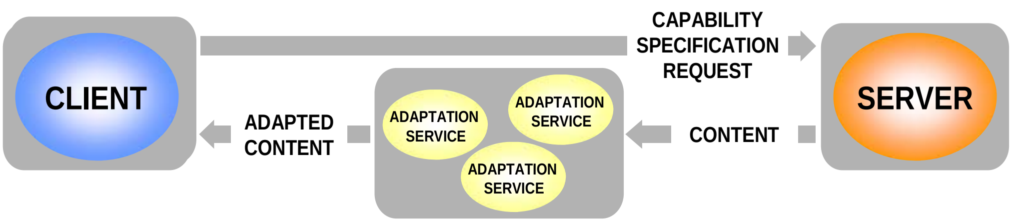

Figure 2.4 Service Based Adaptation

The Internet Content Adaptation Protocol (iCAP) [37] is closely related to this approach. ICAP distributes Internet-based content from the origin servers, via proxy caches (iCAP clients) to dedicated iCAP servers. For example, simple transformations of content can be performed near the edge of the network instead of requiring an updated copy of an object from an origin server, such as a different advertisement by a content provider, every time the page is viewed. Moreover, it avoids proxy caches or origin servers performing expensive operations by shipping the work off to other (iCAP) servers. However, it only defines a method for forwarding HyperText Transfer Protocol (HTTP) messages, i.e. it has no support for other protocols and for streaming media (e.g. audio/video) and only covers the transaction semantics and not the control policy.

2.1.2 Adapting Multimedia

The diversity of the multimedia presentation environment imposes strict requirements on multimedia applications and systems [70] [108]. The emerging growth of mobile services (together with wireless technology such Bluetooth, 802.11, GPRS and UMTS) defines more requirements for the content and service providers [103]. Content, terminal capabilities and underlying networks demand separate service creation processes and mobile services require support for new billing and profiling mechanisms based on the user and the service at hand [47] [78] [81]. In particular, as these devices are becoming more multimedia capable, one of the interesting challenges is the multimedia content delivery on these embedded devices [86].

2.1.2.1 SMIL

Several attempts have been made to standardize the presentation environment and the presentation format for mobile service delivery. Markup languages such as the XML (Extensible Markup Language)

[151] and its applications like SMIL (Synchronized Multimedia Integration Language) [159] developed by the World Wide Web Consortium (W3C) [145], can be applied in modelling structured, document-like multimedia presentations [87]. SMIL plays the same role in a SMIL player that HTML plays in a Web browser (namely providing information on how to layout and format a page). A SMIL presentation can consist of multiple components of different media types (such as video, audio, text, and graphics) linked via a synchronized timeline. For example, in a slide show the corresponding slide can be displayed when the narrator in the audio starts talking about it.

SMIL 2.0 is the main representation in Web technology for describing timing and synchronization of multimedia presentations. Careful attention has been paid, in the design of SMIL, to modularity and extensibility of the recommendation and three language profiles have been proposed. Most notably, SMIL Basic profile is a collection of modules together with a scalable framework, which allows a document profile to be customized for the capabilities of the device. Providing an adaptive content is still under investigation, as some general mechanisms such as content negotiation, universal profile (document, user, network, and terminal) descriptions and processing are not well established yet [82].

SMIL 2.1 [159] is defined as a set of markup modules, which define the semantics and XML syntax for certain areas of SMIL functionality. This specification provides three classes of changes to SMIL 2.0, among the ten functional areas; in particular new models are introduced, former SMIL modules are deprecated and replaced by new ones to allow differentiated features to be implemented in profiles, without necessarily requiring support for all of the functionality of the former SMIL module and former SMIL Modules are revised allowing extended functionalities. All these changes are related to the use of SMIL through mobile devices.

Several simple content selection mechanisms have been introduced in SMIL to provide greater flexibility. However, in most cases, SMIL adaptation is achieved at the client side. This supposes that the client is adaptation-capable and that the profiles and the client capabilities are somehow set. In addition, adaptations do not necessarily belong to the same layer of a document presentation. One can start by designing a device-independent document layer and generate, once the profiles are identified, the SMIL content representation.

It is also possible to perform adaptation within a SMIL document instance beyond the mechanisms which are provided by the format and to modify the content itself to fit bandwidth and display limitations. In fact, SMIL language itself contains an "adaptation" or "alternate content" mechanism. Using the `<switch>` tag and "test attributes" it is possible to have a SMIL player choice between alternative content. Examples of attributes that the player can use, are "`systemBitrate`" to select content that fits the current network bandwidth, "`systemCaptions`" to choose between video with or without captions, "`systemLanguage`" to select content in a given language, "`systemScreenDepth`", "`systemScreenSize`", etc [159]. These adaptation features enable a

SMIL player to fit to technical circumstances and some fairly static user preferences. SMIL integrates both HTML and SVG to add timing features to pages and vector graphics respectively. SMIL with SVG elements offers support for Web animations [158].

2.1.2.2 MPEG-21

MPEG-21 [93] is an open standards-based framework for multimedia delivery and consumption by all the players in the delivery and consumption chain [16]. It is the newest of a series of standards being developed by the Moving Picture Experts Group, after a long history of producing multimedia standards. The goal of MPEG-21 can thus be redefined as the technology needed to support users to exchange, access, consume, trade and otherwise manipulate Digital Items in an efficient, transparent and interoperable way. Interoperability is the driving force behind all multimedia standards. It is a necessary requirement for any application that requires guaranteed communication between two or more parties. Interoperability expresses the users' dream of easily exchanging any type of information without technical barriers.

The basic concepts in MPEG-21 relate to *what* and *who* within the multimedia framework. *What* is a Digital Item, i.e. a structured digital object with a standard representation, identification, and metadata within the MPEG-21 framework. *Who* is a user who interacts in the MPEG-21 environment or uses a Digital Item, including individuals, consumers, communities, organizations, corporations, consortia, governments and other standards bodies and initiatives around the world [93]. The users can be creators, consumers, rights holders, content providers or distributors, etc. There is no technical distinction between providers and consumers: all parties that must interact within MPEG-21 are categorized equally as users. They assume specific rights and responsibilities according to their interaction with other users. All users must also express and manage their interests in Digital Items [92].

In practice, a Digital Item is a combination of resources, metadata, and structure. The resources are the individual assets or content. The metadata describes data about or pertaining to the Digital Item as a whole or also to the individual resources in the Digital Item. The structure relates to the relationships among the parts of the Digital Item, both resources and metadata. For example, a Digital Item can be a video collection or a music album. The Digital Item is thus the fundamental unit of distribution and transaction within the MPEG-21 framework [92].

MPEG-21 is organized into several independent parts, primarily to allow various slices of the technology to be useful as stand-alone. This maximizes their usage and lets the users to implement them outside MPEG-21 as a whole, in conjunction with proprietary technologies. The MPEG-21 parts already developed or currently under development are as follows:

1. *Vision, technologies, and strategy*: this part describes the multimedia framework and its architectural elements with the functional requirements for their specification.
2. *Digital Item Declaration (DID)*: this second part provides a uniform and flexible abstraction and interoperable framework for

declaring Digital Items. By means of the Digital Item Declaration Language (DIDL), it is possible to declare a Digital Item by specifying its resources, metadata, and their interrelationships.

3. *Digital Item Identification (DII)*: the third part of MPEG-21 defines the framework for identifying any entity regardless of its nature, type or granularity.
4. *Intellectual Property Management and Protection (IPMP)*: this part provides the means to reliably manage and protect content across networks and devices.
5. *Rights Expression Language (REL)*: this specifies a machine-readable language that can declare rights and permissions using the terms as defined in the Rights Data Dictionary.
6. *Rights Data Dictionary (RDD)*: this is a dictionary of key terms required to describe users' rights.
7. *Digital Item Adaptation (DIA)*: this identifies all the description tools for usage environment and content format features that might influence transparent access to the multimedia content (notably terminals, networks, users and the natural environment where users and terminals are located).
8. *Reference software*: this includes software that implements the tools specified in the other MPEG-21 parts.
9. *File format*: defines a file format for storing and distributing Digital Items.

10. *Digital Item Processing (DIP)*: this defines mechanisms for standardized and interoperable processing of the information in Digital Items.
11. *Evaluation methods for persistent association technologies*: documents best practices in evaluating persistent association technologies using a common methodology (rather than standardizing the technologies themselves). These technologies link information that identifies and describes content directly to the content itself.
12. *Test bed for MPEG-21 resource delivery*: this last part provides a software-based test bed for delivering scalable media and testing/evaluating this scalable media delivery in streaming environments.

2.1.1.2.1 Digital Item Adaptation

This seventh part of MPEG-21 [92] specifies all the tools for the adaptation of Digital Items. One of the goals of MPEG-21 is to achieve interoperable transparent access to (distributed) advanced multimedia content by shielding users from network and terminal installation, management, and implementation issues [93]. Achieving this goal requires the adaptation of Digital Items (see figure 2.5) [137]. As shown in this conceptual architecture, a Digital Item may be subject to a resource adaptation engine, a description adaptation engine, or a DID adaptation engine, which produces the adapted Digital Item [92].

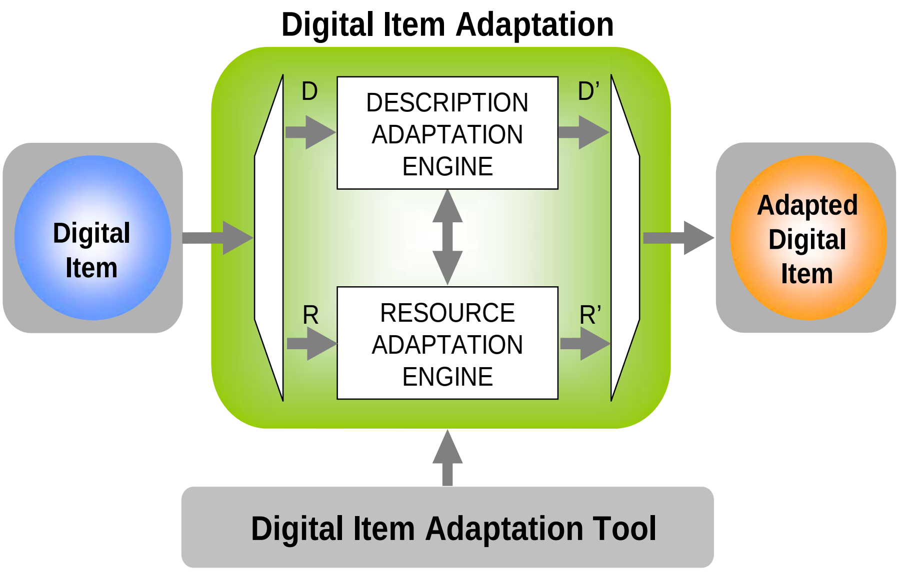

Figure 2.5 Digital Item Adaptation Architecture

The usage environment description tools describe the terminal capabilities (such as codec and input-output capabilities, and device properties) as well as network characteristics (such as network capabilities and network conditions), user (for example user info, usage preferences and usage history, presentation preferences, accessibility characteristics, including visual or audio impairments, and location characteristics) and natural environment. In this context, natural environment relates to the physical environmental conditions around a user such as lighting or noise levels, or circumstances such as the time and location [137].

This part of MPEG-21 [92] also includes the following specific items:

- *Resource adaptability*: tools to assist with the adaptation of resources, including the adaptation of binary resources in a generic way and metadata adaptation. In addition, tools that assist in making resource complexity trade-offs and associations between descriptions and resource characteristics for Quality of Service are also targeted [137].
- *Session mobility*: tools that specify how to transfer the state of Digital Items from one user to another. More specifically, the capture, transfer and reconstruction of state information.

2.2 Accessibility

The term "*accessibility*" usually points out the facility of computer systems to provide information and services to people who access them by using assistive technologies or special computer configurations often necessary to accommodate a disability [113] [123].

Assistive technologies (both hardware and software ones) have been designed and developed to make Personal Computers accessible to people with disabilities, in order to promoting integration in everyday life, education, and work [142].

2.2.1 Rich Media Accessibility

Even if rich media presents numerous accessibility challenges, they can be made accessible if all the elements are developed with accessibility in mind and the end product is used or viewed on accessible media players. Accessible rich media typically includes captioning, audio description, and navigation using a keyboard [125].

Accessible media players are those that can be operated by all users, including those using assistive technologies. They must also provide authors with the means to add captions, audio descriptions, extended audio descriptions, and subtitles [147]. The current level of accessibility for media players creates interesting situations. Some media players allow video descriptions to be created and played but have an inaccessible interface that users of screen readers cannot operate [11].

Moreover captions may look different when created on one player and then played back on another. For instance, captions developed using QuickTime may look fine when viewed in QuickTime but then they appear larger or smaller when later viewed in RealPlayer.

Several media players have also made considerable progress in improving accessibility of their products. The National Center for Accessible Media [96] provides information and tutorials on captioning audio, descriptive video, making maps and other forms of rich media accessible; strategies for dealing with player and cross-platform issues; links to tools for rich media authoring and viewing; links to latest news; and much more.

2.2.2 Web Accessibility

The explosive growth of Internet services has had a great impact on people's lives. The Internet is making distances smaller and smaller, connecting people anytime, anywhere and reaching to the far corners of the earth. Access to the Internet and Web resources is becoming a part of everyday life for a large portion of the population in the developed world: in employment, education, health care, commerce, and recreation. As such, "an accessible Web can also help people with disabilities more actively participate in society" [107]. Printed information or content that is delivered through audio or video media will often be inaccessible for some groups of people with sensory impairments. The Web can make media available to these individuals through alternative formats such as text, captioning, and descriptive audio [143]. Furthermore, the availability of services and information on the Web can help people with mobility impairments overcome difficulties of physically reaching onsite services. The Web often allows these individuals to bypass the limits of their disabilities [11].

Web accessibility also provides benefits to other groups of users in addition to those with disabilities, including:

- older people with age-related changes in ability,
- people using non-conventional devices, such as PDAs or smart phones to access the Internet,
- people in areas of the world where the Web access bandwidth is limited,
- people who are working in situations where their senses or hands are busy, for example, while driving or watching a video in noisy surroundings.

Another dimension of Web accessibility is the responsibility of Web authors, Web developers, Web designers, and technologies they use to develop Web content [123] [133]. Many organizations have been working on defining guidelines that ensure that Web Content will be accessible and their efforts have resulted in a Web accessibility specification created by the Web Accessibility Initiative [142] of the World Wide Web Consortium [145]. They have produced the first set of accessibility principles to be accepted worldwide.

2.2.3 Standards and laws

Many guidelines and requirements have been defined to support the production of accessible Web applications and Web content. The W3C has leaded the Web Accessibility Initiative (WAI) since 1997, which develops guidelines and resources specifically devoted to Web accessibility [142]. The best-known document produced by this group is Web Content Accessibility Guidelines (WCAG), which defines a collection of authoring guidelines related to several main themes of accessible design [161] [162]. The guidelines make recommendations that foster the development of accessible Web content, such as providing equivalent alternatives to non-textual content and using appropriate markup and style sheet elements [74]. WCAG 1.0 [161] directly refers to practical techniques that explain and define how to design and implement accessible HTML and CSS based content [124], while WCAG 2.0 [162] is intended to be technology independent and applied to all Web technologies.

Other WAI guideline documents make recommendations for developing:

i) accessible user agents, including Web browsers, media players and assistive technologies (User Agent Accessibility Guidelines “UAAG”) [160] and
ii) accessible authoring tools that produce accessible content (Authoring Tool Accessibility Guidelines, “ATAG”) [148].

All the W3C Recommendations could be considered as a worldwide reference for Web accessibility, though their use has been primarily voluntary.

In order to promote the ethical issues associated with inclusion, accessibility is frequently encouraged and often enforced by law. Many countries have added regulations to existing accessibility laws, including chapters related specifically to Web accessibility. In 1998 the United States Government added ICT (Information and Communication Technologies) accessibility, through Section 508 of the Rehabilitation Act [136], imposing hardware, software and Web accessibility constraints upon federal agencies and their suppliers. Similarly, the Canadian Government made accessibility mandatory for federal government Web sites by enacting The Common Look and Feel for the Internet legislation [132] in 2000. In the same year, the European Community raised the profile of accessibility in information technologies with the e-Inclusion policy, one of seven “eEurope policy priorities”, intended to sustain participation of all those in the knowledge-based society [38]. In addition, several European countries, like Italy [68], the UK [134], Germany, Portugal and Spain, have enacted their own rules or guidelines to ensure the accessibility of Web content. The number of countries that are addressing Web accessibility issues continues to grow and is being monitored by the W3C.

All the above mentioned guidelines, laws and requirements are based on (X)HTML accessible authoring practices. Two of them are presented below. This non-comprehensive list presents the main practices associated with accessible authoring (mostly related to content adaptation and transcoding subjects) [161] [162]:

1. To provide alternative formats for all non-text content, including graphical information, multimedia, and programmed objects. A user may not be able to use a specific media format due to a sensory disability (e.g. a blind user cannot see an image) or may have difficulties in accessing a resource that requires the use of a specific plug-in or helper application. Audio tracks should be supplemented with synchronized captioning or a transcript and video should include captioning and descriptive audio: the latter used to describe information that cannot be deduced from the audio track of a video. Interface elements in plug-in or add-on software must include a text label so they can be read by assistive technologies. The most common alternative format is the (X)HTML Alt attribute, used to provide a short text description of something visual.
2. Design for device independence, creating Web pages that are accessible both with a mouse and a keyboard. Some people with disabilities may have difficulties using a keyboard, perhaps due to

mobility impairment and others, such as blind users, may be unable to use a mouse. Plug-in or add-on software used to play multimedia content must also be usable in a device-independent manner. A good test is to use the Tab key repeatedly to see if all Web site or interface elements can accessed.

2.2.4 Disabilities and Assistive Technologies

Accessibility is generally dependent on assistive technologies used by people with disabilities to access their PCs, but it also depends on whether people with various disabilities can perform specific tasks on their PCs with the help of their assistive technologies [28] [56].

In this Subsection we introduce how people with disabilities access the Web [142], by considering a few examples of specific disabilities and the assistive technologies that might be used. The examples are not an exhaustive list, but are intended to offer a short overview of some of the more relevant cases, where the type of disability has a significant affect on a person's ability to access the Web [166].

First we consider people who are blind, who will most likely use a *screen reader* to access their computers. A screen reader gathers information from a computer screen and outputs that information as synthesized speech [45]. While accessing the Web, a screen reader may encounter a variety of barriers, such as uncommented images or information whose meaning depends on colour or its position on the screen. Visual information without text alternatives that can be read by a screen reader will otherwise not be accessible to a screen reader user [142].

Similarly, people who have poor vision may use an assistive technology that enhances their residual sight, such as a *screen magnifier*. This tool enlarges the information displayed on the screen and helps the user by appropriately modifying some of its characteristics such as font size, contrast, or colours [15]. The resulting display from a screen magnification tool represents only a portion of the whole screen, which often creates a loss of context. To accommodate those using a screen magnifier authors need to create content that can be easily resized. In addition, it is useful using relative measures (e.g. em, %) instead of absolute measures (e.g. pt, px) to define the size characteristics of their content, allowing it to resize, to fit any size browser window without loosing or distorting the information being presented [31].

Mobility related difficulties range from simply being unable to grasp or handle a mouse, to disabilities that require the use of voice input to control a computer instead of the traditional keyboard and mouse. Generally people with mobility impairments need Web pages that can be fully accessed by using a keyboard or mouse-equivalent input device. A head mouse and single click switches might be used in place of a traditional mouse, controlling the cursor with head movements and clicking on the mouse by leaning on a large button like switch. Alternative mouse input devices might be used together with an onscreen keyboard or voice recognition system for navigating and entering content [166].

Finally, we can consider the less obvious or hidden disabilities such as learning disabilities and dyslexia, disabilities that affect a user's ability to read, write, navigate, comprehend and recall relevant information. People who have such disabilities may use a text-to-speech system that reads text on the screen aloud using synthesized speech [18]. In addition, multimedia are widely used as an accessibility solution for anyone who has difficulty reading and/or understanding information presented in text form [125].

Consistency in presentation is often an important aspect for improving accessibility and usability for those with learning disabilities, such as navigation tools that remain the same throughout a Web site, a consistent look-and-feel, and page layouts that do not change from screen to screen [142].

2.3 E-learning

The evolution of an Information Society has transformed many activities in our everyday lives, including how we work, communicate, entertain, teach and learn [114]. More specifically, in recent years widespread Internet connectivity, together with the development of new Web-based multimedia technologies, has strongly encouraged educational uses of ICT (Information and Communication Technology). All activities that need network technologies to deliver learning and training programs can be considered forms of "*e-learning*" [36]. ICT naturally fuelled the spread of e-learning, forcing the emergence of a society in which economic and social performances are largely judged by a continuous distribution of information and knowledge [127]. E-learning currently represents the most widespread form of "*Distance Education*", which generally refers to educational activities that involve teachers and students remotely located both in time and space. Current distance education is based on a wide range of delivery methods, including traditional correspondence, as well as books, audio/video tapes, interactive TV, CD-ROM and DVD, as well as services that can be offered through the Internet [110]. More generally, "*e-learning*" can be defined as the delivery of education or training programs through electronic means [58].

From a technological point of view, today's e-learning is rooted primarily in a Web-based delivery of educational multimedia content, coupled with synchronous and asynchronous communication features that allow students and teachers to interact [13] [54] [59] [127]. In addition, new e-learning forms are emerging, increasing nomadic and ubiquitous access [27] [53] [120], such as narrowcasting based ones [21].

2.3.1 Standards

A standard description of content structure is needed to ensure that content will be interoperable across different e-learning platforms. Several interoperability specifications have been developed by international organizations such as [8]:

- The IEEE (Institute of Electrical and Electronics Engineers), with a specific working group, the Learning Technology Standards Committee, which is working on e-learning standardization [60].
- The IMS (Instructional Management System) Global Learning Consortium, a collaboration of government organizations that are defining specifications to ensure interoperability between e-learning products [62].
- The ADL (Advanced Distributed Learning) initiative [1], lead by the U.S. Department of Defence, which has developed the SCORM (*Shareable Content Object Reference Model*) standard, one of the more widely used e-learning specifications. ADL has based its work on that of IEEE and IMS, and has created a more encompassing interoperability standard that takes into consideration recommendations from those and other standards [5].
- The AICC (Aviation Industry CBT – Computer Based Training – Committee) [6], which is an international association of technology-based training professionals and develops AICC's AGR's (AICC Guidelines and recommendations) [7]. Such specification defines both hardware and software requirements in CBT environments.

The goal of such standards is to define metadata, data structures, and communication protocols that will make learning content work across different platforms, by providing specific guidelines to be used throughout the design, development and delivery of learning content.

In order to describe general learner characteristics, the IMS Learner Information Profile (IMS LIP) [66] is devoted to define a set of packages that can be used to import data into and extract data from an IMS compliant Learner Information server. The main aim of such a specification is to address the interoperability of Internet-based Learner Information systems with other systems that support the Internet learning environment. By using LIP, it is possible to define learner's information about: accessibilities; activities; affiliations; competencies; goals; identifications; interests; qualifications, certifications and licences; relationship; security keys; and transcripts.

Another relevant role is played by the de-facto standard SCORM (Shareable Content Object Reference Model) [5], which is based on some specifications previously defined by IEEE-LTSC and IMS. SCORM includes a de-facto standard for defining a SCO (Sharable Content Object). A SCO is a learning resource that can be presented in any SCORM compliant system, displaying and sequencing content, and tracking student progress. Each SCO is made up of one or more assets or resources, which are electronic representations of media (e.g. text, images, sound, video), web pages or other types of data. An SCO can be described with metadata and found by searching for terms in the metadata in online content repositories, thereby enhancing opportunities for their re-use.

Metadata and structural information about a unit of learning content is usually contained within a "*manifest*", an XML file that describes the learning content in a standard manner. A SCORM manifest generally contains the content's semantic description (metadata), together with its navigation or structural description (organizations), and the locations of each of the contained assets (resources). The SCORM main specifications are [5]:

- The Content Aggregation Model (CAM) [2] that defines the content structure and describes the content with metadata (based on the IMS Content Packaging specification).
- The Run-time Environment (RTE) [3], a JavaScript API (Application Programming Interface) that delivers real time information to the Learning Management System (LMS)/Learning Content Management System (LCMS) about user actions within a SCO, including exercise solving and tracking through resources.
- The Sequencing and Navigation (SN) [4] specification describes rule-based definitions of possible paths through learning content.

The collected standards can be applied to learning content and to learning platforms, e.g. Learning Management System (LMS), Learning Content Management System (LCMS) and Virtual Learning Environment (VLE) with the aim of fully supporting the reuse of content across systems.

2.3.2 E-learning Accessibility

E-learning materials are often used with a specific technology, or configuration, making them less available to people who have limited access capabilities or are using non-standard computer equipment.

Learners with disabilities using assistive technologies can greatly benefit from e-learning, not just because it allows distance and flexible learning activities, but also because it helps students with disabilities to access resources which would otherwise present significant barriers for them [9] [14] [46] [48] [64] [71] [72] [88] [89].

New e-learning paradigms will consider student's individual abilities and learning goals, where learning is occurring, and through which particular device learning is taking place. Learning will be adapted for each individual learner [67] [115] [116] [117] [118]. The IMS Global Learning Consortium [62] has developed a sub-specifications that attempt to address the personalization or transformation of e-learning content: the IMS Accessibility Learner Profile (IMS ACCLIP) [65], which is a part of IMS LIP [66], is devoted to describing students' accessibility constraints [51]. ACCLIP describes the user in terms of accessibility needs, without considering the device characteristics. ACCLIP enables the description of user preferences (visual, aural or device) that can be exploited for tailoring learning contents (e.g. preferred/required input/output devices or preferred content alternatives). In other words, this personal user profile provides a means to describe how learners interact with an e-learning environment, by focusing on accessibility requirements. The ACCLIP Specification defines the required elements to represent accessibility preferences, which can be grouped into four sections:

- *display information* (`<display>`), which describe how the user prefers to have information displayed or presented; for example, it is possible to define preferences related to cursor, fonts and colors characteristics (`<cursorSize>`, `<fontFace>`, `<fontSize>`, `<cursorColor>`, `<foregroundColor>`, `<backgroundColor>`). In addition, it is possible to declare the need of using a screen reader (`<screenReader>`), specifying the interaction preferences, such as the speech rate, the pitch and the volume (`<speechRate>`, `<pitch>` and `<volume>`), or the need of visual alerts instead of aural ones (`<visualAlert>`);

- *control information* (`<control>`), which define how a user prefers to control the device; for example, it is possible to define preferences related to standard keyboard usage (`<keyboardEnhanced>`). In addition, it is possible to declare the need of using non typical control mechanism, such as onscreen keyboard (`<onscreenKeyboard>`), alternative keyboard (`<alternativeKeyboard>`), mouse emulation (`<mouseEmulation>`), alternative pointing mechanism (`<alternativePointing>`) and voice recognition (`<voiceRecognition>`);
- *content information* (`<content>`), which describe what enhanced, alternative or equivalent content the learner requires; for example, it is possible to define how to present visual, textual and auditory contents in different modalities (`<alternativesToVisual>`, `<alternativesToAuditory>`, `<alternativesToText>`) and the need of personal style sheets (`<personalStylesheet>`);

- *accommodations* (`<accomodation>`), which allow recording of requests for and authorization of accessibility accommodations for testing or assessment; for example, it is possible to declare the request for accommodations and the accommodation description (`<requestForAccomodations>`, `<accomodationDescription>`).

An ACCLIP profile would be presented to an e-learning application by a learner, perhaps using a smart card, a memory stick or perhaps automatically retrieved from a database. The system in turn would serve up the appropriately customized content adapted specifically for that person.

The IMS Global Learning Consortium specifies also standards devoted to provide content metadata, to define content alternatives and to drive authors in producing contents, in order to improve didactical materials accessibility:

- the IMS AccessForAll Meta-data (ACCMD) specification [63] describes adaptable learning content by specifying, for example, what form the content will be presented in. The ACCMD specification might be implemented in an LMS. The LMS would receive an ACCLIP profile from a user, then based on that profile, use an ACCMD application in the LMS to retrieve content appropriate for that person’s needs. ACCMD is the mirror of ACCLIP, providing an interpreter for ACCLIP profiles and choosing the appropriate content based on that interpretation.
- The IMS Guidelines for Developing Accessible Learning Applications specification [64] defines a set of guidelines, which

provide a framework for the distributed learning community. This framework will set the stage for what solutions currently exist, what the opportunities and possibilities are for implementing them, and the areas where more development and innovation are still needed in educational technologies to ensure education that is truly accessible to anyone, anytime, anywhere.

The AccessForAll Meta-data specification is intended to make possible for systems to identify resources that match a user's stated preferences or needs. ACCMD describes the adaptability of learning content by specifying alternative formats for each content element, such as text alternatives for images, descriptive audio for video content, transcripts or captioning for audio tracks, visual alternatives for text, colour alternatives to increase contract, reduced alternatives for small screens and a variety of other potential alternative formats. By entering an XML profile string when entering an ACCLIP aware Web site or application, a blind user viewing a video, for example, will automatically receive that video with descriptive audio. A deaf user will receive the same video but with captioning instead. A user on a cell phone may use an ACCLIP profile to display the video at a lower resolution. A typical user will receive just the video without any transformation. Similarly, an ACCLIP profile can be used to configure a computer work station with the appropriate assistive technologies, or reconfigure a web application perhaps simplifying it for a person with a learning disability or a cognitive impairment, all simply by inserting a USB memory stick, or swiping a smart card with an ACCLIP profile on it [65].

2.4 Content Negotiation and Device Identification

Before adapting content to different devices, we need to know something about the device and to negotiate between the adaptation system and the device [154]. There are currently two main standardized methods of performing content negotiation, which are described in the following Subsections: the HTTP request header field and the Resource Description Framework (RDF) Profiles (Composite Capabilities/Preferences Profile and User Agent Profile).

2.4.1 HTTP Request Header Message

The HTTP request header field is a unique identifier sent from a client device to a server when asking for a service. It can be used for statistical measurements, and can also be used to provide device-specific content for different Web browsers. In order to increase the use of the HTTP request header, its format can be extended. But there has been no standard framework for defining extensions yet, the HTTP Extension Framework (HTTPext) has been moved to Experimental RFC2774 [98].

The information in the HTTP request header is often added differently by different browsers, and even wrongly expressed. For example, the Microsoft IE Browser can be described as Mozilla in the HTTP user agent string; Opera browser can appear as Microsoft IE, Mozilla or Opera, because the user agent identification can be configured in its settings menu. Therefore, if a browser is unknown or identifies itself incorrectly, content adaptation depending on the HTTP request header may generate unexpected results.

2.4.2 Resource Description Framework

The World Wide Web was designed for human use and all the data on the Internet can only be read but not understood by machines. There is so much information already available that managing and updating it becomes unrealistic. The W3C proposed the Resource Description Framework (RDF) [156] to improve the maintenance and flexibility of Web resources. RDF uses metadata to describe the data in the Web and makes it much easier to automatically manage and process the Web data and resources [157]. RDF provides interoperability between applications interchanging machine understandable information on the Web, and also between individual servers and clients.

The main aim of RDF is to define a mechanism to describe resources without making any assumption about the application domain and its semantics, in order to make the work easier for autonomous agents. RDF is based on XML in a standardized and interoperable manner and it is also possible for RDF to use other syntax.

2.4.3 RDF Profile

There are different RDF profiles, such as CC/PP [150] and User Agent Profile (UAProf) [102]. These are two related standards, recommended by the W3C and the Open Mobile Alliance (OMA). As the diversity of devices increases, the device capability and preference for content negotiation and adaptation must be known. The goal of these profiles is to allow client devices to tell servers their capabilities. The CC/PP and UAProf data formats are based on RDF models and describe device capabilities with two-level hierarchies consisting of components and attributes. When we parse these profiles, RDF is an abstraction level over XML, so it must validate both XML and RDF [156].

CC/PP and UAProf are useful for device independence, content negotiation and adaptation, as they allow different devices to specify their capabilities in a uniform way.

2.4.3.1 CC/PP Profile

The Composite Capabilities/Preference Profile (CC/PP) provides a standard way for devices to transmit their profiles when requesting Web content. Servers and proxies can then provide adapted content appropriate to a particular device [150].

A CC/PP vocabulary is defined by using RDF [156] [157] and specifies components and attributes of these components used by the application to describe a certain context. The three main components specify the hardware platform, software platform and browser user agent. In particular:

- *Hardware Platform*: this component defines the device (mobile device, personal computer, palmtop, tablet PC, etc…) in terms of hardware capabilities, such as `displaywidth` and `displayheight` (that specify display width and display height resolution), `audio` (that specifies audio board presence), `imagecapable` (that specifies images support), `brailledisplay` (that specifies Braille display presence), `keyboard` (that specifies keyboard type).

- *Software Platform*: this component specifies the device software capabilities, such as `name` (which specifies operating system name), `version` (which specifies operating system version), `tool` (which specifies present assistive tools), `audio` (which specifies supported audio types), `video` (specifies supported video types), `SMILplayer` (which specifies present SMIL players).
- *Browser User Agent*: this component describes the browser user agent capabilities, such as `name` (specifies user agent name), `version` (specifies user agent version), `javascriptversion` (specifies javascript versions supported), `CSS` (specifies CSS versions supported), `htmlsupported` (specifies HTML versions supported), `mimesupported` (specifies mime types supported), `language` (specifies languages supported).

The protocol for transmitting CC/PP profiles is based on an experimental HTTP extension framework. Many existing servers do not support this protocol, so developers have to adjust it to make it compatible in some way.

There are two key problems related to device independence which are beyond CC/PP working group scope:

1. CC/PP profile does not provide a standard vocabulary for Web clients to communicate their capabilities to servers.
2. It does not describe the type of adaptation methods that servers should perform on behalf of devices based on their capabilities.

Such problems needs to be solved in order for the protocol to be used in practice.

2.4.3.2 User Agent Profile

UAProf is defined as a standard between Wireless Application Protocol (WAP) devices and servers. The profile can be used for better content adaptation for different types of WAP devices [102]. UAProf profile also describes the next generation of WAP phones. The advantage of UAProf is that it defines different categories of mobile device capability [99]:

- *HardwarePlatform Component*: as the related CC/PP component, this category provides information about the hardware capabilities of the mobile device, such as color capability (by using `ColorCapable` and `BitsPerPixel` attributes), model name of mobile device (by using `Model` and `Vendor` attributes), text input capability (by using `TextInputCapable` attribute), screen size (by using `ScreenSize` and `ScreenSizeChar` attributes) and sound capability (by using `SoundOutputCapable` attribute).
- *SoftwarePlatform Component*: as the related CC/PP component, this category provides information about the software characteristics of the mobile device, such as audio and video

encoders supported (by using `AudioInputEncoder` and `VideoInputEncoder` attributes), character sets accepted (by using `CcppAccept-Charset` attribute), Java capability (by using `JavaEnabled`, `JavaPlatform` and `JVMVersion` attributes), acceptable content types / MIME types (by using `CcppAccept` attribute) and operating system name and version (by using `OSName`, `OSVendor` and `OSVersion` attributes).

- *BrowserUA Component*: as the related CC/PP component, this category specifies information about the browser of the mobile device. For example, mobile browser name and version (by using `BrowserName` and `BrowserVersion` attributes), HTML version supported (by using `HtmlVersion` attribute), XHTML version supported (by using `XhtmlVersion` and `XhtmlModules` attributes) and JavaScript capability (by using `JavaScriptEnabled` and `JavaScriptVersion` attributes).
- *NetworkCharacteristics Component*: this category specifies information about the capabilities of the mobile device for network connection. For example, bearers supported (CSD, GPRS, SMS, EDGE, etcetera, by using `SupportedBearers` attribute) and encryption methods supported (WTLS, SSL, TLS, etcetera, by using `SecuritySupport` attribute).
- *WapCharacteristics Component*: this category provides information about the WAP features supported by the mobile device. For example, DRM (Digital Rights Management)

capability (by using `DrmClass` and `DrmConstraints` attributes), maximum WML deck size (by using `WmlDeckSize` attribute), WAP version supported (by using `WapVersion` attribute) and WMLScript libraries supported (by using `WmlScriptVersion` and `WmlScriptLibraries` attributes).

- *PushCharacteristics Component*: this category specifies information about the WAP Push capabilities of the mobile device. For example, character encodings supported (by using `PushAcceptEncoding` attribute), character sets supported (by using `PushAcceptCharset` attribute), content types / MIME types supported (by using `PushAccept` attribute) and maximum WAP Push message size (by using `PushMsgSize` attribute).
- *MmSCharacteristics Component*: this category provides information about the MMS (Multimedia Messaging Service) capabilities of the mobile device. For example, maximum MMS message size supported (by using `MmsMaxMessageSize` attribute), maximum image resolution supported (by using `MmsMaxImageResolution` attribute) and character sets supported (by using `MmsCcppAcceptCharSet` attribute).

The weakness of this standard is that it does not resolve how servers and proxies should use the UAProf profile, as well as CC/PP profile.

Chapter 3

3. A novel proposal for adapting rich Learning Objects

This Chapter summarizes the idea that has driven our work in designing and developing a system which faces the problem statement described in Section 1.1, by delivering personalized video-lectures, automatically computed to meet user access capabilities.

We concentrate our efforts on video-lectures, considered as multimedia contents which contemporaneously reveal the complexity and the potentiality of delivering rich media to learners who work in restricted conditions. A video-lecture is basically composed by two continuous flows (audio and video) synchronized with a slides sequence and all the textual information (captions and slide descriptions) needed to ensure complete accessibility. In this context we used SMIL [159] synchronization format to represent this synchronous resource, described by metadata and packaged in a SCORM [5] Learning Object (LO), which represents a rich LO.

In order to provide the user with a video-lecture that fit his/her needs, it is necessary to transform the rich LO so that it can correctly work on the device in use and respect accessibility issues. In order to plan an appropriate adaptation activity, we needed a system to adequately define the contexts constrains, which are due to two main sets of characteristics:

i) learner's needs, in terms of possible disabilities, needed and/or preferred interface interaction options (which involves both input and output aspects), and
ii) device capabilities, in terms of hardware characteristics, installed software, user agent equipment and supported connectivity.

As we pointed out in the previous Chapter, different existing profiling standards are available, and, hence what we have to do is identifying the most adequate and complete ones, which have to take into account the above two mentioned sets of characteristics, and appropriately combining them. By considering standardization as a main guideline in the design of mobile and accessible e-learning, we identify two currently available proposals (CC/PP [150] and IMS ACCLIP [65]) to be combined in our approach (see Chapter 4).

On the basis of users' and devices profiles, rich LOs could need one or more transformations. Contextual constrains (which are dictated by learners needs and device characteristics, as already mentioned) may impose single media adaptation in terms of size, display dimensions, format, presentation, compression, transformation into different kind of media, etc. Certainly, the most complex situation emerges when rich media are involved, since limited conditions require a set of hard transformations, which undermine media synchronicity.

This activity is performed, as shown in Figure 3.1 below, by using the limited adaptation capabilities of currently available device and formats and by supporting a complex service-oriented transcoding activity. In literature (as mentioned in the previous Chapter), several content transcoding and adapting approaches and mechanisms exists. Such scenario implies that we have to design a system which considers the most wide and complete set of rich media transformations (which have to involve single media and entire presentations), allowing the possible addition of new kind of operations and maintaining media synchronicity or degrading it in the most appropriate way. Our proposal in terms of rich LOs transcoding is illustrated in Chapter 5.

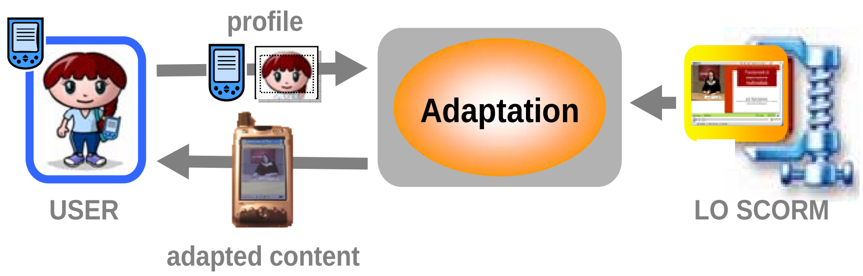

Figure 3.1 LO adaptation scheme

According to main literature, it is worth noting that the involved techniques are rather well-known and already existing, but their combination is original and the system as a whole actually represents the novelty of our work.

Chapter 6 will show how the above mentioned techniques are put together, characterizing our system.

Chapter 4

4. On Profiling Learners and Devices

In this Chapter we are going to discuss our proposal in terms of profiling learner's context, which is described in Section 4.1. In Section 4.2 we will present four scenarios illustrating different use cases according to which, different learners' and devices profiles need to be considered.

4.1 Composing Learners' Profiles

In this Section, we sketch how the learner profile is utilized in order to produce accessible LOs, which can be fully enjoyed by learners. The basic idea is that such a profile must describe both the device in use and all the learner's characteristics, which are needed to identify accessibility issues.

As mentioned in the previous Chapter, some projects have been done in the direction of managing Learning Objects (LOs), based on the idea of adapting contents and their presentation in a suitable way. Yet, none of them took into account device capabilities. As a consequence, LO adaptation can not be effectively completed so as to meet mobile users' requirements.

A profiling mechanism is required in order to describe both users and devices, so that users preferences and needs are defined. Several standards and solutions have been proposed (such as IMS Global Learning Consortium LIP [66] and ACCLIP [65], W3C CC/PP [150] and OMA UAProf [102]), without generating an exhaustive and fully supported solution. In fact, while CC/PP offers an open profiling mechanism, it defines a "common vocabulary" that fully describes only the device. On the other side, ACCLIP outlines the user in terms of accessibility needs, without considering device characteristics. To completely profile learners and devices, we need to consider both the user needs and the device capabilities. Hence, we coupled these two standards. It is worth noting that profiling procedures based on learners' didactical preferences are out of this thesis scope.

In order to profile learners, we used the IMS Accessibility for Learner Information Package (ACCLIP) Specification [65]. ACCLIP is an XML-based standard and enables the description of user preferences (visual, aural, device), which can be used for tailoring learning content (e.g. preferred/required input/output devices or preferred content alternatives). In other words, it provides a means to describe how learners interact with an e-learning environment, by focusing on accessibility requirements. The ACCLIP Specification defines the required elements to represent accessibility preferences, which may be grouped into four sections, as mentioned in Chapter 2: display information, control information, content information and accommodations which a learner is eligible for. However, no elements to characterize client devices are provided by such a standard. Thus, we refer to the W3C's CC/PP (*Composite Capabilities/Profile Preferences*) standard to profile devices [150]. A CC/PP profile is composed of a set of CC/PP attribute names and related values, assessed to describe device capabilities and characteristics. CC/PP is based on RDF (*Resource Description Framework*) [156], which is designed by the W3C as a metadata and machine understandable properties description language.

A comparative analysis of ACCLIP and CC/PP shows that the whole set of characteristics they cover is the same that we need to exhaustively profile any learner's context. The joining of such two sets of descriptions represents a complete profile of the dyad (learner, device).

$$\boxed{\text{ACCLIP Profile} \cup \text{CC/PP Profile} = \text{Complete Profile}}$$

It is worth noting that the intersection of ACCLIP and CC/PP is not an empty set.

$$\boxed{\text{ACCLIP Profile} \cap \text{CC/PP Profile} \neq \emptyset}$$

In particular, the overlapping of the two sets of characteristics includes all the assistive technologies that are declared in CC/PP as hardware and software components, while in ACCLIP it defines accessibility tools used by learners.

By comparing such two descriptions we can observe that:

i) Assistive technologies declared in CC/PP represent hardware and software in use on the device. An assistive technology can be installed on a device in use by people without any disabilities (e.g. people who test accessibility application, people who share a device with someone else with a disability).

ii) Assistive technologies which are specified in ACCLIP, declare hardware and software needed by the learner (i.e. currently in use).

In order to face such an overlapping, our profiling approach considers assistive technologies as they are defined in ACCLIP (ii), by discarding analogous information provided by CC/PP (i).

Now, in the following Subsections, we are going to show four use cases (*A*, *B*, *C* and *D*), which illustrate four different learners using different hardware and software platforms. We will describe the related IMS ACCLIP and CC/PP descriptions. We will maintain the original XML-based format for the ACCLIP and RDF-based format for CC/PP, in order to enhance readability of the provided profiling code and to enforce the compliance to existing standards.

4.2 Some use cases

In order to give emphasis to all involved aspects, in this Section we are going to provide four scenarios, by illustrating different use cases according to which different learners and devices profiles need to be considered.

4.2.1 Scenario A: a Fully Equipped, User with no disabilities

As a first scenario, let us consider a user (say *A*) which gains access to the lecture from his home. *A* user utilizes a fully equipped PC with any support for high quality audio/video and SMIL players. Within his profile, the user specifies a preference for having video encoded with a Real Video code, while MPEGs are exploited in the LO.

A learner's ACCLIP profile is shown in Figure 4.1. In such a scenario no transcoding operations are required to deliver a LO, which can be due to accessibility user's needs. Thus, in the related ACCLIP profile, the element `<AccessForAll>` is kept empty, i.e., no accessibility issues must be taken into account.

The figures 4.2, 4.3 and 4.4 show some fragments of the three main CC/PP components codes, which define a fully equipped platform.

```
...
<accessForAll schemaVersion="1.0.29"
       xmlns="http://www.imsglobal.org/xsd/acclip"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.imsglobal.org/xsd/acclip
      AccessForAllv1p0d29.xsd">
      <context identifier="userA" xml:lang="it"/>
</accessForAll>
...
```

Figure 4.1 IMS ACCLIP in Scenario A

```
[sfa:AProfile]
 |
 +-ccpp:component-> [sfa:TerminalHardware]
 |                  |
 |                  +-rdf:type-----------> [sfa:HardwarePlatform]
 |                  +-ex:displayWidth----> "1024"
 |                  +-ex:displayHeight---> "768"
 |                  +-sfa:audio----------> “yes”
 |                  +-sfa:imagecapable---> “yes”
 |                  +-sfa:brailledisplay-> “no”
 |                  +-sfa:keyboard-------> “yes”
...
```

Figure 4.2 CC/PP Hardware Platform Component Profile in Scenario A

```
...
+-ccpp:component-> [sfa:TerminalSoftware]
 |                   |
 |                   +-rdf:type--------> [sfa:SoftwarePlatform]
 |                   +-ccpp:defaults---> [sfa:SWDefaults]
 |                   +-sfa:name--------> “Windows XP Professional”
...
 |                   +--sfa:audio-----------> [ ]
 |                   |                          |
 |                   |  ----------------------
 |                   |  |
 |                   |  +--rdf:type---> [rdf:Bag]
 |                   |  +--rdf:_1-----> “mp3”
 |                   |  +--rdf:_2-----> “ra”
...
 |                   |
 |                   +--sfa:video-----------> [ ]
 |                   |                          |
 |                   |  ----------------------
 |                   |  |
 |                   |  +--rdf:type---> [rdf:Bag]
 |                   |  +--rdf:_1-----> “rm”
...
 |                   |
 |                   +--sfa:SMILplayer------> [ ]
 |                   |                          |
 |                   |  ----------------------
 |                   |  |
 |                   |  +--rdf:type---> [rdf:Bag]
 |                   |  +--rdf:_1-----> “RealOne”
 |                   |
...
```

Figure 4.3 CC/PP Software Platform Component Profile in Scenario A

```
...
 +--ccpp:component-->[sfa:TerminalBrowser]
                     |
                     +--rdf:type------------> [sfa:BrowserUA]
                     +--ccpp:defaults-------> [sfa:UADefaults]
                     +--sfa:name------------> "Internet Explorer"
                     +--sfa:version---------> "6.0"
                     +--sfa:javascriptversion-> [ ]
                     |                            |
                     |  ------------------------
                     |  |
                     |  +--rdf:type---> [rdf:Bag]
                     |  +--rdf:_1-----> "1.1"
 ...
                     |
                     +--sfa:CSS-------------> [ ]
                     |                          |
                     |  ----------------------
                     |  |
                     |  +--rdf:type---> [rdf:Bag]
                     |  +--rdf:_1-----> "2.0"
 ...
                     |
                     +--sfa:htmlsupported---> [ ]
                     |                          |
                     |  ----------------------
                     |  |
                     |  +--rdf:type---> [rdf:Bag]
                     |  +--rdf:_1-----> "3.2"
                     |  +--rdf:_2-----> "4.01"
                     |
                     +--sfa:mimesupported---> [ ]
                     |                          |
                     |  ----------------------
                     |  |
                     |  +--rdf:type---> [rdf:Bag]
                     |  +--rdf:_1-----> "text/html"
                     |  +--rdf:_2-----> "text/plain"
                     |  +--rdf:_3-----> "text/css"
 ...
                     |
                     +--sfa:language--------> [ ]
                                                |
                        ----------------------
                        |
                        +--rdf:type---> [rdf:Seq]
                        +--rdf:_1-----> "it"
```

Figure 4.4 CC/PP Browser User Agent Component Profile in Scenario A

4.2.2 Scenario B: a Fully Equipped, Deaf User

Let consider, instead, the case of a deaf user (say *B*) which gains access to the lecture by means of a fully equipped PC. A SMIL player is installed on her system.

Figure 4.5 depicts the *B* user profile. In this ACCLIP profile the user defines a set of preferences about visual alters instead of generic audio ones (see element `<visualAlert>` inside `<display>` element).

```
...
<accessForAll schemaVersion="1.0.29"
xmlns="http://www.imsglobal.org/xsd/acclip"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="
http://www.imsglobal.org/xsd/AccessForAllv1p0.xsd">
      <context identifier="userB" xml:lang="it">
        <display>
           <visualAlert>
              <visualAlertGeneric>
                 <systemSounds value="captionBar"/>
                    <captions value="true"/>
              </visualAlertGeneric>
           </visualAlert>
        </display>
</accessForAll>
...
```

Figure 4.5 - IMS ACCLIP in Scenario B

The related three main CC/PP components chunks of code are shown in the previous figures 4.2, 4.3 and 4.4, which define a fully equipped platform.

4.2.3 Scenario C: a Fully Equipped, Blind User

Let *C* be a blind user who gains access to the Internet with a PC equipped with a screen reader and a Braille display (i.e., the assistive technologies that enable blind people to use a computer). A SMIL player is installed on the system.

A simplified version of user *C* profile is depicted in Figure 4.7. Here, the user declares a set of preferences about its used screen reader (see element `<screenReader>` inside `<display>` element), as well as its Braille display characteristics (see `<braille>` element, partially omitted). All these elements are included inside the accessibility LIP element (`<AccessForAll>`) which drives the system transcoding process. Based on this profile, the system produces an alternative version of each graphical and visual content.

```
[sfa:CProfile]
 |
 +-ccpp:component-> [sfa:TerminalHardware]
 |                  |
 |                  +-rdf:type-----------> [sfa:HardwarePlatform]
 |                  +-ex:displayWidth----> "1024"
 |                  +-ex:displayHeight---> "768"
 |                  +-sfa:audio----------> “yes”
 |                  +-sfa:imagecapable---> “yes”
 |                  +-sfa:brailledisplay-> “yes”
 |                  +-sfa:keyboard-------> “yes”
...
```

Figure 4.6 CC/PP Hardware Platform Component Profile in Scenario C

Figures 4.6 and 4.8 show the Hardware Platform and the Software Platform CC/PP components code. In this scenario hardware and software assistive technologies (Braille display and Jaws, a screen reader) are provided in order to allow a blind user to utilize such platform.

```
...
<accessForAll schemaVersion="1.0.29"
xmlns="http://www.imsglobal.org/xsd/acclip"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=
        "http://www.imsglobal.org/xsd/AccessForAllv1p0.xsd">
      <context identifier="userC" xml:lang="it">
         <display>
           <screenReader>
              <screenReaderGeneric>
                <link value="speakLink"/>
                <link value="differentVoice"/>
                <speechRate value="500"/>
                <pitch value="0.8"/>
                <volume value="0.5"/>
              </screenReaderGeneric>
           </screenReader>
           <braille>…</braille>
          </display>
          <control>
             <keyboardEnhanced>…</keyboardEnhanced>
             <mouseEmulation>…</mouseEmulation>
             <voiceRecognition>…</voiceRecognition>
          </control>
      </context>
</accessForAll>
...
```

Figure 4.7 - IMS ACCLIP in scenario C

```
...
+-ccpp:component-> [sfa:TerminalSoftware]
 |                   |
 |                   +-rdf:type--------> [sfa:SoftwarePlatform]
 |                   +-ccpp:defaults---> [sfa:SWDefaults]
 |                   +-sfa:name--------> “Windows XP Professional”
...
 |                   +-sfa:tool-------->[ ]
 |                   |                   |
 |                   |  ----------------
 |                   |  |
 |                   |  +--rdf:type---->[rdf:Seq]
 |                   |  +--rdf:_1------>“jaws5.0”
...
 |                   +-sfa:audio------>[ ]
 |                   |                  |
 |                   |  ---------------
 |                   |  |
 |                   |  +--rdf:type-->[rdf:Bag]
 |                   |  +--rdf:_1---->“wav”
 |                   |  +--rdf:_2---->“mp3”
 |                   |  +--rdf:_3---->“wma”
 |                   |  +--rdf:_4---->“mid”
 |                   |  +--rdf:_5---->“ra”
 |                   |
 |                   +-sfa:video------>[ ]
 |                   |                  |
 |                   |  ---------------
 |                   |  |
 |                   |  +--rdf:type--->[rdf:Bag]
 |                   |  +--rdf:_1----->“avi”
 |                   |  +--rdf:_2----->“mpeg”
 |                   |
 |                   +--sfa:SMILplayer->[ ]
 |                   |                   |
 |                   |  ----------------
 |                   |  |
 |                   |  +--rdf:type->[rdf:Bag]
 |                   |  +--rdf:_1--->“RealOne”
 |                   |  +--rdf:_2--->“QuickTime”
 |                   |
...
```

Figure 4.8 CC/PP Software Platform Component Profile in Scenario C

4.2.4 Scenario D: a Mobile User with no disabilities

Finally, say a user *D* gains access to the lecture by means of a smart phone. Her handheld device has a small screen, reduced computational capabilities and it does not support the SMIL technology.

Figure 4.9 depicts the *D* user profile. In this ACCLIP profile the user defines a set of preferences about different input control systems, due to the use of a PDA, in such a way to allow mouse emulation (see element `<mouseEmulation>` inside `<control>` element).

We can observe that IMS ACCLIP defines a set of means to describe just the device control, but no information about supported formats and display dimensions are provided. Thus, we need to involve CC/PP in order to express such device capabilities.

The figures 4.10, 4.11 and 4.12 show some code fragments of the three main CC/PP components, which define mobile device platform characteristics.

```
...
<accessForAll schemaVersion="1.0.29"
xmlns="http://www.imsglobal.org/xsd/acclip"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="
http://www.imsglobal.org/xsd/AccessForAllv1p0.xsd">
      <context identifier="userD" xml:lang="it">
        <control>
           <mouseEmulation>
              <mouseEmulationGeneric>
              <speed value="0.5"/>
                <acceleration value="0.5"/>
                <device value="keypad"/>
              </mouseEmulationGeneric>
           </mouseEmulation>
        </control>
      </context>
</accessForAll>
...
```

Figure 4.9 - IMS ACCLIP (Scenario D)

```
[sfa:DProfile]
 |
 +-ccpp:component-> [sfa:TerminalHardware]
 |                  |
 |                  +-rdf:type-----------> [sfa:HardwarePlatform]
 |                  +--ex:displayWidth----> "240"
 |                  +--ex:displayHeight---> "320"
 |                  +--sfa:audio----------> “yes”
 |                  +--sfa:imagecapable---> “yes”
 |                  +--sfa:brailledisplay-> “no”
 |                  +--sfa:keyboard-------> “no”
 |
...
```

Figure 4.10 D’s CC/PP Hardware Platform Component Profile

```
...
 +-ccpp:component-> [sfa:TerminalSoftware]
 |                  |
 |                  +--rdf:type----------> [sfa:SoftwarePlatform]
 |                  +--ccpp:defaults-----> [sfa:SWDefaults]
 |                  +--sfa:name----------> “Pocket PC”
...
 |                  +--sfa:audio-----------> [ ]
 |                  |                         |
 |                  |  ----------------------
 |                  |  |
 |                  |  +--rdf:type---> [rdf:Bag]
 |                  |  +--rdf:_1-----> “wav”
 |                  |  +--rdf:_2-----> “mp3”
 |                  |  +--rdf:_3-----> “mid”
 |                  |
...
```

Figure 4.11 D’s CC/PP Software Platform Component Profile

```
...
 |
 +-ccpp:component->[sfa:TerminalBrowser]
                  |
                  +-rdf:type-------> [sfa:BrowserUA]
                  +-ccpp:defaults--> [sfa:UADefaults]
                  +-sfa:name-------> “Pocket Internet Explorer”
                  +--sfa:version---------> “4.1”
                  +--sfa:javascriptversion-> [ ]
                  |                            |
                  |  -------------------------
                  |  |
                  |  +--rdf:type---> [rdf:Bag]
                  |  +--rdf:_1-----> “1.5”
                  |
                  +--sfa:CSS-------------> [ ]
                  |                          |
                  |  -----------------------
                  |  |
                  |  +--rdf:type---> [rdf:Bag]
                  |  +--rdf:_1-----> “1.0”
                  |
                  +--sfa:htmlsupported---> [ ]
                  |                          |
                  |  -----------------------
                  |  |
                  |  +--rdf:type---> [rdf:Bag]
                  |  +--rdf:_1-----> “3.2”
                  |  +--rdf:_2-----> “4.01”
                  |
                  +--sfa:mimesupported---> [ ]
                  |                          |
                  |  -----------------------
                  |  |
                  |  +--rdf:type---> [rdf:Bag]
                  |  +--rdf:_1-----> “text/html”
                  |  +--rdf:_2-----> “text/plain”
                  |  +--rdf:_2-----> “audio/mpeg”
                  |  +--rdf:_2-----> “text/css”
                 ...
                  |
                  +--sfa:language--------> [ ]
                  |                          |
                  |  -----------------------
                  |  |
                  |  +--rdf:type---> [rdf:Seq]
                  |  +--rdf:_1-----> “it”
```

Figure 4.12 D’s CC/PP Browser User Agent Component Profile

Chapter 5

5. On Transcoding Rich Media LOs

This Chapter details a suitable solution for dynamic adaptation and transcoding of widely different SCORM-compliant LOs before their delivery to users. First, in Section 5.1 we discuss the LOs transcoding main issues. Second, Section 5.2 illustrates the strategy we designed for the LOs adaptation. Third, in Section 5.3 we present four scenarios in which LOs are adapted on the basis of learners and devices profiles (as defined in the previous Chapter).

5.1 Transcoding LOs

A transcoding process consists of a set of conversion steps, each of them involving one of the media which is included in the whole complex synchronized multimedia presentation. Practically speaking, as pointed out in the previous Chapter, two main factors are considered during such a transcoding activity:

i) the computational capabilities of the user device, together with the software installed on the client device, and the networking capabilities of the mobile client, i.e., those networking technologies which are supported by the mobile terminal and/or are actually available at the moment of the rich media delivery;

ii) the user characteristics.

As a matter of facts, due to the numerousness of possible user scenarios, it results that finding the most appropriate transcoding strategy is not a simple task [40]. Indeed, such a scheduled conversion process must respect all the constraints imposed by the system, network and the learner capabilities.

Customization of LOs, based on user needs might be performed in different ways. Basically, the main strategies can be summarized in three categories:

1. Adding metadata: ad hoc metadata can be associated to content to specify its characteristics. They can be matched with learner profiles once a given LO has been requested. Metadata are typically used to engage a selection on the content database; this way, only contents which are consistent with the learner profile are considered to be delivered to the user.
2. Use of customization primitives. The idea is based on maintaining a set of alternatives inside the content, which can be used once the learner specifies his preferences. To this aim, the content is formatted so that evaluation methods and selection primitives are exploited to permit to select the most suitable alternatives for the specific learner.

3. Real-time adaptation of contents. This strategy is mainly utilized in mobile rich-media applications to ensure that content can be delivered and managed despite widely varying characteristics of mobile devices.

As to the use of metadata (1), this approach has been extensively employed by the IMS Global Learning Consortium [62], which has proposed the IMS AccessForAll MetaData (ACCMD) [63]. In that proposal, it is suggested to describe accessible learning contents by specifying, for example, which kind of content is being presented and weather there is an equivalent or alternative form for that content. Besides, ACCMD provides support to functional interoperability, i.e., any resource can be substituted or coupled with an alternative. To this aim, each media resource is associated with a description of a set of additional resources, which are somehow equivalent to the primary one [63]. ACCMD is typically used together with the IMS Accessibility for Learner Information Package (LIP) [66] which describes preferences that should be stored in a user profile (e.g., preferred/required input devices or preferred content alternatives). However, the problem with ACCMD is that it can be only partially applied to rich media contents. Indeed, ACCMD requires that contents are managed as unique, atomic components, while rich media are, by definition, complex ones, i.e., a composition of synchronized media resources. Summing up, based on ACCMD, content is considered either as accessible or as not accessible as a whole. Thus, it is not possible to specify alternative versions of single media composing the rich multimedia. This obviously represents an important limitation.

In rich media e-learning content, primary resources and their related alternatives are composed and synchronized by making use of time-based aggregation methods, such as parallelization and sequentialization primitives. The use of alternative versions of the content must be dynamically evaluated. In this sense, the strategy of including customization primitives in the content (2) seems to fit this goal. Some encoding formats, such as SMIL [159], for example, include accessibility issues inside their primitives in order to automatically offer support to synchronous alternatives. However, this approach shows limitations on other fronts. In fact, only a static set of limited, pre-defined preferences is provided to the user; this hampers the development of sophisticated customization mechanisms, able to select among alternative contents. Moreover, the presence of an alternative for a specific primary resource could cause cognitive overload to the user. Such limitations are overcome by resorting to approach able to adapt contents at real-time (3).

5.2 LOs Transcoding Strategy

In this Subsection, we are going to focus on the strategy we devised for the dynamic adaptation of SCORM-compliant LOs, before their delivery to users.

Based on our previous considerations, a need emerges for a system able to exploit transcoding strategies for the automatic production of SCORM-compliant LOs, encoded as synchronized multimedia presentations [41] [104] [113]. Such a system must be able to synchronously combine different discrete and continuous media, according to any user profile, which takes into account both user tastes, physical capabilities and devices technical characteristics. The produced LOs should include all the accessibility metadata, in such a way to ensure accessibility and portability of LOs. Metadata are retrieved from the LO; they are embedded in a traditional SCORM manifest and IMS ACCMD. Based on these inspected data, the system selects appropriate alternative versions of contents for the learner and schedules a transcoding strategy for computing appropriate rich media content [39] [90]. This adaptation process is performed to

i) modify characteristics of media so as to perfectly fit all the device and user requirements (e.g., resizing screen dimension),
ii) automatically compute a missing alternative (e.g., captioning a speech).

All the involved media content need to be synchronized, according to the temporal and spatial dimensions. As a synchronization among media contents composing a given LO, we assume that this is accomplished by making use of the SMIL technology [159], i.e., a well-known mark-up language for the specification of temporal and spatial synchronization relationships among media contents composing a multimedia presentation. Needless to say, complex situations arise when not only the transcoding strategy involves modifications on media content composing a LO, but also the synchronization specification of newly obtained media content needs to be modified. Thus, methods are needed to manage SMIL documents and to possibly transform these SMIL-based multimedia presentations into other formats (e.g., XHTML documents or video tracks).

Basically, a broking service has been designed. From a logical point of view, three different phases characterize the conversion of multimedia presentations representing unpacked SCORM LOs (see Figure 5.1):

i) a *Recoding Phase*,

ii) a *Media Transcoding Phase* and

iii) a *Postproduction Phase*.

These three phases are individually discussed in the following Subsections.

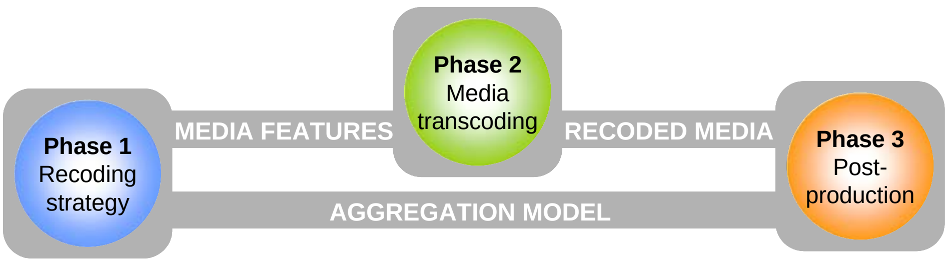

Figure 5.1 - Conversion Phase Sequence

5.2.1 Recoding Phase

The *Recoding Phase* states which typologies of media should be delivered to a given learner, according to her/his profile, and whether the synchronization specification needs to be modified. Suitable conversion rules for such an adaptation are described in Figure 5.2.

```
0 if (SMIL supported)
1    use SMIL presentation
2    pass in Phase 2 to transcode media
3 else //no SMIL techonologies exploited
4    if (video player supported)
5       create a single video (in Phases 2,3)
        merging original contents
6    else // playout of a sequence of contents
7       transcode media (in Phase 2)
        and create a linear sequence of
        contents to be played-out in sequence (Phase 3)
8    fi
9 fi
```

Figure 5.2 Recoding Phase

In particular, based on the learner and device profiles, a first check is performed to verify whether a SMIL player is supported on the client device. In the positive case, a SMIL specification is used for the final play-out of media contents composing the LO. Subsequent checks (along other phases) will be needed to check whether some media contents composing the synchronized multimedia presentation should be transcoded.

When SMIL technologies cannot be utilized (e.g., due to missing software at the client-side), a check is performed to verify whether video is supported. In the positive case, a transcoding activity is scheduled according to which all media contents associated to the LO are merged in a unique video file. In the negative case, instead, a different transcoding process is scheduled to convert the multimedia presentation into a discrete set of separate media contents which complies with the software installed on the client device. Such contents will be played-out in sequence.

5.2.2 Media Transcoding Phase

The *Media Transcoding Phase* is in charge of determining which media format must be used for each component of the LO. Depending on the client profile, each media may be left in its original format or, alternatively, converted into other formats, scaled, translated or discarded.

Given a specific kind of media, a match between the encoding format of that media and the capabilities of the client terminal is accomplished. If the actual encoding format is not supported by the client device, the system converts such a media content into another (supported) format. If no encoding format is supported for such a kind of media, that media content is converted into text. In certain cases conversion of media contents can be automatically accomplished without any additional information associated to the considered media.

Examples involve *conversion* between encoding formats of the same media type (e.g., audio files from WAV to MP3), but also *degradations* of contents to different media types, such as text-to-speech, speech-to-text or transformations from animations to images. In other cases, instead, additional information must be provided to substitute the media content with another one. An example, in this sense, is the translation from images to text, according to which images are simply substituted with their alternative text description. Examples of *scaling* are concerned with compression of media contents, reduction of their dimensions, quality or data rate. For instance, as to images and video, a check on terminal display size is carried out. Based on such a check result, videos and images may be resized, when necessary.

Other kinds of transformations may be accomplished in this phase. *Translation* can be employed on text in order to transform it from its original language into a different one, according to the user profile. Finally, *deletion* of media contents is accomplished for those contents which are useless for the user, when they cannot be played out by the client device and they cannot be transformed or substituted with some additional information being present inside the SCORM package.

Table 5.1 shows some important considerations linking different media to the characteristics of learners/devices profiles. Blank cells in the table correspond to absence of limitations.

Table 5.1: Media vs Client Profile

	Network	**Computation Capabilities / Software**	**Learner Profile**
Audio	Medium/high bandwidth required, or audio compression needed	Codec required For people who is hard of hearing, speech-to-text could be of help	Conversion needed for learners who is hard of hearing → use of captioning, additional textual description, speech-to-text
Video	High bandwidth required	Codec required	Not useful for blind people → use additional textual description + text-to- speech (synchronized with other audio contents)
Text		For blind people, text-to-speech software required	Text-to-speech needed for blind people
SMIL		SMIL player required; otherwise, transcode to a single media presentation	

As an example, typically, textual data use (as well as SMIL specifications, which are text-based) does not represent an issue from a networking point of view. Hence, text is considered as the most lightweight media and it can be easily transmitted, whatever the networking technology used by the learner.

Some considerations are in order. First, as expected the higher the available bandwidth is, the richer (in terms of kinds of media) the multimedia presentation is, which can be responsively delivered to the learner.

Second, learner preferences play an important role in the media adaptation process. Indeed, conversion of (not audio-based) media to audio contents is needed for blind users; then, obtained audio content needs to be presented according to a sequential play-out. Conversely, captions and additional textual information must be presented to people who is hard of hearing, who cannot enjoy audio contents.

Needless to say, contents which are not useful for a specific user during the LO visualization, must not be delivered to the client device, in order to save network bandwidth.

5.2.3 Postproduction Phase

The *Postproduction Phase* is in charge of recomposing and packaging all (transcoded) media contents to obtain a SCORM-compliant LO. Depending on the identified Recoding Phase, contents composing the LO can be structured as SMIL documents. Alternatively, when a single video track must be provided for final presentation, contents are merged together into a video. Finally, if a sequence of discrete contents must be played-out, a sequence of Web pages is automatically generated. Text is inserted within Web pages. In case of audio files, instead, links to these contents are created and placed into the documents; they will be played-out after an explicit request by the user.

5.3 Some use cases

In order to better emphasize all involved aspects, in this Subsection we are going to consider the four learners which have been described as use cases in the previous Chapter. Such scenarios illustrate situations according to which different transcoding strategies need to be employed. In particular, we are going to consider a LO which involves rich media synchronization. In the next Subsections, we will show how such a LO is properly adapted, based on learners' needs and their device characteristics.

The original LO is composed by the following media contents:

i) a video content showing the lecturer,
ii) an audio content embodying the lecturer's talk,
iii) a sequence of static images, representing the lecture slides.

Moreover, two other information flows are added and maintained synchronized with the others:

iv) a caption sequence used to store the lecturer's speech in a textual format, and finally,

v) an additional textual description of content, which are associated to each slide.

The two last additional content types are added to the LO to ensure portability and accessibility of the encoded contents [161]. Indeed, the captioning process results as an essential tool for students who are deaf or hard of hearing, as well as for foreign students; moreover, the use of captions is useful also whenever students gain access to the LO thanks to devices unsupplied with audio capabilities. The additional textual description of each slide, instead, can be exploited as alternative information to the media composing the lecture [112].

Figure 5.3 shows a screenshot of a developed lecture.

Figure 5.3 - The Synchronized Multimedia Lecture (a frame)

In Figure 5.4, a portion of the related original SMIL code is reported, which describes the information and data corresponding to a single slide of the lecture. With this lecture, which can be considered as a LO, we can now hypothesize different situations of use by students with very different user profiles.

Figures 5.5, 5.6, 5.7 and 5.8 show the results of the necessary transcoding processes for each use case which has been detailed above.

```
...
<par>
  <video src="video/video.mpg" region="region_video"/>
  <seq>
    ...
    <par>
      <img region="region_slide" src="img/2.jpg" dur="60s"
        alt="Perche studiare questa tecnologia, 1"
        longdesc="img/2.htm"/>
      <audio region="region_audio" src="audio/2.wav"/>
      <textstream src="caption/1.rt”
        "region="region_subtitle" stem-captions="on"/>
    </par>
    ...
  </seq>
</par>
...
```

Figure 5.4 - The Synchronized Video-Lecture SMIL Code Fragment (Corresponding to One Slide)

5.3.1 Scenario A: a Fully Equipped, User with no disabilities

In such a scenario, A user gains access to the lecture from his home, by using a fully equipped PC with support for high quality audio/video and SMIL players. In his profile, the user specifies a preference for having video encoded with a RealVideo Encode, while MPEGs are exploited in the LO. Based on individual user tastes, a conversion from MPEG to RealVideo is performed on video files. These files will substitute original ones in a new LO provided to the user. In this specific case, no synchronization relaxation is necessary for him. This way, *A* will enjoy an adapted, complete SMIL presentation with high quality media files (see Figure 5.5).

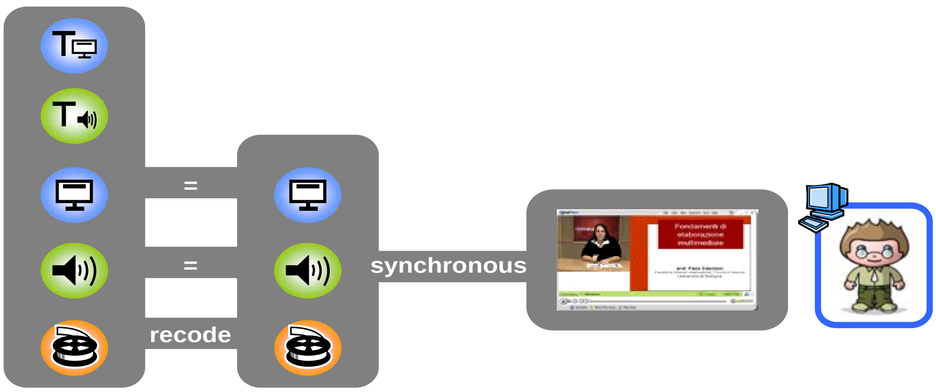

Figure 5.5 - Use Case A: Transcoding Processes and Final Result

5.3.2 Scenario B: a Fully Equipped, Deaf User

In such a scenario, B user is deaf and she gains access to the lecture by means of a fully equipped PC. A SMIL player is installed on her system. Since *B* is deaf, it results that transcoding of media contents are needed to meet user preferences. Hence audio is simply deleted while caption sequences are exploited (see Figure 5.6).

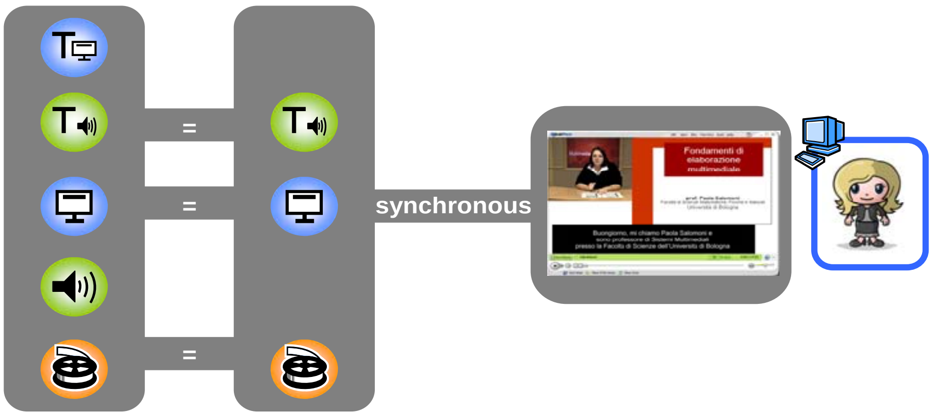

Figure 5.6 - Use Case B: Transcoding Processes and Final Result

5.3.3 Scenario C: a Fully Equipped, Blind User

In the third scenario, let us consider C user, who is blind and gains access to the Internet with a PC equipped with a screen reader and a Braille display (i.e., the assistive technologies that enable blind people to use a computer). A SMIL player is installed on the system. Due to the user blindness, only audio flows can be utilized along the presentation. Thus, all detailed visual information is omitted and substituted, whenever possible, with audio or alternative text. Use of text is admitted since such a kind of media can be converted to audio at the client side by means of the screen reader. However, actually current SMIL players and screen readers are not compatible. Hence, in this case a need emerges to face with the inability of screen readers to read text showed by the SMIL player. Furthermore, the system cannot simultaneously play-out an auditory content (i.e., the talk) while the screen reader is reading a text (i.e., the slide description). A new synchronization specification (not SMIL-based) must be set in order to obtain a linear sequence of contents. In particular, text and audio data are managed to be presented as a XHTML slide show (see Figure 5.7).

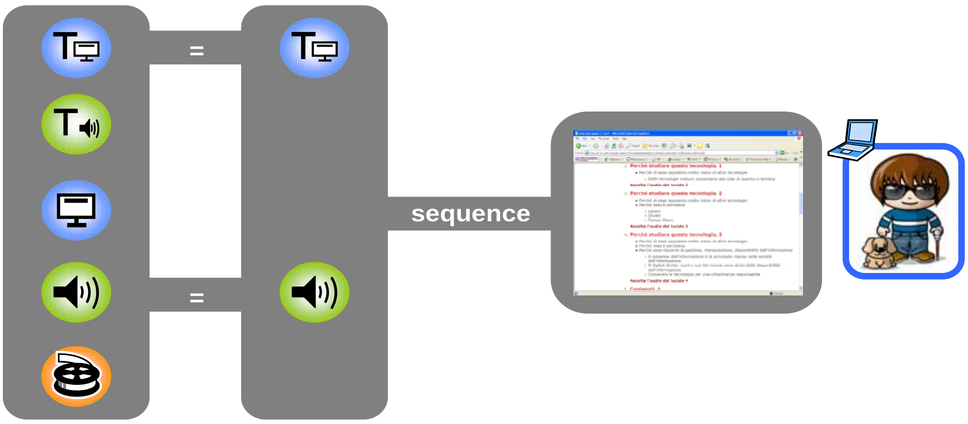

Figure 5.7 - Use Case C: Transcoding Processes and Final Result

Summing up, transcoding steps for the support of unsighted people are as follows:

i) video and images are omitted since they are useless for blind users;

ii) alternative textual descriptions substitute images on slides (while video is simply discarded);

iii) the audio talk is divided into portions which are merged with textual description of the slides.

All these mentioned use cases point out the need for a planning phase that decides how to adapt media contents based on the user profile.

5.3.4 Scenario D: a Mobile, User with no disabilities

Finally, consider user *D* who gains access to the lecture by means of a smart phone. As already described, her handheld device has a small screen, reduced computational capabilities and the platform does not support the SMIL technology. *D* is connected via an 802.11 WLAN network. Such a network guarantees an adequate bandwidth for a fluent transmission of the video clip reproducing the lecturer.

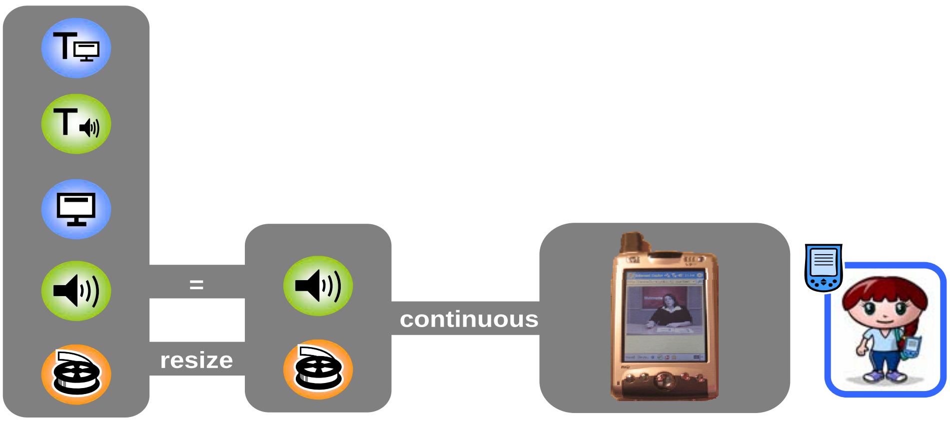

Figure 5.8 - Use Case D: Transcoding Processes and Final Result

In this context, transcoding of media contents are needed to meet device capabilities (absence of SMIL player on the PDA). In particular, a reduction of video and images sizes is necessary to meet PDA's display resolution. Finally, since no SMIL players are installed on D's PDA, the multimedia presentation needs to be transformed into a single video, which comprises all contents constituting the new LO for *D* (see Figure 5.8). Needless to say, since a single video is presented, which incorporates all the original information, the additional descriptions for images become useless.

Chapter 6

6. System Architecture

The aim of this Chapter is to point out the main system architecture issues, in order to transcode LOs meeting learners' needs and their device capabilities. First, in Section 6.1, we are going to illustrate how such a system works. Second, in Section 6.2, we present a system implementation.

6.1 How the system works

This Subsection is devoted to describe the whole system. Summarizing, in substance, such a system is endowed with methods to:

i) retrieve a LO, once it has been requested by a user,
ii) unpack such a LO,
iii) schedule and execute a transcoding strategy, on the basis of user and device profiles,
iv) re-pack all recoded media contents to obtain a new video-lecture and, finally,
v) deliver such a new content to the user.

We can summarize our system activities as follows:

- *Broking activity*: the system behaves as a broker that manages each specific user request to provide it with the best version of a LO.
- *Profiling activity*: user profiles are stored and managed so as to provide users with properly tailored LOs.
- *Transcoding activity*: the system orchestrates a set of specific transcoding Web Services to obtain the required form of the entire LO.
- *Unpackaging activity*: the system decomposes the original LO, which is encapsulated according to the SCORM packaging standard [5].

Each of these four activities is associated to a specific software component; these components are deployed in a software architecture, as discussed in the following Subsection.

A typical interaction between a client and such a system is similar to a Client/Server context. To obtain a tailored video-lecture, the client contacts the system by sending the learner profile together with a set of used device settings. Then the client will receive a video-lecture, which is optimized for the declared profiles and appropriately encapsulated. The system is able to recall previously connected users information. Thus, during their first connection, users have to specify device (hardware and software) capabilities and personal settings which will be recorded for future requests. Once the user and device profile have been received or have been retrieved by the database, the system compares its related user settings with the requested LO, and then defines a transcoding strategy.

Based on this transcoding strategy, the system computes a new version of the LO and sends it back to the client. Since the system manages SCORM-compliant LOs, it is able to un-package requested LOs.

6.2 An Implementation

As to the architectural design of our system, based also on the related work, it turns out that the best choice is probably structuring it as a service-oriented distributed architecture. A central component of such an architecture acts as a broking service (as it will be described in the following Subsection) in charge of scheduling the needed conversion steps to adequately transcode a multimedia presentation before its delivery and presentation to the user. Conversion rules are identified based on the user preferences and client device capabilities, i.e., based on the client profiles. Degradation of media (in the most graceful way) should be performed by issuing conversion requests to specific Web services, distributed over the network. This solution has the great advantage of distributing tasks, load and competences over the network, thus improving scalability of the system.

In particular, the system is made up of different software components (as depicted in Figure 6.1) which correspond to the functional activities mentioned in the previous Section:

- a *Media Broker* (MB), which manages users accesses to our system; schedules the transcoding activity;
- a *Profile Manager* (PM), which manages the Profile DB;

- a *Package Manager* (PaM), which un-packages SCORM-compliant LOs;
- a *Transcoding Unit* (TU), which executes the transcoding strategy.

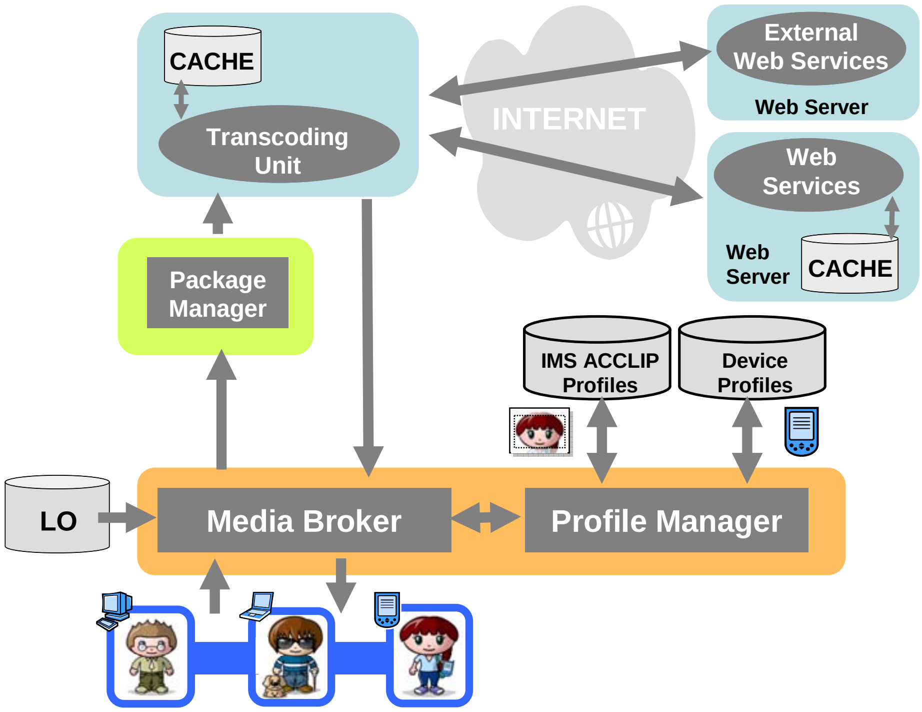

Figure 6.1 General System Architecture

The communication flow among system components is depicted in Figure 6.2. Basically, as soon as the user requests a LO, the client application authenticates to MB. MB interrogates PM, which retrieves the user's profile. Then, MB passes the request to PaM, which retrieves and un-packages the requested LO. Moreover, MB schedules a transcoding strategy, by matching the user and device profiles with the specific media resources composing the LO, based on steps described in the previous Chapter. The planned transcoding strategies with media that need conversion are forwarded to TU. TU executes the planned transcoding activities and, once these operations are completed, it forwards the adapted and recomposed resources back to the user, through MB.

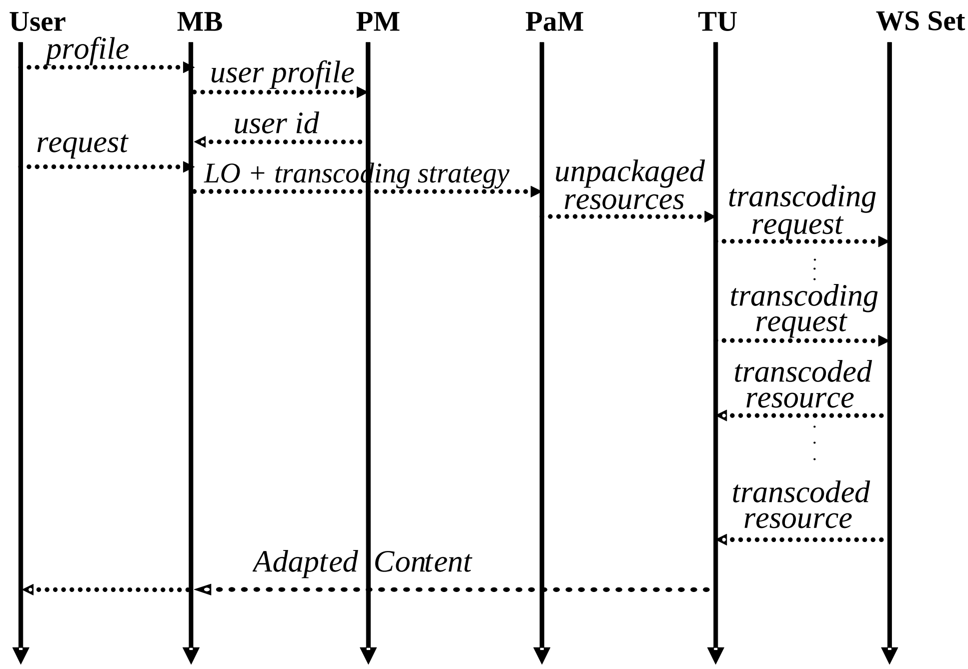

Figure 6.2 Communication Flow

Going into a more detailed discussion of the system implementation, it is worth noting that TU embeds some Web Services designed to locally accomplish specific transcoding processes. In particular, a single specific Web Service (named *Transcoding Unit Web Service*, TUWS) manages the SMIL document specification and (when needed) transcodes it into a new SMIL document. Then, a set of other local Web Services is used to perform different transcoding operations on single media resources. Summing up, each satellite Web Service is able to perform a simple transformation on a single media (which could possibly be a computationally heavy operation, e.g., transforming a video from a specific size to another). Finally, external Web Services can be exploited to perform transformation tasks which are not offered locally (placed on the same LAN of TU).

Summing up, TU is implemented as a two-level Web Services architecture in order to meet requirements derived from the dynamic nature of adaptation mechanisms. Indeed, the number of content adaptation typologies, as well as the set of multiple formats and related conversion schemes is still increasing. Thus, a notable advantage is gained by distributing all the adaptation activities over different Web Services and by consequently spreading the computational load.

Clearly, the use of Web Services guarantees flexibility, modularity and platform independence. Moreover, new Web Services might be easily plugged into the system so as to augment the available types of transformations.

The system performances of TU have been improved by using a two-level caching system (see Figure 6.1). Specifically, TU is supplied with a first level cache which maintains recently managed files, such as SMIL structures. A second level cache is provided to store recently produced media files which have been transcoded by each local Web

Service. The system takes advantage of caching systems by transcoding resources once and delivering them to users with similar profiles.

Once the learner has requested a LO, the client application contacts the system by authenticating itself to MB. As already mentioned, at its first access, the user specifies two profiles: an ACCLIP profile (to describe his/her accessibility preferences and related needs) and a CC/PP profile (to describe device capabilities).

As to PM, instead, this component is able to recall all the previously connected users. A database is utilized, which contains device characteristics and user preferences, with a set of mobile device hardware capability descriptions derived from Wireless Universal Resource File Library (WURFL) [167]. WURFL is an open source project that focuses on the problem of presenting content on the wide variety of wireless devices. The WURFL is an XML configuration file which contains information about device capabilities and features for a variety of mobile devices. Device information is contributed by developers around the world and the WURFL is updated frequently, reflecting new wireless devices coming on the market.

Moreover, we set a number of pre-configured standard profiles, in order to simplify the definition of user preferences. Users can decide whether to maintain a pre-set profile or to modify it by creating a new, personal and customized one. Each profile is identified by a unique user ID; during every access to the system, a user is simply required to specify only such unique ID.

Chapter 7

7. Experimental Assessment

This Chapter is devoted to assess the performances of the presented system. Due to its peculiarities, three notable aspects result to be of interest in our investigation. First, transcoding facilities on single media resources need to be assessed (see Section 7.2). Second, since we implemented the presented system as a distributed service-oriented architecture, a relevant issue is concerned with the efficacy of having all transcoding facilities distributed (see Section 7.3). Third, the efficacy of our caching subsystem is to be assessed (see Section 7.4). Final, the obtained LOs were subjected to the qualitative evaluation of real impaired learners (see Section 7.5). In Section 7.1 we introduce experimental scenarios.

7.1 Experimental Scenarios

Experiments have been conducted by transcoding different SMIL-based presentations packaged as SCORM-compliant LOs. Media comprised within these presentations have been chosen among a set of 20 resources such as videos, audios, images and text files. Random requests have been generated for presentations. Three hosts have been exploited to distribute all the components of our architecture. In particular, one node hosted the system components, while the other two ones have been devoted to host our local Web Services. Utilized machines have the following hardware characteristics: Pentium 4 – 2,5 GHz – with 1GB of RAM and hard-disk of 80 GB. Servers were running a LAMP (Linux, Apache, MySQL and PHP). To generate user requests, we exploited two personal computers equipped with Microsoft Windows XP, 2,5 GHz Pentium 4 CPU, 1 GB RAM, 80 GB hard disk. These hosts have been used to generate user requests. As to the client emulation, tests have been performed by using SOAtest Load Tester, an automated Web Service testing software, which is distributed by Parasoft [105]. During each single trial, the maximum number of (emulated) users connected to our system was set to 500.

For each request, a random user profile has been created. Exploited profiles might differ in screen dimensions settings (chosen from 784 entries, i.e., 28 possible screen widths and 28 possible screen heights), supported media formats (i.e., 7 different image formats, 3 video formats), accessibility constraints (e.g., use of assistive technologies, such as Braille display or screen reader, preference for not utilizing specific types of media, such as audio files or images), etc. All simulations had been taking place for one hour.

As to the exploited Web Services, we implemented three Web Services for specific transcoding operations, but also a third-party Web Service already available on the Internet has been used, which is able to convert text from a specific language to another one [144].

Our first Web Service manages and transforms several image formats; it is based on the open source library for media conversion ImageMagick [61].

Our second developed Web Service is able to convert a wide set of continuous media formats, based on the open source library for media conversion FFmpeg [42].

Three other developed Web Services are devoted to convert a SMIL presentation into several XHTML documents; multiple documents are created whenever display dimensions impose a split of the whole content. In particular, according to the scheduled transcoding strategy, such Web Services are typically utilized to create XHTML documents, possibly enclosing, respectively:

i) audio and text,
ii) images and text, or
iii) only text.

Finally, a Web Service provides a fragmentation of a unique content into several XHTML pages; it is used whenever display dimensions impose a split of the whole content.

7.2 On Assessing Single Transcoding and Adaptation Facilities

In this Subsection we are going to report on results related to the conversion of single media resources composing LOs in our assessments.

Basically, considered media comprise also those ones which are sketched in the use cases we described in Chapter 5.

Specifically, Table 7.1 shows times for conversion of a specific video file encoded as a MPEG file (800x600, size of 3,75 MB) to another MPEG video (with different dimensions) and Real Video formats, with varying dimensions. As reported in the table, times of conversion vary from 1.8 sec to 2.7 sec. As to audio, instead, time to convert a .wma file of 798 KB into the mp3 format, involves 0.7 sec. As to images, it results that compressing a 378 KB, 1024x768 JPEG image into a 240x320 one involves about 0.2 sec. Finally, conversion of a SMIL based document into a XHTML one involves, on average, 0.3 sec. These results demonstrate that viable transcoding strategies can be built, which exploit single resource conversions as building blocks for complex transcoding schemes.

Table 7.1 Times of Conversion for Different Media Resources

Media	***Original Dimensions***	***Original Format***	***Final Dimensions***	***Final Format***	***Transcoding Time (msec)***
Video	800x600	.mpg	240x320	.mpg	1856
Video	800x600	.mpg	800x600	.rm	2400
Video	800x600	.mpg	240x320	.rm	2730
Audio		.wma		.mp3	754
Image	1024x768	.jpg	240x320	.jpg	224
SMIL presentation		.smil		.xhtml	303

7.3 On Assessing the Distributed System: Results

To assess the efficacy of our distributed implementation of the system, we contrasted it against a transcoding system which has been implemented enclosing all local transcoding facilities in a unique local software component, i.e., without using any Web Services. Hereinafter we refer to this configuration as “monolithic”. Practically speaking, such a system configuration can be thought as a proxy-based architecture enclosing all facilities offered by our system.

As shown in Tables 7.2, 7.3 and Figures 7.1, 7.2, let us observe that our distributed system performs better than the monolithic system.

Table 7.2 Times of Transcoding LOs with distributed system vs Monolithic. Uniform Distribution

Trascoding system	***Min (msec)***	***Max (msec)***	***Avg (msec)***	***Completed Reqs***
Monolithic System	157	34970	6293	8421
Distributed System	143	15874	3631	8836

Table 7.3 Times of Transcoding LOs with distributed system vs Monolithic. Bell Curve

Trascoding system	***Min (msec)***	***Max (msec)***	***Avg (msec)***	***Completed Reqs***
Monolithic System	140	31249	2512	2020
Distributed System	135	11086	1824	2586

In particular, Table 7.2 reports results which have been obtained when the number of user requests is uniformly distributed during the period of simulation. In other words, we have assessed the system behavior during a steady trend of users' accesses. We can notice lower average, minimum and maximum response times for the system. In addition, also the number of user requests has been completed within the time of observation is higher.

Table 7.3 reports results we have obtained when the number of requests has been shaped as a typical bell curve, so as to assess the scalability of the system depending on the request number and to simulate a peak of requests. Peaks on the number of requests are quite common in the Web. For instance, such a conjuncture typically happens when a particularly popular resource is made accessible by some provider. In that case, service responsiveness becomes a real issue to be faced so as to avoid the "Slashdot" effect [77]. Even in this case, lower minimum, maximum and average response times have been obtained using the distributed system. Moreover, a distributed solution has shown to augment the number of completed transcoding processes.

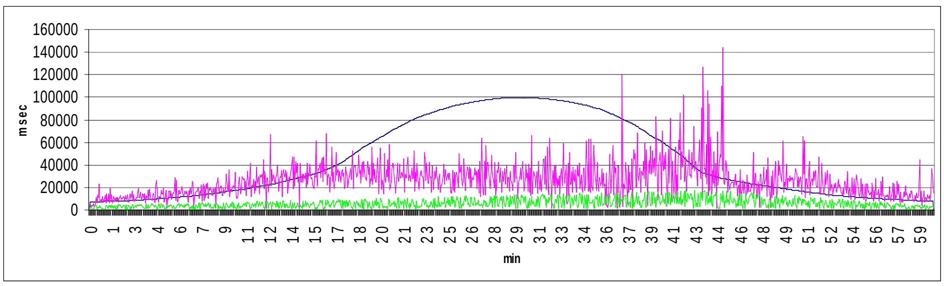

Figure 7.1 Average Execution Time (Monolithic System)

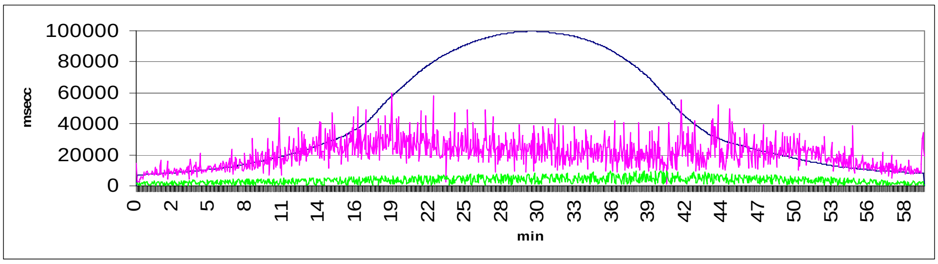

Figure 7.2 Average Execution Time (Distributed System)

Figures 7.1 and 7.2 report the average response times observed during an hour trail using, respectively, the monolithic system and the distributed system (see the upper non-regular curves depicted in pink within the two Figures). The regular dark (blue) line reproduces (out of scale) the number of users who are connected at a given time, waiting for requested contents. The lower non-regular light (green) line, instead, represents the average packaging time, calculated by measuring times to un-package and re-package (transcoded) LOs.

From these Figures, it is possible to observe a more regular behavior using our distributed implementation of the system. Indeed, the monolithic system presents a peak in response times, thus underlining a bottle-neck in the system, after a higher number of users have issued a request. Practically speaking, when the number of contemporary requests is above a threshold number, the monolithic system presents performance degradation.

7.4 On the Efficacy of Using Caches

In this Subsection, we are going to show results we obtained through the use of different cache settings. In particular, we have contrasted four different caching policies:

i) first and second level caches both disabled; this represents a worst-case scenario that does not resort to caches.

ii) First level cache enabled while second level cache disabled. This scenario represents a typical situation of use when external Web Services without local caching policies are utilized.

iii) Second level cache enabled while first level cache disabled, i.e., we assessed the situation where TU does not resort to caching systems.

iv) Both first level and second level caches enabled (our default); this scenario measures the efficacy of combining local and global caching policies.

All caches have been refreshed every 20 minutes. Table 7.4 shows the average execution times and the average number of completed transcoding processes, which have been obtained by resorting to the different caching policies. It is worth noticing that higher performances (i.e., a lower average time and a higher number of completed requests) have been obtained when both types of caches are utilized. Furthermore, results show that higher improvements have been obtained when caches locally employed at Web Services have been activated.

Table 7.4 Transcoding Times Enabling and Disabling Two Level Caching System

Caching System	***Avg (msec)***	***Completed Requests***
Disabled Caches	1824	2495
First Enabled Level (TUWS Cache)	1645	2573
Second Enabled Level (Web Services Caches)	1392	2588
Two Cache Types Enabled	1367	2602

It is important to notice that the cache size at each node has been set to 5-8 GB. We point out that the distributed system can trigger conversion of rich media such as high definition videos. Thus, the cache size must be properly set to avoid, on one side, that (final) large sized resources are continuously transcoded at each request (since no space is available on the cache for those resources), and, on the other side, to maintain a huge cache at each host (in some sense, this solution corresponds to maintaining every resource format pre-processed at the server-side).

In this respect, however, it is also important to notice that, typically, in scenarios of use in mobile, accessible e-learning, video compression is performed to transform high quality videos into very compressed ones (for instance, videos to be displayed on mobile terminals). These are, probably, the most computation intensive transcoding operations in our system. In other words, only few high quality video formats are delivered to fully equipped users. (It is quite uncommon to have requests for high quality videos that need to be slightly compressed.) Instead, a plethora of possible low quality video formats can be provided for delivery to (mobile or network-constrained) users. These last ones are small sized files which are easily maintained within nodes caches. Summing up, the higher the requested degradation is, the higher the computation is, which is needed for the transformation, but the lower the file size will be. Thus, our two-level caching system can be put of real good use to support system activities and augment scalability, as confirmed by our experiments.

7.5 Subjective Evaluation

Dozens of users with disabilities were invited to interact with our system in order to enjoy lectures transcoded on the basis of their profiles. In particular, blind users and users with motion impairments were enrolled. After the test phase, users were asked to assign a score (from 1 to 6, the higher the better) to the system accessibility. Average scores are shown in Table 7.5. All the users gave a positive score and, in particular, blind users showed an enthusiastic reaction at the lecture provided based on media alternative to video. Also people with motion impairments gave a positive evaluation on the system, mainly due to the fact that they were allowed to interact with it exploiting alternative interaction methods (e.g., vocal commands). Some of users' comments were (translated from Italian to English):

- "*The final produced content is very good. It is possible to notice the hard work done in order to reach the maximum level of accessibility*".
- "*The lecture accessibility overcomes all my expectations. The synchrony between resource media alternatives (audio and caption based) is perfect and fully enjoyable*".

Table 7.5 Users evaluation

Users	***Average Assigned Scores***
Blind users	6
Users with mobility impairments	5

Chapter 8

8. Related Works

The aim of this Chapter is illustrating main related works and discussing analogies and differences in comparison with the presented work.

First, Section 8.1 discusses main adaptation architectural solutions, by comparing them with our system. Second, in Section 8.2, we will consider schemes devised to schedule content adaptation. Third, Section 8.3 will debate techniques for structuring multimedia contents and modeling multimedia adaptation processes. Fourth and final, in Section 8.4, we will presents some learners' profiling application projects.

8.1 Adaptation Architectural Solutions

As described in Chapter 2, the most significant distributed architectural solutions for content adaptation and transcoding are grouped into four main categories [30] [80]:

i) solutions applying *client-side* approaches.
ii) Solutions applying *server-side* approaches.
iii) Solutions applying *proxy-based* approaches.
iv) Solutions applying *service-oriented* approaches.

In the following Subsections, we are going to present main advantages and disadvantages of solutions which applied such approaches.

8.1.1 Client-side approach

The main advantages of the client-side adaptation of resources are due to the obvious knowledge that client applications may obtain from their device capabilities. Several kinds of adaptations and adjustments may be performed on the client-side, by occurring in the content delivery device (typically the Web browser). Many browsers, for example, let the user increase or decrease document font sizes.

Client-side adaptations can also be computed based on directives contained in the content itself. The most prominent example of author-controlled adaptation performed at the client-side is the use of Cascading Style Sheets (CSS) [149], which authors use to style HTML [153] (or XHTML [163]) documents, Scalable Vector Graphics (SVG) [158], or even plain XML content. Separating style from content is accepted as a good practice for managing data and enables authors to provide different styles to suit different devices. In CSS, authors can define different styling rules for different media types. CSS media types are names that identify different devices, such as screen, handheld, TV, print, projection, aural, and Braille display. Based on the use of CSS media types, for example, user agents (on smaller devices) may be forced to omit the visualization of those parts of Web pages which are useless. Needless to say, this kind of approach presents several limitations. First, it is often not practicable to send information over a network that must not be displayed, or even different versions of the same media encoded in different formats. This limitation is particularly relevant when the communication is performed according to one-to-one transmission protocols. Second, this solution increases the computational overheads on the client side. This kind of approach is therefore inadequate when clients gain access to those multimedia resources by means of low end devices.

A different behaviour of such an approach consists in sending multiple different formats to clients, who have to choose the most adequate one in order to play it out [81] [169].

To conclude, adaptations that can benefit a group of clients with similar needs can be more efficiently implemented with server-side or proxy-based approaches. Furthermore, not all clients may be able to implement content adaptation techniques due to processor, memory resource constraints and limited network bandwidth.

Considering our context, such solutions are too naive and do not completely and effectively meet learners requirements in providing adequately adapted rich LOs. However, our system takes into account a basic set of client-side adaptations, in order to provide typical and overall diffused mechanisms related to CSS and SMIL standards.

8.1.2 Server-side approach

These solutions are clearly more flexible and general than a client-side approach and minimize the use of network. However, dynamic transcoding skills must be installed at the server, which has to provide contents and to perform the additional functional of content adaptation [79] [91] [100] [128].

The main advantages of using server-side adaptation architecture are as follows:

- The content can be originally written in XML, and then transformed to other markup languages according to browser capabilities.
- The server usually has much more processing power than the client devices.
- The system is simple and easy to connect with databases or applications installed on the server.
- With content negotiation of a client device, the server can control the presentation layer and send content that the device is able to present.
- The server can have full knowledge of its content, thus increasing the possibility of displaying content on most browsers.

On the other hand, the most important defects of using server-based content adaptation architecture are:

- Not all browsers support content negotiation or the server may not recognize all browsers data. So the server must make assumptions or use default parameters on the browser’s ability to present the content.

- Scalability is a key issue because the centralized server has to manage all content adaptation requests. Heavy server-side applications may slow down the server.

Such solutions, as well as the previous one, are not adequate in order to support rich LOs adaptation, hence we have decided to consider different and newest solutions, which are described in the following Subsections.

8.1.3 Proxy-based approach

One of the main problems in applying a proxy-based solution is that these adaptation approaches focus on particular types of adaptation such as image transcoding, HTML [153] to WML [140] conversion, etc. and these are specific applications [19] [20] [26] [49] [50] [83] [84] [121]. In addition, if all adaptations are done at the proxy, it results in computational overload, as some adaptations are computationally intensive and this degrades the performance of information delivery, just as in the server-side approach.

This approach has evolved through many forms, most of them related to caching. More recently, the community refers to this intermediate node with multiple definitions, such as edge server, surrogate sever and secondary server, with the implicit meaning that it can support active functionalities beyond caching in network locations that are closer to the client [30] [80]. In intermediary-based adaptation, most work is carried out by the nodes placed between the platform of the provider and the client and can benefit caching of already adapted resources.

Certainly, such solutions are the most effective among the previously described ones. Moreover, comparing them with our context points out the need of adding new kind of adaptation and transcoding operations. Proxy-based solutions do not support an easy mechanism in order to modify the set of available transformations.

8.1.4 Service-based approach

The main aim of this approach is to distribute roles and computational load more efficiently [10] [69] [70], in order to obtain a modular architecture and allow new transcoding services to be added if needed. This is one of the main advantages in distributing all the adaptation activities over different Web Services.

While this approach would provide a valuable service for the end customer, the service provider and the content provider, it is important to have an architectural framework which is simple, scalable, flexible and interoperable [10]. On one hand, Web Services are becoming popular technologies for publishing various services on the Internet [101]. On the other hand, there is a trend in developing content adaptations as value added services. However, the link between them has not yet been explored, i.e. using Web Services for the purposes of developing content adaptation services. Moreover, deciding what adaptations to perform and which services to select in order to maximize performance and minimize costs can be a complex constraint satisfaction problem.

Introducing content adaptation as a service distributes the activities and results in performance enhancement especially for computational intensive applications. For example, a server that handles only language translation is inherently more efficient than any standard Web server performing many additional tasks. It also opens new opportunities to service providers as additional revenue. However, it is very important to have an architectural framework to enable a content delivery system to incorporate such functionalities. Such a system needs a basic mechanism to configure and run various services by selecting suitable ones from a list of those available and deciding on the most appropriate configuration.

Some work has been done in this direction. For instance, in [10] the authors present an architecture that enables the use of third-party adaptation services by means of content negotiation and adaptation models. The devised adaptation system is devoted to transforming images, video, audio and text. In [139], the authors argue that semantic Web Services can serve as a key to enable technology to achieve the goal of "universal multimedia access", so that users can consume any multimedia resources anywhere, at any time, and using any device. Therefore, in their paper the authors stress the importance of changing classic multimedia adaptation functionalities into a set of effectively selected Web Services.

Such solutions seem finally meet the requirements our context imposes. As described in Chapter 6, we have developed a system architecture which leans on service-based approach, by exploiting Web Services features [104].

8.2 Scheduling the Content Adaptation Process

An important issue is how the software component that performs content adaptation is organized, i.e. how the transcoding tasks are scheduled in order to adapt contents for a proper play-out to a specific user.

A straightforward solution for implementing a content adaptation process is that of structuring it as a pipeline [31] [36] [46] [69]. In this solution, transcoding and adaptation activities are performed sequentially. Such a solution facilitates the composition of all the necessary steps to adapt and transcode contents in compliance with client device profiles and user preferences. Moreover, it is possible to introduce new modules to the pipeline for different typologies of adaptation and transcoding activities, taking into account the availability of new encoding formats as well as new types of device.

Starting from a pipeline-based structure, a more interoperable solution has been suggested which resorts to the introduction of a broker within the system architecture [49] [55]. A broker is an intermediate system in charge of identifying a user's needs and facilities offered by media adaptation components. This allows a complete match between constraints imposed by clients and provided resources, hence it is the approach we have chosen in order to schedule adapting and transcoding operation in our system [39] [90] [111].

8.3 Structuring Multimedia Contents

In recent years, a large number of projects have focused on multimedia transcoding, each one presenting its own, often very different, approach. Several works have presented transcoding systems for optimized management of just a single type of media [23] [24] [25] [49] [86] [108] [138]. Similar transcoding mechanisms are commonly used at Web servers to customize the size of objects that make up a Web page, to provide a low-latency access to contents and differentiated quality of services. However, these schemes lack the capabilities to simultaneously manage multiple kinds of media. Moreover, they statically convert contents in an off-line approach.

Other examples have been devised to dynamically decide whether it is the case to degrade rich multimedia contents into poorer versions in order to meet device capabilities and user preferences. A seminal work in this direction is presented in [91] [128]. In this work, multimedia contents are represented using a structure, called InfoPyramid. By exploiting this structure, the system is able to transcode video, image, audio and text in different resolutions and different modalities. Contents can thus be played-out on a variety of devices. The main limitation of the devised solution is that the transcoding process is done off-line. Moreover, the authors suppose that each media item is embodied in a single Web object, as an atomic element. Thus, no temporal or spatial relationships among different media, nor synchronization or layout issues are considered.

A more recent work [43] [55] fills this gap in part, by considering spatial relationships among media components of a multimedia presentation. In this project media (text, image, audio and video) are described using a specific XML-based grammar. However, no focus is placed on temporal relationships among media, nor synchronization issues related to alternative multimedia presentations produced by adopting general transcoding processes, which considerably alter the structure of the content.

As regards synchronization between media and their different transformations in adaptive multimedia presentations, SMIL represents a promising new technology [159]. In particular, in SMIL 2.0 specifications, the *Timing and Synchronization Module* offers a set of elements and attributes which are devoted to managing media synchronization in multimedia presentations. This language enables the transcoding of single media included in the SMIL presentation, while maintaining the original synchronization specification.

Due to its special features, SMIL is becoming a widely exploited technology in multimedia adaptation. For example, in [82] the authors propose a SMIL content adaptation framework for mobile devices, based on a three-tier scheme for content access. The tool is able to adapt contents to meet client profiles and manage the presentation layout. However, this system is not able to convert a given type of media into another (e.g. from video to a set of images, from an image format to alternative text or from audio to its related captions).

As a matter of fact, SMIL represents a key technology in this specific research field that must be certainly taken into account when designing a novel system for content adaptation. However, while researchers push towards a massive use of this language, customers seem to be still unaware of its existence. Thus, several questions arise regarding what happens when users have no SMIL player on their devices or how can providers distribute contents in this situation or, finally, how can media synchronization be maintained. With this in view we have designed and developed the presented system.

8.4 Profiling Learners

In this Subsection we are going to illustrate some e-learning projects devoted to profile learners. The Inclusive Learning Exchange (TILE) [97] [131] is a learning object repository developed by the Adaptive Technology Resource Centre at the University of Toronto, which implements both ACCMD [63] and ACCLIP [65]. Whenever authors use the TILE authoring tool to aggregate and publish learning objects, they are supported in creating and appropriately labelling transformable aggregated lessons (codified by the TILE system using ACCMD). Learners are enabled to define their learner preferences, which are then stored as IMS ACCLIP records. Thanks to such information, TILE inspects the state preferences of the learner and computes the best resource configuration by transforming or re-aggregating the lesson.

The Web-4-All project [97] [141] is a collaboration between the Adaptive Technology Resource Centre, the Web Accessibility Office of Industry Canada and the IMS Global Learning Consortium. This project allows learners to automatically configure a public access computer by using a learner preferences profile implemented with the ACCLIP and stored on a smartcard. Thanks to information stored within its smartcard, each learner can freely switch from one public workstation to another. When the smartcard is read by the workstation, the Web4All software automatically configures the operating system, the browser and all the necessary assistive technologies, based on the learner profile. If the assistive technology requested by a learner is not available on a workstation, the program launches and configures the closest approximation.

The PEARL (Practical Experimentation by Accessible Remote Learning) project [97] [109] is a European Commission funded project led by the Open University, in the UK. A technical framework teaching laboratory for science and engineering has been developed to be offered to remote students. The project's main aim is that of increasing the participation of disabled students in these subjects by offering increased access to practical work. Thus, interfaces are generated "on the fly", based on XML elements describing single interface components and based on the supported types of interaction.

Such projects have been done in the direction of managing Learning Objects (LOs) based on the idea of adapting contents and their presentation in a suitable way. Yet, none of these ones took into account device capabilities, contrary to our system. As a consequence, their LO adaptation can not be effectively completed so as to effectively meet more general users requirements.

Chapter 9

9. Conclusions

E-learning systems represent a fundamental means to offer educational services to people with disabilities, who typically have difficulties to attend traditional on-site learning programs or to gain access to traditional printed learning materials. Moreover, mobile e-technologies represent effective means to match skills of disabled learners and requirements/demands of the environment surrounding them, because of devices limited capabilities.

In order to face these issues, we developed an automatic system for the production of accessible and portable learning materials, which may be of real help to surmount physical and environmental barriers that users can encounter during their learning activities. The system offers a broking service to transcode digital video-lectures based on the specific student and device profile. Thus, students with disabilities may gain access to contents by means of assistive adaptive technologies.

What is new in this system is that both device and human limitations are dynamically considered during the transcoding process. By coupling these two issues, the whole "anytime, anywhere, anyone and any device" slogan can be achieved.

Such a system works on (SMIL-based) rich media contents, which are widely utilized to improve the efficacy of Web-based learning systems, but, at the same time, are typically difficult to be ported from a device to another and present several characteristics that compromise accessibility. These difficulties are surmounted by resorting to a distributed service oriented architecture. A broker is responsible to analyze user and client device profiles and to produce a suitable transcoding strategy to adapt the requested rich media content (i.e., the LO). It is also able to manage (i.e., unpackage and create) SCORM-compliant LOs so as to comply with this e-learning standard.

Transcoding steps are accomplished by different distributed Web Services, which can be dynamically plugged into the system. Separation of capabilities and tasks becomes very important in such a particular context, where different possible user profiles, transcoding preferences and media content types exist.

Results obtained from a real experimental assessment confirm the viability of our approach and that the distribution of all transcoding facilities represents an important means to augment scalability and system performances. Moreover, we also showed that the use of a caching system can be put of good use to improve the overall system performances. As a matter of facts, a smart use of the caching system could be developed enabling the maintenance of topic, highly requested encoding formats (e.g., high definition videos, 800x600 MPEG videos), which can be easily converted to other formats. This way, once a target media format is requested, the system can retrieve the more similar one.

9.1 Future Works

Main future efforts will be devoted to exploit presented mechanisms in new emerging e-learning systems. More specifically, we have pointed out two different technologies which could offer support to the provision of mobile and accessible learning materials. The first one is the client-server architecture used in podcasting. Our proposal could easily improve both accessibility and mobility of currently podcasted lectures, by maintaining the same architectural approach. A more complex architectural challenge is represented by the idea of re-design the whole system to be used in a P2P environment.

Another interesting future work will regard the employ of different multimedia formats to code the rich media source (video-lecture), such as MPEG-21.

Finally, in order better prove the whole presented work feasibility it is useful to provide an adequate e-learning content authoring system. Such a system support authors in producing rich media contents, driving them in creating content metadata and media alternatives. Such efforts introduce an obvious overhead in authoring activities. We have already done some works in this direction [12] [34] [35], which are still on progress.

References

[1] Advanced Distributed Learning. Available from: http://www.adlnet.org, 2006.

[2] Advanced Distributed Learning. Content Aggregation Model (CAM), Sharable Content Object Aggregation Model (SCORM) Version 1.3, retrieved October, 2006. Available from http://www.adlnet.org/downloads/files/67.cfm, 2004.

[3] Advanced Distributed Learning. Run Time Environment (RTE), Sharable Content Object Aggregation Model (SCORM) Version 1.3, retrieved October, 2006. Available from: http://www.adlnet.gov/downloads/files/194.cfm, 2004.

[4] Advanced Distributed Learning. Sequencing and Navigation (SN), Sharable Content Object Aggregation Model (SCORM) Version 1.3, retrieved October, 2006. Available from: http://www.adlnet.gov/downloads/files/300.cfm, 2004.

[5] Advanced Distributed Learning. Sharable Content Object Reference Model (SCORM) 2004 2nd Edition Document Suite, retrieved October, 2006. Available from: http://www.adlnet.org/downloads/70.cfm, 2004.

[6] AICC Aviation Industry CBT Committee. Available from: http://www.aicc.org, 2006.

[7] AICC Aviation Industry CBT Committee. AICC Guidelines and Recommendations (AGR's). Available from: http://www.aicc.org/pages/down-docs-index.htm#AGR, 2006.

[8] L. Anido. An observatory for e-learning technology standards. In *Advanced Technology for Learning archive*, Vol. 3, No. 2, pages 99-108, ACTA Press, April 2006.

[9] J. A. Barron, L. Fleetwood and A. E. Barron. E-Learning for Everyone: Addressing Accessibility. In *Journal of Interactive Instruction Delivery*, Vol. 16, No. 4, pages 3-10, 2004.

[10] G. Berhe, L. Brunje and J. Pierson. Modeling service-based multimedia content adaptation in pervasive computing. In *Proceedings of the first Conference On Computing Frontiers* (*CF '04)*, pages 60-69, April 2004.

[11] P. Bohman. Introduction to Web Accessibility. Available from: http://www.webaim.org/intro/, 2003.

[12] M. Boni, S. Cenni, S. Mirri, L. A. Muratori and P. Salomoni. Automatically Producing IMS AccessForAll Metadata. In *Proceeding of the International Cross-Disciplinary Workshop on Web Accessibility - Located at the Fifteenth International World Wide Web Conference (WWW '06)*, pages 92-97, May 2006.

[13] C. Bouras and M. Nani. A Web-based tool for building and accessing learning objects and online courses. In *Proceedings of the IEEE International Conference on Advanced Learning Technologies*, pages 645-647, August 2004.

[14] R. K. Boyd and B. Moulton. Universal design for online education: access for all. In *The distance education evolution: issues and case studies*, pages 67-115, Idea Group Publishing, 2004.

[15] D. C. Brown, E. Burbano, J. Minski and I. E. Cruz. Evaluating Web page color and layout adaptations. In *IEEE Multimedia*, Vol. 9, No. 1, pages 86-89, IEEE Communications Society, January-March 2002.

[16] I. Burnett, R. Van de Walle, K. Hill, J. Bormans and F. Pereira. MPEG-21: Goals and Achievements. In *IEEE Multimedia*, Vol. 10, No. 6, pages 60-70, IEEE Communications Society, October - December 2003.

[17] M. Butler, F. Giannetti, R. Gimson and T. Wiley. Device Independence and the Web. In *IEEE Internet Computing*, Vol. 6, No. 5, pages 81-86, IEEE Communications Society, September - October 2002.

[18] S. Cain. Accessing Technology - Using technology to support the learning and employment opportunities for visually impaired users. Royal National Institute for the Blind, 2001.

[19] C. Canali, V. Cardellini, M. Colajanni, R. Lancellotti and P. S. Yu. Cooperative architectures and algorithms for discovery and transcoding of multi-version content. In the *Proceedings of the 8th International Workshop on Web content caching and distribution*, pages 205-221, October 2004.

[20] C. Canali, V. Cardellini and R. Lancellotti. Content Adaptation Architectures Based on Squid Proxy Server. In *World Wide Web*, Vol. 9, No. 1, pages 63-92, Kluwer, March 2006.

[21] F. Cardinali. Towards Narrowcasting & Ambient Learning: New Mobile, Location and Context Aware Formats for the Content Industry towards the Knowledge Society. In *Proceedings of the E-Learning Conference*, May 2005.

[22] A. Celentano and O. Gaggi. Context-aware design of adaptable multimodal documents. In *Multimedia Tools and Applications*, Vol. 29, No. 1, pages 7-28, Kluwer, April 2006.

[23] S. Chandra, C. S. Ellis and A. Vahdat. Differentiated Multimedia Web Services Using Quality Aware Transcoding. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00)*, IEEE Communications Society, March 2000.

[24] S. Chandra, C. S. Ellis and A. Vahdat. Application-Level Differentiated Multimedia Web Services Using Quality Aware Transcoding. In *IEEE Journal on Selected Areas in Communication*, Vol. 18, No. 12, pages 2544-2564, IEEE Communications Society, December 2000.

[25] S. Chandra, A. Gehani, C. S. Ellis and A. Vahdat. Transcoding Characteristics of Web Images. In *Proceedings of the SPIE Conference on Multi-Media Computing and Networking (MMCN '01)*, Vol. 4312, pages 135-149, January 2001.

[26] C. Y. Chang and M. S. Chen. On exploring aggregate effect for efficient cache replacement in transcoding proxies. In *IEEE Transactions on Parallel and Distributed Systems*, Vol. 14, No. 6, pages 611-624, IEEE Communications Society, June 2003.

[27] S. C. Cheng, C. W. Su and Y. T. Lin. Mobile Learning with Intelligent Download Suggestions. In *Learning Technology*, Vol. 7, No. 2, IEEE Computer Society Technical Committee on Learning Technology (LTTC), April 2005.

[28] J. Clark. Building Accessible Websites. New Riders Publishing, Indianapolis, 2002.

[29] R. C. Clark and R. E. Mayer. E-Learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning. Pfeiffer, 2002.

[30] M. Colajanni and R. Lancellotti. System Architectures for Web Content Adaptation Services. In *IEEE Distributed Systems online*, Vol. 5, No. 5, IEEE Communications Society, May 2004.

[31] A. Cook and S. Hussey. Assistive Technologies - Principles and Practice, 2nd Edition. Mosby, 2002.

[32] K. Curran and S. Annesley. Transcoding media for bandwidth constrained mobile devices. In the *International Journal of Network Management*, Vol. 15, No. 2, pages 75-88, John Wiley & Sons, March-April 2005.

[33] E. de Lara, Y. Chopra, R. Kumar, N. Vaghela, D. S. Wallach and W. Zwaenepoel. Iterative adaptation for mobile clients using existing APIs. In *IEEE Transactions on Parallel and Distributed Systems*, Vol. 16, No. 10, pages 966-981, IEEE Communications Society, October 2005.

[34] A. Di Iorio, A. A. Feliziani, S. Mirri, P. Salomoni and F. Vitali. Automatically Producing Accessible Learning Objects. To appear in a *Special Issue of the Journal of Educational Technology & Society*.

[35] A. Di Iorio, A. A. Feliziani, S. Mirri, P. Salomoni and F. Vitali. Simply Creating Accessible Learning Object. In *Proceedings of eLearning and Human-Computer Interaction: Exploring Design Synergies for more Effective Learning Experiences (INTERACT 2005 Workshop)*, September 2005.

[36] M. Driscoll and S. Carliner. Advanced Web-Based Training Strategies: Unlocking Instructionally Sound Online Learning. Pfeiffer, 2005.

[37] L. Elson and A. Cerpa. ICAP - the Internet Content Adaptation Protocol. RFC 3507, the ICAP Protocol Group, 2003.

[38] European Union. eInclusion & eAccessibility. Available from: http://europa.eu.int/information_society/policy/accessibility/index_en.htm, 2006.

[39] S. Ferretti, S. Mirri, M. Roccetti and P. Salomoni. A Multimedia Broker to support Accessible and Mobile Learning through Learning Objects Adaptation, submitted to ACM TOIT Special Issue on Internet Technologies for Distance Education.

[40] S. Ferretti, S. Mirri and P. Salomoni. Adapting Rich Multimedia Contents for Context-Aware Delivery Services,

submitted to International Journal of Virtual Technology and Multimedia.

[41] S. Ferretti, S. Mirri and P. Salomoni. Transcoding in Online Rich-media Edutainment: a Dynamic Approach based on Content/Context Awareness. To appear in *Proceedings of the 3rd IEEE International Workshop on Networking Issues in Multimedia Entertainment (NIME'07) - 3th IEEE Communications and Networking Conference (CCNC 2007)*, IEEE Communications Society, January 2007.

[42] FFMPEG. FFmpeg Multimedia Systems. Available from: http://ffmpeg.sourceforge.net/index.php, 2006.

[43] Z. Fiala, M. Hinz, K. Meisner, and F. Wehner. A Component Based Approach for Adaptive Dynamic Web Documents. In *Journal of Web Engineering*, Vol. 2, No. 1&2, pages 58-73, September 2003.

[44] D. Forman. Cultural change for the e-world. In *Proceedings of the International Conference on Computers in Education*, Vol. 2, pages 1412-1413, December 2002.

[45] Freedom Scientific. Jaws for Windows Overview. Available from: http://www.freedomscientific.com/fs_products/software_jaws.asp, 2006.

[46] S. Gabrielli, V. Mirabella, S. Kimani and T. Catarci. eLearning Accessibility: A Comprehensive Approach to Content Development. In *Proceedings of eLearning and Human-Computer Interaction: Exploring Design Synergies for more Effective Learning Experiences (INTERACT)*, September 2005.

[47] R. Guan. Content Adaptation on Mobile Phones. Master's Thesis. Technical University of Denmark Kgs. Lyngby, Denmark, 2003.

[48] M. L. Guenaga, D. Burger and J. Oliver. Accessibility for e-Learning Environments. In *Computers Helping People with Special Needs, Lecture Notes in Computer Science*, pages 157-163, Springer Berlin / Heidelberg, 2004.

[49] R. Han, P. Bhagwat, R. LaMaire, T. Mummert, V. Perret and J. Rubas. Dynamic Adaptation in an Image Transcoding Proxy for Mobile WWW Browsing. In *IEEE Personal Communication*, Vol. 5, No. 6, pages 8-17, IEEE Communications Society, December 1998.

[50] R. Han and J. R. Smith. Transcoding of the Internet's multimedia content for universal access. In *Communications, Networking, And Multimedia, Multimedia communications: directions and innovations*, pages 261-296, Academic Press, 2001.

[51] L. Harrison and J. Treviranus. Accessible E-Learning - Demystifying IMS Specifications. In *G. Richards (Ed.), Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education (ELEARN)*, pages 2000-2003, 2003.

[52] K. Harumoto, T. Nakano, S. Fukumura, S. Shimojo and S. Nishio. Effective Web browsing through content delivery adaptation. In *ACM Transactions on Internet Technology (TOIT)*, Vol. 5, No. 4, pages 571-600, ACM Press, November 2005.

[53] B. P. Heath, R. L. Herman, G. G. Lugo, J. H. Reeves, R. J. Vetter and C. R. Ward. Project Numina: Enhancing Student Learning with Handheld Computers. In *Computer*, Vol. 38, No. 6, pages 46-53, IEEE Computer Society, June 2005.

[54] E. Heinrich and J. Chen. A Framework for the Multi-modal Description of Learning Objects. In *Proceeding of the International Conference on Dublin Core and Metadata Applications website*, page 32?, October 2001.

[55] M. Hinz and Z. Fiala. AMACONT: A System Architecture for Adaptive Multimedia Web Applications. In *Workshop XML Technologies for Semantic Web (XSW 2004)*, October 2004.

[56] C. Hofstader. Internet accessibility: beyond disability. In *Computer*, Vol. 37, No. 9, pages 103-105, September 2004.

[57] S. Hoh, S. Gillies and M. R. Gardner. Device Personalization - Where Content Meets Device. In *BT Technology Journal*, Vol. 21, No. 1, pages 67-74, Kluwer, January 2003.

[58] W. Horton and K. Horton. E-learning Tools and Technologies: A consumer's guide for trainers, teachers, educators, and instructional designers. John Wiley & Sons, 2003.

[59] M. Hricko. Design and Implementation of Web-Enabled Teaching Tools. Information Science Publishing, October 2002.

[60] IEEE. IEEE Learning Technology Standards Committee Home page. Available from: http://ieeeltsc.org/, 2006.

[61] ImageMagick. ImageMagick - Convert, Edit, and Compose Images. Available from: http://www.imagemagick.org/, 2006.

[62] IMS Global Learning Consortium. Available from: http://www.imsglobal.org, 2006.

[63] IMS Global Learning Consortium. IMS AccessForAll Meta-data Specification. Available from: http://www.imsglobal.org/specificationdownload.cfm, 2002.

[64] IMS Global Learning Consortium. IMS Guidelines for Developing Accessible Learning Applications. http://www.imsglobal.org/accessibility/, 2006.

[65] IMS Global Learning Consortium. IMS Learner Information Package Accessibility for LIP. Available from: http://www.imsglobal.org/specificationdownload.cfm, 2002.

[66] IMS Global Learning Consortium. IMS Learner Information Profile (LIP). Available from: http://www.imsglobal.org/specificationdownload.cfm, 2002.

[67] ISO/IEC JTC1 Information Technology for Learning, Education, and Training. Text of FCD 24751-1, Individualized Adaptability and Accessibility in E-learning, Education and Training Part 1: Framework. November 2001.

[68] Italian parliament. Law nr. 4 – 01/09/2004. Official Journal nr. 13 – 01/17/2004, January 2004.

[69] D. Jannach, K. Leopold, C. Timmerer and H. Hellwagner. Toward Semantic Web Services for Multimedia Adaptation. In *Proceedings of Web Information Systems Engineering (WISE '04)*, pages 641-652, November 2004.

[70] D. Jannach, K. Leopold, C. Timmerer and H. Hellwagner. A knowledge-based framework for multimedia adaptation. In *Applied Intelligence*, Vol. 24, No. 2, pages 109-125, Kluwer, April 2006.

[71] P. Karampiperis and D Sampson. Learning Object Metadata for Learning Content Accessibility. In *Proceedings of the 16th World Conference on Educational Multimedia, Hypermedia and Telecommunications (EDMEDIA)*, pages 5204-5211, June 2004.

[72] P. Karampiperis and D. Sampson. Supporting accessible hypermedia in web-based educational systems: defining an accessibility application profile for learning resources. In *The New Review of Hypermedia and Multimedia*, Vol. 10, No. 2, pages 181-197, Taylor & Francis, December 2004.

[73] B. Kelly and L. Phipps. Holistic approaches to e-learning accessibility. In *ALT-J, Research in Learning Technology*, Vol. 14, No. 1, pages 69-78, Routledge, part of the Taylor & Francis Group, March 2006.

[74] B. Kelly, D. Sloan, L. Phipps, H. Petrie and F. Hamilton. Forcing standardization or accommodating diversity?: a framework for applying the WCAG in the real world. In *Proceedings of the 2005 International Cross-Disciplinary Workshop on Web Accessibility (W4A)*, pages 46-54, ACM Press, May 2005.

[75] B. H. Khan. Managing E-Learning Strategies: Design, Delivery, Implementation and Evaluation. Information Science Publishing, 2005.

[76] L. K. Kan, X. Peng and I. King. A user profile-based approach for personal information access: shaping your information portfolio. In the *Proceedings of the 15th international conference on World Wide Web (WWW2006)*, Poster Session "Browsers and UI, web engineering, hypermedia & multimedia, security, and accessibility", pages 921-922, ACM Press, May 2006.

[77] L. Kleinrock and R. Gail. Queueing Systems: Problems and Solutions. John Wiley and Sons, 1996.

[78] J. Korva, J. Plomp, P. Määttä and M. Metso. On-line service adaptation for mobile and fixed terminal devices. In *Proceedings of the 2nd International Conference on Mobile Data Management*, pages 252-259, January 2001.

[79] B. Knutsson, H. Lu, J. Mogul and B. Hopkins. Architecture and performance of server-directed transcoding. In *ACM Transactions on Internet Technology (TOIT)*, Vol. 3, No. 4, pages 392-424, ACM Press, November 2003.

[80] T. Laakko and T. Hiltunen. Adapting Web Content to Mobile User Agents. In *IEEE Internet Computing*, Vol. 9, No. 2, pages 46-53, IEEE Communications Society, March-April 2005.

[81] Z. Lei and N. D. Georganas. Context Based Media Adaptation in Pervasive Computing. In *Proceedings of IEEE*

Canadian Conference on Electrical and Computer (CCECE '01), pages 913-918, IEEE Communications Society, May 2001.

[82] T. Lemlouma and N. Layaida. SMIL Content Adaptation for Embedded Devices. In *Proceedings of the Synchronized Multimedia Integration Language European Conference*, February 2003.

[83] K. Li and H. Shen. Coordinated enroute multimedia object caching in transcoding proxies for tree networks. In *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, Vol. 1, No. 3, pages 289-314, ACM Press, August 2005.

[84] W.Y. Lum and F.C.M. Lau. A Context-Aware Decision Engine for Content Adaptation. In *IEEE Pervasive Computing*, Vol. 1, No. 3, pages 41-49, IEEE Communications Society, July-September 2002.

[85] D. S. Metcalf. M-Learning: Mobile E-Learning, HRD Press, Inc, 2006.

[86] M. Metso, A. Koivisto and J. Sauvola. A Content Model for the Mobile Adaptation of Multimedia Information. In *Journal of VLSI Signal Processing Systems*, Vol. 29, No. 1-2, pages 115-128, Kluwer, August-September 2001.

[87] M. Metso, M. Loytynoja, J. Korva, P. Määttä and J. Sauvola. Mobile Multimedia Services – Content Adaptation. In *Proceedings of the 3rd International Conference on Information, Communications and Signal Processing (ICICS '01)*, October 2001.

[88] V. Mirabella, S. Kimani, S. Gabrielli and T. Catarci. Accessible e-learning material: a no-frills avenue for didactical experts. In *The New Review of Hypermedia and Multimedia*, Vol. 10, No. 2, pages 165-180, Taylor & Francis, December 2004.

[89] V. Mirabella, S. Kimani and T. Catarci. A no-frills approach for accessible Web-based learning material. In *Proceedings of the 2004 international cross-disciplinary workshop on Web accessibility (W4A)*, pages 19-27, ACM Press, May 2004.

[90] S. Mirri, D. Pantieri, M. Roccetti and P. Salomoni. A Multimedia Broker for Accessible Learning Objects Transcoding. In *Proceedings of the 2nd IEEE International Conference on Information and Communication Technologies: from Theory to Applications (ICTTA '06)*, pages 641-646, IEEE Communications Society, April 2006.

[91] R. Mohan, J. Smith J and C.S. Li. Adapting Multimedia Internet Content for Universal Access. In *IEEE Transactions*

on Multimedia, pages 104-114, IEEE Communications Society, March 1999.

[92] MPEG MDS Group. MPEG-21 Multimedia Framework, Part 7: Digital Item Adaptation. ISO/MPEG N5845. Available from: http://www.chiariglione.org/mpeg/working_documents/mpeg-21/dia/dia_fcd.zip, 2003.

[93] MPEG Requirements Group. MPEG-21 Overview. ISO/MPEG N4991, 2002.

[94] D. Mukherjee, E. Delfosse, K. Jae-Gon and W. Yong. Optimal adaptation decision-taking for terminal and network quality-of-service. In *IEEE Transactions on Multimedia*, Vol. 7, No. 3, pages 454-462, IEEE Communications Society, June 2005.

[95] K. Nagao, Y. Shirai and K. Squire. Semantic Annotation and Transcoding: Making Web Content More Accessible. In *IEEE Multimedia*, Vol. 8, No. 2, pages 69-81, IEEE Computer Society, April 2001.

[96] National Center for Accessible Media. Available from: http://ncam.wgbh.org/index.html, 2006.

[97] L. Nevile, M. Rothberg, M. Cooper, A. Heath and J. Treviranus. Learner-centered Accessibility for Interoperable Web-based Educational Systems. In *Proceedings of*

Interoperability of Web-Based Educational Systems Workshop, *14th International World Wide Web Conference (WWW2005),* May 2005.

[98] H. Nielsen, P. Leach and S. Lawrence. An HTTP Extension Framework. RFC 2774, Agranat Systems, 2000.

[99] M. Nilsson. UAProf An overview. Available from: http://www.w3.org/2002/02/DIWS/presentations/nilsson/nilsson.pdf, 2002.

[100] B. D. Noble, M. Price and N. Satyanarayanan. A Programming Interface for Application-Aware Adaptation in Mobile Computing. In *Proceedings of the 2nd USENIX Symposium on Mobile and Location-Independent Computing*, April 1995.

[101] W.L. Jr. Oellermanna. Architecting Web Services. Springer-Verlag New York, 175 Fifth Avenue, New York, NY, 2001.

[102] Open Mobile Alliance (OMA). User Agent Profile v. 1.1 Approved Enabler. Available from: http://www.openmobilealliance.org/release_program/uap_v11.html, 2002.

[103] V. Pandey, D. Ghosal, and B Mukherjee. Exploiting user profiles to support differentiated services in next-generation wireless networks, In *IEEE Network*, Vol. 18, No. 5, pages

40-48, IEEE Communications Society, September-October 2004.

[104] D. Pantieri, P. Salomoni and S. Mirri. RMob: Transcoding Rich Multimedia Contents through Web Services. In *Proceedings of the 2nd IEEE International Workshop on Networking Issues in Multimedia Entertainment (NIME '06), CCNC 2006 Satellite Workshop*, pages 1168- 1172, IEEE Communications Society, January 2006.

[105] Parasoft. SOAtest. Available from: http://www.parasoft.com/jsp/products/home.jsp?product=SOAP, 2006.

[106] T. Phan, G. Zorpas and R. Bagrodia. An Extensible and Scalable Content Adaptation Pipeline Architecture to Support Heterogeneous Clients. In *Proceedings of the 22nd IEEE International Conference on Distributed Computing Systems (ICDCS '02)*, pages 507- 516, IEEE Communications Society, 2002.

[107] M. Pilgrim. Dive Into Accessibility. Available from: http://diveintoaccessibility.org/, 2002.

[108] B. Prabhakaran. Adaptive Multimedia Presentation Strategies. In *Multimedia Tools and Applications*, Vol. 12, No. 2-3, pages 281-298, Kluwer, November 2000.

[109] Practical Experimentation by Accessible Remote Learning Project (PEARL). Available from: http://iet.open.ac.uk/pearl, 2003.

[110] M. J. Rosenberg. E-Learning: Strategies for Delivering Knowledge in the Digital Age. McGraw-Hill, 2000.

[111] P. Salomoni and S. Mirri. A multimedia broker for ubiquitous and accessible rich media content transcoding. In *Proceeding of 1st IEEE International Workshop on Networking Issues in Multimedia Entertainment (NIME '04), Globecom 2004 Satellite Workshop*, pages 186- 191, IEEE Communications Society, November 2004.

[112] P. Salomoni and S. Mirri. Providing Accessible and Portable Video-Lecture from Content transcoding. In *Proceedings of 11th Euromedia Conference*, pages 97-101, February 2005.

[113] P. Salomoni, S. Mirri and G. R. Gay. Accessibility. To appear in *Encyclopaedia of Internet Technologies And Applications*, 2007.

[114] P. Salomoni, S. Mirri and G. R. Gay. E-learning. To appear in *Encyclopaedia of Internet Technologies And Applications*, 2007.

[115] A. Savidis, D. Grammenos and C. Stephanidis. Developing inclusive e-learning systems. In *Universal Access in the*

Information Society, Vol. 5, No. 1, pages 51-72, Springer-Verlag Berlin, Heidelberg, July 2006.

[116] A. Savidis and C. Stephanidis. Developing inclusive e-learning and e-entertainment to effectively accommodate learning difficulties. In *ACM SIGACCESS Accessibility and Computing*, No. 83, pages 42-54, ACM Press, September 2005.

[117] J. Seale. The development of accessibility practices in e-learning: an exploration of communities of practice. In *ALT-J Research in Learning Technology*, Vol. 12, No. 1, pages 51-63, Routledge, part of the Taylor & Francis Group, March 2004.

[118] J. Seale. The development of accessibility practices in e-learning: new roles for learners and teachers? In P. Kommers & G. Richards (Eds.), *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications*, page 4856, June 2005.

[119] C. Severance. What is a profile? [computer standards]. In *Computer*, Vol. 28, No. 9, page 66, September 1995.

[120] M. Sharples, D. Corlett and O. Westmancott. The Design and Implementation of a Mobile Learning Resource. In *Personal and Ubiquitous Computing*, Vol. 6, No. 3, pages 220-234, Springer-Verlag, May 2002.

[121] A. Singh, A. Trivedi, K. Ramamritham and P. Shenoy. PTC: Proxies that Transcode and Cache in Heterogeneous Web Client Environments. In *World Wide Web*, Vol. 7, No. 1, pages 7-28, Kluwer, March 2004.

[122] D. Sloan. Creating accessible e-learning content. In: *(eds Seale J., Phipps L. and Sutherland A.) Access All Areas: disability, technology and learning.* JISC, TechDis Service and ALT, pages 35-41, May 2002.

[123] D. Sloan, P. Gregor, M. Rowan and P. Booth. Accessible accessibility. In the *Proceedings on the 2000 conference on Universal Usability*, pages 96-101, ACM Press, November 2000.

[124] D. Sloan, A. Heath, F. Hamilton, B. Kelly, H. Petrie and L. Phipps. Contextual web accessibility - maximizing the benefit of accessibility guidelines. In *Proceedings of the 2006 international cross-disciplinary workshop on Web accessibility (W4A): Building the mobile web: rediscovering accessibility?*, pages: 121-131, ACM Press, May 2006.

[125] D. Sloan, J. Stratford and P. Gregor. Using multimedia to enhance the accessibility of the learning environment for disabled students: reflections form the Skills for Access project. In *ALT-J Research in Learning Technology*, Vol. 14,

No. 1, pages 39-54, Routledge, part of the Taylor & Francis Group, March 2006.

[126] D. Sloan, J. Stratford and S. Stone. Creating engaging, accessible multimedia for learning. In *(Eds. M. Adams and S. Brown) Towards Inclusive Learning in Higher Education*. London: Routledge, pages 129-143, 2006.

[127] M. Sloman. The E-Learning Revolution: How Technology is Driving a New Training Paradigm. American Management Association, 2002.

[128] J. R. Smith, R. Mohan, C. S. Li. Transcoding Internet Content for Heterogeneous Client Devices. In *Proceedings of IEEE International Conference On Circuits and Systems (ISCAS '98)*, pages 599-602, IEEE Communications Society, June 1998.

[129] C. Stephanidis. Adaptive Techniques for Universal Access. In *User Modeling and User-Adapted Interaction*, Vol. 11, No. 1-2, pages 159-179, Kluwer, March 2001.

[130] E. Stergarsek. Accessible e-learning platform. In *Proceedings of 1st IEEE-EESTEC Technical Conference*, IEEE Communications Society, September 2004.

[131] The Inclusive Learning Exchange (TILE). Available from: http://www.barrierfree.ca/tile/, 2006.

[132] Treasury Board of Canada Secretariat, Chief Information Officer. Common look and feel for the Internet – Accessibility Section. Available from: http://www.tbs-sct.gc.ca/clf-nsi/index_e.asp, 2000.

[133] J. Treviranus and J. Brewer. Developing and Reusing Accessible Content and Applications. In *Chapter 10 of: Reusing Online Resources: A Sustainable Approach to eLearning*, Allison Littlejohn, Kogan Page, London, April 2003.

[134] U.K. Parliament. Disability Discrimination Act. Available from: http://www.opsi.gov.uk/acts/acts1995/1995050.htm, 1995.

[135] University of California, Berkeley. UC Berkeley offers courses and symposia through Google Video. Available from: http://www.berkeley.edu/news/media/releases/2006/09/26_google.shtml, 2006.

[136] U.S. Rehabilitation Act Amendments. Section 508. Available from: http://www.webaim.org/standards/508/checklist, 1998.

[137] A. Vetro. MPEG-21 digital item adaptation: enabling universal multimedia access. In *IEEE Multimedia*, Vol. 11, No. 1, pages 84-87, IEEE Communications Society, January-March 2004.

[138] A. Vetro, C. Christopoulos and S. Huifang. Video transcoding architectures and techniques: an overview. In *IEEE Signal Processing Magazine*, Vol. 20, No 2, pages 18-29, March 2003.

[139] M. Wagner and W. Kellerer. Web Services Selection for Distributed Composition of Multimedia Content. In *Proceedings of the 12th annual ACM International Conference on Multimedia 2004*, pages 104 - 107, ACM Press, October 2004.

[140] WAP Forum. Wireless Markup Language version 2 Specification. Technical Specification, available from: http://www.wapforum.org/what/technical.htm, 2001.

[141] Web-4-All Project. Available from: http://web4all.atrc.utoronto.ca/, 2006.

[142] Web Accessibility Initiative. Introduction to Web Accessibility. Available from: http://www.w3.org/WAI/intro/accessibility.php, 2005.

[143] Web Accessibility Initiative. Social Factors in Developing a Web Accessibility Business Case for Your Organization. http://www.w3.org/WAI/bcase/soc, 2005.

[144] WebServiceX Translation Engine: http://www.webservicex.net/, 2006.

[145] M. Weiser. The Computer for the 21st Century. In *Scientific American*, Vol.265, No.3, pages 94-104, September 1991.

[146] World Wide Web Consortium. Available from: http://www.w3.org, 2006.

[147] World Wide Web Consortium. Accessibility Features of SMIL. Available from: http://www.w3.org/TR/SMILaccess/, 1999

[148] World Wide Web Consortium. Authoring Tool Accessibility Guidelines 1.0, W3C Recommendation 3 February 2000. Available from: http://www.w3.org/TR/WAI-AUTOOLS/, 2000.

[149] World Wide Web Consortium. Cascading Style Sheets, level 2 CSS2 Specification. Available from: http://www.w3.org/TR/REC -CSS2/, 1998.

[150] World Wide Web Consortium. Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0. Available from: http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115, 2004.

[151] World Wide Web Consortium, Extensible Markup Language (XML) 1.0. Available from: http://www.w3.org/TR/REC-xml, 2000.

[152] World Wide Web Consortium, Extensible Stylesheet Language (XSL) Version 1.0. Available from: http://www.w3.org/TR/xsl/, 2001.

[153] World Wide Web Consortium. HTML 4.01 Specification. Available from: http://www.w3.org/TR/html, 1999.

[154] World Wide Web Consortium. Implementation of OPS Over HTTP. Available from: http://www.w3.org/TR/NOTE-OPS-OverHTTP, 1997.

[155] World Wide Web Consortium. Mobile Web Best Practices 1.0, Basic Guidelines, W3C Working Draft 18 May 2006. Available from: http://www.w3.org/TR/mobile-bp/, 2006.

[156] World Wide Web Consortium. Resource Description Framework (RDF): Concepts and Abstract Syntax. Available from: http://www.w3.org/TR/rdf-concepts/, 2004.

[157] World Wide Web Consortium. Resource Description Framework (RDF) Model and Syntax Specification. Available from: http://www.w3.org/TR/1999/REC-rdf-syntax-19990222, 1999.

[158] World Wide Web Consortium. Scalable Vector Graphics (SVG) 1.1 Specification. Available from: http://www.w3.org/TR/SVG11, 2003.

[159] World Wide Web Consortium. Synchronized Multimedia Integration Language 2.1. Available from: http://www.w3.org/TR/2005/REC-SMIL2-20051213/, 2005.

[160] World Wide Web Consortium. User Agent Accessibility Guidelines 1.0, W3C Recommendation 17 December 2002. Available from: http://www.w3.org/TR/WAI-USERANGENT/, 2002.

[161] World Wide Web Consortium. Web Content Accessibility Guidelines 1.0. Available from: http://www.w3.org/TR/WCAG10/, 1999.

[162] World Wide Web Consortium. Web Content Accessibility Guidelines 2.0 W3C Working Draft 2005. Available from: http://www.w3.org/TR/WCAG20/, 2006.

[163] World Wide Web Consortium. XHTML™ 1.0 The Extensible HyperText Markup Language Specification. Available from: http://www.w3.org/TR/xhtml1/, 2002.

[164] World Wide Web Consortium. XSL Transformations (XSLT) Version 1.0. Available from: http://www.w3.org/TR/xslt, 1999.

[165] World Wide Web Consortium - Device Independence Activity. An Introduction to Device Independence. Available from: http://www.w3.org/2001/di/IntroToDI.html, 2001.

[166] World Wide Web Consortium - Education and Outreach Working Group. How People with Disabilities Use the Web W3C Working Draft. Available from: http://www.w3.org/WAI/EO/Drafts/PWD-Use-Web, 2001.

[167] WURFL. Wireless Universal Resource File Library. Available from: http://wurfl.sourceforge.net, 2006.

[168] H. Yonghyun, K. Jihong and S. Eunkyong. Structure-aware Web transcoding for mobile devices. In *IEEE Internet Computing*, Vol. 7, No. 5, pages 14-21, IEEE Communications Society, September-October 2003.

[169] C. Yoshikawa, B. Chun, P. Eastam, A. Vahdat, T. Anderson and D. Culler. Using Smart Clients to Build Scalable Services. In *Proceedings of Winter 1997 USENIX Technical Conference*, January 1997.

[170] D. Zhang, J. L. Zhao, L. Zhou and J. F. Nunamaker Jr. Can e-learning replace classroom learning? *In Communications of the ACM*, Vol. 47, No. 5, pages 75-79, ACM Press, May 2004.