Adaptation Technologies in Mobile Learning

Paola Salomoni, Silvia Mirri
Computer Science Department, University of Bologna, Italy.

ABSTRACT
The explosive growth of Internet services and the widespread diffusion of mobile devices are offering ubiquitous and time-independent access to a huge amount of online resources. In particular, educational experiences on non-conventional contexts have been made possible and the use of mobile terminals has created a new form of e-learning, called mobile learning (or “m-learning”). The availability of open technologies has participated in the diffusion of such a trend. Needless to say that standards, metadata and adaptation mechanisms have to be applied to the m-learning environment and content, in order to meet learner’s needs and preferences, as well as device characteristics.

This chapter will introduce issues and standards for adaptation mechanisms for mobile learning resources and it will describe some notable existing projects and software tools. Finally, it will present some case studies and open trends about m-learning.

INTRODUCTION
The term m-learning is meant to refer to learning by using mobile devices, both in anywhere/anytime context and in traditional classrooms (Clothier, 2005). During the last decade, mobile learning has grown from a research topic to a plethora of concrete projects in schools, museums, cities and workplaces. This transformation has been facilitated by the widespread diffusion of mobile devices and connectivity. Moreover, the availability of open source technologies in this field strongly supports the diffusion of mobile learning. More recently, Web 2.0 technologies and applications have increased users’ capabilities to collaborate and to share content and knowledge through mobile devices, constructing kinds of the so-called “collective intelligence” (Levy, 1997).

The diversity of mobile device capabilities, learner’s preferences (and needs), supported content formats and contexts of use has called for standards, metadata and adaptation mechanisms. They have to be applied to the learning environment and content, in order to provide the best educational experience to learners. In particular:

- device capabilities and learner’s needs have to be profiled on the basis of standards; open standards are strategically used in such a context;
- original learning contents have to be described on the basis of metadata and standards;
- learning contents have to be transcoded through adaptation mechanisms on the basis of device capabilities, learner’s preferences, contexts of use and original learning content characteristics.

This chapter will motivate the need of standards and adaptation mechanisms in the mobile learning field and it will describe related standards, open source projects and libraries. Finally, it will present how Learning Content Management Systems face m-learning instances and it will introduce m-learning open issues and trends.

The remainder of the chapter is organized as follows. The section named “Mobile device profiling standards” presents main profiling mobile device standards (W3C CC/PP and OMA UAProf) which describe mobile device capabilities and drive learning content transcoding operations. This section introduces also open source repositories of mobile device descriptions (such as WURFL) compliant to the W3C Device Description Repository Standard (DDR). The section titled “Mobile learning content adaptation mechanisms” describes main concepts related to content adaptation and transcoding, by introducing architectural approaches, standards and formats which apply adaptation mechanisms and open source projects and libraries in charge of operating adaptations to the whole learning content or to single media which compose it. The section named “Mobile issues in LCMSs” introduces main strategies to face mobile context issues applied by the two of the most widespread and well-known open source Learning Content Management Systems: Moodle and ATutor. Finally, the section titled “Open Issues and Future Research Directions” concludes the chapter, by describing some open issues and future trends. In particular, it presents the growing need of tools and applications in order to let learners collaborate to each other, create and share mobile learning content. Moreover, the relationships between mobile learning and accessibility topics are introduced in this last section.

MOBILE DEVICE PROFILING STANDARDS

In order to meet the needs of learners who are equipped with mobile devices, the adaptation of didactical materials has to be driven by device characteristics. Hence, profiling such characteristics is fundamental in mobile learning. In fact, standards which describe mobile device capabilities allow the identification and the definition of the characteristics (i.e. formats, sizes, etc.) the learning contents should have after the transcoding operations. There are currently two main standards which are devoted to perform device profiling: W3C CC/PP (Composite Capabilities/Preferences Profile) and OMA UAProf (User Agent Profile). Both of them are based on Resource Description Framework (RDF). The RDF format implies the document schemas are extensible.

As the diversity of devices increases, the device capability and preference for content negotiation and adaptation must be known. The goal of CC/PP (World Wide Web Consortium, 2004) and UAProf (Open Mobile Alliance, 2007) profiles is letting 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. Whenever these profiles are parsed, RDF represents an abstraction level over XML, so it must validate both XML and RDF. 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.

W3C Composite Capabilities/Preference Profile (CC/PP)

The Composite Capabilities/Preference Profile (CC/PP) provides a standard way for devices to transmit their profiles when requesting Web content (World Wide Web Consortium, 2004). Servers and proxies can then provide adapted content which is appropriate to a particular device. A CC/PP vocabulary is defined by using RDF 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, the software platform and the 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 image 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 need to be solved so that the protocol can be used in practice.

OMA User Agent Profile (UAProf)

The second profiling standard we are going to introduce is UAProf, which has been defined as living between Wireless Application Protocol (WAP) devices and servers (Open Mobile Alliance, 2007). The UAProf was born to be used for better content adaptation on different types of WAP devices. The UAProfile also describes the next generation of WAP phones. The advantage of UAProf is that it defines different categories of mobile device capabilities:
- *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, etc., 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 the CC/PP one. Moreover, the UAProf production (and the CC/PP too) for a device is voluntary: for GSM devices, the UAProf is normally produced by the vendor of the device whereas for CDMA/BREW devices, it's more common for the UAProf to be produced by the telecommunications company. Some drawbacks are listed in the following:

1. Not all devices have UAProfs.
2. Not all advertised UAProfs are available.
3. Real-time retrieving and parsing UAProfs is slow and can add substantial overhead to any Web request.
4. There is no industry-wide data quality standard for the data within each field in an UAProf.

All these issues and the huge amount of mobile device capabilities have made necessary the use of a repository of mobile device descriptions. This provides a quick and easy way to match learner's device types and their characteristics.

Device Description Repositories

The W3C MWI (Mobile Web Initiative) and the DDWG (Device Description Working Group), recognizing the difficulty in collecting and keeping track of UAProf, CC/PP and device profile information, and the practical shortcomings in the implementation of UAProf and CC/PP across the industry, have outlined specifications for a Device Description Repository (DDR). DDR is supported by an initial core vocabulary of device properties (World Wide Web Consortium, 2008a). Implementations of the proposed repository are expected to contain information about Web-enabled devices (specially, but not limited to, mobile devices). By means of DDR compliant repositories, authors of Web content would be able to make use of repositories to adapt their content to best suit the requesting device. This would facilitate the interaction and viewing of Web content and applications across devices with widely varying capabilities. Open and commercial implementations of the DDR Simple API are available (World Wide Web Consortium, 2008b). In particular, the open source "MyMobileWeb" (MORFEO Project, 2007) was presented by Telefónica I+D, as we will describe in the following section (in subsection "Open source projects and libraries"). This implementation includes a Web-interface to the API implementation, which was developed in Java language.

An example of a device description repository is the Wireless Universal Resource File Library (WURFL), based on UAProf and CC/PP profiles (WURFL, 2010). The WURFL is an open source project that focuses on the problem of presenting content on a 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, thereby reflecting new wireless devices coming on the market. Currently, WURFL stores more than 500 capabilities related to thousands of mobile device models.

It is worth mentioning that whenever an m-learning project takes into account mobile device profiling, it should provide an integration with the learners' profiling system. In fact, learners' profiles have to be compliant to IMS Standards, such as the IMS Learner Information Profile (IMS Global Learning Consortium, 2002a) and the IMS Accessibility Learner Profile (IMS Global Learning Consortium, 2002b). The integration of such profiles guarantees a complete description of the learner's context, but it could cause some overlaps that the profiling system should manage. A solution has been proposed in (Ferretti et al., 2009).

MOBILE LEARNING CONTENT ADAPTATION MECHANISMS

The growing diffusion of devices coupled with the ability to deliver Web content and specifically online learning content anywhere at any time, has improved the learning process flexibility and quality of services (Anido, 2006). As a result, new techniques for delivering didactical materials according to device features and even specific languages emerged (Pandey et al., 2004). Adapting typical Web content and services (originally made for PCs) to small and mobile devices is still one of the content adaptation hot topics (Curran et al., 2005). 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.

The main obstacles to Web content interoperability are: possible application bugs; some devices don't support functions, such as new mobile phones that only support Java and non-standard proprietary markup language extensions. The final result is that the same online learning content might have a great variety of appearances and could run in several ways (or in no way), depending on the platform and device (Salomoni et al., 2008). Hence content adaptation and transcoding are necessary and should be based on device capabilities and preferences, on the network characteristics and on the strength of some application-specific parameters; therefore, Web learning content and applications should be generated or adapted for a better learner experience (Harumoto et al., 2005). Device independence principles set aside from any specific markup language, authoring style or adaptation process.

According to the W3C definition, content adaptation is the transformation and the manipulation of Web content (such as images, audio, videos, texts and multimedia presentations) to meet desired targets (defined by the terminal capabilities and the application needs) (Colajanni et al., 2004; Harumoto et al., 2005). They include:

- media conversion (the conversion of content from its original form to another; it can be performed automatically, depending on the type of conversion: e.g., Text to Speech (TTS) or animation to image);
- format transcoding (e.g. XML to HTML, SVG to GIF, WAV to MP3);
- scaling (of images as well as video and audio streams; it can involve recoding and/or compressing specific media content and it has effects in terms of reduction of size, quality and data rate of contents);
- 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);
- re-sampling;

- file size compression;
- textual content fragmentation.

In particular, transcoding is the process of converting a media file or object from one format to another. This process is typically used to convert video, audio and image formats, but it is also used to adapt multimedia presentations and Web content to the constraints of non-standard devices, e.g. mobile devices (Laakko, 2005). It is well-known that mobile devices have limited capabilities, such as smaller screen sizes, lower memory and slower bandwidth rates. Most existing multimedia presentations and Web content are created to be displayed on desktop computers and, usually, Web designers provide complex, detail-rich didactical content, with multimedia experiences. Thus, in mobile learning environments, transcoding must face the diversity of mobile devices. This heterogeneity imposes an intermediate state for content adaptation to ensure a proper presentation on each target device (Pandey et al., 2004; Curran et al., 2005).

In the following of this section we are going to introduce architectural approaches, standards, open source projects and libraries, which can be useful in mobile learning content adaptation and transcoding.

Architectural Approaches

Due to different device capabilities, content adaptation and transcoding need to be implemented before the content is presented to the user. From an architectural point of view, four categories should be mentioned that represent the most significant distributed solutions for content adaptation (Colajanni et al., 2004):

- *Client-side approaches*: the transcoding process is in charge of the client application. Client-side solutions can be classified into two main categories with different behaviours: (i) the clients receive multiple formats and adapt them by selecting the most appropriate one to play-out, or (ii) 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.
- *Server-side approaches*: the server (which provides contents) performs the additional functions of content adaptation. In such an approach, content adaptation can be carried out in an off-line (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) or on-the-fly fashion (adapted contents are dynamically produced before delivering them to the clients).
- *Proxy-based approaches*: In proxy-based approaches, the adaptation process is carried out by a node (i.e. the proxy) placed between the server and the client. In essence, the proxy captures replies by the server to the clients request and performs three main actions: (i) it decides whether performance enhancements are needed, (ii) it performs content adaptations and (iii) it sends the adapted contents to the client. 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.
- *Service-oriented approaches*: 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. 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 types of adaptations; therefore, third-party adaptation services are important.

Standards

The diversity of the content presentation environments imposes strict requirements on multimedia applications and systems (Jannach et al., 2006). The emerging growth of mobile services (together with wireless technology such as Bluetooth, 802.11, GPRS and UMTS) defines more requirements for the content and service providers (Pandey et al., 2004). 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. In particular, as these mobile devices are becoming more multimedia capable, one of the challenges is the multimedia content delivery on these embedded devices.

Hence, in this subsection we are going to introduce adaptation mechanisms applied by W3C Content Selection for Device Independence (World Wide Web, 2007), W3C SMIL markup language (World Wide Web Consortium, 2008c) and by MPEG-21 framework (MPEG Requirements Group, 2002).

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) and its applications like SMIL (Synchronized Multimedia Integration Language) developed by the World Wide Web Consortium (W3C) can be applied in modeling structured, document-like multimedia presentations.

W3C Content Selection for Device Independence

A more recent Candidate Recommendation for standardizing content adaptation was released in 2007 (World Wide Web Consortium, 2007). The Content Selection for Device Independence (DISelect) aims to provide the ability to select between different versions of content. DISelect has been designed to be used within other markup languages and it was born to support content authors in the specification of different versions of the content they produce. The ability to select among different versions of the same content provides one important mechanism the authors can use in order to guarantee the adaptability of their materials, so that they could be enjoyed also by using mobile devices. DISelect is provided with a mechanism for the content selection which is to be expressed when adaptation takes place and which requires only modest computational capability. It defines two profiles of markup: the DISelect Basic and the DISelect Full. The former is a subset of the latter, and it is intended to contain only the necessary markup to be implemented on small-footprint DISelect implementations, running on mobile devices with limited processing power and memory. The latter is the complete set of defined DISelect markup. The main idea is to allow the use of DISelect elements and attributes within a host markup language by means of namespace mechanisms. In this module, attributes and elements for conditional processing have been defined, such as `<if>` (it defines a set of materials which has to be included in the final result if the associated expression has the appropriate value) and `<select>` (it encloses one or more sets of material that are subject to conditional selection. It contains one or more `<when>` elements and an optional `<otherwise>` element. Expressions associated with the `<select>` and `<when>` elements control the conditions under which particular parts of the content are processed). Despite such a markup language and its related adaptation mechanism seem to be very feasible and easy to be used (also in creating suitable m-learning content), it is still a Candidate Recommendation and not yet a standard.

W3C SMIL

A stable and well-known W3C standard which applies client-side adaptation mechanisms is W3C SMIL (World Wide Web Consortium, 2008c). It 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 clip starts talking about it. SMIL 3.0 is the main representation of 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 mobile device.
SMIL 3.0 is defined as a set of markup modules, which define the semantics and XML syntax for certain areas of SMIL functionalities. This specification provides different classes of changes to SMIL 2.1, covering 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 Modules, which are, in turn, revised allowing extended functionalities. Some of these changes are related to the use of SMIL through mobile devices and to the content transcoding mechanisms.

Several simple content selection mechanisms have been introduced in SMIL to provide greater flexibility and to meet different context of use constraints (including those ones which are related to mobile contexts). However, in most cases, SMIL adaptation is achieved at the client side. This implies that the client is adaptation-capable and that the profiles and the device 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 to fit bandwidth and display limitations. In fact, SMIL language contains an “adaptation” or “alternate content” mechanism. By using the `<switch>` tag and the so-called “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. Besides the capabilities of specifying bandwidth, display limitations and learners’ language for each medium, SMIL allows the declaration of author-defined variables to trigger the presentation of contents. The latter ones are a construction of the CustomTestAttributes (CTA) module. Language built-in and author-defined variables could be used by SMIL compliant players to present alternative multimedia items on the strength of their Boolean value. To enhance content adaptability, the SMIL CTA module extends this feature with the definition of author-defined custom test attributes. This module allows the attribute definition and setting in the header of the SMIL document, so that the author can set a default state for each custom test attribute and declare a default attribute which will be play out whenever no customization will be necessary. The customization is done by using a SMIL `<switch>` construct in the `<body>` of the SMIL document. It is used to select media for inclusion in a presentation depending on the values of the custom test attributes. The first object that contains a value true will be rendered. It is possible to set the last option so that it will always resolve true. In this way it will be considered only if no other objects resolve to true. Hence, the CTA module could be used to offer alternative presentations to users who are equipped with mobile devices.

Another interesting SMIL 3.0 feature is the SMIL State Module. It provides mechanisms which permit the author to create more complex control media than the timing and content control modules could do. By allowing the use of variables, a document can have some explicit state along with ways to modify, use and save this state. So, such a module could be used to describe and/or define when different versions of the content should be played out, allowing also content adaptation to mobile learning contexts. It is worth noting that an improvement is the use of XPath expressions instead of boolean ones, which are used in the CTA module.

Finally, the improved SMIL 3.0 MetaInformation Module and the attribute `label` allow the definition of alternative presentation and metadata within the document instead of in the `<head>` element of the presentation. In particular, the `label` attribute specifies the name of a SMIL file, which contains a description or an alternative version of the element. If it is selected, then the original presentation will be paused and a new document instance will be created to display the target SMIL file. The use of this attribute affects the original media synchronization and the provision of new versions of the multimedia presentation. Thus, by means of such an attribute, customized content could be provided to users equipped with mobile devices. Moreover, the new SMIL 3.0 Metainformation module permits the declaration of the element `<metadata>` as a child of media elements inside the `<body>` and not only in the header of the SMIL document.

These adaptation features enable a SMIL player to fit technical circumstances and some fairly static user preferences. This makes SMIL a standard able to provide adapted multimedia presentations to mobile learners.

MPEG-21

Another interesting standard is MPEG-21, which is an open standards-based framework for multimedia delivery and consumption by all the players (MPEG Requirements Group, 2002). 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 involves guaranteed communication between two or more parties. Interoperability expresses the users' need of easily exchanging any type of information without technical barriers, supporting it also in mobile contexts (Burnett et al., 2003).

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. The users can be creators, consumers, rights holders, content providers or distributors, learners, etc. There is no technical distinction between providers and consumers: all parties that have to 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.

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-lecture collection or a music album. The Digital Item is thus the fundamental unit of distribution and transaction within the MPEG-21 framework.

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.

The seventh part of MPEG-21 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 (by considering also mobile devices and contexts), management, and implementation issues. Achieving this goal requires the adaptation of Digital Items (MPEG MDS Group, 2003). 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, according to the learners' context of use.

The usage environment description tools can describe the mobile device capabilities (in terms of codec, 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 the natural environment. In this context, the 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 (Vetro, 2004).

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

- Resource adaptability: i.e. 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.

- Session mobility: i.e. 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.

Concluding, MPEG-21 is an ideal candidate to provide suitable and adaptable content in mobile learning environments.

Open source projects and libraries

Different open source projects and libraries can operate adaptation to online learning content and to single media which compose it. In this subsection we are going to introduce some of them, in particular FFmpeg (FFMPEG, 2010), ImageMagick (ImageMagick, 2010) and GAIA Image Transcoder (Gaia Reply 2010). Finally, we will introduce an open source project that allows the development of mobile applications (MORFEO Project, 2007).

FFmpeg (FFMPEG, 2010) is an open source project which produces libraries and programs for managing multimedia data. The name of the project comes from the MPEG video standards group, together with "FF" for "fast forward". Such a project is composed by different parts, i.e. an audio/video codec library, an audio/video container mux and demux library and the FFmpeg command line program for transcoding multimedia files. FFmpeg is developed under GNU/Linux, but it can be compiled under most operating systems, including Apple Inc. Mac OS X and Microsoft Windows. There are two video codecs (FFV1 and Snow codec) and one video container invented in the FFmpeg project during its development. In particular, such a set of libraries allows different forms of audio and video transcoding, for instance: transforming video and audio files from a format to another one, adding subtitle or caption tracks to a video, re-sizing video width and height, disabling audio recording, degrading audio quality.

ImageMagick (ImageMagick, 2010) is an open source software suite for displaying, converting, and editing raster image files. The software mainly consists of a number of command-line interface utilities for manipulating images. ImageMagick does not have a GUI based interface to edit images, as Adobe Photoshop and GIMP have, but instead it modifies existing images as directed by various command-line parameters. Many applications, such as MediaWiki, and phpBB can use ImageMagick to create image thumbnails if it is installed. ImageMagick is also used by other programs for converting images. In fact, one of the basic and thoroughly-implemented features of ImageMagick is its ability to efficiently convert images between different file formats. In particular, ImageMagick can read and write over 100 image file formats. Another ImageMagick trancoding feature is the color quantization: the number of colors in an image can be reduced to an arbitrary number and this is done by intelligently weighing the most prominent color values present among the pixels of the image. Summarizing, such a library allows different kinds of image transcoding, such as: format conversion, width and height resizing, image rotation, transparency, animation, insertion of text and comments.

The GAIA Image Transcoder (GIT) is an open source library which operates image transcoding for mobile applications (Gaia Reply, 2010). GIT is composed by two parts: a transformation and transcoding library which performs image adaptation to mobile devices (by using information retrieved from the WURFL file) and a JSP tag library which enables library utilization into a J2EE environment. The transcoding library works as a transformation pipeline, which is composed by a set of filters. Such filters could operate directly on the image body or indirectly on the associated meta-information. At the time the authors are writing, GIT supports filters such as: re-sizing the images to conform it to the width and the height of the device screen, optimization of color depth of the image on the basis of device capabilities, transcoding of the images to supported formats, returning the associated image when a corresponding URI is given.

All the previous projects and libraries should be used by applications devoted to providing adapted content to mobile users. MyMobileWeb (MORFEO Project, 2007) is an open source project which allows the development of mobile applications with such features. It is based on Java and J2EE technology and on open-standards. By means of MyMobileWeb it is possible to create applications which adapt their user interfaces according to the characteristics of the device and Web browser used. Such device capabilities are provided by a Device Description Repository (compliant to the W3C DDR Standard), such as the already described WURFL (WURFL, 2010). Content should be described and structured in a declarative language (based on Web standards). The applications developed by using MyMobileWeb automatically perform content fragmentation, when necessary. The applications layout and all the appearance features are controlled through CSS stylesheets and it is possible to define different stylesheets for different families of devices. Moreover, MyMobileWeb provides the selection of user interface parts, on the basis of the characteristics of the delivery context, allowing authors to specify adaptation policies by means of the selection attribute, as mandated by the W3C DISelect specification (World Wide Web Consortium, 2007). Finally, in order to adapt different media content (audio, video, images), selection and transcoding components are incorporated into MyMobileWeb.

MOBILE ISSUES IN LCMSs

In this Section we are going to present how some well-known LCMSs face mobile learners' instances.
Several open source projects are based on Moodle, a well-known open source Learning Management System (Moodle Pty Ltd, 2010). The MOMO (MObile MOodle) project is an add-on to Moodle which is able to implement mobile learning scenarios with Moodle as a backend. Mobile learners may install the MOMO client on their mobile devices and they can access m-learning courses. The MOMO client is a Java based application. Moreover, administrators are allowed to install the MOMO extension on the Moodle server in order to make content available for mobile environments. The MOMO project was designed to allow m-learning experiences by means of generic Java and Internet enabled devices. After some years it has been abandoned and substituted by the MLE-Moodle (Mobile Learning Engine-Moodle) plug-in. This is especially devoted to users who access m-learning content through mobile phones. Learners can either use the mobile browser or a special mobile phone application which was designed for learning on mobile phones (the MLE phone client). When a learner installs this special application on his/her device, he/she has to declare the device producer and model. Then, a customized version of the application is downloaded and installed. In such a context there is no Device Description Repository. The MLE server side architecture includes also a media server, which is devoted to convert media (audio, video and images) to formats suitable for mobile phones.

Another widespread open source Learning Content Management System is ATutor (Adaptive Technology Resource Centre, 2010). ATutor has been designed and developed with accessibility and adaptability in mind. To reach such goals, ATutor supports a large number of accessibility, interoperability and e-learning standards (i.e. W3C WCAG, IMS AccessForAll, IMS ACCLIP, IMS Content Packaging, ADL SCORM, IMS QTI, IMS Common Cartridge) and its layout and interface are liquid and highly configurable by users. Thanks to the application of universal design principles, ATutor is easy to be enjoyed also by using non standard devices. Nevertheless, there is not yet a version explicitly dedicated to mobile learning environments. In order to overcome such a lack, the development team is designing a version of ATutor for mobile platforms (iPhone, Android, Blackberry). The main goal is to extend current Web services in ATutor to allow accessing learning content, network activity, communication tools, etc, by using a mobile device. Such a mobile application will be Java, Javascript and AJAX technologies based.

OPEN ISSUES AND FUTURE RESEARCH DIRECTIONS

This section will introduce some open issues and future trends in adapting technologies in mobile learning environments. A notable trend is related to the so called "E-learning 2.0". Such a term indicates the set of Computer-Supported Collaborative Learning (CSCL) systems and tools. At the moment this is one of the most promising innovations to improve learning by exploiting current information and communication technologies. The main idea is to provide tools students can used to work together on learning tasks or in other words to collaboratively learn. Collaborative learning assumes that learners' knowledge is socially constructed through conversations and comparisons about learning content and idea sharing (Levy, 1997). Social networks, Web 2.0 applications and new communications technologies should be applied to m-learning in order to enhance such a more active role of the learners. New features and tools should be adapted in order to be exploited in a feasible way by mobile devices with limited characteristics (computational capability, connectivity, small display, and so on). Collaboration among learners is an issues of the so-called constructivism, a theory of knowledge which argues that humans create their knowledge from their experiences. Constructivism is often associated with pedagogic approaches that promote active learning, or learning by doing. To apply such a theory, m-learning applications should provide mechanisms to allow learners' content creation, sharing and integration, according to mobile device capabilities.

Another interesting open issue is the accessibility of mobile learning environments. Their limited capabilities could affect the learning experience of people with disabilities (Barron et al., 2004). In fact, small screens, small buttons and keyboards, the lack of alternative input systems, the restriction of configurable display options (i.e. text and background colors, text size, font) may represent barriers for learners with disabilities.

CONCLUSION

Content and application adaptation is fundamental in mobile learning contexts. It can be applied by exploiting different strategies, mechanisms, architectural approaches, device profiling standards and libraries. In order to provide a customized mobile learning experience, all these features and technologies have to be orchestrated. This way, learning content and applications could meet mobile learners' needs and other constraints which affect the m-learning context. In this scenario, standards and open source resources play a strategic role, because they significantly improve the tools to reach m-learning goals. In particular, the development of open source LCMS, the definition of profiling devices open standards and repositories, and content transcoding open mechanisms and libraries are leading the field of adaptation in mobile learning environments. By means of such open technologies, m-learning designers and developers could effectively provide learners with adaptable and interoperable content and applications.

REFERENCES

Adaptive Technology Resource Centre (2010). ATutor Learning Content Management System. Retrieved March 21, 2010, from: http://www.atutor.ca/.

Advanced Distributed Learning (2004). Sharable Content Object Reference Model (SCORM) 2004 4nd Edition Document Suite. Retrieved March 21, 2010, from: http://www.adlnet.gov/Technologies/scorm/SCORMSDocuments/2004%204th%20Edition/Documentation.aspx.

Anido, L. (2006). An observatory for e-learning technology standards. *Advanced Technology for Learning*, 3(2), 99-108.

Barron, J. A., Fleetwood, L. & Barron, A. E. (2004). E-Learning for Everyone: Addressing Accessibility. *Journal of Interactive Instruction Development*, 26(4), 3-10.

Burnett, I., Van de Walle, R., Hill, K., Bormans, J. & Pereira, F. (2003). MPEG-21: Goals and Achievements. *IEEE Multimedia*, 10(6), 60-70.

Clothier, P. (2005). An Introduction to m-Learning: An Interview with Ellen Wagner. *The E Learning Guild*. Retrieved March 21, 2010, from: http://www.elearningguild.com/.

Colajanni, M. & Lancellotti, R. (2004). System Architectures for Web Content Adaptation Services. *IEEE Distributed Systems online*, 5(5).

Curran, K. & Annesley, S. (2005). Transcoding Media for Bandwidth Constrained Mobile Devices. *International Journal of Network Management*, 15(2), 75-88.

Ferretti, S., Roccetti, M., Salomoni, P. & Mirri, S. (2009). Custom E-learning Experiences: Working with Profiles for Multiple Content Sources Access and Adaptation. *Journal of Access Services*, 6(1&2), 174-192.

FFMPEG (2010). FFmpeg Multimedia Systems. Retrieved March 21, 2010, from: http://ffmpeg.org/.

Gaia Reply (2010). Gaia Image Trascoder (GIT). Retrieved March 21, 2010, from: http://gaia-git.sourceforge.net/.

Harumoto, K., Nakano, T., Fukumura, S., Shimojo, S. & Nishio, S. (2005). Effective Web Browsing through Content Delivery Adaptation. *ACM Transactions on Internet Technology (TOIT)*, 5(4), 571-600.

ImageMagick (2010). ImageMagick - Convert, Edit, and Compose Images. Retrieved March 2010, from: http://www.imagemagick.org/.

IMS Global Learning Consortium (2002a). IMS Learner Information Package Accessibility for LIP. Retrieved March 21, 2010, from: http://www.imsglobal.org/specificationdownload.cfm.

IMS Global Learning Consortium (2002b). IMS Learner Information Profile (LIP). Retrieved March 21, 2010, from: http://www.imsglobal.org/specificationdownload.cfm.

Jannach, D., Leopold, K., Timmerer, C. & Hellwagner, H. (2006). A knowledge-based framework for multimedia adaptation. *Applied Intelligence*, 24(2), 109-125.

Laakko, T. & Hiltunen, T. (2005). Adapting Web Content to Mobile User Agents. *IEEE Internet Computing*, 9(2), 46-53.

Levy, P. (1997). *Collective Intelligence: Mankind's Emerging World in Cyberspace*. New York, NY, USA: Plenum Publishing Corporation.

Moodle Pty Ltd (2010). Moodle. Retrieved March 21, 2010, from: http://moodle.org/.

MORFEO Project (2007). MyMobileWeb Project. Retrieved March 21, 2010, from: http://mymobileweb.morfeo-project.org/.

MPEG MDS Group (2003). MPEG-21 Multimedia Framework, Part 7: Digital Item Adaptation. ISO/MPEG N5845. Retrieved March 21, 2010, from: http://www.chiariglione.org/mpeg/working_documents/mpeg-21/dia/dia_fcd.zip.

MPEG Requirements Group (2002). *MPEG-21 Overview*. ISO/MPEG N4991.

Open Mobile Alliance (2007). User Agent Profile v. 2.0 Approved Enabler. Retrieved March 21, 2010, from: http://www.openmobilealliance.org/Technical/release_program/uap_v2_0.aspx.

Pandey, V., Ghosal, D. & Mukherjee, B. (2004). Exploiting User Profiles to Support Differentiated Services in Next-Generation Wireless Networks. *IEEE Network*, 18(5), 40-48.

Salomoni, P., Mirri, S., Ferretti, S. & Roccetti, M. (2008). A Multimedia Broker to support Accessible and Mobile Learning through Learning Objects Adaptation. *ACM Transactions on Internet Technology (TOIT)*, 8(2), 9-23.

Vetro, A. (2004). MPEG-21 digital item adaptation: enabling universal multimedia access. *IEEE Multimedia*, 11(1), 84-87.

World Wide Web Consortium (2004). Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0. Retrieved March 21, 2010: http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115.

World Wide Web Consortium (2007). Content Selection for Device Independence (DISelect) 1.0. Retrieved March 2010: http://www.w3.org/TR/cselection/.

World Wide Web Consortium (2008a). Device Description Repository Core Vocabulary. Retrieved March 21, 2010: http://www.w3.org/TR/ddr-core-vocabulary/.

World Wide Web Consortium (2008b). Device Description Repository Simple API. Retrieved March 21, 2010: http://www.w3.org/TR/DDR-Simple-API/.

World Wide Web Consortium (2008c). Synchronized Multimedia Integration Language 3.0. Retrieved March 21, 2010: http://www.w3.org/TR/2008/REC-SMIL3-20081201/.

WURFL (2010) Wireless Universal Resource File Library. Retrieved March 21, 2010, from: http://wurfl.sourceforge.