A Technological Framework for TV-supported Collaborative Learning*

Martín López-Nores, Andrés Elexpuru-Eguia, Yolanda Blanco-Fernández, José J. Pazos-Arias, Alberto Gil-Solla, Jorge García-Duque, Belén Barragáns-Martínez and Manuel Ramos-Cabrera

Department of Telematic Engineering, University of Vigo, 36310, Spain
{mlnores, elexpuru, yolanda, jose, agil, jgd, belen, mramos}@det.uvigo.es

Abstract

Interactive Digital TV is emerging as a potentially important medium to create opportunities for learning at home. To date, the offer has been mostly based on the contents available through broadcast, but this is expected to change in the near future. The increasing availability of high-quality bidirectional networks, together with the fact that IDTV users are abandoning their passive habits, envisages a new range of highly interactive services that may enhance greatly the prospects of distance education.

This paper introduces a technological framework for the development and deployment of distributed and collaborative educational services for IDTV, proposing an extension to the Multimedia Home Platform standard. The framework is based on a selection of freely available technologies, which we integrate into a CASE tool that bridges the gap between course-authoring and programming tasks. We also discuss the possible market implications of our approach, because the ideas presented here contribute to openness in the field of IDTV services, so far monopolised by mainstream broadcasters.

Key words: Interactive Digital TV, collaborative learning, distributed multimedia services, authoring tools.

1. Introduction

The development of Internet-based learning initiatives has revealed some shortages, including difficulties in the use of computers for some social sectors, the limited penetration of computers in homes and the uneven presence of broadband infrastructure. In response to that, two initiatives are being taken to port educational services to other mediums than personal computers [14]: t-learning, supported by Interactive Digital TV (IDTV), and m-learning, based on the use of modern mobile devices, like mobile phones or PDAs.

Convergence of these mediums is expected for the near future, so as to set the foundations for continuous and ubiquitous learning. However, this process is hampered by the fact that the users’ habits and expectancies are greatly different for each particular medium. As a result, the situation today suggests that different solutions should be crafted to adapt the characteristics of each medium, while keeping the goal of convergence in mind.

Our working group has been involved with IDTV architectures and services for several years, and we have recently turned our attention to the t-learning field. In [9], we presented an approach to developing educational applications, addressing the specifics of broadcast services and focusing on the entertainment aspect. According to pjb Associates [13], such services play a significant role at the initial stage of t-learning in which we are today. However, it is commonly agreed that the pedagogical value of t-learning will be greatly enhanced by supporting higher levels of interaction among users and service providers. The goal of this paper is to anticipate the needs of the future range of services, by introducing a technological framework for the development and deployment of distributed educational services, with special emphasis on collaborative ones. We do not comment pedagogical or organisational issues involved with the use of the proposed technology.

Our work is based on the Multimedia Home Platform (MHP) standard [4], which is the proposal of the DVB Consortium to normalise the field of IDTV services. Since it was published in year 2000, MHP has been steadily growing in popularity, and now it faces worldwide adoption, ensuring the interoperability of future developments. In [9], we analysed the support MHP provides for broadcast-based t-learning services, concluding that it is quite satisfactory. But matters change when considering the needs of distributed services, because the mechanisms at the disposal of application developers to handle the return channel are still rather simplistic, only adequate by themselves for limited feedback from single users. Thus, we propose here an extension to the standard that would improve its support for interaction and collaborative work.

* Work supported by the Ministerio de Educación y Ciencia, research project TSI2004-03677.
The paper is organised as follows. Section 2 comments past and future trends of t-learning, explaining its most important peculiarities. Section 3 details our approach for distributed t-learning services, describing how they are structured and the technologies we use for their implementation. Section 4 introduces a CASE tool intended to assist in the development of services within our approach. Section 5 comments the current status of our work and how we are putting it into practice. Finally, in Sect. 6, we discuss the relevance of our work regarding learning and business opportunities, and also analyse future lines of research.

2. T-learning experiences and promises

As a platform for education, Interactive Digital TV is considered a key to reach the widest audiences. This is because there is at least one television in nearly every household in developed countries, while the average penetration of personal computers is not expected to go beyond 70% in the short to medium term (see [14]). Moreover, being an enhancement to traditional TV sets, IDTV is easy to use and well known for everybody, meeting the socially important need to offer online learning services to people who cannot afford to buy a computer, do not have Internet access or lack the knowledge to use such technologies.

In the migration from analogue to digital TV, it was soon envisaged that t-learning could help fulfill the emptiness in the set of applications that TV users would be willing to pay for (see [14]). As a result, broadcasters started to provide simple, pioneering services as a complement to their channel offerings [1]. These services only provided means for simulated interactivity (interaction with the contents available in the broadcast streams), making little use of feedback mechanisms. The emphasis was placed on informal learning through edutainment (education and entertainment), coherent with the consideration of TV as a medium for entertainment and the traditionally passive habits of TV users.

The importance of broadcast services is stressed by the fact that their interactive features are driving users towards more active profiles in the use of the television, not being passive spectators anymore. This, together with the increasing availability of high-quality bidirectional networks, makes it possible to start thinking on highly interactive services, based on the profuse exploitation of return channels.

The increased interactivity (referred to as real interactivity) brings up opportunities for more engaged learning. It provides for a user to actively interact with others, promoting the creation of virtual learning communities [3] that gather together people with common interests and learning needs. This contributes to mitigate the feeling of isolation that is typical in distance education. Moreover, it sets the basis for the provision of new ways of collaborative learning, which, as argued in [13], is likely to be a more fruitful way of widening access to and participation in learning in the medium term than just focusing on the interactive TV offerings through broadcast.

2.1. The peculiarities of t-learning

The extensive research done in the e-learning field provides useful theoretical insight for t-learning [10], as well as a number of standards and tools for content management and student monitoring, such as the SCORM and IMS standards. Most of these solutions are also valid for t-learning, since there are no fundamental distinctions regarding evaluation, profiling, content management, etc. What is more, these tasks represent the major source of commonalities between e-learning and t-learning (especially distributed t-learning), because there is much work to be done in accommodating the peculiarities of the IDTV environment, that range from the limited computing power of the receivers (referred to as set-top boxes) to the limited interaction capabilities achievable through a remote control (see [5]).

The most remarkable peculiarity stems from the fact that, whereas e-learning courses have text and graphics as a central axis, t-learning ones should be naturally based on audio and video, to exploit the inherent multimedia capabilities of a TV set. This wealth of multimedia contents suggests a distinction between user-driven and media-driven strategies for interactivity [9]. In user-driven strategies, applications respond to the users’ actions, as usual in e-learning services. In media-driven ones, the evolution of pieces of media controls the flow of applications and guides users, who take part in a reduced number of decisions. Which approach is most suitable for a given service depends on the role that users are expected to play: user-driven strategies are recommended for active roles, and media-driven strategies for more passive ones (particularly, media-driven strategies are the best option for edutainment). However, it is agreed that the approach should always be mixed, to always leave some control in the users’ hands.

3. A scheme for distributed t-learning

The work we presented in [9] addresses the needs of broadcast-based t-learning services, offering mechanisms to efficiently exploit simulated interactivity. The extension to real interactivity and collaborative learning that we propose here demands additional solutions: services are conceived as distributed ones and, as commented in Sect. 2, mechanisms are needed to control the way users interact and self-organise into virtual learning communities. This also requires means to differentiate roles among users, group management capabilities, etc.

This section details our approach to the development and deployment of distributed t-learning services. We begin by
introducing the architecture we propose for educational services (referred to as courses). Then, we present the technologies we consider most appropriate to implement our approach over the foundations of the MHP standard. Finally, we describe a network architecture targeted at supporting collaborative services by making good use of the different communication media available to set-top boxes.

3.1. The architecture of a course

We have devised a simple and flexible structure for the courses, which are made up of a set of pedagogical units (PUs) that define most of the service’s logic. PUs are the primary level for contents organisation, and they can be arbitrarily complex, containing any kind of elements (text, graphics, media clips, user interface controls, etc.). As illustrated in Fig. 1, those elements can be laid out in multiple sceneries and scenes, and several types of interrelations can be established among them. The sceneries determine how the logic of a PU is distributed among a number of different machines —those of users and service providers (spatial organisation). On the other hand, scenes provide for the temporal organisation of activities.

Scenes were already introduced in [9], but sceneries are a specific need of distributed courses. Besides supporting distribution, they provide for the definition of roles among users, setting the basis for the establishment of virtual learning communities. For example, in a remote lecturing service, the lecturer and his audience run different sceneries, so that only the lecturer is given elements to control the sequencing of the slides.

In a given PU there may exist as many roles as sceneries have been defined, and how a user is assigned a given role is service-dependent: it may be decided by the user himself, by other users, etc. Among other properties, it can be defined how many instances of a given scenery can be active at the same time within a virtual community (only one lecturer and up to fifteen people in the audience, for instance).

The Course Manager and conditional access The composition of a course defines the ordering of its PUs and the conditional access dependencies among them. These can be used, for example, to reject access to one PU until some knowledge has been proved on the preceding ones.

Conditional access is supervised by an entity called the Course Manager (CM in Fig. 1). In contrast with the work presented in [9], in a distributed t-learning service the CM is not an unique entity, but a replicated object running on every machine inside a virtual community where the course is active. This makes it possible for access conditions to be evaluated for every single user, for all the members of a community with a given role, etc.

The Scenery Managers and contextual binding Every scenery runs a Scenery Manager (SM in Fig. 1), that acts as a centralised point for the communication with other machines. It is also in charge of controlling the coordinated presentation of multiple pieces of information, which, as commented in Sect. 2.1, is an essential feature for t-learning. For this purpose, we intentionally avoid standards for multimedia synchronisation available to Web services (like SMIL because, as commented later, they do not fit well with the characteristics of an IDTV receiver. Instead, we apply the contextual binding mechanism we introduced in [9], conveniently extended to support synchronisation among different sceneries. In the remote lecturing example, this permits all the members of the audience to follow the explanations of the lecturer as he browses a slide show, by automatically keeping their screens up-to-date.

Our mechanism is based on the definition of contexts, which are identifiers bound to the scenes of a pedagogical unit or to specific parts of an informational resource (delimited by anchors in text files, by timestamps in pieces of audio and video, etc.). Synchronisation works in a way that, when a context change occurs in an element of a PU, the corresponding Scenery Manager tells the rest of the elements of the same scenery to switch to the new context, and also propagates the change to the other SMs, which do the same in their sceneries. Having done this, all the elements throughout the PU that recognise the new context switch to it by taking appropriate actions.
3.2. A few notes about MHP

Prior to describing the technological framework that we propose to implement t-learning services within the scheme of Sect. 3, we discuss the support provided by the MHP standard regarding the aspects mostly related to the purposes of this paper: the fundamental programming technologies and the communication protocols.

Application models MHP defines two types of applications: DVB-HTML and DVB-J. The DVB-HTML model follows a declarative approach, adapting technologies from the Internet world to address the specifics of the television context. As commented in [11], DVB-HTML is being used for very few developments, because it has been found too difficult to use and implement [5]. Besides, delivering applications as tagged files to be parsed and interpreted can have a serious impact on their performance, due to the low computing power of a set-top box. This explains why we avoid using SMIL to synchronise the presentation of contents, as explained in Sect. 3.1.

The other approach, DVB-J, has been much more successful. DVB-J applications (commonly referred to as xlets) are written in Java, being subject to two main restrictions, related to the APIs they can make use of and the life cycle they must implement. The Java language, enhanced with the extensive set of MHP APIs, is more powerful than its declarative counterpart and provides more specific support for the characteristic features of the IDTV medium. Therefore, it should be no surprise that we have chosen DVB-J for our implementation.

Communication protocols MHP rules about the protocols to be used for accessing broadcast data and for the return channel. As shown in Fig. 2, broadcast is based on MPEG-2 elementary streams for live multimedia content and DSM-CC object carousels for other resources (mostly files). On the other hand, the return channel is mostly operated by means of the TCP/IP suite of protocols.

The MHP communications solution does not provide explicit support for group communication, being only suitable by itself for limited feedback from individual users to service providers. Moreover, the protocol stacks were designed by assuming a rigid separation between broadcast networks and return channels. However, it has been recently argued (see [6]) that some advantages can be achieved by making transparent use of both types of networks, benefitting from the robustness and high quality of service of broadcast networks and from the flexibility and bidirectionality of the Internet. By the way, remark that most of the current set-top boxes (including MHP ones) do not support audio or video streaming through the return channel, so any audiovisual contents shall always be broadcast.

3.3. Implementation technologies

Having selected the DVB-J approach, we completed the technological framework for our implementation with several freely available technologies that fit well within it: XML, JavaBeans, LOTOS [8] and JXTA [12]:

- We use XML for many different purposes, such as to communicate with IMS databases and content repositories, to define the message types needed for an application, and to express all those entities with a predefined structure, like the composition of a course in terms of its pedagogical units or the decomposition of a piece of information in contexts.

- We resort to JavaBeans as the simplest way to construct applications. It defines an architecture of components (beans) that allows development to be carried out in an entirely visual way, provided that a sufficiently rich set of beans is available together with a suitable environment for their composition.

- We have found the LOTOS process algebra suitable to define interaction patterns among the users in a virtual community, taking into account the diverse roles they may take up. Briefly speaking, our approach is to model users as agents who behave independently of others, although they may have to agree on certain actions. LOTOS' capabilities to describe synchronisation and concurrence make it easy to state, for example, that a lecturer can only go on with the next slide when all the members of the audience allow him to (because they have no more questions to ask about the current slide), or that the only possibility for the members of the audience is to go home after the lecturer finishes the presentation.

- Finally, JXTA is an open-source initiative aimed at providing a thin layer on top of which peer-to-peer (P2P) applications and services can be built. We consider P2P technologies to be suitable for the purposes of collaborative t-learning because they promote decentralisation, in a way that end users and devices are given much more relevance than in classical client-server architectures. Moreover, they provide for direct interaction among users, and offer powerful mechanisms for the establishment of virtual communities.

JXTA technology fulfills the needs of the sceneries approach for distributed logic commented in Sect. 3. We use it as the basis for communication among the users in a virtual community, because it natively supports multicasting and addresses important issues such as resource discovery or group management.

XML and JavaBeans were introduced in the context of broadcast-based t-learning services in [9], and their use is slightly enhanced for distributed ones. On another hand, a
3.4. Virtual networks and virtual communities

JXTA technology is network and language-independent, and it can be used on a wide range of devices (including PersonalJava-compliant ones, such as MHP set-top boxes). It defines a set of protocols that provide flexible mechanisms for group communication, and also make it possible to establish virtual networks on top of Internet and non-IP networks. Thus, we consider JXTA to be an ideal candidate to overcome the lacks of the MHP communications solution, commented at the end of Sect. 3.2.

In our approach, a virtual network maps directly to a virtual learning community, with all the agents involved being treated as peers (see Fig. 3).

All the entities in a virtual network (services, informational resources, user profiles, etc.) are given an identifier, that is published by means of advertisements. These are XML descriptors with information about the advertised entities and pointers indicating where they can be found.

The mechanism used for discovery is based on rendezvous super-peers, i.e., peers in well-known locations designed to cache advertisements. Due to the limited computing power of set-top boxes, we propose to use service providers’ computers as rendezvous nodes (see Fig. 3). For this purpose, these nodes maintain state information about the services being executed at any given moment, managing a set of distributed hash tables [15] that contain information about the users who are currently active in the different virtual communities, the roles they are playing, the resources they are accessing, etc.

JXTA defines a second type of super-peers (relays) to deal with heterogeneous networks and protocols, allowing applications to exchange messages with no concern about the networks they traverse. We exploit this idea to introduce broadcast relays, which provide for the combined use of broadcast and IP networks for communication inside a virtual community. Broadcasting is a natural option for information flows that must be served to all the members of a virtual community, as it happens with the explanations of the lecturer in the remote lecturing example. Besides, as commented in Sect. 3.2, it is currently the only option to transmit any audiovisual content generated by a user. The messages to be broadcast must be sent to a broadcast relay, which forwards them onto the broadcast networks addressed to the corresponding virtual community.

Following these ideas, we have implemented a JXTA communications layer to make transparent use of broadcast networks and return channels. This is shown in Fig. 4, where the gray arrows only exist in broadcast relays.

Regarding the return channel, our communications layer uses the TCP and UDP protocols. As for broadcast, it uses MPEG-2 sections for streaming audio and video content, object carousels for files, and UDP over Multiprotocol Encapsulation for messages due to bindings between elements in different sceneries (remember Sect. 3). Thus, all the messages exchanged among peers are subject to being broadcast—especially when they are targeted to many others—, tak-
ing advantage of the inherent capabilities of broadcast networks for multicasting. The mechanism to use is decided considering developers’ guidelines and traffic conditions, in order to use the available networks in the best possible way.

3.4.1. Templating As we commented in [9], our approach promotes the use of templates to construct services, with explicit separation of their functionality, appearance and contents in order to facilitate reuse. This leads to great reductions in the size of the information needed to run a service, which, in the context of broadcast-based t-learning services, serves to mitigate the effect of the latencies due to the transmission of files in object carousels. Now, we use the same approach to speed up the loading of the resources that are most frequently accessed in the virtual networks.

Taking advantage of the state information they store (remember Sect. 3.4), rendezvous nodes can detect which resources are the most utilised ones. The results of this survey are periodically sent to broadcast relays, where they are taken into account to schedule the broadcast emissions. On the receiver’s side, the JXTA communications layer monitors the information that is being received through the broadcast networks. Thus, when an application requests a given resource, it is immediately known whether that resource is available through broadcast, in which case it can be accessed with no need to issue a query through the return channel. Clearly, this helps to speed up accesses and avoids Internet bandwidth squandering, witnessing the advantages of combining broadcast networks and the Internet.

It is not difficult to envisage that a same set of Java classes may be reused for many services, displaying different contents under different appearances. Again, this would be the case of the classes that implement sceneries for remote lecturing services.

4. Integration in a development environment

Having decided on the technologies that would be used to construct services, we integrated them all into a CASE tool that assists developers all through the creative process, from the design of individual pedagogical units to their integration in courses. The utility of such a tool is clear after taking a look at the existing IDTV development environments, which do not provide an adequate support for the particular needs of t-learning (see [9]), nor for the design of distributed services.

Our tool is actually an extension of the one we presented in [9] for broadcast-based services. It has been implemented on top of the NetBeans platform, which provides quite an adequate framework to define environments for the visual development of applications. This liberates service designers from technological details, so they can concentrate on the aspects that make up a value-added application.

Developing with JavaBeans Our tool implements the sceneries approach for distributed logic presented in Sect. 3. Applications are created by placing Java beans in the different sceneries, setting up their properties and establishing bindings among them. It is up to our tool to generate the necessary Java code.

It is worth noting that MHP receivers do not have a window manager, and the only way to have a service span over multiple frames (for example, one offering controls to browse a multimedia library and other displaying the selected contents) is to build each frame as an xlet. This separation faces the problem that, for security reasons, it is impossible for one xlet to get references to objects in others. In order to communicate, they have to use a version of the Java Remote Method Invocation (RMI) technology, adapted to local communication.

Thus, three cases are possible when it comes to generate the code needed to interrelate beans, with different technologies being used in each case. These are illustrated in Fig. 5, where, for simplicity, we do not represent the many scenes that may exist in each of the frames.

1. For local bindings between beans that belong to the same frame of the same scenery, we use the classical JavaBeans mechanism of event adapters: an object is created that registers itself as a listener for certain events in the source bean and invokes the adequate methods in the destination one.

2. Local bindings between beans in different frames of the same scenery are dealt with by means of the inter-xlet communication mechanism discussed above. Basically, a wrapper for the destination bean is created that implements the java.rmi.Remote interface.

3. Finally, we handle all bindings that extend to different machines by exchanging JXTA messages. Each Scenery Manager (Figure 1) collects the events of its scenery that are programmed to cause actions in other sceneries (or in other instances of the same one), and issues messages notifying their occurrence. On the re-
receiving sides, the SMs distribute incoming messages by invoking appropriate actions in the target objects.

We initially looked at other solutions for interrelating distributed objects, like Java RMI and CORBA, but we rejected them for being too resource-intensive (see [7]) and complex for our needs. JXTA provides a much lighter solution, and also sets the basis for the whole peer-to-peer approach.

We have created a set of specialised beans, taking into account the specifics of the MHP standard, that range from extensions to the user interface elements to context-aware objects that manage access to and representation of multiple media resources. Course Managers and Scenery Managers are also implemented as beans, but they are not visible to application designers.

About contextual binding Our implementation of the contextual binding mechanism (Section 3) is based on interrelations between Scenery Managers and the beans that control the presentation of contents, in a way that designing pedagogical units as user-driven or media-driven ones (see Sect. 2.1) only depends on which events are handled and what contexts are defined. In this regard, our development tool provides a number of wizards with which to define the decomposition of pieces of information in contexts, defined ad-hoc for the different content types.

Some additional features

- An editor deals with the composition of the courses, by describing the ordering of the pedagogical units and the conditional access dependencies among them, as commented in Sect. 3.

- Another editor is used to define interaction patterns among the peers in a virtual community, according to what we commented about LOTOS in Sect. 3.3.

- There are wizards to operate databases and content repositories following the IMS specifications. One of them provides for the automatic creation of multiple-choice tests involving any kind of elements (text, graphics, video clips, etc.) in questions and options, with any desired layout.

- There is also a set-top box emulator to test the applications at design-time, of which multiple instances can be started to simulate a distributed environment.

5. Work in progress

Nowadays, there are not regular digital TV emissions including MHP applications in Spain and, by extension, there are no set-top boxes at homes capable of executing them. We have only been able to test the mechanisms proposed in this paper over the metropolitan area network of the Vigo University Campus, using a limited number of terminals. However, we are currently negotiating the possibility of doing experimental tests over cable broadcast networks.

The results of the tests have been quite satisfactory so far, but we do not have sufficient basis to evaluate the scalability of our approach. On the one hand, scalability is favoured by the decentralisation caused by P2P technologies, the transparent use we make of broadcast networks and return channels, and the reuse promoted by the templating mechanism. In this regard, as we pointed out in Sect. 3.4.1, it is expectable that larger deployments of collaborative t-learning services will be achieved in the future by reusing great amounts of information, especially Java classes implementing sceneries with generic functionality.

On the other hand, the main obstacle to scalability lies in the need to transmit all the multimedia flows through broadcast, because a few large communities could generate an overwhelming amount of data. It would be highly advisable to support streaming through the Internet in future revisions of the MHP standard.

Regarding our development tool, some work still remains to be done in its design, mainly in eliminating completely the need to write any line of Java code. As soon as it is finished, we have planned to release it as a commercial application. On the contrary, the JXTA communications layer commented in Sect. 3.4 will be made available as free software for the JXTA community, in the hope that it will serve to inspire further work.

6. Discussion and future work

In this paper, we have presented an architecture for distributed IDTV services, and a selection of freely avail-
able technologies suitable for its implementation. This work comes as an extension to the mechanisms defined in the MHP standard to operate the return channel. Our proposal is based on peer-to-peer technologies, as they natively provide for the organisation of users into virtual communities of people with shared interests. We have also discussed the convenience of establishing a virtual network on top of the physical ones; in this context, the use of broadcast networks for multicast inside a community is a very interesting feature, particularly when dealing with multimedia content.

The introduction of P2P technologies in IDTV will probably have a significant impact on its business models. The development of t-learning initiatives has been so far controlled by mainstream broadcasters, leaving most of the educative community apart. Because P2P promotes decentralisation in networks and services, the approach we suggest here contributes to openness in the educational market, leaving also place for private enterprises, and even individual users, to offer learning services (advertising them is just part of the P2P framework, as commented in Sect. 3.4). This opens up several new ways for broadcasters to make business: a) by hiring bandwidth in their broadcast networks to provide high-quality streaming of multimedia content; b) by hiring computational power and storage capacity for complex computations and large databases; or c) by giving access to their content repositories, that store numerous selected pieces of learning or audiovisual material.

Some important future lines of research in the t-learning field deal with personalisation and convergence with other mediums. Personalisation is a need in order to handle the increasing amount of information available, and also to target services and products to users who may be interested in them. The TV-Anytime Forum is working on solutions for the definition of user profiles and mechanisms for the markup, storage and retrieval of multimedia content. In this regard, our working group is now applying TV-Anytime specifications in designing a recommender of personalised contents that combines different strategies to infer knowledge from user profiles and viewing histories [2].

On its part, the desired convergence of IDTV with other mediums requires further advances in interoperability of software and contents. The most relevant roles here will surely be played by the Java language, XML and the MPEG-4 standard. Java is unquestionable because of its portability and adaptability to a range of different devices. XML is steadily gaining acceptance as a common format for structuring, exchanging and processing information (for example, it is at the core of the SCORM and IMS standards). Finally, MPEG-4 is called to revolutionise the creation, distribution and use of multimedia content. Unfortunately, the MHP standard does not currently support MPEG-4, but it probably will in future versions; in fact, set-top boxes following the MHP mas-

ter lines and including MPEG-4 decoders have already been demonstrated (see [6]).

The inclusion of MPEG-4 support will cause some changes in the way of doing things, probably extending the focus in development to media authoring tools. In any case, tools such as the one presented here will still be needed, for example, to deal with distribution, content management, the definition of differentiated roles among users, and any other additional logic.

References