

Research Article

Web Content Search and Adaptation for IDTV: One Step Forward in the Mediamorphosis Process toward Personal-TV

Stefano Ferretti,¹ Marco Rocchetti,¹ and Claudio E. Palazzi^{1,2}

¹ *Dipartimento di Scienze dell'Informazione, Università di Bologna, Mura Anteo Zamboni 7, 40127 Bologna, Italy*

² *Computer Science Department, University of California, Los Angeles, Boelter Hall, Los Angeles, CA 90095, USA*

Received 27 December 2006; Accepted 3 April 2007

Recommended by Hao Yin

We are on the threshold of a mediamorphosis that will revolutionize the way we interact with our TV sets. The combination between interactive digital TV (IDTV) and the Web fosters the development of new interactive multimedia services enjoyable even through a TV screen and a remote control. Yet, several design constraints complicate the deployment of this new pattern of services. Prominent unresolved issues involve macro-problems such as collecting information on the Web based on users' preferences and appropriately presenting retrieved Web contents on the TV screen. To this aim, we propose a system able to dynamically convey contents from the Web to IDTV systems. Our system presents solutions both for personalized Web content search and automatic TV-format adaptation of retrieved documents. As we demonstrate through two case study applications, our system merges the best of IDTV and Web domains spinning the TV mediamorphosis toward the creation of the personal-TV concept.

Copyright © 2007 Stefano Ferretti et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

Television (TV) still represents the most widespread domestic appliance with a 99% of penetration in western world's homes [1]. Currently, interactive digital television (IDTV) embodies the first step in the ongoing evolution of this medium through which TV users are becoming active subjects able to interact with broadcast contents, sending back various kinds of feedback.

Through the IDTV *mediamorphosis* [2, 3], traditional TV watching will be coupled with new interactive digital applications that may be developed to run on TV. As a result, TV watchers will pass from a passive role to an active one [4]. To this aim, broadcast interactive digital applications will be retrieved by the set top box (STB) and executed on the TV screen [5, 6]. These applications may be complementary to TV programs (e.g., voting systems, electronic program guides), or even completely uncorrelated to on-air shows, thus representing new services (e.g., games, news, t-learning, t-commerce, delivery of Web contents) exploitable while TV programs go on.

Digital transmission technology has the merit of pushing forward the frontier of experimentation; however, it does

not yet possess all the resources that are required to sustain the whole process of TV mediamorphosis. To complete this process, researchers are now called to combine IDTV with the most powerful and complete medium able to carry information: the Web. Indeed, also thanks to the larger and larger diffusion of broadband connectivity, the Web has gained a tremendous importance since its development, generating new forms of business models and digital entertainment applications such as teleworking, online magazines, advertisement systems, online auctions, blogs, instant messaging, online games, interactive audiovisual services, and IPTV.

The integration of IDTV and Web worlds represents an exceptional opportunity to inject new fuel into both systems creating a huge, accessible container of all sorts of digital services. Advantages of this combination would be reciprocal, for example, the enlargement of the Web audience to TV watchers, interactive services provided to the TV audience, Web services brought to people who cannot afford to buy a fully equipped computer to browse the Internet or refrain from using one since they lack the knowledge to manage this kind of technology [3].

Nevertheless, IDTV and the Web have been developed as two different technologies, thus posing several challenges

to their combination and requiring the development of new ad hoc solutions. For instance, to allow an effective Web browsing through the TV, two macro-problems have yet to be solved:

- (i) the retrieval of Web contents performed on the basis of precise requests of TV watchers;
- (ii) the proper presentation of these contents in a format suitable for a TV screen.

Indeed, since the limited space size of *TV carousels* [2, 7], it is not feasible to naively hypothesize that all the Web is converted a priori into a TV-friendly format and then broadcast over TV channels. At the same time, a centralized approach that let the IDTV providers select, retrieve, convert, and finally present Web contents to users does not embody either an adequate solution as it would strongly limit the level of autonomy and interaction provided to users.

Instead, we propose a novel approach to address the two aforementioned macro-problems. Our solution exploits the DVB multimedia home platform (MHP) standard [6] and is composed of four main modules:

- (i) a user interface to facilitate users in expressing their preferences about contents they are interested in;
- (ii) a mobile agent system that permits to dynamically retrieve Web documents on behalf of the user and pushes them onto the TV carousel;
- (iii) an automatic transcoding service able to adapt Web contents in a format suitable for IDTV;
- (iv) a DVB-Java (DVB-J) browser application broadcast via the carousel and locally executed on each STB to properly manage transcoded contents and visualize them on a TV screen.

As it is evident, the first two modules address the first macro-problem, whereas the third and fourth modules focus on the second one.

Based on our system, various novel applications and services can be devised. In order to assess the viability of our solution, we have developed two exemplar case applications: a TV dashboard and a Web news presenter. The former involves the development of several *widgets* (i.e., mini-applications) that run over MHP-compliant STBs, allowing users to configure mobile agents so as to dynamically inject their own personal messages (e.g., advertisements, pictures, text) into the TV carousel. Furthermore, users can also specify which kinds of messages/advertisements they are interested in. This way, a client application (i.e., an *Xlet*) running on STBs can filter broadcast data items and provide users with pertaining contents. Instead, the latter is utilized to watch on a TV screen sport news that are published on dedicated Web sites.

Our system is able to promote the development of new forms of advertisement and t-commerce. Moreover, a vast plethora of new applications and services becomes feasible through it. By exploiting the peculiarities of both the Web and interactive TV, our solution is able to provide the best of both worlds thus paving the way for the development of new sophisticated, integrated IDTV-Web services which rep-

resents the essence of a TV-mediomorphosis toward a new medium: the personal-TV.

The remainder of this paper is organized as follows. Section 2 provides background related to our work. Section 3 describes in detail our solution. In Section 4, we report on features of two specific applications we have developed as proof-of-concept to test our system. Finally, Section 5 concludes this paper.

2. BACKGROUND

This section presents concepts that are at the basis of our approach. Specifically, we first provide a general overview of the IDTV world in Section 2.1. Then, we devote Sections 2.2 and 2.3 to discuss the state-of-the-art of the two macro-problems stated in Section 1.

2.1. The IDTV world

With the term IDTV experts refer to TV with interactive contents and digital enhancements. The idea is that these technologies can be exploited to let TV watchers move from a passive role to a more active one [4, 8].

Interactive TV is not a new concept. In the past, this term has been widely employed as the top-level classification for all kinds of interactivity that involved TV [9]. For instance, it included video-on-demand systems as proposed in the U.S. since the 80's, which enabled users to interactively choose programs to be watched. However, those early interactive solutions for TV were soon abandoned due to the high cost of bringing two-way networks into people's homes [10]. With the recent advent of always on domestic broadband connectivity (xDSL), new proprietary solutions have been made available by companies such as OpenTV, NDS, Canal+, PowerTV, and Microsoft. However, neither these initiatives represent the perfect solution able to engender a general and persistent interactive model for TV utilization. Indeed, all these initiatives lack an open standard and propose, instead, a proprietary solution based on a "vertical market" notion. Simply stated, each network provider distributes its own STBs already endowed with peculiar features (not only applications and services, but contents too). Needless to say, this represents a design inadequacy that prevents interoperability and the emerging of a new standard able to drive the personal-TV revolution.

As of today, the adoption of IDTV technology is a phenomenon that spreads all over the world. In Europe, Asia, and Australia, the IDTV market development is lead by the Digital Video Broadcasting (DVB) Project [5], whereas the OpenCable Application Platform (OCAP) [11] and the ATSC's Advanced Common Application Platform (ACAP) [12, 13] represent the standards that are leading the TV innovation in the U.S., and the Integrated Services Digital Broadcasting (ISDB) initiative [14] is active in Japan. In our work, we focus on the European DVB standard, as it was the standard available for our experimentation; however, considerations expressed in the paper can be easily extended also to other IDTV standards in the world.

IDTV systems enable the broadcast of data via several transmission media. As to DVB, for example, data can be broadcast via cable (DVB-C), satellite (DVB-S), digital terrestrial (DVB-T), and wireless signals for handheld devices (DVB-H). As a common feature to all these different transmission channels, DVB specifies that data need to be broadcast by exploiting *carousels* [2]. In essence, a carousel may be represented as a cyclic transmission of data structured as in a common file system [5].

Not only has DVB specified protocols for data transmission over different networking media, but it has also provided specifications on the structure of the software inside an STB, that is, the multimedia home platform (MHP) [6]. MHP defines a programmable interface between interactive digital applications and the terminals on which those applications are executed [15]. Specifically, Java applications can be created to run over MHP-compliant STBs. These are commonly termed DVB-Java (DVB-J) applications or, more simply, *Xlets*, due to their similarities to Java applets. Basically, Xlets are composed of a set of class files and other resources, which are broadcast within the DVB data carousel to be executed over STBs.

Thanks to Xlet applications, data broadcast on the carousel can be managed and presented to watchers directly on their TV screens. Furthermore, users are allowed to interact with Internet servers and TV broadcasters through an Internet-based return channel, thus providing means for the development of a whole plethora of interactive services (e.g., advertisement, pay-per-view, voting, gaming, t-commerce, home shopping) finally available even through the TV screen.

2.2. Automatic Web content retrieval

In order to combine IDTV with the Web, the first macro-problem that needs to be solved is represented by how to retrieve Web contents that correspond to TV user's preferences/subscriptions given the huge amount of available information. To this aim a feasible solution could make use of mobile agents.

Mobile agents are programs that can migrate from host to host in the Internet. Times (and places) at which agents decide to migrate are due to autonomous agents' decisions, based on the specific tasks they have been instructed to accomplish.

Indeed, two main reasons suggest to exploit this technology to retrieve Web contents on the behalf of users. First, the explosive and continuous growth of the Web may overwhelm the ability of the average user in actually performing a proficient search [3, 16]. This is especially true in the depicted scenario, as we are considering (TV) users that could even had no practical experience with Web browsing. Therefore, smart mobile agents can be really helpful by performing information discovery, processing, and filtering on behalf of their users. Second, an application thought to be executed over IDTV-based terminals has to deal with the limited (computational and networking) resources of current STBs [6]. This consideration gains even further importance if one thinks of

emerging IDTV-based handheld devices, for example, DVB-H compliant mobile terminals [2].

Summing up, a system where users directly browse the Web through Xlets operating on STBs does not represent a feasible solution. The actual resource (e.g., time, bandwidth, processing) consuming task of retrieving Web contents of interest for the user has to be moved closer to Web servers. Smart mobile agents can be really of service in this sense making beneficial the fusion of IDTV and Web domains.

2.3. Formatting Web contents for IDTV presentation

The second macro-problem concerns the proper presentation of Web contents in a format suitable for a TV screen. Indeed, whereas it is widely accepted that TV contents can be delivered through the Internet (and IPTV initiatives demonstrate this claim) [5, 7, 13, 17, 18], yet the reversed case, that is, making Web contents available on TV screens, is far from being a service effectively available.

Examples of authoring tools exist that allow TV providers to create enhanced audiovisual (hyperlinked) contents that can be interactively enjoyed by TV users; for instance, GMF4iTV [19], Hypercafé [10], and Viper [1, 20]. Similarly, the Jules Verne project is currently working on the integration of MPEG4 and MHP [21]. However, none of these solutions can be directly exploited to browse existing Web contents on a TV screen, as they all require a manual editing of contents that have to be visualized.

Focusing on specific DVB-based solutions for transcoding Web contents for TV visualization, the DVB consortium has identified in DVB-HTML the new markup language for MHP-based IDTV systems [6]. Indeed, DVB-HTML follows the declarative approach inspired by the W3C and specifies how Web pages should be organized to be displayed on a TV screen [22]. Yet, opinions about DVB-HTML are widely controversial. In fact, DVB-HTML is regarded as a complex specification, not suitable for devices with poor computational capabilities (e.g., current STBs and new DVB-H devices) [8, 23]. Moreover, no official testing procedure exists for this language, thus entailing risks for the interoperability among different implementations of MHP.

As a result, DVB-HTML is not even supported by current off-the-shelf MHP-STBs and, to overcome this limitation, ad hoc solutions have to be implemented such as developing specific DVB-J applications able to display DVB-HTML documents. The Nionex pontegra [24] and the ORTIKON ACE MHP [7] platforms represent two prominent exemplars of this kind of initiatives. However, due to the inherent complexity of DVB-HTML, only a specific subset of DVB-HTML is supported by these platforms. Furthermore, since standard HTML pages are out of the scope of DVB-HTML, a negligible fraction of Web pages is factually displayable on current STBs through this approach.

These considerations provide evidence to the following two claims:

- (i) DVB-HTML's specification is too complex to be supported by current STBs (i.e., resulting Xlets would pro-

vide coverage only for a small percentage of pages defined according with this standard);

- (ii) current (HTML-based) Web pages cannot be displayed on TV screens unless they are recoded from scratch utilizing DVB-HTML.

Consequently, a transcoding service able to automatically convert (HTML-based) Web contents for display on TV screens represents an indispensable, yet still effectively missing, tool to fuel the mediamorphosis process toward personal-TV.

3. PROVIDING WEB CONTENTS ON IDTV: AN EFFECTIVE SYSTEM ARCHITECTURE

The Web and IDTV represent two worlds that have been developed through distinct tracks; merging them now passes through finding appropriate solutions for the Web content retrieving and visualization macro-problems stated in Section 1. This is exactly what our system accomplishes.

Furthermore, whereas alternative approaches leave to IDTV providers the complete decision power about which contents have to be transmitted through the TV carousel, our system leaves this decision to TV watchers, which can hence receive personalized information.

Our approach clearly opens up new business scenarios. New forms of applications and services can be devised such as, for instance, advertisement, TV/online auctions, TV dashboards, and Web-TV.

The remainder of this section is devoted to describe in detail our devised solution; in particular, we subdivide the explanation of our system components into two subsections, each corresponding to one of the two aforementioned macro-problems.

3.1. Gathering contents from the Web

The first macro-problem we address in our work is represented by how to collect information based on user's preferences. This can be subdivided into two smaller tasks: (i) enabling users in expressing their preferences and (ii) retrieving Web contents based on these preferences.

To cope with the first task we have developed an Xlet that runs on a user's STB to let her/him specify contents of interest. In particular, user's preferences are collected through a graphical interface and encoded based on the specification protocol. Instead, the second task is performed by the means of mobile agents that are instructed according to user's specifications to search the Web on user's behalf.

With the help of Figure 1 we clarify the functioning of our system. When a new request is expressed by a TV watcher (step 1), the Xlet running on the STB receives this request and transmits it to a specific host (e.g., host A in Figure 1). This way, host A can spawn a new mobile agent, instructed following user's preferences as reported by the Xlet (step 2). The mobile agent moves in the Internet visiting different hosts (steps 3 and 4) and, at each host, it performs a query for requested contents; results are managed and stored within the agent. Once the agent has visited all nodes it planned to visit,

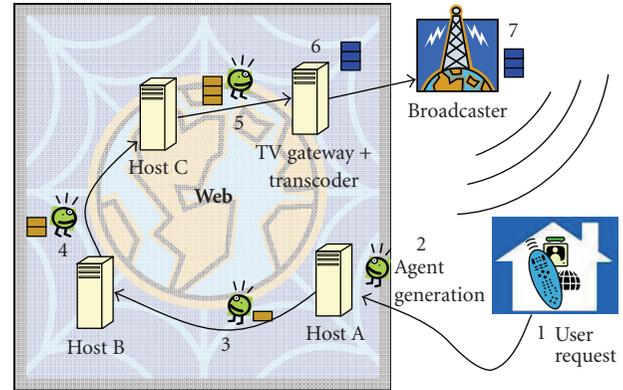


FIGURE 1: The system architecture.

it moves to a final host, which acts as a gateway to the IDTV domain (step 5) to deliver all retrieved data. These data are then checked for duplicates and corruptions (step 6) and automatically transcoded from their Web format to a format suitable for IDTV. We dedicate Section 3.2 to discuss this task at length, as this represents one of the main contributions of our system. Finally, transcoded contents are pushed onto the TV carousel and broadcast to TV watchers (step 7).

The TV carousel obviously contains a plethora of documents, as different users request different kinds of contents, which are all included in the same TV carousel. Therefore, the carousel is filtered at client side to locally present only contents requested/subscribed by each user. In particular, information filtering is accomplished by resorting to metadata associated to XML documents related to broadcast contents (see Figure 2).

A crucial issue regards the discovery of those hosts that are enabled to accept the arrival and the execution of mobile agents and, among them, those which may contain data of some interest for TV users. Finding an optimal solution for this issue is out of the topic of this paper; we just adopt a naive (even if effective) solution, according to which, the agent gains knowledge of hosts to visit at its generation on the home host (i.e., host A in Figure 1). This implies that the host that generates mobile agents embodies also the functionalities of a centralized discovery server. However, several alternative and more sophisticated solutions may be deployed. For instance, DHT-based systems could be employed (e.g., Chord [25], Pastry [26], CAN [23], Tapestry [27]), thus, having the agent performing a *lookup(keyword)* operation to retrieve the complete lists of nodes matching its query.

Focusing on the factual implementation of mobile agents, we developed the prototype of our system resorting to the Java agent development (JADE) framework [28], a free, FIPA-compliant platform for mobile agents [29]. We have chosen a Java-based framework to create our mobile agents as it allows a simpler integration with Java-based Xlets that are executed on the MHP STB.

Figure 3 reports a sketch of the Java source code implementing the mobile agent (DVBAgent), which is in charge of searching Web contents based on the user preferences.

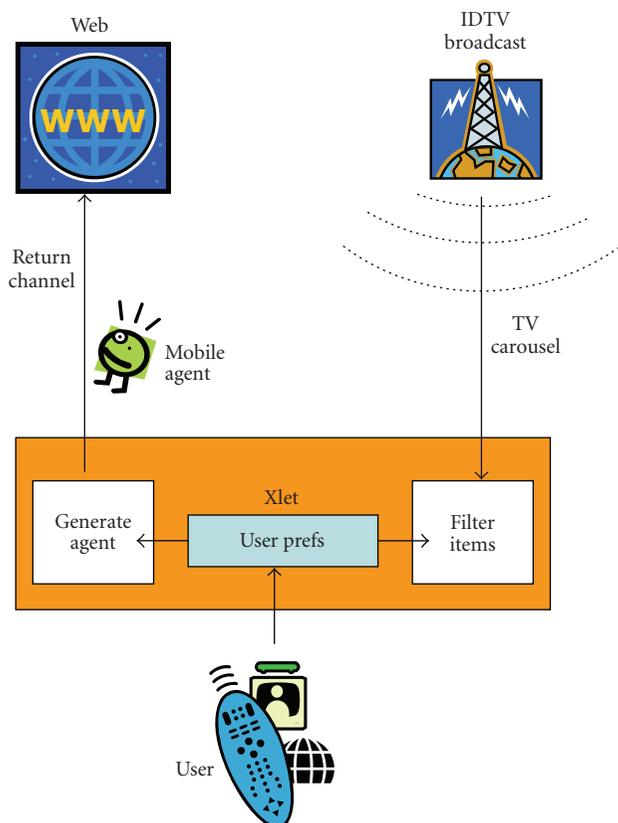


FIGURE 2: Xlets interaction with IDTV and the Web.

As soon as a user has requested its intervention, an agent is created and then initialized (see Figure 3, `setup()` method, line 2). Prior to becoming operative, an agent behavior has to be specified to be used by that agent (`addBehavior()`, line 4). This task is accomplished by means of the `WaitSettingsAndGo` class. Obviously, the agent waits for the user preferences to be specified before it can move towards some hosts to search for contents (lines 7–9). To move from one host to the next, a `go()` method has been implemented, which is devoted to select the next host to visit (line 11). Upon arrival at a given host, a query is performed to search for contents, based on the user preferences (`queryForContents()` method, line 12).

We resorted to XML-based technologies coupled with the FIPA ACL (agent communication language) to represent keywords, users' preferences, and results managed by mobile agents. In particular, we created a simple specification protocol that is used by agents in our system. Even in this case, a more elaborated solution can be exploited such as those inspired by the Semantic Web research groups or FIPA-compliant querying communication protocols [29, 30].

As an example of how user preferences and requests may be specified within our system, we report in Figure 4 the XML code produced by our Xlet, which specifies the request a user has issued to exploit the function of a weather forecast application we have developed for use on the TV. This request amounts to (i) the list of cities in the U.S. for which the

user wants to know the temperature (lines 7–9); (ii) the list of weather forecast sites which should be asked for temperatures based on the user preferences (lines 1–3). Obviously, to interpret the XML document of Figure 4, our system manages a correspondent DTD which is reported for the sake of completeness in Figure 5.

As an example of how our system responds to the request issued by the user, we provide in Figure 6 a document comprised of the values of the temperatures retrieved by our mobile agent after searching the Web. Also in this case, an XML syntax is exploited (the related DTD is reported in Figure 7). Basically, for each city the mobile agent has retrieved one or more forecasts, which are embodied in the XML document. The resulting data are then managed at the TV Gateway host (see Figure 1) for insertion in the TV carousel.

3.2. Browsing transcoded Web contents on IDTV

The second macro-problem we address is represented by the proper visualization of retrieved Web contents on the TV screen. The solution adopted by our system passes through performing two smaller tasks. The first task regards automatic transcoding of Web contents retrieved by the mobile agent into well-formed XHTML Basic documents that allow visualization on IDTV screens. However, as TV sets possess specific visualization and interaction characteristics, a specific browser has to be developed to allow a correct interpretation of transcoded documents and their management through IDTV screens and remote control.

Focusing on the first task, our implemented transcoding service steps through three different phases (branch (i) of Figure 8). In the first phase, well-formed XHTML Basic documents are created so as to have elements that can be easily managed by means of a standard XML parser (e.g., SAX, DOM, KXML, nanoXML) and standard XHTML browsers. This result is obtained by the following:

- (i) closing all start-tags with the corresponding end-tags;
- (ii) putting all tags in lowercase;
- (iii) properly nesting all the XHTML elements.

All these steps are accomplished by resorting to the facilities offered by the JTidy library, an open source project that can be used as a tool for cleaning up malformed and faulty HTML, and as a DOM parser [31].

In the second phase, XHTML tags and/or attributes that cannot be displayed on IDTV screen are removed. A detailed list of tags modified by our transcoding service and corresponding actions is reported in Table 1. The same is done for all tag attributes related to font styles (e.g., `font-style`, `font-weight`, `font-family`), as they cannot either be exploited for TV screen visualization. Indeed, MHP currently includes only *Tiresias* as a possible font to be displayed. As an additional benefit, this phase reduces the total amount of coded data. This is positive especially considering the fact that broadcasting several Web pages and media objects may easily saturate the bandwidth reserved for data transmission via carousel.

```

(0) public class DVBAgent extends Agent {
(1)     ...
(2)     public void setup() {
(3)         ...
(4)         addBehaviour(new WaitSettingsAndGo());
(5)     }
(6)
(7)     class WaitSettingsAndGo extends CyclicBehaviour {
(8)         public void action() {...}
(9)     }
(10)    ...
(11)    public void go() {...}
(12)    public void queryForContents() {...}
(13)    ...
(14) }

```

FIGURE 3: Mobile agent implementation.

```

(0) <application id="tempWidget">
(1)   <url> myPreferredForecastSite_1 </url>
(2)   <url> myPreferredForecastSite_2 </url>
(3)   <url> myPreferredForecastSite_3 </url>
(4)   <name> Temperatures </name>
(5)   <desc> Displays cities temperatures </desc>
(6)   <prefs> 8 </prefs>
(7)   <item id='San_Diego'> San Diego </item>
(8)   <item id='Los_Angeles'> Los Angeles </item>
(9)   <item id='San_Francisco'> San Francisco </item>
(10)  ...
(11) </application>

```

FIGURE 4: User preferences specification.

In the third and final phase, the transcoding service manages links present in the document. Two different kinds of links have to be considered, namely, internal and external. With internal links, we refer to those links that point to contents included in the transmitted carousel. Instead, external links refer to contents that are not included in the carousel. Our system appropriately modifies internal links so that they refer to the correct objects within the carousel. Conversely, external links are left in their original form, as they can only be managed through the Internet-based return channel.

Scripts and other advanced Web objects (e.g., ECMA-script, Javascript, Flash, SVG) are currently not supported by STBs, which typically comply to MHP 1.0.1, whereas ECMA-script support is required with MHP 1.1. Consequently, our system eliminates unsupported scripts and objects and exploits their alternative representations, if present, encoded through the `<noscript>` (or `<noobject>`, `<noembed>`) tag. This process is depicted by branch (ii) in Figure 8. As a matter of fact, W3C recommends the use of alternative HTML versions for contents associated to scripts or objects,

supplied in a form which looks like: `<noscript> alternative Web contents </noscript>`. This is aimed at guaranteeing alternative forms of presentation for users that exploit lightweight browsers (e.g., Lynx), thus, augmenting contents' accessibility. Needless to say, such an approach is required until no support for these technologies will be provided for MHP-compliant TV sets.

Focusing on general media objects (branch (iii) of Figure 8), we resort to a rich media transcoding system as described in [32]. This approach enables the conversion of several kinds of media (i.e., images, audio, video, rich multimedia presentations) into different encoding formats that are MHP-compliant.

The second task for the visualization macro-problem corresponds to properly manage transcoded contents and visualize them on a TV screen. This issue is related to the fact that transcoded data are broadcast to be displayed on very different TV screens (e.g., high definition TV sets, regular TV sets, handheld DVB-H devices). For a comprehensive discussion on these issues, interested readers may refer to [33, 34].

```

(0) <?xml version="1.0" encoding="UTF-8"?>
(1)
(2)   <!DOCTYPE application [
(3)     <!ELEMENT application (url*, name, desc, prefs, item*)>
(4)     <!ATTLIST application id ID #REQUIRED>
(5)     <!ATTLIST item id ID #REQUIRED >
(6)     <!ELEMENT url (#PCDATA)>
(7)     <!ELEMENT name (#PCDATA)>
(8)     <!ELEMENT desc (#PCDATA)>
(9)     <!ELEMENT prefs (#PCDATA)>
(10)    <!ELEMENT item (#PCDATA)>
(11)  ]>

```

FIGURE 5: User preferences specification: the DTD.

```

(0) <tempWidget>
(1)   <city id="San_Diego">
(2)     <temp>78</temp>
(3)     <metr>F</metr>
(4)     <forecast_site> myPreferredForecastSite_1 </forecast_site>
(5)   </city>
(6)   <city id="San_Francisco">
(7)     <temp>68</temp>
(8)     <metr>F</metr>
(9)     <forecast_site> myPreferredForecastSite_1 </forecast_site>
(10)  </city>
(11)  <city id="Los_Angeles">
(12)    <temp>74</temp>
(13)    <metr>F</metr>
(14)    <forecast_site> myPreferredForecastSite_2 </forecast_site>
(15)  </city>
(16)  ...
(17) </tempWidget>

```

FIGURE 6: Contents gathered by the mobile agent.

Here, we discuss the solutions we devised to tackle these visualization issues (a summary is also provided in Table 2). In particular, as regular MHP systems do not support XHTML Basic, we developed another Xlet application that acts as a browser for the XHTML Basic documents produced by our transcoding service. This Xlet is run on the STB to manage and optimize the receiver-side settings for the management and visualization of received contents, specifically.

- (1) First and foremost, a crucial problem is represented by the fact that a Web document is not structured to be displayed on a TV screen. Indeed, a TV screen has a resolution which is typically lower than that available on a PC monitor. Furthermore, even the update frequencies and the pixel size are different on the two devices [22]. In essence, spectators watch TV from a longer distance (some meters) and consequently the font size typically used for a Web page (11 pt) may result too small for TV sets. Consequently, we have de-

signed our DVB-J browser application to utilize enlarged font sizes for presentation and, when needed, to appropriately split each Web page into several TV screenshots.

- (2) Tables within Web documents can be too big to be displayed on a TV screen. Therefore, our system splits each table within the Web document into several subtables that are displayed over subsequent screenshots. To browse subtables, a navigation mechanism is provided which can be manipulated via the remote control. Needless to say, our approach is effective when tables are employed exclusively for representing data in a tabular format, whereas it would not perform well if tables were used for structuring the layout of the Web document. Yet, this latter case does not comply with the guidelines provided for Web documents' layout; rather, it represents a behavior highly deprecated by the W3C.

```

(0) <?xml version="1.0" encoding="UTF-8"?>
(1)
(2) <!DOCTYPE tempWidget [
(3)   <!ELEMENT tempWidget (city*)>
(4)   <!ELEMENT city (temp, metr, forecast_site)>
(5)   <!ATTLIST city id ID #REQUIRED>
(6)   <!ELEMENT temp (#PCDATA)>
(7)   <!ELEMENT metr (#PCDATA)>
(8)   <!ELEMENT forecast_site (#PCDATA)>
(9) ]>

```

FIGURE 7: Weather forecaster application: the DTD.

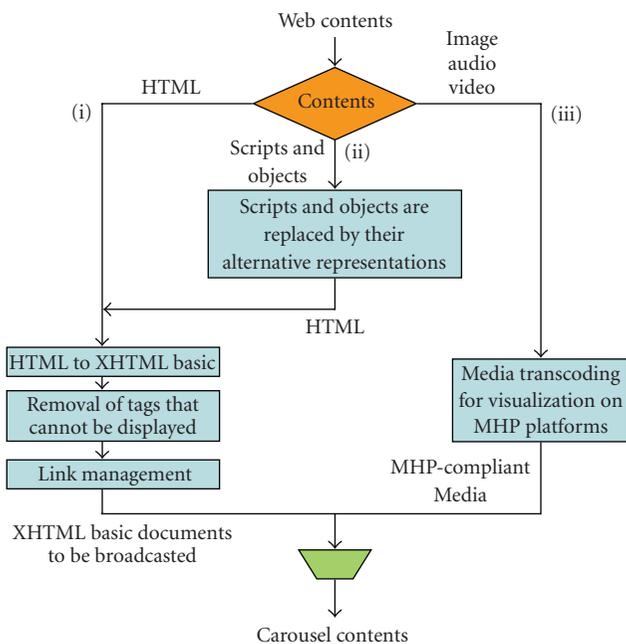


FIGURE 8: Transcoding Web contents for IDTV.

- (3) High flickering effects on the TV screen must be avoided as they may result annoying for TV watchers. To this aim, dark colors for the background and bright colors for characters are employed. In essence, our DVB-J browser aims at improving readability (as well as accessibility for partially sighted people) by employing dark blue and white as default colors for the background and characters, respectively.
- (4) Focusing on the usability issue, we have to keep in mind that the TV remote control is the device that has to be utilized to control the system at client side. Consequently, the implementation of all functions that are typically provided by a mouse (e.g., pointing, dragging, clicking) have to be replaced by buttons on the TV remote control. Considering the example of the dragging function, scroll bars are replaced by the sub-

sequent visualization of different screenshots executed through the arrow keys on the remote control.

- (5) Finally, links present within the displayed page are made available in a separate frame on the top-right corner of the screen.

As a final comment on our Xlet, it is worth mentioning that this software application has been implemented using the Java programming language and by resorting to those Java libraries needed to make it DVB-compliant, that is, JavaTV, HAVi, DVB, JTidy [5, 6, 31]. This way, the Xlet is perfectly operative when executed over DVB-compliant STBs.

4. AN EXPERIMENTAL ASSESSMENT OF OUR SOLUTION ON A REAL DVB-T SYSTEM

Aimed at verifying the viability of our approach, we evaluated our solution on a real DVB-T system. The DVB-T system was provided by CINECA [35], an Italian consortium for ICT research. In the following two subsections we report on two sets of experiments that were devoted to test the various components of our system architecture. We present two possible applications that may be implemented thanks to our architecture. In particular, we focus on the ability of our system in (i) retrieving Web documents based on user's preferences and (ii) automatic transcoding fetched Web pages for visualization on a TV screen.

Each single test was conducted as follows. Upon completion of the transcoding process, a carousel was generated by multiplexing an MPEG-2 video stream with the transcoded data provided by our solution. The obtained multiplexed stream was then passed to a DVB-T modulator, in charge of converting the generated carousel into an analog coded orthogonal frequency division multiplex (COFDM) signal. To perform controlled experiments, we connected the utilized MHP-STB with the DVB-T modulator via a TV cable. On-the-air transmissions were hence replaced by transmissions transported on the TV cable. However, this did not reduce the significance of our assessment, since it is well known that on-the-air reliable communication is guaranteed by DVB-T systems, thus, being equivalent to those carried out by means of a cable.

TABLE 1: Tag transcoding.

Tag	Action
<!--...-->	Removed
<a>	Maintained. Internal links, managed as values of the href attribute, are modified to point to the linked contents within the carousel
<area>	Removed
	Tag removed, text maintained
<bdo>	Tag removed, text maintained
<button>	Removed
<col>	Removed
	Tag removed, text maintained
<fieldset>	Removed
<frame>	Maintained. Frames are managed as linked contents. The effect is that only a single frame is displayed at once.
<hr>	Removed
<i>	Tag removed, text maintained
<kbd>	Tag removed, text maintained
<label>	Removed
<link>	Removed
<map>	Removed
<object>	Replaced by the alternative representation
<option>	Removed
<p>	Maintained if not void. Removed otherwise
<param>	Removed
<script>	Replaced by the alternative representation
	Tag removed, text maintained
<style>	Removed
<textarea>	Removed
<var>	Tag removed, text maintained
*	All other tags are left in their original form

TABLE 2: TV visualization: issues and solutions.

Issue	Solution
(1) TV set has a low resolution and is watched from a larger distance (with respect to a PC monitor)	Larger font sizes are used, Web page to be fragmented over several TV screenshots
(2) Tables can be too large to be displayed on TV	Tables are fragmented into subtables
(3) High flickering and visualization problems for (partially sighted) people	When not specified in the document, blue backgrounds and white fonts are employed
(4) A TV remote control has limited input modalities	Scroll bars are replaced by the use of TV remote control to navigate through different screenshots
(5) Positions of links in the screen	Links are moved to separate frame displayed on the top-right side of the screen

4.1. The TV dashboard application

To provide validation of our system architecture and, in particular, of the mobile agents' effectiveness in retrieving Web contents of interest for the user, we developed an ap-

plication that takes inspiration from the well-known Dashboard available for Mac operating systems [36, 37]. The main idea is that of providing the STB with a platform able to host several concurrently running, dynamic, light-weight, mini-applications (named widgets in the Dashboard jargon).

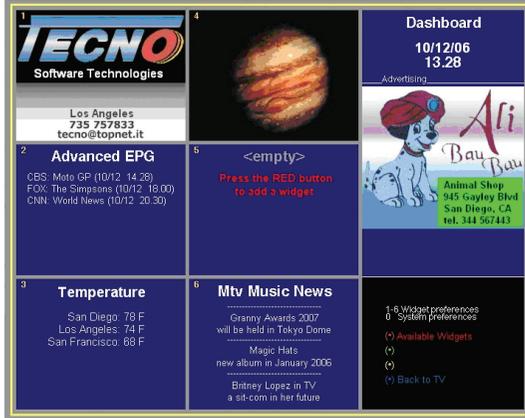


FIGURE 9: A Screenshot of the dashboard application.

Widgets are supposed to periodically retrieve some information from the Web to update their statuses. In our system, each widget is associated to a mobile agent that periodically goes out in the Internet to retrieve fresh information. This information is included in the TV carousel and broadcast.

Since the TV carousel is dynamically updated and data are cyclically broadcast, each Xlet at the client TV side periodically checks for new items available that match its interest. When new relevant data are found in the carousel, the information visualized on the TV screen is updated accordingly.

As a proof-of-concept of our solution, we have devised several widgets that are simultaneously run in our system, namely, *image-viewer*, *rss-viewer*, *weather-forecaster*, *TV advertiser*. A screenshot of our Xlet, as it appears during execution on the Xletview emulator [38], is reported in Figure 9.

More in detail, *image-viewer* is a simple application for the visualization of an image on the screen. The TV carousel periodically includes a new image that matches some specific user's preference (e.g., the astronomy picture of the day, the preferred daily comic strip, some image took from a certain Web camera) and that is utilized to replace the currently displayed one. To support this application, the mobile agent is simply responsible to retrieve an image from specific Web sites and insert it into the TV carousel (a check for duplicates is performed).

Rss-viewer displays contents encoded as RSS feeds that are acquired from specific news Web sites. Even in this case, the mobile agent is simply responsible to retrieve RSS feeds from a specific set of Web sites and to insert them into the carousel, if not already present.

The *weather-forecaster* application takes care of presenting weather forecasts on the TV screen. Again, this is accomplished through a mobile agent that explores the Web to collect weather forecast information related to the geographical area where the user lives.

Our solution opens the door for a whole set of new applications that involves even commerce and direct participation of users in pushing their contents on air. To this aim, the more interesting widget we developed is certainly represented by the *TV advertiser*. Through this application, users

TABLE 3: Time needed during the content search activity (msec).

Min value	Max value	Average value	Median value
5	49	10.86	12

TABLE 4: Reduction of the HTML document size.

Min value	Max value	Average value
-4.5%	-43.50%	-33.07%

TABLE 5: Transcoding data rate (KB/s).

Min value	Max value	Average value	Median value
31.60	111.52	54.21	47.06

are enabled to dynamically inject their own advertisements into the TV carousel. In essence, users are able to provide contents (or links to contents available on the Web) to their agents. In our case, these contents are related to users' business or general activities.

Then, these advertisements can be seen by other TV watchers that instruct their mobile agents to collect specific kinds of advertisement and uploads them on the TV carousel. Therefore, advertisement messages included in the carousel by some user are presented to other users that are exactly interested in that particular product/service. To this aim, the Xlet running on the STB of each of these potential customers filters data present in the TV carousel and provides the user with only pertaining contents. Clearly, this approach promotes a new form of advertisement "from the bottom," according to which users can be engaged in new forms of business (or just communication) involving contents that do not need to be provided by the IDTV broadcaster.

The service model adopted by the *TV advertiser* application can be easily extended to create a wide gamma of new services. As an exemplar case, think for instance to online auction systems on TV. The user's agent could be configured to follow a given item in an auction and periodically report updates on its status while competing in the auction.

To assess the viability of our solutions, we conducted several experiments. In particular, since Web content searching represents a major issue from a performances view point, the goal of our experiments was that of verifying whether agents could be able to efficiently retrieve contents from the Web.

For the sake of conciseness, we report here a summary of performance results obtained using the weather forecaster widget (obviously, similar results were obtained using also other widgets). In this case, the mobile agent size was of about 12 KB. As times needed to migrate agents depend only on the size of the agent and on factors external to our system (e.g., host configuration, external load balancing mechanisms, network latency, etc.), we focus here only on the time spent by mobile agents at a given host, during the content search process.

Table 3 shows the minimum, maximum, average, and median values of the time that mobile agents spent at a given host during our experiments. Considering that the time needed to move an agent to a host to another on the

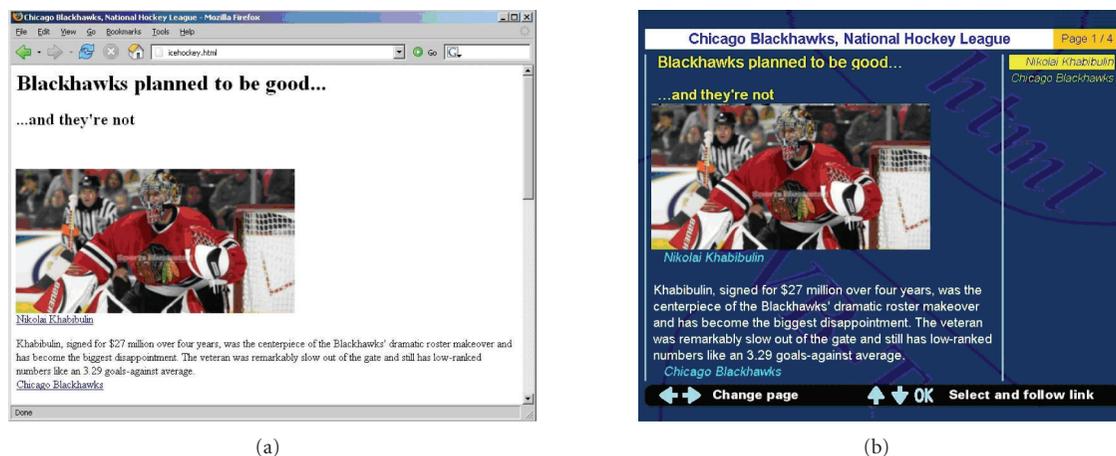


FIGURE 10: Web page visualization on the Web (leftmost screenshot) and on a TV set (rightmost screenshot).

Internet may take some dozens of milliseconds based on the network latencies between those hosts, we can conclude that the overhead imposed by our system is negligible. This confirms the efficacy of our approach.

4.2. News from the Web

The second application we implemented on the DVB-T system is concerned with presenting news retrieved from a Web site on the TV screen.

We carried out experimental tests on tens of different Web pages. Depending on the selected HTML document, during the conversion process different size reduction percentages were experienced, ranging from 4.5% to 43.5% (see Table 4). On average, a 33% of reduction in the size of converted Web pages was obtained. As previously mentioned, this is a positive effect due to the automatic removal of tags that could not be displayed on a TV screen and would hence be useless in this context. Since DVB is intended to transmit also in narrowband scenarios (i.e., mobile transmissions with only a few hundreds of Kbits available for data [4, 5]), diminishing the size of Web pages up to 30% can be of great help to ameliorate system performances. In particular, since carousels obtained during these tests were comprised only of images, text, and bytecode, the resulting average size amounted to 771 KB.

During our preliminary tests, we noticed a transcoding data rate from about 32 KB/s to about 112 KB/s, depending on the contents, as reported in Table 5. In particular, in the table we show the minimum, maximum, average, and median values of the data rate obtained during the transcoding activities of our experiments. However, it is worth noticing that such a transcoding process is performed just once, before insertion of contents into the carousel. Consequently, performances of the transcoding system do not influence final content retrieval time experienced by users. Indeed, to complete a transmission cycle from the request to the final visualization on the TV screen, only about 2.5 seconds were required, on average.

In order to permit also a subjective evaluation of our solution, we report in Figure 10 an example of Web page fetched from the Web (leftmost part of the figure) and then converted and presented on a TV screen (rightmost part of the figure) by means of our system. As anticipated, Web pages displayed on the TV screen present a dark background color with shiny, large fonts to improve the visualization of contents. Web pages have been split into multiple TV screenshots that can be browsed through the TV remote control. Furthermore, links to other Web pages are included in a separate frame on the top-right corner of the screen and can be navigated by resorting to the arrow key buttons of the TV remote control.

Summarizing, these tests confirmed the ability of our solution in dynamically transcoding Web contents before inserting them into the carousel and in optimizing the presentation of these contents over TV screens. With respect to other proposals, our solution has the main advantage of permitting autonomous fetching and adaptation of contents from the Web to be inserted into the TV broadcast system without any need for manual reconfigurations.

5. CONCLUSION

Bringing Web contents to IDTV systems represents a new opportunity that enables TV users to enjoy new interactive services while watching TV. It may be envisaged that new Web-inspired applications for entertainment will be developed on IDTV platforms as soon as viable technological solutions will be available. Unfortunately, the current state of the art still presents several unresolved issues that impedes a proficient combination of the Web and IDTV able to spin a shared world of opportunities.

To this aim, in this work we have addressed the two main macro-problems in this context: collecting information on the Web based on user's preferences and appropriately present retrieved Web contents on the user's TV screen. We have hence deployed a new system architecture which finally enables TV watchers to characterize the information

broadcast to their IDTV sets. As a main contribution, our system narrows the disconnect between the Web and IDTV worlds with the aim of merging them together. We claim that our solution represents an important step forward that may inject new fuel in IDTV and complete the ongoing metamorphosis from TV, as it is currently known, toward the novel concept of personal-TV.

Our architecture exploits the MHP standard for DVB and is composed by four main components:

- (i) a user interface to facilitate users in expressing their preferences about contents in a specification format;
- (ii) mobile agents dynamically generated from users' preferences to retrieve Web documents on users' behalf;
- (iii) an automatic transcoding service able to convert Web contents into a format suitable for IDTV presentation;
- (iv) a DVB-J (or Xlet) browser application broadcast via the carousel and locally executed on each STB to properly manage and visualize received transcoded contents through the IDTV screen and remote control.

Furthermore, we have demonstrated the feasibility and qualities of our system by developing two exemplar case applications (a TV dashboard and a Web news presenter), and experimenting with them on a real DVB-T system.

Last but not least, a note worth of mention is that almost all responsive TV technologies now comprise new standards for receiving and enjoying broadcast TV contents over mobile, handheld devices (e.g., DVB-H) [2, 3, 14]. Therefore, our interpretation of the personal-TV concept also embodies a new way to carry tailored Web contents to nomadic users by exploiting broadband IDTV communication networks.

ACKNOWLEDGMENTS

The authors feel indebted toward CINECA for their support during the experimental evaluation of our system with a real DVB-T environment. The authors are also very grateful to J. Andrich and A. Michetti for their fundamental help during the implementation of their first system prototype. This work is financially supported by the Italian M.I.U.R. and R.E.R. under the MOMA, DAMASCO, and SWIMM initiatives.

REFERENCES

- [1] A. Carmichael, M. Rice, and D. Sloan, "Digital interactive television in the UK: is the opportunity for 'Inclusivity' being missed?" in *Proceedings of the Accessible Design in the Digital World Conference (ADDW '05)*, pp. 1–9, Dundee, UK, August 2005.
- [2] R. Pulles and P. Sasno, "A set top box combining MHP and MPEG-4 interactivity," in *Proceedings of the 2nd European Union Symposium on Ambient Intelligence (EUSAI '04)*, pp. 31–34, Eindhoven, The Netherlands, November 2004.
- [3] R. Filder, *Mediamorphosis: Understanding New Media*, Pine Forge Press, Thousand Oaks, Calif, USA, 1997.
- [4] C. Herrero, P. Cesar, and P. Vuorimaa, "Delivering MHP applications into a real DVB-T network: OtaDigi," in *Proceedings of the 6th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Service (TELSIKS '03)*, vol. 1, pp. 231–234, Nis, Serbia - Montenegro, Yugoslavia, October 2003.
- [5] DVB Consortium: <http://www.dvb.org/>.
- [6] Multimedia Home Platform, <http://www.mhp.org/>.
- [7] ORTIKON Interactive, "Broadcast Manager," 2005, <http://www.ortikon.com/>.
- [8] A. Gil, J. Pazos, C. Lopez, et al., "Surfing the Web on TV: the MHP approach," in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME '02)*, vol. 2, pp. 285–288, Lausanne, Switzerland, August 2002.
- [9] K. Chorianopoulos, "The digital set-top box as a virtual channel provider," in *Proceedings of the Human Factors in Computing Systems Conference (CHI '03)*, pp. 666–667, Fort Lauderdale, Fla, USA, April 2003.
- [10] MsnTV: <http://developer.msntv.com/Overview/default.asp>.
- [11] Open Cable Initiative: <http://www.opencable.com/>.
- [12] ATSC: Advanced Television Systems Committee, <http://www.atsc.org/>.
- [13] "Proceedings of the IPTV Workshop," in *International World Wide Web Conference (WWW '06)*, Edinburgh, UK, May 2006.
- [14] S. Morris and A. Smith-Chaigneau, *Interactive TV Standards: A Guide to MHP, OCAP, and JavaTV*, Focal Press, Oxford, UK, 2005.
- [15] J. Jones, "DVB-MHP/Java TV data transport mechanisms," in *Proceedings of the 40th International Conference on Technology of Object-Oriented Languages and Systems (TOOLS Pacific '02)*, vol. 10, pp. 115–121, Sydney, Australia, February 2002.
- [16] S. Arumugam, A. Helal, and A. Nalla, "aZIMAS: web mobile agent system," in *Proceedings of the 6th International Conference on Mobile Agents*, vol. 2535 of *Lecture Notes in Computer Science*, pp. 1–15, Springer, Barcelona, Spain, October 2002.
- [17] M. Lopez-Nores, A. Fernandez-Vilas, R. P. Diaz-Redondo, et al., "A mixed XML-JavaBeans approach to developing T-learning applications for the multimedia home platform," in *Proceedings of the 1st International Workshop on Multimedia Interactive Protocols and Systems (MIPS '03)*, vol. 2899 of *Lecture Notes in Computer Science*, pp. 376–387, Napoli, Italy, November 2003.
- [18] N. Sawhney, D. Balcom, and I. Smith, "Authoring and navigating video in space and time," *IEEE Multimedia*, vol. 4, no. 4, pp. 30–39, 1997.
- [19] B. Cardoso, F. de Carvalho, L. Carvalho, et al., "Hyperlinked video with moving objects in digital television," in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME '05)*, pp. 486–489, Amsterdam, The Netherlands, July 2005.
- [20] S. Agamanolis and V. M. Bove Jr., "Viper: a framework for responsive television," *IEEE Multimedia*, vol. 10, no. 3, pp. 88–98, 2003.
- [21] K. Baker, "Jules Verne project: realization of reactive media using MPEG4 on MHP," in *Proceedings of IEEE International Conference on Consumer Electronics (ICCE '06)*, pp. 439–440, Las Vegas, Nev, USA, January 2006.
- [22] J. Masthoff, "Group modeling: selecting a sequence of television items to suit a group of viewers," in *User Modeling and User-Adapted Interaction*, vol. 14, pp. 37–85, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2004.
- [23] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *Proceedings of the International Conference of the Special Interest Group on Data Communication (SIGCOMM '01)*, pp. 161–172, San Diego, Calif, USA, August 2001.

- [24] NIONEX GmbH, “Pontegra Browser,” 2004, <http://www.nionex.de/index.jsp>.
- [25] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, “Chord: a scalable peer-to-peer lookup service for Internet applications,” in *Proceedings of the International Conference of the Special Interest Group on Data Communication (SIGCOMM '01)*, pp. 149–160, San Diego, Calif, USA, August 2001.
- [26] A. Rowstron and P. Druschel, “Pastry: scalable, distributed object location and routing for large-scale peer-to-peer systems,” in *Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware '01)*, pp. 329–350, Heidelberg, Germany, November 2001.
- [27] B. Zhao, J. Kubiawicz, and A. Joseph, “Tapestry: an infrastructure for fault-tolerant wide-area location routing,” Tech. Rep. UCB/CSD-01-1141, Computer Science Division, University of California, Berkeley, Calif, USA, April 2001.
- [28] Telecom Italia Lab, Java Agent Development framework, <http://jade.tilab.com/>.
- [29] Foundation for Intelligent Physical Agents, FIPA Agent Management Specification, October 2000, <http://www.fipa.org/>.
- [30] DAML Query Language: <http://www.daml.org/2002/08/dql/dql>.
- [31] JTidy Project: <http://sourceforge.net/projects/jtidy>.
- [32] D. Pantieri, S. Mirri, and P. Salomoni, “RMob: transcoding rich multimedia contents through web services,” in *Proceedings of the 2nd IEEE International Workshop on Networking Issues in Multimedia Entertainment (NIME '06)*, Paris, France, June 2006.
- [33] S. Ferretti and M. Roccetti, “MHP meets the web: bringing web contents to digital TV for interactive entertainment,” in *Proceedings of the 8th IEEE International Symposium on Multimedia (ISM '06)*, pp. 169–176, San Diego, Calif, USA, December 2006.
- [34] S. Ferretti, M. Roccetti, and J. Andrich, “Living the TV revolution: unite MHP to the web or face IDTV irrelevance,” in *Proceedings of the 15th International Conference on World Wide Web (WWW '06)*, pp. 899–900, Edinburgh, UK, May 2006.
- [35] Cineca Web site: <http://www.cineca.it/>.
- [36] A. Marcus, “Dashboards in your future,” *Interactions*, vol. 13, no. 1, pp. 48–60, 2006.
- [37] Apple Computers, Dashboard <http://www.apple.com/macosex/features/dashboard/>.
- [38] Xletview emulator, <http://sourceforge.net/projects/xletview/>.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

