INTEGRATION BETWEEN DIGITAL TERRESTRIAL TELEVISION AND INTERNET BY MEANS OF A DVB-MHP WEB BROWSER

Irene Amerini, Roberto Caldelli, Rudy Becarelli, Francesco Filippini
MICC (Media Integration Communication Center), Univ. of Florence, Viale Morgagni 65, 50134 Florence, Italy

Giovanni Ballocca, Roberto Borri
CSP, Turin, Italy

Keywords: Digital TV, XHTML, Web browser, DVB-MHP, Set-top-box, Java.

Abstract: The process of digital TV convergence, expected in Europe to the end of 2012, makes desirable to develop a system that can provide access to the Internet offering web content on a television screen. In this paper we present *WebClimb*, a web browser that would pursue an effective integration of Digital Terrestrial Television (DTT) and Internet in the DVB-MHP platform. *WebClimb* is a Java-based web browser and enables users to browse the web interacting with an easy to use Graphical User Interface (GUI), driven by a TV remote control. We explain the system architecture of this novel web browser and we evaluate the operating modes of a *WebClimb* prototype, over commercial digital terrestrial set-top-boxes.

1 INTRODUCTION

The digitization and convergence between broadcasting, telecommunications and the Internet are bringing deep and rapid changes to broadcasting environment. These changes introduce significant opportunities for broadcasting consumers in term of choice and access to the content.

The broadcasting infrastructure, with the advent of digital terrestrial television (DTT) in the ’90s, is enabled to provide access to telematic services and distribute interactive applications to communities not yet reached by broadband services (e.g. people in rural areas). All the tools and applications designed to enrich the user experience in the DTT suddenly become a factor enabling a first step in the reduction of the digital divide, at least helping to bridge the gap separating users groups (due, for example to economical, cultural or age reasons) from the access to information and communication technologies.

When digital TV convergence will be completed (in Europe the completion of the transition from analog to digital signal has been set to the end of 2012) a few hundred of millions of digital terrestrial television terminals will be present in the European households, so it would be desirable to develop a multifunction platform that can provide access to both interactive TV and also to Internet services. In this paper we demonstrate the feasibility of Internet and Digital Terrestrial Television integration proposing a web browser prototype which is able to access web content toward TV.

There are many standard to manage interactive and digital terrestrial television technology and in our work we focus on DVB (Digital Video Broadcasting) standard. This standard is used in Europe, Asia, Australia, part of Latin America and in some country of Africa. In other country others standard exist: OpenCable Application Platform (OCAP) and the ATSC’s Advanced Common Application Platform (ACAP) for the United States, Integrated Services Digital Broadcasting (ISDB) is active in Japan, finally DMB (Digital Multimedia Broadcasting) is used in China. In 2000, the DVB consortium completed the standardization of an open standard platform for interactive digital TV and multimedia services called the Multimedia Home Platform (MHP). MHP defines a generic interface between interactive digital applications and the terminals where these applications run. DVB has adopted Java as the interoperable application format for the MHP, developing a version called DVB-J that includes the core of the standard Java language and provides extensions appropriate to the broadcast TV environment. These extensions include a suite of APIs that contains most of the Java TV API,

Amerini I., Caldelli R., Becarelli R., Filippini F., Ballocca G. and Borri R.
INTEGRATION BETWEEN DIGITAL TERRESTRIAL TELEVISION AND INTERNET BY MEANS OF A DVB-MHP WEB BROWSER.
DOI: 10.5220/0001837003230328
In *Proceedings of the Fifth International Conference on Web Information Systems and Technologies (WEBIST 2009)*, page
ISBN: 978-989-8111-81-4

HAVi User Interface, DAVIC (Digital Audio Video Council) APIs and DVB APIs. DVB also enclosed a new application model, a security framework, and a broadcast data transport system for the delivery of MHP applications (also known as “Xlets”). A Xlet can be either resident in the STB or downloaded from an object carousel of a TV channel.

Two MHP specification exists: MHP 1.0.x (DVB, 2003) and MHP 1.1.x (DVB, 2005). The MHP 1.0.x included the definition of two profiles; the first, the Enhanced Broadcast profile is the baseline profile of the MHP. It is intended for use in broadcast only systems where there is a unidirectional flow of data to the receiver. The second profile is called the Interactive Broadcast profile and provides support for a bi-directional data channel using the return channel systems currently defined by DVB. The MHP 1.1 specification includes the definition of a new profile called the Internet Access profile and a new optional content format called DVB-HTML (Gil et al., 2002). The Internet Access profile is intended to provide access to the most common types of Internet services, as currently found on the Internet i.e. WWW and e-mail. Today set-top-boxes on the market give support only for MHP 1.0, so there are no boxes that support the Internet Access Profile. A possible solution is to allow the user to download from the TV object carousel a web browser as a DVB-J application.

In this paper such a solution is presented providing the Internet access functionality on DVB-MHP platform. We develop a DVB-J application, MHP 1.0.x compliant, available for all the STBs today on the market or already previously acquired. This application, named *WebClimb*, permits to access the Internet through the use of the return channel provided by the Interactive Broadcast Profile. *WebClimb* could be embedded on the set-top-box itself or downloaded from the broadcast TV carousel similarly to an usual MHP application.

The paper is organized as it follows. After reviewing some related works in Section 2, we present in Section 3 the proposed system architecture. Finally the application’s prototype is evaluated and the results are presented in Section 4, conclusions are provided in Section 5.

2 RELATED WORKS

This section explores the related works concerning web browser for digital terrestrial television.

Some DVB-MHP commercial applications exist, named Interactive TV platforms, that manage XML pages and act as a browser.

Example of these are: Pontegra HTML browser developed from the Nionex company, Goonie Browser (NordCom Interactive), Ace Browser (Ortikon Interactive), Sofia Backstage Browser Platform (Sofia Digital), Evo Browser (Espial Escape) and Yambo provided by CINECA. Such platforms were born mainly for the development of MHP services by acting as an engine to render applications written by using standard well-known languages such as HTML or XHTML and avoiding to deal with Java programming language (as explained in Section 1). Limitations to these system are that some of them are not broadcast application; in fact they are only embedded in STB and the browser is not an add-on application on the channel. Others applications don’t present a GUI interface and some of them display a subset of DVB-HTML format or non-standard XML pages.

In (Vuorimaa et al., 2002) a Java based XML browser named X-Smiles is presented, intended both for desktop use and embedded network devices, like palmtop, mobile and TV, to support multimedia services. There is an attempt for porting a limited version of X-Smiles on a MHP compliant digital television set-top-boxes but the current state of this attempt is unknown.

In (Cho and Park, 2005) the authors design an integrated web browser for digital home networking and digital data broadcasting based on ACAP standard instead of DVB standard. In (Ferretti et al., 2007), the authors introduce a system to access the WWW that requires the usage of a server-side application gateway, acting as a web-spider, to collect a predefined set of web contents and reformatting them to fit the constraints of the TV environment. This approach is useful in the production and management of the so called “walled garden services”, where the broadcaster or service operator is able to control and validate the contents to be accessed by the user but can become extremely cumbersome in the access to the open Internet.

So these issues influenced our thinking in developing *WebClimb* that is, as we said in Section 1, a DVB-J application, MHP 1.0.x compliant, and can be downloaded by the broadcast TV object carousel or, alternatively, can be embedded in a STB, and locally executed on each STB. It permits to properly manage web contents, without a server-side conversion, and visualize them on a TV screen, by means of a Graphic User Interface (GUI). *WebClimb* is a basically XHTML browser and assures a fine visualization for XHTML web page but it can also manage HTML pages composed by XHTML tag (W3C, 2000).

In particular during *WebClimb* design, specific care has been dedicated to consider the limitations im-

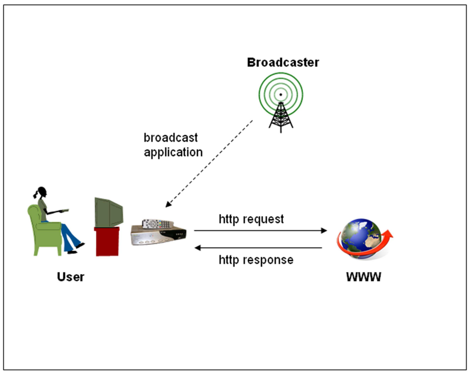


Figure 1: An use case.

posed by the TV set: low screen resolution and reduced interactivity in order to develop an intuitive graphical user interface.

3 SYSTEM ARCHITECTURE

In Figure 1 an use case of the whole process is depicted: a TV broadcaster transmits the *WebClimb* application multiplexed on its TV signal to all the receiving STBs.
An user, by selecting that specific TV channel supporting such an Internet service, can access to the list of applications provided by the channel and choose *WebClimb* to navigate the web via the return channel (e.g. telephone line) and the remote control unit of the STB. For instance, if the user, while looking at a football match, wants to know information about a certain football player, he can switch on *WebClimb* application and, comfortably sitting on his armchair, he can access to the net for searching.
WebClimb browser system architecture is a stack composed by three layers:

- Networking
- Rendering/Layout
- GUI (Graphical User Interface)

Each layer is described in detail hereafter.

3.1 Networking

The *Networking* layer is a module to handle network communication and it is in charge of managing HTTP/HTTPS connection over return channel. In detail, these layer takes as input an URL (Uniform Resource Locator) or URI (Uniform Resource Identifier) and outputs the content of the specified URL passing through the following steps: URL Resolver, Ethernet/PPP Resolver, HTTPClient. The first one is a module that resolves URL and URI and returns a stream that allows the client to send and receive data.
The Ethernet/PPP Resolver hides the type of data link protocol (Ethernet or PPP) used for the return channel connection to the higher level, and handles the session management connection through the DVB API.
The third module is the HTTPClient which implements the application level protocol HTTP. HTTPClient, a third party library made available by Apache (http://hc.apache.org/httpclient-3.x), is employed since support for HTTP is not mandatory in DVB-MHP platform, as specified in (DVB, 2003).

3.2 Rendering/Layout

In Figure 2 the steps composing *Rendering/Layout* layer is displayed.

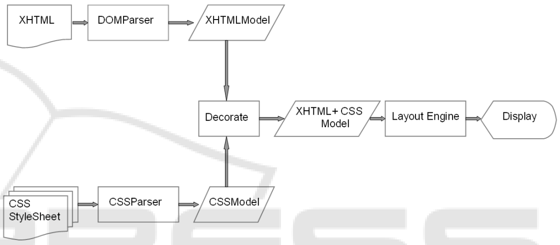
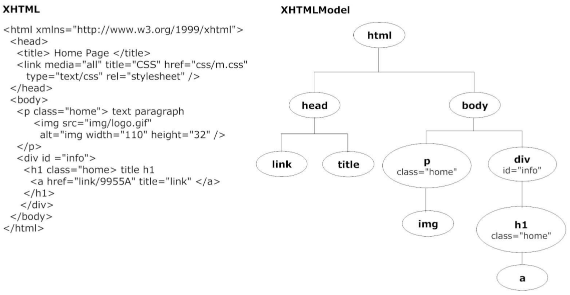


Figure 2: The Rendering/Layout layer flow representation.

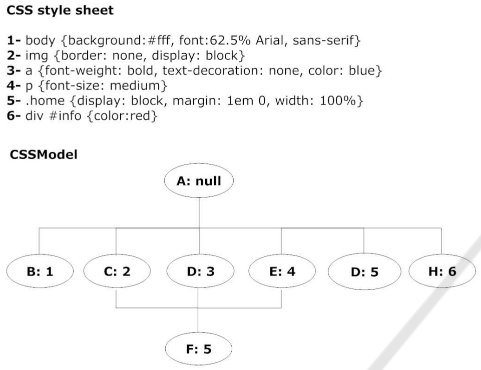
The first step consists in the document parsing. The input data streams coming from the lower layer are parsed and converted in a easily handled tree data structure (see the flowchart branches in Figure 2 called *DOMParser* and *CSSParser*).
Two kinds of input are handled: the web page in XHTML or HTML format and the CSS style sheets linked to the web page. Then for our purpose, there is the need of a parser that take as input XHTML or CSS formats and outputs a Document Object Model (DOM). JTidy (http://jtidy.sourceforge.net) has been chosen to parse XHTML format thanks to his small code dimension in terms of memory occupation and to his capacity of manage non-standard XML like HTML.
The *XHTMLModel* in Figure 3 is created through one-to-one association with a DOM model obtained through the use of JTidy parser. Also SACParser (Simple API for CSS, http://www.w3.org/Style/CSS/SAC), a third party CSS parser library, has been select to generate a Document Object Model style tree called *CSSModel* (Figure 4).
After that, the data model (*XHTMLModel*) is decorated with CSS information through the assignment



XHTML

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title> Home Page </title>
    <link media="all" title="CSS" href="css/m.css"
       type="text/css" rel="stylesheet" />
  </head>
  <body>
    <p class="home"> text paragraph
         <img src="img/logo.gif"
          alt="img width="110" height="32" />
    </p>
    <div id ="info">
      <h1 class="home> title h1
        <a href="link/9955A" title="link" </a>
      </h1>
     </div>
  </body>
</html>
```

Figure 3: XHTML web page example and on the right the *XHTMLModel*.



CSS style sheet

1- body {background:#fff, font:62.5% Arial, sans-serif}
2- img {border: none, display: block}
3- a {font-weight: bold, text-decoration: none, color: blue}
4- p {font-size: medium}
5- .home {display: block, margin: 1em 0, width: 100%}
6- div #info {color:red}

Figure 4: CSS style sheet linked to XHTML and the *CSSModel*.

algorithm till to obtain a *XHTML+CSS Model* (Figure 5). The assignment algorithm links CSS selector to elements in the *XHTML Model* and associates CSS style rule composed by property's value according to cascading (weight-origin of a style sheet and specificity of a selector) and inheritance rules defined in the Cascading Style Sheet level 2 Specification (W3C, 1998).

After XHTML data parsing has been performed and data model has been created, it is necessary to display the data on the screen so the core of the web browser architecture, the layout engine, is built. A layout engine, or rendering engine, is a module that takes web content (such as XHTML, image files, etc.) and format information (such as CSS) (the *XHTML+CSS Model* in Figure 5), and displays the formatted content on the screen, following the visual formatting model specification described in CSS Style Sheet Specification (W3C, 1998). All the elements in the *XHTML+CSS Model* know which are the right positions (coordinate points) to fill in the main window on the screen (viewport) due to the fact that are *inline* or *block* elements following the normal flow positioning scheme. Furthermore each element also knows its own dimensions obtained from dimensions of the inner block that composes a containing block. For example the tag *a* is an *inline* element, and so the rectangular box associated is laid horizontally, after the sibling element box; the tag *div* instead is a *block* element so the box is laid out the previous box vertically. All the boxes are attached to the containing block inside the viewport. WebClimb's viewport resolution is 640 × 480 pixel, due to the low resolution of TV screen, so when the viewport is smaller then the document's initial containing block, the browser offers a scrolling mechanism.

Additionally all elements in the *XHTML+CSS Model* are responsible for rendering themselves drawing appropriate widgets, using the *Viewport GUI Toolkit*, based on AWT and HAVi User Interface API, implemented to satisfy graphical browser's need.

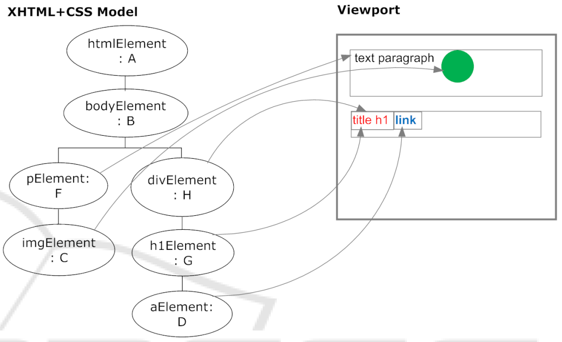


Figure 5: The *XHTML+CSS Model* and its visualization on the viewport.

3.3 Graphical User Interface

Focusing on usability and yielding the user interface as intuitive as possible, we have kept in mind that the TV remote control is the only device that has to be utilized to control the system at client side. Consequently the implementation of all functions that are usually provided by a mouse (pointing, clicking, dragging) have to be replaced by actions on the remote control buttons.

In particular, for text editing, a virtual keyboard interface has been devised, according to the criteria described in (Ambiente Digitale, 2005). The navigation pointer on the keyboard is discrete (key-to-key). The user navigates the virtual keyboard on the TV screen using the arrows keys of the interactive keypad and selects the desired alpha-numeric character pressing the OK button. Once he has selected the character, this appears on the text box area.

The *WebClimb* GUI (see Figure 6) has two navigation levels: the first concerning application user interface and the second concerning the navigation inside the content area (viewport) through which the user can move throughout a web document. The users can switch from one to the other with the pressure of the

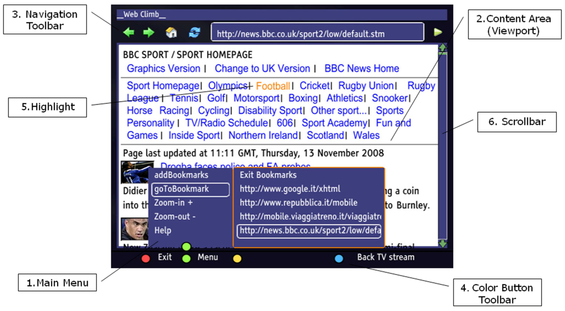

3. Navigation Toolbar
2. Content Area (Viewport)
5. Highlight
6. Scrollbar
1. Main Menu
4. Color Button Toolbar
Web Climb
http://news.bbc.co.uk/sport2/low/default.stm
BBC SPORT / SPORT HOMEPAGE
Graphics Version | Change to UK Version | BBC News Home
Sport Homepage| Olympics| Football| Cricket| Rugby Union| Rugby League| Tennis| Golf| Motorsport| Boxing| Athletics| Snooker| Horse Racing| Cycling| Disability Sport| Other sport...| Sports Personality| TV/Radio Schedule| 606| Sport Academy| Fun and Games| Inside Sport| Northern Ireland| Scotland| Wales
Page last updated at 11:11 GMT, Thursday, 13 November 2008
Drogba faces police and FA probes
Didier
into th
ng a coin
to Burnley.
addBookmarks
goToBookmark
Zoom-in +
Zoom-out -
Help
Exit Bookmarks
http://www.google.it/xhtml
http://www.repubblica.it/mobile
http://mobile.viaggiatreno.it/viaggiatr
http://news.bbc.co.uk/sport2/low/defa
Exit
Menu
Back TV stream

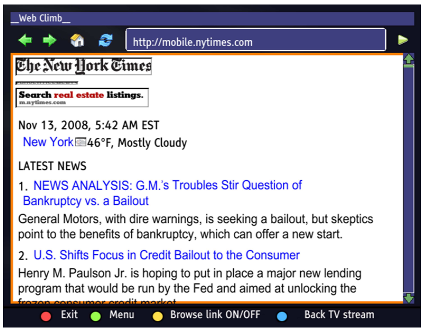
Figure 6: Digital television GUI: 1. Main Menu; 2. Content area (Viewport); 3. Navigation Toolbar; 4. Color Button Toolbar (the yellow button enables link navigation within the viewport); 5. Highlight; 6. Scrollbar

yellow key button of the remote control (see Figure 6 "Browse link ON/OFF").

The focus point of the first level of the GUI's components is controlled with the arrow buttons of the remote control and selection is made with OK button. As shown in Figure 6 the top of the GUI contains the navigation toolbar which shows the current URL and home-page, back and forward buttons as it usually happens in a common browser interface. In the center of the GUI there is the viewport. Dragging function through viewport's scrollbar is performed through the arrow up/down key on the remote control. At the bottom of the GUI there are the color buttons in horizontal position in the same color sequence as in the remote control to satisfied the criteria of effectiveness and usability (Ambiente Digitale, 2005). Some functionalities must be present in all digital terrestrial television applications such as exit the application, go back to TV stream and a help for the user. All these and other functionality are associated at some color buttons in *WebClimb*: *red button* exits the application, *green button* activates the menu, *yellow button* enables switching of the two navigation levels (as previously explained) and *blue button* allows to go back to the TV stream.

Inside the viewport (second navigation level) some GUI components are focusable elements (link, text input, radio button, check box etc.); the user can pass through each focusable element by means of the up-down arrow of the remote control. With the OK button one element is chosen; for example if a XHTML *form* text input is chosen a virtual keyboard for text editing appear. If the OK button is pressed again the the inserted text is sent according as POST or GET method of the tag *form*. The control remains inside the viewport until yellow button is pressed again.

Finally the minimum font sizing (18pt), which is required in digital terrestrial application to avoid graphical flickering effect on the screen has been respected.



Nov 13, 2008, 5:42 AM EST

New York 46°F, Mostly Cloudy

LATEST NEWS

1. NEWS ANALYSIS: G.M.'s Troubles Stir Question of Bankruptcy vs. a Bailout

General Motors, with dire warnings, is seeking a bailout, but skeptics point to the benefits of bankruptcy, which can offer a new start.

2. U.S. Shifts Focus in Credit Bailout to the Consumer

Henry M. Paulson Jr. is hoping to put in place a major new lending program that would be run by the Fed and aimed at unlocking the frozen consumer credit market

Figure 7: *WebClimb* output for the New York Times mobile version web site

The GUI provides other useful features such as the opportunity of adding a web page to bookmarks and the zoom-in/zoom-out font size as a fit-to-width property that automatically reformats and re-sizes content to the viewport.

4 PROTOTYPE AND APPLICATION

We have implemented a prototype of *WebClimb* browser following the Model View Controller (MVC) architectural pattern (Gamma et al., 1994). In our application, the Model is represented by *Facade-Model* class and the main View is the *MBScrollContainer* class that represents the viewport on the television screen (as described in the previous section and shown in Figure 6). The Controller is an object that, through the *controllerLogic()* method, set a new URL to the *FacadeModel*; then a stream is opened and through the use of the parser the data model is created. Once the model is created the data model is visited through the Visitor design pattern; whenever each element of the model is visited a *draw()* method is performed and, if necessary, proper widgets are pictured.

The *WebClimb* prototype performance is evaluated in term of execution on commercial STBs. *WebClimb* behavior has been verified on pages of disparate categories (from XHTML web page designed for PC web browser to mobile XHTML web page) giving positive results generally. In Figure 7 a rendering example is presented. *WebClimb* shows properly web page's content information and has a clear visualization, due to right handle of character encoding and also due to the arrangement of the graphical elements on the screen.

WebClimb has been evaluated on a real DVB-T system, downloading it on commercial STBs by means of a TV broadcasting platform. To test *WebClimb*'s compatibility, the prototype has been verified on the following STBs: DiPro DP-001, ADB X75-T, Elsag IDTV 1000, Telesystem TS7.0DT and Humax DTT-3500. The first and the second are two ADB's different models of STBs. ADB is one of the main producer of STBs, with over 10 million digital set-top-boxes sold worldwide, so this test was particularly fundamental to check WebClimb compatibility. The main differences among these STBs are the memory, which ranges from 8 MByte memory to 64 Mbyte, and the Internet connection that in some case take place with a modem connection and in others with ethernet and modem connection.

From this experiment rises up that *WebClimb* browser, with its 900K byte footprint, needs low memory requirements for execution and works on all STBs tested, both for ethernet and modem connection. *WebClimb* shows the same result in term of graphical elements visualization on TV screen among different STBs. So the test performed on commercial STBs prove the effectiveness of our prototype demonstrating its efficiency on different model and brand.

5 CONCLUSIONS

In this paper we presented the *WebClimb* browser, a Java-based DVB-MHP web browser for digital terrestrial television STBs. The system model architecture has been described and then evaluated in order to verify its performance. We developed a browser that can be broadcast via a TV carousel or embedded on a STB and locally executed; it collects web contents and displays them without asking for reformatting of such a content from the server side.

Experimental tests over commercial STBs have given positive results especially in prospect of a real usage of the system. The *WebClimb* browser, as well, can function as *Interactive TV platform* enabled broadcasters and network operators to create content for enhanced and Interactive Television using standard web tools.

We are working on compare *WebClimb* functionalities with others DVB-MHP web browsers today on the market, like those described in Section 2. In the near future *WebClimb* will be optimized in term of dimension and an additional positioning scheme of visual formatting model will be implemented (floats and absolute positioning).

ACKNOWLEDGEMENTS

The authors would like to thank Tuscany Region (Italy) for supporting the DTT Competence Center which has carried out such a research.

REFERENCES

Ambiente Digitale, F. U. B. (2005). *[Italian Version] Raccomandazioni per le interfacce dei servizi interattivi della televisione digitale*.

Cho, S. and Park, J.-S. (2005). Design of an integrated web browser for digital tvs. *Advanced Communication Technology, 2005, ICACT 2005. The 7th International Conference on*, 2:1089–1092.

DVB (2003). *DVB Multimedia Home Platform (MHP) Specification 1.0.3, ETSI*.

DVB (2005). *DVB Multimedia Home Platform (MHP) Specification 1.1.2, ETSI*.

Ferretti, S., Roccetti, M., and Palazzi, C. (2007). Web content and adaptation for idtv: One step forward in the mediamorphosis process toward personal-tv. *Advances in Multimedia*, 2007:1–13.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*.

Gil, A., Pazos, J., Lopez, C., Lopez, J., Rubio, R., Ramos, M., and Diaz, R. (2002). Surfing the web on tv: the mhp approach. *Multimedia and Expo, 2002. ICME '02. Proceedings. 2002 IEEE International Conference on*, 2:285–288 vol.2.

Vuorimaa, P., Ropponen, T., von Knorring, N., and Honkala, M. (2002). A java based xml browser for consumer devices. In *SAC '02: Proceedings of the 2002 ACM symposium on Applied computing*, pages 1094–1099, New York, NY, USA. ACM.

W3C (1998). *Cascading Style Sheets, level 2 CSS2 Specification*.

W3C (2000). *XHTML 1.0: Extensible HyperText Markup Language*.