OPEN-STANDARDS RICH MEDIA MOBILE PLATFORM & RAPID SERVICE CREATION TOOL

Rhidian Lloyd Cody, Prof. John Cosmas, Dr. Emmanuel Tsekleves
Brunel University, Uxbridge, UB8 3PH, United Kingdom

Abstract

This paper builds upon the work carried out by Brunel University in the field of "Fast Prototyping And Semi-automated User Interface And Application Generation for Converged Broadcast and Cellular Terminals" [1]. This work involved the development of a service creation application for interactive services on mobile devices and methodologies and tools to speed up and deskill the deployment process. This paper aims at further enhancing these tools and presents an enhanced open standards reference platform for mobile digital TV and rich media services. By using a Scalar Vector Graphics (SVG)-driven Java MIDP application (as opposed to bitmapped raster graphics-driven MHP), rich media services can be broadcast to mobile devices running various Java-supported platforms with a user interface scalable to any screen size. Moreover, the Rich Media Mobile Browser is integrated into a service creation tool, therefore enabling rapid testing and deployment of rich mobile media services.

The following sections detail the motivation behind the need for a platform which allows for rich media play-out on mobile devices, along with the rich media mobile viewing application and the tools used to create and test rich media with speed and ease.

Motivation

There is plentiful evidence proving that mobile usage is on the increase. With so many new handsets coming to market, each new mobile device is becoming more of a mobile multimedia computer than just simply a voice communications device [2]. At this early stage in service roll out it, is difficult to predict consumers' views on mobile digital television. However, early reports indicate that mobile TV is likely to be the next commonplace integrated technology in modern mobile devices. The new business models that the mobile television platform can provide are predicted to generate US$6.1bn in revenue by 2011 [3].

With the convergence of broadcast networks and telecommunication networks imminent, the market for rich media on mobile devices broadens. From digital mobile television and interactive teleshopping to multi-participant live gaming and national surveys, rich media is the next logical step forward in mobile digital communications. For these rich media services to gain a good foothold in being a mainstream integration into next generation mobile devices, their development and deployment needs to become a much simpler process. Currently, rich media services are only available on a small percentage of devices using proprietary end-user software. This is mainly due to the complexities involved in service creation which must be carried out by developers who specialise in mobile platform development (e.g. Java MIDP).

Unlike the Multimedia Home Platform (MHP) standard [4] found in set-top boxes for conventional television sets, no all-encompassing standard exists for rich media services within a mobile environment. This is due to the myriad devices in the market including and not limited to PDAs, mobile cell phones, handheld gaming devices, Ultra Mobile PCs (UMPCs) and in-vehicular infotainment systems each with their own operating platforms, power constraints, screen sizes and screen resolutions. Network operators and content providers demand a service deployment system which can be seamlessly integrated across the majority of platforms using open standards.

This paper presents a complete Rich Media Mobile Platform from service creation through to end-user consumption using open, internationally standardised formats.

Mobile Interactive Services Using Open Standards

Mobile Interactive Services can accompany broadcasted audiovisual content or can be standalone services. These services are similar to those currently available using MHP on standard television set-top digiboxes to provide services such as interactive

voting on a live quiz show or ordering a pizza when a particular advertisement is broadcasted. There is currently no standard format of delivering mobile-centric interactive services. Content providers and network operators demand a format which offers rich interactive content and can be viewed by the majority of its customers, irrespective of the viewing device.

As full duplex communications are natively available on mobile devices, this type of interactivity lends itself to the mobile environment very well. Although it can be argued that HTML and similar browser compatible formats already support this kind of interaction, no format currently supports direct rich interaction with the current live TV broadcast feed or allows for full freedom of design.

The Rich Media Mobile Platform introduces a Mobile Interactive Services format which is proposed to give a rich user interface experience to mobile device end-users. The platform defines a *rich* user interface as one incorporating media such as video, sound and scalable graphics alongside conventional images and text. The format enables a move away from the standard page-by-page structure of the Web by providing more compelling and intuitive user interfaces.

The Mobile Interactive Services format uses an enhanced media format based on the XML-based open-standards SVG Tiny 1.1 (SVGT) profile [5]. Owing to its internationally recognised standardisation, use of SVG is fast becoming the universal method of displaying and storing scalar vector graphics. The advantages of using SVG for graphic rendering are clear [6]: As Figure 1 demonstrates, SVG graphics can be scaled to infinity without loss of quality making it ideal for scaling to varying device screen sizes. SVGT also sports smaller file sizes compared with rasterised graphics making it ideal for a data bandwidth-constrained mobile environment. Animated graphics are supported as standard and SVGT also supports inline or linked external media, HTML-style hyperlinking and is based on XML - a vendor-neutral, platform-neutral, universally supported interchange format. Above all SVGT is an open standard which will improve and increase in functionality over time.

Figure 1: *Demonstration of the infinite scalability of vector graphics compared with raster graphics*

Currently, SVG Tiny 1.1 support on mobile devices is limited to static image rendering and animation [5]. Full SVG features such as scripting and gradients are not supported. It is seen as a graphics format as opposed to an interactive media format. The Mobile Interactive Services format enhances SVGT documents to create fully interactive rich media documents.

Through the use of added XML meta data, attributes and SVG Id naming conventions, graphical elements within the XML code of the SVGT document are recognised as interactive user interface components. User interface components and features such as buttons, textfields, menus, hyperlinks, textblocks, video, forms and scrollable pages can be included as part of the SVGT document to provide interactivity. Should the user interface component require further properties that SVGT does not provide, added XML attributes are used with a Mobile Interactive Services XML namespace, which detail the extra required information.

Scalar vector graphics enables the user interface to expand (or indeed shrink) to the size of the mobile terminal's screen. In addition, the Mobile Interactive Services format reformats and repositions certain graphical elements to gain the most efficient use of screen real estate. SVGT graphics can be specified as a background object and will therefore fill the mobile device's screen regardless of screen orientation, for example. As a further example, standard SVGT does not support multiline blocks of text or any text wrapping abilities. The Mobile Interactive Services format overcomes this by automatically reformatting specified "textblocks". In standard SVGT a `<text>` element is used to render a graphic of a series of characters to the screen. To achieve a text paragraph effect, we convert the `<text>` element into a block of multilined text by setting its `id` attribute to `textblock_textblockname`. The Rich Media Mobile Browser will then render the element as a paragraph of wrapped text instead of a single line of characters. Moreover, the textblock will be re-rendered should the device's screen orientation change providing a best-fit efficient use of screen real estate at all times. Standard text properties such as font typeface, size and colour are set in the usual SVGT way.

Actions for user interface components and device keypad button presses are supported in the Mobile Interactive Services format through embedded XML meta data within the SVGT document. Actions can include navigational options such as a change of

scene, opening of a Web hyperlink, opening an external SVG document or starting/stopping animations; retrieval of external data; submissions of form data, or hiding/highlighting other graphics or user interface components. For example, pressing key 5 on the device's keypad could be specified to bring up an on-screen menu.

As depicted in Figure 2, The Mobile Interactive Services format introduces methods of retrieving data externally to fill textual content automatically. As a result, SVGT documents can be created as templates for displaying data which is to be obtained later via an XML information exchange format. For example, the content provider could have an XML document containing sports score data. The Mobile Interactive Service can be assigned to obtain this data to use as its content when viewed. This means the content provider can just provide data and not have to create a whole new Interactive Service each time the score data changes. This significantly increases Interactive Service efficiency by considerably reducing the service development time.

Figure 2: *Mobile Interactive Service external data retrieval*

Using the Rich Media Mobile Interactive Services format, network operators and content providers now have a method of efficiently creating Mobile Interactive Services that not only scale and best-fit render to the mobile device's screen but also provides a consistent look-and-feel on every device they are viewed on, regardless of the viewing device's operating platform or screen size, resolution or orientation. Mobile Interactive Services are viewed by means of the Rich Media Mobile Browser, which is discussed in the next section.

Rich Media Mobile Browser

The Rich Media Mobile Browser is a full Mobile Interactive Services and audiovisual play-out application developed to operate on mobile devices. It enables the user to graphically view TV schedules, watch streamed digital mobile TV via DVB-H (or other methods) and download and use Mobile Interactive Services. Figure 3 shows an example of the Rich Mobile Media Browser in action.

The Rich Media Mobile Browser is developed using the Java 2 Mobile Edition (J2ME) language subset using the Mobile Information Device Profile version 2.0 (MIDP 2.0) [7] allowing it to be installed on any mobile device supporting MIDP applications. MIDP allows audiovisual playback using the Java Mobile Media API (MMAPI) [8] and user interfaces are rendered using the Java M2G API [9] using SVG Tiny profile version 1.1. Java MIDP is currently integrated as standard on over 1.2bn mobile handsets [10] with its latest incarnation available on the vast majority of modern smartphones and PDAs.

Upon opening the browser application the user is presented with a listing of available services to consume via an intuitive scalable SVGT-based user interface. The user interface is navigated using device point-and-click hardware or navigation keys. The presented services are the result of parsing a retrieved XML-based Electronic Service Guide (ESG) [11] via a specified URI provided by mobile network operators or content providers. The location of the URI maybe a Web URL, a local IP address obtained from a hardware DVB-H stack via DVB-H datacasting [12], or obtained through an IPTV system. Services within the listing can include traditional digital TV channel broadcasts and radio channel broadcasts, or on-demand services such as Mobile Interactive Services and audio/video clips.

After the user selects the desired service to consume he/she is presented with a synopsis. Along with a description of the selected service, the synopsis details the multiple forms of consumable multimedia available to the service in the form of web links, images, multiple video feeds, multiple audio feeds, Mobile Interactive Services, and so forth. Depending on the type of media selected by the user the Rich Media Mobile Browser commences play-out: audiovisual playback or streaming, Mobile Interactive Service play-out or, if an external link, the browser will call the local device's default Web browser to view the content.

The Rich Media Mobile Browser also features an Interactive Service alert system. If an audiovisual broadcast stream has further media associated with it, the browser will alert the user with a message or on-screen graphic. The message can either be dismissed or acted upon where the user will be presented with a featured Mobile Interactive Service. This feature is especially useful for advertisements. For example, if a movie trailer is shown, a Mobile Interactive Service could broadcast alongside it with movie reviews, local cinemas currently screening the movie and links to the movie's website.

Play-out of Mobile Interactive Services is a key part of the Rich Media Mobile Browser. The browser can render and provide interactivity based on the proposed SVGT-based Mobile Interactive Services format by parsing through the retrieved document and identifying user interface elements and performs their associated actions. For example, if the browser discovers a graphical element such as a `<rect>` (a rectangle graphic) has been specified to act as a button, it will obtain the button's required action through its `action` attribute. The `action` value will provide the details of what action to take should the button be clicked on or selected using the device's navigation keys or keypad. The tie-up between the user interface graphics and their actions for interactive services are created using the Rich Media Mobile Platform's Service Creation Tool, which is discussed in the next section.

Figure 3: *The Rich Media Mobile Browser in action showing the Electronic Service Guide listing available services to consume*

Service Creation Tool

Rich media on mobile devices is currently left in the hands of mobile programmers. This often means designers are unable express their ideas and are forced to use standard graphical user interface components, resulting in an ugly, uninviting user interface design and layout.

Presented as part of the Rich Media Mobile Platform is a rapid Mobile Interactive Service development tool. The tool is a workstation-based application developed to significantly speed up the process of Mobile Interactive Service creation. It removes the need for skilled mobile application developers by letting the designer quickly and easily add functionality to their graphical user interface designs.

The tool is intentionally designed to be a "second stage" element within a Mobile Interactive Service production cycle. It enables the designer to use an SVG industry-standard authoring tool of their choice to initially design the user interface. Such authoring tool examples include Oracle's Mobile Designer, Adobe's Creative Suite package and Corel's design packages. Using the design authoring tools, rich animation and attractive user interfaces can be created with no restrictions on the designer. User interface components can be placed anywhere on the screen and can have any desired look-and-feel. Once the project has been visually designed it can be exported as a standard SVGT document ready for manipulation.

The created SVGT document is then imported into the Rich Media Mobile Platform's Mobile Interactive Service Creation Tool to begin adding interactivity.

The tool initially exposes every graphical element within the imported SVGT document to the user by means of a Document Object Model tree alongside a preview of the graphically rendered document. From here the designer can choose which graphical elements to act as user interface components. Through use of an intuitive drag-and-drop interface, SVGT graphical elements quickly become user interface components. From there each component can be assigned actions such as jumping to a new scene in the document or opening a Web hyperlink. A makeshift on-screen keypad is also available to tie actions to certain device key presses.

The tool creates the interactivity by means of SVG Id naming conventions and added XML attributes, which are automatically being inserted into the document on-the-fly as the designer drags-and-drops.

Once the interactivity has been tied to the graphical elements the user can simply draft preview the newly created Mobile Interactive Service with the click of a button. The preview mode essentially opens a workstation emulation version of the Rich Media Mobile Browser and also enables the user to view the Interactive Service in a range of screen orientations, dimensions and resolutions to visualise how it will appear on various mobile devices. Finally, it can be sent to a real or emulated mobile device for actual device testing.

For example, to convert a `<circle>` element into a button, the user would first select the item from the presented Document Object Model tree. A properties box then displays the attributes of the circle: attributes such as x and y coordinates and radius data are shown. With the click of a button, the user can choose for the circle to act as a button. An action dialogue box opens ready for the user to assign actions to the button. An intuitive list of possible actions is also presented for ease of access. By selecting a `gotoscene` action, the user can specify that this circle button now jumps to a pre-determined animation scene in the SVGT document. Behind the scenes, the tool is generating all the extra XML code to add into the SVGT document. The circle's `Id` attribute would be set to `menuItem_button` to specify it is now an interactive button item and its `action` attribute set to `gotoscene_scene4`, for example. If the user then chose a `<rect>` item from the presented Document Object Model tree and set the element to be a `background` object, this particular `<rect>` graphic will now be rendered full screen, regardless of the viewing device's resolution or screen orientation. With a click of the tool's preview button the user can see the rendered interface with the newly added interactivity along with ability to change the displayed rendering's resolution and orientation. This Mobile Interactive Service document could now be saved and viewed in the Rich Media Mobile Browser on a physical mobile device.

Figure 4 visually represents a typical Mobile Interactive Service development scenario from initial design to end-user consumption.

This rapid "what you see is what you get" approach to Mobile Interactive Service prototyping , creation and deployment, along with the design freedom and an open-standards format, is proposed to enable mobile telecommunications network operators and content providers to get new services into customers' hands with swiftness and simplicity.

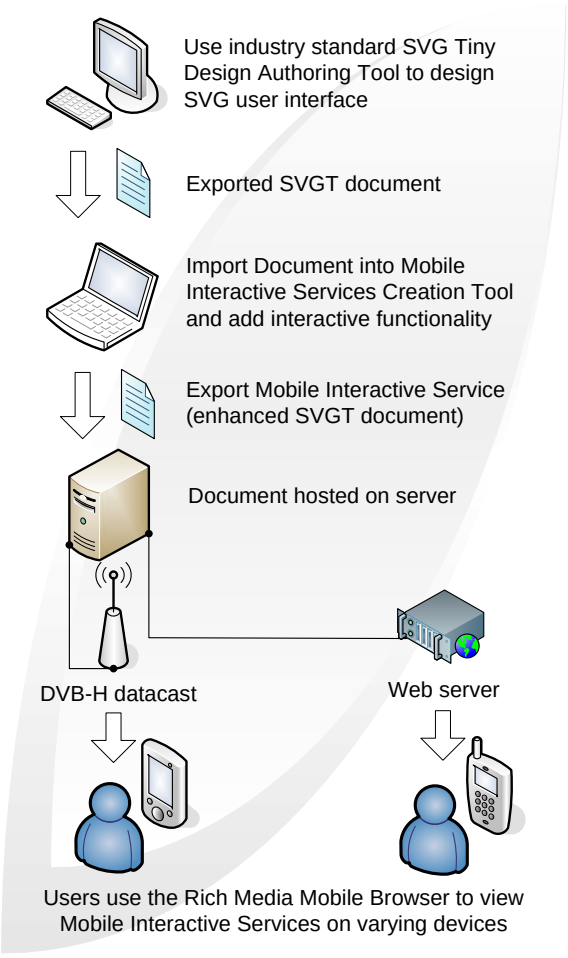

Figure 4: *Mobile Interactive Service Development Scenario from initial user interface design to end-user consumption*

Conclusion

Digital Television and rich media have exploded into our homes and across the internet over the past half-decade making it one of the biggest types of media in the world. With recent advances in broadcast and IP technology media service providers can now enhance users' experience of digital media even further so they are in touch with the consumer wherever they are. However, building rich mobile media services requires much skill and programming expertise in the field. The Rich Media Mobile Platform is proposed to provide digital TV content providers and mobile telecommunications network operators a rapid and convenient, open-standards method of getting interactive services into the hands of their connected mobile customers, opening up a wealth of new

business and revenue-generating opportunities. This paper has not only presented a fast-prototyping and emulation tool but a full workable Rich Media Mobile Browser for real-world application. The developments represent an advance in the interactive services domain for the rapid development of mobile digital TV services for users on the move.

References

[1] Emmanuel Tsekleves, Brunel University (2006), Fast Prototyping And Semi-automated User Interface And Application Generation for Converged Broadcast and Cellular Terminals

[2] BBC News (August 2004), Boom Times Ahead For Smartphones
http://news.bbc.co.uk/1/hi/technology/3602718.stm

[3] BBC News (May 2007), Mobile TV Predicted To Be A Hit
http://news.bbc.co.uk/1/hi/technology/6639249.stm

[4] Multimedia Home Platform (MHP)
http://www.mhp.org/

[5] W3C (2003), Mobile SVG Profiles: SVG Tiny and SVG Basic
http://www.w3.org/TR/SVGMobile/

[6] W3C (August 2000), Features of SVG
http://www.w3.org/TR/SVG-access/

[7] Sun Developer Network, Mobile Information Device Profile (MIDP); JSR 37, JSR 118
http://java.sun.com/products/midp/

[8] Sun Developer Network, Mobile Media API (MMAPI); JSR 135
http://java.sun.com/products/mmapi/

[9] Java Community Process, JSR 226: Scalable 2D Vector Graphics API for J2ME
http://jcp.org/en/jsr/detail?id=226

[10] Sun Developer Network, Java ME Technology - Powering your Devices Everywhere
http://java.sun.com/javame/technology/index.jsp

[11] ETSI TS 102 470 v1.1.1 (2006), Digital Video Broadcasting (DVB); IP Datacast over DVB-H: Program Specific Information (PSI)/Service Information (SI)
available at http://www.dvb-h.org/

[12] ETSI TS 102 471 v1.2.1 (2006), Digital Video Broadcasting (DVB); IP Datacast over DVB-H: Electronic Service Guide (ESG)
available at http://www.dvb-h.org/