Mobile Rich Media Technologies: Current Status and Future Directions

Jaeyeon Song[1] and Byoung-Dai Lee[2]
[1] Multimedia Global Standard Group, Samsung Electronics
Suwon, Korea
[e-mail: jy_song@samsung.com]
[2] Department of Computer Science, Kyonggi University
Suwon, Korea
[e-mail: blee@kgu.ac.kr]
*Corresponding author: Byoung-Dai Lee

Received June 1, 2010; revised July 20, 2010; revised December 17, 2010;
accepted January 29, 2011; published December 28, 2011

Abstract

Demand for rich media services is rapidly increasing and, as a result, several technologies are competing to deliver rich media services to mobile, resource-constrained devices. In this paper, we explore the existing rich media technologies and analyze their key features. In addition, in order to accommodate the new requirements introduced by emerging service models, such as service convergence, these technologies must evolve from their current status. We examine how the community intends to achieve this.

Keywords: Convergence, rich media, service platform

DOI: 10.3837/tiis.2011.02.007

1. Introduction

Recently, resource constraints imposed on mobile devices have become more relaxed. For instance, new mobile phones are equipped with larger color displays, advanced graphics rendering platforms and various forms of convenient human interfaces, such as haptics. User are thus demanding a similar real-time rich media experience from new services and applications that they get from a PC.

Rich media is a dynamic, interactive collection of various types of media content such as audiovisual materials, graphics, text, and hyperlinks. It provides the means for defining the temporal and spatial organization of the media components as well as their possible interactions. In particular, interactivity, ranging from user-device interactivity to client-server interactivity, is the essence of rich media. **Fig. 1** demonstrates examples of rich media services currently available. The interactive mobile TV service [5][13], illustrated in **Fig. 1-(a)**, provides access to traditional TV content and to additional interactive services aligned with the TV program, such as online voting in a live show. **Fig. 1-(b)** shows a rich media map service [22] that provides real-time traffic and street-plan information. This service allows users to acquire more topographical information via a button menu. In addition, clicking on different regions of the map plays a corresponding informative video. Another example is the user interface of the device enhanced with rich media [13]. This provides better presentation with fluid interaction, and enables the device resources and the connected resources to be blended seamlessly into a coherent interface. As an example of this class, **Fig. 1-(c)** shows a phone-screen with a weather popup.

Several technologies are competing to achieve the vision of rich media services to mobile, resource-constrained devices. To bring this vision to fruition, these technologies must evolve so the requirements for the next generation of mobile services are fulfilled. High demand for effective user experiences and the support of service-level convergence are critical requirements in the mobile application and service domain. Users prioritize intuitive usage and accessibility to services when evaluating the effectiveness of the services. On the move, for instance, users would prefer to obtain the information they need with one click rather than following a menu tree of several branches. In addition, the service interfaces must leverage online experiences, as users are accustomed to quality Web interfaces in their daily business

Some day in the far off future, when women and men from Earth are living and working on Mars, they may well look back on Jan. 14, 2004, as the first step of their amazing journey.

(a) (b) (c)

Fig. 1. Examples of rich media services: **(a)** interactive mobile TV service, **(b)** rich media map service, and **(c)** device user interface.

and life. Convergence is one of the frequently quoted buzzwords in today's IT industry. The evident benefit of converged services is that the users can seamlessly enjoy the same services on their mobile Web, mobile TV and home network, which in turn may generate new revenues for the service providers.

Current rich media technologies provide a more flexible, dynamic, interactive, and rich user experience than traditional user interface applications, which are usually implemented using program languages best suited to the target devices. However, most of the technologies focus on a single, independent service domain, so they are insufficient to be applicable to converged environments. For instance, when rich media content is communicated and shared among heterogeneous devices, individual devices must be able to present it adaptively based on their resource capabilities. This adaptability is one of the important features that current rich media technologies lack.

In this paper, we explore the existing technologies for enabling rich media services and identify their key features in Section 2. We consider only the technologies based on open standards and proprietary solutions, such as Flash, are not covered. In Section 3, we investigate the requirements that will have a direct impact on the evolution of rich media technologies. We summarize by presenting the directions that the next-generation rich media technologies will follow, and the current activities of the development community attempting to achieve it.

2. The Current Rich Media Technology

In rich media, a multimedia presentation is a collection of individual audiovisual content such as still images, audio, video, and, possibily, fonts, presented in a scene description that specifies the organization of the media objects. In particular, a scene description describes four aspects of a presentation [14]:

- how the scene elements (media or graphic) are organized spatially, e.g., the spatial layout of the visual elements;
- how the scene elements are organized temporally, i.e., if and how they are synchronized, when they start or end;
- how to interact with the elements in the scene, e.g., when a user clicks on an image;
- and how the scene changes happen.

A scene description may change by means of animations. The different states of the scene during the whole animation may be deterministic (i.e., known when the animation starts) or not. The former case is illustrated by parametric animations. The latter case is illustrated by, for instance, a server sending modification to the scene on the fly. The sequence of a scene description and its timed modifications is called a scene description stream [10]. Therefore, the application logic or scenarios for the user interface can be entirely defined in the scene description stream and only a generic runtime component (e.g., a rich media engine) is required on the terminal.

The rich media technology is usually composed of three parts: a presentation format, a packaging format, and a transport format (see **Fig. 2**). The presentation format defines the scene description; it must be simple yet efficient to improve delivery and parsing times, as well as storage sizes. The packaging format defines how rich media data is grouped together. This must cope with high latency networks and support synchronization among rich media data with a very low overhead. The transport format defines the delivery mechanisms of the rich media data such as download-and-play, progressive download, streaming, and broadcasting.

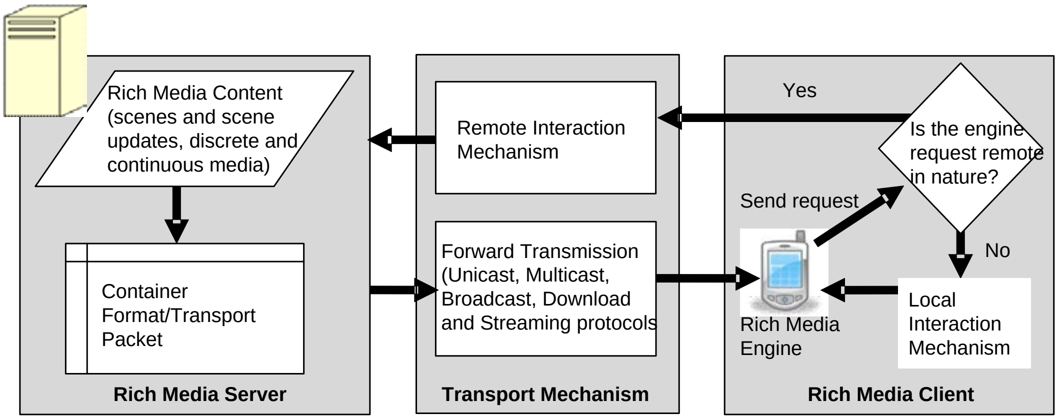


Fig. 2. General architecture of the rich media system.

In what follows, several competing technologies for enabling mobile rich media services are presented and analyzed.

2.1 SVG

SVG (Scalable Vector Graphics) [15], defined by W3C, is a language for describing two-dimensional vector graphics and graphical applications in XML, combined with raster graphics and multimedia. SVG allows for three types of graphic objects: vector graphic shapes (e.g., paths consisting of straight lines and curves), images, and text. Graphical objects can be grouped, styled, transformed, and composited into previously rendered objects. The feature set includes nested transformations, clipping paths, alpha masks, filter effects, and template objects.

SVG supports the ability to change vector graphics over time. SVG content can be animated either by embedding SVG animation elements in SVG content or via scripting. In particular, SVG's animation elements employ the animation features defined in SMIL (Synchronized Multimedia Integration Language) [20], which is an XML markup language for describing multimedia presentation such as timing, animations, visual transition and media embedding, among other things. It is also possible to describe the animation by use of a scripting language (e.g., ECMAScript) that accesses SVG uDOM (micro Document Object Model), which provides complete access to all elements, attributes and properties. A rich set of event handlers such as "onmouseover" and "onclick" can be assigned to any SVG graphical object, thus being able to achieve any kind of animations.

SVG defines simplified profiles for mobile devices, and mobile industries such as 3GPP (the 3rd Generation Partnership Program) and OMA (Open Mobile Alliance) have adopted them for their mobile profiles. For instance, SVG includes two simplified mobile profiles: SVG Tiny (SGVT) for highly restricted mobile devices such as cell phones and SVG Basic (SVGB) for high-level mobile devices such as PDAs. 3GPP has adopted SVGT as the mandatory vector graphics media format for MMS (Multimedia Messaging Service), PSS (Packet-Switched Streaming Service) and IMS (IP Multimedia Subsystem).

Note that both SVG and SMIL rely on the HTML model for content consumption (e.g., download-and-play and progressive download-and-rendering) and they do not specify the streaming of SVG and SMIL content. Therefore these languages are not suitable for fast, dynamic, interactive, and interoperable content.

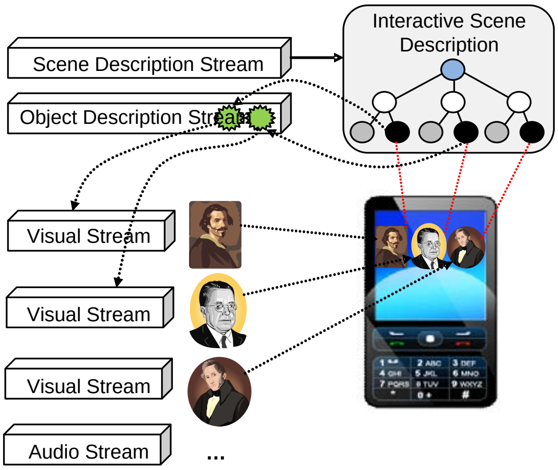


Fig. 3. Various video and audio streams are composed on top of a fixed-background still picture according to a scene description. An additional concept is the object descriptor, which associates sets of elementary streams to audio or visual objects.

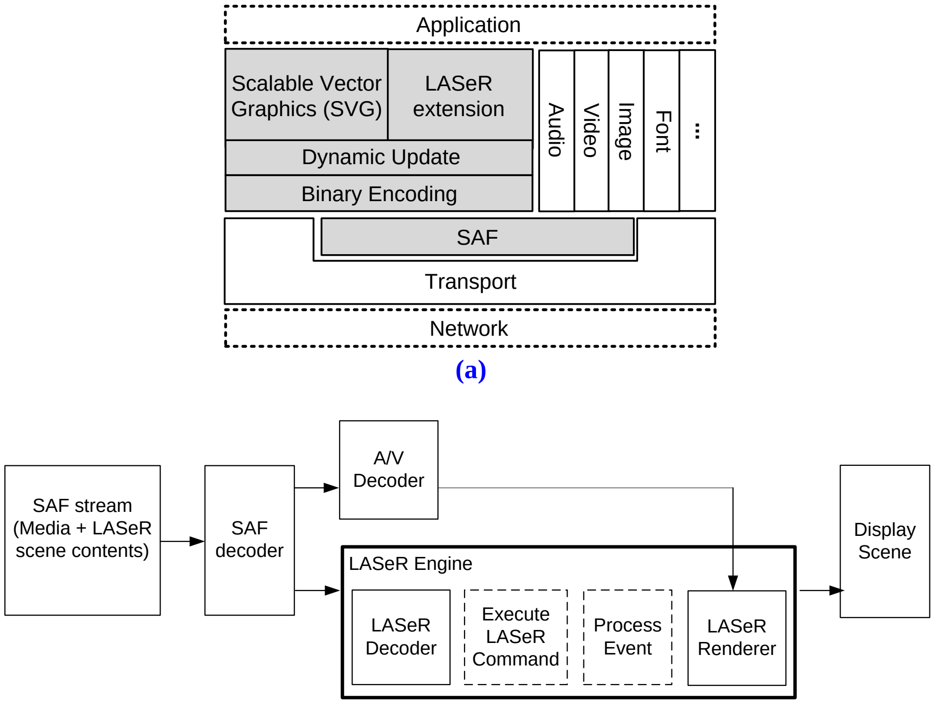
2.2 BIFS

BIFS (Binary Format for Scenes) [8] is a complete framework for encoding scene data in the MPEG-4 standard and it aims at providing a corpus of technology to be used by various types of multimedia services and networks. It integrates rich media content in a unique framework so that it can be seamlessly manipulated by the content authors as well as by the users.

BIFS describes the scene in a scene graph, which is a hierarchical representation of audio, video, and graphical objects, each represented by a BIFS node abstracting the interfaces to those objects. Building on the VRML (Virtual Reality Modeling Language) [11] scene graph, BIFS defines 62 new nodes on top of the 54 nodes defined by VRML. The major extensions include 2D/3D scene composition, special nodes for facial and body animation, extended sound composition and a query node for terminal resources. Compression is the most obvious area where BIFS enhances VRML. BIFS files are stored in compressed binary format, which results in a 10 to 20 times smaller file size than their VRML equivalents. This capability is useful in optimizing content delivery. In addition, BIFS defines several scene description profiles: Basic2D, Simple2D, Core2D, Main2D, Advanced2D, Complete2D, Audio, 3DAudio, and Complete.

BIFS introduces the innovative notion of incremental updates of the scene, enabling streaming of long running scenes. For this purpose, BIFS defines commands that can modify, delete or replace objects in the scene in a timely manner, and the BIFS-Update protocol to send the commands acting on a scene from a server to the client terminal where the scene resides. In particular, in MPEG-4, the audiovisual scene can be split into several elementary streams and control information in the form of object descriptors. This is used to let the receivers know what type of information is contained in each stream. These descriptors provide links between the scene description and the streams of the audiovisual objects. In other words, the presentation itself is a stream that updates the scene graph and relies on a dynamic set of object descriptors, which allow referencing to the actual media streams (see **Fig. 3**).

The spectrum of MPEG-4 end-devices goes from standard computers to mobile devices, through interactive TV sets. For instance, the Core2D profile of BIFS has been adopted for interactive data services in T-DMB (Terrestrial Digital Multimedia Broadcasting) [7] and has



(a)

(b)

Fig. 4. LASeR engine architecture and execution flow: **(a)** LASeR engine architecture and **(b)** LASeR decoder flow.

been deployed on various personal devices with different resource capabilities. However, the processing complexity due to the inherent content and binary encoding makes BIFS inappropriate for mobile devices. Therefore, instead of compromising on the technology's performance, the MPEG forum decided to create a rich media standard for constrained devices.

2.3 LASeR

MPEG-4 Part 20 [10] is the MPEG forum's latest solution for representing and delivering rich media services to resource-constrained devices and it is the most prevalent technology for enabling rich media services. The standard consists of two specifications: LASeR (Lightweight Application Scene Representation), which specifies coded representation of multimedia presentation for rich media services; and SAF (Simple Aggregation Format), which provides aggregation methods to efficiently transport LASeR data together with media data over various delivery channels.

LASeR uses an SVG scene tree at its core. It imports composition primitives from SVG Tiny and SMIL, and uses the SVG rendering model to present the scene tree. It also defines compatible extensions over SVG to allow the development of efficient services in mobile environments. Examples of the extensions include simple axis-aligned rectangular clipping, definition of new events for key press, simple text underlying, and a new timing model. In addition, LASeR defines two profiles—LASeR Mini for mid- and lower-end embedded devices and LASeR Full for higher-end devices—to specify the allowed elements in the profile and the list of their attributes, possibly with restrictions when describing the scenes.

Along with the extensions over SVG, dynamic updates, and binary encoding, as in BIFS, are keys features of LASeR. With a dynamic update mechanism that uses LASeR commands manipulating SVG uDOM, it becomes possible for a remote server to modify the scenes in reactive, smooth and continuous ways. LASeR commands and scene description elements are encoded in a binary format before transmission. Due to the limited processing and storage capabilities of mobile terminals, the design goal of LASeR's binary encoding is to support simple, yet efficient compression in order to improve delivery and parsing times, as well as storage size.

Binary encoded LASeR content together with media is eventually delivered using SAF or other transport mechanisms. SAF defines the binary representation for multiplexing LASeR data together with various media data, thus making one logical SAF stream. As such, it fulfills efficient distribution of rich media content, which would, otherwise, require separate delivery by establishing different connections. Note that, however, LASeR can be used independently from SAF. **Fig. 4** illustrates the LASeR engine architecture. When a terminal receives the SAF stream, the SAF decoder first demultiplexes the stream into media content and scene descriptions. Individual media elements are decoded in corresponding A/V coders, whereas the LASeR engine decodes the scene descriptions, executes LASeR commands, and processes events. Finally, the LASeR renderer composites all scene components, including media content, and present the scene to the user.

Though several functional similarities, including incremental updates of scenes and binary encoding of the scene description, exist between LASeR and BIFS, LASeR has enhanced the end-to-end rich media publication chain more efficiently. This has resulted in ease of content creation, optimized rich media delivery and enhanced rendering on the devices, as its design has specifically targeted mobile devices and constrained networks. Furthermore, as LASeR focuses on 2D scenes only, it has a much simpler object set, leading to much simpler decoding.

LASeR has been adopted for a wide variety of mobile applications and services in different domains, ranging from device user interfaces to mobile broadcasting. For instance, a French mobile operator, SFR (Société Française de Radiotéléphonie), provides on-device portal services that are implemented using LASeR technology. In addition, several mobile operators in Europe, such as T-mobile in Germany, H3G in Italy, and Swisscom in Switzerland, have adopted LASeR for enabling interactive data services that are provided as part of DVB-H (Digital Video Broadcasting-Handheld) [6] based mobile TV services.

2.4 MORE

MORE (Mobile Open Richmedia Environment) [22] is an open suite of W3C, OMA, 3GPP, IETF (Internet Engineering Task Force) technologies combined to meet the requirements for formatting, packaging, compressing, transporting, rendering, and interacting with rich media files and streams. In fact, there are significant similarities between MORE and LASeR with respect to technologies and functionalities. For instance, MORE uses SVG Tiny 1.2 as a basis for its scene presentation format. In addition, dynamic, incremental scene updates through SVG uDOM is supported to achieve a reactive, smooth and continuous service. However, the scene update mechanism in MORE relies on REX (Remote Events for XML) [18]. It is based on a set of requirements that are intended to maintain compatibility with DOM events, declarative in nature, and it integrates well with the WWW architecture and the general requirements of the wireless application environment. As a result, the syntax for the update mechanism is not limited only to SVG but is also extensible to other markup languages. MORE does not support local management of preference data for users/applications and mechanisms to permanently store a small amount of information securely.

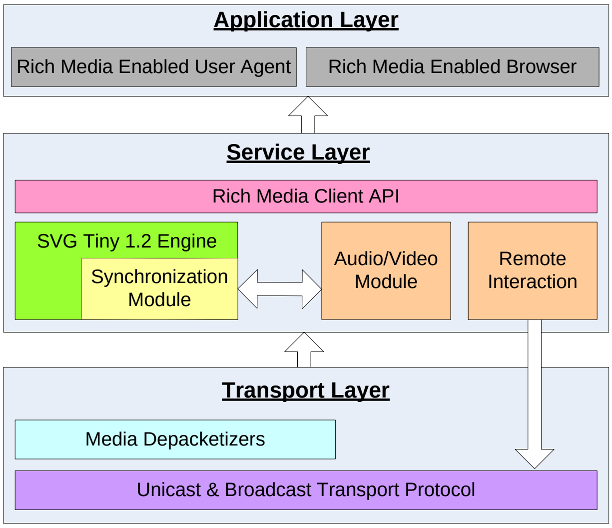


Fig. 5. MORE client architecture.

A typical MORE client, depicted in **Fig. 5**, is a lightweight entity present on the mobile terminal. This is possible due to the fact that it builds on top of existing application enablers such as SVG Tiny 1.2 and XHTML-basic and thereby reuses their associated underlying components such as the XML parser, rendering libraries, media decoders, and compression libraries. The client uses media depacketizers to obtain the different media that constitute the scene and scene updates in the case of real time streaming. The synchronization module helps synchronize the frame rate and timing of continuous media with that of the non-frame based SVG content. The SVG engine, in turn, takes the different media and timing information as input to compose dynamically the rich multimedia presentation. The client is also responsible for transmitting any feedback occurring during interaction.

The primary advantage of MORE over LASeR lies in its strong separation of its components and their interfaces. By enforcing such a separation, it is extensible, allowing it to incorporate the best solutions within the architecture, such as new compression techniques.

MORE is backed up by Nokia and it is much less IPR (Intellectual Property Rights) -encumbered, which may encourage its wide adoption. In particular, MORE, along with 3GPP DIMS (Dynamic and Interactive Multimedia Scenes), was proposed for consideration to meet the requirements defined in OMA-RME (Rich Media Environment). However, to our knowledge, no MORE based services have been deployed and widespread in the mobile environment, except for those in research projects.

2.5 3GPP DIMS and OMA-RME

Since its release, LASeR has received great attention from the mobile industry, due to its simple, yet effective and efficient architectural design. As a result, 3GPP and OMA have adopted LASeR as the technical foundation for their respective rich media standards - 3GPP DIMS [1] and OMA-RME [17]. Furthermore, many aspects of OMA-RME rely on 3GPP DIMS. For instance, OMA-RME supports scene description extensions defined in 3GPP DIMS in addition to SVG Tiny 1.2. They also share the same timing and processing model to handle the time and synchronization of scenes. However, individual standards show differences mainly in packaging and delivery of rich media content. For instance, OMA-RME supports packaging of rich media data into 3GP files [2], whereas 3GPP DIMS streams are

Table 1. Comparison of representative rich media technologies.

	SVG Tiny 1.2	**MPEG-4 BIFS**	**MPEG-4 LASeR**	**MORE**	**3GPP DIMS**	**OMA RME**
Scene Description	Describing 2D vector graphics in XML, combined with raster graphics and multimedia	Based on VRML with extensions	Based on SVGT 1.2 with XML extensions	Based on SVGT 1.2 with XML extensions	Based on SVGT 1.2 with XML extensions	Based on SVGT 1.2 with XML extensions
Animation	SMIL/ Scripting	BIFS-Anim	SMIL/ ECMScript	SMIL/ ECMScript	SMIL/ ECMScript	SMIL/ ECMScript
Open Type font	Not defined	Not defined	Supported	Supported (Optional)	Supported (Optional)	Supported
Dynamic Update	Not defined	Supported (BIFS-Update)	Supported (DOM tree update)	Supported (DOM tree update)	Supported (DOM tree update)	Supported (DOM tree update)
Compression	Not defined	Binary encoding	Binary encoding	GZIP compression	GZIP compression	GZIP compression
Packaging Format	Not defined	Object descriptors	SAF	ISO Base Media File	ISO Base Media File	3GP
Delivery Format	Not defined	IP/RTP/ MPEG-2 TS	RTP/HTTP/ FLUTE [21]	RTP/HTTP/ FLUTE	RTP/HTTP/ FLUTE	RTP/HTTP/ FLUTE
Error Recovery	Not defined	Not defined	Supported	Supported	Supported	Supported
Profiles	Not defined	9 scene description profiles defined	LASeR Mini LASeR Full	Not defined	Mobile profile	Not defined

carried in files of the ISO Base Media File format [9], including 3GP files.

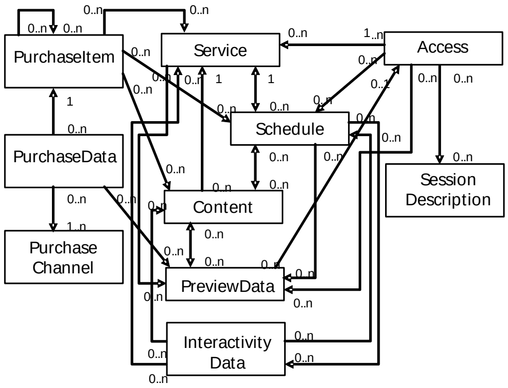
3GPP DIMS and OMA-RME are rich media standards defined by leading mobile industry consortiums. Therefore, it is expected that mobile services based on these technologies will be widespread. For instance, OMA-RME has been adopted for the interactive application framework in ATSC (Advanced Television Systems Committee) Mobile DTV services [4]. Applying this interactive application framework, the broadcaster, service provider, or content provider is able to create and control the presentation aspects of the service.

The primary features of the aforementioned rich media technologies are briefly summarized in **Table 1**. In the next section, we explore critical requirements for next generation mobile services, and suggest future directions to which the current rich media technologies must evolve to accommodate those requirements.

3. The Future of Rich Media Technology

3.1 A Separation between How to Present and What to Present

The approach that rich media technologies have taken to provide interactive, information rich user experiences is to separate “what to present” from “how to present”. Information on how to



(a)

```
<?xml version=”1.0">
<ESGMainBCAST>
  <Service id=”BBCThree” type=”1" version=”1"
ServiceProtection=”0">
    <Name>BBC Three</Name>
  </Service>

  <Content id=”urn:bbc:2890" version=”2">
    <Name>The Ferocious …</Name>
  <Content>

  <Schedule id=”BBCThree_d0e1024" version=”2"
ServiceIDRef=”BBCThree”>
    <ContentIDRef idRef=”urn:bbc:2809">
      <Presentation_Window
Presentation_Start_Time=”3346533600"
Presentation_End_Time=”3346537200"/>
  </Schedule>
</ESGMainBCAST>
```

(b)

Fig. 6. An example of SI: **(a)** the data model of OMA-BCAST ESG (each box represents a well-formed XML document), and **(b)** a snippet of an XML document.

present is commonly called PI (Presentation Information) and scene descriptions correspond to it, as it declaratively specifies how to present various scene components spatially and temporally. SI (Structured Information), on the other hand, contains metadata information describing the scene components and thus specifies what to present. SI is typically organized as a pre-defined texture structure such as XML. A strong separation between PI and SI makes it possible for service providers to control the user experiences of a service on the fly without modifying the devices’ software components that have been deployed in the first place. In particular, the devices are required to deploy only the generic runtime component (e.g., rich media engine) as the presentation logic is defined in separate scene description streams that have rich media composition capabilities on top of the capabilities common to classic multimedia players.

By its nature, SI varies significantly according to services and it can be subject to change over time. Furthermore, as complexities of services increase, it is more likely that individual services use their own syntax, data formats and delivery mechanisms to define, package and transmit SI for flexibility and efficiency reasons. For instance, OMA-BCAST [16], a representative standard for enabling mobile TV services, defines its own syntax for SI, called ESG (Electronic Service Guide), to provide descriptive information for the available services (e.g., titles and the synopses of the services), as well as service acquisition (e.g., IP addresses for audio/video streams). **Fig. 6** illustrates the data model and an example XML document of ESG used in OMA-BCAST.

When it comes to the degree of the separation between SI and PI, the current rich media technologies still need more development because some elements of SI are required to be embedded in scene descriptions. For instance, a LASeR scene description consists of text, graphics, animation, interactivity, and spatial and temporal layout. Therefore, textual content to be displayed must be present within a scene description. As the complexity and the amount of SI of a rich media service increase, the negative impact caused by this requirement becomes more obvious, especially due to the delivery of a huge amount of scene description data. This clearly suggests that in order for the next generation rich media technology to be applicable

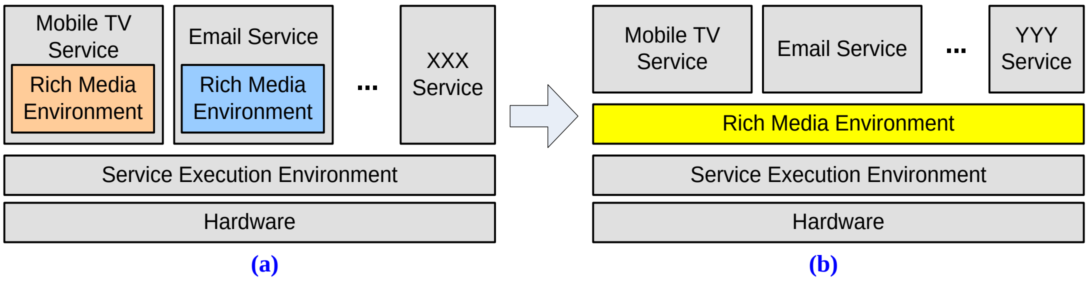


Fig. 7. Extension of rich media environment as a service platform.

over a wide variety of services, it must provide generic, flexible mechanisms to reference SI from PI to consume SI in a consistent manner.

Recent efforts to address the problem can be found in the literature [5][19]. These approaches add advanced features of existing XML technologies, such as XPath and XLink, into the LASeR technology to systematically reference texture content in SI and to provide mechanisms to handle periodic changes of SI. For complex services, however, more sophisticated application logic may be required to acquire pieces of information from SI. In the data model shown in **Fig. 6**, for the channel navigation scene to present the image logos of the individual TV channels, the locations of the logos, which are defined in the “PreviewData” fragment, must be searched using the unique identifiers of the channels that can be found in the “Service” fragment. This process can hardly be modeled and expressed in a declarative way. Therefore, extensive research is still required to handle such complex use cases.

3.2 Rich Media Environments as a Service Platform

The initial scope of the mobile rich media technologies focuses on provisioning of dynamic, interactive, information rich services on mobile, resource-constrained, and optionally connected devices. In particular, they were developed on the basis of the independent service model, in which individual services are independent from one another and thereby there are no interactions among them. However, as more and more services become rich media enabled and multitasking capability is in great demand, we must consider interactions among different rich media services and the underlying service execution environment. For instance, when a user wants to email while watching a TV show on a mobile phone, it would be preferrable that both services appear on the screen of the mobile phone at the same time, through dynamic scene recomposition by the rich media environment, instead of a single service monopolizing all resources.

With the current rich media technology, an instance of the rich media environment is created and integrated on a per service basis, and its primary job is to take care of only presentational aspects of the service (see **Fig. 7-(a)**). In addition, due to the lack of the capability for the rich media environment to provide systematic ways to deliver events from the service exeuction environment, the rich media service ignores other events or the same event is presented differently by individual services, which decrease the degree of consistency in the user experience. Therefore, the next generation rich media environment must be extended as a service platform, an intermediary entity between rich media services and the underlying service execution environment (see **Fig. 7-(b)**). As such, the rich media environment becomes capable of coordinating behaviors of different rich media services in a consistent and controlled manner. Furthermore, interfacing with the underlying service execution environment on behalf of the rich media services assures that device-specific

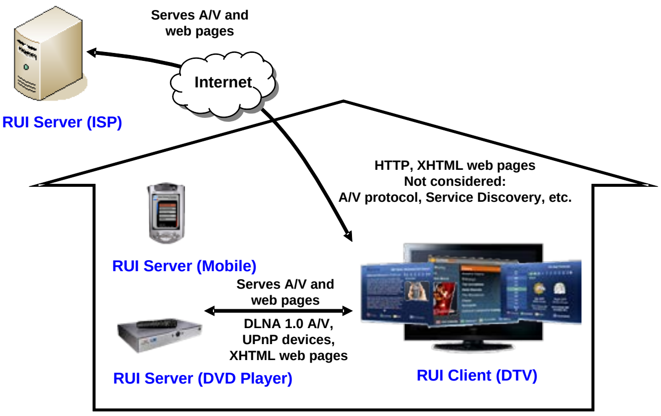


Fig. 8. The scope of Web4CE: the Web4CE framework adopts a number of technologies for both in-home and the Internet that span multiple domains.

capabilities (e.g., voice call, GPS-based navigation) are well integrated into individual services.

Several interesting approaches are imaginable to address these issues. As an approach for controlled presentation of multiple, independent services at the same time, individual services may provide different scene descriptions, each of which is optimized for a specific use. For instance, the mobile TV service can provide two scene descriptions; one for the case where the mobile TV service uses the screen of a mobile phone exclusively, and the other for the case where the screen is shared with an additional service, for example, an email service. Another approach is to allow the rich media environment to select dynamically the best scene composition in consideration of various QoE (Quality of Experience) metrics such as the user perference. Along with this, the next generation rich media technology needs to provide generic, flexible mechanisms to receive events from the underlying service execution environment, forward them to corresponding rich media services, and/or process them internally on behalf of services. Examples of such events include reception of incoming calls and alert of low battery power.

As an example of relevant on-going projects, MPEG-U (MPEG-Rich Media UI) [12] standardizes widget packaging, delivery, representation and communication formats. This standard enables communications among widgets on the same device or different devices, and other applications to better support connected environments in a controlled manner.

3.3 Support of Converged Services

As converged environments, where all devices will truly interoperate and cross the boundaries of different service domains, have emerged as the future service environment, it is envisioned that rich media services will be “communicated” and “shared” among heterogeneous devices belonging to different service domains, which typically have different resource capabilities. For instance, a mobile phone is accessing the rich media enabled TV service, but since there is a big screen TV in the vicinity, the TV is used as output with the user input provided on the mobile phone. In particular, presentation of rich media service on the TV is not a simple rendering of audiovisual content along with graphics received from the mobile phone. Instead, it must be well optimized for the screen size of the TV.

In such contexts, the rich media engines deployed on individual devices will play the role of the central hub for device discovery, resource negotiation and session management, as well as presentation of rich media content. The following shows some of the challenging issues to be addressed for sharing rich media content:

- infrastructure for discovering clients and servers in the network for data exchange (e.g., rich media content and control information);
- mechanisms to deal with session migration for streaming of rich media content;
- mechanisms to exchange capability information between devices, and to adapt the rich media content based on these capabilities;

An example of recent efforts in this area is Web4CE [3], developed by the CEA (Consumer Electronics Association). Web4CE enables rich media services on the converged environment consisting of the home network and Internet services (see **Fig. 8**). It introduces a new concept, called RUI (Remote User Interface), to allow a user interface to be remotely displayed on and controlled by devices or control points other than the one hosting the logic, and a major component of RUI is the CE-HTML profile that uses existing Web technologies (i.e., XHTML, CSS TV profile, AJAX, etc.) to define user interfaces to be rendered on screens with different resolutions and sizes, ranging from HDTV screens to mobile phone displays. Web4CE also supports broadcasting of audiovisual data over Internet and the home network.

As another example of the evolution of rich media technologies, the MPEG forum also comes to the finish of revising the LASeR standard. The amendment will include static and dynamic adaptation for scene presentation by selectively rendering and replacing the scene components based on the resource capabilities of the terminals, such as the display size and available memory. In particular, this ability enables the rich media content to be broadcast and shared by the various types of receiving terminals, thus resulting in significant reduction in bandwidth consumption for data transmission. Otherwise, rich media content targeted for individual terminal types would need to be authored and broadcast separately.

The requirements introduced by the vision of converged services are most complex because various service domains must be involved that have intrinsically different characteristics. Therefore, although several innovative attempts have been proposed to address the related issues, we believe that extensive research still needs to be conducted in such areas as scene adaptation, adaptive streaming, and seamless handover.

4. Concluding Remark

In this paper, we explored representative open standards for enabling rich media services. The current rich media technologies aim at providing efficient and effective ways of representing and delivering rich media services in various service environments, and thus they address architectural and functional areas which are generic enough to be common to many services, such as the scene description, packaging, and delivery of rich media data.

Converged environments have emerged as the future service environment, therefore the current rich media technologies must advance accordingly. In this paper, we present three important requirements that the next generation rich media technology must fulfill: 1) a strong separation between what to present and how to present; 2) the rich media environment as a service platform to provide controlled and consistent presentation of rich media services; and 3) rich media content being shared and communicated across heterogeneous devices with different resource capability. We analyzed the progress being made in addressing each of the requirements.

References

[1] 3GPP TS 26.142, "Technical Specification Group Services and System Aspects; Dynamic and Interactive Multimedia Scenes (Release 7)," *3rd Generation Partnership Project*, Sep. 2007.

[2] 3GPP TS 26.244, "Transparent End-to-End Packet Switched Streaming Services (PSS): 3GPP file format (3GP)," *3rd Generation Partnership Project*, Sep. 2007.

[3] ANSI CEA-2014, "Web-based Protocol and Framework for Remote User Interface on UPnP Networks and the Internet (Web4CE)," *Consumer Electronics Association*, June 2006.

[4] ATSC A/153 Part 1:2009, "ATSC-Mobile DTV Standard, Part 1 – ATSC Mobile Digital Television System," *Advanced Television Standard Committee*, Oct. 2009.

[5] B. Lee, J. Song and Y. Nam, "Converged Mobile TV Services Supporting Rich Media in Cellular and DVB-H Systems," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 3, pp. 1091-1097, Aug. 2008. Article (CrossRef Link)

[6] ETSI EN 302 304 V1.1.1, "Digital Video Broadcasting (DVB); Transmission System for Hand-held Terminals (DVB-H)," *European Telecommunications Standards Institute*, Nov. 2004.

[7] ETSI TS 102 427 V1.1.1, "Digital Audio Broadcasting (DAB); Data Broadcasting – MPEG-2 TS Streaming," *European Telecommunications Standards Institute*, July 2005.

[8] ISO/IEC 14496-11, "Information Technology – Coding of Audio-Visual Objects – Part 11: Scene Description and Application Engine," *Int'l Organization for Standardization/Int'l Electrotechnical Commission*, 2005.

[9] ISO/IEC 14496-12, "Information Technology – Coding of Audio-Visual Objects – Part 12: ISO Base Media File Format," *Int'l Organization for Standardization/Int'l Electrotechnical Commission*, 2004.

[10] ISO/IEC 14496-20, "Information Technology – Coding of Audio-Visual Objects – Part 20: Lightweight Application Scene Representation (LASeR) and Simple Aggregation Format (SAF)," *Int'l Organization for Standardization/Int'l Electrotechnical Commission*, 2004.

[11] ISO/IEC 14772-1, "Information Technology – Computer Graphics and Image Processing – The Virtual Reality Modeling Language – Part 1: Fundamental Specification and UTF-8 Encoding," *Int'l Organization for Standardization/Int'l Electrotechnical Commission*, 1997.

[12] ISO/IEC JTC CD 23007-1, "Information Technology – Rich Media User Interface – Part 1: Widgets," *Int'l Organization for Standardization/Int'l Electrotechnical Commission*, 2009.

[13] J. Dufourd, "LASeR: The Lightweight Rich Media Representation Standard," *IEEE Signaling Processing Magazine*, vol. 25, no. 6, pp. 164-168, Nov. 2008. Article (CrossRef Link)

[14] J. Dufourd, O. Avaro and C. Concolato, "An MPEG Standard for Rich Media Services," *IEEE Multimedia*, vol. 12, no. 4, pp. 60-68, Oct. 2005. Article (CrossRef Link)

[15] "Mobile SVG Profiles: SVG Tiny and SVG Basic," *World Wide Web Recommendation*, Jan. 2003. Article (CrossRef Link).

[16] OMA-TS-BCAST_DVB_Adaptation_V1.0, "Broadcast Distribution System Adaptation – IPDC over DVB-H," *Open Mobile Alliance*, Sep. 2007.

[17] OMA-TS-RME_V1_0-20071017-D, "Rich Media Environment Technical Specification," *Open Mobile Alliance*, Oct. 2007.

[18] REX, "Remote Event for XML," Article (CrossRef Link).

[19] S. Song, I. Lee, H. Lee, J. Hong and J. Cha, "Rich Media Application Using MPEG-4 LASeR PMSI Technique," in *Proc. of 3rd European Symposium on Computer Modeling and Simulation*, pp. 546-551, Nov. 2009. Article (CrossRef Link)

[20] "Synchronized Multimedia Integration Language (SMIL 3.0)," *World Wide Web Recommendation*, Dec. 2008. Article (CrossRef Link).

[21] T. Paila, "FLUTE – File Delivery over Unidirectional Transport," *Internet Engineering Task Force*, Oct. 2004. Article (CrossRef Link).

[22] V. Setlur, T. Capin, S. Chitturi, R. Vedantham, and M. Ingrassia, "MORE: A Mobile Open Rich Media Environment," in *Proc. of 2006 IEEE Int. Conf. on Multimedia and Expo*, pp. 2029-2031, July 2006. Article (CrossRef Link)

Jaeyeon Song received her B.S., M.S. and Ph.D. degrees in Electronic Engineering from Hong-ik University, Korea in 1995, 1997 and 2001 respectively. She joined Samsung Electronics, Co., Ltd as a senior engineer in 2001. She has participated in many projects for mobile broadcast service standardization. Her research interests include mobile IPTV and interactive rich media service. She is co-chair of LASeR ad-hoc group of MPEG committee.

Byoung-Dai Lee is an assistant professor at the department of computer science, Kyonggi University. He received his B.S. and M.S. degrees in Computer Science from Yonsei University, Korea in 1996 and 1998 respectively. He received his Ph.D. degree in Computer Science and Engineering from University of Minnesota, twin cities, USA in 2003. Before he joined the Kyonggi University, He worked at Samsung Electronics, Co., Ltd as a senior engineer from 2003 to 2010. During the period, he has participated in many projects related to mobile broadcast systems. His research interests include open mobile platform, mobile security and mobile multimedia broadcast.