

Flexible Scene Composition Method for Companion Screen Service based MMT-CI

Minjeong Song

Seoul National University of Science and Technology
Graduate School of Nano IT Design Fusion
Seoul, Republic of Korea
mi0_0ng@naver.com

Miri Park

Seoul National University of Science and Technology
Graduate School of Nano IT Design Fusion
Seoul, Republic of Korea
alf18514@naver.com

Sung-geun Yoo

Seoul National University of Science and Technology
Graduate School of Nano IT Design Fusion
Seoul, Republic of Korea
orcogre@gmail.com

Sangil Park

Seoul National University of Science and Technology
Graduate School of Nano IT Design Fusion
Seoul, Republic of Korea
sangilparkmail@gmail.com

Abstract— ATSC 3.0, which is the standard of UHD broadcasting, is defined, and a new web-based broadcasting system is being developed which leads to the active participation of viewers from the traditional broadcasting. Therefore, in this paper, we study about the scene composition method of viewer - selectable type applying the companion screen service among the IBB services which is one of the ATSC 3.0 standards and developed a web application.

Keywords—*Companion Screen Service; HTML5; Websocket; MMT; ATSC 3.0*

I. INTRODUCTION

Recently, UHD broadcasting has been standardized in South Korea, and ATSC 3.0 has become a hot topic. Unlike existing traditional broadcasting, interactive services by viewers have been added in ATSC 3.0. The IBB(Integrated Broadcast Broadband) service is a standard for viewing real-time broadcasting and receiving rich additional services via the network and broadband network [1]. In order to provide this service, some companies develop various personalized services and draw the active participation of viewers.

However, in the case of a multi-channel service in which contents of a plurality of channels can be viewed on one screen, the limit that a viewer can't freely adjust the size and position of a previously designated content There are points.

In order to solve this problem, in this paper, we studied a system that enables viewers to specify the layout of multiple contents that the viewer wants to freely through the second device and can see this on the TV screen. In addition, applying a scene construction method using CI(Composition Information) of MMT(MPEG Media Transport) which is one of ATSC 3.0 standards, implementing a JavaScript-based web application, and experimenting with a viewer selection scene composition was carried out.

II. RELATED WORKS

A. Companion Screen Service

Companion screen service is a technology corresponding to the fourth part of the IBB standard. Companion screen service is also called a second screen service or N-screen service and means a broadcast service that can be additionally used with various devices such as a smartphone and tablet together with a TV [2]. The second device using the supplementary service, that is, the companion device is driven through communication with the TV. In both devices, each application is driven and the communication between the applications is done via a Websocket. The companion application is basically created by the HTML5 web-based application and can be executed on the web browser. When the TV and the companion device can communicate with each other, each device can share the service and content information and can exchange the necessary information [3].

B. Websocket

Websocket is a technology for realizing efficient two-way communication between a server and a client. Websocket maintains socket connection between server and client and is capable of bidirectional communication and data transfer at any time. Since data can be transmitted and received freely by using a socket, data can be exchanged more easily than the related method of request-response [4]. These Web socket communication methods are used by being applied to the connection and data exchange method of the application between the TV and the companion device by the companion screen service[1].

C. MMT CI

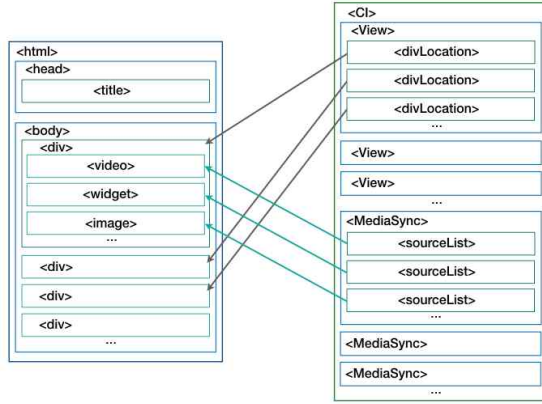


Fig 1. Correlation between html and CI documents of MMT CI

MMT is the standard of MPEG's next-generation broadcasting system that enables IP-based media delivery service. MMT defined MMT CI for screen composition and content selection. This MMT CI has been defined to use HTML 5 and can utilize screen splitting and time event, and various UI and dynamic screen compositions are possible[5].

In the <body> of the HTML document on the left side of Fig. 1, after defining the character of Asset and defining elements such as <video>, <audio> <image>, etc. as content sources in the child element of <div> Describe the spatial information of the media element as an attribute. The CI article on the right side of Figure 1 was created in the XML schema. In the CI document, the <View> element is described first. The <divLocation> element plays a role of drawing on the screen by referring to the content data value of a single MMT asset. The <MediaSync> element describes the start time and ends time of the source by referring to the time information of the media element in the HTML5 with the XML based CI document[6].

III. PROPOSED VIEWER SELECT LAYOUT METHOD

The purpose of this paper is to increase the participation of viewers watching TV. To this end, we have developed a system that can adjust the position and size of contents freely. The system consists of primary devices and applications, and companion devices and applications where viewers can configure their own layouts. .

A. Service Scenario

The scenario of the proposed service is shown in Fig 2. The viewer is watching Video 1 through the TV. During the viewing, the web application, Customized Layout System, is run through a companion device connected to the TV. The viewer selects the desired content from the list of the application and creates a customized layout. After the layout is completed, the viewer presses the send button. Using this button, the layout created by the viewer is transmitted to the TV via websocket. Finally, the TV receiving the layout is set based on the position information and the time information of

the content by manipulating the composition of the scene.

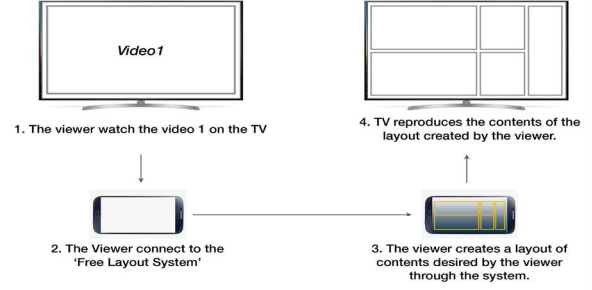


Fig 2. Scenario of proposal service

B. Companion Application

The companion application is a Javascript-based web application that can be run in a web browser. In this paper, the companion application is named Customized Layout Application. The UI of the proposed service's application is shown in Figure 3. The viewer selects contents desired by the viewer through the contents list and move the contents to the empty canvas space by dragging and dropping. The moved contents can freely adjust the size and change the position of the viewer. It also consists of a system that can transmit the layout through a web socket when the submit button is pressed after the layout is completed.

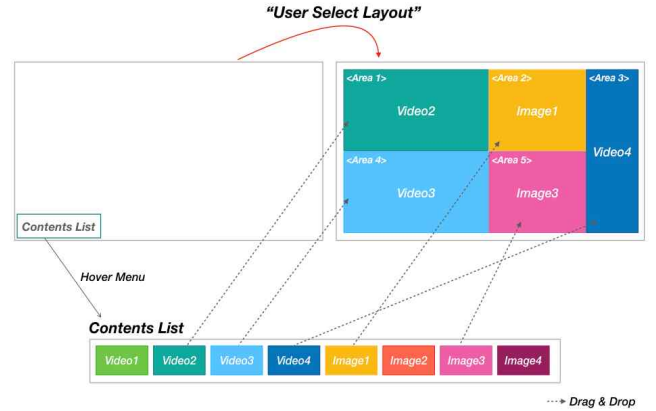


Fig 3. Customized Layout Application UI

C. Communication between each device

Fig 4. shows a communication procedure for exchanging information between devices. The TV first requests the primary application to the web server, and the web server replies to it. The MMT-CI having the space-time configuration information of each content package transmits the information of the view 1 to the TV through the XML parser and executes the scene synthesis of the main application. A companion device, a smartphone, runs a companion application through requests and responses to a web server. As a result, when the viewer completes the various layouts, the CI generator of the application generates the MMT CI and transmits the screen configuration information of the View 2 to the web server. The web server that receives the information causes the main

application to update the screen and display the new scene in the viewers.

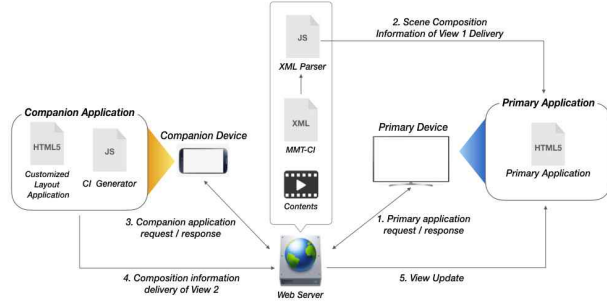


Fig 4. Communication sequence diagram between each device

IV. IMPLEMENTATION



Fig 5. Implementation Result using 4K TV and Smartphone

```
<mmtci:CI>
  <mmtci:view id="view2" style="position:absolute; width:2560px; height:1440px">
    <mmtci:divlocation id="div1" style="top:0px;left:0px;width:1181.1635112858003px;
    height:748.3406225881142px" begin="145.326415"></mmtci:divlocation>
    <mmtci:divlocation id="div2" style="top:707.7128589689993px;left:0px;width:
    1181.1635112858003px;height:732.2883457679444px" begin="149.55">
    </mmtci:divlocation>
    <mmtci:divlocation id="div3" style="top:0px;left:1162.2989729808753px;width:
    682.1668464458434px;height:1440px" begin="145.2977218"></mmtci:divlocation>
    <mmtci:divlocation id="div4" style="top:0px;left:1840.7631197206454px;width:
    719.2348642150215px;height:684.5575508103935px" begin="undefined">
    </mmtci:divlocation>
    <mmtci:divlocation id="div5" style="top:666.7763928137864px;left:
    1837.7711509959388px;width:722.2288400040612px;height:770.5685618729098px" begin=
    "undefined"></mmtci:divlocation>
  </mmtci:view>
</mmtci:CI>
```

Fig 6. The generated MMT-CI XML document

A. Experiment Environment

Primary Application and Companion Application are written in HTML5, javascript and, it' library, jquery 3.3.1 [7]. In addition, in the case of the Companion Application, the drag and drop and the resize function are implemented by applying Interact.js 1.3.4. Also, the web server was developed using Node.js and the web socket communication system was made using Socket.io.

The videos included in the content list are shown in the list through the MPEG-DASH(Dynamic Adaptive Streaming over HTTP) video streaming method. Each video was encoded with 720p, 1080p, and 2160p through FFmpeg [8], and MPD files were generated by dividing into 4-second segments using MP4Box [9].

B. Implementation

For the companion device used for the experiment, Samsung Galaxy s8 and LG G7 ThinQ were used, and as the primary device, Skyworth 39E 780 U 4K TV was used. We run experiments using Mozilla 's Firefox running web application. The experimental results are the same as in Fig 5.

Fig 6 is the View 2 MMT-CI document created using the MMT-CI generator included in the Web application. When the MMT-CI document corresponding to the TV on which the image of View 1 is played is transmitted, the layout of the image and the image changes accordingly.

V. CONCLUSION

In this paper, we have studied and implemented a companion screen service that enables viewers to flexibly advance the composition of scenes of contents. With the widespread use of smartphones and the completion of ATSC 3.0 standards, various TV supplementary services that can escape from traditional broadcasting and pursue bidirectional development are developing. The service developed in this paper is a system that can adjust the content, position and size desired by the viewer, out of the size of the fixed content of the existing multi-channel service, the active participation of viewers Can be brought about. Also, we hope that this service can be applied not only in multi-channel but also in point and VR video.

ACKNOWLEDGMENT

This work was supported by an Institute for Information and communications Technology Promotion (IITP) grant funded by the Korea Government (MSIT) (No. 2016-0-00144, Moving Free-viewpoint 360VR Immersive Media System Design and Component Technologies)

REFERENCES

- [1] TTA, "Standard Guide UHD IBB Service of Terrestrial broadcasting", TTA.KO-07.0128/R1, 2017, pp.46-49
- [2] Jaewon Moon, Taebum Im, Seungwoo Keum, Yunseok Nam, Sangwon Lee "": Design and Implementation of Platform Providing Adaptive Contents based on IBB Companion Screen Service", JBE Vol. 22, No. 1, pp.95-105, January 2017
- [3] Youngmo Ban, Yunseok Nam, "New Broadcasting Service, Terrestrial UHD Second Screen Service", Broadcast and Media, Republic of Korea, vol.23, pp. 45-55, January 2018.
- [4] Sangmin Sim, "Websocket and Socket.IO", September 2011 [online], <https://d2.naver.com/helloworld/1336>.
- [5] Kyungmo Park, Youngkwon Lim, Shuichi Aoki, Gerard Fernando, JinYoung Lee "Study of ISO/IEC CD 23008-1 MPEG Media Transport", October 2012
- [6] Minwoo Jo, Jungwook Park, Kyuheon Kim, "Scene Composition Technology Based on HTML5 in Hybrid Broadcasting Environment", JBE Vol. 18, No 2, pp. 237-248, March 2013.
- [7] JQuery 3.1.4, <https://jquery.com/>
- [8] FFmpeg, <https://www.ffmpeg.org/>
- [9] MP4Box, <https://gpac.wp.imt.fr/mp4box>