System Design and Service Scenario for the Second Screen Service

Joo Hyun Park[1], Soon-Bum Lim[2,*]

Abstract

Today, the proliferation of various mobile devices, such as smart phones and tablet PC, brought changes in the existing TV viewing behavior. People use smart devices as secondary device while watching TV. Researches on a wide range of services linked with second-screen devices around the smart TV in the home network have been actively conducted. While there exist several Web-related technologies for connections between devices, specialized techniques for a second screen service are quite insufficient. There are still some problems related to the display of contents from multiple devices and the efficient transfer of these contents. Considering the characteristics of broadcasting systems In this study, we focus on a second screen service that permits a dynamic transfer of contents by connecting a television (TV) with mobile devices. Here, we propose a second screen service model to enable the personalization of TV contents by combining the existing broadcasting and Web-related techniques.

Key Words: Hybrid Broadcast Network, Personalization, Second Screen Service, Web-based TV Technology.

I. INTRODUCTION

The existing broadcasting systems are starting to display dual contents from original media and the Internet as one broadcasting unit, thereby transforming into a hybrid network. By considering a second screen service, the world broadcast standards such as Hybrid broadcast broadband TV (HbbTV) and Advanced Television Systems Committee (ATSC), and Korea's standard ground-wave networks are being altered to become the standards for the next generation of broadcasting systems [1].

While there exist several Web-related technologies for connections between devices, specialized techniques for a second screen service are quite insufficient [2]. There are still some problems related to the display of contents from multiple devices and the efficient transfer of these contents. Further, the current research on the second screen service has only been applied to testing services, and there are few, if any, successful cases of a commercial service.

Furthermore, considering the characteristics of broadcasting systems, which evolve into a hybrid broadcasting system, in this paper, we propose a second screen service model to enable the personalization of television (TV) contents by combining the existing broadcasting and Web-related techniques.

II. State of the Art

2.1 Web & TV Working Group

We studied the "Web and TV Interest Group" organized by W3C to find research trends related to the second screen service [3]. The Web and TV Interest Group provides a forum for Web and TV technical discussions, reviews the existing work and the relationship between Web services and TV services, and identifies requirements and potential solutions to ensure that the Web will function well with TV. Recently, research on the connection technology between various types of devices and a second screen service using Web techniques has been actively carried out in the Web and TV Interest Group's forum [4]. Some of the major studies are the various use cases of second screen services, LinkedTV, and UPnP.

2.2 Open Annotation Data Model

The Open Annotation Data Model(OADM) specifies an interoperable framework for creating associations between related resources, annotations, using a methodology that conforms to the Architecture of the World Wide Web through W3C Open Annotation Community Group [5].

The primary aim of the Open Annotation Data Model is to provide a standard description mechanism for sharing Annotations between systems. This interoperability may be either for sharing with others, or the migration of private Annotations between devices. The shared Annotations must be able to be integrated into existing collections and reused without loss of significant information. An annotation is considered to be a set of connected resources, typically including a body and target, and conveys that the body is related to the target. The exact nature of this relationship changes according to the intention of the annotation, but most frequently conveys that the body is somehow "about" the target. Other possible relationships include that the body is an identifier for the target, provides a representation of the target, or classifies the target in some way. RDF (Resource Description Framework) is the core technology of the OADM.

- Resource : Everything has a URI(Web pages, Images, Videos, etc)
- Description : Description of the resource's properties and attributes
- Framework : Model, Language and grammar for the description and Resource

III. Design of Personalized Second Screen Service Model

Here, we propose a second screen service model to adapt Web techniques for a personalized TV contents service. The proposed model is composed of four parts. One part specifies the system configuration. Second part describes the extension design of OADM. Third part explain a method of personalized profile extraction. The last part specifies the reconstruction of additional content process.

3.1 Overall System Configuration

A second screen service is a service between a public device such as a smart TV and a private device such as a smartphone. While the main contents are displayed on the public TV, the additional contents are displayed on the secondary devices. These additional contents are related to the main contents played on the TV, and the contents are changed according to the user information transferred from a second screen device to the TV. In order to play the additional contents on the second screen devices, several

techniques are required to connect these devices and to deliver the contents. As shown in Figure. 1, the TV and the second screen devices are connected by using universal plug and play (UPnP) techniques, and the additional contents from the set of broadcasting contents are delivered from the TV to a second screen device by using Web socket techniques [6].
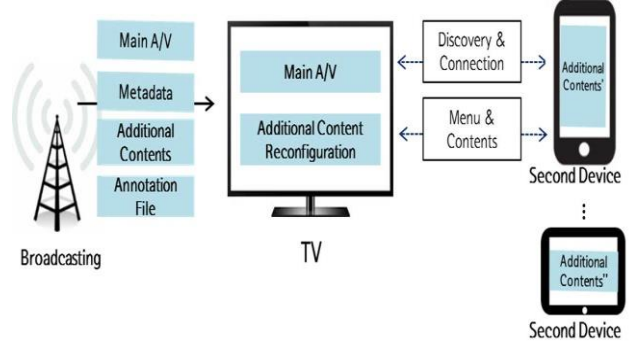

Fig. 1. System Configuration

3.2 Extension Design of Open Annotation Data Model

We applies OADM to the additional information representing the broadcast content for the purpose of the connection between the broadcast content and the additional content. Figure 2 shows a graphical representation of a stat of the plurality of additional content be included in a broadcast program. Annotation that are additional contents can be extended because it is the instant of class. Annotation in square box express additional content information for a specific period.
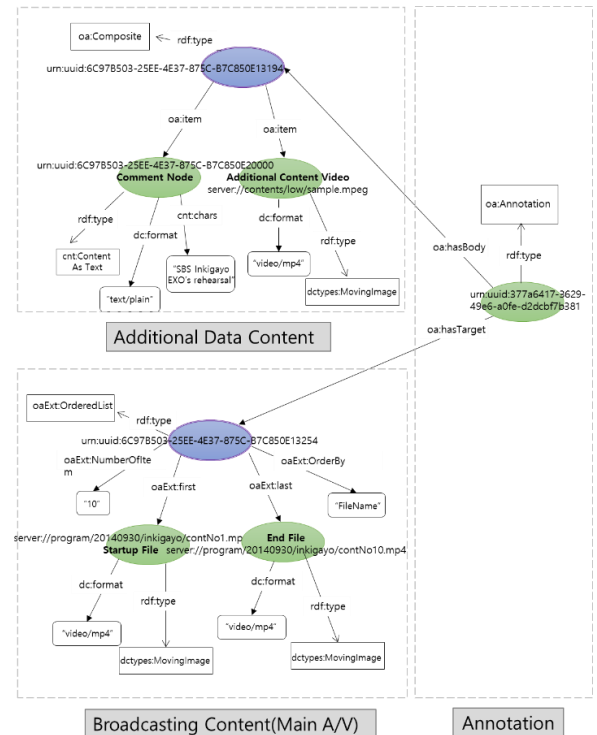

Fig. 2. Additional information represented by the Open

Annotation Data Model Graph

Existing OADM is not suitable to describe the resource list in the order. Although list class can represent a first and last list of the specific period of the additional contents, this class should be displayed individually in the order of the list between the first and last resource. Therefore we extend a class named OrderedList and express the list of resources in the order only by specifying the beginning and end Resources. The extension specification includes a Relationship (first, last) which can specify the first resource and last resource and Property (NumberOfItem, OrderBy) specification which specifies the number and sort of the resource files. <Table 1> is the Open Annotation Ontology Extension for new OrderedList class.

Table 1. Open Annotation Ontology Extension.

Class: oaExt:OrderedList URI: http://mintlab.sookmyung.ac.kr/ns/oaExt#OderedList *OrderedList* – A class representing a collection of resources in order. Subclass of oa:List class. This class represents the beginning and end of resource without having display all of the resources in the order.
Relationship: oaExt:first URI: http://mintlab.sookmyung.ac.kr/ns/oaExt#first *first* – The first resource in the list that has to be expressed in ordered list. Can be represented by a resource URL. The sub property of rdf:first.
Relationship: oaExt:last URI: http://mintlab.sookmyung.ac.kr/ns/oaExt#last *last* – The last resource in the list that has to be expressed in ordered list. Can be represented by a resource URL. The sub property of rdf:last.
Property: oaExt:NumberOfItem URI: http://mintlab.sookmyung.ac.kr/ns/oaExt#NumberOfItem *NumberOfItem* – The total number of resources(Item) The number of resources include first and last resources. The sub property of rdf:value. Value – Integer
Relationship: oaExt:OrderBy URI: http://mintlab.sookmyung.ac.kr/ns/oaExt#OrderBy *OrderBy* – Sort the ordered list of resource. Sub property of rdf:value. The file is limited that the file name is numbered in a sequential number or time. Value – FileName, FileTime

3.3 Personalized Profile Extraction

The user-specific preferences analysis is needed to provide a personalized service. The user profile information is needed in order to provide user-specific additional contents[1]. It utilizes user personal information like primary TV viewing history and additional contents

viewing history information in the user profile information.

3.3.1 User Personal Information

The user personal information contains extracted result of the preferred program category through the user ID, age, sex, favorite category, and priority. The user personal information will utilize the values entered manually by the user in the personal device.

3.3.2 Primary TV Viewing History

In primary TV viewing history information, it extracts the subject / item / Cast information of the program corresponding to the user through the top ratings of the TV program viewing pattern analysis. The more user will watch a lot of programs in a certain category, the more this category will be given a higher scores for the subject / item / Cast of the program. The ratings of a particular program (P) of the user are calculated as follows. After this, it will score the subjects / item / Cast of the program corresponding to the top of the ratings list (see Eq 1.).

The ratings of a particular program(P) of user
$$= \frac{\sum P \, Viewing \, Time \, of \, user}{\sum Total \, Broadcasting \, Time \, of \, P} \quad (1)$$

3.3.3 Additional Contents Viewing History

Additional Contents Viewing History is same with the way of scoring calculation for Primary TV Viewing History. It is calculated separately for additional content viewing history by Purchased contents viewing points and transferred additional contents viewing points, and extracts topic / item / Cast information. In order to calculate scores in Transferred additional contents viewing history, it uses the number, the time of additional contents viewing, etc. The utilization score of Transferred additional contents will be able to calculate as follows.

The score of transferred additional contents
$$= w_1 f_i + w_2 t_i \quad (2)$$

(f_i : Frequency of using transferred additional content,
t_i : Time of using transferred additional content,
w_1:Weight for the frequency, w_2 :Weight for the time)

Purchased Content is calculated by the same way of transferred additional content. But it has a difference that weighing for the purchased content. The way of calculating score is as follows.

The score of purchased content
$$= w_1 f_i + w_2 t_i \quad (3)$$

(f_i = Frequency of using purchased content, t_i =Time of using purchased content, w_1 = Weight for the frequency, w_2 = Weight for the time)

The points which calculated by these three kinds of data listed in the list of each program and additional content by the topic / item / Cast. After this, add the item with the same score each other, then select K of the highest score for affinity. Like this way, it can provide affinity per user.

3.4 Reconstruction of Additional Content

The reconstruction of additional content process is performed on TV. Parsing the Metadata, Annotation F ile and multimedia data received from the broadcast a nd match with user preferences. As a result, system c an provide the personalized additional content accordi ng to the user. Figure 3 shows the reconstruction of additional content process.
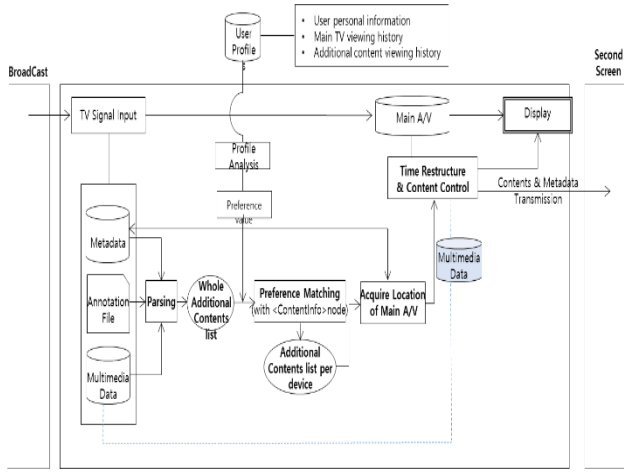

Fig. 3. Reconstruction of Additional Content Process

Ⅳ. System Implementation

We show the applicability of the proposed second screen service by implementing its prototype. The prototype carries out the following three processes:

- Device Discovery and Connection
- TV Contents Separation
- Real-time Contents Delivery

4.1 Device Discovery and Connection

A second screen device sends a message confirming the connection on the TV screen when the TV requests a connection or when the connection is completed. In this step, the HTML connection page uses a network service discovery API to find a service in the UPnP control device [6].

UPnP consists of a control device, a controlled device, and a service. In this study, the TV is set as the controlled device because it performs the role of a server that provides the second screen device with additional contents. The second screen device is a control device playing the role of

a client receiving the additional contents from the TV.

The UPnP networking process performs the following six-stage process to find the abovementioned devices and connect them: assign addresses, discover devices, device description, control, event, and presentation [3]. In the assign address stage, when the TV and the second screen devices access a common network, they search the dynamic host configuration protocol (DHCP) and receive an IP address. In the discover device stage, second screen devices use the simple service discovery protocol (SSDP) to search for a TV. When a TV is discovered, the second screen device requests for the technical description and service description of the TV device. Then, in the control stage, a second screen device analyzes the information obtained from the previous stage. The device also sends a command to the TV through the simple object access protocol (SOAP). In the event step, the TV notifies the stat of the variables and the second screen devices are operated [Figure. 4].
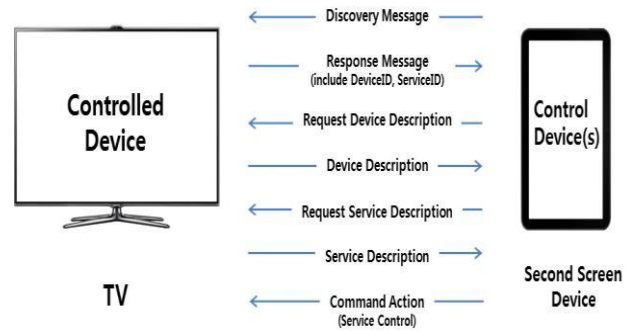

Fig. 4. Process of UPnP Connection

4.2 TV Contents Separation

After the device discovery and connection step, a reconstruction process can be performed for additional contents.
The original broadcasting contents are separated into the main TV contents and the additional contents. The annotation file written in the open annotation data model is interpreted to separate the main contents and the additional contents [7]. Figure 5 shows the reconstruction process and the processes are as follow;

- TV Signal Input → metadata, annotation file, multimedia data Parsing → Extraction of full additional contents list → Matching with User Preference → Obtaining the additional contents list for each device → Obtaining Main A/V location that corresponds to the acquired additional content → Time Reconfiguration and Content Control between Main A/V and

Acquired Additional Content → Delivery of Content to Second Device /Display on TV.

An additional contents list and the corresponding menu are created through a reconfiguration process. This menu is displayed on each second screen device adaptively by using the CSS and WebSocket technology.

The menu consists of three items: List of Contents of My Interest, Current Broadcast Content, and Contents of My Interest. On the section of List of Contents of My Interest, an additional content list that matches the user's preferences is displayed. This section's size is flexible according to the content quantity. The Current Broadcast Content section displays a list of additional contents included in the current channel irrespective of the user's preference. This section can be automatically updated whenever a TV scene changes. The Contents of My Interest section plays the contents selected by the user from the List of Contents of My Interest menu or the Current Broadcast Content menu.
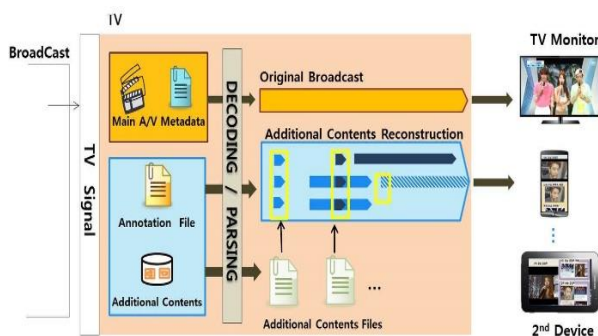

Fig.5. Reconstruction Process

4.3 Real-Time Contents Delivery Step

After the second screen device and the TV are connected, the additional contents can be sent to the viewer's second screen device according to the viewer's request by using WebSocket[8]. WebSocket enables a real-time transmission of the additional contents whenever the TV scene is switched. Next, we implement a server that can send binary data with the Node.js module and the contents to a client's webpage through WebSocket. In this section, we describe the server and client configuration and the implementation method for the server/client side for real-time contents delivery.

A server is built into the TV and a webpage plays a broadcast on the TV. The TV server executes a series of functions and stores the additional contents data. When the TV sends the segmented file number corresponding to the current channel to the TV server, the TV server verifies the existence of the additional contents. If the file has additional contents, the server sends a list of these additional contents to the second screen application according to the user's preference. The list is sorted and displayed in an appropriate section. Then, the selected contents are streamed through WebSocket from the TV server. Figure 6 shows the configuration of the server and the client.
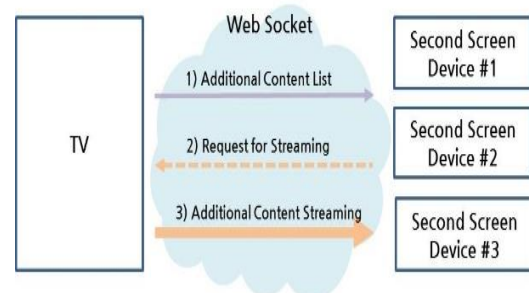

Fig. 6. Composition Diagram of Server and Client

Next, we create a server that performs binary data communication with the BinaryServer() function in the server-side javascript. It designates a variable that defines the current situation and executes certain functions according to the situation. When a connection is established, the server receives the metadata file from the client. At this instance, the file is handled with the pipe() function.

In the client-side javascript file, the IP address of the server that will connect to the BinaryClient() function becomes a factor for connection. Further, the protocol for WebSocket communication is written with "ws," and not "http." When the statement of a client becomes "open," the stream and streaming file is set by the createStream() function.

V. Implementation of Sample Prototype

In this section, we suggest a scenario for the realization of a personalized service and confirm the results of the sample prototype.

5.1 Personalized Service Scenario

We present a service scenario that can be applied to an actual Korean TV program. This scenario introduces a service that encourages a viewer to stay on the current channel. In this scenario, there are three viewers watching a music program; two viewers have common interests, and

the third viewer has different interests. While the viewers are watching the music program, various additional services such as preview video, paid contents, and program information are provided on the individual devices according to the viewer's interest. Figure 7 shows the flow of the scenario.
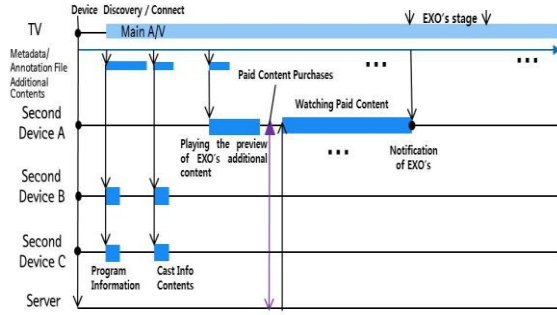

Fig. 7.   Flow of the Scenario

There are three viewers. Viewer A is a person who likes the group "EXO" and uses a tablet as the second device. Viewers B and C do not have a specific favorite singer, but they usually like watching a music program on the TV. However, the second screen devices of these two viewers are different; viewer B uses a laptop, and viewer C uses a smartphone.

On viewer A's second screen device, EXO's rehearsal preview video is delivered, while EXO is not on the stage. Viewer A purchases and watches the full version of EXO's rehearsal video waiting for the EXO's on-stage time. Upon EXO's turn, a notification is sent to viewer A's screen. When EXO's video is being played on viewer A's screen, the program and cast information contents are delivered on viewer B's and viewer C's screens. Although viewers B and C are provided the same contents, the layout of these contents varies according to the device size.

5.2 Implementation Result

We set the precondition for the sample prototype of the proposed second screen service scenario. A laptop is used instead of a TV in the hybrid network environment. This laptop stores additional contents on its server. Further, we assume that the TV receives the user's profile from a second screen service application.

The prototype execution is as follows: The connection buttons on the services pages of the second screen devices enable these devices to access UPnP. When the saved segment files start playing, the laptop sends a message to the second screen devices. This message shows that the laptop has each of the second screen devices' contents. If a user selects the desired contents from the menu, a request to assign the additional contents for real-time streaming is sent to the server. If the server accepts this request, the related additional contents are streamed.
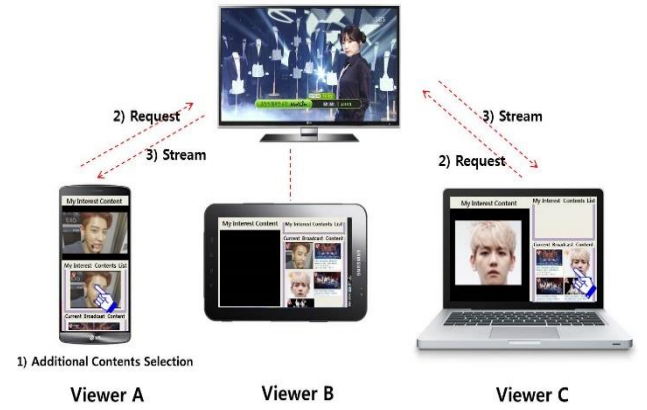

Fig. 8.   Result Screen

VI. CONCLUSION

In conclusion, we confirmed the basic Web-related technology for the realization of a second screen service, such as the connection technology between different devices and the real-time contents transmission technology. Then, we designed a service model and a service scenario for a personalized second screen service and confirmed the possibility of this service through the prototype implementation. In the process, we make it easy to express a large amount of content by extending the OADM class. This study hopes for a more advanced and efficient technology to extract Web contents from a set of broadcasting contents and deliver them to the user-desired device. Moreover, we expect the advanced second screen service to be commercialized in the future.

REFERENCES

[1] JH. Lee, DH. Choi, DJ. Lee, and KS. Kim, "2nd Screen Service Technology of KBS-TVut Platform," *Proceedings of the 2014 Korean Society of Broadcast Engineers Summer Conference*, pp.15–17, 2014

[2] JH. Park, SY. Jeon, SA. Jung, HJ. Yim, BJ. Bae and SB. Lim, "Development of a Second Screen Service for Personalized Content Delivery," *Int. J. Control and Automation*. Vol.8, pp.321–330, 2015

[3] Web and TV Interest Group, Available: http://www.w3.org/2011/webtv/, 2016

[4] N. Borch, B. Cheng, D. Raggett, and M. Zorrilla, "An architecture for second screen experiences based upon distributed social networks of people, devices and programs," *The Fourth W3C Web and TV Workshop*, 2014

[5] Open Annotation Data Model, Available: http://www.openannotation.org/spec/core/, 2016

[6] J. Clarke Stevens, "A Flexible Multi--Screen Solution Based on UPnP," *The Fourth W3C Web and TV Workshop*, 2014

[7] Open Annotation Data model(W3C), Available: http://www.openannotation.org/spec/core/, 2016

[8] The WebSocket API(W3C), Available: http://www.w3.org/TR/websockets/, 2016

Authors

Joo Hyun Park (park.joohyun5@gmail.com)

She is currently attending on the Ph.D program in the Dept. of IT Engineering at Sookmyung Women's University in Kore. Her main research interests are electronic publishing for reading-disabled people, web and mobile multimedia contents and user interface.

Soon-Bum Lim (sblim@sm.ac.kr) received his BS degree from Seoul National University , Korea, in 1982 and his MS and Ph.D degrees in computer science from KAIST(Korea Advanced Institute of Science and Technology), Korea, in 1983 and 1992, respectively. From 1989 to 1997, he was the engineering director for the font technology and printer division at Human Computer, Inc, and Trigem Computer, Inc. From 1997, he was an assistant professor in the Dept. of Computer Science at Konkuk University in Korea. From 2001, he is currently a professor in the Dept. of IT Engineering at Sookmyung Women's University in Korea. His main research interests are computer graphics, web and mobile multimedia contents, user interface, and electronic publishing such as font, eBook, and XML documents.