

W3C Web and TV : Issue on Web technologies for N-Screen

Jinhong Yang¹, Park, Sangmin², Junkyun Choi³

Information and Communication¹, Department of Electronics Engineering^{2,3}

KAIST

103-6 Munji-Dong, Yuseong-Gu, Daejeon 305-714, Korea

sunupnet@kaist.ac.kr¹, deadpk@kaist.ac.kr², jkchoi59@kaist.edu³

1. Introduction

From 2006, we have performed researches on both technical and business aspects around IPTV service funded by Korean Ministry of Knowledge and Economics and Korea Telecom. Especially, we researched on the technical issues to provide IPTV service through the Web under the project, "Research on the Web-based open IPTV service technologies". Part of the results, content aggregation and syndication system, are currently considered to be adapted to the Korea Telecom's IPTV service infrastructure [1].

Nowadays we research issues on Web technologies for TV and new media environment, which includes device and service discovery and synchronization, media sharing between primary and second screen devices on second screen scenarios.

In this paper we describe the issues on Web technologies for N-screen environment and our approach.

2. Point of View

2.1 Device Discovery & Synchronization

In home networking environment Web technologies are used to combine inter device services and the previous approach, such as DLNA or UPnP. That's because the most recent devices are Web enabled and the Web technologies are developed enough to provide an easier way to construct network system. These inter device network have some features, which the network provides services to users by interconnecting several devices. The service gets elaborated when the number of connected devices increases.

By the way, it is very important that we should investigate devices and know the features and functions of them when we construct a network. Each device has its own features and we have to give proper actions and command when the device is deployed to the network [2]. That's why we should be aware of their locations and functions. Adding or removing devices implies the service which the system can provide to user changes. Therefore devices are to be checked every moment to provide proper service.

2.1.1 System structure

It is an important issue that how we can provide device discovery. I think there are several network

structures about notifying location and sending information among devices. As their implementation differs, their pros and cons also differs. Hence, we figure out some structures and describe below.

- Server oriented structure: Server takes a role of managing devices' information and location. When new device is added or device is removed then the server checks this out and notifies connected devices.
 - Server maintains devices' location and information.
 - Device do not need to search other devices and can know other devices by querying server.
- There is no center server, so each device detects other devices by itself.
 - Devices exchange their information and location via p2p transport.
 - It is easy to expand and deploy network to other networks [3].

2.1.2 Device discovery policy

Devices should get to know other networks and it can be done by device discovery.

So the discovery policy may include when or how to discover. It varies according to network purpose and network structure. Here are some issues which we can think of generally.

- The way of device discover the adjacent devices:
 - Device or server discovers devices periodically. This method can catch the broken device. Thus the system can maintain network properly. But it has a disadvantage that the network consumes battery or resources continuously so we have to be careful.
 - User or manager has a role of adding or removing devices. Manager should take care of device modification manually. It is possible to prevent devices sending discover message continuously but it takes a time. And also one device's failure should be notified in some way or the whole network system will fail.
 - How far device discovers adjacent devices. If a device searches in wide area it can catch more devices but it might consume more energy. However we cannot predict what device will come and we cannot know its place. People can handle it manually; otherwise the device should take proper action.

2.2 Second Screen

N-screen enables users to see objects of main screen in Web browser. Users can control or manage those things in a second screen. The screens are interconnected elaborately so one can manage both screens with handling only one screen. User should use first and second screen without confusing so it can be compared to using TV with remote control.

2.2.1 System structure

Second screen can be applied in a different situation. Second screen can take a remote control role or just show some part of first screen. It is also possible that second screen show additional view which is not in first screen, so the users can obtain optional information or data.

Therefore the N-screen can be implemented in a different ways.

- Server sends different information according to the screens.
 - Sending different information in a server level has some advantages. Server sends only necessary part of page so the clients just see the received page. It is faster and simple for clients but the server should spend more time to generate different page show. However, most implementation should be taken on a server side and it might cause problems.
- Client receives the same page from server but just see some part of it.
 - Clients themselves control and filter pages. Server doesn't need to consider which client is. The client can hide needless part and see necessary part. This scheme does not require server side implementation but the client should spend more time to process the page.

2.2.2 Session control policy

Since the N-screen services show pages, controls, or optional pages, we should consider some points.

- We should keep track on users of main screen and second screen. Server can determine which page must be shown on second screen after seeing what's on main screen. Second screen will be chosen by users.
- It is not necessary that the user of main screen and second screen always the same. The relationship of users between main screen and second screen can be a relationship between presenter and audience. For this reason user of second screen is not necessarily the user of main screen, but the users should be authenticated.
- Sometimes second screen should show different screen pages while they connect to same main screen services. When we see contents of main screen and second screen. We should consider the session information at the same time. Therefore we check the state changes of main and second screen continuously.

2.2.3 Implementation

Now we describe our approach to N-screen. We examine a case of an object movement in multi-screen environment.

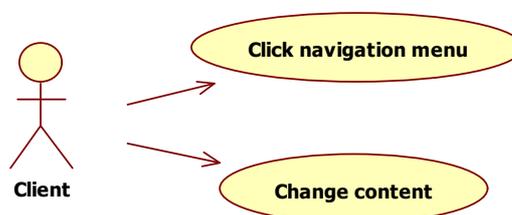


Figure 1 Use case of client

Above figure 1 briefly describes client use case. Client can surf the Web as usual, but he can navigate Web page by clicking navigation menu. This triggers the content of Web page changes. How the content changes is like this. We think of an idea from the DOM tree. Since Web page is a tree structured form the user can navigate by following DOM tree structure. User starts to navigate from BODY tag in HTML. Then the navigation menu shows the element which is child of BODY tag. If the client clicks one of them then the element is shown while the BODY disappears and the navigation menu changes to a list of the child's children elements. Therefore, the user can see an element that he really wants to see.

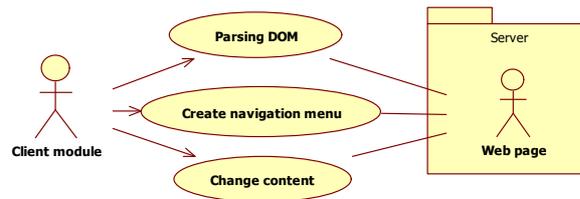


Figure 2 Use case of client side module

Client module makes navigation menu and changes content as the above figure 2 shows. It requires the module having the DOM tree structure and parsing page source should be done in advance. When the module obtain page elements and get view element then it make navigation menu. Now user can click the navigation menu and is able to see only necessary element.

Our goal is described below Figure 3.

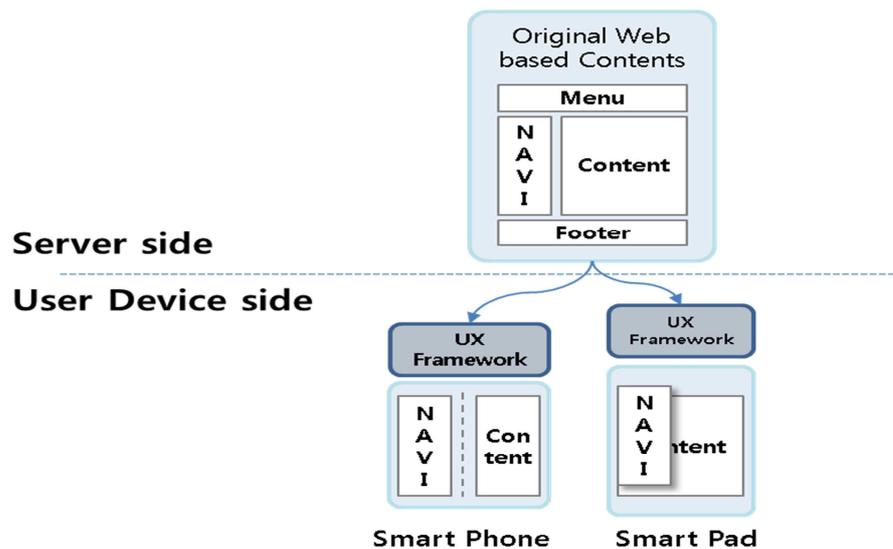


Figure 3 Diagram of system overview

Server side contains all the page contents in a one page. However, it is not appropriate to show the page in smart phone or smart pad because the resolution or screen size is much smaller than the entire page. Just resizing them to a small size doesn't works nicely. Thus, we only show the necessary part and make some menus so that user can change other views or hide some views.

User can see what he really wants to see and the other elements of the pages are hidden. User can navigate the pages by clicking buttons or changing settings.

3. Conclusion

So far, we covered some technical issues on N-screen environment. We figured out some issues about device discovers and N-screen environment. It includes some system structures and policy when dealing with device discover and N-screen. Additionally, we added some system overview of our implementation.

We have interest of Web and TV and we have studied on smart TV, N-screen, multi-screen communication, DAP, and so on. We are encouraged to talk about the shape of these new technologies and curious about what will comes next.

References

- [1] Jinhong Yang, Hyojin Park, Jun Kyun Choi, 'A Web-based Content Syndication Platform for IPTV,' ICOIN 2011, Page 494-497
- [2] Urs Bischoff, Gerd Kortuem, 'Programming the Ubiquitous Network: A Top-Down Approach', Proceedings of the System Support for Ubiquitous Computing Workshop (UbiSys'06), 2006.
- [3] Michiharu Takemoto, Hiroshi Sunaga, Kenichiro Tanaka, Hiroaki Matsumura and Eiji Shinohara, 'The Ubiquitous Service-Oriented Network (USON) - An Approach for a Ubiquitous World based on P2P Technology -', second International Conference on Peer-to-Peer Computing IEEE (2002)