Mobile Contents Handling System for Ubiquitous Environments Using UML

Haeng-Kon Kim[1] and YounKy Chung[2]*

[1]*School of Information Technology, Catholic University of Deagu, Korea*

[2*]*Department of Computer Engineering, Kyungil University, Republic of Korea*

hangkon@cu.ac.kr, ykchung@kiu.ac.kr

Abstract

The expanded bandwidth for air interfaces has made a solid ground for streaming media application on wireless networks. With the advantages of wireless system in time and place, mobile streaming media service is very attractive. With the further increasing popularity of multimedia devices, such as laptop computers, personal digital assistants (PDAs), and smart phones, there is a rapidly growing demand for efficient real-time playback and transmission of large amounts of multimedia data, but most users' connections are not fast enough to download large chunks of multimedia data. Moreover, when switching to another device while in the middle of downloading content, users have to download the same multimedia content all over again. A streaming service technology is therefore needed in which users can seamlessly download on the network large multimedia data based on the user's history of preferred content.

This study suggests a server–client streaming system that enables seamless multimedia content download that allows the user accessing the data to switch conveniently to another device or client without having to repeat the download to the beginning. The proposed system is also designed to be compatible with many users and different device environments through the home network by storing/managing information in the home server factors such as user preferences, device character information, the indication on the exact point the download was interrupted, and other related information.

Keywords: *mobile streaming media service, streaming service, server–client streaming system*

1. Introduction

As multimedia technologies develop and high-speed network technology advances, large amounts of multimedia content are produced and distributed. Moreover, as home multimedia devices become increasingly commonplace, receiving and storing multimedia content through various channels even on a home network becomes possible.

Various multimedia devices, such as personal computers (PCs), televisions, laptop computers, and personal digital assistants (PDAs) connect to the home network, along with set-top boxes for satellite or cable broadcasting, and devices such as personal video recorder (PVR) for recording/playing multimedia content.

For example, a multimedia client such as a PC or a PDA, which can connect to a wired or

* Correspondence Author

wireless communication network that constitutes a home network, can also receive multimedia content in various formats on such a network, that is, the Internet. The set-top box or PVR can also receive multimedia content through the satellite or cable network. The office server [1] is a server that implements data streaming in various formats to various multimedia clients on a home network through wired or wireless protocols, such as BcN, HFC, FTTH, ATC protocol, and wireless LAN. Figure 1 shows the system structure that conducts IP (Internet protocol) streaming of multimedia content stored in the office server and received through the external wired and wireless network. Figure 1 shows the structure of the mobile streaming media system, including four basic parts: the server, the client, the transmission channel and the content source. The server adopts concurrent server of cluster; the client is a cell phone supporting BREW application; the transmission channel uses CDMA1x wireless network of China Unicom; and the content source are the streaming type documents or real-time audio/video data. In addition to providing the basic streaming service, the server deals with the additional function such as file management, access certification, and charging [3, 4].

Internet protocol (IP) is a multimedia data streaming service that provides data in various formats to multimedia clients in real time through a home server using the RTP control protocol (RTCP) [2] that transmits controlling information and real-time transport protocol (RTP)[3], a transmission protocol for real-time applicable data. In such cases, the home server monitors the home network to inform the home network mode, and guarantees quality of service (QoS) by conducting a data streaming service according to the proper bandwidth.

However, in most cases, users themselves are capable of conducting data search and management, that is, users know how to deal with the multimedia file they have searched and downloaded on the Internet. Moreover, with the widespread proliferation of multimedia devices such as PCs, laptop computers, PDAs, and smart phones, there is an increasing demand for the efficient transmission and real-time playback of large amounts of multimedia data.

However, most users do not have fast enough connections to quickly download multimedia data. Especially, when users receive the multimedia content via data streaming through a wide-screen client such as a TV or a PC, then switch to a different small-screen device such as a mobile phone or PDA, they have to download the same multimedia content all over again.

This study proposes a data streaming system that enables device users to get an intelligent recommendation regarding the media content they want based on the user preferences stored on the home server, and that users receiving the data streaming service with a multimedia client on a home network are assured of seamless access to such multimedia content even if they switch devices.

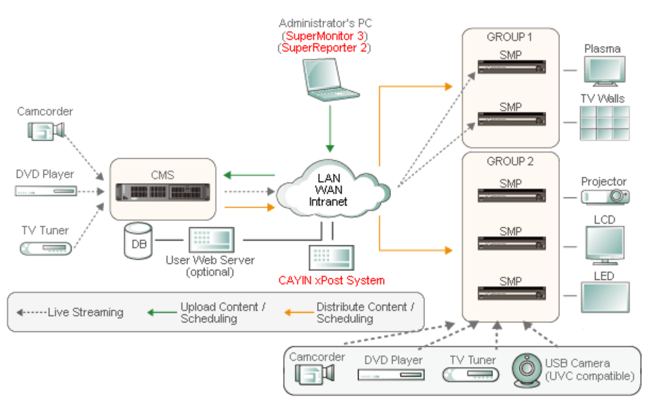

Figure 1. Content Streaming System Structure on a Office-based Network

This system is stored in a home server so users can get media content recommendation depending on their preferences regarding factors such as actors/actresses, directors, producers, genres, content title, and key words. Therefore, users themselves don't have to search or manage the content that they want from the home server for large multimedia content transmitted to the client on the home network. By implementing streaming based on user identification and device character information of many multimedia clients on the home network, users are assured of seamless access to the streamed content even if they switch to another different client on the home and office network.

2. Structure of Streaming System and Constituting Factors

Data streaming technology, which enables the real-time playback of video without having to download the file, is now common. Real Network's Real Player System [4] or Microsoft Corporation's NetShow Service [5] represent some of the leading real-time streaming technology. However, the technologies mentioned have a weak point: they are platform dependent, which means they cannot be ported to a different platform easily.

To complement such a weakness, the streaming system that this study proposes uses Simple Object Access Protocol (SOAP) [6, 7]. SOAP is Microsoft Corporation's protocol to access objects on the Web, and uses XML language with HTTP to send text commands. Therefore, implementing message communication using SOAP requires a server–client communication protocol for SML Web service based on the IP Network so it does not cause problems in getting ported between different devices.

Figure 2 shows SOAP class in a two-way communication network using UML notation.

Moreover, users receive intelligent recommendations on the media content they might like based on their recorded user preferences, and the data on user's preferences are managed by Metadata. It has to be constituted according to the XML Metadata Schema [8] defined in TV-Anytime [9] and filtered, and transmitted to the client so that it can use XML-based SOAP communication message.

The system enables seamless data streaming service based on the user's preferences and device information is constituted of the Server Streaming Manager; User Database, Contents Database, Pumping Server constitute the server; and Client Streaming Manager, User Interface, Media Player constitute the client.

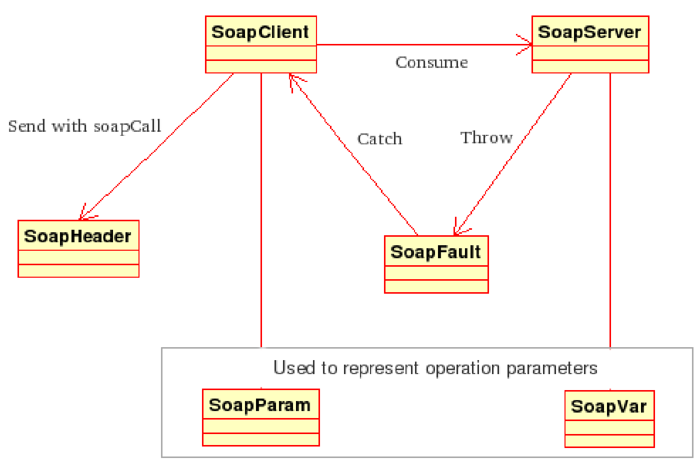

Figure 2. The Bi-directional SOAP Transport Stack using UML

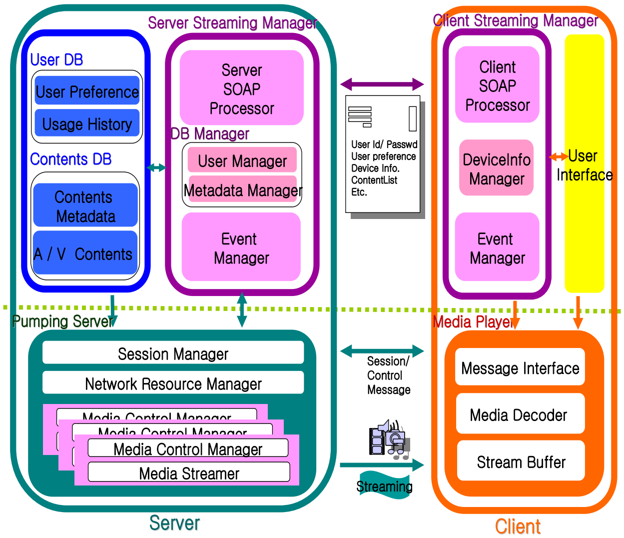

Figure 3. Mobile Streaming System Architecture

The SOAP message communication is implemented between the Server Streaming Manager and Client Streaming Manager, and TCP/IP is used for streaming content between the Pumping Server and the Media Player. Figure 3 shows the module that constitutes the streaming system and respective functions.

2.1. User Interface

The user interface handles the user's input via devices such as the keyboard or mouse. It gives the Client Streaming Manager information on a user's preferences, which includes the user's favorite actor/actress, producer, genres, content title, and key words entered through the keyboard when the user registered his or her preferences, as well as log-in information (user ID and password) when the user logs on. Also, it sends Trickplay events such as Play, Pause, Stop, Quit, etc., which were entered through the mouse to the Event Manager while content streaming is in progress.

2.2. Client Streaming Manager

2.2.1. Event Manager. The Event Manager controls how content is displayed on the media player according to the Trickplay event from the User Interface. Especially, it sends the Client Soap Processor's stop point, which is the point the media is stopped or interrupted.

2.2.2. Device Info Manager. The Device Info manager determines the client device information such as media codec, CPU capability, display size, and other such factors from the media player codec or devices' system information, then sends them to the Client Soap Processor.

2.2.3. Client Soap Processor. The Client Soap Processor parses the textual information from the client into XML by fitting the SOAP message format, and sends it to the Server Soap Processor through SOAP. It requests the server for connection by sending user information, user preferences, and client device information to the Soap Processor. It controls by creating and nullifying the media player that can display content by streaming. Also, the stop point from the Event Manager is to be sent to the Server Soap Processor when the user logs off.

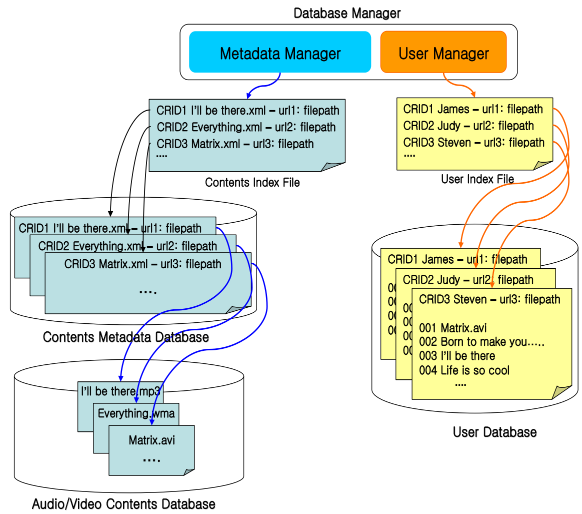

Figure 4. Repository for User Information and Contents

2.3. Media Player

The media player, created from the Client Soap Processor, requests the Pumping Server to start the service, and chooses the proper decoder and display for the streaming media content from the server.

Figure 4 shows how user information and content are efficiently stored and managed for seamless data streaming based on the user's preferences and device information from the server into Repository.

2.4. Sever Streaming Manager

The Server Streaming Manager handles all the client's connection processing and management, and is always ready to receive client request with its server demon.

2.4.1. Event Manager. With the content request from the Server Soap Processor, the requested Content Database file route is given to the Pumping Server.

2.4.2. Database Manager. The Database Manager is composed of the User Manager and Metadata Manager. The User Manager manages the Mapping Table that stores the user ID, password, and device information. The Mapping Table indexes the user's preferences based on the User DB and Usage History, and manages client device information. User Manager also handles the Active User Table that manages information on the currently connected user. It records and manages user conditions, such as content information that the user currently uses, and content stop point that indicates the exact point the content was stopped.

It manages files in the Content DB by indexing the Content Metadata. The Content Metadata with metadata on Audio/Video contents file and metadata on respective Audio/Video contents constitute separate databases.

Content Metadata is a file in XML, which includes content details that allows the Metadata Manager to do its job by creating a Mapping Table that indexes contents of the Metadata for faster search.

Table 1. Genre Type to Classify User Preference

Genre Type	Details
Fiction	Romance, Action, Fantasy, Erotica, Comedy Drama
Sports	Football (soccer), Baseball, Basketball, Golf, Swimming, Skiing
News	Daily news, Foreign/International, Local/regional, Weather forecasts
Amusement	Game Show, Quiz/Contest, Variety Show, Comedy

2.4.3. Server Soap Processor. The Server Soap Processor handles the backward processing of Client Soap Processor. It gets information in text by XML Parsing of SOAP messages transmitted from the client through SOAP.

User information and preferences, and client device information received from SOAP is stored in the User Manager. Content List created from the Database Manager is sent to Client Soap Processor to choose one content. Upon every request of media streaming of the Client Streaming Manager, the Pumping Server is created in the thread to stream the selected media to the respective client. It sends the event between the Pumping Server and the Client Streaming Manager.

2.5. User Database

Table 1 represents genre details used in this study based on the user preferences content classification provided by TV-Anytime Specification version 1.1. It classifies Genre Type as a higher classification then Details as a lower classification.

User History DB stores/manages content list and the timestamp of the user's most recent log-on to the server and received streaming. It is managed by Metadata and follows the schema defined in TV-Anytime Specification 006.

2.6. Content Database

The database constitutes of content file and content Metadata with details such as actor/actress, director, producer, genre, content list, key word, and storyline to each audio/video content and audio/video contents file in various formats such as AVI, ASF/WMV/WMA, MPEG-2, DivX that the database possesses. Used Metadata Schema follows the Schema defined in TV-Anytime Specification 006.

2.7. Pumping Server

The Media Pumping Server created in the thread from the Server Soap Processor upon the client's request. With the file route from the Event Manager, content is buffered from the Content Database and streams data to the client. The process of content streaming and streaming via Trickplay between the Pumping Server and Media Player is similar to the existing streaming system.

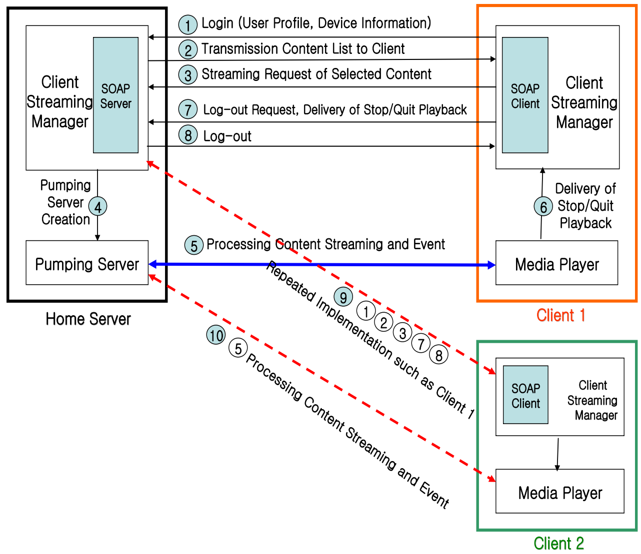

Figure 5. Seamless Streaming Service Using Stop Point using UML

3. Seamless Content Streaming and Display Implementation

The server streaming manager implements the content streaming service to many multimedia clients based on user preferences and clients' device character information on the home network and user identification, and seamlessly provides multimedia content even if users switch client devices. Figure 5 shows the seamless streaming service procedure, streaming services using stop points and how media playback is implemented.

Information and contents is saved in the database. Stop streaming information can be divided into three categories.

- Frame number of multimedia content received via streaming
- Playing time information of multimedia content received via streaming
- Ratio calculated by the frame number and playing time of multimedia received via streaming

3.1. Streaming Resuming Request

The server agent receives a resumption request regarding the streaming that was stopped from client 2. Similar to the process with client 1, client 2 identifies the user to know if the user is allowed to use the streaming service. It identifies if the client 1 user and client 2 user are the same, and if they are, the part of the media data that is yet to be sent to client 1 when it stopped is continued to be provided via streaming to client 2.

In resuming content streaming, the user of client 1 and the user of client 2 are identified, and if both users are actually one and the same, streaming is resumed with client 2 with the content after the stop point of client 1. The device character information, such as media codec, CPU capacity, display size, bit transmission rate, media playable time, and network bandwidth is stored in the Mapping Table, making it possible to know the media codec of the multimedia content that had been sent via streaming to client 2 or to know if the CPU or the display can handle the playback.

- Streaming is implemented after playback is determined possible.

- If playback is not possible, client 2 will be informed or will be given a recommendation on other playable content in a created content list.
- If there is no codec, it is possible to stream by transcoding to the media format of client 2. A transcoding module plugin is required.

Multimedia content that was stopped will be displayed on top of the user interface and a pop-up window appears asking if data streaming is to be continued. If the user selects "consecutive playing," the server agent is notified and streaming resumes. If the user does not select consecutive playing, content will be started all over from the beginning. Figure 7 shows the consecutive playing of the movie 'Matrix' selected by Client 2 user.

Figure 6. Example of UI Provided by Client 2

3.2. Log-out Request and Repeated Implementation

If the request to stop streaming through client 2 is received, and the server agent commands to stop the streaming, the pumping server thread becomes null and only the demon server operates. The Log out request and repeated implementation can be seamlessly provided to many clients on the home network with this process.

4. Conclusion

This study proposes the implementation of a server–client streaming system that enables the seamless data streaming of large amounts of multimedia content even if the user switches to another device or client. The proposed system is also capable of sending highly customized content to different users or different device environments on a home network by storing/managing user preference information, device character information, and the content stop point on the home server. Further studies on determining the preferred content by automatically figuring out the user's preferences and device character information are necessary.

Acknowledgements

This study was supported by the Kyungil University Grant.

References

[1] A. Roopalakshmi and A. kumar, “Hybrid Chaining Scheme for Video-on-Demand Applications Based on Popularity”, Proceedings of the 8th conference on Applied informatics and communications, **(2008)**, pp. 546-553.
[2] H. S. Guruprasad, M. Dakshayini, P. Jayarekha, H. D. Maheshappa and P. Geethavani, “Dynamic Buffer Allocation for Video-on-Demand System Based on Popularity”, *NCIIT-2006*, **(2006)**, pp. 13-17.
[3] B. K. Lee and J. Y. Song, "Home Server and Appliance Terminal Technology for Development of Digital Home", Korea Information Processing Society, vol. 11, no. 3, **(2004)**, pp. 46-54.
[4] UCL RTP/RTCP library, http://www.mice.cs.url.ac.uk/multimedia/software/common.
[5] Schulzrinne, et al., "RTP : A Transport Protocol for Real-time Applications", RFC 1889, **(1996)** January.
[6] http://www.real.com.
[7] http://www.microsoft.com.
[8] http://www.w3schools.com/soap/default.asp.
[9] SOAP Version 1.2 Part 1, http://www.w3.org/TR/2003/REC-soap12-part1-20030624/.
[10] “The TV-Anytime Forum Specification”, http://www.tv-anytime.org.