APPLICATION LEVEL SESSION HAND-OFF MANAGEMENT IN A UBIQUITOUS MULTIMEDIA ENVIRONMENT

Letian Rong, Ian Burnett
CRC SIT (Smart Internet Technology), University of Wollongong, Wollongong, Australia

Keywords: Universal multimedia adaptation, Mobile multimedia applications, Internet services and applications

Abstract: This paper focuses on one of the most important aspects of user mobility in a ubiquitous mobile environment: application session hand-off management. Here we use the term Session Mobility to define the ability of handling application session hand-offs among mobile devices. The paper summarizes the current research in the field and addresses the important facets and the missing "ingredients" of these treatments. We then propose an architecture to support and manage application session transfers based on the MPEG-21 multimedia framework. This takes advantage of Digital Items and adaptation metadata to provide a standards-based approach to the problem. Finally, we validated our framework using a test-bed which provides for dynamic multimedia adaptation.

1 INTRODUCTION

Mobile devices are becoming increasingly popular. In the near future, users will be using a series of devices rather than a single but will also be using them while changing locations rapidly. This is often called personal mobility and is therefore the aim of allowing users to consume services through multiple devices while changing location. This paper focuses on one area of personal mobility, Session Mobility (SM) management, which deals specifically with the issues involved in consuming a session continuously or later, following storage of session state, in a device independent manner.

2 RELATED WORK

To date, the widest published projects in the personal mobility area have been that of the Berkeley ICEBERG project (B. Raman et Al., 1999), Stanford's Mobile People Architecture (MPA) (M. Roussopoulos et Al., 1999), AT&Ts Telephony Over Packet networkS (TOPS) (N. Anerousis et Al., 1999), the Session Initiation Protocol (SIP) (M. Handley et Al., 2001), and the Hand-off Manager middleware from University of Illinois and Purdue University (Y. Cui et Al., 2003).

The first four of these are similar in that they were designed primarily for voice mail, email and teleconferencing related services. The main feature of this type of application is that they are usually short-lived and require only basic resource adaptations and modality conversions (voice-to-text, voice-to-video with fixed configurations). Thus, the main focus of these systems is to deal with issues involved with user location and they are not designed for the complex adaptation procedures that might be required in multimedia applications.

Cui et al, (Y. Cui et Al., 2003) developed a more extensive middleware framework to manage the session hand-off issues of personal mobility missing from the above systems. Importantly, it was also designed to target multimedia applications. Their system supports personal mobility by inserting hand-off manager middleware between multimedia applications and the underlying network infrastructures. They also introduced a User Metadata Server for storing session state related information. Their hand-off manager is then responsible for transferring the session state of the primary application to the hand-off manager of another application directly or alternatively to instruct the User Metadata Server to perform the operation. They also introduced a QoS mapping algorithm to facilitate resource transcoding during session hand-offs.

There are, however, several limitations to Cui et al's contribution (Y. Cui et Al., 2003). First the transcoding-based QoS adaptation approach from the hand-off manager middleware is too complex

Rong L. and Burnett I. (2004).
APPLICATION LEVEL SESSION HAND-OFF MANAGEMENT IN A UBIQUITOUS MULTIMEDIA ENVIRONMENT.
In *Proceedings of the First International Conference on E-Business and Telecommunication Networks*, pages 223-229
DOI: 10.5220/0001394602230229

and is limited to one specific element of the multimedia content adaptation process (i.e., transcoding). A more comprehensive adaptation process is therefore required to support dynamic adaptation of multimedia resources and sessions. This is particularly important during the process of session hand-off amongst a broad variety of devices, and the middleware should be able to support universal multimedia in particular, user preference, terminal capability, network capability, natural environment characteristics and other requirements.

In addition, the Cui approach takes a session sender approach that requires a session to be stopped before the hand-off of the session can occur. In certain cases the consumer might like to have data seamlessly streamed from one device to another by initiating it from the session receiver. On other occasions the consumer might like to continue the same session on multiple devices. For example, a person might like to continue watching a movie on his PDA while his kids continue watching the same movie on the TV. Cui's approach does not facilitate these possibilities.

Finally, all reported approaches to session mobility so far deal only with one to one session hand-off. In real life situations, however, it is quite possible that the hand-off will involve N to M session transfers. For example, the session state of a presentation may need to be synchronized to other devices so that others can either carry on the presentation or view the updated presentation session. It is also important that a device, on receiving several sessions from multiple session senders try to conserve their previous application session states (i.e., window size, window position etc.,). This requires a more complex session hand-off manager.

The architecture proposed in this paper addresses these issues and facilitates session hand-off with particular focus on the following requirements:

1. Enabling sessions to be transferred amongst a large variety of devices and allowing those sessions to be adapted to the new environment dynamically. This involves resource(s) adaptation according to the specific usage environment and session states. Furthermore, a mechanism is required to identify which application data are session related (e.g., how long a movie has been playing) and which are usage environment related (e.g., the resolution of a movie). This would enable the device to identify the application data which should be transferred to the session receiver.

2. Enabling sessions to be stored remotely for latter consumption or to be transferred seamlessly between devices. The management software should intelligently support those options.

3. Enabling N to M session transfers and managing multiple received sessions in an appropriate way so as to conserve their previous session states.

3 SESSION MOBILITY ARCHITECTURE

The proposed session mobility architecture (see Figure 1) was designed based on the MPEG-21 multimedia framework. This approach takes advantage of Digital Items and adaptation metadata to provide a standards-based approach to the problem. The architecture consists primarily of two parts: the M21 middleware which is installed on all the consumer devices and the SM server. We shall now discuss MPEG-21 and the roles of these two parts of the architecture.

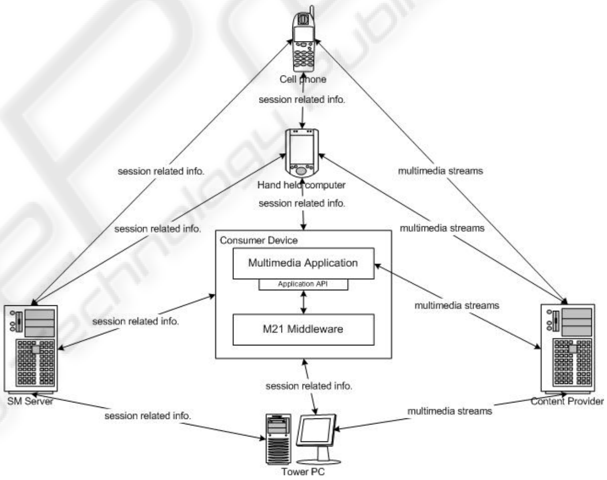

Figure 1: Session mobility architecture

3.1 MPEG-21

MPEG-21 is a new multimedia framework from the Moving Picture Experts Group (MPEG) that supports multimedia access and delivery using heterogeneous networks and terminals in an inter-operable and highly automated manner (J. Bormans et Al, 2003). MPEG-21 addresses the requirements of Universal Multimedia Access (UMA) (A. Perkis et Al., 2001) by providing a normative open framework for multimedia delivery and consumption.

The fundamental unit of distribution and transaction in the MPEG-21 framework is the Digital Item (DI). It can be considered as a structured digital object which consists of resource(s) (e.g., a photo album, a web page) and related information for the manipulation of the resource(s) (e.g., terminal capabilities, intellectual properties). DI is represented as a Digital Item Declaration (DID) in MPEG-21 through the Digital Item Declaration Language (DIDL) (V. Iverson et Al, 2002) which conforms to the XML standard. While a DI can have wide ranging contents (e.g. Rights information), for our purposes we consider a DI to contain resource(s), i.e. a list of choices that correspond to the various adaptation aspects of those resources and Digital Item Adaptation (DIA) information. DID (ISO 21000-2) has already become an International Standard.

The MPEG-21 Digital Item Adaptation (DIA) Tools are a collection of descriptions and format independent mechanisms that steer the multimedia content adaptation process (A. Vetro et Al., 2004). The descriptors are represented in XML and can be either wrapped in a DID or be used independently. Currently the DIA Tools are clustered into eight major categories (see Figure 2) and our architecture mainly uses three of those tools: Usage Environment Description Tools, Session Mobility and DIA Configuration Tools.

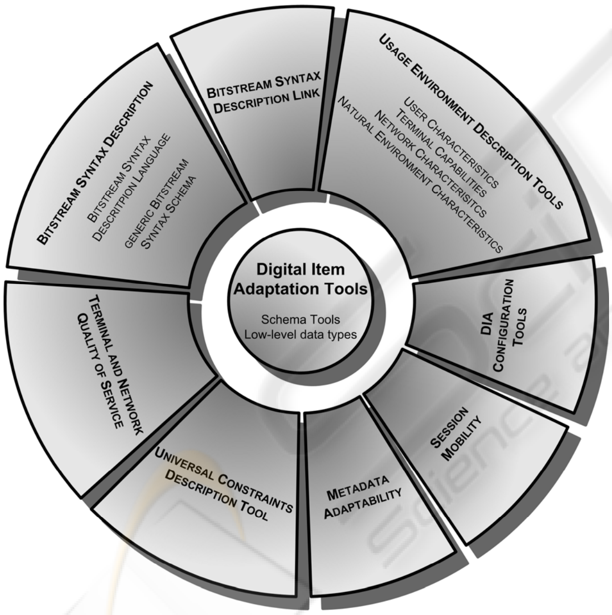

Figure 2: DIA tools architecture (A. Vetro et Al., 2004)

The *Usage Environment Description Tools* includes descriptors for the various dimensions of the usage environment. Those are user characteristics, terminal capabilities, network characteristics and natural environment characteristics.

The *DIA Configuration Tools* specifies how and where the related usage environment information can be used for the adaptation of DIs, also it identifies which configuration state of resources are session related or should be configured again during session hand-offs.

Session Mobility describes the application session state information that pertains to the consumption of a Digital Item in real time, allowing a Digital Item to be consumed continuously when it is transferred from one device to another.

The usage of these related tools are explained in the following section. Since the scope of MPEG-21 is very broad, interested readers are referred to references for information related to MPEG-21 (J. Bormans et Al, 2003) (A. Perkis et Al., 2001).

3.2 M21 middleware

M21 is a middleware application developed at the University of Wollongong that implements the base concepts of MPEG-21. It was used in our previous work (L. Rong and I. Burnett, 2004) for enabling multimedia contents to be adapted to various devices/terminals in a ubiquitous environment according to usage environment attributes. Its structure is illustrated in Figure 3.

In the multimedia content adaptation approach, the DI was used as a “Menu” to contain a link or links to multiple pre-existing variations of resources and a list of choices which correspond to the resource or which can be selected so as to configure those resources further. Furthermore, the DIAC related information is also encapsulated in the DI to provide further guidelines on the adaptation process. The guidelines specify information on the location of adaptations e.g. at the consumer device, intermediate nodes or the provider, the types of descriptors required for adaptations and how the choices should be selected (by the device automatically or by the user manually). The consumer then requests a multimedia resource by first requesting its corresponding Digital Item and then performs the resource related configuration and selection according to his/her usage environment attributes and the “guidelines” in the DI. The usage environment attributes are expressed in the Usage Environment Description Tools. The consumer then sends that configuration and selection information as the second request to the provider who in turn, performs the required resource related adaptations on their side (according to the same DI) before sending the resource to the consumer. The M21 middleware is installed on both the consumer and provider side to perform the DID parsing, DIA processing and resource adaptation processes. Also it handles launching of applications to “play” the resources on the consumer side.

We adopted the existing M21 middleware and the content adaptation approach into the session mobility architecture. This gives us two main advantages. The first advantage is that we can use the existing content adaptation functionality to reconfigure and adapt resources as they are transferred to a session receiver device. The second advantage is that the Digital Item is used as the fundamental unit for multimedia delivery in the session state transfer process. This broadens the concept of session state because a DI can then consist of multiple resources and hence the session state related to a DI can contain multiple media session states – each associated with media resources in the DI. Thus, we define the session state of a DI as a user's current state of interaction with a DI. An example of this concept would be the session state of a digital CD album which contains audio clips, movies and photos (i.e., a DI with multiple resources). The session state of the DI (in our case the CD) would contain the audio track playing position, the movie playing position, the photo being displayed etc. The session state of a DI is inserted into the DI as a SM schema of the DID Adaptation Tools when it is transferred to the session receiver device. Also DIAC is used in the DI to identify which configuration state is session related.

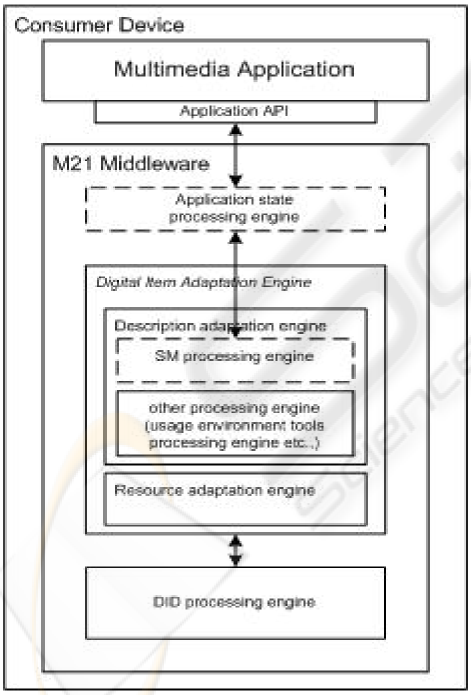

Figure 3: M21 middleware and its components

A SM processing engine and application state processing engine are thus added into the middleware to manage session hand-offs (see as the dot line boxes in Figure 3). The SM processing engine is used for generating and parsing session state information as SM schemas, and it also acts as the decision maker as to where sessions should be fetched from. Its decision making mechanism is explained in section 3.4. It should be noted here that the SM processing engine links directly with the application state processing engine by fetching application session information from it on the session sender side and passing this information to the engine on the session receiver side.

The application state processing engine is used to extract application session information through the application APIs on the session sender side and launching appropriate applications on the receiver side. As mentioned above, the engine connects to the SM processing engine to transfer the information, so that session states can be encapsulated/extracted from DIs as SM schemas. Again, we believe it is necessary to support multiple session transfers. The application state processing engine performs this task by launching appropriate applications with resources in their preserved session state and arranging those applications to reflect their previous application session states. It therefore takes advantage of the middleware architecture and only requires different application "drivers" to support various applications through their APIs.

3.3 SM Server

The second major part of our architecture is the Session Mobility (SM) server that resides in the network for storing, redirecting and processing multimedia sessions. It uses a database to store session related information as indicated in Table 1.

The SM server either transfers stored sessions directly to any session receivers or directs session senders to perform the operation. Its session steering mechanism is explained in the section below.

Table 1: Session Mobility server information

User ID	User identification
IP Address	IP address of the device
Application Name	Application for running the media
Resource Name	Name of media resource
Media Status	Playing, Paused or Stopped
Stored Session	Exist or empty

3.4 Multimedia session hand-off management

We categorize session hand-offs into two main types, session sender driven and session receiver driven. The two approaches are shown in Figures 4 and 5.

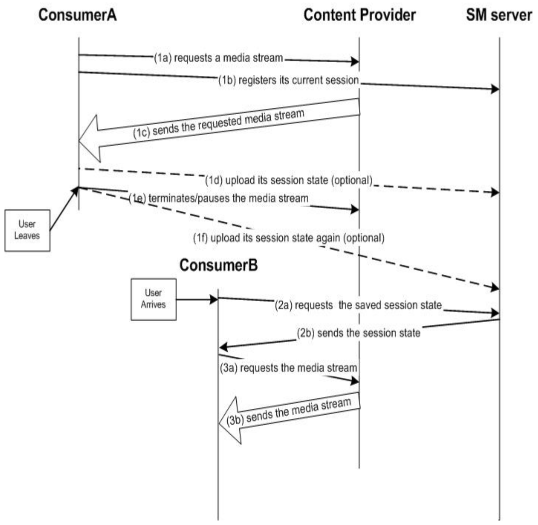

Figure 4: session sender driven approach

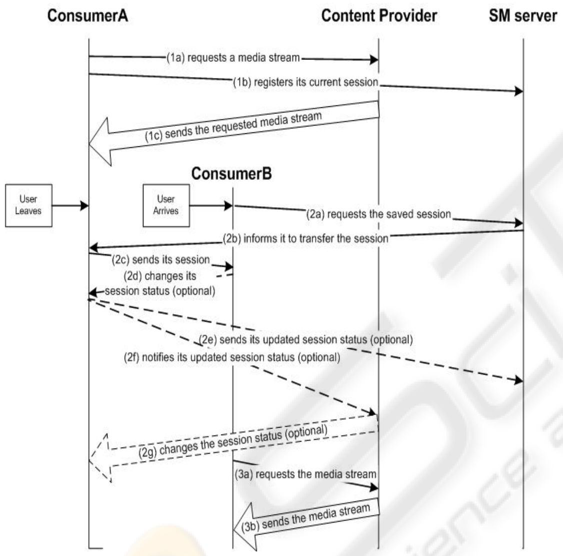

Figure 5: session receiver driven approach

Sender driven session hand-off is when the session sender steers the session hand-off process. In this case the sending terminal changes its current session status (i.e., play, pause or stop) before transferring its session to the session receiver or the SM server. The alternative, receiver driven approach, enables the session receiver to control the session status of the session sender and initiate the session hand-off process. The steps of each algorithm are shown in Figure 4 and 5.

In the steps detailed, a content provider is used in the case of streaming media sessions being transferred during hand-offs. Those steps however can equally be used for transferring local data without the content provider.

Both approaches have been incorporated into our current architecture and a simple algorithm is used by the SM processing engine for determining which approach it should use according to the current status of the media. If a media resource is currently playing and there are no corresponding sessions saved on the SM server, then the session receiver would contact the session sender to retrieve the media session directly. Conversely if the media session has already been saved on the SM server and it shows the media status as being stopped, paused or killed, then the session receiver would retrieve the session from the SM server instead. The advantages of facilitating both approaches in the architecture are that: 1. Consumers are given the flexibility to initiate session hand-offs from different devices; 2. It reduces SM server load by offsetting some session hand-offs to peer-to-peer session transfers (i.e., in the session receiver driven approach). Finally, several enhanced approaches from (Y. Cui et Al., 2003) can be easily modified to be incorporated into our proposals.

4 EXPERIMENTAL SETTINGS AND VALIDATION

The M21 middleware was installed on a test-bed as the middleware layer application on three separate computers and a fourth was used as both the SM server and RTP streaming server.

Two different applications were used in the experiment to demonstrate multiple session management. These were a RTP streaming client and a Java web browser. The RTP streaming program pair, a RTPTransmitter and RTPReceiver, have been used previously in our dynamic multimedia adaptation work (L. Rong and I. Burnett, 2004), and the Java web browser is implemented based on NetClue (NetClue Java web browser v4.2) APIs. All programs were written in Java for common application interaction and easy code deployment.

Both session sender driven and session receiver driven approaches with single or multiple session transfer(s) have been evaluated in the test-bed. The session sender devices were able to register their current sessions with the SM server and transfer their session to either the SM server or session receiver devices through the M21 middleware. The session receiver devices were then able to launch corresponding applications, adapt contents according to their usage environment attributes and resume processing of the Digital Item session. The session receivers can also effectively control the media status of the session sender through the M21 middleware as shown in Figure 6.

Furthermore the application session processing engine arranges and displays applications according to their previous application states. Currently the supported application session information includes window size, position, state i.e., maximized, minimized or user defined and size-to-resolution ratio.

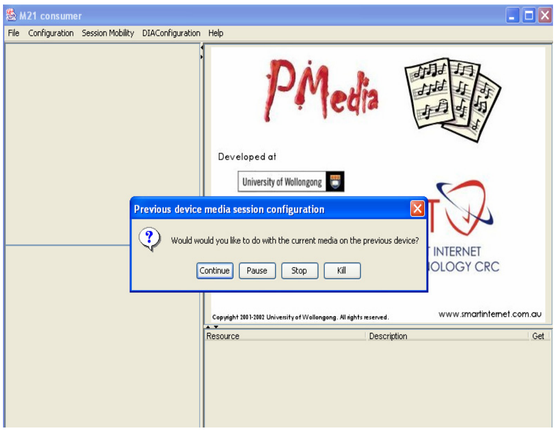

Figure 6: screen shot shows session receiver choosing media status of session sender through M21

In addition, the test-bed supports the following usage environment descriptors and choices for streaming medias: MediaTime, Coarse Language, CharacterSet and Resolution. The descriptors MediaTime and Coarse Language are transferred to the session receiver device as they are session related, while CharacterSet and Resolution are configured locally according to the capabilities of the session receiver. For the web browser, content adaptation was demonstrated by adapting web pages of different resolutions to various terminals according to their resolutions. The generation and processing overheads of MPEG-21 Digital Items are shown to be relatively low in the experiment. We used 83ms and 2ms on average for generating Digital Items with application session state information for the RTP streaming client and web browser respectively, and 279ms for parsing the DIs on the receiver end. The generation of DI for the RTP streaming client is relatively higher due to fact that session information is required to be extracted from the client program during media streaming.

5 CONCLUSION

In this paper, we proposed a session mobility architecture to improve the area of session hand-off management with the aim of targeting multimedia applications. There are some similarities between our work and that of Cui's. We believe that we have substantiated what is missing from their work and broader session transfer experience with MPEG-21.

This architecture enables session transfers to be performed through two different types of approaches (session sender driven and session receiver driven) and this then facilitates different types of session transfer needs. We also adopt the Digital Item concept to facilitate dynamic session adaptations to the session receiver, without complex content negotiation/matching algorithms. Further, an application state processing engine with different "application drivers" is used to manage multiple applications during session transfers. This can be easily expanded to cater for other multimedia applications by writing application "drivers" and incorporating them into the application state processing engine.

Several Digital Item Adaptation Tools (i.e., DIA Configuration Tools, Session Mobility and some of Usage Environment Description Tools), are the end-result of our work and they are now included in the Final Draft International Standard (FDIS) version of DIA Tools and will become an International Standard in 2004.

As for our future work, we shall implement a more complex Session Mobility engine that uses more comprehensive mobility characteristic descriptors (Z. Sahinoglu and A. Vectro et Al., 2003) for facilitating session hand-offs and their adaptation processes under more complex mobility situations.

REFERENCES

B. Raman, R. H. Katz and A. D. Joseph, *"Personal Mobility in the ICEBERG Integrated Communication Network"*, Report No. UCB/CSD-99-1048, May, 1999.

M. Roussopoulos, P. Maniatis, E. Swierk, K. Lai, G. Appenzeller and M. Baker, *"Person-level Routing in the Mobile People Architecture"*, 2nd USENIX Symposium on Internet Technologies and Systems (USITS 1999), October, 1999.

N. Anerousis, R. Gopalakrishnan, C. R. Kalmanek, A. E. Kaplan, W. T. Marshall, P. P. Mishra, P. Z. Onufryk, K. K. Ramakrishnan, and C. J. Sreenan, *"TOPS: An Architecture for Telephony Over Packet Networks"*, IEEE Journal of Selected Areas in Communications, 17(1), January 1999.

M. Handley, H. Schulzrinne, E. Schooler and J. Rosenberg, *"SIP: Session Initiation Protocol"*, RFC2543, April, 2001.

Y. Cui, K. Nahrstedt and D. Xu, *"Seamless User-level Hand-off in Ubiquitous Multimedia Service"*, Multimedia Tools and Applications Journal, Special Issue on Mobile Multimedia and e-Commerce, Kluwer, 2003.

J. Bormans, J. Gelissen, and A. Perkis, *"MPEG-21: The 21st Century Multimedia Framework"*, IEEE Signal Processing Magazine, March 2003.

A. Perkis, Y. Abdelijaoued, C. Christopoulos, T. Ebrahimi, *"Universal Multimedia Access from Wired and Wireless systems"*, Birkhauser Boston, transactions on Circuits, Systems and Signal Processing, Special issue on Multimedia Communications, Vol. 20., No.3, 2001, pp. 387-402.

V. Iverson, T. Schwartz, Y. Song, R. Walle, E. Santos and D. Chang, *"MPEG-21 Digital Item Declaration FDIS"*, *ISO/IEC JTC1/SC29/WG11/N4813*, May, 2002.

A. Vetro, S. Devillers and C. Timmerer, "ISO/IEC 21000-7 FDIS – Part 7: Digital Item Adaptation", ISO/IEC JTC1/SC29/WG11/N6168, January 2004.

L. Rong and I. Burnett, *"Dynamic Multimedia Adaptation with MPEG-21"*, Consumer Communication and Networking Conference (CCNC 2004), January, 2004.

NetClue Java web browser v4.2, NetClue, www.netcluesoft.com.

Z. Sahinoglu and A. Vectro, *"Mobility Characteristics for Multimedia Service Adaptation"*, EURASIP Journal of Image Communications, Special Issue on Multimedia Adaptation, October, 2003.