Recording and Delivery of HbbTV Applications

Jean-Claude Dufourd
Telecom ParisTech
37 rue Dareau
75014 Paris, France
+33145817733
dufourd@telecom-paristech.fr

Stéphane Thomas
Telecom ParisTech
37 rue Dareau
75014 Paris, France
+33145818057
thomas@telecom-paristech.fr

Cyril Concolato
Telecom ParisTech
37 rue Dareau
75014 Paris, France
+33145817991
concolato@telecom-paristech.fr

ABSTRACT

As the newly created standard for interactive television, HbbTV (for Hybrid broadband broadcast TV), is getting traction, the problem of recording TV services takes on new dimensions with interactive and hybrid services. This paper explores the different issues raised by the recording and on-demand playback of broadcast HbbTV services containing interactive applications. Adaptive streaming tools provide part of the solution. Guidelines to application developers and small extensions to HbbTV are also proposed. We describe possible implementations based on MPEG-2 TS and discuss an alternative based on ISOBMFF.

Categories and Subject Descriptors

H.3.5 [**Information Storage and Retrieval**]: Online Information Services – *Commercial services, Web-based services;* H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems; H.5.4 [**Information Interfaces and Presentation**]: Hypertext / Hypermedia – *User issues*

General Terms

Algorithms, Experimentation, Standardization.

Keywords

HbbTV, interactive TV applications, catch-up TV services, time shifting, adaptive streaming, MPEG DASH

1. INTRODUCTION

A new standard for interactive TV, Hybrid Broadband Broadcast TV (HbbTV), is getting more and more traction since its inception in 2009 and standardization by ETSI in June 2010 [1]. This standard differs from previous attempts in that its scope includes both broadcast and broadband delivery mechanisms. However, one of the current limitations of this standard is that it only specifies how interactive applications (composed of HTML, CSS and JS in addition with other media resources) are delivered within a broadcast service. In this paper, we are interested in the processing of HbbTV services either for time-shifting/recording or for catch-up service. It should be noted that we distinguish HbbTV service from HbbTV application, where the service is the set of audio-visual content synchronized with the associated application(s), the application containing all the interactivity.

HTTP Streaming is also becoming more and more popular for delivering audiovisual content over broadband links. The same technologies are being worked on in a large standardization effort (MPEG DASH [2], 3GP AHS [3]). The overlap of underlying technologies between HbbTV and DASH, and in particular the use of MPEG-2 Transport Stream (TS) or the ISO Base Media File Format (ISOBMFF) suggests considering the use of HTTP Streaming for the delivery of interactive HbbTV services.

Our goal in this paper is to analyze the different issues raised by the recording and on-demand playback of broadcast HbbTV services containing interactive applications. The analysis is carried from three different perspectives. We first look at the management of resources, then at transport and carousel issues. Finally, we look at the modifications that need to be performed at the application level to ensure its correct playback.

Recording of interactive TV services has been explored before at the time of MHP [5][6], with some similar constraints and solutions, but as MHP did not envisage broadband links, the management of resources as discussed in 3.1 was not studied. [7] describes a format for the recording and playback of interactive applications in the context of video conferencing where no data carousel is necessary. [8] describes adding live annotations to a multimedia document, which has similarities with the capture of Stream Events into a timed data stream described in section 3.2.2. By describing the addition of comments to a TV program, [9] also shows that our problems have similarities with live program annotation. Finally, the repurposing of broadcast interactive applications over DASH was studied in [4] but only in the context of MPEG-4 BIFS applications.

This paper is organized as follows. Section 2 presents use cases and requirements. Section 3 is the core of the paper and presents our study of the different problems, while Section 4 discusses possible implementations and associated limitations. Finally, Section 5 concludes this paper and presents future work.

2. USE CASES AND REQUIREMENTS

We are considering the use case of delayed playback or time shifting, which covers two variants: catch-up TV are an organized version of delayed playback which many TV channels offer or want to offer; and user-recorded services.

The goal of catch-up TV is to make available to the user a version of the target content that will produce the same effects as the playback of the broadcast stream, at minimal cost. The process may repurpose the broadcast stream or operate on its basic

EuroITV'11, June 29–July 1, 2011, Lisbon, Portugal.
Copyright 2011 ACM 978-1-4503-0602-7/11/06...$10.00.

components and create a version of the content in another packaging format.

The problem of recording content as it is broadcast so that it can be played back later is similar to that of creating a catch-up service. The goal is the same. The recording process has the broadcast stream to work on, as well as broadband resources.

The requirements that we have derived for solutions to the above use cases are:

R1. Compatibility with existing HbbTV terminals: any solution to the above use cases should work on existing HbbTV set-top-boxes or TVs, or require minimal extensions.

R2. No loss of service interactivity: if there was interactivity in the broadcast service, the same interactivity should still be available.

R3. Always show the same information: the information shown to the user should be the exact same information as when the content was broadcast, regardless of whether the information changed during the broadcast or since then.

R4. Keep the synchronization of interactive events with the media: if when broadcast, the interactive application appeared or changed at a certain point in the media, the same interactive application should appear or change at the same point in the media when played back.

R5. Simplify the process: any solution should be as simple as possible and consume as little resource as possible, so as to allow concurrent processing on the device; for example, it should be possible to record a show while playing another show, without impact on the playback quality of experience.

R6. Improve service interaction: since the service is played back over a bidirectional medium, it should be possible to add features not possible in broadcast, such as random access, VCR-like controls, etc.

R2, R3 and R4 are understood to be applicable unless the service author decides otherwise.

3. PROBLEM ANALYSIS

In this section, we provide an analysis of the problem of the repurposing of broadcast interactive services for the use cases described in Section 2. We structure this analysis by considering first resource issues, then transport and carousel issues and ending with high-level application issues.

3.1 Resources

Concerning resource formats, we assume that the same coding formats for audio, video or metadata content are available in both broadcast and broadband modes. If it was not the case, a media transcoding step would be necessary, but such transcoding is out of scope of this paper.

Resource issues in HbbTV emerge from the combination of hybrid delivery and interactivity. Interactive HbbTV applications typically consist in HTML, CSS and script files, together with associated resources, such as images, clips, fonts, XML data, etc. These files and resources may be delivered in broadcast within the object carousel, or in broadband. It is the application designer's responsibility to make sure that all resources are available at the time of the broadcast, but the continued availability of resources delivered in broadband is not guaranteed. This is the first resource relative issue. To record an HbbTV service and ensure R2/R3, we need to record the resources delivered in the object carousel and the resources delivered in broadband whose continued availability is uncertain.

The second issue is that there is no list of resources of the HbbTV application. An HbbTV application is signaled in the Application Information Table (AIT), and the table entry includes the URL of the root document and the domain name. In order to recreate the list of resources necessary to the application, we analyze recursively the root document to find links to resources. The main issue here is that resource URLs may be computed, in which case the analysis will usually fail to get a resource URL. As a result, our first guideline to application developers is: ***(G1) if you want your application to be recording-friendly, do not compute URLs***. Another welcome solution to this problem would be for HbbTV to define a manifest for HbbTV applications, in the spirit of the manifest of widgets, defined in section 7 of [10]. This manifest would be the perfect place for information about the longevity of broadband resources.

We have to limit the list of resources to the domain boundary defined in the AIT, and to a certain depth of recursion. In the case of user recording, the amount of data that needs to be stored may be too large for a set-top-box with limited resources, if the application has links to many pages. We have to let the user decide how much data will be stored in the terminal.

When played back, the application needs to point to stored resources. There are two solutions to achieve this: 1) we rewrite the URLs in the application to point to stored resources; 2) we define an indirection mechanism similar to that of the iloc box of ISOBMFF [11]. The iloc box contains a table with two columns: a resource URL used in the application, and the actual location of the corresponding resource. The browser, before fetching any resource, needs to check that table: if it finds the URL in the table, it fetches the actual location; otherwise, it goes on fetching the original URL. Depending on the context, we may have to use one solution or the other, e.g. if the application is signed and cannot be modified, we have to use the indirection solution.

We also have to consider the possibility that some broadband-delivered resources may change frequently during the broadcast. An example is a file containing the score of a sports match, which is polled by the script of the application. It can be argued that this file should be delivered in the object carousel. However, the presence of such a resource in broadband would make an accurate playback of the application impossible in the user recording case: it would be impractical to analyze the application to detect the polling frequency, specially since the polling may be dependent on user interaction; running the application in background while recording would not be guaranteed to work either, regardless of problems of competition for device resources in case the device is playing another channel while recording. So this leads us to a second guideline to application developers: ***(G2) do not use changing resources delivered in broadband, use the carousel for such resources***.

3.2 Transport and Carousel

We concentrate now on the multiplexing formats because they have a direct impact on catch-up services. Since the HbbTV and DASH standards both refer to the MPEG-2 TS and the ISOBMFF, we study only these two formats.

3.2.1 Use of MPEG-2 TS

Given that most broadcast digital TV systems, including those specified by the HbbTV standard, rely on MPEG-2 TS for the delivery of interactive applications, this technology is a natural choice for recording. However, some issues must be taken into account. First, the HbbTV standard only requires the support of Single Program Transport Stream (SPTS) in broadband delivery. Therefore, recording equipment must transform the input data delivered in broadcast as Multiple Program Transport Stream (MPTS) into an SPTS, and in particular remove unnecessary programs and fix the timing, such as time stamps and program clock reference values, as detailed in [5]. Second, the delivery format should enable random access. MPEG-2 TS is not meant for efficient random access, but DASH indexing tools can compensate for that. One option is that the TS is segmented and the DASH manifest indicates that each segment starts with a random access point. Another option is to have the DASH manifest point to a separate file documenting all locations of random access points in the TS. Apart from these aspects, from a transport format view, all the signaling data can remain unchanged.

A carousel is a periodic delivery of a set of information, with versioning information. In broadcast environments, it enables late users to tune in, retrieve and possibly update, in a timely manner, the non-streamed elements of the program, such as the interactive application. When recording to MPEG-2 TS, the received carousel may be kept as it is and stored in the output TS, possibly filtered of data intended for non-recorded programs.

3.2.2 Use of ISO Base Media File Format

The ISOBMFF is also an interesting choice for the recording of HbbTV applications because of its ability to store timed and non-timed media data and metadata, with efficient random access features and because the DASH standard enables efficient streaming of ISOBM files. However, also for this format, some issues should be considered. The ISOBMFF does not support 1) the signaling of the HbbTV application and in particular the delivery of the Application Information Table (AIT); 2) the delivery of Stream Events synchronously with the associated audiovisual content; and 3) the object carousel.

A possible solution for the signaling is the storage of the XML AIT in a "meta" box. Similarly, the HbbTV standard relies on signaling of timed events, related to the audio/video content, by means of the so-called Stream Events. Stream Events and object carousel can be easily stored in a timed data track, together with their associated times as computed by the presentation time of the A/V streams at the moment the event or object was received. This is a lighter alternative to the SMIL solution suggested in [8].

However, HbbTV players will not recognize such augmented ISOBM files as containing HbbTV services. An extension of HbbTV is required for such files to be used in the context of catch-up TV, extension consisting in:

- the recognition of the ISOMB file as a recording of HbbTV program with interactive applications, triggering the extra behavior; this could be signaled by the use of a new brand in such ISO files.
- the retrieval and processing of an AIT equivalent at the beginning of playback;
- the retrieval and processing of the Stream Events track and of the object carousel track, both synchronized with the A/V streams, to ensure a presentation as close as possible to the original broadcast presentation.

We manage to avoid the definition of an external file as the MHP info file in [6], yet there is no way to avoid the extension of the behavior of the player. To conclude on the possible use of ISOBMFF for recording HbbTV services, in a pure HbbTV context, the cost of the extension of the player to deal with augmented ISOBMFF files is not justified when considering the much cheaper solution of using MPEG-2 TS. It would only make sense to use ISOBMFF for recording HbbTV services for use on non-HbbTV devices, such as mobile phones and tablets, for example in the context of convergence within the home.

3.3 Application Handling

In the process of recording the content, we have dealt with resources, then transport and carousel, we can now process the application. From an application level perspective, we see three issues to be dealt with when repurposing applications: management of resources, described in 3.1; programming interface changes and application relevance handling.

3.3.1 Application Programming Interface changes

One aspect we have to consider is the interaction of the application with the video object. As the service will be delivered in broadband or played locally instead of its original broadcast delivery, the controls that an application can have on the video object change. For instance, it is impossible to change the channel but on the other hand, it is possible to rewind or pause. Therefore, the programming interfaces used in the script of an application need to change: the video cannot be controlled through a video/broadcast interface, but through an AVcontrol interface. Calls to the video/broadcast interface will fail in broadband or local delivery. We need to rewrite any use of the video/broadcast interface into a use of AVcontrol.

In order to simplify this process, a welcome extension of HbbTV would be to deprecate both AVcontrol and video/broadcast interfaces and design a common superset interface for video, ideally in sync with the work on HTML5 video.

3.3.2 Application relevance

Finally we have to take into consideration the relevance of the application in a delayed playback situation. Some broadcast applications will stay meaningful, whereas others will be partly or completely irrelevant. Additional information on a type of rare fish just shown on video, or the biography of a movie actor will still be meaningful in playback, as will be all links to reference material. But let us consider a voting application: the first part of the application displays a question and a certain number of buttons to answer the question; as soon as the viewer has voted, the second part of the application shows the intermediate results of the on-going vote. It usually makes no sense to let the viewer vote in a catch-up scenario, as it is too late for the vote to be taken into account. However, showing the final results of the vote may be of interest to the viewer. So this voting application needs not be fully active when played back: it just needs to show the question and possible answers, then the final results. Finally live news widgets displayed during the show, e.g. a stock ticker, may not have any relevance during catch-up playback.

In the case of catch-up TV, the application may have to be manually modified. In the case of user recording, the device has no way to know the application relevance so a welcome addition to the HbbTV specification would be to include the DVB Application Recording Descriptor, specified in DVB TS 102 809, in the AIT, which holds related information.

It is possible for the application developer to improve the recording process: in the HTML of the application, there is a video object of type “video/broadcast” when the service is delivered in broadcast, and of type “video/mpeg” when the service is delivered in broadband or locally recorded. So ***(G3) the application can be designed to be seamlessly adaptable to both situations*** – another guideline for application developers. The recording system just has to modify the video object appropriately, and then the application can adapt to its execution environment by checking the type attribute of the video object.

4. DISCUSSIONS

In the previous section, we presented an analysis of the problem and proposed a set of solutions to the recording of HbbTV interactive services. In this section, we describe a few realistic implementation choices and their limitations.

A bare bone functionality allowing both catch-up and user recording is:

- consider that broadband resources stay available: no resource management is necessary;
- demux the MPTS, filter it, remux it as SPTS, keeping the received object carousel unchanged (or just filtered if the carousel is shared between a few channels);
- process the application for API changes and application relevance.

The advantages of such a system are that it is simple, and will work for time-shifting and relatively short-term playback. This may be acceptable for user-recording, but would not meet the quality requirement of a catch-up TV system for TV channels. Of our requirements, all but R2/R3 are satisfied, R5 particularly well, R6 through the use of DASH, but R2 is partial and R3 is left to chance.

The inclusion of resource management as described in 3.1 improves a lot the quality and longevity of recordings. Together with the adoption of our proposed guidelines (no computed URLs, no changing resources on broadband, design the application for both situations), an excellent compromise between the system complexity and the preservation of the application original functionality can be reached for a professional catch-up TV system. A catch-up TV system would then consist in a plain web server with, per recorded program:

- a SPTS processed by DASH tools (i.e. an MPD file and some TS segments, possibly a RAP index file), containing everything that was initially broadcast, A/V, AIT, stream events and object carousel;
- a set of application resources initially delivered in broadband that need to be cached because their “web” version may have changed.

With that addition, R2 and R3 are now satisfied.

Our guidelines already constitute a limitation on the freedom of application developers. Here is another limitation related to conflicts in the user interface. When playing back a recorded HbbTV program, we expect to have access to play-pause controls applicable to the recorded program. Imagine that the interactive application in the recorded program proposes to view a broadband video: during the original broadcast, the play and pause buttons applied naturally to the broadband video. During playback, there is an ambiguity on the use of those buttons. To resolve this, we can, for example, use a key to switch the focus of the controls between the broadband video and the general recording, together with a visual cue about which is currently active. This can be implemented either as an automatic augmentation of the interactive application, or as an extension of the HbbTV player behavior if this situation is deemed important enough.

5. CONCLUSION

In this paper, we have explored the different issues raised by the recording and on-demand playback of broadcast HbbTV services containing interactive applications. We proposed a set of HbbTV native tools for the base of the system, DASH tools for adding random access to MPEG-2 TS in the user recording case, and also adding HTTP streaming in catch-up systems. We proposed guidelines to application developers and small extensions to HbbTV. We described possible implementations based on MPEG-2 TS and discussed an alternative based on ISOBMFF.

Future work includes the implementation of a recording functionality in our in-progress open source HbbTV player, the checking of recording guidelines in our HbbTV application validation software, proposing the extensions described above for standardization in the next version of HbbTV, and exploring the playback of HbbTV services on non-HbbTV devices such as mobiles and tablets in the context of convergence in the home.

6. ACKNOWLEDGEMENTS

Part of this work was funded by the French project openHbb.

7. REFERENCES

[1] HBBTV, ETSI Technical Specification 102 796, “Hybrid Broadcast Broadband TV”

[2] MPEG DASH ISO/IEC 23001-6 Draft International Standard

[3] 3GPP Adaptive HTTP Streaming, TS 26.234 http://www.3gpp.org/ftp/specs/html-info/26234.htm

[4] C. Concolato, J. Le Feuvre and R. Bouqueau, “Usages of DASH for Rich Media Services”, *Multimedia Systems Conference*, San Jose, USA, February 2011

[5] L. Gan, F. Gan, J. Tan, T. Tan, and J. Shi, “A new A/V content storage format for recording interactive TV,” *Int. Conf. on Consumer Electronics,* 2005, pp. 41-42.

[6] P. Newton, D. Kelly, J. Tan, J. Shi, and L. Gan, “Recording interactive TV,” *IEEE Transactions on Consumer Electronics*, vol. 51, May. 2005, pp. 540-544.

[7] D.A. Vega-Oliveros, M. de G.C. Pimentel, and D. Santana Martins, “‘ This Conversation Will Be Recorded ’: Automatically Generating Interactive Documents from Captured Media,” *ACM Symposium on Document Engineering*, Manchester, United Kingdom: 2010, pp. 37-40.

[8] D.C. Bulterman, “Using SMIL to encode interactive, peer-level multimedia annotations,” *2003 ACM symposium on Document engineering - DocEng ’03*, 2003, p. 32.

[9] R.G. Cattelan, C. Teixeira, R. Goularte, and M.D.G.C. Pimentel, “Watch-and-comment as a paradigm toward ubiquitous interactive video editing,” *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 4, Oct. 2008, pp. 1-24.

[10] W3C, “Widgets 1.0: Packaging and Configuration”, available at http://www.w3.org/TR/widgets/

[11] ISO/IEC 14496-12:2005 | 15444-12:2005: "Information technology – Coding of audio-visual objects – Part 12: ISO base media file format" | "Information technology – JPEG 2000 image coding system – Part 12: ISO base media file format".