2-IMMERSE: A PLATFORM FOR PRODUCTION, DELIVERY AND ORCHESTRATION OF DISTRIBUTED MEDIA APPLICATIONS

J. Walker[1], D.L. Williams[2], I.C. Kegel[2], A.P. Gower[2], J. Jansen[3], M. Lomas[4] and S. Fjellsten[5]

[1] Cisco, UK; [2] BT, UK; [3] CWI, NL; [4] BBC, UK and [5] ChyronHego, Sweden

ABSTRACT

We describe an overview of the architecture of an evaluated multi-screen experience based on MotoGP sports content, which was developed using an object-based broadcasting approach. This architecture enables experiences to be implemented as distributed media applications consisting of components orchestrated by a cloud-hosted service platform. The MotoGP service prototype was evaluated by more than 90 MotoGP fans and assessments of the features and the experience have been broadly positive.

INTRODUCTION

This paper describes an evaluated system for delivering personalised TV experiences synchronised across multiple screens. We use an object-based broadcasting approach which allows a multi-device media experience to be conceived and presented as a single coherent experience. This releases TV service providers from the constraints of rendering a broadcast stream onto a single 16:9 frame, and allows them to develop compelling experiences that combine synchronised, interactive and customisable media objects across multiple screens.

We will introduce the platform, as developed for theatre- and sports-based service prototypes, which has been designed, developed and trialled in partnership with major content providers. The key features of the platform will be highlighted and the results from evaluations with users will be reported.

This paper provides an illustrated example of an object-based multi-screen personalised TV experience. In the correspondingly-labelled sections, we will describe: the benefits and challenges of the object-based approach; the distributed media application and service platform we have built to enable it; a service prototype based on MotoGP content from Dorna Sports; and the results from the evaluation of this service prototype.

The service prototypes and the technical infrastructure described have been developed within the EU-funded H2020 ICT project '2-IMMERSE' (1).

BENEFITS AND CHALLENGES

Object-based broadcasting, as discussed in Armstrong et al (3) enables TV programming to be customised to suit the available client devices, the available bandwidth and the capabilities of the devices. It allows broadcasters to provide content that can be adapted for display in a range of different environments enabling, for example, the presentation of sport in a public space like a fan-zone or a pub to be quite different to that presentation of the same sport within the home.

The object-based broadcasting approach can be applied across more than a single screen, to include all the devices that users may wish to involve in the presentation of the available content. The approach thus releases TV service providers from the constraints of rendering a broadcast stream onto a single 16:9 frame and pre-mixed audio stream, and allows them to develop compelling experiences that combine synchronised, interactive and customisable media objects across multiple screens.

There are, of course, significant challenges associated with conceiving and delivering personalised multi-screen experiences using object-based broadcasting. Television has had over 80 years to master storytelling within a single rectangular frame. Our key challenge has therefore been to conceive and deliver experiences that build on, and do not undermine or obscure, the technical craft and ingenuity that have been used to create compelling entertainment experiences. New production tools are therefore required to enable the definition of rules that control the way objects may be arranged on the available screens and to schedule when such objects may be presented.

Further challenges are posed when delivering live productions using object-based broadcasting. It must be possible to integrate the new tools with the live workflows of graphics operators, vision and audio mixers so that objects can be made available in the multi-screen environment with the same levels of fidelity and control as for a conventional live broadcast stream.

The 2-IMMERSE project is developing production tools to support the pre-production authoring of object-based experiences, and also to trigger the presentation of objects during live broadcasts. These will be reported in future papers.

DISTRIBUTED MEDIA APPLICATION

We implement our object-based broadcasting experiences as Distributed Media Applications (DMApps). These comprise a set of software components which are orchestrated across the participating devices by a cloud service platform. These components typically render media objects created as part of the experience production, support user interaction and/or implement application logic.

A pre-requisite for enabling these components to run on any or all of the participating devices is a common client software environment. Within the 2-IMMERSE project we have adopted HTML/CSS/JavaScript, and in particular HbbTV2.0 (5) since it provides additional APIs and services which are useful in a multi-device environment.

For our service prototypes we have developed an HbbTV2.0 emulator, as we have found that the emerging OEM HbbTV2.0 products tend to have feature and resource constraints that limit some of the richer object-based experiences we seek to enable. However, we want these experiences to adapt to work on devices with a range of capabilities, including these commercially-available devices. The HbbTV2.0 emulator also implements a comprehensive onboarding flow, supporting user network configuration, sign-in, device pairing, experience discovery and experience launch. Figure 1 shows how our common client software environment maps onto companion devices, our HbbTV2.0 emulator and OEM HbbTV devices.

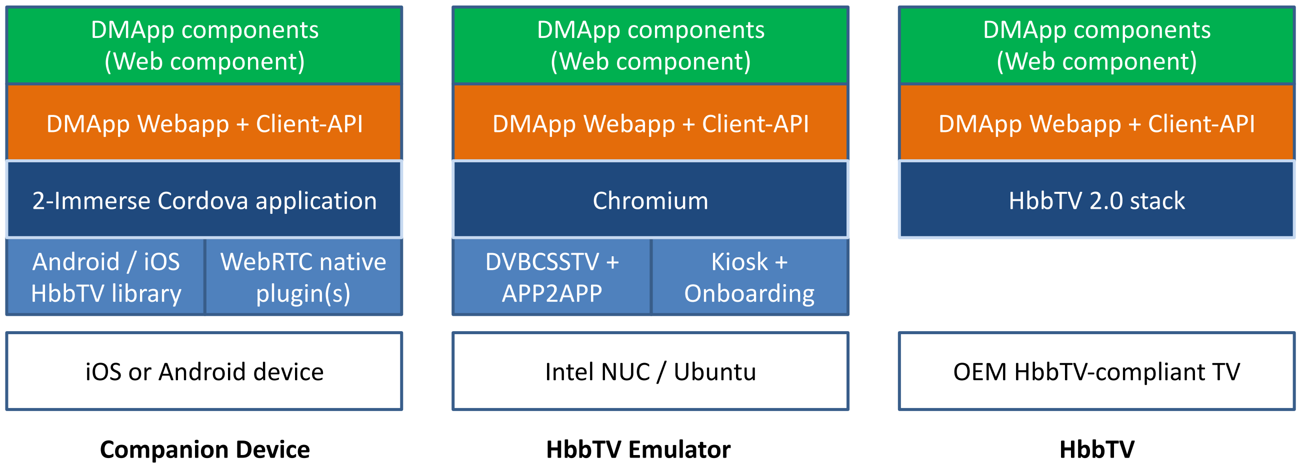


Figure 1. Client Software Environment

A DMApp is more than a just set of HTML components. t also includes timeline and layout documents defining the component orchestration, and HTML/CSS/JavaScript to host the Client-API and components. The Client-API is a JavaScript library that manages interaction with the platform, component management and presentation, including media synchronisation.

SERVICE PLATFORM

As noted in the previous section, the components within a running DMApp need to be orchestrated across the participating devices, which we refer to as a *context*. Within the project we have developed a cloud-hosted service platform to facilitate this, as shown in Figure 2.

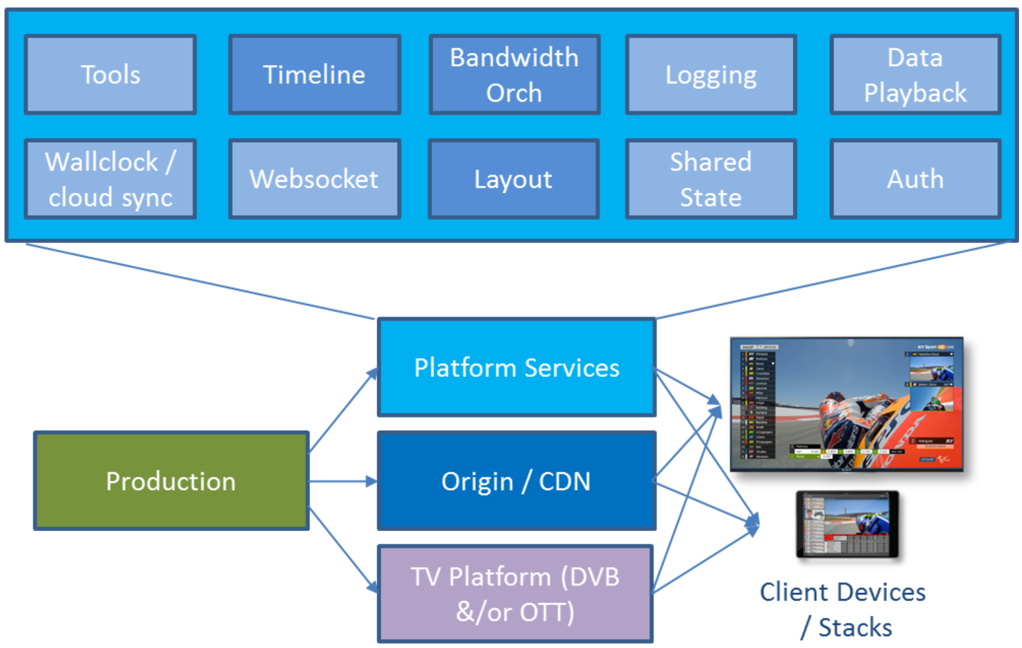


Figure 2. The 2-IMMERSE Service Platform

The platform comprises a set of containerised services, managed using the open-source Rancher container management platform (9), deployed on Amazon Web Services. Many of the services are common to typical IP-based TV platforms, such as logging and authentication. Services that make this architecture distinctive are concerned with DMApp orchestration and include the *timeline service*, the *layout service* and the *bandwidth orchestration service*, which all work together to determine the best possible presentation of the DMApp components given the available presentation devices and delivery bandwidth – i.e. these services put the right media asset in the right part of the right screen at the right time. The DMApp components are orchestrated following a lifecycle model progressing through a series of states: initialised, started, stopped and destroyed.

Timeline

The timeline service deals with temporal orchestration. It interprets an XML timeline document that controls the temporal composition of all the media items and components in a 2-IMMERSE context. Under control of the timeline document, it sends commands to create, start, stop and remove the individual components and media items at the appropriate time. It also supports live editing of documents for live production, as detailled in Jansen et al (6).

The service allows the triggering of sections of the timeline document based on user interaction through a *timelineEvent* mechanism. This allows, say, user interaction on a companion device to start a (possibly long) sequence of events in the experience. A *repeat* container (in addition to *parallel* and *sequential* containers) allows for such user interaction to happen multiple times.

An *update* facility supports changes to parameters or to layout aspects of active components under timeline control. Together with the *timelineEvent* this allows for seamless adaptation of the experience on user interaction.

Layout

The layout service is responsible for spatial orchestration; that is managing and optimising the presentation of the set of active DMApp Components across a set of participating devices. Given a set of active DMApp Components, authored layout requirements, user preferences, and the set of participating devices and their capabilities, the layout service will determine an optimum layout of components for that configuration. It may be that the layout cannot accommodate presentation of all available components concurrently.

The service instance maintains a model of the participating devices in the environment, and their capabilities e.g. screen size, screen resolution, colour depth, number of audio channels and interaction capabilities.

The layout requirements will specify, for each component, constraints such as minimum and preferred sizes, audio capability, interaction support, and whether the user can override these constraints. Some of these constraints may be expressed relative to other components (priority, position, etc.).

Layout changes can be triggered by a number of events, and layout updates are pushed to affected client devices via the websocket service.

Bandwidth Orchestration

The role of this service is to monitor the bandwidth consumption of ABR audio and video components in a running DMApp, and where necessary, to reallocate bandwidth between these streams to optimise the overall quality of experience. It does this in collaboration with layout service to determine the relative priorities and layout of active components. This uses the generalised architecture and protocols defined in MPEG SAND (7).

Related Standards Activity

The MPEG MORE draft specification (8) provides a generalised architectural framework for object-based media orchestration. It introduces the concepts of a Media Orchestration Session, sources, sinks, controllers and M-Processors. The service platform developed within 2-IMMERSE is broadly compatible, and it is possible to map our services, clients and data flows onto this framework. The temporal and synchronisation architecture adopted by MPEG MORE is DVB-CSS (4), which has also formed the basis of our synchronisation approach. It should be noted however that our implementation does not use the specific data protocols and formats of MPEG MORE, as our platform was developed independently of the specification effort.

MOTOGP SERVICE PROTOTYPE

The MotoGP experience (2) is based upon BT Sport's presentation of a MotoGP race using content from Dorna Sports. It has been designed and built as a multi-screen experience in which the race is shown on a communal TV and additional synchronised content is available on both tablets and smartphones ("companion" devices). The experience has a timeline that covers three phases: the pre-race build up ("Inside MotoGP"), the race itself ("Watch Live") and the post-race analysis ("Race Review"). The features available to the users in these phases vary and are described in the following sections.

Inside MotoGP

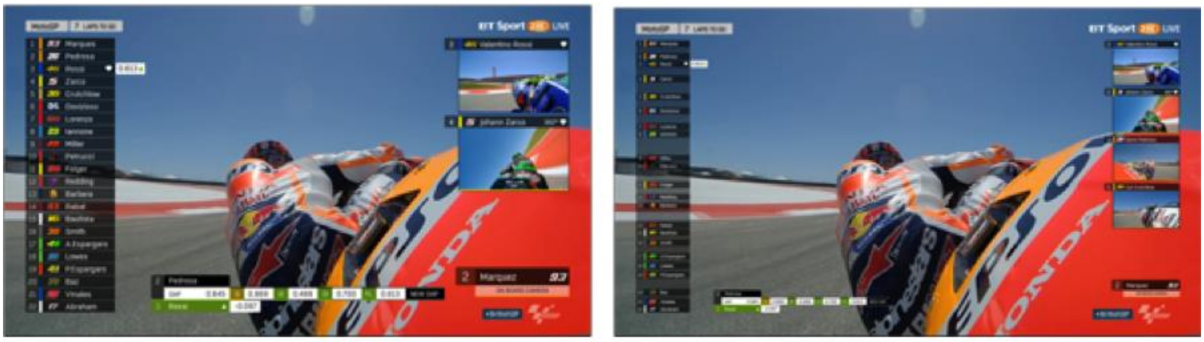
Alongside the commentary-led exposition on the TV prior to the race, access is provided to a variety of optional short-form video on demand (VoD) clips on the companion device to enhance the user experience. These include guide videos that explain how to use the experience, catch-up videos that bring viewers up to speed with recent MotoGP events and technical videos that help viewers understand some of the more technical aspects of the sport. In addition, during this phase users are prompted to use a menu option to personalise their experience, for example by selecting a favourite rider who is then highlighted within the leader board and on timing data.

Watch Live

During the race itself, some of interactions on the companion devices can affect what is shown on both the main TV and on the device. Users can switch between an interactive leader-board and mosaic of configurable video streams, a growing chronological list of events that can be viewed on demand as replays on the TV and tablet, and a list of viewing options which enable additional live camera streams, tracking and timing data to be configured for presentation on the tablet. There is also a casting option that enables users to select an additional live stream to display as a picture-in-picture view on the TV.

The presentation of media objects can be adapted to suit the context of the viewer through menu options, where context may include the size of the TV or whether the viewer has declared themselves an expert, fan or a novice as shown in Figure 3. A menu option also allows personalisation of the audio mix, by independent volume control of the commentary and ambient audio.

Responsive TV Expert Mode layout optimised for 32", 50" and 65" TVs



Responsive TV Novice Mode layout optimised for 32" and 65" TVs

Figure 3. This figure shows how, using the object-based broadcasting approach, the size and layout of the on-screen graphics can be adapted to better suit the context of TV size and to provide information suited to the specific needs of expert and novice viewers

Race Review

Alongside the commentary-led post-race coverage presented on the TV, access is provided to a set of optional VoD replays that can be selected by the user. These are a super-set of the replays that were made available in the Watch Live section. These replays are only presented on the user's companion device but often consist of synchronised views from multiple cameras angles.

EVALUATION

The evaluation of the MotoGP experience took place between November 2017 and January 2018. We sought to evaluate users' responses to the overall experience and specifically to the new features offered by the object-based approach. It also provided more generic insights that should be valuable for subsequent prototypes.

The evaluation was carried out with 44 households and 4 lab trials using a fully-developed prototype. Overall, we received assessments of the experience from 93 individual users. Evaluations were based on questionnaires, qualitative semi-structured interviews with triallists and on analytics of application-use based on instrumentation of the DMApp.

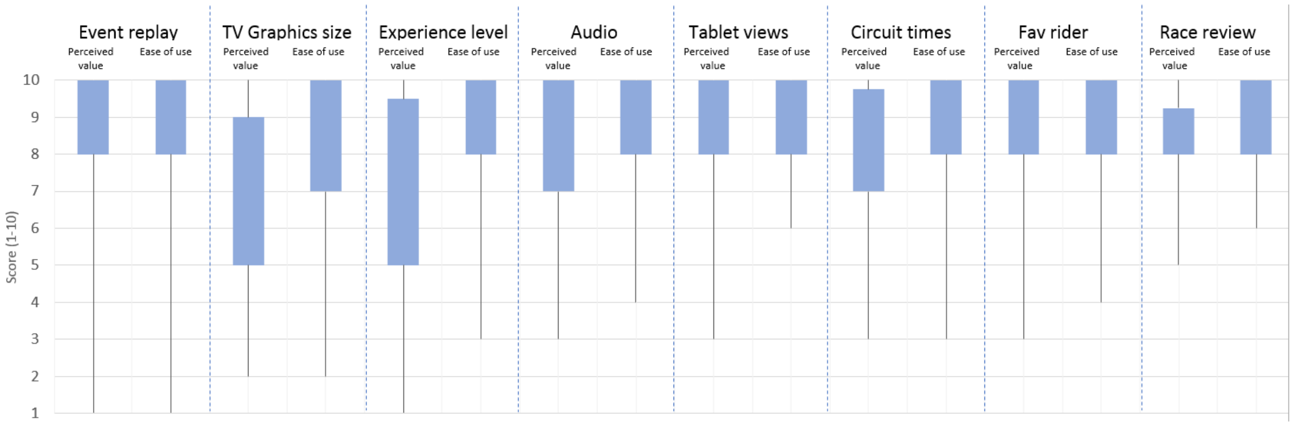


Figure 4. Summary results from the assessments of value and ease of use for the broad categories of features enabled in the multi-screen MotoGP experience.

Evaluation results

Users provided quantitative feedback on the perceived value and ease of use of different functions. These assessments are shown in Figure 4. Users scored nearly all features at around 9/10 which we regard as good. The feature regarded as most valuable (according to the mean score) was the Tablet Views, with Event Replays in second and Race Review in third. In commentaries, users were particularly taken with ability to select and view additional live synchronised camera views; additional camera views was also the feature most likely to be mentioned by users when probing spontaneous recall of the features available in the experience.

> *"It was more interesting interacting with the race than just sitting watching it because I felt more involved."*
>
> *"I believe the different viewpoints made the experience more interesting to watch and for me created an involvement in the race."*

The verbatim feedback relating to the overall experience has assured us that our key design goal – not undermining or obscuring the technical craft and ingenuity that have been used to create compelling entertainment experiences – has been met. Indeed, it is tempting to believe that, for many viewers, the multi-screen experience is better than the traditional TV experience.

CONCLUSIONS

We have introduced a technical architecture used for the delivery of object-based broadcasting to multiple screens together with a description and evaluation of a multi-screen service prototype developed using MotoGP content from Dorna Sports and BT Sport. We have introduced the concept of a component-based Distributed Media Application, and described the cloud-based microservice platform used to orchestrate DMApp components across participating presentation devices, with a focus on the orchestration services: timeline, layout and bandwidth orchestration. A multi-screen service prototype developed using MotoGP content was evaluated very positively by over 90 respondents. The most memorable and valued features of the service were the additional synchronised camera views that were made available and placed under the control of the users.

REFERENCES

1. 2-IMMERSE. 2015. *2-IMMERSE.* Accessed May 2018. www.2immerse.eu.

2. 2-IMMERSE. 2018. *The 2-IMMERSE MotoGP at Home multi-screen experience.* 24 April. https://www.youtube.com/watch?v=FZIhrnGzC4I.

3. Armstrong, M., Brooks, M., Churnside, A., Evans, M., Melchior, F., and Shotton, M. 2014. “Object-based broadcasting - curation responsiveness and user experience.” Amsterdam: IBC 2014.

4. DVB. 2015. Digital Video Broadcasting (DVB); Companion Screens and Streams; Part 1: Concepts, roles and overall architecture. s.l. : ETSI, 2015. ETSI TS 103 286-1 V1.1.1.

5. HbbTV Association. 2018**.** *HbbTV 2.0.2 Specification.* s.l. : ETSI, 2018. TS 102 796 V1.4.1.

6. J. Jansen, D. Bulterman and P. Cesar. 2018. “Workflow Support for Live Object Based Broadcasting.” *DocEng.* Halifax, Canada: ACM.

7. MPEG. 2017. *Information technology -- Dynamic adaptive streaming over HTTP (DASH) -- Part 5: Server and network assisted DASH (SAND).* s.l. : ISO/IEC, 2017. 23009-5.

8. MPEG. 2016. “MPEG-B (Media Orchestration) w16194.zip .” *MPEG-B Media Orchestration.* June. Accessed May 01, 2018. https://mpeg.chiariglione.org/standards/mpeg-b/media-orchestration.

9. Rancher. n.d. *Rancher.* Accessed May 01, 2018. https://rancher.com.

ACKNOWLEDGEMENTS

The work presented in this paper was supported by the EU funded ICT project 2-IMMERSE, under contract 687655