THE QUEEN'S AWARDS
FOR ENTERPRISE:
INNOVATION
2011

B B C

Research & Development

White Paper

WHP 243

March 2013

AN EXPERIMENT IN SOCIAL TV FOR AUTOMATING CONTENT SHARING VIA SOCIAL NETWORKS

Vinoba Vinayagamoorthy and Jerry Kramskoy

BRITISH BROADCASTING CORPORATION

White Paper WHP 243

An Experiment in Social TV for Automating Content Sharing via Social Networks

Vinoba Vinayagamoorthy and Jerry Kramskoy

Abstract

This paper presents an overview of prototypes built in order to enable users to share content across a social network such as Twitter in an automated fashion. Currently, it is a laborious process to manually extract the information needed to refer to a programme being viewed on a connected TV unambiguously in order to construct a short tweet or post on a mobile device. This process is made more complex if the user wants to refer to a specific part of the programme. Our prototypes use the Universal Control API to extract information about what programme a user is watching on their connected television. The extracted information is then used to automatically 'pull' programme-relevant content to a companion device such as a mobile phone, tablet or laptop. The user can then choose to share the online content on social networks. Conversely, the user can look-up a programme online and then 'push' relevant information to the connected television in order to tune it to the right channel.

Additional key words: dual screen, interface

An Experiment in Social TV for Automating Content Sharing via Social Networks

Vinoba Vinayagamoorthy and Jerry Kramskoy

1   Introduction

Watching television has been considered a shared experience (1) (2). The advent of the connected TV means that sharing the TV experience can include a circle beyond the confines of the living room and is no longer limited to 'water cooler moments' (3) (4) (5). Some viewers create their own parallel narrative of the show, albeit guided by events in the show, through social networks (6). The way in which they do this varies depending on the TV show. For instance, in a series of the X-Factor, audiences preferred to tweet during adverts if the episode was a performance show, however, if the episode was a results show then audiences tweeted during the show (6). Also audiences engage in different types of conversations based on the type of show they are watching. Recent findings (5), suggest that audiences preferred to connect with celebrities and stars of the show while watching 'Strictly Come Dancing' (7). On the other hand, while watching BBC Question Time (8), audiences prefer to connect to friends. Current trends indicate that shows that are watched live often have an associated greater social buzz. The type of conversations audiences have on social networks about TV shows and the factors that affect them is an interesting research area by itself but it is clear that the desire to share TV experiences is quite strong. Industry interest is growing around companion screens and associated services to the point where standards bodies are now considering the possibilities of standardising various information flows across broadcast networks, home networks, and the Internet.

This white paper explores a common use case in the area of social TV where users share links to webpages relevant to the programme currently being viewed on TV. This paper attempts to describe a prototype implementation to automate the process and discuss the challenges met in trying to do so using currently deployed broadcast technologies.

2   Push, Pull and Share

User A would like to send a message to User B about a particular scene in a piece of video content being broadcast on TV. This is generally accepted as normal user behaviour in today's world of connected devices. It allows users to connect to each other across geographical and chronological spaces, using media as a way to keep in touch and maintain personal relationships (9) (10) (11) (12). This is typically a manual process for both User A and User B. Moreover, the methods, timings, and reasons associated with the sharing process of different types of content are quite varied. There is no way for User A to predict how or when User B chooses to receive their message. We investigated whether this situation could be improved.
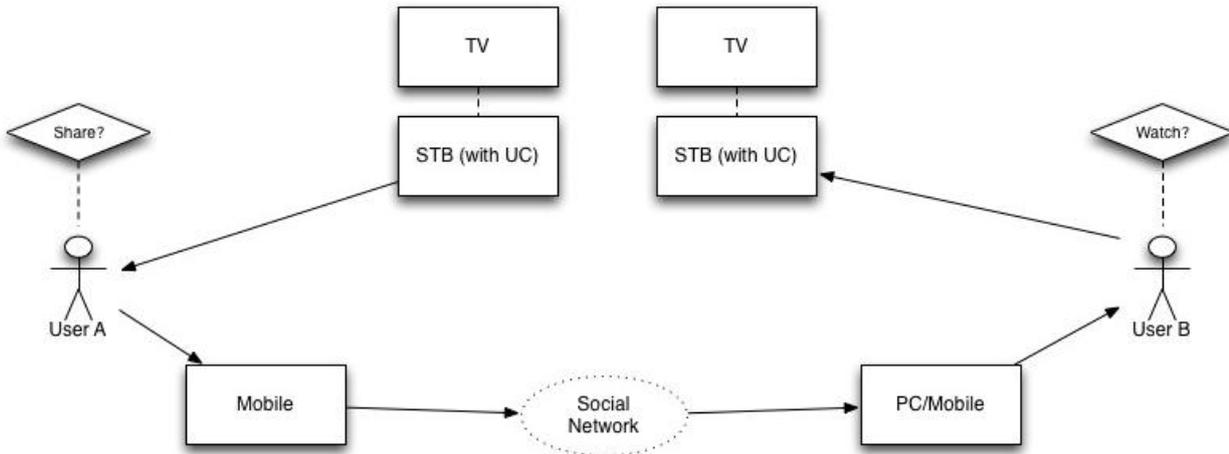
3   Current Process

Figure 1: Depiction of the flow of actions in a typical content share process between users

Usually, users on a mobile device (User A in Figure 1) wishing to share thoughts about a programme they were watching on a connected TV would have to invoke a third party social network app such as Twitter, type in their message and submit it. In order to be of use to an intended recipient (User B in Figure 1) the message would have to contain information about the programme which should include:

- Clues to which program User A is referring to
- The part of the program User A specifically finds interesting (optional)

Similarly users viewing the message at the receiving end would have to read and interpret the message on the social network. However, User B (Figure 1) has a choice of devices to receive the message on and a choice of devices for watching the programme on.

If User B chose to access the programme content on a mobile device or a PC/laptop, they would have to find it and navigate to it, such as via a web site search. To watch it on their connected TV, they would have to manually search available sources (either Live or on-demand) via the EPG or catch-up interface to access the content. Whilst the process is not difficult, it is tedious and requires quite a bit of motivation. Also there is no guarantee that the content is available on the device User B tries to find it on.

With home-network based protocols like Universal Control[1] (13) and identifiers such as the CRID (14), it is possible to automate parts of the chain described, making the experience effortless and seamless. The main technical challenge to using the Universal Control (UC) is that it requires server software to be incorporated in Set Top Boxes (STBs) or connected TVs. The advantage is that, once the server is present, it can expose functionality of the STB or TV to authorised client software on other devices connected on the same home network.

Alternatives, such as audio watermarking or fingerprinting technologies may also be used to enable part of the communication pathway between devices but there are associated costs and constraints (15).

3.1   Conveying the Message

Currently, in order to ease access to the content for User B, User A might have to state the title of the programme, episode title/number, and perhaps channel details if the message was constructed purely in natural language. If the information was conveyed through a relevant URI, it makes it easier to convey a lot of the message in one short phrase but currently User A must search and find a relevant URI to attach to their message. For example, User A may find a relevant URI on the BBC iPlayer site (16) or on the BBC's /programmes (17) pages with the aid of a search engine

---

[1] Universal Control is here after referred to as UC

such as Google or Bing. If User A wanted to convey a message relevant to a particular segment of the programme, information about the time offset during the programme will also have to be embedded in the message. Currently, most programmes broadcast on BBC channels have an associated /iplayer (16) and/or /programmes (17) resource which is mostly accessible through a /i redirection link. At the time of writing http://www.bbc.co.uk/i/b01n70f3/ redirected to the 5th episode of the 7th series of Dr. Who on iPlayer[2]. The iPlayer URI can also be embedded with a time offset from which to begin playback. So http://www.bbc.co.uk/i/b01n70f3/?t=5m00s redirects the user to a point 5 minutes into the aforementioned episode of Dr. Who. This makes the iPlayer link a viable candidate for purposes of sharing but it is hard for the viewer to construct such a link manually.

4   Prototype

We created a prototype based on Universal Control for seamlessly sharing (a section of) content between User A and User B via Twitter.

User A uses our bespoke companion app that wraps UC-aware web content in PhoneGap (18) to provide a native look-and-feel and simple porting to iOS and Android. User B uses a browser that has been enhanced with GreaseMonkey (19) to allow web page editing before presentation. User A's app queries the UC-enabled STB for information about what's on and its current time point, and constructs a link encapsulating this, which User A sends to Twitter. User B reads the Twitter message as usual and clicks on the link. The browser launches on User B's companion device that acquires the iPlayer or /programmes web page from the link. In our prototype, before presentation, the web page is dynamically edited. This adds additional buttons to the web-page that trigger Javascript when clicked. When User B clicks on one of these, the web page communicates with a UC-enabled STB to request the box to play back the content from the specified time point. In a real deployment, the buttons could be implemented as part of the web page to appear if a suitable connected TV or STB is detected on the same network.

The following sections depict the three disparate but inter-related acts supported by the prototype that allow User A to share a message about a piece of content with User B over a social network, so user B can experience that content.
- Pull – Act of pulling enough information from a connected TV to a companion app
- Share – Act of conveying a message from a companion device about a piece of content to enable the recipient to engage with the experience via the recipient's companion device (possibly rendering the content on another device such as a connected TV)
- Push - Act of pushing enough information to a connected TV from a companion app

4.1   Pull

In order for a user to retrieve relevant information on a companion device about a programme on a connected TV, the user has to 'find' it. In a scenario in which the user has access to a set top box (STB) with a UC server on a home network, companion devices installed with UC enabled client applications and connected to the same home network can post queries to uncover information about what programme the user is currently viewing.

In the BBC, there are two different identifier namespaces available for identifying content. One namespace deals with broadcast content, which uses Content Reference Identifiers (CRID) embedded in the broadcast multiplex to uniquely and permanently identify TV programmes. The other identifier namespace identifies online content resources. A resource may be additional information about a programme, found in /programmes, or it may be AV material available via iPlayer during the catch up window. Programme identifiers (PID)[3] are used for the identifier namespace of these resources. A URI such as 'http://www.bbc.co.uk/iplayer/PID' or 'http://www.bbc.co.uk/programmes/PID' names a particular resource.

---

[2] http://www.bbc.co.uk/iplayer/episode/b01n70f3/Doctor_Who_Series_7_The_Angels_Take_Manhattan/
[3] This use of the abbreviation PID should not be confused with the MPEG-2 transport stream packet identifier.

A UC server can acquire CRIDs extracted by the STB from the multiplex. Information about the current 'play head' position of the programme can also be queried from the UC server on the STB. This is particularly useful as it allows a UC-aware companion app to uniquely identify broadcast content and provide time offset information about the content they are watching in an automated fashion.

However there is no direct mapping system between these two namespaces. Online content does not contain CRIDs and the broadcast multiplex does not carry PIDs.

In order to bridge this problem, HTTP requests can be made by the UC companion app to a resolving service (20) (21) (22) on a web server which maps between CRIDs and PIDs. Currently, this service is not officially proffered to BBC audiences/users, However, a limited version of this service is being developed within BBC R&D.

To provide the user with relevant online data, information about the broadcast content has to be *pulled* from the STB and fed to a resolving service. The '**Pull**' phase involves the companion app obtaining CRID information relevant to the broadcast content from the UC server on the STB and sending it via a query to the resolver service. The query to the resolver service returns a PID to the companion app which constructs an appropriate iPlayer URI which includes the time offset information. Generally, the URI allows access to the content on BBC iPlayer at the corresponding point in time offset from the beginning of the programme. If a BBC iPlayer version of the content is not available due to right issues, the link gets redirected to the appropriate /programmes link by iPlayer.
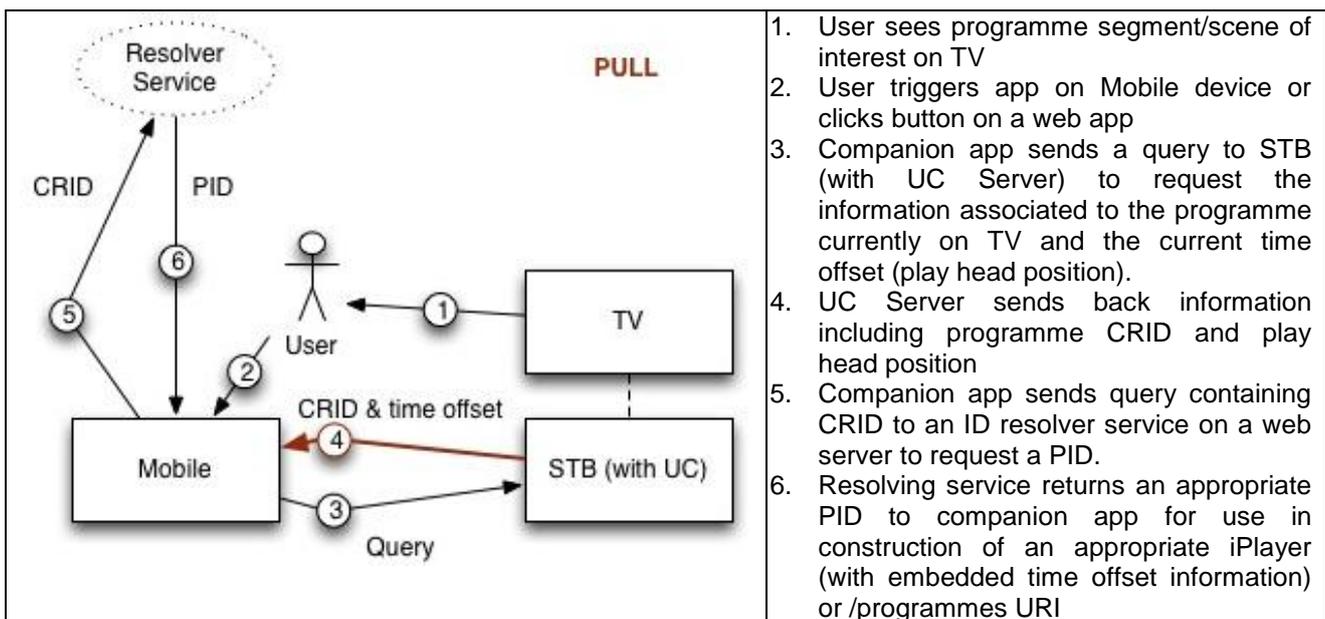

	1. User sees programme segment/scene of interest on TV
	2. User triggers app on Mobile device or clicks button on a web app
	3. Companion app sends a query to STB (with UC Server) to request the information associated to the programme currently on TV and the current time offset (play head position).
	4. UC Server sends back information including programme CRID and play head position
	5. Companion app sends query containing CRID to an ID resolver service on a web server to request a PID.
	6. Resolving service returns an appropriate PID to companion app for use in construction of an appropriate iPlayer (with embedded time offset information) or /programmes URI

Figure 2: Key data flow during the act of a 'Pull' in this prototype

4.2 Share: Choice of Platform

Once the URI has been constructed by the companion app, it can be shared out in a message to an intended recipient using email or social networking sites such as Twitter (23), Facebook (24), Google+ (25) etc.

The method of sharing is best left to the user since it normally depends on their personal preferences and the social network platform in which their 'friends' are active. Twitter is well known as a platform which is often used by TV watchers to share comments and opinions on (26) (27). Many programmes use specific hashtags (keywords prefixed with a '#') to enable users to easily follow comments from others relevant to the programme (28) and engage in social chatter about the content. However, Twitter also limits the number of characters in each message to 140 characters.

In order to minimise the number of characters used in embedding the constructed URI in a message, the companion app sends a query with the URI to a URI shortening and bookmarking service – in this case bit.ly (29). The companion app gets back a much shorter bit.ly link whose target can never be altered nor do the links expire. When users manually post a message to Twitter through their website, bit.ly is automatically invoked to ensure links embedded in the message are shortened, however, this facility is not integrated into Twitter's API yet. The companion app currently utilises personal user accounts with bit.ly. Furthermore, bit.ly uses the suffix 'http://bbc.in' whenever a BBC webpage is shortened. The shortened link is embedded into an editable message which also includes a possible hashtag for the programme and is presented to the user (User A in Figure 3). Once User A clicks the submit button in the companion app, the message is published on Twitter.

Generally, messages sent on the Twitter platforms can be viewed by anyone even if it is sent addressed to a particular user (User B in Figure 3). Once User B receives the Twitter message with embedded bit.ly link and clicks on it, they are redirected through the URL shortening service (bit.ly) to the iPlayer or /programmes page presented by the browser on User B's companion device.
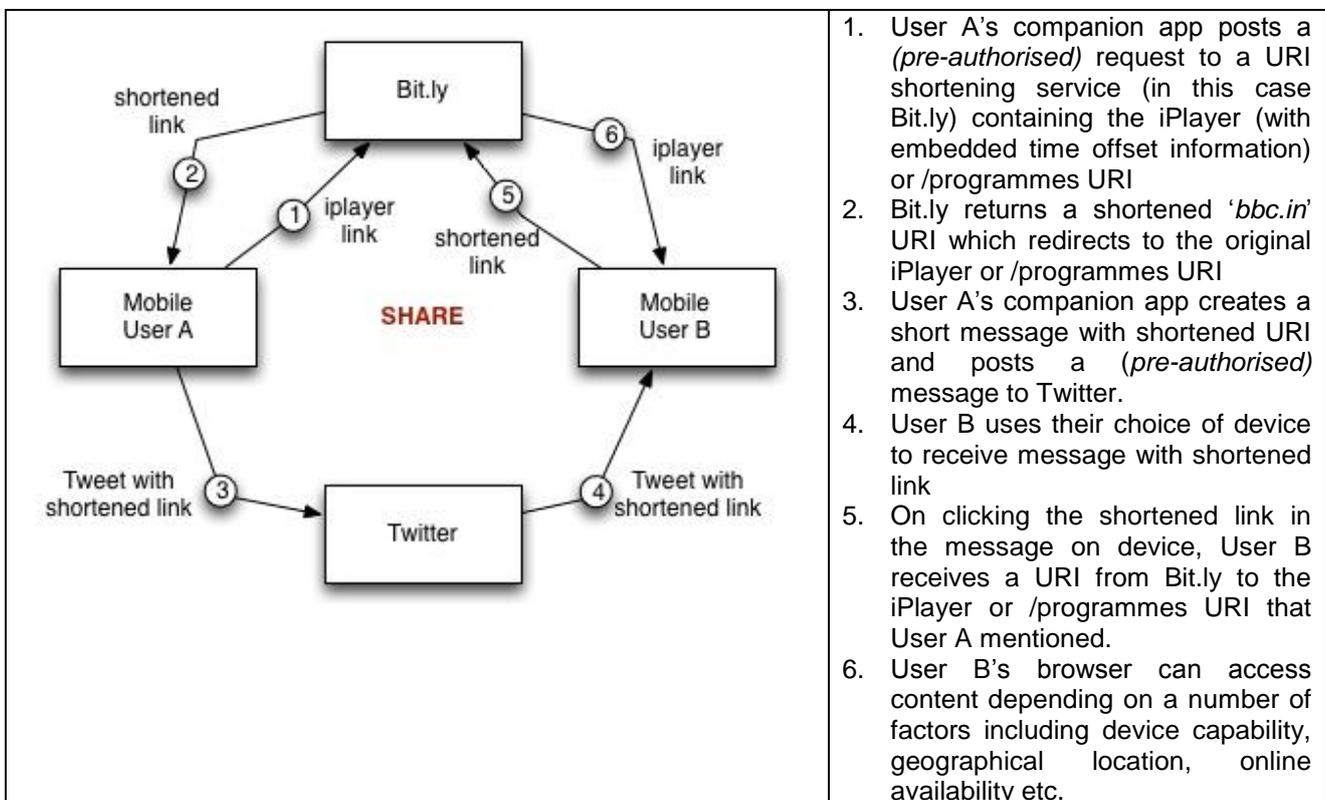

1. User A's companion app posts a *(pre-authorised)* request to a URI shortening service (in this case Bit.ly) containing the iPlayer (with embedded time offset information) or /programmes URI
2. Bit.ly returns a shortened '*bbc.in*' URI which redirects to the original iPlayer or /programmes URI
3. User A's companion app creates a short message with shortened URI and posts a (*pre-authorised*) message to Twitter.
4. User B uses their choice of device to receive message with shortened link
5. On clicking the shortened link in the message on device, User B receives a URI from Bit.ly to the iPlayer or /programmes URI that User A mentioned.
6. User B's browser can access content depending on a number of factors including device capability, geographical location, online availability etc.

Figure 3: Key data flow during the act of 'Sharing' in this prototype

4.3 Push

The webpage presented to the User B's device is dynamically constructed at the completion of the share process mentioned above. The web page returned from the BBC is edited through the use of a Grease Monkey (30) script to include additional UI controls and Javascript triggered by those controls.

The Javascript is UC-aware and also employs the resolution service described in section 5.1 (20). Together, these enable an automated "**push**" of the content to a connected TV. The Javascript extracts the PID from the URI for the web page and sends the PID in a query to the resolver service (21) on a web server which returns a relevant CRID. The CRID (as discussed in 4.1) enable references to broadcast content on a STB.

The Javascript can then discover the UC-enabled box, and '**push**' a request to its UC server to play a specific piece of content identified by the CRID (at a particular point in the programme as identified by the play head position).
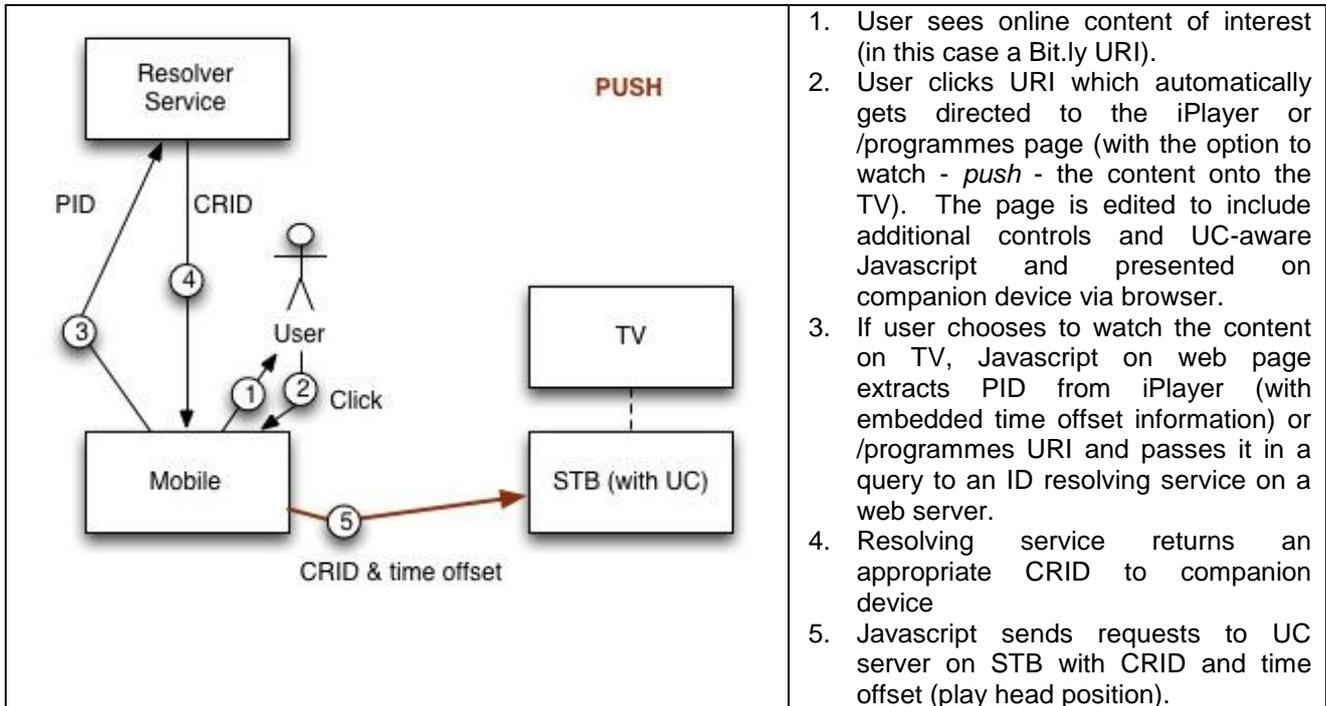

	1. User sees online content of interest (in this case a Bit.ly URI).
	2. User clicks URI which automatically gets directed to the iPlayer or /programmes page (with the option to watch - *push* - the content onto the TV). The page is edited to include additional controls and UC-aware Javascript and presented on companion device via browser.
	3. If user chooses to watch the content on TV, Javascript on web page extracts PID from iPlayer (with embedded time offset information) or /programmes URI and passes it in a query to an ID resolving service on a web server.
	4. Resolving service returns an appropriate CRID to companion device
	5. Javascript sends requests to UC server on STB with CRID and time offset (play head position).

Figure 4: Key data flow during the act of a 'Push' in this prototype

4.4 Prototype

In the case of the prototype, a laptop connected to the same home network as the STB with the UC server acts as the companion device. Figure 5 and Figure 6 show the '**Push to TV**' button which allows the user to watch Waybuloo and Dr. Who on a TV connected to a UC-enabled STB on the home network.

Figure 5: iPlayer page with the 'Push from TV' and 'Pull from TV' buttons

Figure 6: /programmes page with 'Push to TV' button

Figure 5 shows the 'Pull from TV' button which allows a user to retrieve the iPlayer page relevant to the BBC programme currently being viewed on the TV. However, this button might be better placed at other pages on the iPlayer website such as the home page in order to fit in better with more probably user stories.

5 Discussion

This section presents some of the issues encountered during the implementation of the prototype.

5.1 Challenges

There are considerations that need to be taken into account with respect to the timing accuracy of broadcast signalling events and how these map to indicate the beginning of a programme. Often there is a time offset between the signalled start time of the programme and the actual time at which the programme starts. This means that the timings extracted from the broadcast might not match up with the equivalent timings on the associated content on iPlayer. Combined with the lag associated with a user using the pull functionality on a client app, the time embedded into the published link might be a few seconds after the point in time in the programme to which the user intended to refer to. If the offset is consistent, then the problem can be mitigated by adjusting the timings obtained from the broadcast to give a better estimate.

The prototype depends on the availability of CRIDs and the means to access them through Universal Control. CRIDs are permanently allocated to a particular programme and unique to it. This makes them a good candidate to query and find out about the programme currently on the TV. If CRIDs are not available, it is possible to identify the programme the user is currently watching live by inference based on other metadata available in the STB, for instance using the DVB triplet[4] in conjunction with the event ID[5]. However, this information is not currently accessed by UC and unlike CRIDs multiple pieces of metadata need to be used in combination to uniquely identify a programme. An alternative (back-up) way of dealing with the problem of missing metadata on the STB might be to present the user with a current electronic program guide (EPG) for manual selection. However, this won't be a viable solution if the programme being watched is not 'Live'.

---

[4] The DVB triplet distinguishes between the same service carried by different networks. For instance, the triplet differentiates between the channel BBC1 carried by BskyB and by Freeview.
[5] The event id refers to a particular broadcast of a programme within one service

There are risks associated with relying on a resolution service. Such a service may fail either due to technical errors at the server end or due to network issues. In these cases, appropriate error messages need to be in place.

There might not be a specific iPlayer clip associated with the program. In this case, the user could be redirected to the brand page of the program. It is normally BBC policy to have microsites (editorially approved webpages) that are relevant to BBC programmes. The microsites tend to have the appropriate links to relevant iPlayer pages.

If an iPlayer URI is accessed while the programme is 'Live', currently the URI automatically redirects to an iPlayer placeholder page which will hold the content once the broadcast has finished. Provided appropriate rights have been secured for the content, the content is encoded and made available for on-demand viewing to BBC audiences living in the UK. However, there is a period of time in between the end of the live broadcast and the content becoming available on the placeholder associated with the programme on the iPlayer platform. At the time of writing, during the live broadcast, the content on some channels is available as part of iPlayer's 'Watch Live' functionality (31). In addition, if the viewer attempts to watch a programme being broadcast live on a PC, iPlayer's 'Live Restart' allows the view to rewind and begin watching the programme from the beginning (as long as the beginning is less than two hours prior to the current time) without waiting for the programme to end and become available on the associated iPlayer placeholder (32). 'Live Restart' also allows the viewer to simply rewind to a point up to two hours prior to the current time on the 'Watch Live'-enabled BBC channels online.

The content on iPlayer is often coupled with an expiry period so if the link is accessed after the date of expiry, the user will be unable to access the content. It is possible that issues of expiry will not be applicable to all content as the BBC considers means of making more of its archived content available to the public.

There are a number of relevant BBC online pages related to the content broadcast on the TV platform. At the beginning of the project, the microsite URI was considered as an appropriate home platform to which to direct users who received recommendation URIs through third party (or other) social networking platforms. However, if conversations about content were to include time offsets, a direct URI to the iPlayer placeholder page would enable users to follow the exact point of interest in the content. This allows the user to watch the content directly on the iPlayer site on a companion device provided they have the right to. Since the URI is being shared to a third party public platform, it is important to ensure the URI 'works', regardless of where the user is, when they click on the link and which device they use to do so. Needless to say the URI shared would have to be platform agnostic.

In our prototype, we use bit.ly to shorten the link due to the restriction of characters placed by Twitter in sending messages. Although, bit.ly is useful for shortening purposes, the resultant URI is quite nondescript which means the user cannot form an impression about what type of content they are going to be presented with when they follow the link. This can be remedied to some extent by adding information about the title and broadcaster in the message.

5.2 Other Considerations

Other than sharing out a reference URI to video content in order to author social chatter about it on a third party network, there are other user scenarios that might require the user to bookmark a specific point in the programme. For instance, the user may want to continue watching a piece of content on their companion device while they are out and about from the point at which they paused watching earlier on the connected TV in their home. Whatever the user story is, there are two problems when dealing with users looking for a specific programme at a specific point in time. The first is to do with finding what the user perceives as the same piece of content in order to ensure continuity in the user experience. The second is to enable the user to join the experience at a point of their choice.

Figure 7: Time factor versus content availability

The time at which users want to access the content relative to when the content is broadcast is an important factor to consider in sourcing it. For instance, it probably makes good design sense to only proffer users with the option to 'push' content to the connected TV if it is available on the on-demand platform or is being broadcast 'live'.

If the content is being broadcast at the time, the TV provides no choice but to join the broadcast "now" (so, at a point out of their control). On iPlayer, the 'Watch Live' and 'Live Restart' functions allow the user to either join the broadcast "now", rewind to "the start of the current programme" with the click of a button or to use a slider to rewind to a point in time up to two hours prior to "now". Some broadcasters have a second time delayed (generally by one hour) channel which broadcasts everything from their main channel with an hour delay. If the programme is a repeat, then the broadcast content might already have been made available as catch-up content. However, in these cases, services need to be in place to allow the user to find these alternative sources and the user's app appropriately designed.

If the broadcast has ended, then the content might be available on demand (in the case of the BBC on iPlayer) and therefore adding the 'push' button on the companion device makes design sense. However, the content does not become available immediately on the relevant iPlayer link. The prototype deals with this period by directing the user to the placeholder page which contains a suitable 'coming soon' message.

Other than too much time lapse between the time of broadcast and the time at which the user chooses to access the content, there are other reasons why a user might not be able to access a particular piece of content – for instance if the user happens to be outside the UK (due to geographical restrictions on the content). This is generally dealt with by providing appropriate feedback to the user; however, a redirect to alternative relevant content could be a less jarring solution. Sometimes the content is distributed through other broadcasters (networks), for example *Top Gear* on Dave or *The Blue Planet* on Eden TV. If the user has access to these sources, a future iteration of client applications similar to the UC-Client prototype could re-route the user to those sources. The trend for non-linear programming and access to infinite channel choices means that retaining user loyalty is a difficult task when the same content is available elsewhere. Similarly, broadcasts of BBC commissioned and resold content on other channels may generate social chatter which may lead to users attempting to access BBC online services for content relevant to relatively older programmes.

Similar content from different sources often contains variations which might be caused due to the insertion/deletion of adverts, regional variations (33), variations due to nation-specific programming (17), geographical variations which may involve other broadcasters etc. Currently, within the UK, iPlayer allows users to view programmes not broadcast in their region. Users will have no issues during a 'pull' phase in order to get a relevant online URI to a programme that is specific to an English Region or UK Nation such as a viewer in England trying to access *River City*[6] (34). If the URI for *River City* is shared onto a social network, iPlayer allows the viewer in England to access the content, however, the viewer will not be able to 'push' the content to their connected TV in order to view a live broadcast of *River City* or a recorded version of it. Geographical constraints are another important factor to take into account in implementing 'push' and 'pull' services. Current connected TVs have access to on-demand services like iPlayer but investigating the

_____

[6] River City is a drama series broadcast on BBC One Scotland

implementation of 'push' services to incorporate iPlayer on the TV platform is beyond the scope of this paper.

The availability of content across regions is not the only issue. Allowing the user to access the correct 'version' of the content is also problematic. Consider the situation in which user A in region A wants to share content across a social network to user B in region B. If the content is the same across both regions then User B will receive the message as User A intended it. If the content in region A is not available in region B, then as it stands User B will not be able to view it on a connected TV as a broadcast but they will still be able to access it on iPlayer. However, if the content is available in region B but in a slightly different form (for instance with added 'how the program was made' commentary at the end), User B might want that version instead. On the other hand, if User A had included a specific time in the content and written a message which references that point then User B might want the version User A watched. The variation in content introduces a different timeline for the 'same' content from multiple sources. Consistently referring to a specific point of interest within the content using a time stamp alone becomes difficult when the correlation between the timeline and scenes is not set.

6   Conclusion

We have described a prototype which allows the seamless sharing of (a section of) content between two connected-TV users via an existing third party social network. This has been made possible by allowing users to easily find iPlayer or /programmes pages associated with a specific piece of broadcast content through the use of the Universal Control server in the connected TV and a cloud-based resolution service.

Universal Control is used to identify the content being viewed by a user on a connected-TV using an App on a companion device such as a mobile phone. Alternative technologies, such as audio watermarking and audio fingerprinting, can also be used to extract this information. If none of these options were available, at the minimum, the user could even use an EPG to indicate manually which programme they are watching. However, with Universal Control, the process is much simpler and the user can also extract a specific point of interest in the form of time since the start of the programme. In addition, Universal Control enables two-way communication between the connected TV and an app on the user's companion device. This allows users to both 'pull' and 'push' information about content from/to their connected TV.

The resolution service is used to map programme CRIDs to iPlayer links (identified by PIDs). The service allows users to easily retrieve an iPlayer link which can also contain timing information to specify a particular point in the content. The link can then be shared or bookmarked as the user sees fit. In the absence of a resolution service, users could search for a relevant iPlayer link using the name of the programme and attach an estimated time from the start of the programme to indicate point of interest. The resolution service attempts to ease the manual process.

In a world where the connected TV and devices communicate seamlessly within a home network, the 'push' and 'pull' functionality can be embedded in relevant microsites, /programmes and iPlayer pages as deemed appropriate for the specific dual screen experience at hand.

7   References

1. *The Role of Smartphones in Mass Participation TV.* Lochrie, Mark and Coulton, Paul. 2012. The 10th European Interactive (EuroITV) TV Conference.
2. The Economist. The lazy medium - How people really watch television. *The Economist: Special Report - Television.* [Online] 29 April 2010. [Cited: 6 November 2012.] http://www.economist.com/node/15980817.
3. Casciato, Paul. New TV and social media trend amonf the youth - study. *Reuters: UK.* [Online] 8 March 2011. [Cited: 2012 November 2012.] http://uk.reuters.com/article/2011/03/08/uk-digital-social-tv-idUKTRE7275RZ20110308.
4. Deloitte. The schedule dominates, still. *Deloitte: TMT Predictions 2012 - Media.* [Online] [Cited: 6 November 2012.] http://www.deloitte.com/view/en_GX/global/industries/technology-media-

telecommunications/tmt-predictions-2012/media-2012/d3da068df67a4310VgnVCM1000001a56f00aRCRD.htm.

5. *Who is on Your Sofa? TV Audience Communities and Second Screening Social Networks.* Doughty, Mark, Rowland, Duncan and Lawson, Shaun. 2012. The 10th European Interactive (EuroITV) TV Conference. pp. 80-86.

6. *Sharing the Viewing Experience through Second Screens.* Lochrie, Mark and Coulton, Paul. 2012. The 10th European Interactive (EuroITV) TV Conference. pp. 199-202.

7. BBC. BBC One - Strictly Come Dancing. *BBC Homepage.* [Online] [Cited: 28 November 2012.] http://www.bbc.co.uk/programmes/b006m8dq.

8. —. BBC One - Question Time. *BBC Homepage.* [Online] [Cited: 28 November 2012.] http://www.bbc.co.uk/programmes/b006t1q9.

9. *Social television and user interaction.* Cesar, P., Chorianopolous, K. and Jensen, J. J. 1, s.l. : ACM, May 2008, Computers in Entertainment, Vol. 6, pp. 4:1-4:10. ISSN: 1544-3574.

10. *Content-Enriched Communication – Supporting the Social Uses of TV.* Chorianopoulos, Konstantinos. 1, 2007, The Journal of The Communications Network, Vol. 6, pp. 23-30.

11. *Past, Present and Future of Social TV: A Categorization.* Cesar, Pablo and Geerts, David. Las Vegas : IEEE, 2011. IEEE Consumer Communications and Networking Conference (CCNC): 3rd IEEE International Workshop on Social TV - the Next Wave. pp. 347-351. ISBN: 978-1-4244-8789-9 .

12. Kramskoy, Jerry. Orchestrated Media and Genres. *BBC Research & Development Blog.* [Online] BBC, 7 October 2011. [Cited: 11 October 2012.] http://www.bbc.co.uk/blogs/researchanddevelopment/2011/10/om-genres.shtml.

13. Jolly, Steve. BBC - Research and Development: Universal Control. *BBC R&D Blog.* [Online] 28 February 2011. [Cited: 22 August 2011.] http://www.bbc.co.uk/blogs/researchanddevelopment/2011/02/universal-control.shtml.

14. Earnshaw, N, et al., et al. The TV-Anytime Content Reference Identifier (CRID). *The Internet Engineering Task Force (IETF) Tools.* [Online] May 2005. [Cited: 31 July 2012.] http://tools.ietf.org/html/rfc4078. IETF RFC 4078.

15. Jolly, Steve. BBC - Research and Development Blog: Dual Screen Experiences: the Secret Fortune Pilot and the Sync API. *BBC R & D Blog.* [Online] 18 October 2011. [Cited: 24 July 2012.] http://www.bbc.co.uk/blogs/researchanddevelopment/2011/10/dual-screen-secret-fortune-sync-api.shtml.

16. BBC iPlayer - iPlayer TV Home. [Online] [Cited: 12 October 2012.] http://www.bbc.co.uk/iplayer/.

17. BBC - Programmes. [Online] BBC. [Cited: 16 August 2012.] http://www.bbc.co.uk/programmes.

18. Home Page. *PhoneGap.* [Online] [Cited: 20 November 2012.] http://phonegap.com/.

19. *Greasemonkey :: Add-ons for Firefox.* [Online] [Cited: 20 November 2012.] https://addons.mozilla.org/en-US/firefox/addon/greasemonkey/.

20. McRoberts, Mo, Jackson, Chris and Miller, Libby. Resolution service for Uniform Resource Identifiers (URIs) identifying broadcasts. *Project Baird.* [Online] 1 July 2010. [Cited: 7 August 2012.] http://projectbaird.com/specs/draft-mcroberts-uri-resolver.html.

21. Miller, Libby and McRoberts, Mo. Resolve a crid. [Online] [Cited: 7 August 2012.] http://services.notu.be/resolver.

22. Miller, Libby. Connecting Broadcast TV and the Web using a resolver. *NoTube | NoTube is a European research project exploring Semantic Web and TV.* [Online] 26 August 2010. [Cited: 6 August 2012.] http://notube.tv/2010/08/26/connecting-broadcast-tv-and-the-web-using-a-resolver/.

23. Twitter. [Online] [Cited: 12 April 2011.] http://www.twitter.com/.

24. Facebook. [Online] [Cited: 10 April 2011.] http://www.facebook.com.

25. Google+. [Online] [Cited: 3 August 2011.] https://plus.google.com/.

26. Twitter, TV and You. *Twitter Blog.* [Online] 25 October 2011. [Cited: 8 August 2012.] http://blog.twitter.com/2011/10/twitter-tv-and-you.html.

27. *Mobile phones as second screen for TV, enabling inter-audience interaction.* Lochrie, Mark and Coulton, Paul. Lisbon, Portugal : ACM, 2011. Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology. pp. 73:1-2. ISBN: 978-1-4503-0827-4.

28. Twitter on TV: A Producer's Guide. *Twitter Developers.* [Online] Twitter. [Cited: 8 August 2012.] https://dev.twitter.com/media/twitter-tv.

29. bitly. *Love your bitmarks.* [Online] [Cited: 9 August 2012.] http://bitly.com/.

30. Greasemonkey :: Add-ons for Firefox. [Online] Mozilla Firefox. [Cited: 16 August 2012.] https://addons.mozilla.org/en-US/firefox/addon/greasemonkey/.

31. BBC. BBC iPlayer - Help - Frequent questions. *BBC iPlayer - Help - Which channels can I watch live online (simulcast) and how do I watch them?* [Online] [Cited: 29 January 2013.] http://iplayerhelp.external.bbc.co.uk/help/prog_availability/what_simul.

32. —. BBC iPlayer - Help - Frequent Questions. *BBC iPlayer - Help - Live Restart - Frequent Questions.* [Online] [Cited: 29 January 2013.] http://iplayerhelp.external.bbc.co.uk/help/playing_tv_progs/live_restart.

33. *BBC - England - English Regions Regional TV.* [Online] [Cited: 30 August 2012.] http://www.bbc.co.uk/england/tv/index.shtml.

34. *BBC One - River City.* [Online] [Cited: 30 January 2013.] http://www.bbc.co.uk/programmes/b006p2xl.