NoTube

Networks and Ontologies for the Transformation and Unification of Broadcasting and the Internet

FP7 – 231761

D7c.4 Social TV Integrated Prototype of v.1 and v.2

Coordinator: Libby Miller (BBC)
With contributions from: Vicky Buser (BBC), Dan Brickley (VU)
Quality Assessor: Luca Vignaroli
Quality Controller: Lyndon Nixon

Document Identifier:	NoTube/2009/D7c.4/V0.9
Class Deliverable:	NoTube EU-IST-2009-231761
Version:	0.9
Date:	October 14, 2011
State:	final
Distribution:	public

Executive Summary

This document summarises the work done so far in the workpackage 'TV and the Social web' (WP7c). It describes the approach we have taken, the prototypes we have made, and our plans for the final three months of the project.

DOCUMENT INFORMATION

IST Project Number	FP7 – 231761	**Acronym**	NoTube
Full Title	Networks and Ontologies for the Transformation and Unification of Broadcasting and the Internet		
Project URL	`http://www.notube.tv/`		
Document URL			
EU Project Officer	Leonhard Maqua		

Deliverable	**Number**	7c.4	**Title**	Social TV Integrated Prototype of v.1 and v.2
Work Package	**Number**	7c	**Title**	Internet TV in the Social Web

Date of Delivery	**Contractual**	M33	**Actual**	31-10-11
Status	0.9		final ☐	
Nature	prototype ☐ report ☒ dissemination ☐			
Dissemination Level	public ☒ consortium ☐			

Authors (Partner)	Libby Miller (BBC), Vicky Buser (BBC), Dan Brickley (VU)		
Resp. Author	Libby Miller (BBC)	**E-mail**	libby.miller@bbc.co.uk
	Partner BBC	**Phone**	+44 (787) 6565561

Abstract (for dissemination)	In the workpackage 'TV and the Social web' (WP7c) we have used lightweight prototyping techniques in order to explore social TV in on-demand, live TV, and second screen authoring environments, using multi-screen environments. Our conclusions are that using metadata as an advert for TV programmes, unique URLs for TV, and APIs to TV and TV-like devices would make second screen applications significantly more useful and interesting.
Keywords	TV, television, set top box, media centre, iphone, ipad, XMPP, jabber, strophe, voting, companion device, second screen, authoring tools, HTML5

Version Log			
Issue Date	**Rev No.**	**Author**	**Change**
October 14, 2011	1	Libby Miller	0.9

Project Consortium Information

Participant's name	Partner	Contact
Vrije Universiteit Amsterdam		Guus Schreiber Phone: +31 20 598 7739/7718 Email: schreiber@cs.vu.nl
British Broadcasting Corporation		Libby Miller Phone: +44 787 65 65 561 Email: Libby.Miller@bbc.co.uk
Pronetics S.p.A.		TBA Phone: TBA Email: TBA
Engin Medya Hizmetleri A.S.		Ron van der Heiden Phone: +31 6 2003 2006 Email: ron@engin.tv
Institut fuer Rundfunktechnik GmbH		Christoph Dosch Phone: +49 89 32399 349 Email: dosch@irt.de
Ontotext AD		Atanas Kiryakov Phone: +35 928 091 565 Email: naso@sirma.bg
Open University		John Domingue Phone: +44 1908 655 014 Email: j.b.domingue@open.ac.uk
RAI Radiotelevisione Italiana SPA		Alberto Morello Phone: +39 011 810 31 07 Email: a.morello@rai.it
Semantic Technology Institute International		Lyndon Nixon Phone: +43 1 23 64 002 Email: lyndon.nixon@sti2.org
Stoneroos B.V.		Annelies Kaptein Phone: +31 35 628 47 22 Email: annelies.kaptein@stoneroos
Thomson Grass Valley France SA		Raoul Monnier Phone: +33 2 99 27 30 57 Email: raoul.monnier@thomson.net
TXT Polymedia SPA		Sergio Gusmeroli Phone: +39 02 2577 1310 Email: sergio.gusmeroli@txtgroup.com
KT Corporation		Myoung-Wan Koo Phone: +82 2 526 6347 Email: mskim@kt.co.kr

Table of Contents

List of Figures

1. Introduction

1.1 Deliverable Objective

The aim of this deliverable, D7c.4, is to:

- Describe the approach we have taken throughout the project

- Summarise the work we have done so far

- Describe the principles behind the work

- Describe the results

- Describe our plans for the final months

1.2 Approach

As defined in our first deliverable, our approach throughout the project has been to:

- Enhance TV user experience by helping users find content they enjoy by making TV more social

by

- Being able to flexibly and quickly demonstrate ideas in a compelling fashion

- Allowing users to share information and recommendations and create textual user generated content - essentially metadata about television - using appropriate social media tools

- Enabling users to share their profile information while respecting user privacy

- Testing the result with BBC content

Two things became clear early on in this project:

- Social media and TV is a very fast-moving area, and our ideas were in danger of being overtaken by commercial projects

- Second screens were the model of interaction that was both obvious and increasing in popularity

The first issue meant that it was very important for us to be very flexible, and also to try to think beyond the obvious connections between social media and TV around Twitter and Facebook.

The second issue threw up a big problem: designing for second screens for TV requires overcoming the lack of two-way communication between most TVs and any other devices. Most TVs and TV-like devices currently have nothing like this, which means no simple or direct way of syncing the second screen device and the TV or PVR.

We have taken the view throughout the project that *if* APIs to TV existed, *then* social TV would become more compelling, and created multiple *user-focused* prototypes which (we hope) show this, as well as pursuing standards-based solutions to the problem via W3C's Web and TV Interest Group, and working with the BBC's 'UC' API.[1] Throughout, we have tried to use tools and frameworks that allow us to create prototypes as quickly as possible, while using diverse techniques for coming up with interesting and relevant usecases and scenarios.[2]
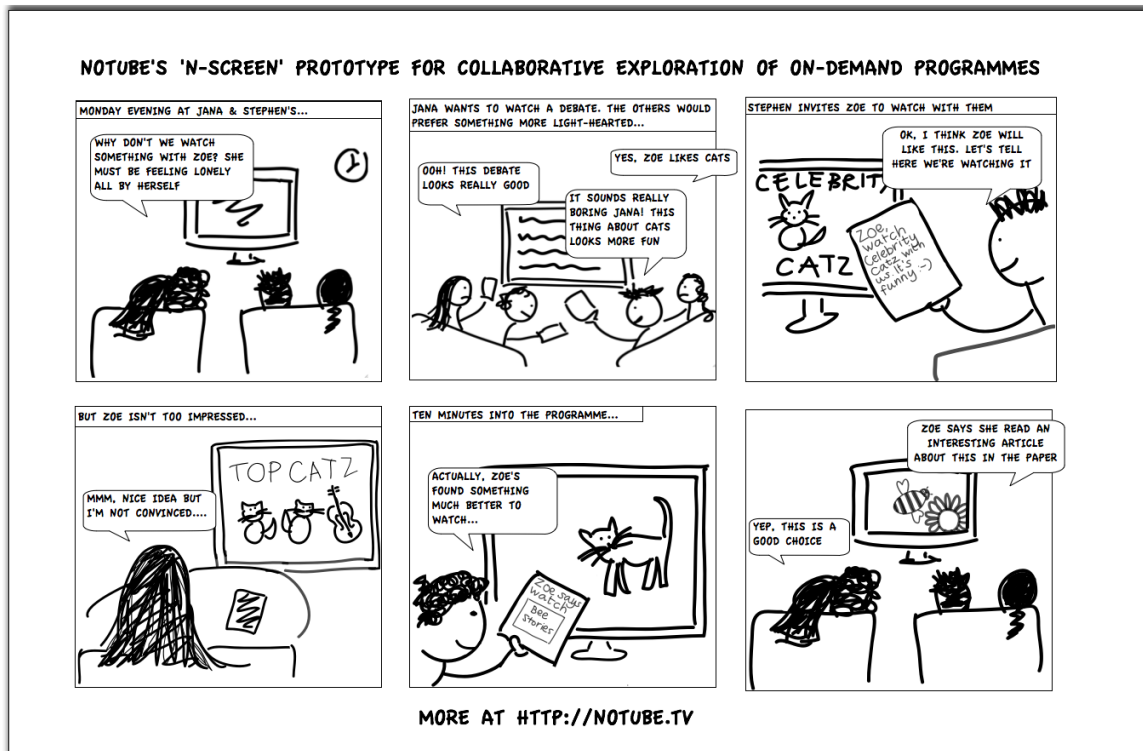

Figure 1.1: A sample NoTube Postcard showing a user scenario

1.3   What have we built?

We have made prototypes in three main areas:

- Live TV Companion

- On-demand browsing

- Authoring second screen content

---

[1] http://www.bbc.co.uk/rd/publications/whitepaper194.shtml
[2] http://notube.tv/author/vickybuser/

1.4   Live TV Companion

1.4.1   iPhone Remote

Our first prototype was a second screen iPhone application that worked with MythTV, an open source software TV receiver, to display information related to the programme. Instead of being specifically designed for that particular piece of software, we designed an API ('Buttons') to generalise over multiple devices and applications. The underlying backend was XMPP (Extensible Messaging and Presence Protocol), an XML messaging framework with identity, contact and group management.

Figure 1.2: iPhone remote in action

From this prototype we learned several things:

- IPhone prototyping is comparatively slow

- It is difficult to automate finding content related to a particular programme, and although identification of a programme takes place in the production of metadata, this identification is not made available in a useful form to end users

- XMPP is a promising technology for syncing between devices, allowing us to punch through firewalls and host infrastructure on external servers

- A substantial amount of infrastructure was missing to make social TV real beyond the obvious and already-existing techniques (such as having a 'Twitter' or 'Facebook' button)

In order to be able to create prototypes quickly, after the first year we moved to HTML / Javascript using an XMPP backend infrastructure and APIs over the top of that. We built some of the infrastructure we needed (for example a crid to pid

mapper[3], allowing us to go from over the air identifiers to HTML, linkable, human-readable content) and tried to persuade people in the industry of the importance of URLs for programmes for word-of-mouth communication, metadata as an advert for content, and the usefulness of APIs to TV.[4]

Although we did create some prototypes to be used with live TV (for example, 'Votey', a application to allow real-time voting and visualisation of votes for a particular programme [5]), and an autocompleting twitter API for TV[6] most effort after this period has been spent on considering the various problems of on-demand content, particularly the problem of collective choice within large corpuses of content.
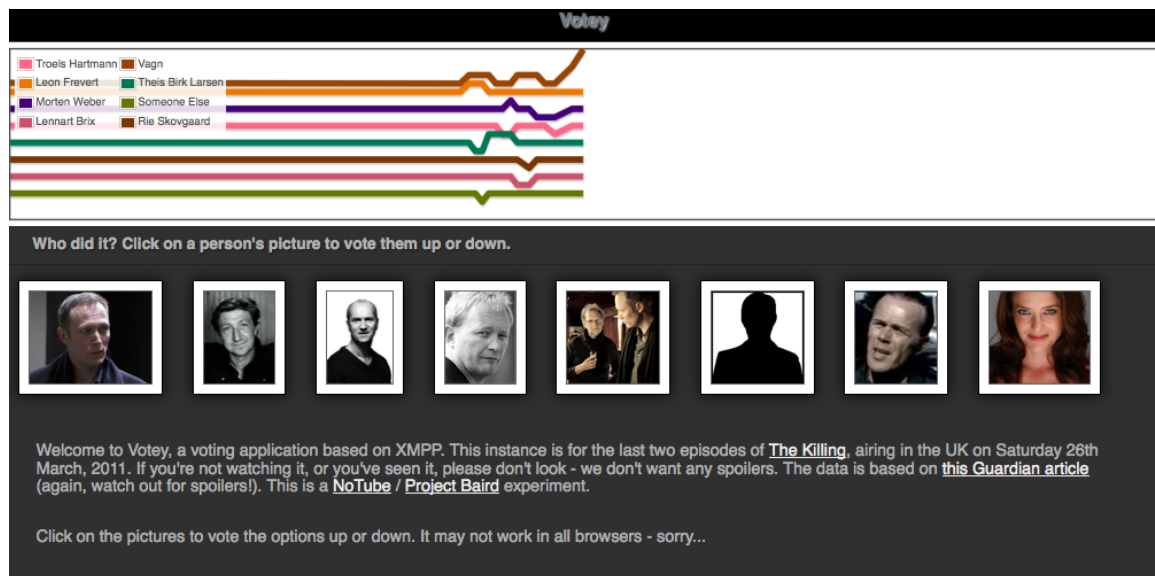

Figure 1.3: 'Votey' - Real-time graphing of votes

1.5   On-Demand Prototypes

We have made two main prototypes in the on-demand area, both centred around problems of choice.

1.5.1   Archive Browser

This is an HTML-based browser of a subset of BBC's Redux project[7] designed to test experimentally if people spent more time browsing and adding to a playlist if related content was presented to them[8]. We did a randomised trial using 90 volunteers from within the BBC. The results were inconclusive, although people liked the approach whichever group they were placed in - but one interesting thing did emerge: people

---

[3]http://notube.tv/2010/08/26/connecting-broadcast-tv-and-the-web-using-a-resolver/

[4]http://www.w3.org/2010/11/web-and-tv/papers/webtv2_submission_60.pdf

[5]http://jabber.notu.be/thekilling/

[6]http://notube.tv/2010/11/22/epg-autocompleter/

[7]http://www.bbc.co.uk/blogs/bbcinternet/2008/10/history_of_the_bbc_redux_proje.html

[8]http://notube.tv/2011/07/01/finding-interesting-new-programmes-trial-results/

like having a 'random' (or 'shuffle') button. We also made the data for this experiment open for others to download and use. [9].
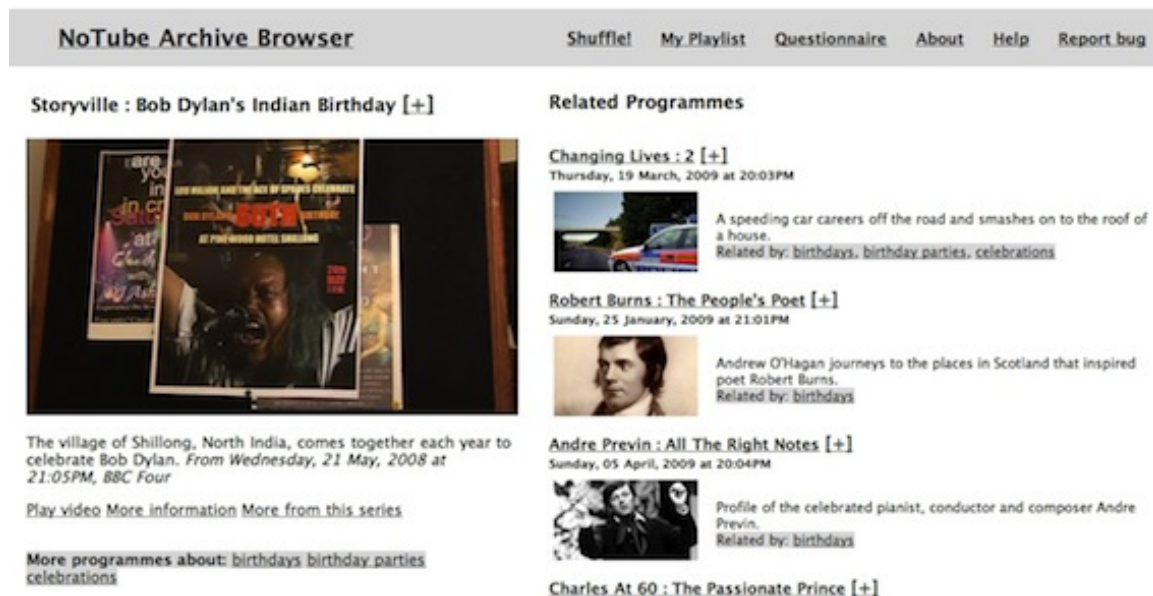

Figure 1.4: NoTube Archive Browser

1.5.2 'N-Screen'

N-Screen explicitly addresses issues of choice in a social situation. Because people have preferences about each others' mental states (i.e. whether they will also enjoy a programme), choosing in small groups can be very difficult.

N-Screen is a second screen HTML / Javascript application that allows people to express their preferences to each other directly, by dragging and dropping content to each other individually or as a group. Group members can be in the same room or physically distant, and one or more TVs can be synced to watch together remotely. It is primarily for tablets and laptops, but runs on anything with a modern Web browser; from smartphones to touch-tables and desktop PCs. It is designed for multiple screens in use while watching TV, explicitly addressing the 'second screen' issues (but taking it a logical step further), but also some of the social aspects of TV watching, ones that we believe have not been addressed elsewhere.

We have made three different version of N-Screen with three different sets of content: Redux, TED Talks, and iPlayer[10], with different approaches to content to content recommendation encapsulated within each, depending on the data available[11].

We are indebted to the VU for their work in arranging for design help, and to Fabrique[12] for designs to improve the appearance of N-Screen.

---

[9]http://notube.tv/2011/02/15/linking-wikipedia-and-bbc-programmes/

[10]http://wp.me/p1fkjI-tR

[11]http://wp.me/p1fkjI-uq
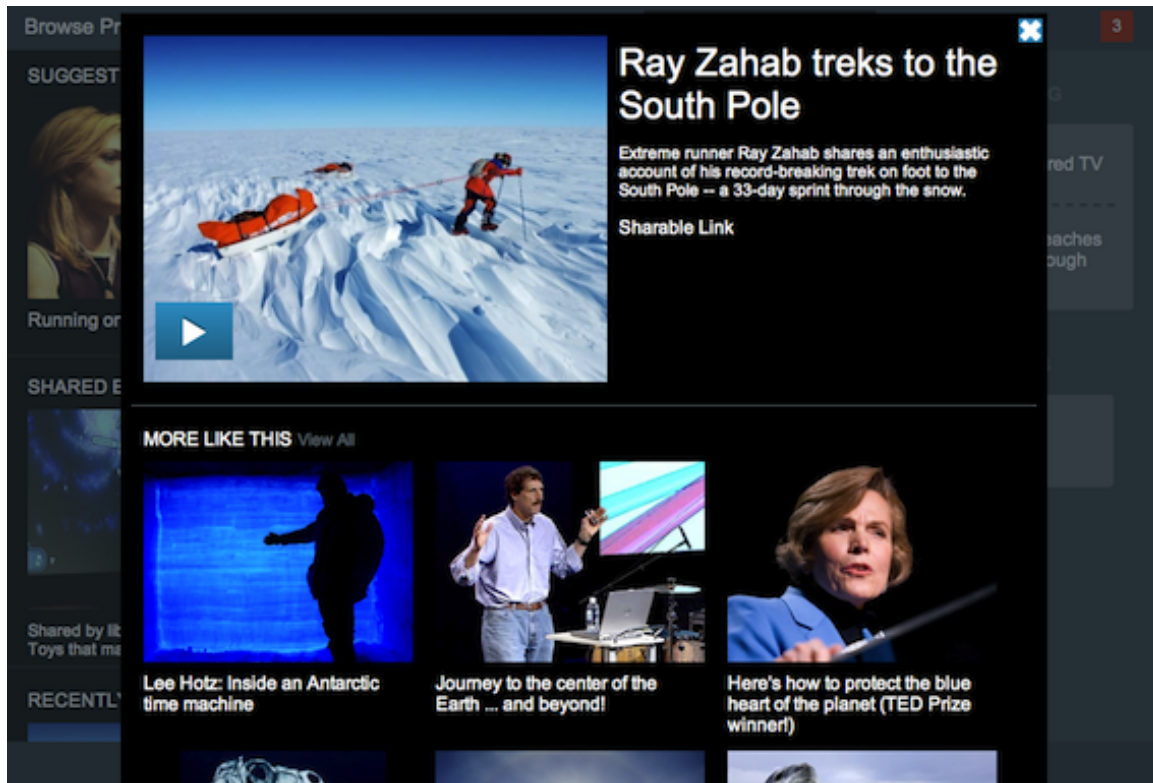
[12]http://www.fabrique.nl/

Figure 1.5: N-Screen - small-group collaborative browsing of video collections

1.6   Tools for Authoring and Playback of Second Screen Content

Our final set of demonstrators starts to consider some of the problems with authoring second screen content, and whether collaboration can help. It uses the same infrastructure as N-Screen but shows some of the different ways it can be used.

1.6.1   TV Extras Authoring ('TEA')

TEA lets you author basic extras for the second screen, as you watch a video, using drag and drop from searches of the web. It is a web application built using HTML5. The usecases concern:

- Professional, common-place second screen content authoring within organisations

- Enthusiast annotation

Currently we only address on-demand authoring.

In the application, two or more people can see each others' suggestions for second screen links and review it. Either can then sync the video for playback, so they can review it together. Finally, they can save their work to an intermediate file format for playback on other devices. This intermediate file is in some ways the most interesting part, as it enables expert developers for particular platforms to create second screen playback tools without also having to handle the authoring side.
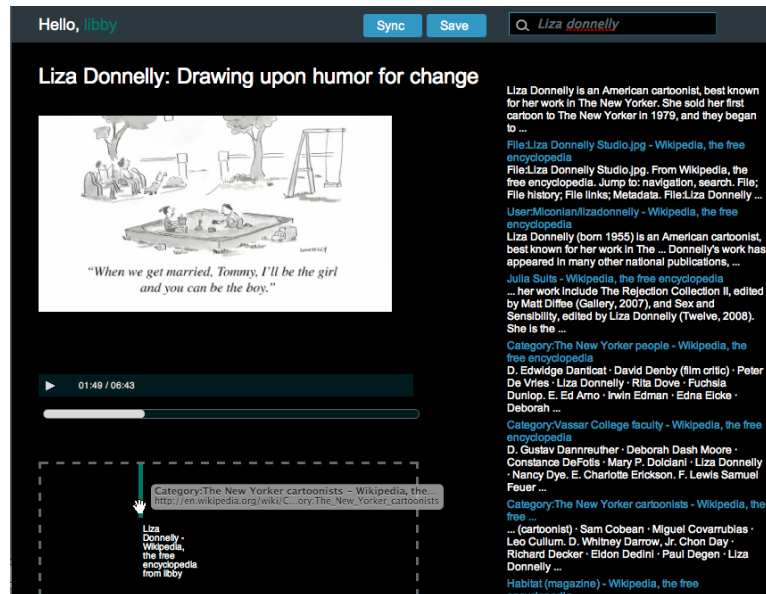
Figure 1.6: 'TEA' - TV Extras Authoring

1.6.2 TEAPlayer

TEAPlayer is an experimental iPhone application built with Phonegap[13](so the bulk of it is HTML) to play the output format of TEA on XBMC (XBox Media Centre). The idea was to see if the information in the file format produced using TEA was sufficient to be made into a useful tool. The result is an iPhone application that looks for XBMC (a video player) instances on the same network, and allows the user to choose from a selection of videos on an iPhone to play on it. If she chooses an annotated video, it loads the URLs on the iPhone at the correct time during playback. It also enables her to pause and play the currently playing video.

Figure 1.7: 'TEAPlayer' - Second screen video content player
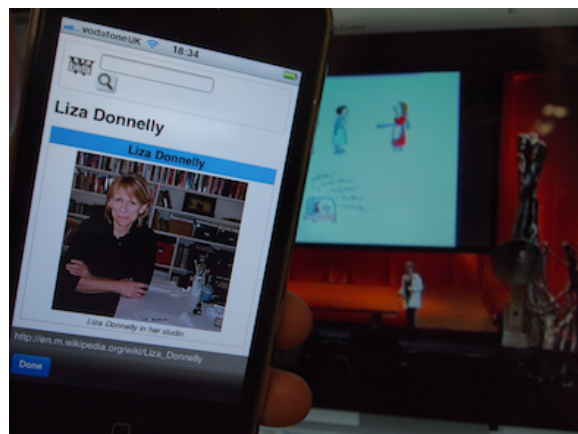
---

[13]http://www.phonegap.com/

Figure 1.8: 'TEAPlayer' - Second screen video content player

1.7   What have we learned?

Our conclusions remain the same as those expressed a year ago in our paper to the Web and TV workshop in Berlin (with Mo McRoberts)[14]. We enumerated the key problems facing developers of second screen content:

- (a) How do we know what the person is watching?

- (b) How do we locate additional information about the programme?

- (c) How do we locate apps or Web pages related to it?

- (d) How can we manipulate it?

We proposed three parts of a solution

1.7.1   Use metadata as an advert for the content that can flow out into the public Web in search of viewers

1.7.2   Create URLs for the content items

1.7.3   Agree on open APIs for controlling the TV and getting access to metadata from it

The only thing we would add to that at this stage is that we failed to completely understand and emphasise the huge gap between web developers and TV developers. To harness the creativity of the vast number of web developers, a TV API needs to be extremely easy to use and understandable within a page or two of documentation. This is a very, very different way of working to TV developers, and there is a large chasm of understanding between the two groups.

It was very disappointing to see a plethora of 'Apps' and APIs to TV at IBC 2011, with no standard in place to mitigate the effect of this diversity on consumers and developers.

---

[14]http://www.w3.org/2010/11/web-and-tv/papers/webtv2_submission_60.pdf (pdf)

1.8   Next steps

In the final three months of the project we want to tackle the following final tasks:

1.8.1   Integration of personalised recommendations

N-Screen provides the capacity for several different kind of recommendations, including personalised ones from a source such as Beancounter. We are mid-way though a lightweight integration of Beancounter and N-Screen, using Beancounter as a data-source for N-Screen and vice-versa.

1.8.2   Evaluation and user testing

Two pieces of user testing are planned:

- General comparison of second screen and on-screen information

- User evaluation of N-Screen

1.8.3   Refining the TEA file format

We are researching possible existing formats for describing URL-based annotation at points in a video.

1.8.4   Completing and publishing an implementation of the 'Buttons' Javascript API

This will allow developers to use second screen applications without worrying about the detail of the XMPP infrastructure.