



NoTube

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

FP7 – 231761

D7c.4 Social TV Final Integrated Prototype

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:	1.0
Date:	October 26, 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 Final Integrated Prototype
Work Package	Number	7c	Title	Internet TV in the Social Web

Date of Delivery	Contractual	M33	Actual	31-10-11
Status	1.0		final <input type="checkbox"/>	
Nature	prototype <input type="checkbox"/> report <input checked="" type="checkbox"/> dissemination <input type="checkbox"/>			
Dissemination Level	public <input checked="" type="checkbox"/> consortium <input type="checkbox"/>			

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 multiple connected screens. We have gone significantly beyond the state of the art by creating standards-based prototypes that allow small and large groups to collaborate and interact in real time while watching TV or finding interesting things to watch and share. We argue that using metadata as an advert for TV programmes, unique URLs for TV, and device independent APIs to TV and TV-like devices 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 26, 2011	1	Libby Miller	1.0

PROJECT CONSORTIUM INFORMATION













Participant's name	Partner	Contact
Vrije Universiteit Amsterdam	 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.		Marco Bruni Phone: +39 06 45472503 Email: marco.bruni@pro-netics.com
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-networks.com
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	6
1 INTRODUCTION	7
1.1 Deliverable Objective	7
1.2 Approach	7
2 WHAT HAVE WE BUILT?	8
2.1 Live TV Companion	8
2.2 On-Demand Prototypes	11
2.2.1 Archive Browser	12
2.2.2 ‘N-Screen’	12
2.3 Tools for Authoring and Playback of Second Screen Content	16
2.3.1 TV Extras Authoring (‘TEA’)	16
2.3.2 TEAPlayer	16
2.4 Next steps	16
2.5 Integration of personalised recommendations	16
2.6 Evaluation and user testing	18
2.7 Refining the TEA file format	18
2.8 Completing and publishing an implementation of the ‘Buttons’ Javascript API	18
3 WHAT HAVE WE LEARNED?	19
3.1 Use metadata as an advert for the content that can flow out into the public Web in search of viewers	19
3.2 Create URLs for the content items	19
3.3 Agree on open APIs for controlling the TV and getting access to meta-data from it	19

LIST OF FIGURES

2.1	Usecase: Do you want to know more?	9
2.2	Live TV Companion in action	10
2.3	‘Votey’ - Real-time graphing of votes	12
2.4	Usecase: Finding a programme to watch later	13
2.5	NoTube Archive Browser	13
2.6	NoTube Archive Browser with Korean categories	14
2.7	Usecase: Finding things to watch in small groups	14
2.8	N-Screen - small-group collaborative browsing of video collections . . .	15
2.9	‘TEA’ - TV Extras Authoring	17
2.10	‘TEAPlayer’ - TV Extras Authoring Playback	17

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

We defined our approach in our first deliverable, summarising the text of the proposal about this part of the project, which had become outdated :

- 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 was and continues to be 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 (i.e. not just ‘Tweet this to your friends’).

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

This deliverable is about how we chose some obvious problems viewers have, and tried to solve them using ideas that are a step further than what is commercially being produced, but using technology that largely already exists.

2. What have we built?

We have made prototypes in three main areas:

- Live TV Companion
- On-demand browsing
- Authoring second screen content

2.1 Live TV Companion

The basis of our first prototype was: ‘Do you want to know more?’, a reference to the film ‘Starship Troopers’ in which each news item ends with various options for further related exploration, producing the possibility of a non-linear exploration through the available content. We wanted to create the means for viewers to explore what they are watching more deeply using a connected second screen. This type of exploration is a very common reason for using a non-connected second screen while watching TV¹. Our view was that if this could be automated or part-automated, there would be many opportunities for making television more interactive and informative, and also more social. The social aspect of TV watching comes from talking about the programme: opinions but also facts, connections and referencing related material on the Web. An example usecase is illustrated in Figure 2.1.

Our goal was to use unique programme identification, combined with NoTube services, coupled with a direct connection to the TV, to produce relevant and interesting information about the programme being viewed. This information could then be explored on a second screen, and shared. This turned out to be more difficult than we expected.

It was clear even at this early point in the project that second screens were the most appropriate mechanism from a user experience point of view for exploring programmes more deeply, including more socially. However, typically, consumer-orientated TVs and TV-like devices such as PVRs (Personal Video Recorders) only have one external-developer-accessible control interface, which is the infra-red remote control. For our prototype to work we needed both to be able to control the TV and also to be able to identify what programme was showing on the TV. In the past year this problem of synchronisation of screens has been tackled using work-arounds such as audio-watermarking². We took a more direct approach, and identified a TV software platform that we could develop with, which also had control and feedback APIs accessible to the more casual developer.

Our first prototype was a second screen application that worked with MythTV, an open source software TV receiver, to display information related to the programme. Instead of being specifically designed for any particular platform, we designed an API (‘Buttons’) to generalise over multiple devices and applications. We were able to use MythTV’s sockets API to find connect the two devices in both directions. We created a native iPhone application in order to use the device’s capabilities for discovery of

¹See <http://www.lostremote.com/2011/10/13/what-tv-viewers-are-doing-on-their-tablets-smartphones/> and <http://ovum.com/2011/10/04/unlocking-the-second-screen/>

²For example <http://www.bbc.co.uk/blogs/researchanddevelopment/2011/09/orchestrated-media-based-tv--.shtml>

JANA IS LISTENING TO THE RADIO...



Figure 2.1: Usecase: Do you want to know more?

available TV devices. The underlying synchronisation platform was XMPP (Extensible Messaging and Presence Protocol), an XML messaging framework with identity, contact and group management.



Figure 2.2: Live TV Companion in action

From this prototype we learned several things:

- It is difficult to automate finding content related to a particular programme, and although unique identification of a programme takes place in the production of metadata, this identification is not made available in a useful form to end users. In order to be able to present the user with interesting information about the programme they are watching it needs to be specific to that programme, and not about another programme from the same series, or about the series itself, or about something else which happens to have the same name.
- XMPP is a promising technology for syncing between devices, allowing us to punch through firewalls (enabling usecases like ‘record this programme for me at home when I’m at work’) and host infrastructure on external servers
- iPhone prototyping is comparatively slow, and (obviously) limits the devices that can be used
- 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)

The main piece of infrastructure missing had to do with the specific identification of programmes. We needed the ability to go from the ‘CRID’ (Content Reference Identifier³) which is broadcast over the air as part of the EIT metadata (Event Information Table⁴) to information that people could read or share. Heavily inspired

³http://en.wikipedia.org/wiki/Content_Reference_Identifier

⁴http://en.wikipedia.org/wiki/Program_and_System_Information_Protocol

by RadioDNS⁵, and collaborating with Mo McRoberts of Project Baird⁶, we wrote a resolver for BBC programmes that takes a CRID and returns the unique BBC URL for the programme and vice-versa⁷, bridging the gap between the two worlds in the sense that if a user has the CRID and access to the Web they can find out more about the programme, and share that link with others using social media; and in some circumstances if they have the programmes URL they can also switch to watching the programme itself via the CRID.

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 on top of that. We built some of the infrastructure we needed and tried to persuade people in the industry of the importance of URLs for programmes for social web communication, metadata as an advert for content, and the usefulness of APIs to TV.⁸

Although we have created other prototypes to be used with live TV (for example, an application to allow real-time voting and visualisation of votes⁹, Figure 2.3), and an autocompleting twitter API for TV¹⁰, most effort after this period has been spent on considering the various problems of on-demand content, particularly the problem of choice within large corpuses of content. There are a number of interesting problems in this area, particularly as the number of on-demand videos is growing very rapidly¹¹. This decision was also taken because of the problems we had creating reliable and sustainable demonstrators using the MythTV setup and the lack of alternatives for playing live TV, and that we could not make much more interesting progress in this area.

2.2 On-Demand Prototypes

We have made two main prototypes for on-demand video content, both centred around problems of choice.

When faced with a large number of potential options, many people find it difficult to choose, and give up quickly¹². For video content owners this means that much of their content may never get the attention of people who are interested in it, either directly, or via social media sharing.

The first prototype ('Archive Browser') was designed to test whether presenting *related* content could make people browse longer. The second one ('N-Screen') attempts to harness the 'look what I found' impulse for the same purpose.

Both of these applications are Web-based, and can be used as second screen applications, with another web-page acting as the 'TV'.

⁵<http://radiodns.org/>

⁶<http://projectbaird.com/discovery/tvdns/>

⁷<http://notube.tv/2010/08/26/connecting-broadcast-tv-and-the-web-using-a-resolver/>

⁸http://www.w3.org/2010/11/web-and-tv/papers/webtv2_submission_60.pdf

⁹<http://jabber.notu.be/thekilling/>

¹⁰<http://notube.tv/2010/11/22/epg-autocompleter/>

¹¹http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html

¹²see <http://flowtv.org/2010/10/choice-fatigue-community-and-the-mutations-of-television/> and http://en.wikipedia.org/wiki/Least_objectionable_program

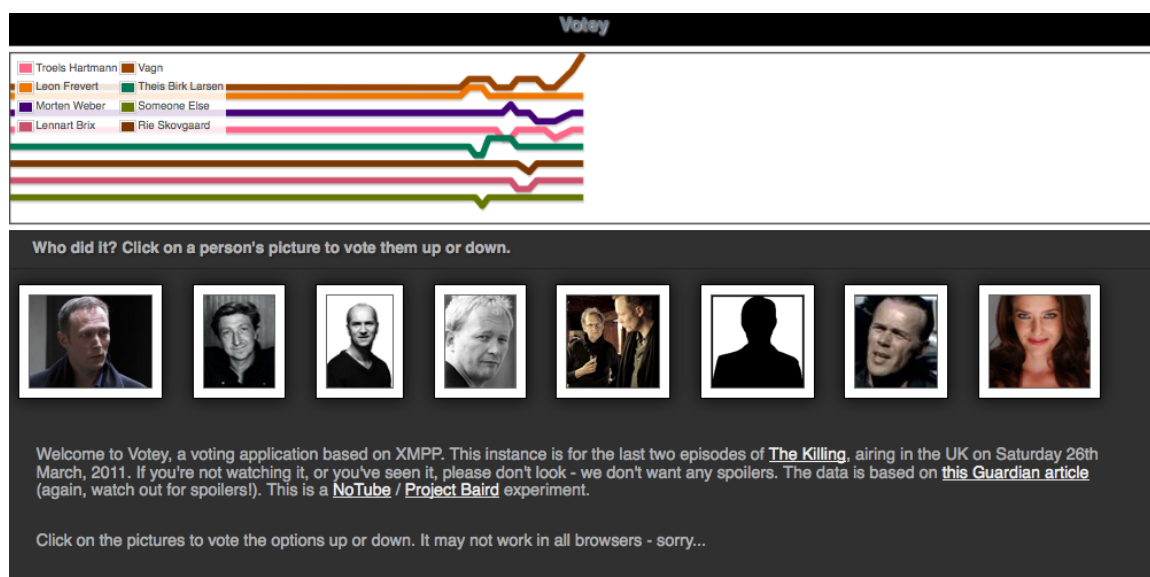


Figure 2.3: ‘Votey’ - Real-time graphing of votes

2.2.1 Archive Browser

The Archive Browser addresses the usecase of finding something interesting to watch later. It could be applicable to any on-demand archive, for example BBC’s iPlayer system. It is illustrated in Figure 2.5.

This particular version is an HTML-based browser of a subset of BBC’s Redux project¹³. We designed it to test experimentally whether people spent more time browsing and adding to a playlist if related content was presented to them. We tested this in 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. One interesting thing did emerge: people like having a ‘random’ (or ‘shuffle’) button¹⁴.

We also made the data for this experiment open for others to download and use¹⁵. We were also able to combine this prototype with the kt Korlex Korean Wordnet using mappings between the BBC’s classifications system, Lonclass¹⁶, in order to be able to search and browse the subset of this archive in Korean (see Figure 2.6).

2.2.2 ‘N-Screen’

N-Screen explicitly addresses issues of choice in a social situation. Because people have preferences about each others’ mental states as well as their own (i.e. in small groups they make decisions based on whether other people present will also enjoy a programme), choosing in small groups can be very difficult. The usecase is illustrated in Figure 2.7.

N-Screen is a second screen HTML / Javascript application that allows people to express their preferences to each other directly, by dragging and dropping items to

¹³http://www.bbc.co.uk/blogs/bbcinternet/2008/10/history_of_the_bbc_redux_project.html

¹⁴<http://notube.tv/2011/07/01/finding-interesting-new-programmes-trial-results/>

¹⁵<http://notube.tv/2011/02/15/linking-wikipedia-and-bbc-programmes/>

¹⁶<http://en.wikipedia.org/wiki/Lonclass>

NOTUBE ARCHIVE BROWSER PROTOTYPE

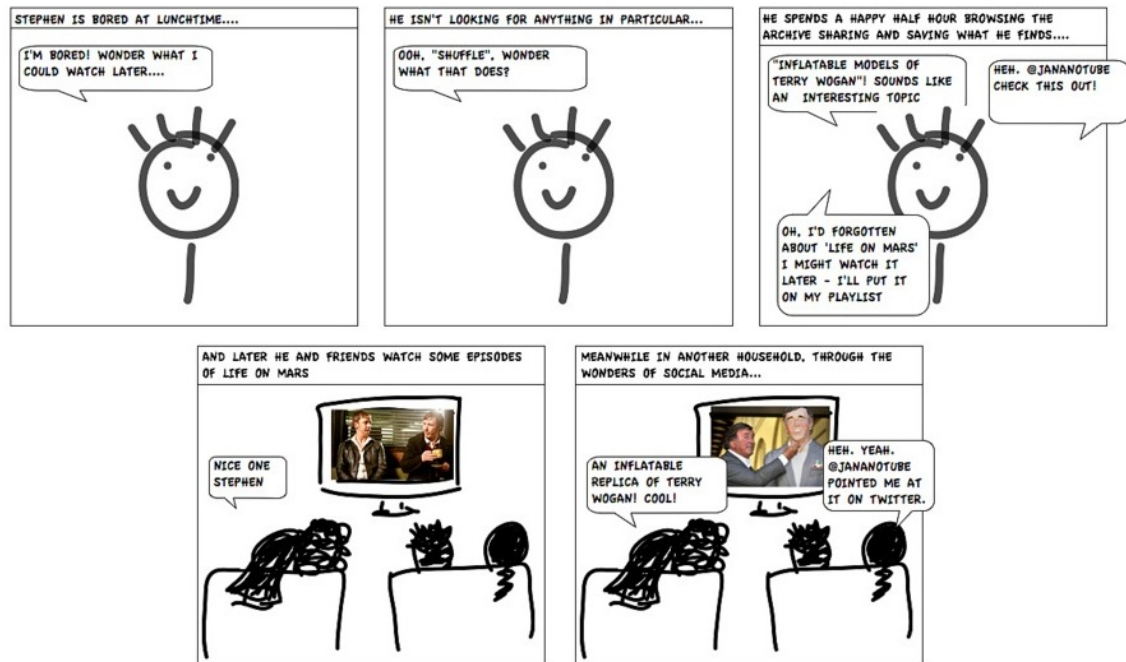


Figure 2.4: Usecase: Finding a programme to watch later

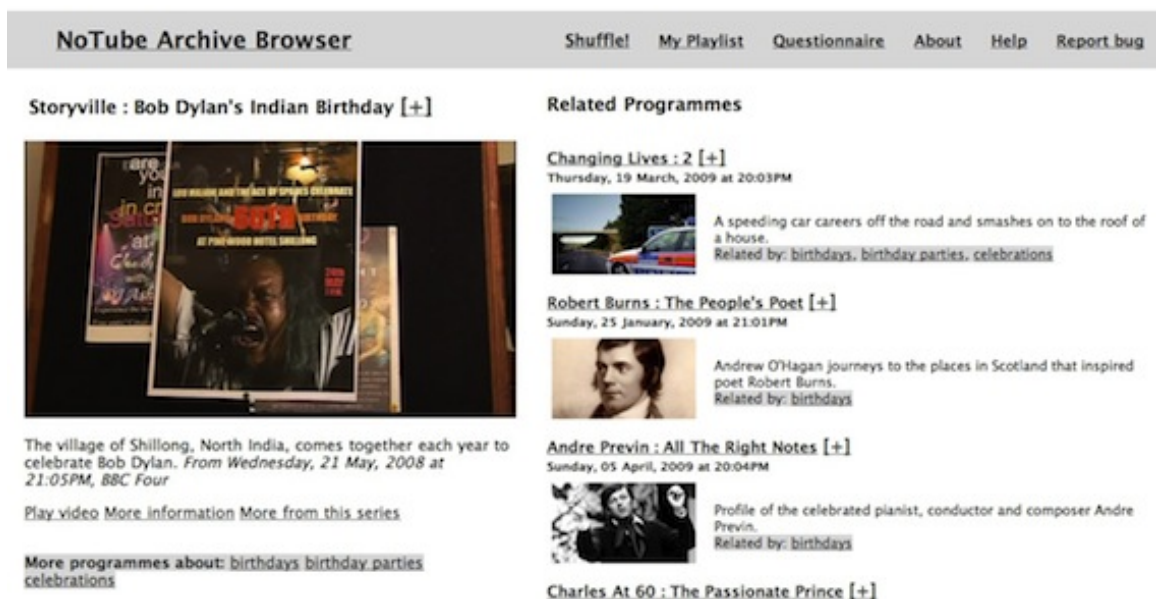


Figure 2.5: NoTube Archive Browser

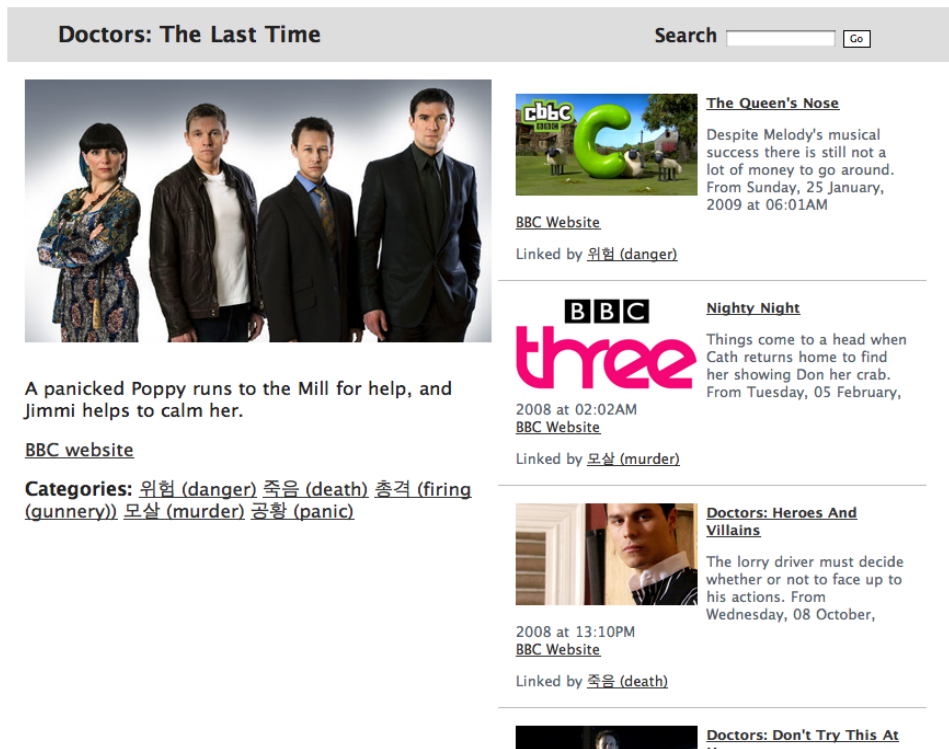


Figure 2.6: NoTube Archive Browser with Korean categories

NOTUBE'S 'N-SCREEN' PROTOTYPE FOR COLLABORATIVE EXPLORATION OF ON-DEMAND PROGRAMMES

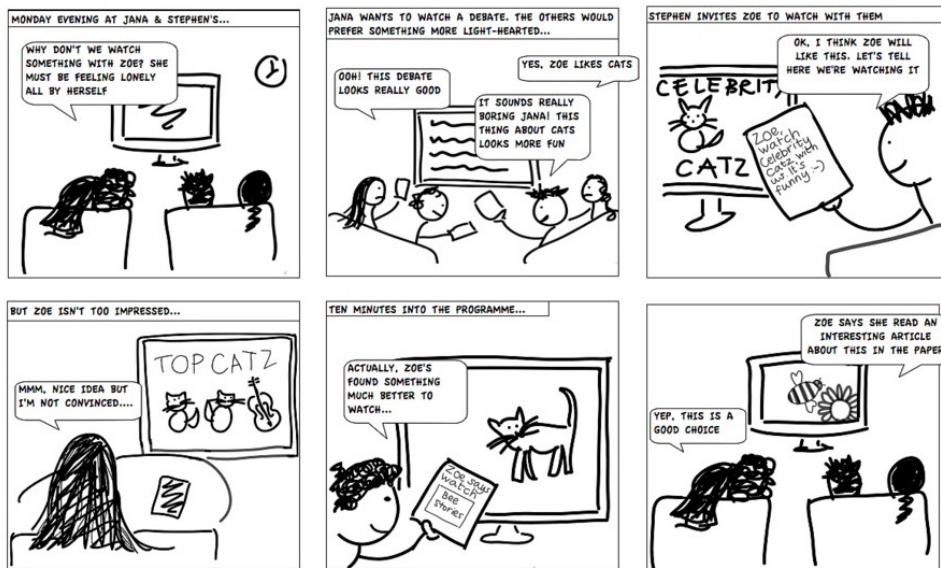


Figure 2.7: Usecase: Finding things to watch in small groups

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.

We have made three different version of N-Screen with three different sets of content: Redux, TED Talks, and iPlayer¹⁷, with different approaches to content to content recommendation encapsulated within each, depending on the data available¹⁸. Figure 2.8 is a screenshot of N-Screen.

We are indebted to the VU for their work in arranging for design help, to Fabrique¹⁹ for designs to improve the appearance of N-Screen, and to Danny Ayers²⁰ for creating custom sounds to improve the drag and drop user experience.

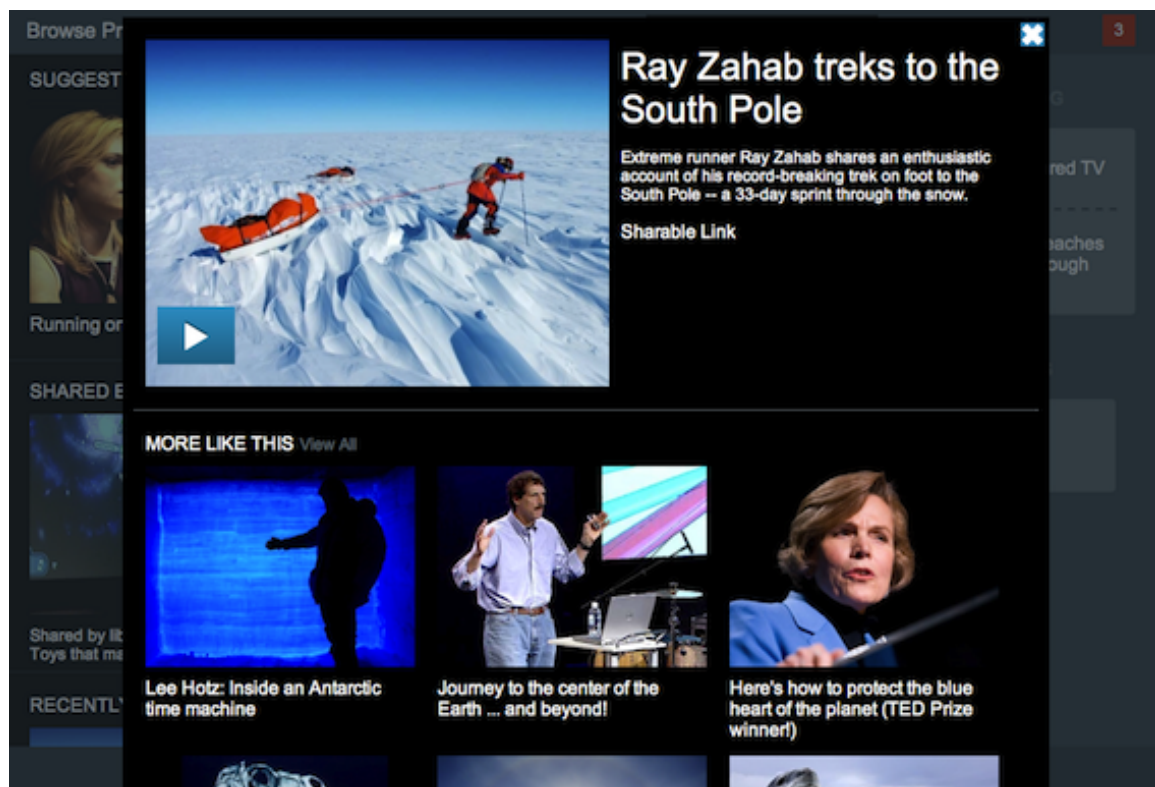


Figure 2.8: N-Screen - small-group collaborative browsing of video collections

2.3 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.

¹⁷<http://notube.tv/2011/10/10/n-screen-a-second-screen-application-for-small-group-exploration-of-on-demand-content/>

¹⁸<http://notube.tv/2011/10/14/algorithms-for-recommendations-in-various-n-screen-implementations/>

¹⁹<http://www.fabrique.nl/>

²⁰<http://dannyyayers.com>

2.3.1 TV Extras Authoring ('TEA')

TEA lets people author basic extras for the second screen, as they watch a video, by dragging and dropping links from the web into the correct position. It is a web application built using HTML5. The usecases we considered include:

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

Currently we only address authoring for on-demand video.

In the application, two or more people can see each others' suggestions for second screen links as they are added in position, in real time. Either person 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.

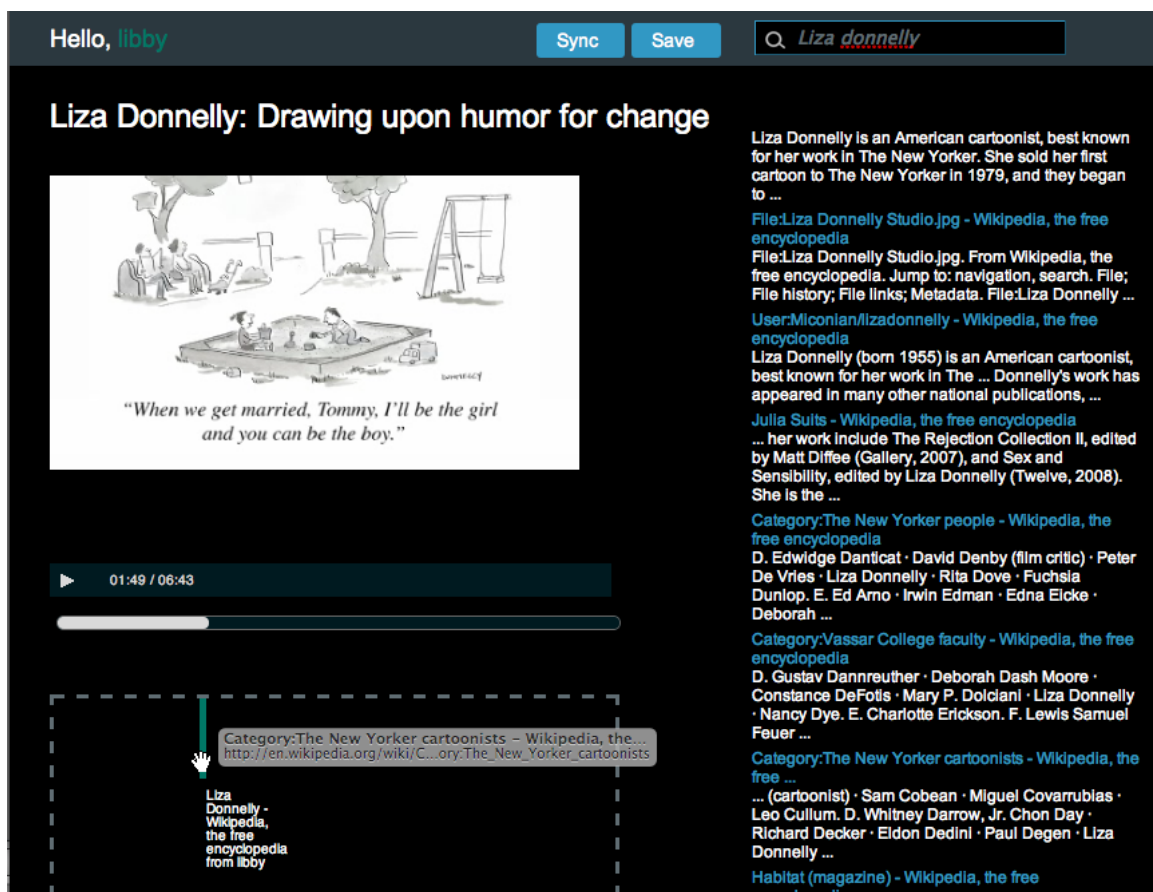
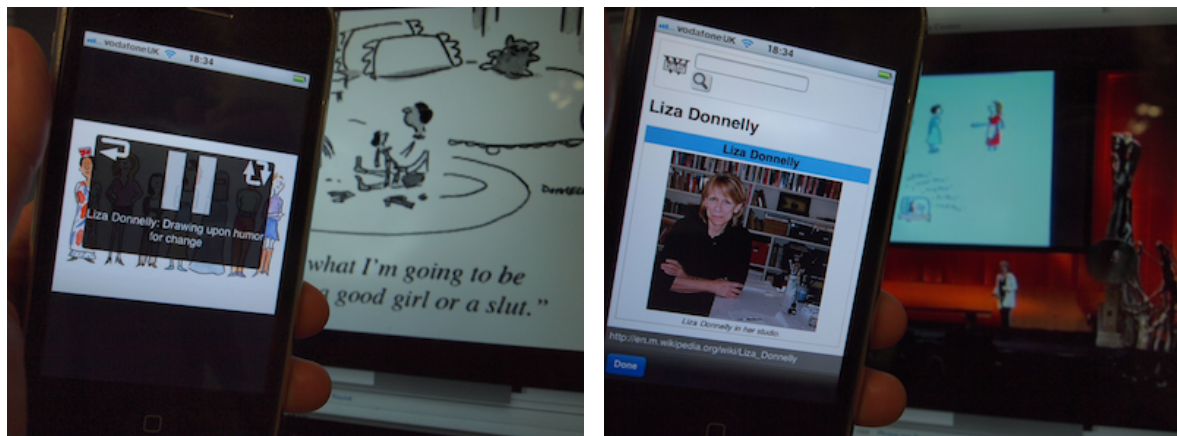


Figure 2.9: 'TEA' - TV Extras Authoring



(a) TEAPlayer showing remote functionality (b) TEAPlayer showing link appearing on second screen

Figure 2.10: 'TEAPlayer' - TV Extras Authoring Playback

2.3.2 TEAPlayer

TEAPlayer is an experimental iPhone application built with Phonegap²¹ (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 - see Figure 2.10b.

2.4 Next steps

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

2.5 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 through a lightweight integration of Beancounter and N-Screen, using Beancounter as a data-source for N-Screen and vice-versa.

2.6 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

²¹<http://www.phonegap.com/>

2.7 Refining the TEA file format

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

2.8 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.

3. What have we learned?

Our conclusions remain the same as those expressed in February 2011 in our paper to the Web and TV workshop in Berlin (with Mo McRoberts)¹. 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

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

3.2 Create URLs for the content items

These first two two help social TV discovery, by giving people a Web reference to use and by letting people see something about what they are talking about even if they cannot (or don't feel inclined to) watch it right away. APIs to see what is playing and identify it, to pause, play and otherwise manipulate a remote TV player enable multiple interesting second screen applications, by enabling synchronisation of second screen content.

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

As we have gradually realised within the project, the worlds of TV and the Web are far apart in many ways, but particularly important to our work has been the difference between Web developers and TV developers, which can be summarised very crudely as experimentation versus reliability. The Web is a platform that rewards and generates novelty and change. The TV world is largely about making reliable appliances. The specifications in each world are appropriately weighted. The attention spans of the developers are sized proportionately to the specifications. Add to this the concerns of content owners about the accessibility of their content to non-rights holders - this applies to metadata as well as video - and it is not surprising that control over the device is restricted. In retrospect, making anything that spans these two worlds in any direct sense was going to be difficult.

What is also clear is that if aspects of the TV experience could be made available to web developers, there could be an explosion of related applications for second screens.

¹http://www.w3.org/2010/11/web-and-tv/papers/webtv2_submission_60.pdf (pdf)

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.

We have tried to demonstrate 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.² 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.³ We have documented our approach and technical work as blog posts, photos and video, and by creating a list of links on delicious, and we have worked in public where possible, so that the work will continue to be useful in the future.

²<http://www.bbc.co.uk/rd/publications/whitepaper194.shtml>

³<http://notube.tv/author/vickybuser/>