

Buttons: Device Control and Communication and its application to the Social Web

Dan Brickley, Libby Miller, Vicky Buser, NoTube Project

December 16, 2010

Integration of TV and the social web is already happening. People are increasingly using online social networks to talk about TV, nearly all via second screens. We believe that this is a very important trend because it means that integration of the Web and TV is already happening - and occurring in an unexpected direction. This trend started without any specific tools to support it, but increasingly, applications are being built specially.

For live content, a key problem with applications such as these is synchronisation to a greater or lesser degree: what's playing now, and how can applications identify it, enhance it, and allow user interaction pertaining to it.

Meanwhile, APIs to new forms of TV such as XBMC, Boxee and MythTV are in common use. Boxee in particular has taken a lead in identification of programmes by URL, using links such as BBC iPlayer and IMDB. All these APIs be used with second screen applications to control the software and display information about what's playing, and having open APIs has enabled the creativity in kinds of remote control that can be made. However the proliferation of APIs has meant that each television application has to have a separate control application, duplicating the problem of multiple physical remote controls but in application form.

We argue that a common, two-way open API for TV would enable people to watch, control and interact with television in a way that increases their use and enjoyment of it. In this position paper we briefly describe some features of a straw-man API, 'Buttons' that allows control and feedback from TV. We outline why Buttons or something like it is needed for social web interaction, and briefly describe the current protocol and our implementations of it using MythTV, iPhone and the Web.

Buttons was created as a collaboration between the NoTube and FOAF projects. NoTube is an European Project about the Ubiquitous Web and TV. FOAF is a project to describe people, their friends and interests, on the Semantic Web.

1 TV Trends and the Problem Buttons Solves

The initial problem Buttons solves is that televisions accept control but only communicate back information in visual way. This limits how much information from the TV can be combined with other applications, particularly when using second screens. Here are some of the reasons why we think this is an important problem for TV:

1.1 TV-related Social Networking is Increasing

People are spending more time talking about TV on social networks, particularly regarding live TV events, football, Big Brother, Eurovision and so on - but also smaller local trending topics, particularly in the UK (@@ show example). However because it's *my* social network but *our* TV, shared and private content displayed on a large screen may be uncomfortable socially.

1.2 Second Screens are Growing in Use

The use of a networked second screen is becoming common while watching the TV[@@]. People are using a second screen because they want to interact with the internet while watching. This activity is often, but not always, related to the TV program they are watching. In many cases it is social interaction related to the programme, and sometimes lightweight 'research' (actors' names, locations etc) about the programme, or an unrelated activity with the TV as noise in the background.

1.3 TVs Are Becoming Smarter and More Networked

New kinds of set top boxes are appearing on the market which allow the delivery of TV over IP, i.e. they are connected to the Internet. Some of these have network APIs for control; others are much more closed. Some television sets themselves are now networked.

1.4 TVs Are Still Not Easy to Control

Controlling a TV can involve a number of complex tasks, and some tasks are very difficult or tedious without a keyboard (for example non-numeric text input). Complex tasks with interactions displayed on a screen some distance away are physically more difficult than nearby (@@ref?). For people with any disability these problems are compounded.

2 Buttons

Buttons is an initial set of verbs and a protocol to enable people to control their TVs using the most comfortable and appropriate means available to them, and to make it easier for them to communicate about their media experiences using social networks.

The key features are *control of the TV* and *feedback from the TV about what's playing now*. Our first implementation is in XMPP (a standards-based XML messaging format), chosen for your implementations (see below) because it has a built-in access control mechanism, anonymous capability and LAN discovery possibilities, but other transports would be feasible, for example HTTP. Buttons is not stateful.

A typical message / response for play a programme looks like this:

```
<body xmlns='http://jabber.org/protocol/httpbind'>
  <message to='gumbovi@jabber.notu.be' from='19298842691292502097401981@jabber.notu.be'
    type='iq' xmlns='jabber:client'>
    <body>
      {
        'command': 'play',
        'channel': 'http://www.bbc.co.uk/bbctwo/'
      }
    </body>
  </message>
</body>
```

```
<body xmlns='http://jabber.org/protocol/httpbind'>
  <message to='gumbovi@jabber.notu.be' from='19298842691292502097401981@jabber.notu.be'
    type='iq' xmlns='jabber:client'>
    <body>
      {
        "url": "http://www.bbc.co.uk/programmes/b00wwdpr",
        "title": "The Daily Politics",
      }
    </body>
  </message>
</body>
```

```
"channel": "http://www.bbc.co.uk/bbctwo/",
"description": "Andrew Neil and Anita Anand present the top political stories of the day.",
"start_time": "2010-12-16T12:00Z",
"end_time": "2010-12-16T12:30:00Z",
"time_elapsed": "1628"
}
</body>
</message>
</body>
```

3 Applications

We have built several applications around the general framework of XMPP, using both XMPP over HTML and application specific libraries to control both TV applications (MythTV, XBMC) and web-based video players.

@@descriptions and pictures@@

4 Conclusions

@@blah@@

5 Links and References