R&D White Paper

WHP 127

December 2005

Digital TV Hacks

S.J.E. Jolly, G.D. Smith *and* A.K. Sheikh

Research & Development
BRITISH BROADCASTING CORPORATION

BBC Research & Development
White Paper WHP 127

Digital TV Receiver Projects
"Digital TV Hacks"

Stephen Jolly, Gareth Smith, Alia Sheikh

Abstract

We describe a simple, cost-effective way of receiving DVB Transport Streams and processing them, to extract their internal data streams and repurpose (or "hack") them in new and interesting ways. An example processing chain is detailed, in which information from subtitles is used to generate a graph depicting the social relationships within a TV programme. Some suggestions for innovative uses of broadcast content are offered to the reader.

Additional key words: PieSpy, Project X, DVB Tools, DVB, VLC, SubStream, Open Tech 2005

BBC Research & Development
White Paper WHP 127

Digital TV Receiver Projects
"Digital TV Hacks"

Stephen Jolly, Gareth Smith, Alia Sheikh

1 Introduction

As the cost of technology continues to fall, people are increasingly choosing to install DVB[1] receiver cards in their home PCs. In addition, Personal Video Recorders (PVRs) and Set-Top Boxes (STBs) now exist that give their users direct access to the raw digital TV datastream that is transmitted to their homes. Free tools for processing this data have been developed, and home processing of digital TV broadcasts is now perfectly feasible. In sections 2 and 3 of this document, the necessary software and hardware tools are listed. The rest of the paper is devoted to a number of suggested projects, or "hacks" in the sense of the word popularised by O'Reilly: "A clever solution to an interesting problem." [1]. A full example is described in section 4. A number of suggestions for other projects are given in section 5.

Be aware that there may be copyright implications to the kind of "hacking" advocated in this paper, if extended beyond the purposes of private study and serious research. Many of the projects generate what are known legally as "derivative works" (new content that incorporates other people's, or other people's content in a new form). Broadcasters and copyright-holders generally forbid the publication and distribution of derivative works without explicit permission. Laws and the attitudes of broadcasters vary considerably – please take care.

In this document, we will often talk about "content", by which we mean any data with a creative component – video, audio, subtitles, textual descriptions, etc.

1.1 Digital Television: Multiplexes and Transport Streams

Digital television is broadcast in "multiplexes", each containing a number of TV and radio channels. One reason for this is to allow easy integration with analogue TV broadcasts, since a single multiplex takes up roughly the same bandwidth as a single analogue TV channel.

The DVB group [2] has standardised the format of the stream of data that conveys digital TV multiplexes to the home. The standard is known as the "DVB Transport Stream" (TS). The same stream format is used for satellite, cable and terrestrial broadcasting. The contents of a TS are as shown in Fig. 1. Each TV channel in the multiplex is represented by a "service" in the multiplex, which contains the various audio, video and interactive "elementary streams". The nature of the elementary streams depicted should be obvious, with the exception of the "Audio Description" (AD) stream. This is a service provided with the partially-sighted in mind; it consists of a commentator describing the action taking place on-screen during dialogue-free parts of a programme. Compatible DVB receivers mix the AD track with the normal programme sound.

[1] "Digital Video Broadcasting", the standard used for broadcasting digital television in many parts of the world, including the UK. The acronym also refers to the group responsible for that standardisation.

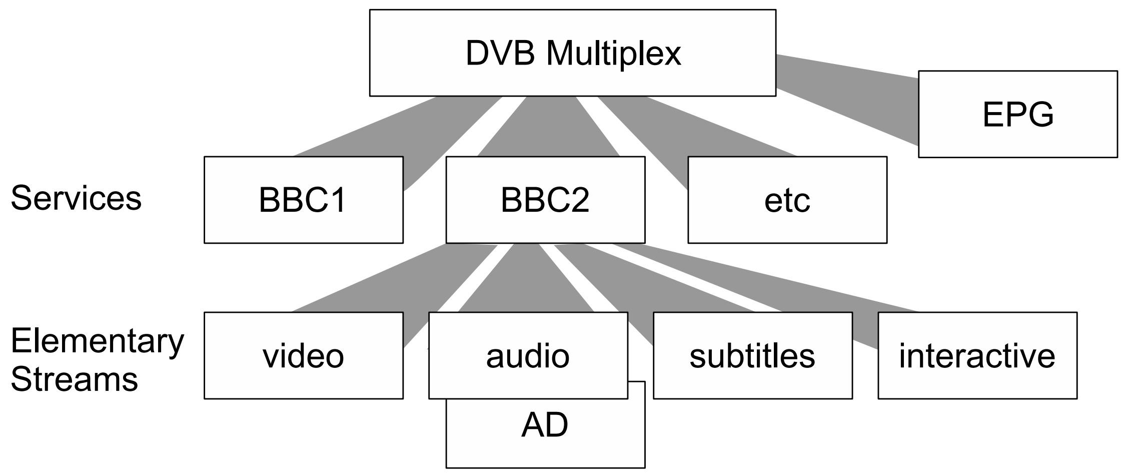

Figure 1: A simplified representation of the main components of a DVB Transport Stream. "AD" refers to an (optional) "Audio Description" audio stream.

There exist "Partial Transport Streams". These consist of a DVB TS with all the elements unrelated to a particular service filtered out. DVB PVRs generally save programmes as partial streams (or as Program Streams, described below), to minimise processsing.

Partial or otherwise, the format of a DVB Transport Stream is that of an MPEG-2 Transport Stream, so tools developed for the latter will usually work on the former. Note however that since DVB streams are an *extension* of MPEG-2 streams, such tools will not support DVB-specific features such as subtitles, EPG data, channel names, teletext etc.

One final complication is the existence of "MPEG-2 Program Streams". A Program Stream is little more than a different way of representing the data in a Transport Stream. Program streams are designed for applications such as DVD Video, in which the elementary streams are synchronous and tightly coupled and the received data can be guaranteed to be practically uncorrupted. Program streams are an alternative to Partial Transport Streams for storing DVB services for later playback, especially if the user may also wish to turn them into DVDs.

2 Acquisition

The first step in the processing chain is to acquire a Transport Stream. There are three straightforward ways to do this:

2.1 DVB Receiver Card

A large number of PCI and USB devices are now available that allow you to use your PC to receive Digital TV. At the time of writing, prices in the UK started from around £50.

If you have a Linux PC, this is probably the simplest and easiest way to start experimenting with Digital TV. The Linux 2.6.x kernel contains a well-regarded DVB driver subsystem [3] with support for the majority of PCI cards and USB devices on the market. A compatibility list is maintained at `http://www.linuxtv.org/wiki/index.php/Supported_DVB_cards`.

Mac users are also well-served by open-source software – drivers for a number of DVB cards are available, along with software capable of saving MPEG-2 program streams to disk [4].

Windows users are less lucky: a search revealed no free software capable of saving MPEG streams to disk intact. Most applications (including the excellent Media Portal [5] PVR software) make use of libraries provided by Microsoft, which only support saving to disk in the proprietary DVR-MS format.

2.2 PVRs

Most Personal Video Recorders (PVRs) store their recorded content in the form of Partial Transport Streams, but gaining access to them often requires the physical removal of the devices' hard disks. A notable exception is the Topfield TF5800PVR [6], which amongst many other unique features permits partial streams to be transferred to and from it via a USB link.

One disadvantage of adopting such a solution is that live, unfiltered streams are unavailable.

2.3 STBs

Some set-top boxes allow DVB streams to be sent to a PC for processing or storage. Dream-Multimedia-TV GmbH [7] make the DreamBox [8] family of devices, which are capable of sending transport streams to a PC via a local network.

3 Transport Stream Processing

Transport stream processing is the extraction of the desired components from a transport stream containing a DVB multiplex (or partial multiplex). Currently, only a limited number of free tools are available for this purpose.

3.1 Project X

Project X [9] is a Java application designed specifically for demultiplexing DVB transport stream files. It has both command-line and GUI interfaces, and supports DVB-specific extensions to the MPEG-2 Transport Stream standard such as teletext and DVB subtitles. Being written in Java, it runs on all major platforms, including Linux, Windows and Mac. The software is open-sourced, with the source code available under the GNU General Public License [10] (GPL).

3.2 VLC

VLC [11] is an extremely flexible media player, transcoder and streaming server. It can extract audio and video from a transport stream, transcode[2] them on-the-fly and serve the result to multiple users over a network using one of a number of common streaming protocols.

The source code to VLC is also available under the GPL.

3.3 DVB Tools

"DVB Tools" [15] is another GPL-licensed Open Source project, providing tools that interface to Linux's DVB subsystem. It contains a number of applications at various stages of development: a tuning utiltiy called "dvbtune", an RTP streaming server called "dvbstream" (with the additional ability to pipe transport streams to files and other tools) and other utilties for converting transport

[2]Transcoding is the process of converting media into a different format. A common transcoding operation is to convert MPEG-2 video into a more efficient format such as MPEG-4 layer 2 (using the DivX [12] or XviD [13] codecs) or Dirac [14].

streams to program and elementary streams, extracting DVB subtitles, etc. The project appears to be updated infrequently.

3.4 PVAStrumento

PVAStrumento [16] is a closed-source freeware tool for Windows for converting DVB transport streams into MPEG-2 program streams. It is mentioned for completeness.

4 Example

As an example of the kind of interesting project that it is possible to create with free tools and limited time, a social network analyser for TV programmes was created.

Social networks are the webs of relationships that we build every day as we interact with friends, family and colleagues. Tools exist for automatically graphing these relationships, given a suitable method of monitoring interactions. One such tool is "PieSpy" [17], a GPL-licensed tool originally written for graphing social networks on IRC[3] channels.

We chose to generate social network graphs for TV programmes by using subtitles as the input to a modified version of PieSpy. We started by copying the partial transport stream containing a recorded programme from a Topfield TF5800. We ran this through Project X to extract the subtitles as a sequence of bitmapped image files. This is the native format for the DVB subtitles broadcast terrestrially in the UK, and presents a problem, since the social networks can only be inferred from the contents of the subtitles: the text.

To convert the images into text, an OCR package was used. GOCR [18] was chosen for this task, being the only suitable package available. The results it gave were variable, especially where spacing was concerned, but since social network graphs are qualitative rather than quantitative, the corresponding (fairly low) rate of errors could be tolerated.

Once the subtitle text had been recovered it was turned into complete sentences, which were fed one-by-one into PieSpy. Piespy itself was modified in three significant ways. Its input was provided by the subtitle stream rather than an IRC channel, its analysis "heuristics" were replaced by a simple algorithm looking for occasions when more than one pre-defined keyword occurred in a sentence, and the nodes in the graph were those same keywords, rather than being a list of the members of an IRC channel.

An example social network graph is shown in Fig. 2. The subtitles were taken from the 2005 French Grand Prix. Since the subtitles were transcribed from a *commentary* feed, the nodes in the graph reperesent the social entities under discussion: the drivers and teams taking part in the race. A link between two nodes is created whenever two entities are talked about in the same sentence. On the grounds that there has to be some reason why the commentators choose to make those links, we therefore infer that they represent "interactions" between social entities that have affected the outcome of the race in some sense.

The code used to generate Fig. 2 was written in a weekend, to demonstrate how easy such projects can be. A "cleaned-up" version that additionally allowed the state of the subtitle stream to be inspected graphically between each processing stage was written in another two weeks, and demonstrated at Open Tech 2005 [19]. The source code for this demonstration has been released [20] under the GPL for the benefit of anyone who wants to try experimenting for themselves.

[3]Internet Relay Chat, a community-oriented instant messaging system.

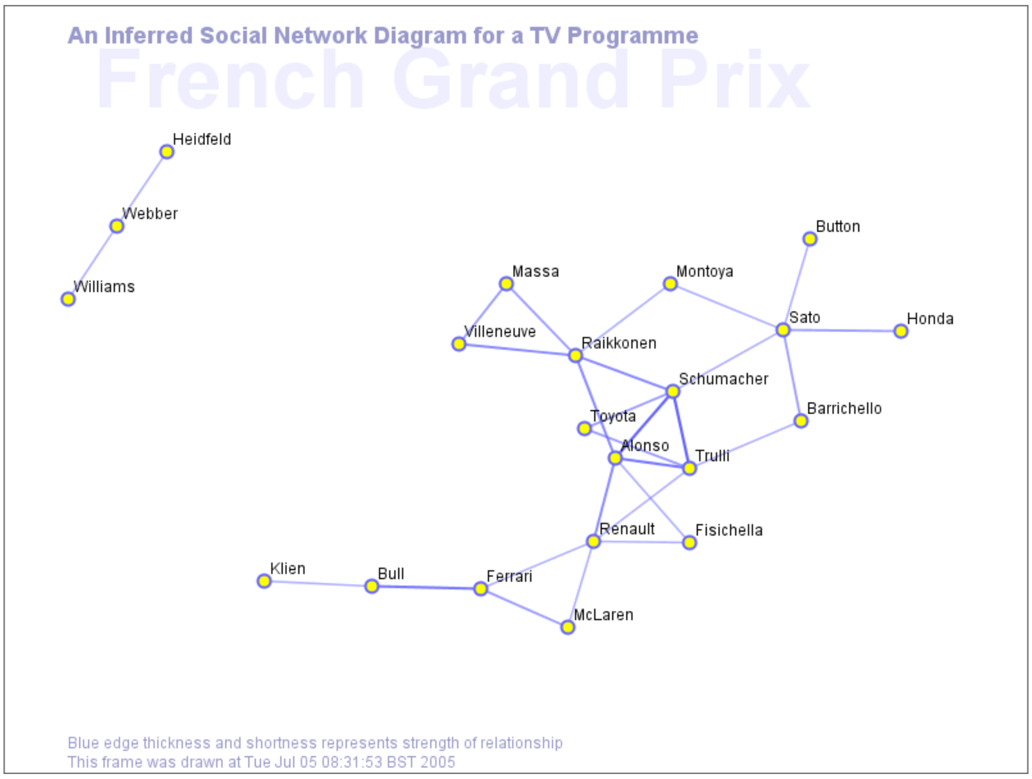
An Inferred Social Network Diagram for a TV Programme
French Grand Prix
Heidfeld
Webber
Williams
Button
Massa
Montoya
Sato
Honda
Villeneuve
Raikkonen
Schumacher
Barrichello
Toyota
Alonso
Trulli
Renault
Fisichella
Klien
Bull
Ferrari
McLaren
Blue edge thickness and shortness represents strength of relationship
This frame was drawn at Tue Jul 05 08:31:53 BST 2005

Figure 2: A social network graph showing the interactions between drivers and teams in a Formula 1 race

5 Possibilities

The options for doing new, interesting, exciting and unexpected things with Digtal TV content are endless. Such possibilites greatly increase the utility of broadcast TV content for private study purposes and serious research. A few ideas are listed below for inspiration.

5.1 Personalise the EPG

- You could track TV trends by looking at the incidence of specific words (or groups of words) in the EPG data that describes upcoming TV programmes. Plot a histogram showing how the trends change over time. Use it to justify accusations of "celebritisation" of television simply by tracking the word "celebrity". (Assuming you observe an increase, that is!)

- Create a "top ten" list of people on television according to how many programmes per month they appear in. Again, you'd be tracking their appearance in EPG programme descriptions. Watch people's careers rise and fall.

- Create a new EPG, with blank spaces where programmes starring people you hate should be. Implement procmail for television.

5.2 Watch TV in New Ways

- Create "podcasts" for your mp3 player from TV programmes by mixing together the soundtrack and Audio Description track (described in section 1.1. "Watch" television on devices without displays.

- Create a slideshow by taking a screenshot of the video and subtitles every time the subtitles change. This lets you watch TV on devices with displays but no video or audio capabilities such as budget PDAs.
- Transcode the video to MPEG-4 (or Dirac! [14]) and watch TV on devices that can handle video, but not in full TV quality. Your mobile phone, iPod or PlayStation Portable, perhaps.
- Automatically create your own edited highlights - search through the programme subtitles for keywords (eg "pony", "goal", "kiss") and cut out the thirty seconds of video surrounding it. Concatenate the chunks of video into a new, shorter programme.
- Create a "scrapbook" programme by extending the previous idea to a whole week's TV viewing - you could watch every goal in every football match shown on television that week, for example.

5.3 Other Ideas

- Extract recipes from the subtitles of cookery programmes and format them for printing.
- Run beat-detection algorithms against soundtracks. Where you detect a beat, there's probably music. Extract it and do something cool with it.
- Run a cut-detector [21] on TV programmes. Determine the rate at which cuts are made. The faster the cut-rate, the more "pacy" the programme. Add "paciness" information to your EPG and watch TV that suits your mood.
- Run stress analysis software against interviews with politicians. See how uncomfortable they are with the questions.
- Finally, there is currently no open-source implementation of an interpreter for MHEG [22][23], the language in which interactive content is distributed on Digital Terrestrial Television in the UK (and other countries). This is a significant component missing from all Linux PVRs and would allow interactive content to be repurposed in similar ways. Creating one would be require a large open source project rather than a quick hack, but it would be the most useful of these suggestions to implement by far.

6 Conclusions

The availability of cheap hardware and free software make it practical to take a Digital TV stream and do whatever you like with it. There are still desirable tools that have yet to be written: a complete set of stream-processing tools would be very useful, for example, as would a free MHEG interpreter. A number of new and interesting things to do with Digital TV have been described: the authors [24][25][26] look forward to receiving any feedback you may have.

References

[1] O'Reilly Hacks. `http://hacks.oreilly.com`.

[2] The Digital Video Broadcasting Project. `http://www.dvb.org`.

[3] The LinuxTV Project. `http://www.linuxtv.org`.

[4] DVB and ATSC on the Mac. `http://www.defyne.org/dvb/`.

[5] Media Portal. `http://mediaportal.sourceforge.net`.

[6] Topfield TF5800PVR Product Information page. `http://www.topfield.co.kr/product_e/pr_feature.asp?cb=DTR\&cm=PVR\&pn=TF5800PVR`.

[7] Dream-Multimedia-TV GmbH. `http://www.dream-multimedia-tv.de`.

[8] The DreamBox Product Family. `http://www.dream-multimedia-tv.de/Bereiche/Produkte/index.php`.

[9] Project X - DVB demux Tool. `http://sourceforge.net/projects/project-x`.

[10] The GNU General Public License. `http://www.gnu.org/copyleft/gpl.html`.

[11] Video Lan Client. `http://www.videolan.org/vlc`.

[12] The DivX MPEG-4 Codec. `http://www.divx.com`.

[13] The XviD MPEG-4 Codec. `http://www.xvid.org`.

[14] The Dirac Video Codec. `http://dirac.sourceforge.net`.

[15] DVB tools. `http://sourceforge.net/projects/dvbtools/`.

[16] PVAStrumento. `http://www.offeryn.de/dv.htm`.

[17] PieSpy. `http://www.jibble.org/piespy`.

[18] GOCR. `http://jocr.sourceforge.net` (sic).

[19] Open Tech 2005. `http://www.ukuug.org/events/opentech2005/`.

[20] SubStream. `http://sourceforge.net/projects/substream`.

[21] BBC R&D Video Shot Change Detector. `http://www.bbc.co.uk/opensource/projects/shot_change/`.

[22] MHEG-5. `http://en.wikipedia.org/wiki/MHEG-5`.

[23] “MHEG-5 UK Profile, version 1.06”. `http://www.dtg.org.uk/publications/books/mheg_profile_106.pdf`.

[24] Steve Jolly. `mailto:stephen.jolly@rd.bbc.co.uk`.

[25] Gareth Smith. `mailto:gareth.smith@rd.bbc.co.uk`.

[26] Alia Sheikh. `mailto:alia.sheikh@rd.bbc.co.uk`.