DIGITAL TELEVISION MASHUPS
—
INTERNET USAGE PARADIGMS FOR TV VIEWERS

by

Henning Pohl

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF SCIENCE

AT

UNIVERSITY OF BREMEN
BREMEN, GERMANY
JANUARY, 2007

Contents

License

Abstract

The television world and the computer world are converging. We can now browse the internet on our television screens, access all our media from the TV, watch television on our PCs or have television shows available via video on demand. Technical aspects of that convergence have been addressed exhaustively but the usage aspects have not. Initially, I asked myself how our usage of the internet would change the way we use other media. In this report I would like to concentrate on the idea that as television and computers converge the usage paradigms of both worlds will ultimately converge as well. I will provide examples of what such a convergence might look like.

For that purpose I will give an overview of the current state of the TV / PC landscape and evaluate how it addresses the possibilities of the convergence mentioned before. I will outline the main relevant features of the internet and digital television and how they could influence the potential converged experience. I will have a look at the software solutions in place at the moment and evaluate their suitability for usage convergence scenarios. Finally, I will present new concepts for of usage convergence. While the examples are conceptual in nature I will also present a prototype implementation and outline its technical aspects.

German Abstract

Internet und Fernsehen kommen sich immer näher. Am Fernseher kann man im Internet surfen oder alle seine digitalen Medien aus dem Heimnetzwerk abrufen. Auch am Computer kann ferngesehen oder können Videos, on-demand, aus dem Internet geladen werden. Während die technischen Aspekte der Konvergenz dieser beiden Welten schon ausführlich behandelt wurden, sind noch viele Fragen bei der Konvergenz der Nutzungsmuster offen. Angefangen habe ich bei der Frage, wie die Nutzung des Internets unseren Umgang mit anderen Medien verändert. In diesem Report gehe ich von der Annahme auss, dass Internet und Fernsehen konvergieren und damit auch eine Konvergenz der Nutzungsmuster eintreten wird. Ich werde im weiteren Verlauf Beispiele dafür vorstellen, wie solche Konvergenz aussehen könnte.

Dazu wird dieser Report zunächst eine Übersicht über den aktuellen Stand der Fernseh- und Computerwelt in diesem Bereich bieten und im Besonderen darauf eingehen, inwiefern dieser Stand erweiterte Nutzungsmuster bereits ermöglicht. Relevante Eigenschaften des Internets und des digitalen Fernsehens werden erläutert und ihr Einfluss auf eine konvergierte Nutzung evaluiert. Der Report soll auch eine Bestandsaufnahme der aktuellen Softwarelandschaft in diesem Bereich sein und untersuchen wie sehr bestehende Softwarelösungen schon auf sich ändernde Nutzungsmuster eingehen. Schlussendlich werden mehrere neue Konzepte möglicher Konvergenz vorgestellt. Diese sind zu großen Teilen konzeptueller Natur, wurden aber auch teilweise als Prototyp umgesetzt.

Acknowledgements

First of all I would like to thank Frieder Nake for continuous inspiration since I first met him in my second semester at the computer graphics course he was offering. His talks on computer science and digital media have been the most important ones in giving me an understanding of what this program I'm studying in is all about. His view on what and how we learn at our university always got me thinking and proved very helpful in developing my own understanding of studying. I would especially like like to thank him for the guidance while homing in on a more refined report topic.
Second I would like to thank Rainer Malaka for being the second reader of this report and for providing valuable feedback during my work on it.
Thanks as well to my family who supported me financially and morally during my whole time at this university and without whom all of this would not have been possible.
I would also like to thank Mareike Magers for proof reading this report and pointing out possible refinements in my writing.
Last but not least I would like to thank my friends for their continuous support and encouragement.

Ehrenwörtliche Erklärung

Hiermit erkläre ich, daß ich die vorliegende Arbeit selbständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Schriften entnommen wurden, sind als solche kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen.

Leiden, NL, den 25. Januar 2007 ______________________________

Henning Pohl

Chapter 1

Introduction

"Digital Television Mashups — Internet Usage Paradigms for TV Viewers" — I would like to explain this title a little bit. Apparently this report is about *digital television* and the *internet*. More specifically I mention *mashups* in the title. So it deals with a specific subset of the internet already. For example it is not about email or FTP transfers. It deals with the social and media aspects of the internet. Something that lately has been labeled "Web 2.0" (a term that by the way was coined to a huge extend by Tim O'Reilly[1]).

I furthermore mention *Usage Paradigms* that come from the internet and are somehow applied to *TV viewers*. This is the specific part I would like to concentrate on in this report which is largely dealing with usage convergence. I would argue that TV usage paradigms have already manifested on the internet but not vice-versa.

I'm using the word paradigm in the title and I think that because of its inflationary use that demands a small explanation. The Merriam-Webster online dictionary states that a paradigm is:

> "a philosophical and theoretical framework of a scientific school or discipline within which theories, laws, and generalizations and the experiments performed in support of them are formulated"[2].

That sets a pretty high bar, so what makes my think that my report is about *usage paradigms* and not just *usage patterns*? The decision was primarily based on my notion that the word pattern only partially describes what I'm writing about. The word pattern suggests that there is predictability and regularity in the matter described. However that does not apply here. I'm dealing with usage behaviors — There is no ultimate certainty there. I largely look at the bigger picture and try to deduct general trends and interests from that. I think this more closely resembles

[1] See e.g. `http://www.oreilly.de/artikel/web20.html` ("What is the Web 2.0?") for his views

[2] `http://www.m-w.com/dictionary/paradigm` (visited 16th November 2006)

Thomas Kuhn is idea of a paradigm than it does the meaning of the word pattern.

1.1 The Problem

But let us get back to television and the internet. At the moment television is in the middle of a transition from analog to digital broadcasting. This transition arguably will bring the computer world and the television world closer together thus creating some kind of hybrid medium as Vivi Theodoropoulou notes:

> "DTV is a new medium, which brings together various old and new media through a joined delivery mechanism. It could be seen as a 'hybrid' medium that has two components: that of television and programming and that of interactivity or in different words, TV as TV and TV as a computer."[Theodoropoulou, 2003]

While both have been distinct markets for most of the time we now see devices that are both: television receivers and computers. Sometimes more of the first, sometimes more of the later. When they move closer together they will start to influence each other to a larger and larger extend. Digitalization is the leveler that has it is root in the computer and will transform the television realm.
As mentioned in the abstract, I would like to explore the usage aspects of this convergence. I think that the transition to digital television has not yet been viewed under that aspect exhaustively enough. Both worlds have their distinct paradigms and it is not clear which will prevail in a converged media home.

1.1.1 Convergence

The word convergence is very prominent in this report and I would like to use the chance to elaborate it a little bit early on. Convergence can be seen under a lot of different aspects. It is used to describe the expected benefits of mergers and acquisitions, in advertising for cross-promotion or in the field of digital television (which is the one I will be dealing with in this report). Convergence is a word often used for an abstract vision; a word that is called buzzword by some. Thus the notion of convergence varies widely. As an example when Microsoft and NBC joined together, their idea of convergence was to "marry TV and the Internet"

[McPherson, 2003, p.176]. While that cooperation might be primarily business motivated Tara McPherson argues that:

> "[. . .] the union of the computer and the television is not simply about the **marriage of computing and televisual technologies**, that is, the marriage of machines. Rather, this union brings together computing technology with the American broadcast system [. . .]: commercial, unidirectional broadcast. Thus, "convergence"[. . .] already presupposes the commercial nature of the Internet as a medium." [McPherson, 2003, p.177] (emphasis by author of this report)

McPherson here assumes that it will be companies with their legitimate commercial interests which will be pushing the convergence of TV and the internet forward. In Europe there are also the public TV stations that are important players in shaping the convergence and which do not have to lend to financial reasoning only. Convergence on the hardware side however is clearly driven by commercial interests of home entertainment and computer manufacturers.

While the way the internet and TV converge is primarily business driven I would like to mainly draw on a different type of convergence in this report. Not only are two businesses converging but there are two distinct cultures converging as well. The TV culture and the television culture. While a business convergence might come first the culture convergence is following hard on. This is what I would like to focus on. This convergence will take place after the initial technical convergence. Therefore the technical convergence will set the rules of the game and define the scope of that second convergence. For that reason an explanation of the technical convergence is necessary before an attempt can be made to guess what the culture convergence might look like. Vivi Theodoropoulou noted in 2002 that:

> "[. . .] technological convergence between the PC and the TV will happen but the two media will for the foreseeable future remain two separate devices depending on their uses and crucially the context of use. As an interviewee stated 'TV is there to be watched. It is the queen of the living room. That's what it's there for. To be stared at. If I want to email I will go to my room and log on the Internet'."[Theodoropoulou, 2003]

I think while she is right about the technical convergence I do not share her doubts on media convergence. While we may not see email on the TV I'm sure we will at least see computer inspired applications on the TV. Theodoropoulou herself notes that "despite marketing predictions and the industry hype on interactivity, the 'converged viewer' has not yet been shaped."[Theodoropoulou, 2003] acknowledging that there can not yet be certainty on those consumer's demands. It is pointless to restrict the possible converged media experience to the sum of the media experiences it emerged from. It will be interesting to observe the changes that will most likely occur in this area in the future.

1.2 Approach

After this small introduction I will describe the inherent characteristics of the internet and the digital television world in more detail. Following I will present the details of the prototype I devised for the purpose of internet and TV convergence. I will also give an overview of the current software landscape and why its state is one of the reasons no far reaching usage convergence solutions have been implemented yet. Finally, I will present multiple examples of internet and TV usage convergence. I will detail the roots of each example in both fields and how each example facilitates said convergence.

Chapter 2

Digital Television

In this chapter I would like to provide some insight into digital television. I will first give an overview of the transition process and its stakeholders. After that I would like to present the features of digital TV with more detail. There are some distinct new features of digital television that make it different it from analog television. For this report I will focus on digital television over-the-air (DVB-T). Note that while most aspects also apply to other transport channels some aspects, like the error correction used, are specific to over-the-air transmission.

I will also give a small general overview on TV as a medium. This is certainly not intended to be a complete overview on the world of TV. I rather intend to shed some light especially on how TV is used at the moment and the implications of the medium.

2.1 The Digital Transition in Television

While I will explain the features of digital television later I would first like to give a short introduction on the transition process itself. This transition has already taken a long time and we are still right in the middle of it. For example the ATSC digital television standard was already adopted in the US by the FCC in December 1996 [Book, 2004, p. 8]. But the US is still in the middle of that transition. Over-the-air broadcasting of analog television is currently scheduled to end February 2009[1]. In Germany it is planned to have digital over-the-air broadcasting for at least 90% of the population by the end of 2008[2]. To achieve that mark digital broadcasting will first be available to people living in larger cities and the more densily populated areas of the country. Other countries all over the world are also in the process of this transition and not only terrestrial but also cable and satellite broadcasting are part of that transition.

[1] `http://www.dtv.gov/consumercorner.html#whencomplete` (visited 10th November 2006)

[2] `http://www.ueberallfernsehen.de/data/empfangsgebiete.pdf` (visited 10th November 2006)

There are already numerous benefits of switching to digital television. Ulrich Reimers for example provides a list of reasons for DVB in [Reimers, 2005, pp. 10–11]. However I think that, being still in the middle of the transition, a lot of changes are yet to take place. For example Interactive television and advanced television services are now feasible but are not yet widely adopted. Some standards for interactive television already exist but, like the overall development of digital television shows, it will probably take a couple more years for this technology to mature. Artur Lugmayr et al. describe the current state of DTV as

> "[. . .] an excellent example of being at the starting point of the S-Curve [See [Abraham and Knight, 2002]]. It is a new innovative branch with new large revenue potentials."[Lugmayr et al., 2004, p. 142]

I think that what we see today is only a glimpse on what is yet to come. I'm writing this to specifically underline that this is an area where far-reaching innovation can be expected to take place and it is one of the characteristics of a transition that within the process, the result is never fully predictable. As this report is written in the middle of that transition it aims to look at what has already happened and tries to deduct predictions for the future from there. I see this work as my own interpretation of where I think the transition should take us and what areas of the transition have not yet received their required focus.

2.1.1 Different Cultures

There are two driving forces behind the digital television transition. The computer industry and the television industry. One should keep in mind that those two players have different interests and the final design of digital television has to reflect this. Larry Press mentions several differences between those two cultures (he specifically is talking about the interactive TV culture which derived from the general TV culture) in [Press, 1993, pp. 22-23]. I have picked some from his list for this report to give a general idea of how these two worlds differ:

- Applications — TV's initial focus is on movies and home shopping while the internet culture started with a focus on the educational community and communication services.

- Users — TV is targeting the user at their home while the internet started as a utility for researchers and later spread into the businesses and homes.
- Data Type — TV is the premier medium for video while the internet was text based initially and can incorporate any digital media type.
- Terminal Device — TV is build for a living room experience while the internet was initially targeted at desktop computers.
- Geographic Scope — While TV is regional the internet is global.
- Access Ethic — TV uses a centralized producer & broadcaster model while on the internet every user can also assume the role of a producer and every computer can be become a broadcaster.

While the differences between those two cultures are not the main focus they should be kept in mind when reading this report. The design of digital television reflects the struggle between these different cultures. A design decision that might not seem ideal from e.g. a computer standpoint might make more sense when remembering the TV standpoint. In the long run, we will probably see a convergence between those two cultures. However, in 1992 Mountford et al. already noted that "The shift from broadcast television culture to computer culture is likely to be slow."[Mountford et al., 1992, p. 227]. From todays standpoint this still holds up and the convergence is sure to still take more time.

2.1.2 The Digital Television Transition and the Computer World

As mentioned earlier the television world and the computer world are moving closer together. For a long time both worlds had their own standards. For example Television used interlaced scan for its images while computer monitors worked in progressive mode[3]. A television signal was very different from a video signal in the computer world. When a user wanted to use television signals on a computer he had to utilize an add-on card to bridge the gap between these two worlds. The analog and interlaced image had to be converted to a digital and progressive image. There were no information on the signal available in a given broadcast. The only additional service was teletext which is not easily

[3] See [Owen, 1999, pp. 291–293] for a short overview on the "Interlace Controversy"

consumable by computer programs but made for human viewers.
After the transition DTV has a lot in common with the computer world. The compression algorithm used is MPEG-2, which for example is also used for DVDs. Thus the video stream is easily consumable by a computer. Computers that are able to play DVDs are also able to decompress the video stream from a DTV signal. There also is metadata available on the channel a tuner is tuned to. A computer program could consume that data to enhance the viewing experience. I will show what the convergence of these two worlds could result in more in depth later on. At this point I would like to cite Lugmayr et al. who described the transition like this:

> "Recent years have brought many changes to the world of mass media. The Internet and mobile communications technology have provided consumers with interactive digital services. Television is catching up with this trend through the digitalization process. Digital television and the Internet, providing modern multimedia services on a familiar platform. In short, digital TV is a gateway to the world of interactive media." [Lugmayr et al., 2004, p. VII]

They focus on the interactive capabilities but also see television converging with services provided by the computer world. They are mainly talking about the internet but also include mobile devices. The core idea is that of multimedia services. Their idea of a multimedia service is not coupled to only one platform but is transient to multiple platforms and could therefore also be consumed in different ways. In this report I would also like to build on that definition. As I mentioned earlier I think that the attention usage convergence attracts is only marginal at this point and it is time to bring together the TV and the internet world even more. One approach for that would be to create service mappings for services from one world to the other one. Together those "converted" services would form such an above mentioned multimedia service. They might not be identical (they most likely can not be as they must at the end be tailored to a specific platform) but they would share a common idea and enable users to map their service usage from one world to another.
So what kind of services could be transient between both worlds? Lugmayr et al. organized possible DTV services into groups. See Table 2.1 for a complete

Service Type	Description
PC Migrated Services	
communication	E-Mail and newsgroups, instant messaging, chat, teleconferencing
collaboration	gaming, content synchronized chats
Standard Digital TV Services	
informational	TV portal, EPGs
regionalized	regional weather, regional news, merchandising
personalized	custom news, advertisements, personalized EPGs, automatic video recording
interactive	game shows, interactive advertisements, eKnowledge platform, knowledge visualization, transactional services, access point to the digital smart home, user authentication (smart cards, conditional access, fingerprints, voice identification), payment schemes
Visionary Services	
perceptional	active content and content manipulation, perceptive digital items, adaptive content, personalized characters and actors
collaborative	digital communities, contextual computer games, TV as an artificial companion
narrative	interactive narratives, knowledge socialization
intelligent	smart TV, observation of user habits, user group identification

Table 2.1: Digital TV services, taken from [Lugmayr et al., 2004, p.138]

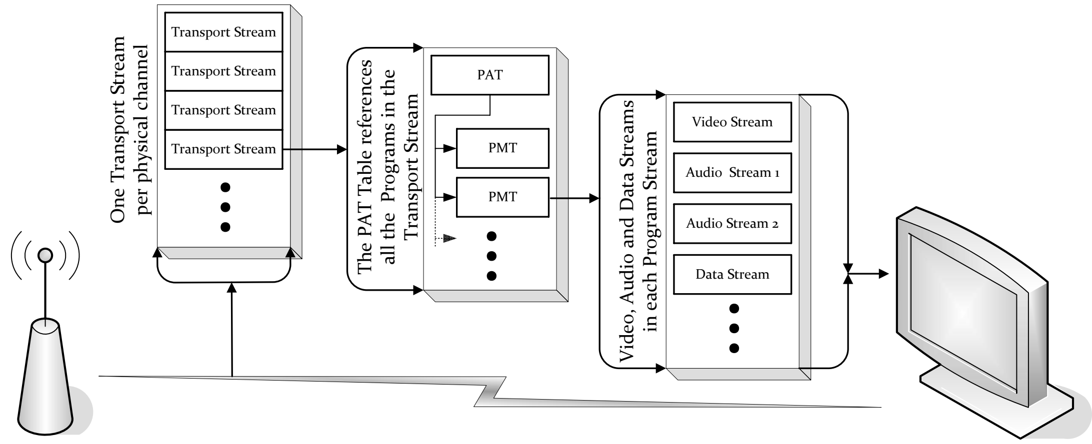

Figure 2.1: Structure of a DVB-T broadcasting signal

listing taken from [Lugmayr et al., 2004, p.138]. This provides a great general starting point and I will extend this later on when I present my ideas for possible convergence services.

2.2 Technical Features of Digital Television

Digital television has some features that have merited the transition from analog to digital television. While a complete guide to the technical aspects of digital television can to found [Fischer, 2004] or [Reimers, 2005] I would just like to briefly describe the most important differences. For the general structure of a DVB-T broadcast signal please see figure 2.1.

Multicasting

In analog television only one TV program is broadcasted per channel. For example in terrestrial television a channel is usually a 6,7 or 8 MHz part of the radio spectrum. If TV signals are digitized they can be compressed. Thus up to 4 standard programs can be fitted into one channel. A HD program however takes up more space and almost takes up an entire channel. In Germany it was decided not to broadcast HD content over-the-air. A higher number of available channels was deemed more important than HD programming.
By multicasting the available spectrum can be utilized more efficiently. This is important as television has to share the available spectrum with cellphone

networks, broadcast radio and many other participants. Broadcasters can not just obtain more frequencies but have to utilize their assigned spectrum more efficiently. Thus multicasting is a major incentive for companies to switch to digital broadcasting. For consumers multicasting is a major incentive for switching to digital television. Surveys show that the biggest reason for switching for a lot of consumers has been "the bigger choice of channels it allows."[Theodoropoulou, 2003].

Error Resistance

The broadcasted digital signal contains error corrections codes. Receivers are therefore able to correct erroneous input. Digital receivers also handle situations gracefully where there is more than one source (for example when multiple broadcasting towers are in range). This is thanks to the COFDM (Coded Orthogonal Frequency Division Multiplex) modulation technique (See e.g. [Fischer, 2004, pp. 312–313]). Where analog receivers would produce artifacts such as shadowing digital receivers are able to maintain picture quality. On the downside analog signals degrade gracefully while digital signals look good up to a certain threshold and then immediately look bad. For example in DVB systems every MPEG-2 packet has 6 bytes of error protection and up to 8 errors per packet can be repaired using that information [Fischer, 2004, p. 19] but if that error threshold is exceeded a packet has to be discarded.

Video Quality

The digital television signal will support better picture quality. A full frame in analog television has either 625 or 525 lines. In Germany, a 625 line system is in place (note that a frame is interlaced and the odd and even fields are broadcasted successively so the 625 lines are divided between those two). From those 625 lines only 575 are visible as some lines are reserved for the vertical blanking interval (VBI). This 575 lines a the maximal vertical resolution of picture transmitted on analog television. There is no definite relation to pixel values for analog television pictures though. The horizontal resolution will depend on the sampling frequency and is therefore arbitrary but if we assume a $4:3$ aspect ratio and square pixels the horizontal resolution of an analog TV frame would be 768 pixels.

Digital television introduces a range of new formats. Now the resolution of a frame is clearly defined contrary to analog television. Numerous formats exist for digital television. In one system[4] they are: 240p, 288p, 480i, 480p, 576i, 576p, 720p, 1080i & 1080p. Formats ending with an "i" are interlaced while formats ending with a "p" are progressive. The number defines the amount of lines broadcasted. In the interlaced formats the amount of lines is divided between the two fields. Therefore a picture transmitted in 480i only has 240 lines and only together with the next field the 480 lines total resolution are achieved. All the formats mentioned before exist in different horizontal resolutions. There are versions for $4:3$ and $16:9$ television and also for rectangular and square pixels. In digital television the 576i format corresponds to the standard format in German analog television. All formats with a higher number denote formats with higher resolution.

However, in Germany only a marginally better picture will be broadcasted. In general digital transmission of television signals is not a requirement for higher resolution video. HD television would also be possible using analog technology (in fact when HD development started it was not about digital at all). Digitalization however allows for compression of a high throughput HD signal. Therefore a digital HD broadcast is more feasible than an analog one.

An important aspect here is the inclusion of progressive formats in the digital television standards. While TV traditionally has used interlace formats and the PC world has used progressive formats, progressive formats for television bring those two worlds closer together. Processing of progressive formats in a PC is easier and better suited for the PC architecture.

Compression

Digital television is compressed using the MPEG-2 suite of compression algorithms. This compression it what makes multicasting possible in the first place. This also enables broadcasting of higher quality imagery in a given bandwidth which could be a possible incentive for people to quickly switch to digital television. The compression algorithms chosen for DTV are also used e.g. on DVDs

[4]Note that there are numerous ways to label those resolutions. The one given here is mostly used in the USA. Another system would be the MPEG-2 profiles and levels.

or in camcoders. Most computers are already equipped with software to decode that kind of video and audio.

Dynamic Program Structure

A digital television stream can be quite complex. A tuner receives a list of all available programs in the program association table (PAT) in the transport stream (TS). Every program is listed there with an identifier. The tuner then has to wait for the program map table (PMT) with that identifier to get more information on a program. In that second table the identifiers of the video, audio or data streams of that program are available. A program can hold any possible combination of video, audio and data streams. This could be a single video stream and three audio streams, a data and an audio stream or only a data stream. The interesting point here is that the structure of the programs is completely flexible. At every point in time a broadcaster could add a new program, add a data stream to a program or remove an additional audio stream from a program. In theory that offers broadcasters a lot of flexibility that was not available before in analog television. In practice however most receivers do not continually check for the program mapping and a full scan takes quite some time [Fischer, 2004, p. 23]. Therefore, this functionality is not commonly used at the moment.

Encrypted TV Channels

In digital television programs can be encrypted. In that case there is an additional table in the TS with information on the encryption - the conditional access table (CAT). It references the packets that hold the descrambling information. The decryption is done outside of the receiver. For that purpose a common interface is defined and users have to plug in smart cards or other hardware for the descrambling. The receiver will then route the TS through the common interface where the encrypted data is descambled and passed back to the receiver for further processing. The encryption system in that way enables new possibilities. First of all, encryption of programs is a very important part of Pay-TV. Pay-TV in analog television scenarios was limited to some basic tricks on the television signal like not broadcasting sync information. In the digital world far more sophisticated encryption is possible.

Encryption is also used for copyright reasons. If for example a TV station in the Netherlands buys the rights for an American TV series the price for that series will depend on the amount of possible viewers. It will be broadcasted in English so not only Dutch people could watch the series but most of Europe if the show is broadcasted over satellite. If the channel is encrypted and only Dutch people have access to the descrambling smart cards the amount of possible viewers and therefore the price of the series for the TV station will decrease.
Digital television enables business models that have not been feasible before in analog TV. As encryption is performed on the TS level, and could therefore also be used for data services, we might also see new data services for interactive television that build on a pay-per-use or similar model.

Additional Data

In digital television the TS does not only carry the video and audio data but also contains other data. This includes the old fashioned teletext for programs in the TS and metadata on the TS itself. In section 2.2 I already mentioned the PAT and the PMT which carry information on the programs available and what kind of data constitutes the program. In addition to those the DVB Group defined some other tables, labeled service information (SI) tables. Those tables provide more general and not program specific information. The available tables are:

- Bouquet association table (BAT)
- Event information table (EIT)
- Network information table (NIT)
- Running status table (RST)
- Service descriptor table (SDT)
- Time & date table (TDT)
- Time offset table (TOT)
- Stuffing table (ST)

For example the SDT describes the services provided in a TS. For example this includes the human readable name of the program but also encompasses a lot of other descriptors[5]. This also includes the electronic program guide (EPG) which is distributed in the EIT table. That table offers means to e.g. broadcast the program guide for the next week with the TV program. It is however mostly limited by broadcasters to the currently selected TS. Therefore only the TV program from programs in the current TS can be accessed. While according to the standard it is possible to include guide data for other TS most broadcasters chose not to do so. All the channels in one TS are usually related (like all channels from the RTL group or all channels from Pro Sieben / Sat.1) and broadcasters are not obliged to provide guide data for their competitors channels. In theory however a receiver can read a lot of information out of the incoming TV signal. Extensive data is not only available on the current program but also on the upcoming program. It is up to the TV software or set-top-box to present that data in a meaningful way.

Additionally, broadcasters are free to include any data they want in a program of their own in the form of a data stream. However there is no standardized data format for that data. Therefore only a few very specific applications exist which rely on matching player software to make use of the embedded data. Data can be broadcasted as part of a program stream mux or as an individual data stream in the TS.

Data broadcasting in DVB is handled by data and object carousels[6]. Data carousels can only carry individual files while object carousels are more flexible and could contain directories or events referencing other streams in the TS. In both carousels data is constantly and repeatedly broadcasted. Like the tables in the TS there are tables in the carousels that are used to identify individual files or objects. Via those tables receivers can find out about the kind of data being broadcasted and can fetch individual files or objects given their IDs.

[5]See EN 300 468 standard “Specification for Service Information (SI) in DVB systems” from the DVB Group p. 28 for details on available descriptors

[6]For a complete overview on data and object carousels please see [Reimers, 2005, pp. 289–295]

2.3 Media Aspects of Television

"Television is a unidirectional and visual mass medium" would be a short description of television. I would like to use this section to write about those aspects of television. This is of course not intended to be a full cultural analysis of television. Instead I would like to only turn to some key areas of television. For a short history of television Jochen Hörisch's book "Der Sinn und die Sinne. Eine Geschichte der Medien [Hörisch, 2001, pp. 339–359] also is a good starting point.

Television and couch-potatos are two words that often go together. Television viewers want to "turn their brain off" (according to Steve Jobs via [Boddy, 2004, p. 90]). The fact that content on television can easily be consumed is what brought along that reputation. However that easiness of consumption is also what makes the medium television accessible to almost everyone. Interaction with the TV set is limited and people do not need to learn much to start TV viewing. Once viewing their decisions are limited to what channel to watch and whether to continue watching. That is in stark contrast to the internet where a much higher level of expertise is needed to use the medium. Entry barriers are thus much lower in television than on the internet.

Another important aspect of television is it's liveness. Of course not everything on television is live but for some parts of TV (e.g. sports broadcasting) liveness is a central aspect. It is important to note that liveness on TV is different from liveness on the internet. The difference becomes clear if we e.g. have a look at the events on 9/11. On television all stations immediately switched to news reporting mode once the events started to occur. In the following hours all the TV viewers were remotely connected to New York City. They shared a common experience and it did not make a difference how many kilometers people were away from New York. Of course people also could follow the events on the internet. However here no liveness in the television sense could be created. People would browse around, look at pieces of information and continue. Everyone would become their own director, more busy with collecting information than just "experiencing" the event. Thomas Bjoerner states that: "It is well known how television has a special immediacy [snip], which is created in that moment, when 'the actual broadcast and the reception of it are synchronous' "[Bjoerner, 2003, p. 97].

TV can also be a group experience. While a family gathers in the living room to watch a TV series together they would not gather in the den to surf the internet together. TV viewing certainly does not require a group but it is an optional way of viewing. Viewing behavior therefore also needs to be differentiated into sole viewing behavior and group viewing behavior. Even if people watch TV alone they often later on share that experience with friends and colleagues and thus create a non-local TV viewing group. The structure of TV facilitates this as the fact that a show is broadcasted in a fixed timeslot ensures that people watch it at the same time. Hence the next morning in the office they all shared an experience the evening before even while they were not together. Nicholas Abercrombie notes on that subject that:

> "As I have implied already, one of the critical qualities of television seems to be its capacity to provoke conversation, to encourage talk. Indeed, television often seems to be about talk. [. . .] Television is intended to be received in a domestic context, which, as I have already said, is characterized by conversational interchanges." [Abercrombie, 1996, pp. 173–174]

2.4 Interactive Television

For this report the area of interactive television is especially important. Development in interactive television (ITV) has a lot of overlap with development of digital television in general. However the television industry only partly build on existing interaction solutions from the computer world but tried to create new solutions specifically for television. Looking at the research in that field we can draw conclusions for the mapping of internet interactivity concepts to the television.

But first of all I would like to come back to the term *interactive television*. This term is not clearly defined. Célia Quico for example states that:

> "**Interactive Television** can be defined as the result of the convergence between television and interactive technologies, allowing the user to customize the programming schedule, to access video-on-demand, pay-per-view, and the Internet, to send and receive e-mails,

> to play network games, to shop and even to make financial transactions, through a television set connected to a set-top box."[Quico, 2003, p.100]

In that vision a lot of different aspects play a role. ITV is supposed to break up the linear structure of television or allow internet functionality on the television set. Television in that vision becomes a personal computer. Other researchers oppose that position and Vivi Theodoropoulou for example makes the point that consumers "do not, however use their TV as a PC and have clear perceptions about what functions each medium is supposed to cover."[Theodoropoulou, 2003].
I do think that this discussion is to some extend pointless if we assume that the hardware is converging. At the moment the associations bound to TV and computers are very different. While the TV is associated with leisure time the computer is often seen as a machine targeted at the work environment. This perception varies between different groups. Ellen Seiter for example mentions the gender influenced view of computers and the largely varying amount of resentments they face[Seiter, 1999, pp. 121–130]. In my opinion this will change with computer interfaces specifically designed for a living room experience and the rise of a perception of the computer more as a medium and less as a machine. For early adopters the convergence of the hardware is even one of the primary incentives to buy into the new PC/TV realm:

> "[. . .] when considering the advantages of adopting the PCTV device, most owners indicate three primary functions: the ability to record programming off the air, **the ability to multitask and converge multimedia activities into one viewing station/environment** and the ability to receive high-definition pictures affordably." [Book, 2004, pp. 235–236] (emphasis by author of this report)

2.4.1 Goals and Adaption

The purpose of interactive television is not a clear one. Like I wrote above there is dispute regarding the specific problems interactive television is supposed to solve and the enhancing properties it is supposed to carry. Table 2.2 shows an example from Sheri Lamont how interactive television could enhance traditional television. Note that in contrast to Célia Quico vision of ITV given above, interactivity here

Sports Broadcast (hockey)	
Reason People Watch TV	**Interactive Content**
- Viewers love the hockey team, their logos, and everything about them. They want to see more	- Provide extra stats about the sporting league, and/or each team playing.
- Viewers turn the game into a social setting by inviting friends and family over to watch.	- Provide extra score updates for other teams playing in the league, or show current standings
- Viewers love to watch the actual game play to see the speed, skills and playing technique	- Provide camera angle switching to watch the game in different views
Viewers want to see if the team beats the rival team – there is a sense of competition between the other teams and who their friends are cheering for	- Provide a pre-game chat so that fans and rivals of the two teams playing can express their emotion to each other
- Viewers love a particular player and follow their every move	- Have biographical information on athletes - Provide scoring statistics for the top scorers on each team

Table 2.2: “Reasons people watch television translated in interactive content”[Lamont, 2003, p. 135]

is an additional layer for standard TV and not a service on its own.

Let us take a step back from individual services and look at interactive television in a broader view, defining properties of interactive television. Constance Book distinguishes between dual screen and single screen interactivity (see [Book, 2004, p. 224]) including interactivity that takes place off the television screen. This is e.g. the case if viewers visit a website after they viewed a show and get additional content there. I would like to exclude that notion of ITV from this report as I would like to concentrate on ITV services on the same screen. Note however that this is an interesting aspect as much of TV's traditional *interactivity* takes place off screen between the viewers themselves (see section 2.3).

Starting with that kind of services we can split them in two groups: synchronous and asynchronous [7]. In section 2.3 I already explained the special immediacy of television. Interactive services can strengthen that temporal coherence by being synchronous and tailored to the current program like the ones proposed above by Sheri Lamont. Asynchronous services on the other hand are independent of the current programming and exist in a seperate interactive service space.

[7]Note that others like Vivi Theodoropoulou differentiate between ‘contextual’ and ‘non-contextual’ services (See [Theodoropoulou, 2003]) and mean the same thing

Thomas Bjoerner gives another example of such synchronous services (See [Bjoerner, 2003, p. 97]) and brings up a problem of that approach: viewers were disappointed that they could not access the additional content for a show after it had ended. The recipes for a cooking show were provided via an ITV solution and could only be accessible during the show itself.

I would like to use the above example to introduce another differentiator for interactive services: information immediacy. Information like the recipe is of interest not only during the show but also afterwards. This kind of information should be provided via an asynchronous service or outside of the ITV environment. A website or a book could prove to be the more suitable medium for this kind of information. On the other hand there is information that has a certain kind of immediacy. Current sports information adapting to the program like in the example above is one example for that kind of information.

The final differentiator of interactive services I would like to introduce is the overlay factor. Services can be designed to take up the whole screen and completely replace the television content or introduce only some smaller overlays. See table 2.3 by Sheri Lamont for an overview on the advantages and disadvantages of both options.

So what kind of service type is ideal for ITV applications? In my opinion **synchronous**, **immediate** and **non-intrusive** services are ideal for the enhancing of the television experience. I see television services as a way to enhance television and I do not think that the TV set should be an entry point to the world wide web — other devices can offer that functionality in a much better way. I do think that ITV services should also adhere to the immediateness of the medium and that information should be context sensitive either in regard to the current program or to the current viewing situation. Finally I think that ITV services should be an addition to the TV content and should not replace it. Therefore overlays are the way to go for TV enhancing services.

At the end interactivity for interactivity's sake is not what will attract customers. Vivi Theodoropoulou already stated that: "**Interactivity** was not a motive for participants in their decision to join a DTV company. Only 12.4% considered interactive services [. . .] as one among other influential incentives for subscription."[Theodoropoulou, 2003]. Interactivity that does not provide additional value to traditional television but tries to be of value on its own does not seem to

Design	Advantages	Disadvantages
Overlay	• Size of TV window is the same as regular TV • Content feels more integrated into show • Close proximity of TV and content may facilitate divided attention • Close proximity of TV and content may enhance memory for show	• Distracting because content on top of TV • Users try to look through content to see TV • Close proximity of TV and content may inhibit focused attention
Embedded	• Easy to separate TV and content when viewing • Easy to focus on content or watch show • Separation of content and TV may facilitate memory • Separation of content from TV may facilitate divided attention	• Size of TV window reduced • Viewer can more easily multi-task and direct attention in 2 locations • Importance may be given to content when not necessary • Separated elements make focused attention difficult

Table 2.3: "Advantages and Disadvantages to Using the Overlay and Embedded Designs"[Lamont, 2003, p. 135]

generate much interest. Customers are still largely drawn by the basic properties of the medium television.

One way to strengthen those basic properties would be to use ITV services to provide communication services for television viewers. Célia Quico already explored that area [Quico, 2003] and found out that in an ITV test scenario (aimed at sport broadcasting) communication services ranked among the favorite ones of the user base. She notes that while "The most accessed functionality was [. . .] an interactive video mosaic [. . .]. The second most utilized feature was the debate forum, followed in third place by the voting functionality."[Quico, 2003, p. 104] and especially underlines for the forum, that "An interesting aspect of the football match forum is that is a very lively debate, with new messages being posted every minute."[Quico, 2003, p.103].

All this should be kept in mind for a lot of design decisions of the prototype are based on this kind of assumptions. The example services implemented all adhere to the principles stated above. Services that enable communication between users are also part of this work and partly build on the findings of Quico mentioned above.

2.4.2 MHP

The Multimedia Home Platform (MHP) is the DVB Project's attempt at a solution for interactive television. MHP provides a common middleware solution for applications targeted at TV sets. MHP provides a set of APIs to applications and hides the details of the underlying hardware. MHP tries to unify the ITV landscape which has been divided among numerous different solutions (See e.g. [Reimers, 2005, pp. 332–337]).

MHP offers two ways for applications to be developed. The first option is to use Java and implement the `javax.tv.xlet.Xlet` interface. Those Xlets can then run on the Java virtual machine (VM) that MHP provides. This VM offers parts of the Java core functionality plus a set of APIs targeted at ITV development specifically. Xlets have a pretty wide range of tools at their disposal. They can control the tuner, control the graphics output, access the SI tables or handle user input.

The second option is to create applications in DVB-HTML which is based on standard XHTML. Applications are a set of pages and interaction is handled by

ECMA-Script[8]. Xlets can also be embedded in DVB-HTML pages for increased functionality. The MHP provides a browser-like application to display those DVB-HTML applications to the user.
MHP provides an easy way to build rich ITV applications. However there are some problems that did not make MHP a viable option for this report. First of all broadcasters have control on the applications available on set-top boxes and there is no way to put ones own applications on them. On PCs there are two free MHP runtime environments available[9] but neither of them offers a full implementation of the MHP standard. Additionally they are only a runtime environment for the applications themselves but not a complete TV solution. No TV tuner can be attached to the runtime and thus no application can interact with real television content. Thus for the development of a working prototype this platform was not a viable option.

[8]Scripting language by Ecma International that is the base of languages like JavaScript or JScript

[9]xleTView (http://xletview.sourceforge.net) and OpenMHP (http://www.openmhp.org)

Chapter 3

Internet

The internet is a fast-growing medium. According to the German "Forschungsgruppe Wahlen" (See [1] for data) 66% of all Germans older than 18 have access to the internet (3rd quarter 2006) while early in 1999 only 12% of all Germans had access to the internet. The internet also is becoming ubiquitous more & more. People have internet access at home, at work and even on-the-go you can go online via one of the numerous offers of the telecommunication companies. The internet has swept the media landscape by storm and for a lot of younger people has already become their primary medium. Sites like Myspace[2] or Facebook[3] have attracted a whole generation of high school and university students and sites like YouTube[4] build on the content created and shared by its users.
For this report I would like to describe the different forms of "Internet TV" currently available. I will also try to explain the idea of a mashup and how this could relate to changing TV viewing behavior.

3.1 Traditional Televisions adoption of the Internet

Before dealing with the internet's adoption of video services it is interesting to have a look at the adoption of the internet by traditional television and the internets benefits for traditional broadcasters.
When the internet developed it was soon welcomed by hardcore television fans. If you want to find out something about that TV show you liked in the 70s the Internet is the place to go. Myriads of fan sites are all around the World Wide Web. Ellen Seiter notes that:

> [...] television was one of the first topics people turned to when

[1] `http://www.fgw-online.de/Ergebnisse/Internet-Strukturdaten` (accessed on November 13th 2006)

[2] `http://www.myspace.com`

[3] `http://www.facebook.com`

[4] `http://www.youtube.com`

> trying to think of something to interest a large and anonymous group of potential readers — other Internet users. [Seiter, 1999, p. 117]

The television presence on the internet does not only consist of fan sites though. Every TV show has at least one official web site on the Internet and no TV network could afford not being online. Companies with a television background form joint ventures with computer companies or they even merge. The most famous example here is certainly the merger of Time Warner with AOL in 2001. Another example would be MSNBC, a joint venture between Microsoft and NBC (see also section 1.1.1). The websites of both MSNBC and Time Warner's CNN form an important point in the company strategy. For example a lot of CNN's content is available online on an on-demand basis and there is even a subscription service dubbed CNN (Pipeline) that offers live video feeds.
In general television content is often available online as well. Since 2005 consumers can purchase TV shows via the iTunes Music Store[5]. In 2006 ABC (one of the television network that sells shows via the iTunes Music Store) went one step further and started offering its TV shows via an online application[6] as well. Internet users now can watch the latest episodes of the offered shows online one day after they have been broadcasted. Deutsche Welle[7] goes one mere step further and also broadcasts their television program on the internet simultaneous to the standard TV broadcasting.
The internet also changed how television stations can get feedback on their shows. Almost every show's website features a message board. People for example already post on the "Desperate Housewives" message board during the show, discussing the twists in the story and what they like and dislike. At May 21st 2006 during the season's finale one of the characters, Mike, was hit by a car. Immediately following that incident at 07:56 PM the user "Xteam_meredithX" started a new thread[8] in the official Desperate Housewives forum and wrote:

> "MIKE JUST GOT HIT?!?!??! UGHHHH WHAT'S GOING ON???"

Apparently this viewer has been watching television and browsing the web simultaneously. Also he/she does not seem to be the only one engaging in this

[5] `http://www.apple.com/itunes/store`
[6] `http://dynamic.abc.go.com/streaming`
[7] `http://www.dw-world.de`
[8] `http://forums.go.com/abc/primetime/desperatehousewives/thread?threadID=1089146` (Accessed on December 13th 2006)

behavior as several other users immediately responded to her message. Like I already mentioned in section 2.3 people seem to enjoy communicating about what they watch on television. The Internet is just anther mean to get involved in that kind of communication. However, the internet makes this communication processes visible and available to outside observers more clearly.

3.2 Television-Like Services on the Internet

Delivering video over the internet is becoming more and more an option with increasing bandwidth available to consumers and the market penetration of broadband offerings. In the "Deutschland Online 3" survey[9] the average bandwidth of a broadband connection is estimated to go up to 11.5 MBit/s in 2010 and to 30.3 MBit/s in 2015 from 1.0 MBit/s in 2004. When asked for the influence of broadband applications on broadband adaptation, 66.3% of the people interviewed for that survey mentioned video on demand and 57.0% mentioned TV over DSL as having a big or very big impact on it. Experts estimate that by 2015 23.5% of the German broadband households will use video on demand services (see survey section 4.1).
Video on demand covers a broad range of services but TV on the internet is not limited to only video on demand. Therefore in the following sections I would like to give an overview of the available services. As a sidenote Bruce Owen already wrote about video on the web in 1999 and provides some examples in [Owen, 1999, pp. 311–325]. While this does not directly translate to todays situation it is still interesting to look back at the state of the development just a mere 7–8 years ago.

3.2.1 Vidcasts

Vidcasts are also known under the name vodcasts, netcasts or podcasts. All terms are a combination with the word broadcasting. Vidcasts play on the term video while vodcasts are more specifically referring to video on demand. Netcasts tries to underline the internet roots of the concept. The name podcasts stems from the name of Apples iPod. Basically all four denote the same concept. I will continue

[9] `http://www.studie-deutschland-online.de/do3/0000.html` (accessed on December 9th 2006)

to refer to vidcasts as I think the term if more generic than vodcasts and more precise than netcasts. The term podcasts should be avoided for several reasons. First of all it is ambiguous as it can refer to pure audio broadcasting and video broadcasting. It also specifically refers to one playback device while it does not matter at all whether you consume vidcasts on an iPod or any other device.

The basic idea of the vidcasting concept is that users can subscribe to internet TV channels made up of on-demand video. Channels here are somehow different from the channels known from traditional television. In vidcasting a channel is a loose group of individual videos with no necessary linear order. When clients subscribe to a channel they do so by subscribing to the corresponding RSS or Atom feed for that channel. In that feed each individual video belonging to that channel will be referenced by it's URI. Videos could be aggregated as files or as streams where the choice here depends on the desired use. When users subscribe to a channel their vidcasting client will present them with a list of videos in that channel. They may then choose a video from that list for viewing. Vidcasting clients will constantly check at the feeds server if a change has occurred in the feed. Maybe a new video has been added or an old one was removed. The list in the vidcasting client will then reflect that changes and the user can watch the newly available content.

Code 1 shows an extract from such a feed. There are multiple notable elements to this. First of all one can see that the feed is basically a standard RSS feed but is extended with the "itunes" namespace (`xmlns:itunes=http://-www.itunes.com/DTDs/Podcast-1.0.dtd`"). It should be noted that Apple does not provide a DTD at the specified location. However most of the tags introduced are pretty self-explanatory. Following the namespace inclusion are general channel information. Line 5 to 18 provide data like the title, author or category of the channel. Following that general information comes any number of items. Each item in a vidcast refers to a video file or stream. In our example the video is referenced in the `enclosure` tag. There is similar metadata available on each item as for the whole channel, but there are also some additional fields like the duration of the referenced media or the time the item was added.

Vidcasts are interesting as they translate the channel metaphor from television to the internet. The expectations here are high. Players like the "Democracy"

Code 1 Extract from exemplary vidcasting feed from VODcast (`http://www.vodcast.nl/feeds/jsdemo.xml`)

```
<?xml version="1.0" encoding="UTF-8"?>
<rss xmlns:itunes="http://www.itunes.com/DTDs/Podcast-1.0.dtd" version="2.0">

  <channel>
    <title>Jet Stream iTuner – VODcast technology demo</title>
    <itunes:author>Jet Stream BV</itunes:author>
    <link>http://www.vodcast.nl</link>
    <description>Jet Stream iTuner is a demo that shows [...]</description>
    <itunes:subtitle>There is more to iTunes Podcast features than you imagined</
        itunes:subtitle>
    <itunes:summary>Jet Stream iTuner is a demo [...]</itunes:summary>
    <language>en–us</language>
    <copyright>Jet Stream BV 2005</copyright>
        <itunes:owner>
            <itunes:name>Stef van der Ziel</itunes:name>
            <itunes:email>stef@jet–stream.nl</itunes:email>
        </itunes:owner>
    <category>Technology</category>
        <itunes:category text="Technology"></itunes:category>
```

...

```
     <item>
      <title>iTuner concept at VODcast Forum</title>
            <itunes:author>Jet Stream BV</itunes:author>
      <description>Bringing professional VODcasting to the living room.</description>
      <itunes:subtitle>Bringing professional VODcasting to the living room.</
          itunes:subtitle>
      <itunes:summary>English spoken. Stef van der Ziel, founder of Jet Stream BV
          [...] </itunes:summary>
      <enclosure url="rtsp://qt1.streamzilla.jet-stream.nl/stef/stef-vodcast.mov"
          length="1024" type="video/quicktime" />
      <guid>rtsp://qt1.streamzilla.jet–stream.nl/stef/stef–vodcast.mov</guid>
      <pubDate>Sat, 17 Sep 2005 12:43:00 GMT</pubDate>
      <itunes:explicit>no</itunes:explicit>
      <itunes:duration>23:04</itunes:duration>
      <itunes:keywords>MPEG–4, video, stream, VOD</itunes:keywords>
    </item>
```

...

```
</channel>

</rss>
```

player[10] claim that they would revolutionize television and encourage users to create their own online TV channels. For “Democracy” compatible channels[11] the RSS 2.0 format is extended with the Yahoo Media extensions to reference video files.

I will not delve into the details of programs like “Democracy” here but there are certain problems with the approach that I would like to address. First of all those applications make a strong claim. Specifically that they enable *“Watching TV”*. However, if we take a closer look at this we can see that it is clearly not TV what is watched. Every available channel is nothing but a collection of references to video files. There is a some linear structure but users are free to swerve. Videos have to be downloaded first and are then watched locally (using video streams instead of video files would solve that problem to some extend). At the end this “internet television” turns out to be a download service for videos from the internet. The properties of television addressed in section 2.3 can not be found in that kind of internet television.

One could argue that it is a matter of bandwidth whether real live video could be broadcasted to an audience and that once we could transfer that amount of video data watching video from the internet becomes like watching TV. However, for current users this model involves a lot of effort. For example the channel guide of “Democracy”[12] currently lists several hundred channels. Advanced recommendations and rating mechanisms need to be in place for users to be able to manage that amount of available content. TV suddenly is not a relaxing pastime anymore but something that requires a decent amount of management skills. This immediately becomes obvious when starting the application. While a television set or a television application on a PC right after startup present video content to the user “Democracy” feels more like a media player. Users have to select videos first and start them manually to view anything at all.

3.2.2 Video Sharing

Sites like YouTube attract a lot of users that want to share their videos with others and also view other people’s videos. In that way a network of videos is created

[10] `http://www.getdemocracy.com`

[11] See `http://www.getdemocracy.com/help/feeds.php` for documentation on the channel format (Accessed December 13th 2006)

[12] `https://channelguide.participatoryculture.org`

where users can comment on videos or respond to them with videos of their own. While those kind of services are an interesting area of research on their own they do not really lend to internet television. There are no real channel like structures to be found in the collection of videos. However those kind of sites could become valuable sources for others who would like to create TV channels on the internet. In theory a vast amount of broadcast material is available but quality and copyright problems are a pressing issue in that area.

An interesting aspect however is that those video sharing sites themselves make it back into traditional television. Clips from the internet are referenced in mainstream television media and the German TV group ProSiebenSat.1 actually announced[13] a whole show consisting only of videos from the video sharing site MyVideo.de[14] — a German YouTube clone. In that way internet content makes it back onto the TV screen and is refitted into a traditional show concept. The network properties of video sharing are lost and only the content is taken and placed into a new context.

3.2.3 Video on Demand

Video on demand is one of the services constantly being pushed by the industry. For example in Germany T-Online offers video on demand services[15] and in the US networks like ABC even provide video content for free (see section 3.1). Viewers who use services like video on demand or a personal video recorder (PVR) become independent from the broadcasting schedule. Gordon Bell and Jim Gemmell make this point for the PVR that applies to video on demand as well:

> "Users of PVRs such as TiVo and UltimateTV have indicated they almost always time-shift TV viewing, playing video from their personal cache at their convenience, rather than watching it at the scheduled time—so-called television "prime time" no longer exists for these users."[Bell and Gemmell, 2002, p. 74]

So is video on demand superior to broadcasting television? Does the internet way to consume media — click to access — replace the way of television where

[13] `http://www.dwdl.de/article/news_8633,00.html` (accessed on December 4th 2006)
[14] `http://www.myvideo.de`
[15] `http://vod.t-online.de`

people have to tune in at the right time to view something specific? For a lot of content asynchronous distribution might be the more convenient choice and Nicholas Negroponte already stated in 1995 that: "With the possible exception of sports and elections, technology suggests that TV and radio of the future will be delivered asynchronously" (via [Boddy, 2004, p. 103]). However I would like to refer back to section 2.3 again and point out that a lot of the aspects of traditional television get lost if everybody consumes on demand. Apart from questions of usage this also raises a range of other questions when the more or less free access to television is replaced with a pay-per-view system.

3.2.4 IPTV

In the general sense IPTV denotes television over IP networks. In practice, this term is mostly used for the triple play offers of the broadband service providers. They want to introduce IPTV as an alternative to television over-the-air, via satellite or cable. IPTV usually is offering at least traditional television and video on-demand services to end-users. Additional services depend on the specific distribution platform used.

One of the available middleware solutions for IPTV is Microsoft's "Microsoft TV"[16] product line. Additional services in this solution include a multimedia program guide, faster channel switching or multiple picture-in-picture support. IPTV in this setup can be received by Windows Media Center PCs or via a XBox game consoles or set-top boxes on the television screen.

IPTV does only partially fit into this section though. While IP networks are used for distribution IPTV is not available on the internet (in contrast to other video on-demand solutions like the above mentioned). Usually broadband service providers will offer this services at the last mile to their subscribers. However, this is an interesting field as the all-digital media pipeline and the more flexible nature of the transport method allow for a high level of customization of television. IPTV could be the focal point in a changing TV landscape. In Microsoft TV traditional broadcasting and video on-demand services already exist side by side thus opening up a new level in the competition for viewer's attention.

[16] See `http://www.microsoft.com/tv`

3.2.5 Peer to Peer TV

In contrast to the vidcasting concept, where the channels are meta-representations and consist of references to media files on the internet, peer to peer TV (P2PTV) applications usually stream full channels over the internet. The channel data is distributed via a peer to peer infrastructure.

At the moment this is an area with little proper information available. There are some applications out there but no major one. This is also a very shady area as most of the content is distributed illegally (and consists of actual TV channels redistributed by 3rd parties). It remains to be seen if peer to peer networks are a good way to distribute live video data. Currently severe restrictions in the protocols make that seem quite unlikely. An interesting aspect here is that the distributed video data is live and volatile instead of resource-based like in the above mentioned architectures which makes this approach probably the closest one to television.

3.3 Social Networks

"Social Networks" is a pretty general term used to denote a specific group of networks. In general a social network is a network that facilitates interaction between its users and provides a formal reflection of the social relationships between its members. Therefore such a network needs human users — pure computer to computer networks can not classify as social networks. However social networks can build on normal computer networks i.e. social networks are higher up the OSI model than normal networks.

The internet is currently the prime platform for virtual social networks. Social networks building on the capabilities of the internet can build on a huge amount of possible users that can easily gain access to the network. While most networks in real life require a certain kind of involvement, a social network on the internet usually only asks for a user's email address as an entry requirement. While huge parts of the internet are designed as such kind of network there are no social networks of that kind in the TV world.

Social networks in the virtual incarnation are a phenomenon from the internet that has not occured in any other media yet. While TV facilitates social interaction off-screen, network building is mostly confined to the real life and takes place

outside of the medium. There are a number of fan communities for television programming. Traditionally they would communicate by mail or fanzines but nowadays those communities turn to the internet as well. Thus the internet is providing a social context for television.
It would be an interesting endeavor to transfer those social networks into the television world. ITV applications would be the prime mean to bring social interaction into this medium. Television could learn about the pitfalls of such networks from the internet where a considerable amount of experience has already been accumulated. A number of books have already been written in this area with e.g. Howard Rheingold's "The Virtual Community: Homesteading on the Electronic Frontier" [Rheingold, 1993] being one of the standard works on virtual communities.

3.4 Mashups

The term mashup is generally used to describe the convergence of two or more internet services into one. For example one could combine rentals listings from Craigslist[17] with maps from Google[18] to present a map with all the available rentals on it to a user. Mashups mostly are build on open APIs to access the individual functionality of the mashup's parts. They leverage existing technology to easily create new applications which lowers the burdens on the individual developer.
Usually the parties involved in a mashup are uneven. The API provider in most cases is a bigger company while the mashup is created by smaller companies or individuals. This is most likely due to the fact that providing an API demands a certain amount of work to create it in the first place and to maintain it in the long run. Bigger companies also have the resources to offer services that can cope with a high amount of concurrent users. This becomes a vital requirement if the API is incorporated by many 3rd party developers.
What is the motivation for companies like Google to make such APIs available and enable outside developers to easily incorporate Google's technology into their applications? One reason could be that each mashup will increase brand and product awareness and recognition by the internet user base. If users can derive

[17] `http://www.craigslist.org`
[18] `http://maps.google.com`

great value from the above mentioned rental map mashup, the Google brand will also benefit from the user's appreciation. Publishing APIs can also be a way to become a de-facto standard in the field. If all map based mashups used Google maps users might be tempted to assume the best (maybe even only) map service on the internet is the one offered by Google while in fact there are numerous competitors. Finally, an open API can extend the value of a companies service by allowing users extended flexibility. As an example Flickr[19] (a photo sharing site) allows developers access to almost the full functionality of Flickr. Other companies were thus able to tightly integrate with Flickr. That makes it easy for Flickr users to print photos via a photo printing service or order whole printed photo books with their Flickr photos. They do not need to upload all their photos again but just allow another site to access their Flickr photos. Flickr does not need to provide all that functionality itself. But by making it easy for other companies to provide missing functionality, Flickr can integrate that missing features easily. Thus, they can increase the value of their product without investing into the development themselves.

So how could this apply to TV audiences? One of the prequisits is that TV functionality could be encapsulated in an API to incorporate it in a mashup. On Microsoft Windows using the Microsoft Internet Explorer an ActiveX object that displays television can be embedded into a website. This ActiveX object could then be controlled via JScript and mashed up with other internet services. As far as I know there are no attempts yet to mix those two media in that way. This is probably due to the fairly high complexity of the ActiveX object and the restrictions posed on the type of runtime environment.

Instead of embedding TV in internet applications it could be done the other way round and internet functionality could be embedded in TV applications. This is the way it is done in this report and which will be further highlighted later on.

[19] `http://www.flickr.com`

Chapter 4

Merging Internet and Television

In the last two chapters I tried to show some possibilities to map aspects of one medium to the other. In this chapter I would like to further explore the influence both media have on each other and the possible convergence.
S. Joy Mountford et al. already compiled a list of cross influencing factors in computing and television in 1992 (see [Mountford et al., 1992, p. 229]). I would like to start this chapter with some factors from that list that I found most interesting. First of all, they ask what computing adds to television and mention *democracy*, *flexibility*, *simulation* and *personalization* as defining factors. The term flexibility here is particularly interesting as they use it to describe that viewers, using new technological means, can react to programming by sending in video or audio responses. TV stations would design asynchronous programming around that possibility and viewer-involving formats would evolve. This is closely linked to the factor of democracy that builds on the idea that current communication technology has enabled e.g. voting functionality in shows and thus gives viewers a voice. In Mountford et al.'s view this will lead to programming that is more precisely catered to the respective audience. Their idea of personalization does not address personalization on the viewer's side. It rather focuses on new option for audience members to easily create high quality programming of their own, using computer technology, which in turn can become part of TV itself. In general computing is seen as a mean to break up the monolithic structure of television, tailor it more to the individual and open up TV for access from outside.
Mountford et al. then ask what television in turn could add to computing. In that part they underline the social aspects of television that have not yet found their way into the world of computers. The most prominent here being the "frame of reference" shared by all viewers *on* at a given time (see also section 2.3). As part of the social aspects they also mention TV's focus on content dealing with "moral issues". They contrast TV's storytelling nature that (being a medium centered around the vocal) has a strong emotional component to content on the computer

which they see as emotionally challenged.
As both media come closer each will likely lose some of its distinct properties. The internet has made computing more social while the personalization of TV with new niece channels all the time has weakened the "frame of reference" property of television.
But how exactly are TV and the computer going to influence each other in the future? Is the internet going to become even more important and eventually replace TV? According to Horst Opaschowski both media will continue gaining attention with neither of them replacing the other one (see [Opaschowski, 1999, p. 23]). Passive TV viewing and (more or less) active internet browsing are two activities that apparently go well together and can co-exist side by side.
We should be careful though when having a look at TV viewers that also use the internet. Younger TV viewers exhibit a different TV watching behavior than older TV users. While older generations used to watch TV exclusively younger audiences are multitasking and seldomly *only* watch TV[1]. The decisive point in the change from passive TV viewing to TV as one-in-many attraction probably was the invention of the remote control and its way into the households in the 1980s (See [Owen, 1999, pp. 102–103]). While it gave people more control over their TV experience it also made the whole experience less of a hassle where muting, turning off the TV or switched to a new channel posed no effort anymore.

4.1 PCTV User Profiles

I would quickly like to shed some light on the user profiles of the viewers we are dealing with here. In the section above I mentioned that younger viewers are both computer and TV savvy and are used to multi-tasking. Here I would like to further examine a specific group of viewers — those that watch television on their computers. Those consumers are especially interesting for this report as they have strong roots in both worlds and are especially interested in the convergence of both media.
In "Digital Television — DTV and the Consumer"[Book, 2004] Constance Ledoux Book mentiones a consumer study that I would like to draw upon in this section. All following numbers are taken from that source. Note that those figures were

[1]See e.g. [Opaschowski, 1999, pp. 26–31] or [Book, 2004, pp. 228–236]

assembled in the U.S. and therefore might not fit 100% to the German market. They do however give a general idea of the target audience of digital television. The main point of interest is their simultaneous media usage. More than 50% of the users who have a PC in the living room are online while they are watching TV. If we only look at the 12-24 year old this number goes up to as high as 80% [Book, 2004, p.220]. This user behavior seems to have become more common over time. Bruce Owen in his book "The Internet challenge to television" cites a survey from 1997 that states that "40 percent of all PC households watch television while using the computer" [Owen, 1999, p.206]. So this number seems to be increasing. This might be due to higher PC ownership in the population but also could hint towards a changed usage pattern. The highly different numbers with younger viewers point more towards a behavior dependent pattern.

If we look at the group of people who not only use a PC while watching television but watch TV on their PCs this group features several interesting characteristics [Book, 2004, p.229]. The group was almost exclusively male (97%), 73% were white, 41% earned more than $41.000 a year, 31% held a graduate degree. This numbers are common in early adopters' studies and show that early adopters are a pretty specific subgroup in society. One has to be careful though in examining their usage behavior as they usually are involved far more than standard users. However, early adopters allow a look at a possible more general future.

These consumers also had an average of 3.4 computers in their home which is much higher value than the average computer ownership of 1. All of them also had an internet connection and 76% even had a broadband connection. On average they were online for about 5.9 hours every day which is also significantly higher than the 1 hour national average.

They did not only spend a lot of time online, but also spent a significant amount of time in front of the TV. On average they watched TV for 2.7 hours a day on their standard television set and 2.6 hours on their computer. This amounts to more than 5 hours of television consume every day. Those people clearly do enjoy watching television.

So how do these people use their set-up? Book cites a consumer:

> "I wanted to be able to capture TV shows and watch TV while chatting with friends on MSN messenger, and I like computer gadgets."[Book, 2004, p.230]

25% of these consumers indicated they were chatting or writing emails while watching TV all or most of the time. 19% even said they were working while watching television [Book, 2004, p. 232].

We can see that this group of people exhibits some special characteristics that make them interesting for further research. In some respect their behavior can be seen as a glimpse of our own future television viewing habits. It is common that a small group of early adopters explores a new medium before the masses start using it themselves. While it is interesting to observe the usage patterns of that specific group it is important to keep in mind that most likely there is no direct mapping to the usage behavior of the general public. With a more diverse group of viewers adopting the technology we will also see other usage patterns. Therefore it is important to not cater services too much to the early adopters but think about what aspects could also apply to a more diverse group of viewers.

A very common behavior seems to be to chat while watching TV. This most likely also applies to a more diverse group as instant messaging or chatting in general already are media used by most internet users (especially younger ones). Chatting is a medium that does not require extensive knowledge and is therefore easily approachable for almost every user.

4.2 Television Usage Paradigms on the Internet

> “The success of video content on the Internet depends partly on the familiarity with TV”[Finke and Balfanz, 2004, p. 180]

What specific usage paradigms did the internet take from television? As the early internet was primarily text-based it did not have a lot in common with television. However the rise of broadband internet and the incorporation of different media into the internet has paved the way for TV paradigms. In section 3.2 I presented a range of TV-inspired services on the internet. I would like to use this section to deduct more general findings from that overview and present ideas how the internet could further build on television’s history.

When we take a look at video on the internet we can see that a lot of terms from television have made it to the internet. We see “Channels”, “Shows” and “Episodes”. It is no coincidence that e.g. online cartoons often try to build on traditional models from television. The popular show HappyTreeFriends[2] for example is targeted at the internet with short and quick to load clips but also tries to stay compatible with more traditional audio-visual media which made it easy for the shows creators to e.g. also publish the episodes in DVD format.

Other video sites are more like giant video databases and their primary function is to offer easy ways to browse and view the stored content. This kind of service has no equivalent in traditional television where the process of selection and presentation is done by the TV studios. In those cases we see a transfer of responsibility from traditional TV's producing side to the internet's consumer side. However further exploration of that phenomenon would require research in TV production which is not the focus of this report.

When it comes to the famous *zapping* behavior exhibited by TV viewers it is interesting to look at its representation on the internet. First of all, zapping is not the same as surfing though both have some similarities. When a internet user follows a link that action is not zapping. It is an action that was already anticipated by the creator of the site the user is currently viewing. Following the link, he is not making a completely free decision but he is acting in the constrains of the current page. Internet surfing becomes like zapping when users enter an URL directly or click on one of their bookmarks though. In those cases the user makes a decision independent from the current content provider to switch to a new one. I do not know of any studies examining similarities between TV's "channel surfers" and internet surfers though. It would however be an interesting area of research to find out similarities in their perception of choice and selection of action.

4.3 Internet Usage Paradigms for Television

Which usage paradigms from the internet could map to television? Mainstream television currently sees the internet as an information channel for their programming, as community builder and partly as a secondary market for their programming. Thus the internet is generally not seen as a medium to mix with traditional television. Attempts like the MyVideo show (see section 3.2.2) try to benefit from the boom in internet video sharing. But at the end TV only uses

[2] http://www.happytreefriends.com

the videos in the same way it already handles video for decades in shows like "America's funniest Home Videos". The community based nature of the internet gets lost during production of that kind of shows. Users who are or could be producers now become consumers again.
In the this section I would like to detail this issue of participation and others in more detail. It forms a basic foundation for chapter 6 where those ideas are used to develop new approaches for TV and internet convergence.

4.3.1 Participation Paradigms

Compared to television the internet is a comparatively open system. Everybody can contribute to what forms the world wide web. The level of contribution however varies a lot. Some might create a whole website of their own with custom design and content, rent a webserver and get a custom domain for their site. Others might be content with signing up with one of the bigger blogging providers and only write blog posts. The level of expertise needed is the biggest difference between those two offerings.
People do not need to be designers, video editors or journalists to publish content. People who like to take photographs but do not have the skills to create their own website can just upload their images to e.g. flickr which takes care of all the hosting, backup and technical details. They do not have to write their own comment system to receive comments on their photos.
Access on television is pretty limited. Most content is created by professionals and viewer contribution is mostly limited to things like their funniest home videos. Open access stations however already allow people to put their content on the air. The main obstacle here is that in those cases people have to do most of the work themselves. On the internet a huge amount of pre-build options exist to make their content look good while people are pretty much on their own on TV.
If television stations addressed this kind of problem they might make more use of the creative capabilities of a lot of people. Leveraging user-generated content for television could help in making TV a more democratic medium, giving people a feeling of involvement and offer cost benefits to TV stations.
Problems arising are ones of quality and of structure. While the internet can accommodate a huge amount of low-quality material the capacity of a TV channel is limited by time constrains. This brings us directly to the problem of structure.

The current structure of television with a relatively low number of channels might not be well suited for this kind of scenario. This is especially true for Germany where the station–network hierarchy found in the USA does not exist. Digital television with its increasing number of channels could provide means to alleviate this kind of problem. However, new methods of channel surfing would need to be in place to allow viewers to continue watching TV rather effortlessly.

4.3.2 Communication Paradigms

A lot of people use the internet primarily for communication. Email, newsgroups, chats or instant messaging have become the primary mode of communication for many people. Communicating via those channels has changed the way people express their thoughts. SMS instant messaging for example promotes shortening of messages and increased use of acronyms. We can see a reflection of this on TV where people can send SMS to e.g. a TV music channel and the messages are then displayed on screen. TV in that case has already adopted the shortened language of a new communication medium.

Other live interaction with television programming is often limited to votes where people are required to dial a certain number depending on their choice and call-in shows where people talk to the presenter via a phone. Sometimes email is used as well to send in questions for a studio guest or a similar purposes. I think the more interesting aspect here is communication of viewers with other viewers. The music channel example above is one example where people could use the medium of the TV channel to communicate with each other. However, problems arise with that kind of communication used on TV.

Television is a broadcasting system. Every viewer receives the same data. If the number of people who would like to communicate at the same time exceeds a certain threshold the channel will be too full. Therefore, embedding the communication in the standard broadcast is not a viable option. It also exposes people to a *conversation* they do not want to take part in and maybe are not interested in at all. So how could television address the issue of communication while learning from the internet?

One could argue that television does not need a system of its own at all. That traditional communication channels could fulfill this need and people could just

call each other on their landlines. However, if television would offer communication services of its own the services could be tailored more to a viewing audience and e.g. be context sensitive. The example to look at on the internet maybe is not email or chats but communities. Such networks (which I already wrote about in section 3.3) formalize social relationships and thus make it easy to restrict communication to subgroups of such networks.
In the future we might see ITV solutions that build on that kind of ideas and enable viewers to get into contact with other viewers of the current program easily while limiting their communication to friends or other subgroups. There is a desire to talk about what is on TV (see section 3.1 for details) and if the TV stations can address that desire inside the medium itself it could strengthen television's position in the war for consumer's attention.

4.3.3 Mashups

Mashups offer a way to quickly create new applications by combining existing ones. As I mentioned in section 3.4 mashups are mostly provided by the bigger companies and provide a way to 3rd party developers to access functions of an otherwise closed system. TV companies could benefit from the creativity of those developers if they provided them with APIs to build upon. The question at hand is what TV has to offer that can be easily bundled in an API and is of value to outside developers.
The primary asset of television is it's content. For 3rd party developers access to that content is a huge incentive to build new services for television. Currently television companies follow a rather restricted model where they try to retain maximum control over the content's presentation. This is largely due to their business model that is bases on advertising revenues generated around their content. This model would need to open up. 3rd party developers of course would need to reimburse television companies for the content they use but the way they generate revenue does not need to be advertising based.
As an example we might take a look at online personal video recorders. Companies like Shift.TV[3] offer users the ability to record television online and then later on download it from their website. The user does not even need a television set

[3]http://www.shift.tv

anymore. The legal status of this business model is currently disputed in court in Germany as the TV stations are not compensated by Shift.TV. Instead they could have agreed on a mashup-like model, where the TV stations make their content available and Shift.TV in turn reimburses them with a share of the subscription fees they earn.

Another possibility for mashups is to give viewers the possibility to creatively work with content from television. In turn viewers are required to make their creations available to the station again. This strengthens the viewer-station relationship and tabs into the huge creativity potential of the audience that previously has not been utilized at all. Comedy Central's[4] Steven Colbert for example initiated a "Green Screen Challenge" in 2006 where he made a green screen video of him available on the internet and asked viewers to get creative by using it in a video production of their own. In the following weeks he could present the new videos from his viewers on his show, the Colbert Report, with a huge finale where the winner was crowned.

It could be beneficial for TV stations to make use of the creative potential outside of their corporations. They can increase potential revenue of their content and even acquire new content, based on their original one, for almost no monetary investment. However, to secure the viability of this model digital rights management systems are needed to make sure that TV stations can still be guaranteed their rights on their original content. Making it available would need to follow some rules to make it acceptable as a business model. TV stations will not be willing to provide access to their content if they risk uncontrolled redistribution. There is a thin line in the amount of control they can give away.

4.3.4 Advertising

Compared to advertising on television, advertising on the internet offers some novelties. On TV advertising is tailored towards the expected median audience of a show while on the internet advertising can be tailored to each individual user. Companies like Google have collected huge amounts of data on individual internet users and based on their surfing and searching behavior can create fairly accurate profiles of those users.

[4] A US TV station — See `http://www.comedycentral.com`

Advertising scenarios for digital television can build on that experience. New ITV applications could composite custom advertising breaks on the set-top box in pre-defined ad-slots. ITV applications also could make behavior tracking of TV viewers easier and broaden data collection to a greater amount of viewers.
Like Google adwords[5] television ads could be inserted in the standard television flow and be context sensitive in regard to the running program.
In general, advertising on the internet can be seen as anticipating developments in television. People have already accepted that their usage behavior is tracked on the internet. It might well be possible to make use of that adaption for television and introduce ads tailored more at individual viewers in that medium as well.

[5] See `http://adwords.google.com` for details on Google adwords

Chapter 5

The Framework

5.1 Overview

The framework has several functions to fulfill. First of all it is the runtime for the plugins (note that in this report the terms *plugin* and *module* are used interchangeably) that implement the individual services presented in this report. It provides easy-to-use interfaces to the underlying functionality. The framework manages the available TV channels and performs all the necessary operations to receive television signals. It converts all incoming visual media (television or otherwise) to textures. Those textures are then composited in a scene together with a GUI to generate the final view for the user. The framework provides callback mechanisms for plugins to allow handling of certain events by them.

There are already a lot of software solutions on the market for watching television on a PC and some of them also have plugin architectures. However, I found that none of the existing applications fulfilled the specific needs posed by plugins that want to dig deeper. In this chapter I will present some of those existing solutions and their plugin infrastructures. I will explain why I think the current software range is not catering to media usage convergence enough.

Note that specific details on the implementation of the framework can be found in the appendix at chapter A. In this part I would rather give details on the *why* and not the *how*.

5.2 Similar Applications

In this section I would like to present some TV applications and why I decided not to use them. This is not supposed to be a complete overview of the TV software landscape (something which would need to be done much more exhaustively) but should outline some general criticism I have with existing solutions. Most of the existing software can be put into one of the following groups.

5.2.1 Media Center Software

Media Center software sees TV viewing as one building block of a much broader media experience. As the name suggests this software aims at becoming the center of all media usage. Normally users of software from this group would at least be able to browse their picture collection, listen to music, watch TV or watch their home videos. Some programs might also offer radio services, PVR functionality, remote control compatibility, games, news, weather information or any number of other services. In general this applications will run in fullscreen mode and offer a seamless experience. TV watching is not necessarily the central activity here thus often design decisions are made which cater to a more general purpose and not to TV viewing in particular.
Almost every media center solution incorporates a plugin architecture. This is not limited to open source media centers but also includes proprietary applications. Therefore extensive customization of all those applications is possible. Media center plugins usually are able to provide themes, new input methods and mini-applications or consume status messages. Mini-applications could be games, program guides or a weather information display. In section 5.2.3 I will further explain the existing plugin landscape.

Windows Media Center

The most popular application in this category is Microsoft "Windows Media Center". This application has previously only been availably through OEMs that bundled a special version of Windows XP with certain PCs. That so called "Microsoft Windows Media Center Edition" was designed to specifically for the living room. However that version of Windows was not freely available thus limiting the possible market penetration. If consumers first have to buy a completely new computer this is a higher entry barrier than a software purchase. In the upcoming version of Windows, Windows Vista, the Windows Media Center software will be included by default. This will result in a much larger penetration in the years to come which makes the already attractive platform even more attractive.

Windows Media Center offers multiple ways for developers to extend the application[1]. First of all developers can write HTML applications which are then hosted by the Windows Media Center shell. HTML applications for Windows Media Center can utilize ActiveX controls and .NET applets and use all the scripting languages that are also supported by Microsoft Internet Explorer. Windows Media Center exposes multiple interfaces to the scripting environment so developers can query the applications status or issue commands to Windows Media Center. HTML applications can not run in the background but run fullscreen in Windows Media Center. They do not merge with the existing modes (TV viewing, music listening, etc.) but are a mode on their own.

Developers can also develop so-called Add-ins. Those Add-ins can be integrated into Windows Media Center in two different modes. They either can be set to load up when Windows Media Center starts and run in the background the whole time or can be registered to be run on-demand. In the later case the user has to explicitly start an Add-in.

Add-ins are essentially .NET assemblies and therefore can be created in any language that can be compiled to CIL code. The interfaces available to this Add-ins are similar to the ones exposed to HTML applications. They are available to applications in the `Microsoft.MediaCenter.AddIn` and `Microsoft.MediaCenter.Extensibility` namespaces.

Finally, applications can develop COM servers which can be registered with Windows Media Center and then receive status data from Windows Media Center. This can be used to log Windows Media Center usage or display information on the current state on an auxiliary display.

From those three possibilities only the first two could be used to create applications that actually control functionality of Windows Media Center. HTML applications however are really applications of their own and do not integrate with Windows Media Center as would be required for my purpose. That only leaves Add-ins as a viable option. The problem with those is that they do not enable developers to embed any richer additional GUI elements to Windows Media Center. Add-ins are limited to dialogs and notification dialogs. This severely limits the use of Add-ins for my purposes. For example using notification dialogs

[1]Microsoft's central location for information on developing for Windows Media Center is available at http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnanchor/html/anch_winxpmce.asp

to relay continous information to the user is not a good option. Updating overlays would be a much better choice. With respect to the current state extensibility of Windows Media Center I decided it would not be a practical option to choose it as a platform.
It should be noted that the new release of Windows Media Center for Windows Vista will include new ways to extend it. That version will offer support for "Windows Media Center Presentation Layer Applications" and ".NET Framework 3.0 XAML Browser Applications". According to preliminary documentation both will enable graphical video overlays and in general much richer extensions for Windows Media Center. Sadly, this software is not generally available at the moment which rules out this possibility. In the future this could be an exiting development platform. Built on the .NET framework 2.0, tasks like the embedding of web services would be an easy one.

Sceneo TVcentral & Meedio Essentials

Both program are pretty much complete media centers. They are somewhat related with Sceneo focusing on Europe and Meedio on the USA. Both utilize the same plugin architecture, so a plugin works in both applications. Plugins are distributed as .mpp files which are basically just renamed ZIP archives. Inside the archive are an xml file with a description of the plugin, an xml file describing the screen layout, a dll file with the actual plugin and any number of media and data files the plugin uses. Documentation on the how to develop plugins for Sceneo's TVcentral or Meedio Essentials is available via the Meedio Developers Network[2].
Plugins are implemented as COM objects or .NET assemblies. Hence a number of different languages are supported. Plugins need to either implement the `IMeedioPlugin` or `IMLImportPlugin` interface. After being loaded plugins and the Meedio host application communicate via messages. When the host application sends a message to a plugin there are two parameters: a reference to the host system and a message field. Plugins ought to look at the message field and issue commands to the host system based on the message type.
Unfortunately the plugin SDK is not targeted towards TV extensions. In fact there are no documented calls to query the state of the tuner or tune to different

[2]http://www.meedio.com/mdn/index.php?title=Main_Page

channels at all. The module that embeds TV functionality into Meedio is a plugin as well and its messages are not documented in the SDK. When Meedio is run in debug mode all messages are logged and messages from the TV subsystem can be found there. However it is not a viable option to reverse engineer the functionality of that module. Without documentation an implementation is not an option. Therefore this SDK does not offer the necessary functionality to build the plugins I have in mind as well.

MediaPortal

MediaPortal is an open source media center solution that supports all the major functionality a media center application needs to offer. It is also extensible via plugins. MediaPortal itself is written in C#. Plugins have to be build for .NET as well using any language that can be compiled to CIL. MediaPortal uses four different types of plugins:

Process plugins work in the background and do not use a GUI

Tag Reader plugins provide access to metadata stored in media files

External player plugins enable media playback in external applications

GUI plugins allow for adding GUI elements and user interaction

Unfortunately documentation is not available to an acceptable extend. The central starting point for MediaPortal development is the MediaPortal wiki[3]. However at the moment there is only some documentation on the skinning architecture, general coding standards information and a small article which covers the bare essentials on how to write a plugin, but no API documentation. There is a link to a class library documentation which only offers some information on a mere two classes. The plugin development guide is limited to an example of a very simple GUI plugin. Some plugin sources can also be found on the MediaPortal SVN.
Sadly the current state of the MediaPortal documentation is disappointing. The general architecture makes a very good first impression and the GUI & skinning framework as well. Because both the plugins and MediaPortal itself are written for .NET a very tight integration is possible. Plugins can add event handlers to

[3] `http://wiki.team-mediaportal.com/MediaPortalDevelopment`

core systems and directly interact with the core system. This, in my opinion, is far superior than a message based system like the one utilized in Sceneo's TVcentral (see section 5.2.1). MediaPortal certainly is promising but at the moment is not a platform I found viable for use for this report.

TVedia

TVedia from 8 Dimensions stands out of the crowd of other media center applications as it is the only one that does not support TV tuners. It is solely targeted on making users personal media available everywhere in the house. Therefore, it also includes a build in UPnP client and server. I would especially like to mention it here because of its outstanding support for internet services. Support for Flickr, YouTube, Last.fm and others is included directly out of the box. TVedia in that way strives to converge media from the internet with local media. I think this is a very interesting aspect that other media center application do not yet address that prominently. The TVedia GUI is also one of the most compelling in the media center application landscape.
TVedia can be extended but extension is limited to the front-end. The front-end is completely defined by a combination of XML and JavaScript. All the files that make up the interface are accessible in the TVedia installation folder and can be easily changed by anyone with some knowledge of XML and JavaScript. There is extensive information on the XML format and the JavaScript objects exposed available. While this makes it very easy to change the interface to better suit ones needs, this architecture allows no changes to the back-end. Hence it is not possible to e.g. write a custom TV module. That makes an extension of this application for this report not an option.

5.2.2 Pure TV Viewing & PVR Software

This group of applications usually comes bundled with tuner hardware and therefore has a different approach. Usually the main focus is on the handling of TV viewing and TV recording. Therefore this type of applications does not offer near the functionality of full-blown media center applications. The only additional feature that is offered in a lot of applications is a scheduler to plan the recording of TV show. Most applications in this group do not offer a plugin SDK and those who do are often extremely limited. For the purpose of this report this applications hence are not usable as extending them is either not possible or not to a feasible extend. I still would like to present one applications from this group.

EyeTV

Elegato Systems EyeTV software allows for DTV viewing on systems running Mac OSX. While offering all the functionality needed for PVR software the plugin SDK is too limited for the desired purpose. Plugin developers can write plugins that have direct access to the TS and can enable or disable PIDs. Thus no higher level functionality can be implemented by plugin developers which makes EyeTV no viable platform for this report.

5.2.3 Plugin Architectures

What kind of plugins get developed and what is popular at the moment? For example there are currently 278 plugins available for Meedio based software (See section 5.2.1 for details on this applications) at the Meedio add-in directory[4]. 11 of those 278 are input plugins and mostly provide functionality for several remote controls. 35 are themes (note that themes could be seen as an additional category and do not really have that much in common with the other plugins) and 73 are import plugins. Import plugins populate the media library with data from external sources. This could be the latest movie trailers or weather data from the internet. General plugins (105) and Modules (50) together form the bulk of plugins. Those two provide additional functionality while the application is running. This includes things like birthday reminders, alarm clocks or small games. The 5 most popular plugins are:

1. A theme
2. A plugin to enable conditional switching of media players so some media types could be relayed to external media players
3. An internet browser
4. A theme resembling Windows Media Center

[4] `http://www.meedio.com/maid`

5. A plugin to provide some additional general functionality to other plugins

Other plugin repositories have a similarly diverse mix of plugins.
In my opinion the plugin landscape at the moment is chaotic and does not address the most pressing issues if it comes to media usage and digital television. It is in the nature of media centers that they try to include various media types. Therefore it does not come as a surprise that a lot of plugins extend this even further and add more and more different media sources. In a way this dilutes the media center experience. The development has lead in a direction where media center applications try to be jack of all trades. There is a thin line here: media center applications have to agglomerate a lot of media assets in one place. If used in the living room users want access to all their media. But the question is where to stop. Do people really need to read emails on their media center? Should there be a full web browser included into a media center? I'm in no position to decide this and the market will figure out what media centers ought to offer at the end. However, I would like to make an important point here:
Currently, the applications and plugins available try to include more and more media into a converging media experience. They all become distinct assets in a library from which users then are able to pick media for playback. Plugins that try to enhance media usage are almost non-existant. There is also no explicit focus on television. Television streams are handled just like videos or pictures which can be selected in the library and displayed. If we want true convergence we have to address issues inherent to the media usage and not only the management of media.
I do believe that the plugins described in this report are a step in that direction.

5.3 Why another Framework?

There are several reasons why I thought another framework would be necessary. The main reason here being that none of the existing applications allowed extensibility in the needed extend. The main factor was whether extension of the on-screen GUI was possible. As described above this functionality was almost never present and in the cases this was possible other functionality was missing. Most of the available software also builds on a media library system where TV is only one part of the media experience. However, this approach does not lend well to more flexible approaches where the used sources are taken from diverse contexts on-the-fly.

Current solutions make tradeoffs according to their focus. Either they try to provide a full scale media solution for every possible medium a user might use or they try to mimic a television set and do not offer a lot of options to the user. In the second case there also only seldomly is an SDK available making extension of this kind of applications impossible. My goal has been to create an application that mimics a television set but still offers a lot of extensibility. A sort of hybrid of the existing solutions. Thus I needed to come up with a framework of my own.

To some degree the decision to implement a framework myself was also motivated by the desire to better understand the underlying technology. Implementing a prototype makes the difficulties the technology poses experienceable.

Chapter 6

Example Modules

So what kind of extended services could be offered for such a platform? In section 2.1.2 I mentioned Lugmayr et al.'s idea of multimedia services. In this chapter I would like to provide more detail on possible convergence services. I hope that the ideas outlined here might provide a new viewpoint on the current development in the computer TV convergence and a starting point for other more advanced services.

6.1 General Aspects of Module Development

There are some general design decisions I made for the creation of the modules that I would like to quickly outline here. In section 2.4.1 I presented some design choices for interactive services, that I would like to build upon here. First of all, I think that the modules should not replace television but provide an additional layer on top of the standard TV images. Additional services should integrate with the choices already made at the TV studio. Therefore I chose to use overlays for any display purposes and did not allow plugins to alter the content of the television stream itself. Services also should be synchronous and directly related to the current TV stream. Ideally the modules should facilitate communication between viewers. This could either happen by directly linking them or by providing an extended context that benefits off-screen communication. However, the modules introduced in detail here are not directly targeting communication services. They do incorporate that idea to some extend though.

6.2 Social Tuning

6.2.1 Overview

Social tuning takes the concept of social bookmarking from the internet to the TV. In general social bookmarking is all about creating lists of links and sharing that links with other people. Tedd Bryant describes Social Bookmarking as:

> "Social bookmarking with tools such as Delicious and Furl lets users save Web addresses under their accounts online and tag these sites with keywords to organize them and make them searchable. [. . .] It is the social aspect of bookmarking that is of interest to us as educators, however. In addition to browsing your own bookmarks, you can also browse the bookmarks of others, either by looking at the bookmarks of those marked "friends" or by browsing the sites of those who have bookmarked similar or identical sites." [Bryant, 2006, p. 63]

An extension of this principle are sites like Digg.com[1] where users vote on submitted links and the links with the most votes make it to the front page. Popular links thus will be more in the spotlight and in theory the most interesting links should be the ones visible directly on the front page. Every user who wants to submit links or vote on links at Digg.com needs an account. By listing other people as friends on their accounts, users can form networks. They can then see which links their friends submitted and which links they voted for.

The links on the front page of Digg.com in that way are a representation of the current interests of the majority of users. A look at that front page gives an immediate idea of what kind of stories the users of Digg.com are currently interested in. Of course, it has to be taken into account that the users of a page like Digg.com are far less heterogen than all the internet users in general. However, for a small subset of all surfers this is a focal point of their daily surfing habit.

The content of the front page is constantly changing. New links surface and old links get buried. Thus a look at the front page will always be just a snapshot. So Digg.com will capture both the temporal and spatial behavior of users on the web. It can be seen at what time which users are interested in what kind of links. It can even be seen which specific group of users follow specific links and the relationships between the users is also becoming visible in that process. Thus something like Digg.com is a powerful tool. It is currently a niece tool but the concept is being adopted all over the internet and companies like AOL are trying to build up their own equivalent implementations of this concept. After all this kind of community is a marketers dream. Thousands of users exposing even more

[1] http://www.digg.com

of their web usage behavior than they do to Google.

6.2.2 Application

So how could this concept apply to Digital Television? To answer this question we need to find equivalent artifacts in both worlds. The internet is all about individual pages (actually files, which do not have to be HTML files but we will think of those as pages too in this context) and their connections. Those connections are called hyperlinks. By following them new pages can be reached. A page that nobody links to (and is not listed in a search engine) actually is not really part of the internet. It might exist on some server that itself is part of the internet but it does not exist in the net of pages that forms the accessible internet. A page has several attributes. The most important one is its address. The address (or URL) of the page is what other pages can use to link to it. A page might have other metadata attached to it, like the author or the time it was created. We will not concern ourselves with those here. The address of a page is also a key enabler of the above mentioned social bookmarking.

For all these artifacts we can find expressions in the world of television. The smallest unit here is the individual TV channel. A digital TV channel has a certain amount of metadata attached to it that is not available in the world of analog TV (see section 2.2). The extend of available data however varies from broadcaster to broadcaster. Often EPG data is available and information on the program currently running and programs on other channels is accessible. In a way this is similar to a hyperlink. Like a hyperlink provides my browser with all the informaton it needs to load the new page the EPG provides information my TV software needs to tune to that other channel. It also gives me a brief explanation of what to expect on this other channel. Like I can click on a link to get to another page on the internet I can tune to get to another channel available in the broadcast. Both actions carry a similar meaning in their respective contexts.

So if social bookmarking sites like Digg.com can give people an idea of the current browsing habits of other people there has to be a way to do the same with the current tuning habits of television viewers. The social tuning plugin was created for exactly that purpose. It will notify a central webservice of any tuning actions and will also query that webservice for the current tuning statistics.

The webservice has a database of all the tuning actions that occurred in the last couple of minutes. From that data those listings are generated, that the plugin can then consume. The webservice only takes lately occurred tuning events into account and thus always provides a snapshot of the behavior of other viewers. The plugin will present this listing in an appropriate way to the viewer.
The viewer will now always have an idea of the behavior of other people watching TV at the same time. If a lot of people switch to one channel at a certain point that will be visible to other viewers. They might now also switch to that channel as there could be something interesting being broadcasted there. The webservice also provides data about which channel people left to tune to a new channel. If a lot of people tune away from a channel this could provide clues to viewers that this channel is broadcasting something not worth seeing at the moment and is thus probably also not worth tuning to.

6.2.3 Previous Work

Helena Bilandžićs dissertation "Synchrone Programmauswahl" [Bilandžić, 2004] exhaustively deals with the program selection models of individual TV viewers. It provides a valuable insight in the tuning behavior of single viewers. It is specifically not targeting groups of viewers though. In that case a different set of dynamics would apply. The social tuning model proposed here is kind of in the middle. It is designed for individual viewers but connects them with virtual groups. Therefore behavior patterns from both areas certainly play a role. Nicholas Abercrombie e.g. wrote about group viewing dynamics in family settings in [Abercrombie, 1996, pp. 167–173].
Larry Press noticed that the internet already strengthened remote interaction which would leave less time for physical one and notes for TV that:

> "The shift from local to remote is more pronounced when we consider the mass media. We know Johnny Carson better than we know our neighbors, and we are touched more deeply by televised events than those on our own blocks."[Press, 1993, p. 23]

The social tuning model proposed here would even further strengthen that notion as the remote interaction is stressed even further and isolation of individual users could be growing.

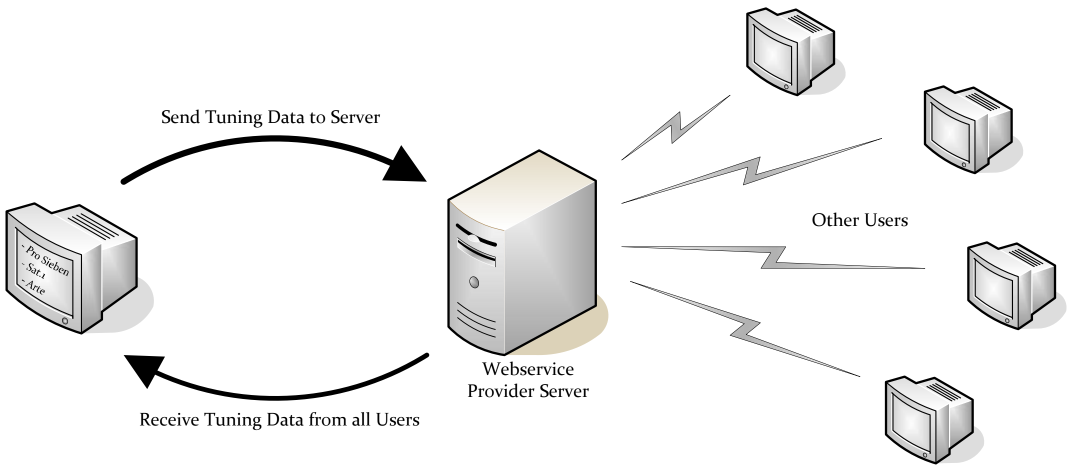

Figure 6.1: Social Tuning Architecture

The whole concept of a "tuning community" heavily builds on the concept of social networks (see section 3.3) and the idea that meaningful results could be generated if a sufficient number of participants contribute (similar to the ideas outlined in James Surowiecki's "Wisdom of Crowds" [Surowiecki, 2004]).

6.2.4 Implementation

There are two components that had to be implemented for this module. One is the plugin for the television framework and the other one is a webservice the plugin can connect to. They communicate via standard HTTP requests using the Representional State Transfer (REST) architectural style. Figure 6.1 shows the general setup of the social tuning architecture.

Webservice

The webservice was implemented in PHP and uses a MySQL database to store the tuning data. It works in two different modes. The basic mode is used if the service is called without any parameters. In that case it will just send a response like the one that can be seen in code 2. First it can be seen that there are two result sets. One is for the channels people tuned to and one is for the channels people left to tune to a new channel. Each result field will specify the number of people who have tuned to or from this channel recently (the timeframe, in which tuning actions will count, is adjustable in the webservice) and the name of the channel. Using the name of the channel as identifier makes sense as the actual tuning parameters a viewer has to use to tune to that channel vary by region and the type of network used to receive this TV signals. The client is responsible for finding the appropriate channel for a given name. The name will correspond to the one that appears in the SI tables[2] so name resolution should be easy for receivers.

The other mode is used if the web service request includes the parameter `newchannelname` plus the optional `oldchannelname`. Sending these parameters to the webservice notifies it of a tuning action that has taken place. The value of each parameter is the name of the channel the viewer tuned to or from. A channel that was left does not have to be specified (for example there is no previous channel when the initial tuning is performed). A tune from ARF to Arte for example looks like this:

Code 2 Example Response from the Social Tuning Webservice

```xml
<?xml version="1.0"?>
<tuneinfo>
  <tuneTo>
          <result count="543" name="Arte" />
          <result count="327" name="Pro Sieben" />
          <result count="212" name="ARD" />
          <result count="205" name="RTL" />
  </tuneTo>
  <tuneFrom>
          <result count="465" name="RTL2" />
          <result count="434" name="RTL" />
          <result count="367" name="NDR" />
          <result count="278" name="Eurosport" />
        </tuneFrom>
</tuneinfo>
```

```
GET /TVtuneInfo.php?newchannelname=Arte&oldchannelname=ARD HTTP/1.1
```

The response will be a simple string acknowledging the tuning action. It is the duty of the webservice to resolve channel name arbitrariness. It is expected that clients use the program name given in the digital channel metadata[3].

[2] See section 2.2 for details

[3] See section 2.2 for details

Figure 6.2: Screenshot of Social Tuning Plugin GUI

Plugin

The plugin itself is fairly small. The only interface it supports apart from the basic `IPluginBase` interface is the `ITVPluginInfo` interface. Interaction with the application is done via the `PluginServiceProvider` (general plugin services), the `GUIServiceProvider` (for custom GUI) and the `ChannelServiceProvider` (to receive tuning events). Upon loading the plugin will register a `TuningEvent-Callback` at the `PluginServiceProvider` and injects a new panel into the GUI. This panel is used to communicate the current TOP4 of channels tuned to and tuned away from. A screenshot of the GUI can be seen in figure 6.2. After the initial startup the plugin checks with the webservice every few seconds for the current tuning data. In case a tuning event takes place the plugin will be notified by the callback and will send a notification to the server.

6.2.5 Results

The social tuning provides an easy way to create a virtual connection with other people who are currently watching TV. Some of the secludedness of sole TV viewers is lifted and the TV experience is enhanced with a sense of connectivity. However, the validation of that thesis would require exhaustive testing for verification. To provide a sense of connection between TV viewers a huge number of users needs to be connected at the same time. If the sample set is too low the listing will be volatile or even empty if a less than adequate number of tuning notifications are processed. User testing with a limited amount of people therefore is not viable and a full scale user test is beyond the scope of this bachelor report. I am still convinced however that the general idea presented here is of interest and could be a starting point for more sophisticated implementations.

Interesting prospects lie in the customization of the social tuning data. Currently everybody's tuning actions have an equal effect on the data send to a user. It would be desirable though to assign greater value to tuning actions from friends or in general to tuning actions of people similar to a user. At Digg.com this is implemented by a friend system where a user can easily see what kind of links his virtual friends liked and which they disliked. Such a two tier system of differently valued datasets could be implemented by creating online profiles for TV viewers and enabling the creation of social networks between those profiles.

Another possible use is that the server's response could include recommendations based on my tuning behavior. This is kind of similar to the recommendation system presented in section 6.3.1 but is not based on a formalization of my viewing behavior but on my currently exhibited switching behavior. This could include my complete switching history but could also be limited to the last hour or so. If the server knew about the content of the currently showing TV channels it could generate quite sophisticated recommendations. For example, a system could detect that a user usually stops switching channels when he encounters a scene where two people kiss. Based on that information the system could notify the user of a show that could satisfy this interest.

The data could also be used outside of the TV experience. It is openly available in XML format and can thus be easily integrated into any application. Webmasters could e.g. display the current tuning data on their website, or developers could create an application that runs on the desktop and notifies the user if a notable peak occurs in the data. With the data freely available it can be mashed up with a range of applications and be used in contexts not even anticipated at the moment.

In general I do not think there is a huge demand for something like social tuning. It does however provide a nice concept of forming on-the-fly virtual communities. I also think that it nicely translates internet concepts of social bookmarking to the TV world. Those concepts are far from mainstream at the moment but I do expect a rise in popularity. User participation currently is the number one key in novel internet services and translating that concept to the television realm could prove to be an interesting way to win back some of the users who already turned away from TV for traditional yet enhanced television.

6.3 Metachannels

6.3.1 Overview

A TV channel can be thought of as a number of successive shows. How a channel is made up is decided by the TV stations. Which shows to show, when to show them, in what order, how often, . . . We can learn about that design of a channel by just tuning to it and watching TV, by reading a TV magazine or by browsing a webpage that offers a program guide. Those channels make up the class of physical channels. But when a TV viewer watches TV he creates his own personal channel. He will switch from channel to channel and in that way create his own individual collection of shows. We could record the TV behavior of a viewer, give the tape to another viewer and this viewer could now also choose between all his normal channels plus the channel of the first viewer. If we did something like that we would introduce a new kind of channel to the television experience.
Each viewers personal channel is not a first class citizen on the TV set. It does not have its own slot, you can not tune to it by pressing a number on a remote control and it is created on the fly while watching the standard channels. I therefore label this personal channel metachannel. It is an abstraction from normal channels. It breaks them down and assembles them again in a new form. A metachannel is a linearly organized collection of references to existing media. Maybe from 13:45 till 15:45 my metachannel shows a show from Pro Sieben. Maybe from 16:15 till 16:20 it shows Arte. Metachannels are a way to formalize viewing behavior and they thus offer a way to study, compare and enhance the viewing experience.
A lot of viewers have their own personal metachannel. Helena Bilandžić for example calls those *TV menues* and states that “there is no doubt that content-based viewing patterns that manifest themselves in individual TV menues exist” [Bilandžić, 2004, p. 38][4]. Maybe not for the whole day but maybe for every

[4]“[. . .] besteht allerdings kein Zweifel an der Existenz inhaltsorientierter Sehmuster, die sich in individuellen Fernsehmenüs niederschlagen” [Bilandžić, 2004, p. 38] — Translation by author

afternoon on a workday or for every Wednesday evening. A lot of people have at least one show they regularly watch and Bilandžić even notes that even if they miss an episode it is not because they watch something else but it is because they do not watch television at all [Bilandžić, 2004, p. 37][5]. Or people routinely turn on the TV to watch the "Tagesschau" every day. The shows we watch might not all be broadcasted on the same channel, so a viewer's personal channel might require him to switch to other channels. If he routinely watches those shows we can assume such a viewer knows exactly when to switch to which channel. If that viewer formalized his tuning / viewing behavior he could just tune to his personal channel which would then transparently do all the tuning for him and would present the desired linear sequence of shows to him.

Furthermore, if a system had access to the formalized viewing behaviors of several viewers it could compare them and generate recommendations for viewers. The concept of automatic recommendations was made very popular by the online bookstore Amazon[6], where users can get recommendations based on their previous purchases. Amazon also enables users to create lists of related titles. A user e.g. might create a list of all the books he thinks are a nice read for a trip to Italy. When another user looks at the product page of a book on Italy, Amazon will display a link to that list with presumably related titles (The book the user is currently viewing needs to be on that list). Both methods enable users to quickly find related media which they might enjoy as well. Users are able to refine their recommendations by providing feedback on them. In that way the system can offer recommendations that are even better tailored towards every individual user. We could adopt that system for TV channels by simply mapping books to shows. The recommendation mechanisms would stay more or less the same.

The next step would be to include non-TV based media into this metachannel model. If we are about to introduce a referential channel model we could as well extend the referential nature to other media. Thus recommendations could include slideshows, video streams or video files from the internet. In that way the metachannel concept enables the definition of custom TV channels that extend

of this report

[5]"Wenn Zuschauer die nächste Folge einer Serie verpassen, dann nicht, weil sie etwas anderes sehen, sondern weil sie zu dem Zeitpunkt überhaupt nicht fernsehen." [Bilandžić, 2004, p. 37] — Translation by author of this report

[6]`http://www.amazon.com`

beyond the TV realm but provide a representation that is specifically tailored towards the TV experience. It is important to keep in mind for the rest of this section that while this concept steams from the introduction of TV referencing metachannels, the notion of metachannels in this report is referring to a more general referential architecture.

6.3.2 Previous Work

The concept of a metachannel is not new. Ballocca et al. for example state that:

> "Our aim is to provide an infrastructure for the user to access seamlessly multimedia content either coming from the broadcast stream or provided/shared by other users thus enabling her/him to assemble a custom program composed of contributions coming from different sources."[Ballocca et al., 2004, p. 282]

In the software landscape the concept of a referential media architecture already manifests e.g. in vidcasting (see section 3.2.1). Some of the vidcasting applications also already specifically draw on a TV-based definition of their purpose. Konstantinos Chorianopoulos uses the term "Virtual Channel" which he sees as a "model that aids the organization and dynamic presentation of digital television programming from a combination of live broadcast, prerecorded content and Internet resources at each set-top box."[Chorianopoulos, 2003, p. 666]. He even goes as far as describing this model as finally empowering TV viewers:

> "The core idea behind the virtual channel proposition [...] is that a part of the decision-making about television programming may shift from the media source to the STB [set-top-box]. The television experience can now be created and controlled at the STB from a combination of local storage, real time broadcast transmissions and Internet resources, thus **offering a set of virtual channels that address each STB user as an important factor in the televised content**."[Chorianopoulos, 2003, pp. 666–667] (Emphasis by author of this report)

Klaus Bruhn Jensen states that according to [Ellis, 1982] and [Williams, 1974] the defining property of TV as a facilitator for communication can not be found in individual shows but in the arrangement of shows, the so-called "Flow" [Jensen, 1996,

p. 183]. He mentions three different kinds of Flow: Channelflow ("Kanalflow"), Viewerflow ("Zuschauerflow") and Superflow. Channelflow is denoting the layout of the shows in one physical TV channel while Viewerflow is the order of shows a viewer creates by switching. The Superflow finally is the combination of those two [Jensen, 1996, p. 183]. Jensen uses the concept of Flow to organize shows into topic buckets. He recorded viewing sessions and then analyzed the switching behavior to deduct similarities between shows (see [Jensen, 1996, p. 183–184]). Hasebrink and Krotz build on the concept of Viewerflow and use recorded switching behavior data to help answer general theoretical questions of TV viewing behavior [Hasebrink and Krotz, 1996, p. 116]. They are particularly interested in the strategy of program selection and in the usage patterns that might be linked to that strategy [Hasebrink and Krotz, 1996, p. 119].

Helena Bilandžić wrote her dissertation [Bilandžić, 2004] on program selection behavior and models. She specifically examines the influence of the structure and content of TV streams on the usage of television and switching behavior in particular.

In general, metachannels / virtual channels are mostly seen as a way to converge different video sources at the TV. The set-top box here becomes a video aggregator. This is quite often mixed with the functionality of a PVR and often presented as a way to break out of the program design of TV executives. This is in contrast to views of other people who are more in favor of the older linear model of television. For example Mountford et al. state that:

> "Conventional linear television, for all its faults, is actually an incredibly powerful, sophisticated and mature medium of communication. I will argue that the best TV [...] is far superior to anything we have yet seen on a computer, Those of us attempting to design the first titles intended for the coming information appliances would, I believe, do well to learn from the success of TV rather than to dismiss it."[Mountford et al., 1992, p. 230]

6.3.3 Application

I would like to introduce and explain my own idea of metachannels and their purpose here. But first of all I would like to state that I think the linear nature of television is vital for the medium. It might not be important for every individual show but I think that it is an important property of the medium. This linear programming does not have a good standing. Mountford et al. even note that: "Conventional linear television is often dismissed by new media evangelists as a contemptible form of passive narcotic. Chewing gum for the eyes. Couch potato mode."[Mountford et al., 1992, p. 230]. So how could this form of a medium be compelling? First of all watching television is very different from surfing the internet. Most viewers do not want to spend any work doing it and television traditionally is seen as "a social relaxing diversion so even the general public do not want to work at it."[Gill and Perera, 2003, p. 85]. So a linear model already offers the desired complexity for this kind of audience. Looking at that we could deduct that every implementation of a metachannel should not introduce excessive usability burdens for TV viewers if it is to be accepted. Additional functionality should fit into the existing channel framework and build on existing TV paradigms.

How could metachannels provide additional value to a TV viewer? The main areas are discoverability of shows, sharing of consumption patterns, integration of diverse media and ease of consumption. I would like to address each aspect individually in the following sections.

Discoverability

> "Consumer surveys indicate that people with 50 channels usually only use about 7 of them."[Sweeney, 1995] via [Gill and Perera, 2003, p. 85]

Why do they not use the other ones? There could be a number of reasons for that. Maybe the viewers only use the number keys on their remote for switching and thus only toggle between channels 1 to 9. Maybe the viewers exclusively use the program guide in their daily newspaper which only includes the main 10 channels. Maybe the viewers stick to channels they are familiar with and that number of channels is that limited. Whatever the reason might be, one can see that it is very unlikely that those viewers will watch programming outside of their everyday viewing patterns. For TV stations this is bad. Even if they had a new show on their channel the likelihood of a viewer discovering that show is low if

that channel is not already one of his favorites.
A metachannel framework that analyzes a viewers usage patterns and then gives recommendations based on that data could eliminate that problem. It would not be important anymore which channel a recommended show is on. If such a system were fine tuned enough the recommendations should be of higher value to viewers as e.g. the recommendations from a TV magazine. The collected data would also be highly interesting for the TV stations themselves. If station executives knew better about the behavior of their viewers they could tailor their program more specifically.
If there was a specific channel to tune to that always showed the show the viewer is most likely interested this could prove to be of high value to viewers. They would be able to watch the content they like without having to invest time finding it. The problem of discoverability disappears. This is similar to search engines on the internet. Without search engines people would need to assemble lists of bookmarks and are very unlikely to surf to any page not already on their list. Managing bookmarks, like managing TV channels, is a tedious task and needs a lot of time of the bookmark set is sufficiently large. Using search engines people do not have to remember URLs but terms to search for. If a user e.g. wants to find out about a new Mercedes car his bookmarks might be limited to the official mercedes site and one to two pages dedicated to cars he frequently visits. By performing an internet search on the appropriate terms the range of pages he has at his disposal immediately becomes much larger. A metachannel also provides the rich result set similar to that of such a web search. Like in a web search much of the content might not have been in the scope of the user before.

Sharing

If a viewer formalized his viewing patterns he can share them. The TV experience becomes transmittable. This opens up a range of possibilities.
For example it provides an easy way for peer groups to share their TV patterns. It is likely that a group of people that share the same interests also would be interested in similar shows. Being able to check what ones best friend watches might provide an novel entry point for the TV experience. Currently, that is done in a non-formalized way. If a member of a peer group likes a certain show, he is sure to tell his friends. Groups might even form solely around the shared interest

for one or a set of shows. However, I would like to argue that a formalized viewing pattern could provide more in-depth information to other viewers. One advantage would be that a system that analyzes a viewer's behavior also can capture patterns that might not be obvious to the viewer himself. While a viewer probably already knows about his favorite shows the system could prove useful for all the less liked shows a viewer watches. An automated solution would be able to provide a much larger scope there. Another advantage could be the time and effort saving properties of formalized viewing patterns.
Sharing of course is not limited to a viewer's peer-group. On the contrary. The true power of the formalization lies in the possibility to share viewing patterns with a much larger audience. For an example it is useful to take a look at Amazon. Like I mentioned above the concept of book lists is somewhat similar to the concept of formalized viewing patterns. Both denote related pieces of media. The true power of those book lists is that they are openly available to everyone. People already do talk with their friends about the books they have read. However, their circle of friends most likely does not encompass all the people that have an interest in that kind of books. All the other people that share that interest are excluded from the book discussions in the peer-group. It makes sense to provide the synopsis of that discussions to those people, too. The final book list, like the viewing patterns, will be a streamlined representation of a conversation that might take place in a peer-group. It is broken down to the essential data and excludes all the redundant parts of the conversation surrounding it (Note that it is another discussion whether that is a desirable effect. Getting ones recommendations from friends in a social manner certainly has its charms and the social aspects can not be mapped to an automated and formalized system).

Media Integration

Referential metachannels allow for the integration of diverse media into the TV experience. As metachannels build on channel concepts the normal television flow is not disturbed. This provides a convenient way to bring media to the TV. It also gives people outside the TV stations the chance to bring their content on the TV screen. All they need to do is put their video content online and build a metachannel around it. Compared to getting air time on television this is an easy task.

Simplification

If a viewer formalized his viewing behavior he would not have to switch the channels anymore. Instead he could just tune to his personal channel where the right orchestration of shows is already running. He still has all the control over his TV experience but he does not need to execute it to have a well-suiting TV experience.
A metachannel could also be extended to provide suiting shows in the timeframes a viewer has not filled with his favorite shows. For example a viewer might formalize his TV viewing behavior from 15:00 till 18:00. Based on that data a system could pick other shows for all the other timeslots. Viewers aren't forced to view that content but it might be a valuable option. In general, the value of metachannels will increase with the number of channels the viewer has available as it becomes harder to keep an overview the more channels are available.

6.3.4 Implementation

For this report I only implemented a limited solution for this problem. There are a number of components here that a full metachannel system would need to implement. First of all there needs to be a way to formalize a users viewing behavior. The same format could also be used to specify a full metachannel that is based on a viewers formalization. At the receiver there needs to be software that creates formalizations, queries metachannel servers, streams content from various sources and finally assembles the custom channels and embeds them in the standard TV interface. At the server side there needs to be software to receive viewer formalization, generate recommendations and make the resulting metachannels available.
For this report I would like to propose a format that enables formalization of TV experiences and also enables metachannel broadcasting to receivers. I also present a webservice providing metachannel management services and a script to provide metachannel access. Finally I would like to present the software component running in my TV framework that embeds that metachannels in the normal TV experience.

Metachannel Formalization

For the metachannel definition I would like to propose an XML format building on the RSS 2.0 specification[7]. This has a number of advantages. First of all, it builds on a standard format that can be understood by a huge number of available applications already. This makes it possible to e.g. also subscribe to a metachannel in an RSS reader software and use it like a program guide. Finally it can be easily extended so adding the desired functionality is quite easy.

Code 3 shows the basic layout of a metachannel RSS feed. Note that the RSS concepts of `channel` and `item` map pretty well to the terms channel and show used in metachannels. In RSS 2.0 there are three mandatory child elements of a channel: `title`, `link` and `description`. For metachannels those will be used respectively for the name of the channel, a link to the channel's homepage and a short description of the channel. If the channel does not have a specific homepage (e.g. it is a private metachannel just for one user) this could also just be a link to the feed's location. Optional channel elements like `copyright`, `language`, `category` or `rating` could prove useful but have been excluded here. A channel also has a required child element called `tv:intermediateMedia`. The media object specified here will be used if no show is currently on the channel. For example in the night there might not be any shows scheduled in the metachannel and a basic image might be displayed informing viewers of that fact. The media object to be displayed is described in the `tv:source` element which I will explain a little bit later.

After the general channel description comes a list of items each denoting one basic show of the channel. In RSS 2.0 all children of an item element are optional. The only requirement is that at least a title or a description is available. For metachannels I decided that the `title`, `link` and `description` elements shall be included to provide a textual representation of the show. This is useful if the feed is not consumed by a TV receiver but by a RSS feed reader. The `item` element has been extended with the `tv:startTime`, `tv:endTime` and `tv:source` elements. While `tv:startTime` is mandatory `tv:endTime` is only optional. Both elements contain a date in RFC822 format[8]. Note that a `dc:date` is also given to provide a time for RSS feed readers.

[7] For the RSS 2.0 specification refer to `http://www.rssboard.org/rss-specification`
[8] `http://www.w3.org/Protocols/rfc822/#z28`

Code 3 Example RSS Feed for a Metachannel

```
<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0"
   xmlns:tv="URL of TV Namespace Definition"
   xmlns:content="http://purl.org/rss/1.0/modules/content/"
   xmlns:dc="http://purl.org/dc/elements/1.1/">
 <channel>
  <title>Station Title</title>
  <link>URL of this station</link>
  <dc:date>Date of last feed update</dc:date>
  <description>Station Description</description>

  <tv:intermediateMedia>
    <tv:source type="image">URL of Image File</tv:source>
  </tv:intermediateMedia>

  <item>
    <title>Show 1</title>
    <link>Link to Page on the Show</link>
    <description>Synopsis of the Show</description>
    <dc:date>Time of the show</dc:date>
    <tv:startTime>RFC 822 time</tv:startTime>
    <tv:endTime>RFC 822 time</tv:endTime>
    <tv:source type="station">Station Name</tv:source>
    <content:encoded xmlns="http://www.w3.org/1999/xhtml">Additional field for description</content:encoded>
  </item>

  <item>
    <title>Show 2</title>
    <link>Link to Page on the Show</link>
    <description>Synopsis of the Show</description>
    <dc:date>Time of the show</dc:date>
    <tv:startTime>RFC 822 time</tv:startTime>
    <tv:endTime>RFC 822 time</tv:endTime>
    <tv:source type="video">URL of a video</tv:source>
    <content:encoded xmlns="http://www.w3.org/1999/xhtml">Additional field for description</content:encoded>
  </item>

 </channel>
</rss>
```

As I mentioned above the `tv:source` element is used to describe individual media objects. To keep the system flexible `tv:source` can be adapted to about every media. The type of the source is specified in the `type` attribute of the `tv:source` element. The element itself will then provide a locator for that media object. Possible types are:

image Locator will be the URL of the image file

stream Locator will be the URL of the stream

station Locator will be the stations name like it appears in the SI tables[9]

video Locator will be the URL of the video file

flash Locator will be the URL of the flash file

metachannel Locator will be the URL of another metachannel

As can be seen the locators in my example mostly are URLs. However the system could be extended to also using data that is broadcasted in a DTV data channel or any other data source. In the case of DTV channels the moduleID and the groupID would need to be specified (see section 2.2 for details on data broadcasting). For this report I will not address this but only deal with available TV channels and media from the internet.
As a sidenote: The RSS specification defines an optional `enclosure` child element that could be used for items. However this solution has some shortcomings that make it less viable in this scenario. The size of the referenced file needs to be specified which makes no sense for streams or TV channels and the locator has to be an URL. Those two could be worked around by simply filling in non-compliant values like 0 for the size and the station name as the URL. However that would most likely confuse other applications that could not properly handle those feeds anymore. The biggest restriction however is that a MIME type[10] has to be specified for the referenced file. There are no MIME types defined yet for TV broadcasts and as MIME is not meant for this purpose, it is not practical to include new types for TV. Therefore I decided not to use the `enclosure` element but create my own element that albeit closely resembles it.

[9]See section 2.2 for details
[10]See `http://www.iana.org/assignments/media-types`

Building the metachannel format on available technology used on the internet has several advantages. First of all it makes metachannels consumable by a huge amount of applications already available. I already mentioned RSS feed readers as one example. There are also a large number of libraries available that make parsing RSS feeds an easy task for developers. Furthermore the format is open and human readable. Users are not restricted to software to create metachannels but could also easily create one manually. Another powerful concept is the referential nature of RSS feeds. No media itself is transmitted in the feed but only referenced in the `tv:source` elements. This makes the creation of new channels as easy as compiling a list of links therefore enabling users to become program directors. Something they normally would never be able to. Referencing of other metachannels enables powerful metachannel hierarchies. Users can build on the programming they like in one metachannel and extend it with their own programming choices making it even easier to tailor a metachannel to ones needs.

Webservice

To offer access to the metachannel database a webservice was implemented. Via the webservice clients can create, edit, remove or query metachannel data. The implementation was done in PHP using the NuSOAP library[11]. Clients can query the webservice for it's WSDL description to find out about available services, the data types used by the services and how to access them. Access to the metachannel database was implemented as a webservice for several reasons. The most important one being that further integration of metachannels in other parts of the project was fairly straightforward. As the interface to the metachannels was formalized in one central location, maintaining it was made significantly easier. For example changes to the database did only affect this one part of the project and were transparent to the rest of it. Having a standard interface to the metachannel functionality also would allow other applications to make use of it. Mashups (see section 3.4) are only possible if such an architecture is in place thus making this an obvious design decision taking into account the goal of this work.

[11] See `http://dietrich.ganx4.com/nusoap` for details

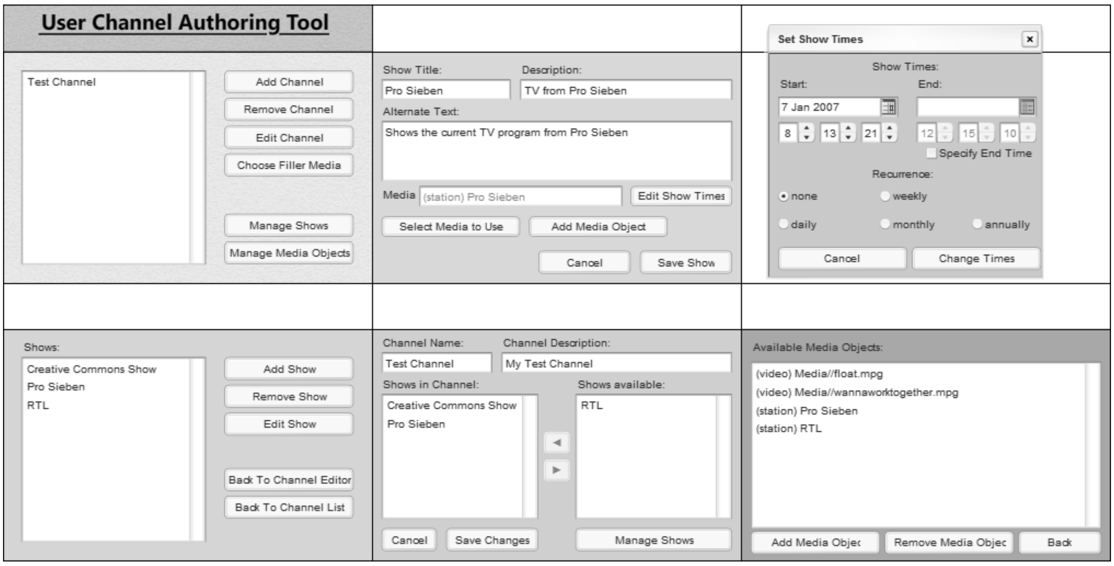

Figure 6.3: Metachannel Authoring Tool

Metachannel Authoring

For the authoring of metachannels I created an application in Macromedia Flash that connects to the metachannel webservice and allows for easy editing of metachannel data. Figure 6.3 shows an overview of the interface. The decision to implement the client in Macromedia Flash was based on it's rapid prototyping capabilities and good support for user interface development. Direct support for webservices formalized in WSDL also made development easier.

Metachannel RSS Feed

To make metachannels easily accessible I implemented a PHP script that makes metachannels available as RSS feeds. In section 6.3.4 I already explained the choice of RSS as format. By default the script will deliver the feed for the first channel saved in the metachannel database. Other channels can be requested by adding a channel identifier to the request. The RSS feed script, like the authoring tool, builds on the webservice presented above. Recurrence settings on shows are automatically translated into individual shows. If e.g. a show is running every week on Monday the RSS feed will have an entry for each occurrence of the show. This makes handling recurrence easier for clients who consume the RSS feed.

Metachannel Receiver Module

The module for the framework is slightly more complex than the social tuning plugin. It basically injects new channels into the channel database of the framework. For that purpose it implements the `IPluginChannelProvider` interface and is notified by the framework when it can inject it's custom channels. The module will then parse a metachannel file from the internet and create a new channel based on the shows contained in it. For that purpose the module utilizes the channel library provided for plugins and the framework itself. See section A.2.3 for details on the channel library. It is the modules duty to properly choose the type of show to create for each metachannel entry. Shows are differentiated by access type which is e.g. TV or file. It is left up to the framework to handle proper show playback.

6.3.5 Results

The plugin currently implements the *sharing*, *simplification* and *media integration* parts described above but doesn not yet help in *discoverability*. It can parse the metachannel files from the internet and translate them to a proper channel for the framework. However, at the moment the only supported media types are TV channel references and media files on the hard disk. For local files several new restrictions apply. As the filtergraph is constructed manually in the framework playback of different file types is a complicated task. Image files for example can currently not be renderer by the framework. There is no accessible image source filter provided with Direct Show and one would need to be written from scratch. There also currently is no way to generate metachannels from viewing behavior alone. A possible solution here would be to incorporate functionality from the social tuning plugin into the metachannel plugin. At the moment metachannels have to be created via an authoring tool. While this is already a convenient way to create formalized viewing patterns it is still far from ideal. The authoring tool also is just a prototype as well.
The prototype at the current stage also does not support recommendations. Finding patterns in the formalized metachannels and deduct recommendations from that data is beyond the scope of this report. It remains up to future work to to explore the possibilities metachannels offer.

While the prototype is still in its early stages there are already some interesting results. First of all the current prototype already provides a "TV-like" feeling while watching metachannels. Tuning to a metachannel is different from simply playing a playlist in a media player. There is a sense of liveness and of programming. The prototype also already facilitates "mashing it up". The webservice interface and standard format make metachannel easily accessible for 3rd-party applications. Thus other services could be build around the available metachannel data already.

6.4 Additional Modules

In the given timeframe of this report only two ideas could be developed up to prototype stage. I would like to use this section to briefly outline two more ideas that fit into the scope of this report but were not implemented.

Daily Soap Extensions

This extensions should give users to possibility to get in touch with other viewers while watching a daily soap. For many people, watching daily soaps, conversation with friends about the watched soaps is common. The soaps are used as a facilitator for communication and exchange on social models. By talking about current events in the show viewers can mediate their own opinions on social situations. A plugin could offer communication and community services to viewers while they are watching the show. While certainly not a replacement for off-screen communication a plugin could make communication with more distant friends easier and provide a fitting context for TV show communities. This could also be used by TV stations to get more immediate feedback from viewers and to communication with them inside the constrains of their medium.

Movie Information Service

Using the metadata available on a channel a plugin could link the current TV stream with data from the Internet Movie Database[12] (imdb). Utilizing imdb would make a huge database full of user-generated content available on the TV

[12] http://www.imdb.com

screen. The trick is to make the data context sensitive. A range of metadata, like the current show's name, is already available on digital television. This could even be extended with image based methods to e.g. analyze the current frame and find out which actors are on-screen. Using that data a plugin could display fitting content on a per-scene level. Mashing up TV and imdb would have benefits for both sides. TV viewers would have on-the-fly data on a movie or its actors available while imdb could leverage its existing content and branch into new target markets.

Chapter 7

Results

7.1 Conclusion

In this report I first presented an overview of the current internet/television landscape, the existing overlap and possible implications of further convergence. After that I presented an application framework and two plugins that make use of it. Both plugins presented showed potential mappings of internet media aspects to the television.
One important question remains that I would like to further detail here:

> *"Is there a need or demand for any of those services?"*

While I wrote about some existing studies here there was none that explicitly stated that people even want to transfer their internet media usage behavior to the television. I did present previous research though that hinted at the fact that people see television and the computer as two media different from each other. I would like to argue though that my vision is not that television should become like the internet. I rather think that the paradigms we deal with in both mediums could be partially mapped to the other and thus bring them closer together on a conceptual level while still being of distinctively different nature. They will not converge on all levels but could benefit from advancements made in each other's area.
Another important aspect of the above question is pointed at by Artur Lugmayr et al. when they state that:

> "Out statement is clear: asking the consumer which new applications he would like is the wrong way to approach new technology. He will answer that he likes to read e-mail, watch movies, listen to music, play DVDs, etc."[Lugmayr et al., 2004, p. 143]

I think that this is a very interesting point. It can not be predicted what kind of technology people want in the future or even what kind of technology today

will become vital in the future. I see this report as one way to probe a possible way television could evolve. Quite possibly my ideas are not what TV is going to look like in 10 years but in my opinion that does not make this work any less interesting.
I do think that it still makes sense though to look at current trends in one are and see how well they translate to the other. If the ideas from the so-called "Web 2.0" movement could be used for television they could reach people currently unable to use the internet. Gathering feedback from that crowd would in turn be quite beneficial for the internet community as well. Many of the "Web 2.0" project have high aiming goals like *revolutionizing television* (see section 3.2.1) but are currently only adopted by a small homogenous crowd. Making the ideas behind those projects available to a more general public could be an interesting test for them.

7.2 Outlook

Were will the future take us? It is safe to say that television will not die. However I think that we will see some changes. First and foremost there seems to be a trend to make television content available in more and more ways. People can already watch a show on live TV, record it and watch it later, view the DVD, access a video on demand service on the internet or download the show to their iPod and watch it on-the-go. It is the high quality content of todays television that makes all this distribution viable. In my opinion there will be a further split between that high quality content (mostly TV series) and lower quality television content (like talk shows or call-in shows). It is not possible to buy episodes of talk shows online or on DVD at the moment and it possibly never will. For that kind of content, that will not be distributed in diverse ways, new usage paradigms might be the most appropriate application.
The internet is the prime example of a medium where huge amounts of bad content make it hard to find some of higher quality. Search engines and link lists were first needed on the internet to cope with the large amount of content. At the same time people have been content with a basic program guide magazine for their TV set. Now more and more channels are being broadcasted. From those channels many are also highly specialized and only target small groups of

people. Thus, TV becomes more overwhelming and the old TV usage paradigms of switching around and having a brief look at the program guide would not work anymore. A mechanism similar to the one presented in section 6.3 might help to handle that amount of content in a better way.

I also think that TV will remain the main medium for live broadcasting for the masses. Unlike the internet, TV can create true mass experiences. Because of the fixed structure it is accessible to anybody and it provides a fixed temporal context as well. As I mentioned in section 2.3 the immediateness of television is an important differentiator to other media. I do not believe that the internet will have a huge influence in that area.

I do however believe that television will become a more referential medium than it already is. Television shows are already full with references to telephone numbers to call, websites to visit or chats to attend. People can turn to television to find out which books to read or which movies to go to. I think that in the future that will be taken even further. In 2004 Matthias Finke and Dirk Balfanz in their paper “A reference architecture supporting hypervideo content for ITV and the internet domain” researched in the same area and presented a concept for hypervideo applications that they see as a “model for the convergence of ITV and the Internet domain” [Finke and Balfanz, 2004, p. 190]. In their work they try to unify video content and the referential structure of the internet. This kind of work could make the referential structure of todays television visible and offer possibilities to extend it even further.

There are a huge number of shows already that make the internet an important part of their show’s concept. This might be just using the internet to facilitate discussion after the show[1] and might go as far as the MyVideo show I wrote about in section 3.2.2. I think that in the future tight integration will become the usual case. This development is hugely influenced by the big media companies that have stakes in both worlds and will try to aim for an even higher level of convergence.

[1] Like the German show “Polylux” that maintains the page Polylog.tv (`http://polylog.tv`) and every week presents a topic with a *pro* and *contra* during the show for the discussion after the show on the internet

Glossary

.NET	Framework by Microsoft that is a combination of the CLR and a suite of class libraries
API	(Application Programming Interface) — Interface allowing code to interact with existing applications or libraries
ATL	(Active Template Library) — Collection of C++ templates to make COM development easier
ATSC	(Advanced Television Systems Committee) — Digital television standard in the USA
BDA	(Broadcast Driver Architecture) — Driver architecture for Microsoft Windows designed to ease development of digital television enabled applications
CAT	(Conditional Access Table) — Table in a MPEG-2 TS containing information on how to descramble a channels video/audio streams
CEGUI	(Crazy Eddie's GUI System) — C++ Library for a GUI subsystem
CIL	(Common Intermediate Language) — Language code in .NET is compiled to and that then in turn is compiled to bytecode that is run by the CLR
CLR	(Common Language Runtime) — Name of the virtual maching in .NET
CLSID	(Class ID) — Identifier for a class in COM based development

COFDM	(Coded Orthogonal Frequency Division Multiplex) — Frequency modulation scheme used e.g. in digital television
COM	(Component Object Model) — Software platform system by Microsoft that facilitates componented development of applications
DLL	(Dynamic-Link Library) — Compiled code that can be shared between applications and is linked at runtime
DTD	(Document Type Definition) — Syntactic schema used to describe the structure of XML files
DTV	(Digital Television)
DVB	(Digital Video Broadcasting) — Collection of standards for digital television
DVB-T	(Digital Video Broadcasting Terrestrial) — Standard for terrestrial broadcasting of digital television
FCC	(Federal Communications Commission) — US agency overseeing the radio spectrum
GDI	(Graphics Device Interface) — GDI is the 2D graphics subsystem in Microsoft Windows
GPU	(Graphics Processing Unit) — Processor for dedicated graphics processing
GUI	(Graphical User Interface)
GUID	(Globally Unique Identifier) — General identifier used e.g. in COM
HD	(High Definition) — Term used to describe new TV standards and TC sets that offer higher resolution images

HTML	(Hypertext Markup Language) — Markup language targeted at websites specifically that facilitates the hyperstructure of the internet with hyperlinks
LGPL	(GNU Lesser General Public License) — Software license created by the Free Software Foundation
MFC	(Microsoft Foundation Classes) — Object-oriented wrapper for the Windows API
MHP	(Multimedia Home Platform) — Platform for interactive TV services by the DVB project
MPEG	(Moving Pictures Expert Group) — Group working on various video and audio encoding standards
MPEG-2	Suite of Audio and Video Compression Standards used for example in DVDs and DTV
OEM	(Original Equipment Manufacturer) — Manufacturer of some parts of a complete system by another company
PAL	(Phase Alternating Line) — One color system in analog TV. Often also used to refer to the corresponding analog TV system in general
PAT	(Program Association Table) — Table containing the PIDs of all the programs in a TS
PID	(Packet ID) — Data stream and tables in a MPEG-2 TS are identified by such IDs. Every Packet belonging to one of them is identified by having the same PID

PMT (Program Map Table) — Table in a TS with references to all the elementary streams associated with one program

PVR (Personal Video Recorder) — Set of devices that record TV to a hard drive in digital form

REST (Representational State Transfer) — Communication style for distributed applications most often on the internet

RSS (Really Simple Syndication) — XML format used to publish digests of websites or other media

SDK (Software Development Kit) — Set of tools and code to develop for a specific platform

SI (Service Information) — Set of tables providing metadata on the available services in DTV

STB (Set-top Box) — Device used to provide additional input choices to older TV sets

SVN (Subversion) — Version control system used to track changes in e.g. software development

TS (Transport Stream) — Multiplexed data stream in MPEG-2 containing video, audio and data packages

UPnP (Universal Plug and Play) — Collection of protocols that enable easy device connectivity and interoperability

URI (Uniform Resource Identifier) — Standard to describe location and/or names of resources

VBI (Vertical Blanking Interval) — Time between two frames introduced to allow the cathode ray to move back to the top of the screen

WSDL (Webservice Description Language) — Language that describes the methods a webservice offers, how to access them and the data types used in the process

XML (Extensible Markup Language) — General-purpose markup language designed as a basis for standardized data interchange

Bibliography

Nicholas Abercrombie. *Television and Society*. Polity Press, Cambridge, UK, 1996. ISBN 0-7456-1436-1.

J.L. Abraham and D.J. Knight. Strategic innovation: leveraging creative action for more profitable growth. *Engineering Management Review, IEEE*, 30(4):23–28, 2002.

Giovanni Ballocca, Luca Broglio, Alessandro Cota, Daniele Peretto, Antonio Ponzetto, and Roberto Politi. A real implementation of personal tv services with deployment in a broadcast demo testbed. In L. Aroyo and C. Tasso, editors, *AH 2004: Workshop Proceedings, Part II*, pages 281–286, 2004.

Gordon Bell and Jim Gemmell. A call for the home media network. *Commun. ACM*, 45(7):71–75, 2002. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/514236.514237.

Helena Bilandžić. *Synchrone Programmauswahl: Der Einfluss formaler und inhaltlicher Merkmale der Fernsehbotschaft auf die Fernsehnutzung*. PhD thesis, Ludwig-Maximilian-University Munich, Munich, Germany, 2004.

Thomas Bjoerner. The early interactive audience of a regional TV-station (DVB-T) in Denmark. In J. Masthoff, R. Griffiths, and L. Pemperton, editors, *Proceedings of the 1st European conference on interactive television: from viewers to actors? (EuroITV)*, pages 91–97, Brighton, UK, April 2–4 2003. University of Brighton.

William Boddy. *New media and popular imagination: launching radio, television, and digital media in the United States*. Oxford University Press, Oxford, UK, 2004. ISBN 0-19-871145-X.

Constance Ledoux Book. *Digital Television - DTV and the Consumer*. Blackwell, Ames, Iowa, 2004. ISBN 0-8138-0927-4.

Todd Bryant. Social software in academia. *EDUCAUSE Quarterly*, 29(2):61–64, 2006.

Konstantinos Chorianopoulos. The digital set-top box as a virtual channel provider. In *CHI '03: CHI '03 extended abstracts on Human factors in computing systems*, pages 666–667, New York, NY, 2003. ACM Press. ISBN 1-58113-637-4. doi: http://doi.acm.org/10.1145/765891.765919.

John Ellis. *Visible fictions*. Routledge Kegan Paul, London, UK, 1982.

Matthias Finke and Dirk Balfanz. A reference architecture supporting hypervideo content for ITV and the internet domain. *Computers & Graphics*, 28(2):179–191, April 2004. doi: 10.1016/j.cag.2003.12.005.

Walter Fischer. *Digital Television: A Practical Guide for Engineers*. Springer, Berlin, Germany, 2004. ISBN 3-540-01155-2.

J. M. Gill and S. A. Perera. Accessible Universal Design of Interactive Digital Television. In *Proceedings of the 1st European conference on interactive television: from viewers to actors? (EuroITV)*, pages 83–89, Brighton, UK, April 2–4 2003. University of Brighton.

Uwe Hasebrink and Friedrich Krotz. Individuelle Nutzungsmuster von Fernsehzuschauern. In Uwe Hasebrink and Friedrich Krotz, editors, *Die Zuschauer als Fernsehregisseure?: zum Verständnis individueller Nutzungs- und Rezeptionsmuster*, pages 116–137. Nomos Verlagsgesellschaft, Baden-Baden, Germany; Hamburg, Germany, 1st edition, 1996. ISBN 3-7890-3600-5.

Jochen Hörisch. *Der Sinn und die Sinne. Eine Geschichte der Medien*. Eichborn, Frankfurt am Main, Germany, 2001. ISBN 3-8218-4195-8.

Klaus Bruhn Jensen. Superthemen der Rezeption: Von Fernsehnachrichten zum Fernsehflow. In Uwe Hasebrink and Friedrich Krotz, editors, *Die Zuschauer als Fernsehregisseure?: zum Verständnis individueller Nutzungs- und Rezeptionsmuster*, pages 178–196. Nomos Verlagsgesellschaft, Baden-Baden, Germany; Hamburg, Germany, 1st edition, 1996. ISBN 3-7890-3600-5.

Sheri Lamont. An 8-step process for creating compelling enhanced television. In *Proceedings of the 1st European conference on interactive television: from viewers to actors? (EuroITV)*, pages 133–136, Brighton, UK, April 2–4 2003. University of Brighton.

Artur Lugmayr, Samuli Niiranen, and Seppo Kalli. *Digital Interactive TV and Metadata*. Springer, New York, 2004. ISBN 0-387-20843-7.

Tara McPherson. Tv predicts its future: On convergence and cybertelevision. In Beth E. Kolko, editor, *Virtual Publics: policy and community in an electronic age*, pages 175–191. Columbia University Press, New York, NY, 2003. ISBN 0-231-11827-9.

Microsoft Corporation. *Windows Server 2003 SP1 Platform SDK Documentation*. `http://msdn.microsoft.com/platformsdk`.

S. Joy Mountford, Peter Mitchell, Pat O'Hara, Joe Sparks, and Max Whitby. When TVs are computers are TVs (panel). In *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 227–230, New York, NY, USA, 1992. ACM Press. ISBN 0-89791-513-5. doi: http://doi.acm.org/10.1145/142750.142796.

Horst W. Opaschowski. *Generation @: Die Medienrevolution entläßt ihre Kinder: Leben im Informationszeitalter*. British-American Tobacco (Germany) GmbH, Hamburg, Germany, 1st edition, 1999. ISBN 3-616-06871-1.

Bruce M. Owen. *The Internet Challenge to Television*. Harvard University Press, Cambridge, Massachusetts, 1999. ISBN 0-674-87299-1.

Mark D. Pesce. *Programming Microsoft DirectShow for Digital Video and Television*. Microsoft Press, Redmond, 2003. ISBN 0-7356-1821-6.

Larry Press. The internet and interactive television. *Communications of the ACM*, 36(12):19–23,140–141, 1993.

Célia Quico. Are communication services the killer application for interactive tv? or “i left my wife because i am in love with the tv set”. In *Proceedings of the 1st European conference on interactive television: from viewers to actors? (EuroITV)*, pages 99–107, Brighton, UK, April 2–4 2003. University of Brighton.

Ulrich Reimers. *DVB: The Family of International Standards for Digital Video Broadcasting*. Springer, Berlin, Germany, 2. ed edition, 2005. ISBN 3-540-43545-X.

Howard Rheingold. *The Virtual Community: Homesteading on the Electronic Frontier*. Addison-Wesley, Reading, Mass., USA, 1993. ISBN 0-201-60870-7.

Ellen Seiter. *Television and New Media Audiences*. Oxford University Press, Oxford, England, 1999. ISBN 0-19-871141-7.

James Surowiecki. *The wisdom of crowds: why the many are smarter than the few and how collective wisdom shapes business, economies, societies, and nations*. Doubleday, New York, NY, USA, 1. ed edition, 2004. ISBN 0-385-50386-5.

J. Sweeney. An introduction to interactive television. In *IBC 94. International Broadcasting Convention*, pages 503–508, 1995.

Vivi Theodoropoulou. The rise or the fall of interactivity? Digital Television and the “first generation” of the digital audience in the UK. Paper presented at the Ripe@2002 Conference, Helsinki/Tampere, Finland, 17-19 January., 2003.

Raymond Williams. *Television: Technology and cultural form*. Fontana, Glasgow, UK, 1974.

Appendix A

Implementation

I would like to use this part of the report to present my implementation. I think that while the implementation details are not important for the understanding of the reports main thesis it still is an important part of my work. Motivation to build this prototype was mainly the desire to have an actual platform to test out my ideas. I also hoped it would provide me with a better understanding of the underlying technology.
For the implementation I used Microsoft Visual C++ 2005 Express Edition which is available for free. During the software development process Subversion was used for source control. The webservice components were hosted on an Apache server running the PHP modules and used a MySQL database in the backend. Thus all tools were freely available. For the development several third party libraries were utilized. First of all the Windows API was used including DirectShow, DirectX and lots of other parts. Parts of the Boost libraries and the STL was used as well. XML handling is done via TinyXML. For the GUI Crazy Eddie's GUI System was used.

A.1 General Software Architecture

The framework was written for the Microsoft Windows XP operating system in the C++ language. The main reason for that was that on the Windows XP operating system applications can use the Microsoft TV Technologies components that make development of applications incorporating TV functionality a lot easier. I will explain Microsoft TV Technologies more in depth in section A.2.1. The framework also utilizes DirectShow for all media handling and Direct3D for the final rendering.
The center of the application is the `TVApplication` singleton. It instantiates all the subsystems and provides access to them. Subsystems include the renderer, the input manager, the plugin manager, the channel manager, the GUI and the

media system. The whole prototype was designed with modularization and ease of extensibility in mind. This especially applies to the plugin architecture which I will present in more detail later on.

A.2 Media Subsystem

The media subsystem utilizes Microsoft's DirectShow for the playback of media assets. In DirectShow there are two different types of objects: filters and filtergraphs. Filters are the basic building blocks used to construct filtergraphs. They can be classified according to their position in a filtergraph. First of all there are source filters which input data from a file, a webcam, a microphone or any other device or source into a filtergraph. At the end of a filtergraph there are renderer filters. They are used to output incoming audio data to the computer speakers, render video data to a window or save incoming data to a file. Between source and renderer filters can be any number of transform filters. Their functions can be anything from converting color from one colorspace to another, decoding video or audio, demultiplexing stream data or adding echo to audio data.
A filter can be connected to multiple other filters. The number of connections a filter can have depends on the number of pins it has. Every filter has zero or more input pins and zero or more output pins. Source filters only have output pins and renderer filters only have input pins. A filter might for example have multiple output pins if it demultiplexes a stream into a video and an audio signal. When the filtergraph is constructed all the filters negotiate their connections. DirectShow tries to match input and output pins that handle the same format.
For more detailed information on DirectShow please refer to the Microsoft Platform SDK documentation [Microsoft Corporation, Graphics and Multimedia / DirectShow] or "Programming Microsoft DirectShow for Digital Video and Television"[Pesce, 2003] from Mark Pesce.

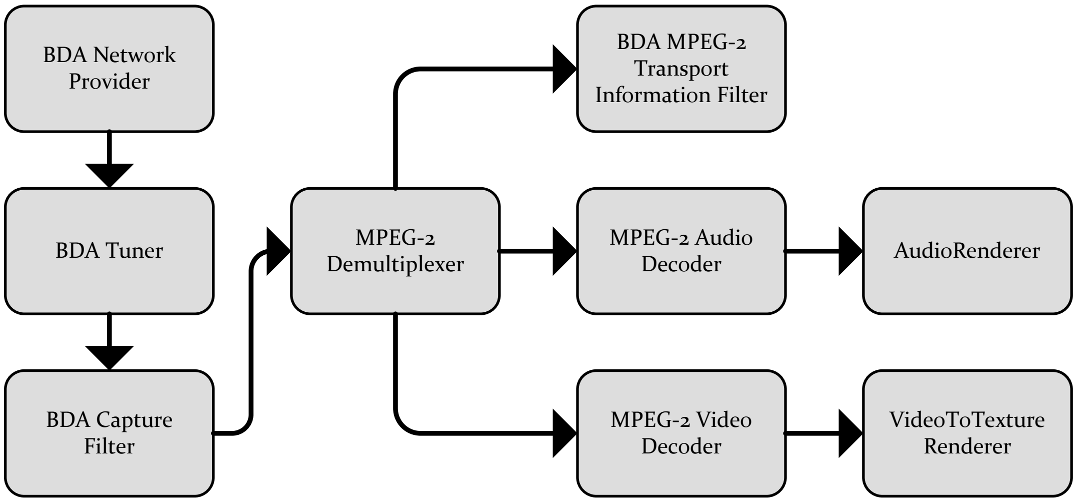

Figure A.1: Structure of the TV Filtergraph

A.2.1 TV Functionality

TV functionality is implemented via Microsoft TV Technologies[1]. Microsoft TV technologies consists of a set of DirectShow filters that can be used to input TV data into a DirectShow filtergraph and the BDA specifications for driver developers. Device manufacturers have to supply BDA compatible drivers for their TV cards to be compatible with Microsoft TV Technologies. BDA drivers are a requirement for Windows Media Center compatibility, so BDA compatible TV cards are already widespread. For application developers the details of the specific TV tuner card are completely transparent which reduces the hassle of developing TV enabled applications.

There are multiple ways to integrate TV functionality in an application. First of all Microsoft provides an ActiveX control for easy integration into applications. The control has to be hosted by an ActiveX container like a Visual Basic form or a Microsoft Word document. It can easily be added to an ATL or MFC based application. As the framework for this report was implemented in pure C++ without ATL or MFC that ActiveX control could not be used. This leaves the second option of building the DirectShow filtergraph manually. Figure A.1 shows the general structure of a such a TV filtergraph.

Constructing a TV filtergraph is pretty straightforward. The BDA Network

[1]See Microsoft Platform SDK documentation [Microsoft Corporation, Graphics and Multimedia / Direct Show / Using Direct Show / Microsoft TV Technologies]

Provider, MPEG-2 Demultiplexer and BDA MPEG-2 Transport Information filters can be created by their CLSID and added to the graph. The BDA Tuner and BDA Capture filters are provided by the device driver of a tuner card. To retrieve them applications need to create instances via the System Device Enumerator. All available BDA Tuners for example can be found in the `KSCATEGORY_BDANETWORK_TUNER` category of the System Device Enumerator. The audio and video decoder filters are added automatically by DirectShow if the MPEG-2 Demultiplexer filter is advised to render its output pins. The renderer filter also does not need to be added manually as DirectShow handles that as well. But as a non-standard video rendering filter is used (see section A.2.2) this one has to be added manually.

A.2.2 Video Textures

There are a number of reasons why I decided to render the video to a texture. First of all I wanted to have the GUI embedded as an overlay on the video surface and I also wanted to have to option to perform transformations and effects on the video. If the video is available as a texture it can be used in any scene and can easily be processed by pixel shaders. I did not make use of that functionality in this report but a design that could provide maximum flexibility was deemed desirable.

So what are the options if those two requirements have to be met? If the video is rendered by one of the standard rendering filters the only options are rendering to a new window, rendering to an existing window's region and rendering to a custom DirectX9 surface. Rendering to a new window is no option as this window ca not be easily controlled. Rendering to a region in an existing window is feasible but rendering overlays over that video via GDI is not a good option. If the video is rendered as a video overlay[2] nothing can be drawn on top of it. Also transformations of that video would not be easily possible.

This leaves rendering to a custom DirectX9 surface. To do that one needs to supply a custom allocator-presenter to the Video Mixing Renderer 9 (VMR9). A custom allocator-presenter is responsible for the allocation of the surfaces for the video renderer, is notified if that surfaces are ready to be presented and can then use this surfaces in a scene. The problem here are multithreading issues

[2]The video data is directly written to GPU memory and thus never passes the CPU. The final frame is composited on the GPU

and handling of device losses. Control over the allocated surfaces is lost once they're handed over to the VMR9 and in case of a device loss it is hard to properly deallocate all surfaces associated with the device. Because the DirectShow parts and the general rendering parts of the system are running in different threads synchronization is also tricky.

Because of those problems I decided to write a custom video renderer filter. Writing a custom renderer provides full control over the memory handling inside the filter. Writing such a DirectShow filter is quite straightforward as stub implementations of base classes are provided. To develope a custom renderer filter one needs to extend `CBaseVideoRenderer` from the DirectShow base classes and provide a couple of simple functions:

```
HRESULT CheckMediaType(const CMediaType * pmt);  // Negotiate Media Type
HRESULT SetMediaType(const CMediaType * pmt);  // Finalize Media Type Choice
HRESULT DoRenderSample(IMediaSample * pMediaSample);  // Render a Sample
```

The filter was written so that it only accepts RGB32 video data (8 bit for each of red, green and blue and 8 padding bits). It is delegated to the upstream filters to provide a proper video stream. This format is a good choice as it is a common texture format and no conversion has to be performed in the rendering filter to copy the video data to the texture. For the same reason interlaced formats and compressed bitmap formats are not accepted.

Once the negotiations have succeeded and the filtergraph is run video data will be streaming to the rendering filter. Every frame will be passed to the filter via the `DoRenderSample` method. The texture surface is locked and the video data is copied row by row to the texture. Because the texture's height and width does not necessarily comply with the video resolution (textures are often required to be a power of two in size) padding has to be taken into account. Because the copying is done in a thread different to the main application thread some precautions have to be taken. First of all the video textures are double buffered. The texture currently used by the application is not the one that is used to render video to. The video renderer filter also makes use of critical sections to protect volatile areas.

The custom video renderer filter has built-in functions for device loss handling. If the device is lost all resources are deallocated so the device can be reset. Video samples arriving at that time are simply discarded.

Using the custom video renderer in an application is quite straightforward. It only needs to be added to a filtergraph and the texture can then be requested for rendering. The filter needs to be advised of device losses but apart from that the application does not need to concern itself with it.
Having the video now available as a texture gives way to a whole new range of options. It could be mapped to about any surface and included in any scene. Like I mentioned above performing image processing on it using the GPU via pixel shaders is a straightforward option as well. In the current implementation of the framework the video is simply used as a sprite. However composition of the video and the GUI is made very easy by that method.

A.2.3 Channel Library

Definitions of the channel class and the classes for shows is done in an extra library. This was done to make this data types available to module developers that want to create their own channels to inject into the framework.
A channel is the basic class in that hierarchy. It has a name and contains a collection of shows. Channels offer functionality to add and remove shows and to query for the currently running show.
Shows all extend from the abstract `show` class and currently exist as incarnations for TV shows and file shows. Each show type can be queried for it's access method that is used by the framework to construct an appropriate filtergraph. Shows also contain various metadata and data on their timing. They can for example be queried whether they are *on* at a given time or if they start before another show. This is used e.g. to efficiently sort shows.

A.3 GUI

For the GUI there were several requirements. It needed to be overlayed on the video and extensible by plugins. For maximal compatibility it was decided that the GUI system should be based on DirectX9 as well. As I mentioned in section A.2.2 that makes integration of the GUI and the video an easy task and solves the overlay issue. Extensibility however is a matter of design. It became clear quite early that building a GUI library of my own would be a huge task and probably not feasible in the given time frame. Therefore I decided to go for an existing

solution and chose Crazy Eddie's GUI System (CEGUI)[3] to be included in the framework.
CEGUI has several advantages. First of all it is freely available under the LGPL license. It works with DirectX9 and allows for easy inclusion into any project. At the base CEGUI is a hierarchical window based GUI system. The framework initializes the general components and creates a root window that spans the entire application window. That root window is then exposed and can be used by other parts of the application. Plugins can also query the root window via the `PluginServiceProvider`. Mouse, window and keyboard events are injected into the CEGUI system by the application and CEGUI then processes and relays them to the proper components.
The choice of CEGUI as GUI system turned out to be quite adequate. Especially the easy extension by plugins was a major point. The GUI interface to the plugins could be limited to a single function that exposes the root window. All interaction with the GUI is then done via the CEGUI system that plugin developers have to include and link as well. For future extensions it would be advisable to include some kind of screen space partitioning system. Currently plugins do not know whether the area of the screen they put there GUI at is already used by another plugin. A request mechanism for GUI placement would therefore be a useful addition.

A.4 Plugin Environment

The plugin environment is modeled after the example of COM[4]. However for the sake of reduced complexity only some aspects of COM have been adopted.
Plugins at the moment can only be implemented in C++. This was a design decision to keep complexity at a lower level. More advanced implementation could fully utilize COM or build on the .NET framework for language independence. But implementing COM servers is not an easy task and the framework currently does not use .NET so those two options were ruled out.
The most basic plugins are only required to provide a class implementing the

[3] See `http://www.cegui.org.uk`

[4] See e.g. Microsofts documentation at `http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnanchor/html/componentobjectmodelanchor.asp` for a number of resources on COM

`IPluginBase` interfaces and export it together with the `GetPlugin` function in a DLL. This DLL is then placed in the `Plugins` directory of the framework. Upon startup the framework will try to load all the DLLs from that directory into its address space by using `LoadLibrary`. If a DLL is loaded successfully the framework will attempt to retrieve the address of the `GetPlugin` function in the DLL via `GetProcAddress`. That function is then responsible for the instantiation of the plugin and should return that instance to the framework. This concept closely resembles that of a class factory in a COM server.

After a plugin is loaded its `Init` function is called. Plugin initialization is advised to be done in that function instead of the constructor as it has a return value and can directly indicate an error. The plugin is then notified of the `PluginServiceProvider` of the framework. After that step a two way binding is established. The framework has loaded the plugin and can query it for its supported functionality and the plugin on the other hand knows about the interface to the framework (the `PluginServiceProvider`) and can query it for supported interfaces.

As mentioned above, plugins and the framework support multiple interfaces. It was decided that only rudimentary functionality is put in the base interfaces and further functionality is only available in extended interfaces. Therefore a plugin can for example query the `PluginServiceProvider` for the `GUIServiceProvider` interface to change the GUI of the framework and inject custom GUI code. This is another adaptation from COM and is done to provide an extensive level of componentization in the framework. Older plugins could still be used with newer versions of the framework if the older interfaces remain unchanged. Also newer plugins could be used with older versions of the framework. Interfaces in the framework are identified by GUIDs like in COM.

Finally the framework provides a callback mechanism for plugins. Plugins can register for callbacks with the `PluginServiceProvider`. Callback identifiers are of type GUID as well and thus also extensible. New interfaces can also introduce new callback IDs which plugins can then utilize.

Why is such a strong componentization desirable? First of all it makes developing plugins easier. If a plugin does not interact with the GUI it does not need to include those header files and also does not need to link with the CEGUI libraries. Plugins do not need to be recompiled for every new version of the framework

and introducing new functionality is easy.
In general the architecture of the plugin framework is kept at a pretty low level and plugin developers need to implement a lot themselves. This provides maximal flexibility and possibilities. As I outlined in section 5.2.3 existing plugin architectures restrict developers quite a lot. Also the focus in existing plugin architectures quite often is a different one. I think the proposed architecture (while closely resembling some of the existing ones) provides an interesting starting point for plugins that want to do more than just import pictures from Flickr into a photo library. The low level architecture provides access to the media handling itself and the GUI. Of course, the functionality of the prototype is limited at the moment but as I mentioned above the general architecture was designed with extensibility in mind.

Appendix B

Attachments

Attached to this report a CD-ROM with the source code of the framework, the webservices and the example plugins plus compiled versions of the application and the plugins can be found. The binaries enclosed are debug builds (and contain no optimization at the moment) for the Microsoft Windows XP operating system. Thus either Microsoft Windows XP or one of its successors is needed to run the compiled binaries. Furthermore the application is currently compiled against the DirectX libraries from the June 2006 SDK DirectX SDK. The DirectX runtime's shared library files of that or a newer version of DirectX are therefore required as well. A redistributable version of the DirectX9 runtime files can be found at Microsoft's DirectX homepage[1].

Furthermore a DVB-T tuner with BDA drivers needs to be present and an MPEG-2 decoder filter needs to be installed on the system. Note that during my testing I noticed that the current prototype does not work well with all MPEG-2 decoders. This is most likely due to the rendering filter requiring RGB32 input which some decoders can not provide properly. The prototype is currently configured for the DVB-T region of Bremen. If a different set of TV channels is desired the `channels.conf` file needs to be replaced with one for the desired target region[2].

The web components require an Apache webserver with the PHP module installed and a MySQL database. The database settings have to be adjusted in the appropriate `config.php` files included with each web component. Load `install.php` to create the necessary database tables. Each web component includes a small start page that gives further hints on how to get started.

[1] `http://www.microsoft.com/directx`

[2] `channels.conf` files for DVB-T in Germany can among others be found at `http://www.vdr-wiki.de/wiki/index.php/DVB-T_channels.conf`