Abstract

It is common practice to use web technologies when creating graphical user interfaces for IPTV set-top boxes. The graphical user interface in TeliaSonera's IPTV service is built with HTML, CSS and Javascript. Lately focus has shifted from just showing video to also provide richer user experiences – coming from requirements of new services and the ongoing shift in resolution from standard definition to high definition. However, the set-top box is a device with very limited resources and it is important to assure good performance while allowing richer user interfaces.

Against that background the thesis asks "how can a web-based graphical user interface for IPTV set-top boxes, such as that of TeliaSonera, be improved and prepared for future IPTV services and richer user experiences?"

To provide an answer two studies were made; one of web technologies and one of future IPTV services. The technologies were evaluated in a Pugh matrix against a set of criteria (the most important being performance, platform independence, and resolution independence) as well as against the results of the IPTV services study.

SVG (Scalable Vector Graphics) scored best in the evaluation. The SVG renderer performed significantly better on the set-top box than the Mozilla web browser that TeliaSonera currently used. To put the technology to test the HTML based user interface was ported to SVG with expectations of a more responsive and graphically rich user interface. The result was satisfying. Loading times had been halved, the user interface works in both SD and HD resolution and richer graphics were added in form of gradients and a skinning system that allow easy change of appearance.

The thesis recommends SVG when building a rich graphical user interface for a set-top box.

Chapter 1
Introduction

TeliaSonera is the leading IPTV operator in Sweden. The graphical user interface (GUI) used in their set-top boxes (STBs) is built with web technologies; HTML, CSS and Javascript.

The company now wants to improve their STB GUI to become faster, withstand the transition from standard definition (SD) graphics to high definition (HD) graphics and also be able to manage new IPTV services as they may arrive in the near to mid future.

1.1 Problem description

Common practice today when creating GUIs for IPTV STBs is to use web technologies; they have been sufficient for providing the functionality required the user interface, they are easy to learn, widely adopted and has a large developer community. The user interface can run in a web browser, which makes it easy to move across different platforms.

Lately it has become important to provide a rich user experience apart from showing video, which is main function of the STB. The STB has dedicated hardware to handle video, but the overall performance is limited. In fact, the STB is a small computer with very limited resources. To provide a richer user interface the performance of the STB and the technology that runs on it is a crucial factor.

The research question for this thesis is defined as follows:

> *How can a web-based GUI for IPTV STBs, such as that of TeliaSonera, be improved and prepared for future IPTV services and richer user experiences?*

A summary of future IPTV services is derived from interviews with leading players in the IPTV STB and service area. A study of available web technologies is then performed with focus on criteria developed in collaboration with IPTV experts at TeliaSonera.

The web technologies are evaluated with respect to the criteria and the new requirement from future IPTV services, and finally a prototype is constructed with the best suited technology.

1.2 Objective

The purpose of the thesis is to present a technology that is well suited for a modern browser-base STB GUI and apply it on a real-world example. The thesis will both give theoretical understanding of what is expected of such a technology and put it into practice to verify the result in a real-world complex system.

1.3 Criteria and scope

Five criteria have been developed together with experts in the IPTV group at TeliaSonera. The criteria form the focus area of the technology study and are used when evaluating the technologies.

Performance The ability to have a fast user interface with minimum response time to user interaction.

Platform independence The user interface should be consistent across multiple terminals with different hardware capabilities.

Resolution independence A better user experience when moving from SD to HD resolution.

Rich user experience The ability to easily customize the appearance of the user interface (skinning), the ability to provide richer graphical effects, such as animations.

Development community A user interface built with tools and technologies that have a large development community facilitates development and maintenance.

The technology research is limited to web based technologies, viewable in a browser, and because of limitations in the lab environment only the technologies that are able to run on the available equipment will be tested.

1.4 Methodology

The methodology used throughout the thesis is a mixture of focused interviews, literature studies, decision making and software implementation.

Chapter 2 Set-top box and IPTV services study is mostly based on focused interviews. That is, the questions are not made up on beforehand; instead the interview focuses on some specific areas that are of interest and lets the interviewee speak more of less freely about these [1]. In this case the areas are STB history and future IPTV services. The interviews are made with three IPTV STB manufactures – ADB, Amino and Motorola – and an IPTV service developer company - Accedo Broadband. Besides interviews whitepapers from the Open IPTV Forum are studied.

Chapter 3 Technology study is based on a literature study over available web-technologies. The literature mostly consists of specifications and papers collected from the different technologies' web sites. To gain knowledge about what technologies exist, online dictionaries, news groups and forums have been used. The criteria from 1.3 Criteria and scope are defined more thoroughly and serve as a basis for what material is presented.

The results from Chapter 2 are analyzed regarding what impact they will have on a STB GUI. Each technology from Chapter 3 is then evaluated with respect to:

- Whether the technology supports the IPTV services.
- Whether the technology fulfills the criteria defined earlier.

The standard format for evaluation in decision theory is called a decision matrix [2]. It evaluates and prioritizes a list of options against a set of weighted criteria. The process of a decision-matrix model may vary depending on sources, but usually the same base structure is used [2] [3] [4].

1. Identify the decision to me made.
2. Identify criteria that are of importance for the decision.
3. Assign individual weights to each criterion.
4. Identify the possible alternatives to be compared.
5. Score the alternatives for each criterion using a decision matrix or another evaluation tool.
6. Choose the alternative with the highest score.

The decision matrix is an L-shaped matrix which tabulates the alternatives to the criteria. It then scores each alternative relative to each criterion according to a predefined scoring system. The Pugh matrix is a special form of a decision matrix that uses one of the alternatives as reference [2]. Every other alternative is then scored relatively the reference using a scale of "worse, same or better" (or in more steps) [2].

The Pugh matrix is fitting in this case because a reference alternative already exists: the current implementation of the GUI.

The results from the evaluation are then applied in Chapter 5 Case TeliaSonera. A prototype based on the current GUI is implemented. The implementation process consists of three steps:

1. Define requirements on the implementation derived from the criteria.
2. Analyze the current GUI to identify what parts (if any) can be reused (the lowest common denominator of the new and old technology).
3. Implement.

Finally the result of the prototype implementation is presented with focus on the requirements.

1.4.1 Qualitative and quantitative methods

Mostly, this thesis is based on quantitative data. The results rely on measurable data, such as capabilities and characteristics of the various technologies. But the most important decisions come from qualitative data that is derived from interviews and discussions. These decisions determine how the quantitative data is used and what data is most important. An example is the technology evaluation. The evaluation is made using quantitative data, but the different criteria and their weights that control the outcome are selected based on qualitative data.

1.5 Related work

In the doctorial thesis "A Graphics Software Architecture for High-End Interactive TV Terminals" [5] Pablo Cesar suggests an architecture for DVB TV terminals based on the Java GEM[1] standard. Though the graphics layers are based on Java he suggests that the client should be able to render XML-based content. However, XML content is only suggested for simple (non interactive) applications and the graphics profiles use XForms in favor of a scripting language.

In the master thesis "STB application development based on modern web technology" [6] Niels Bosma investigates how modern web technologies can inspire development methods of set-top box portals. He states that the browser environment is preferred due to its flexibility, but has drawbacks, such as large memory footprint and low performance. After concluding that STB-portals could benefit of modern web technologies, he creates an application framework to facilitate building graphical STB-portals.

[1] http://www.mhp.org/introduction.htm

The thesis “A feasibility study of building Set-top box user interfaces using Scalable Vector Graphics” [7] by Fredrik Vinkvist concludes that SVG is suited to build a STB GUI. Vinkvist creates an application framework that facilitates development and addresses SVG’s shortcomings, such as lack of other layout modes than absolute positioning. However, he never had the opportunity to run the technology on an STB, leaving performance tests undone.

1.6 Equipment

The hardware used is a Motorola VIP 1910 STB. The STB runs an adapted version of the Mozilla 1.7 web browser and the Ekioh SVG renderer[2].

1.7 Set-top box architecture

The architecture of Motorola 1910 STB is described in the diagram below. The parts that are of especial interest in this thesis are the two top most layers. First the web page, where the GUI application resides, and second the web browser in which the GUI application is executed after the web page has been loaded by the web browser. The web page can control STB functionality through a Javascript API exposed by the web browser or plugins in the web browser. The browser and plugins in turn communicate with the STB middleware using C/C++ APIs.

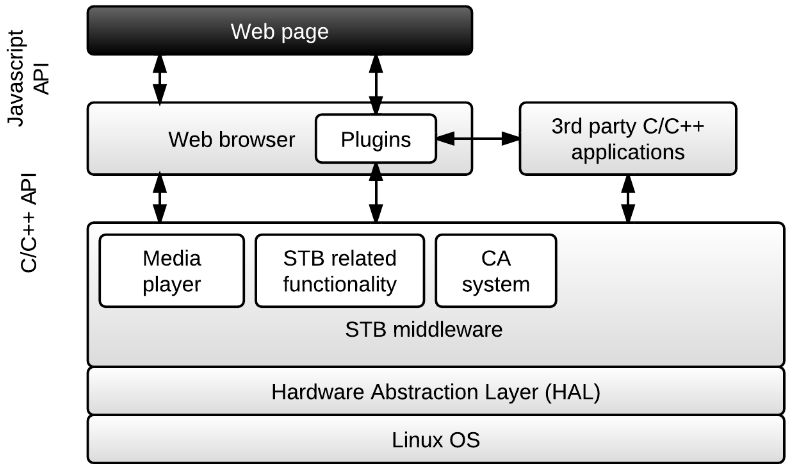

Figure 1-1 Overview of a Motorola STB architecture

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

1.8 Note on performance

It is hard to predict performance of a technology because it depends on many factors. If an application is heavy on computations, performance of the code execution environment is of most importance. If an application is heavy on graphics, performance of the rendering engine is of most importance. Tests can be created to measure different types of performance, such as the Sunspider [8] Javascript benchmark that tests pure Javascript performance, or the GUIMark [9] benchmarks that test the rendering engines of a few technologies.

Results from GUIMark show that both Flash and HTML produce vastly different results on Windows, OSX and Linux, all running on the same device [10]. To make these types of tests meaningful, it is important to have them run on the target device, with the software available on that platform. When performance tests are made in this thesis, only the technologies that currently can run on the STB are tested. This may leave a hole in the comparison, but I find it better than to draw conclusions from unrelated results.

1.9 Terminology and definitions

1.9.1 Graphical user interfaces

In 2004 the Linux Information Project[3] defines a graphical user interface (GUI) as "... a human-computer interface (i.e., a way for humans to interact with computers) that uses windows, icons and menus and which can be manipulated by a mouse (and often to a limited extent by a keyboard as well)." [11]

In the STB case the above definition still holds true with the addition that a remote control is used instead of a mouse.

1.9.2 Web technology

This thesis divides web technologies into two groups; native and plugin based. Server-side technologies are not considered here.

A technology that can run in a web browser is here considered a native web technology. Examples are presentational technologies such as HTML and CSS and the scripting language Javascript.

[3] http://www.linfo.org/

A technology that can run in a web browser by appending a plugin to a web page is here considered a plugin based web technology. Java applets and Adobe Flash are examples.

The main difference between these two is the need for additional software that a plugin based technology needs. There are also security aspects to consider – in this case, where the user interface engine is a web browser, a native web technology is sandboxed to the browser environment while a plugin may have access to other parts of the system.

1.9.3 Browser environment

The browser environment refers to the execution context of a web browser, where the web technology is executed.

Compared to a native environment, which here is considered the operating systems application layer, a technology executed in the browser environment can only access features that the browser exposes, hence not interfere with low level functionality.

Chapter 2
Set-top box and IPTV services study

This chapter seeks an understanding on how IPTV user interfaces and the set-top box hardware have evolved so far and how they will evolve in the next couple of years. The goal is to describe the present situation and provide suggestions of future IPTV services that are relevant for TeliaSonera. The study is mostly based on interviews, which have focused on two areas: set-top box performance and IPTV services.

A historical view over the IPTV set-top box's evolution was created, with CPU performance as main topic. This includes past and present models, as well as the next one or two generation's models. The information will give an understanding on what is possible with coming set-top boxes and can be used to determine how graphical intense the services can be.

Regarding IPTV services, the intension is to understand where the business is heading, to understand what kind of services that are coming and see what impact these will have on the user interface.

The set-top box manufacturers that have been interviewed are Motorola, Tilgin and ADB, all with quite extensive experience in the IPTV industry. An additional interview was performed with Accedo Broadband, a company that provides interactive applications and on-demand content to IPTV platforms.

2.1 Interview with Motorola

At Motorola, Hans Vind was interviewed [12]. He works as senior product specialist and has long experience in the consumer electronics business. He has also previously worked with human-computer interaction.

2.1.1 Set-top box history

Hans Vind explains that the industry's focus from 2005 to 2008 has been on the transition from MPEG-2 to MPEG-4. MPEG-4 enables HD content on the set-top box but requires about four times the data set compared to SD content. Work on encryption has been prioritized and time has been spent on finding approaches to keep channel switch times down. Today, a resolution of 1920x1080i is achieved at 60 fps. The goal is 1920x1080p, which is the Blu-ray standard.

The main CPU had a clock frequency around 300 MHz during the last three to four years. It is first now, when the MPEG-4 problems are basically solved, that focus has moved to increase the performance of the set-top box, Hans Vind continues. The current model, the Motorola IP-STB 1900-9, has a CPU of 266 MHz, two DSPs for video decoding of 400 MHz and 128 MB of RAM of which 32 MB is used for video memory. The next model, to be shipped during 2009 has an improved CPU, faster DSPs and double RAM. Performance is improved even more in future models, with two CPUs and native support for OpenGL and OpenVG. Table 2-1 shows the set-top box history at Motorola.

Hans Vind does not believe that the set-top box will cease to exist. He admits that it probably will be hidden away somewhere but as long as there are large TV operators and content providers that promote themselves by delivering TV content, the set-top box will always prevail. The reason is that otherwise, if the set-top box would be integrated in the television, the end customer could only choose from a limited set of television sets since not all television manufacturers would and could provide support for all TV operators' products. A possibility would be a common interface where the set-top box can be plugged in. Motorola has a working prototype of an IP-STB 1900-9 set-top box that is of the size of a PCMA card. The problem at the moment is that there is no such standard. The Common interface slot that exists on modern TVs does not support video transmission. So for the next five to six years we will have set-top boxes as we know them today. Then we will see, Hans Vind says.

2.1.2 Future IPTV services

Motorola has support for four environments for user interfaces in their set-top boxes. The first approach is using a web browser with HTML and JavaScript to present the user interface. The second is using an SVG renderer with SVG and JavaScript. The third is Adobe Flash, which can be implemented as a standalone user interface renderer or as a web browser plug-in. The last approach is Espial EVO, a proprietary solution which compiles HTML and JavaScript.

Preferably, Motorola likes to support standard technologies, i.e. the browser environment with HTML and JavaScript, SVG and JavaScript or native solutions that operates directly towards the set-top box. Should there be demands from customers about proprietary technologies Motorola will look into it, but they recommend standardized technologies.

The big advantage of the browser environment, according to Hans Vind, is that it uses the same technologies as web services on the Internet, which makes it easy to integrate them with the set-top box. Hans Vind believes that the browser environment is the future just because it allows Web 2.0 services into the set-top box with minimal effort.

He states that the greatest desire at the moment from Motorola's customers is 3D graphics, and efforts are made to include support for that in upcoming set-top boxes. A problem is, though, that the scope of use for 3D graphics is still unclear and the set-top box has far from the computational power that is needed to model a full 3D user interface.

2.2 Interview with Amino (Tilgin)

At Amino Joakim Larsson was interviewed. Joakim is a senior systems architect at Amino (previously at Tilgin) and has 15 years experience in the set-top box industry.

2.2.1 Set-top box history

Tilgin launched its first IPTV set-top box in the year 2000. It was based on x86 architecture with a CPU at 266 MHz. In 2003 they launched its successor Mood 300, now based on a MPIS processor at 324 MHz. For the Mood 300 they implemented graphics acceleration support in a DSP for the purpose of allowing more advanced user interface graphics. But as the industry started to demand support for HD content, which the Mood 300 did not support, it got abandoned for a HD compatible box. In 2005 Mood 400 was released, which is the set-top box that is still used in 2008. It is based on a ST Mirco chipset with a 350 MHz processor and two DSPs of 600 MHz.

The average lifetime for a set-top box, according to Joakim Larsson, is three to four years. During that time the set-top box companies are dependent of that box and cannot afford to release new products every time a new chipset or new demands emerges.

Joakim Larsson says that Tilgin has held discussions with chipset manufacturers regarding increasing performance in the CPUs for a long time. It is just recently that efforts to do so have been made from the chipset manufacturers. The next generation of Tilgin set-top boxes (now labeled Amino), which is set to be release during 2009, will include a slightly better chipset.

2.2.2 Future IPTV services

Tilgin has decided to hold back on investing in new display technologies until the industry seems to have settled for one or a couple of approaches. They have contracted Opera to use their browser as user interface renderer in the meantime.

The next generation set-top box portal, according to Joakim Larsson, will be a widget based system. The operators will provide a portal application with an implemented widget engine. They will also provide standard widgets for services such as EPGs, VoIP and TV content. Third-party providers will then be allowed develop their own widgets for the portal – may it be games, online stores or Web 2.0 services – which will be made available to the users by the operator.

There will probably be a quantity of widget engines based on various technologies, but the overall system will be standardized. Widgets created for a specific engine may be used on all portal systems that implement that widget engine. Joakim Larsson says that this way the set-top box will become a portal to many different services available through the Internet. The system will probably resemble Windows Vista's gadget bar or Mac OS X's Dashboard, and will therefore be familiar to a large part of the end users.

Joakim Larsson presumes that it probably will be a long transition time from a browser environment to a widget-based environment. It is also likely that there will be a widget engine war, and until the winner is known not much can be said about system requirements.

Regarding 3D features, Joakim Larsson just as Hans Vind realizes that today's set-top boxes have insufficient performance to model such a user interface.

2.3 Interview with ADB

At ADB Marcin Pakula was interviewed [13]. Marcin Pakula works as Director of technical marketing at ADB.

2.3.1 Set-top box history

The latest ADB set-top box (as of 2008) is based on the ST Micro 7109 chipset. ADB's approach of building set-top boxes differs somewhat from those of Tilgin and Motorola. According to Marcin Pakula, ADB replaces the microcode from the DSP that handles the audio decoding with code for 3D graphics acceleration to improve performance. To do so they tend to wait a bit longer before embracing a new chipset since they believe that they can get better performance out of it than their competitors.

Marcin believes that the set-top box will get a more prominent role in the future. It will become the central point of the living room, which will provide a universal user interface, through which it would be possible to connect to and control UPnP or DLNA enabled devices in the home.

2.3.2 Future IPTV services

ADB uses the browser environment as a platform for their user interfaces. Marcin Pakula is of the opinion that a web browser that renders HTML and JavaScript is sufficient for creating a set-top box user interface. At the moment they use Mozilla Firefox. On the other hand, he notes that there is a problem with the browser solution because there are no standardized APIs; they depend on both the set-top box vendors and the browser vendors. Marcin Pakula realizes that there is a need for a standard here and feels that the Open IPTV Forum's initiative to create such a standard is the right way to go.

He sees two major functions for the set-top box in the near future: home networking and Internet content. In the first case the set-top box could act as the central point in the living room. Through it one could access all UPnP and DLNA enabled devices connected to the home network, such as DVD players, cameras, computers etc. He claims that the technology exists today, but what is lacking is proper user interfaces. Because the number of devices connected to the home network may be very large, it is likely to become a problem for the user to find the media he is looking for. Most of the devices have their own internal catalogue structure and having to browse through all devices when looking for the media will be cumbersome and very time consuming. He says that there is need for an application that solves this problem and feels that this is an area that has been given too little attention.

Regarding viewing Internet content in the set-top box, he claims that we do not want to use the set-top box for browsing the Internet in the way we are used to in the PC environment since there is no keyboard or mouse available. What we do want, however, is having the set-top box presenting the content of the Internet to us in a way adapted for the big screen and the remote control. ADB is looking at different possibilities to bring Web 2.0 services like Facebook and YouTube into the set-top box.

Marcin Pakula then explains that in many countries there are a lot of information generated by communities about local traffic, weather and such. To be able to present this information through the TV, using a simple user interface, is the future of television. When the user has access to these communities, sharing of content is the next step and the users would then want to upload data to the communities. The technology exists today, he says, but has not been addressed properly by middleware providers or operators.

2.4 Interview with Accedo Broadband

Ted Björling was interviewed at Accedo Broadband. He is Manager of technology and platforms and has experience since the start-up of the company in 2004.

Accedo Broadband is mainly focusing on games for IPTV. Their product portfolio ranges from simple soduku games to more advanced multiplayer games. Most of their applications are based on HTML and JavaScript, but they have some Flash applications as well. Other technologies they are looking into are SVG, Espial EVO and Microsoft Mediaroom. Accedo Broadband investigates new technologies by customer demands and currently, according to Ted Björling, the biggest hype is around SVG.

He believes that community created applications and services are to come to the TV and gives Apple's AppStore as an inspirational model. An example could be an application that fetches information from Wikipedia about the current TV-show and presents the information to the user. Where this application will reside is yet to be determined, Ted Björling says. It could be the set-top box vendor that provides it, middleware vendor or a third-party vendor.

Accedo has done some research about Yahoo's Connected TV and entered a partnership with Yahoo. The idea behind this type of concept is that anyone should be able to create an application for the TV. The television experience will be more interactive if e.g. web services and games get connected to the TV. This is not a unique concept and most CE-vendors are working on similar systems. He thinks, though, that this concept takes a step away from the traditional TV operators since it provides many alternative ways for the user to get TV content. But the end users' expectations on content and services will increase as they get more accustomed to interactive TV services and these expectations must be met in order to keep the customers.

Accedo Broadband's vision is to follow the trends in the business and provide cutting edge solutions. Their main focus areas will be multiplayer games and Web 2.0 services. They have already, according to Ted Björling, made reference implementations that integrate against some Web 2.0 services. He believes that the time to market for these types of IPTV services is between one to three years. It is happening now, he says.

2.5 Open IPTV Forum services

The Open IPTV Forum has the purpose to produce end to end specifications for IPTV. The members come from all business areas associated with IPTV – network operators, content providers, service providers, CE manufacturers and network infrastructure providers [14]. Among these is TeliaSonera.

The Open IPTV Forum has decided upon using a special purpose web browser called CEA-2014 [15], developed by the Consumer Electronics Association. It uses CE-HTML, an extension to XHTML, and ECMAScript to render the user interface [16] and it has standardized APIs for communication with the IPTV devices [16].

The organization has produced a document that summarizes selected services and functions for its second release [17]. Some of those services are of interest for this chapter.

Among the services is PVR functionality with local or network based storage. The Open IPTV Forum suggests that the user should be able to remotely manage content and scheduling using a mobile device. A content guide should present the user with a list tailored to user preferences of scheduled, on demand and recorded content. A personalized channel service is also suggested, which would let the user create his own program line up based on his preferences, habits or the service provider's recommendations.

The Open IPTV Forum also suggests integration of IPTV with communication services such as voice and video telephony and chat with presence state indication. It is also suggested that it should be possible to deliver different components of the media stream to different devices. For instance could the audio stream be delivered to the mobile phone and the video stream to the television screen.

Moreover, it is suggested that interactive applications shall be allowed. These are defined as "applications that allow user interaction via the ITF[4] device or other user devices". This includes applications that are network-based and interact with the IPTV terminal device using web technologies, but also applications that reside in the home network. Mobile devices shall also be able to interact with an application connected to an IPTV service. An example could be to remotely schedule and manage the PVR with a mobile device, as suggested when the PVR functionality was described above.

Home networking is another service that is specified. It states that the IPTV user shall be able to access content stored on DLNA devices connected to the home network and also that IPTV content shall be offered to these devices. The user shall also be able to share content with other users, provided that the DRM policy allows it.

[4] IPTV Terminal Function

2.6 Summary

2.6.1 Set-top box survey

All interviews stated that the set-top boxes have not evolved so much considering CPU capacity during the IPTV-area. All focus has been on transitioning into the MPEG-4 standard that mainly concerns video decoding and is not handled by the CPU. According to all of the interviewees, a change in the industry is happening now. Focus is moved over to CPU performance and in the next models the CPU clock frequencies and the main memory are about doubled. The major set-top box chipset manufacturers are announcing new chipsets which confirms the change [18] [19].

The main chipset manufacturers used for IPTV set-top boxes are ST Microelectronics, Sigma Designs and Broadcom. Recently (late 2008) Intel decided to enter the CE device market by introducing a new chipset, based on the Intel x86 architecture [20].

The next generation chipsets from the some of the biggest players in the industry are listed in Table 2-3. Table 2-1 and Table 2-2 summarize the set-top box history at Motorola and Tilgin.

Table over Motorola set-top boxes	
Model	*Chipset*
IP-STB 400	National Semiconductor G1 (CPU) Sigma Design 7400 (video decoder)
IP-STB 500	National Semiconductor Geode (CPU) Sigma Design 8401 (video decoder)
IP-STB 700	National Semiconductor Geode (CPU) Sigma Design 8405 (video decoder)
IP-STB 1500	ATI Xilleon 210D (comb. unit)
IP-STB 1900-9	ST Micro 7109 (comb. unit) Main CPU: 266 Mhz
IP-STB 19x3 (2009)	ST Micro 7105 (comb. unit) Main CPU: 450 Mhz

Next model (2010-2011)	Main CPUs: ~600 MHz x 2 Built-in support for OpenGL and OpenVG.

Table 2-1 Specifications for the Motorola set-top boxes

Table over Tilgin set-top boxes	
Model	*Chipset*
Mood 200	x86 266 MHz
Mood 300	MIPS 324 MHz
Mood 100	STMicro 200 MHz
Mood 400	STMicro 350 MHz
Next model (2009)	Improved CPU ~450 MHz

Table 2-2 Specifications for the Tilgin set-top boxes

Table over next generation chipsets		
Manufacturer	*Chipset*	*CPU*
STMicro	STi7105	ST40 450 MHz
Broadcom	BCM7405	MIPS 400 MHz
SigmaDesign	SMP8640	MIPS 333 MHz
Intel	CE 3100	Pentium M 800+ MHz

Table 2-3 Next generation's chipsets form some of the major CE chipset manufacturers.

2.6.2 IPTV services

Five major types of services have been identified based on the interviews and the Open IPTV Forum.

Internet content In common for all the interviews is the belief that Web 2.0 services will play a big part in the future of IPTV. The Open IPTV Forum is supporting this approach in their suggestion about interactive applications. It also suggests that users should be able to communicate with each other through video conferencing and chat.

Third-party applications Accedo Broadband suggests community created applications in the likes of the Apple AppStore concept. Tilgin believes that third-party applications will be big when the set-top box portal becomes widget based.

Widget system Both Tilgin and Accedo Broadband suggest a widget system. The widget system is an application environment that will open up the set-top box to small standardized applications, so called widgets. The purpose is that these widgets can be created by anyone with access to the API for that system and that the widgets run in a plug-n-play manner when deployed in the system. This type of system does not only target the set-top box, but all types of consumer electronics that are connected to a TV. Both companies mention that Intel and Yahoo have come quite far on developing such a system.

Home networking The Open IPTV Forum wants to allow the user to access content of other DLNA devices in the home network and vice versa. ADB also suggests this approach.

Mobile devices The Open IPTV Forum suggests that services should be accessible from a mobile device. It also suggests that it should be possible to deliver different parts of the media stream to different devices. These two facts give the opportunity to extend the IPTV experience from one screen into many.

2.6.3 User interface technology

All three set-top box companies have chosen a browser based approach with HTML and JavaScript as user interface technology. Motorola claims that the browser environment is the best way to go since it uses the same technology as the Internet services and will allow for fast integration. ADB and Tilgin are of the same opinion and have both chosen web browsers user interface platforms. Motorola recommends SVG or HTML meanwhile ADB and Tilgin sticks with HTML for now. The Open IPTV Forum has chosen a similar approach with its CEA-2014 browser.

There is, according to Motorola, a desire for 3D graphics. Motorola is to support OpenGL in its future set-top boxes and ADB has implemented some 3D acceleration in its set-top boxes. Both companies seem to realize that there is some confusion about what the 3D graphics purpose is.

Chapter 3
Technology study

This chapter aims to document what web technologies could be of interest for a browser based set-top box GUI. It uses the criteria from section 1.3 Criteria and scope as a base when describing the technologies.

3.1 Criteria

Each criterion is ranked by importance on a scale of 1 to 5. The rank has been set after discussions with the IPTV group at TeliaSonera. The criteria are defined more thoroughly and a set of focus questions has been developed for each criterion. See Table 3-1 for definitions and questions.

Performance The performance of a web technology depends on both the hardware of the device and the runtime environment in which the technology is run. *Weight: 5.*

Platform independence For the same technology to work seamlessly on different devices, it needs to be able to run on different operating system without having to consider the devices specific hardware configurations. *Weight: 4.*

Resolution independence The user interface should render with good quality independent of the screen resolution. *Weight: 3.*

Rich user experience Clear separation between graphics and code is needed. It should be possible to create or change the graphical components of the user interface without altering the code and vice versa. The technology should support rich graphical effect, such as animations and transitions. W*eight: 3.*

Development support Resources in forms of tools and development communities available for the technology. *Weight: 2.*

Table of criteria for the technology study	
PERFORMANCE, WEIGHT 5	
Script performance	How well does the technology perform on the target platform?
Graphics rendering	How well/fast does the technology render on the target platform?
PLATFORM INDEPENDENCE, WEIGHT 4	
Runtime environment	What is the run time environment for the technology? Does the runtime exists for different browsers?
OS support	What OS's does the technology support?
RESOLUTION INDEPENDENCE, WEIGHT 3	
Graphics format	Does the technology support vector graphics?
RICH USER EXPERIENCE, WEIGHT 3	
Skinnability	How does the technology handle separation of graphics and program code to allow easy skinning of the user interface?
Graphical effects	Does the technology support animation and other graphical effects?
DEVELOPMENT SUPPORT, WEIGHT 2	
Development community	Does the technology have a big development community? Is it easy to find information and support?
Tools	Do good tools and frameworks exist?

Table 3-1 Definitions of the criteria used for the technology investigations.

3.2 Web technologies

3.2.1 HTML

HTML version 4.01 is the current W3C recommendation and has been so since 1999. [21](chapter 2.3). The vision when developing HTML has been that all kind of devices should be able to access information on the web, and that HTML documents should be platform and device independent [21](chapter 2.2.1).

HTML has support for text, bitmap images and various layout elements [21](chapter 14.1). HTML is close coupled with other W3C specifications; an HTML document can be manipulated with Javascript by modifying the DOM [22] and the appearance of HTML elements can be change by including CSS style sheets [23].

There are numerous Javascript libraries that provide utility functionality for DOM manipulation, animation, cross-browser issues and more [24].

HTML5

HTML5 is the latest draft of HTML and extends and refines the language by adding some elements, removing and altering other. Among the most notably are the audio and video elements – allowing multimedia content – and the canvas element (see 3.2.2 HTML5 Canvas). HTML5 is often referred to as a collection of technologies, including various newer CSS specifications (such as media queries [25], the CSS images [26] and CSS Transitions [27]) that allow for more control over user interface styling (gradients, animation etc). New Javascript APIs, such as the Websockets API [28] and the File system API [29] are also considered part of HTML5, as well as WebGL [30] and more[5].

Support for HTML5 features varies between different browsers and versions [31]. To get a full picture an analysis of each feature is needed.

3.2.2 HTML5 Canvas

Canvas is an HTML element that represents a resolution dependent bitmap on which it is possible to draw graphics using JavaScript. Graphics may be constructed using a script, thus allowing for dynamic rendering and redrawing, or by inclusion of bitmaps. The canvas API does support text, but only when drawn on one line or along a path (not as a text area). When graphics are updated in a canvas-element the whole element is redrawn on the screen [32](chapter 4.8.11).

The WebGL specification allows for 3D graphics to be drawn on a canvas element [30].

To facilitate developing on the canvas element third party Javascript libraries have appeared that adds functionality on top of the Canvas API. For example, paper.js [33] adds a scene-graph, a DOM and utility functions to draw vector graphics.

The canvas element is supported by newer versions of all major browsers [31].

[5] http://www.html5rocks.com/en/

3.2.3 Scalable vector graphics

SVG is and an XML-based language that aims to provide rich graphics content. It describes two-dimensional vector graphics and text with support for animation, video and audio [34]. It is a W3C standard and currently exists in three different profiles (W3C recommendations) that aim different devices: SVG Full 1.1, SVG Tiny 1.2 and SVG Mobile 1.1 where the two latter are subsets of the first [34].

SVG is compatible with other W3C standards, i.e. it is as already mentioned based on XML, but it can also be used with a subset of CSS for styling, SMIL for animation and has a DOM for document manipulation with ECMAScript/Javascript [34].

As with HTML, SVG is not coupled with a runtime environment, but is dependent on browser makers to integrate native support for it. Browsers are however getting better support for SVG with recent versions, but there are still inconsistencies in the implementations [35].

SVG graphics can be authored in vector graphics editors, such as Inkscape[6]. A few Javascript libraries exist, e.g. Raphaël[7], which facilitates programmatic manipulation of the SVG elements.

3.2.4 JavaFX

JavaFX is according to Oracle their means to improve Java as a rich client platform. The company describes it as “a lightweight, hardware-accelerated Java UI platform for enterprise business applications” [36]. It supports hardware accelerated graphics, audio, video and embedding of HTML documents (using Webkit render) [37].

By reading the documentation [38] it is a bit unclear how JavaFX handles resolution independence. It seems that some components, UI controls and layouts are resizable, while other, such as text and shapes are not [39].

The technology supports separation of code and graphics, by the FXML markup language [40] and a CSS3 subset [41].

Since JavaFX uses the Java language it has a large set of tool to aid development. Oracle also provides some tools specially aimed at JavaFX[8]

[6] http://inkscape.org/

[7] http://raphaeljs.com/

[8] JavaFX tools: http://www.oracle.com/technetwork/java/javafx/tools/index.html

The JavaFX runtime is installed with the Java Runtime Environment [36], and has currently support for Windows with Mac OSX and Linux on the roadmap [42].

3.2.5 Adobe Shockwave Flash

Adobe Flash is a collection of technologies for creating multimedia content. It consists of the swf format, which is a binary format and is used to represent vector graphics, text, video and sound. Flash uses a scripting language called ActionScript (an implementation of the ECMAScript) to add interactivity and animation to the swf-files [43].

To display the swf-files a player such as Adobe Flash Player is required. It is the industry standard and according to Adobe it has a market share of 98 per cent and up, depending on market and version of the player [44]. The Flash Player is installed as a plugin to the web browser. Flash applications can also run outside of the browser, in the runtime environment Adobe Air[9].

Adobe provides tools and frameworks for authoring flash applications[10]. There also exist open source tools, such as the editor Flash develop[11].

3.2.6 Microsoft Silverlight

Silverlight is a cross platform implementation of Microsoft's .NET framework with focus on rich interactive applications and can be run as a plug-in to a web browser or as a native application [45]. It based on vector graphics [46] and supports video, audio, text and graphical effects (such as animations) [45].

It uses a subset of the WPF[12] for user interface elements and the markup language XAML[13] to define the elements [45]. A Silverlight application can either use a Javascript API with Javascript code executed in the browser, or use a managed API with languages such as C# or IronRuby running in Silverlight's runtime environment [47].

[9] Adobe AIR is a system runtime that allows to run Flash outside of the browser (https://www.adobe.com/products/air/faq.html)

[10] Flash tools: http://www.adobe.com/flashplatform/

[11] http://www.flashdevelop.org/

[12] Windows Presentation Foundation (http://msdn.microsoft.com/en-us/library/ms754130.aspx)

[13] Extensible Application Markup Language (http://msdn.microsoft.com/en-us/library/ms752059.aspx)

Microsoft provides authoring tools for Silverlight in their Visual Studio product[14].

Silverlight supports Windows and Mac OSX [48]. Linux is supported by the open source Moonlight[15] project. However, the Moonlight project does not keep up to speed with the Silverlight releases, currently being three versions behind [49].

[14] Silverlight tools: http://www.silverlight.net/downloads

[15] http://www.mono-project.com/