WebC-iTV - Internet TV in Websites

Nuno Magalhães

Abstract. The following document consists in an extended summary of a Dissertation that studies the Internet Television paradigm. It covers the state of the art of this paradigm, analyzes some existing systems, and presents a solution by the development of a framework that has two basic purposes: to allow a producer to manage resources in order to create playlists, and to create a GUI environment for the public to access the content of those created playlists. This project is developed and deployed over an existing web platform, created by SIQuant, called WebComfort. Parallel to the Internet Television paradigm, this document also analyzes the Interactive Video paradigm, although in a superficial manner as it’s a secondary part of this project.

Keywords: Channel, Playlist, Internet-Television (iTV), Resource Management

1 Introduction

With the exponential growth of web technologies and physical resources, devices that used analog systems became rapidly obsolete giving place to the digital ones. Televisions, radios, cameras, frames, among others, experienced a radical change of its inner technology and hardware in order to adopt this new fundamental requirement, that is, to use digital systems.

Digital TV (DTV) is one of the products that born in consequence of this changing and one of its main effects was that the TV (media) content could be authored and managed by computers prior to its broadcasting[1].

Meanwhile, Internet became popular and widely used resulting also in various transformations in the computer science area. One of these was the fact that private (media) resources could easily become public ones as they could be shared over the web. Also, with the expansion of Internet, some tv companies began to try some experiments by broadcasting television over the Internet [9].

These two new technologies put TV in the next step of evolution: the Internet Television (iTV).

As iTV gained popularity, its technologies growth and it was created an innovative way of seeing and interacting with television content: the interactive video. This new paradigm turned advertising more appealing and easier, giving birth to a new tendency in iTV: the impulsive purchase (or impulsive buying). Although it previously existed in other contexts, this kind of purchase, in which a user (spontaneously) buys a product without planning it[2], became popular in the iTV context as it allows a user to purchase a product while seeing its broadcasted advertising movie. By clicking on the movie (or a specific object on the movie), an event is triggered provoking the appearance of metadata about the product in focus appears (like the website where it can be bought, its price, etc.).

1.1 Synonyms for iTV[3]

iTV stands both for Internet-Television or Interactive-Television. This late definition will be explained later and it refers to another paradigm which is frequently used in Internet-Television, Interactive-Television.

Other terms are used when referring to this subject. They are: ITV, i-TV, WebTV, Online TV, Network TV, Broadband TV, Web-based TV, Web-enabled TVs, Net TV InternetTV, Broadband ITV and IPTV.

1.2 Interactive TV and Interactive TV

Internet Television (also commonly known as IPTV) refers to television content that’s placed in an internet repository that can be later broadcasted any device that has Internet access (either if it's a computer with a browser or a set-top box) [1].

---

Interactive-TV refers to the fact that a user might interact with the broadcasted content in various ways: voting programs, teletext, video-on-demand (VoD) [2]. Internet-TV is a case of Interactive-TV since it (normally) allows some level of interaction in its contents.

The scope of this dissertation, regarding interactivity, is the Interactive Video. In this context, Interactive Video is defined as the clicking act over a video in order to obtain information about the content that is being streamed. Normally, it’s used in advertising: a product is being presented and the user might click over the video to view information about that product or even to access a specific website to make the purchase. It can be generalized to other areas such as sports. In this case, a user can click over a sport match (or a player in highlights) in order to see information about it.

1.3 WebComfort platform

It was said before that the project will be delivered and deployed in the WebComfort platform. It’s a Content Management System (CMS) developed in Microsoft ASP.NET 2.0 and it’s available for computer clients (such as internet browsers) and also for mobile clients (cell phones and PDAs, for example).

One of its main purposes is to allow management of web content – not only content in a web application but also content regarding cooperation between various web applications.

All the WebComfort’s features are divided in Modules [3] which can be deployed (and removed) in specific sections. This allows a WebComfort’s webpage to have multiple Modules, or features, running in the same page without having to open a page for each one.

Lastly, this platform provides a visual interface that is normalized. When a project is deployed, it must import the master page and CSS classes already defined [4], so that it can be viewed properly and accordingly to the rest of the WebComfort’s native Modules.

In WebComfort, all pages are divided into tabs which are created and managed by the user. These tabs allows a programmer to deploy various modules into different containers according to its needs.

WebComfort's library has a great amount of interfaces that can be easily used. Just as examples, it's provided a Language API which allows any module (and pages) to have multiple languages easily implemented; there's also Tabs API and Modules API which allows to manipulate the modules in a page and also manage the Tabs. This project is will be developed using WebComfort’s existing features and according to its normalized interface and visual appearance.

1.4 User, producer and viewer

During this dissertation, the terms “user”, “producer” and “viewer” are going to be used very often. Producer is the person that creates and owns a channel and is responsible for its content management. A viewer is the person that accesses the channel to only view the movies its playlists. A user is any person that is using a system disregarding if it is a producer or a viewer since it can be both of them. This last entity is used in systems like YouTube, where a registered user can create a channel but also view content of other users. There’s almost no distinction between a producer and a viewer in this perspective. The scope of this solution, however, focus more on the producer, which is also a user. The viewer has not any representation in the implementation of this project but it's present in this document as an abstract entity which can see the final product – a channel.

---

[4] [http://www.webcomfort.org/portal/Home@1.aspx](http://www.webcomfort.org/portal/Home@1.aspx)
1.5 Objectives

The objective of this project is to create a framework over WebComfort that will become possible for a company’s producer to create an online channel thus allowing a viewer to watch that channel in the web. It will be analyzed the state of the art in the iTV features and elaborated a theoretical solution for this paradigm. This analysis will take into account existing systems and their evaluation. It will be taken into account characteristics such as content management, available features and the GUI.

With this project, a company’s producer will be able to: manage channels, playlists and resources and insert hidden advertisement hyperlinks.

As a direct result from this previous objective, a viewer will be able to navigate in the channel, see the playlist, see the live emission and to click in the resource that's being transmitted in order to navigate to another site (which is decided by the producer).

Another objective of this document is to provide a simple solution for the Interactive Video paradigm. As Interactive Video is also a big subject that is passive of a dissertation for its own, the solution will only deal with the mouse interaction over a whole movie. This objective is present in this document since it's one of the issues that iTV should treat.

2 Reference model and desirable features

By studying and comparing various existing systems that implement iTV, a list of features and characteristics was created and they are mandatory to solve the iTV problems. This list is called the reference model. They are:

- **Channel creator:** a channel is more an identity than anything else. If playlists are scattered in websites or web portals, there’s no identity since these playlists are not connected between them. For a producer to individualize the company’s mission and objectives about iTV, it must have a channel. This way, a viewer can access the playlists, expecting to see a certain type of content that is according to its preferences.

- **Resource management:** a playlist is constituted by various resource entries. These resources must, somehow, be managed so that they can be properly used in the playlist. The resource management requirement also includes other features like playlist management, channel management and everything else that has to be managed by the producer.

- **Playlist creator:** for the iTV paradigm to work as it’s supposed, it must allow the creation of playlists (that, normally, are changed in daily basis), so that viewers can access different resources at different times of the day.

- **Resource importation:** the resources can be stored on a local disk or somewhere in a remote disk for example, on a portal like YouTube or MySpace. It's important that a iTV system allows the importation of resources not only from local disks but also from remote places including portals.

From this model, it can be concluded that are three main entities: channel, playlists and resource. Without any of them the iTV paradigm cannot be solved or approached since it would not allow the creation of an online channel.

Adding to the reference model are some desirable features that are important for the resolution of some of the issues concerning iTV. These features can be grouped by entity:

- **Channel:** manage channel's information, configure the channel's logic and visual.

- **Playlist:** change playlist's date, define the opening time of the playlist, allow the creation of playlist entries with specific associated metadata.

- **Resource:** manage resource's information, have access to resource's metadata, have filters to view only the desirable resources. Apart from these, Video interactivity one of the most desirable feature because the iTV paradigm becomes even more useful when adding some interaction in the broadcasted content. This interaction is important as it changes the role of the viewer from mere spectator to actor or participant.

3 Related work

Nowadays, there are a great number of systems that cover many features of the iTV paradigm. These systems offer a user a large number of different features that goes from TV companies that provide its broadcast content in the Internet for free public access to systems that allow any user to freely create its channel and control its broadcasted content.

Next, it will be analyzed some existing systems that give a solution for the iTV paradigm.
3.1 Existing systems

There are many systems that partially or totally fulfill the iTV requirements. Some of these systems are also used in IPTV. This section will only show a few examples and some features of these systems. They are:

- VideoClix: it’s a platform that enables a producer to create interactive.
- WorldTV: this is one platform that allows the creation of a channel with a playlist. It’s a good starting point for the study of the iTV paradigm and will also be studied later on, although it’s not a complete option. WorldTV is similar to many other systems that are listed in Annex 2.
- YouTube: it’s one of the most complete platforms in terms of iTV, as it also provides Interactive Video authoring. This is another system that will be studied in this document and it’s the most important one.
- Castter: in Portugal, this is a reference platform for iTV since it allows for a common person to literally become a channel’s producer and manager. Many organizations use this platform to create its own channels. Two examples are Querqus and Greenpeace, both environmental organizations.
- Kyte: it’s much like WorldTV. It allows the creation of shows (it represents playlists) and the media content is inserted in the channel via hard drive or webcam.
- WoMa: a website dedicated to create channels with videos only about martial arts. This system will be studied in the thesis.
- Icareus: it’s a system of IPTV that could, in some points, be used in iTV. This is a fully featured platform with several modules that goes from pre-processing media content to the broadcasting feature.

3.2 YouTube

Although mostly known as an online video sharing portal, YouTube’s programmers implemented functions that allow its users to create a channel with all its required features.

The first limitation to be stated is in the creation of a channel. Like WoMa, in YouTube each user has only a single channel. If someone wants to create more than one channel it has to create a new YouTube account in order to do it. This represents a big limitation as it makes channel management almost unfeasible. If a user wants to work with multiple channels it must have each channel opened in different internet browsers as each browser will allow only a single login, hence a single user.

Despite this limitation, YouTube is still a good choice for a producer since its advantages outweigh this disadvantage. For instance, the playlist creator is quite complete as it allows the creation of two different kinds of playlists: various normal playlist and a single fast playlist. The main difference resides on the fact that the fast playlist allows a user to see a movie in YouTube and add it to this temporary playlist so that it can manage the video later. This playlist’s main purpose is to allow a user to now waste time thinking of where to put a movie or even to allow a user to save a movie to see it later. The other playlists are normal ones and there’s no limit on their number – a user can create as many as it wants.

In terms of resource management, YouTube is a five star choice for any producer. Its interface is easy to use and has many features that make the creation of a channel in a whole easier for any user. The media content can be uploaded from the hard disk and it can be of almost any kind. If it’s the will of the user, the video can also be uploaded via a webcam. A playlist can be composed with both videos already published by other users and videos published the user itself. Playlist management is also quite intuitive: a user can decide, first of all, the order of the movies inside each playlist and, later, the order of the playlists in the channel.

As YouTube is not supposed to be a professional channel platform, it does not allow live content or strong interactive features. The only interactivity that YouTube provides consists in annotations and buttons to links (which is more and less what is wanted to this project). Before publishing a video, a user can add features to the video such as this last described. It can also add information to the movie and subtitles.

Each movie has associated a statistics tool. It provides the normal information such as the number of viewers, but it can also provide much complex queries like filter the viewers by region.

The final feature that YouTube has that’s desirable to this project is the Multilingual support. On the bottom of each page, a user can select one from nineteen different languages. One problem with this feature is the fact that is somehow hidden as it only appears in the end of the website and in little letters.

YouTube is presented in this section has a reference example of the project to be developed since it’s one of the most complete and free systems. As it was stated, this platform provides features as:

- Creation and management of a single channel;
- Creation and management of various playlists;
- Video authoring (in terms of adding notes and buttons);
- Statistics tools;
- Multilingual support.
One of YouTube’s few disadvantages, in this specific context, is that it’s not liable to be used by a TV company because of the fact that the site has many users and channels and the creation of one more channel would be a tear drop in a big ocean, not fulfilling most companies’ objectives (such as, reach a certain audience). But it’s free and fully featured and it comprises the whole required and desired features for this project.

4 Solution

The theme followed by the prototype was early abandoned replacing blue based colors to gray based colors in memory of the early cinema resources which followed this pattern.

Three WebComfort tabs exist by default: the main tab, called “Principal” which is the main homepage, the authoring tab, called “Edição” which is the core of the project and the WebComfort's administration tab called “Administração do Portal”. The other existing tabs consist on the created channels. A viewer can only access to the main and channels tabs as they are the only ones with public access.

On the edition's tab, there are three containers. The left container consists in the modules that manage channels, playlists and language support. With the exception of the last module (which was already implemented in WebComfort), the first two have the search option whenever there's data saved in the project's database. This searching option allows the producer to filter the content thus providing a faster way of navigating in large amounts of different data.

In the channel's module, two kinds of search are provided: simple search by channel's name and advanced search by channel's features. The later provides a search by name, genre and main language.

The playlist's module only provides a search by date in which the producer can search a playlist according to its date. This search has numerous combinations as it can filter a whole date or filters dates by days, months, years or combinations of these three, according to what it's typed in the boxes. It's assumed that a null box corresponds to any value (thus, the default null date shows every playlist in the corresponding channel).

The middle container shows the entries of a selected playlist. It's merely informative as nothing can be changed in this module called PlaylistPreview. This table contains two different durations: the first corresponds to the resource's total time, the second corresponds to when the resource will finish in real time.

The final container is reserved to the resources' entity. It has a search engine that only allows to search by name of by location because in this context the main purpose of showing the resources is to allow the producer to visualize which ones are already in the database. A more complete search engine is provided in other context where the producer creates a playlist and needs more information as the duration of the resource, for instance.

Each module can be edited to change the preferences. These preferences consist in letting the producer to choose which columns of each table are shown or hidden.

The preferences option helps the producer to hide unimportant details that would turn the tables more confusing and of difficult handling.

4.1 Authoring Modules/Pages

The previous section described the common features of all modules, with the exception of module PlaylistPreview. In this section and in the next, it will be explained in detail how each module and page work.

The modules have a common function: to show and delete the entries of each table (with the same exception of PlaylistPreview). Add and edit options are available but are just serve to pass arguments that are treated in the modules' associated pages. For this reason, each module has a page which contains all the important operations of creating the entities that are available in the same module.

• Channel Module/Page

The project's channels are created in Channel (module) and ChannelEditor (page). The module shows every existing channel according to the given filter (by default every channel is shown). If a channel is created or edited, the associated page opens providing the desired editing features.

The name of the channel cannot be changed and cannot be repeated. Otherwise it would be difficult to distinguish between two different channels with the same name and in real world there is no company with two channels of the same name.

Although not probable nor common, a channel might change its genre or language. These features are also available for changing.
The logotype of a channel is changed from time to time. Therefore it is possible, at any time, to change channel's logo according to necessity. It's also provided a preview feature to help the producer to see the final result of the appliance of the selected image.

Lastly, the channel's tabs edition allows the producer to make navigation tabs that are automatically created when the channel is created. The names of each tab are also unique inside a channel and their content is variable. The producer decides whether the tab appears in the homepage or not.

When the producer selects the option of creating a new channel, a new WebComfort tab is automatically created. This tab has the channel's name and the common modules are automatically loaded. All tabs that were appointed to appear in the homepage are loaded. Later, this topic will be explained in its appropriate context with more detail.

At the bottom of this web-page, there's a list of all existing channels. This allows the producer to edit another channel without leaving this context.

- **Playlist Module/Page**

After the creation of a channel, a producer creates daily playlists to broadcast shows, movies, news, etc. The PlaylistSelect module is responsible for the creation of sequential playlists. If no playlist exists, it creates a playlist with the current date, else it creates a playlist with the date in the next day of the bigger date existing in the playlists of the company's database. This is the only module that creates entities (in this case, playlists) although it creates empty instances, that is, playlists without entries.

In order to edit the playlists, the producer selects the desired playlist and edits it in the corresponding associated page. This page represents almost all of the effort in this project as it occupied more than 50% of the work done in the whole solution. It provides multiple filters and features for the resources to help the producer to select the desirable ones.

Each entry of a playlist consists of a resource associated to a specific (and mandatory) name. It has a time duration, and ends at a specific hour. If a set of entries has the same name, the solution interprets it as a program. In almost every channel, some programs are repeated a certain number of times. To avoid repeating the creation of the same program, which would waste time, and to prevent the producer to commit mistakes in this repetition, the feature of importing programs is available. It allows the importation of any program saved in the database regardless where it's located. As there can be thousands of programs in an universe of channels and playlists, the producer must specify the channel, the playlist and the name of the program, by this order, to view the resources that were used in this program.

It was focused before that more complete filters for the resources were introduced in this page. This page also allows the filtering of resources by name and location but, as a playlist is being created and time is an invaluable attribute to decide when a resource its played and how much time spends, there's a filter for the minimum and maximum duration of a resource.

Sometimes, a producer might want to create a playlist with resources of a certain theme. To help in the decision of the appropriated resources, there's a filter by topic and keyword, which allows the user to see all resources related with this criteria.

Each set of resources can be insert on top of the existing list or after a selected resource. When this insertion is made all resources update their ending times (with the exception of an insertion at the end of the existing playlist). The same update happens whenever a resource (that's not the last one) is deleted.

Nowadays, almost every channel broadcasts twenty four over twenty four hours and seven days a week, continuously. But some channels with specific themes, such as channels that broadcast debates at a parliament, have a limited schedule. In this example, the playlist doesn't start at 0:00:00 of the current day as it's not business hours. To solve problems similar to this, there's an option that allows a producer to change the starting time of the playlist.

As already stated, the playlist has a date. If the producer wants to change the current playlist's date, it can do that in the appropriate field but it must not match an existing date as it would not be logical to have two playlists broadcasting different programs at the same time in the same channel.

Lastly, an entry might have a link associated. This represents the simple video interactivity which allows a viewer to access an outside website related to the resources that's being broadcast. It's useful specially during the commercials because it helps the viewer to have more information about a certain product or, if it's possible to do it, buy it online.
• **Resource Module/Page**

The last of the Module-Page pair regards the last entity of the solution – resource. Similar to the Channel module, this module doesn't allow creation of any entities. It shows the existing resources but doesn't creates instances of them. For the creation of a new resource the producer selects the appropriate option.

A resource is defined by its location and, therefore, can't be added twice. Any resource, either from the local disk or from a remote location, is saved on a specific folder in the local disk for the extraction of metadata concerning the resource. If the resource is remote, the producer might not want to keep it on its local disk after acquiring the necessary information. In that case, the checkbox that asks about saving the temporary file remains unchecked. Otherwise, the box is can be checked and the resource remains on the local disk.

Any relevant keyword can stochastically be associated to a resource to ease the filtering (already explained). These keywords can be about anything in the resource or related to it. But to define the keyword in an ordered manner, the producer adds topics to it. As previously detailed, these topics have a specific hierarchy defined by the producer.

• **Playlist Preview Module**

This is the only module that only shows information and does not allow any modification in the database.

The module aims to help the producer to keep track of what the playlists have without the need to open the page regarding the edition of these. In order to alter its entries, the producer must open the playlist editor.

4.2 **Channel Modules**

While the previous section gave an explanation about the modules responsible for the creation of the entities, this section covers the modules that appear in the created channels. These are not authoring modules – they just provide information and data. They can be seen as a validation of the previous modules because these show in action the final product: a functional channel with resources to be broadcast.

On the left of a channel's homepage it's located the ChannelMenu module which loads the logotype of the channel and its common and created tabs. By selecting a tab, a viewer can see its content on the container in the right side. However, the Playlist module is always present as a viewer might be navigating on the channel and, at the same time, be informed about the current broadcast emission.

The left container hosts all the modules that show the information about the tabs on the right. On the top is the playlist in its simple view – a view that contains only three records: the previous resource, the resource on air and the next resource, but the playlist can be expanded – it shows the whole playlist in the current day and also colors the resource that's being broadcasted.

5 **Preliminary validation**

This section can be seen as a practical conclusion of the project. It will show if the requirements were, in fact, taken into account and solved in the presented implementation. However, by exposing the implementation, some validation was already shown. Therefore, some of the snapshots already presented can be seen as part of the validation process. Those that fit in this section will be referenced throughout the explanation.

Taking the use cases into account, this section will show a set business scenarios that validate this solution. These business scenarios converge to one final objective: to have a functional channel broadcasting resources.

5.1 **Business scenarios**

• **Scenario A: Create a channel**

In this scenario the producer wants to start using the framework. He wants to create channel “WebChannel” which is a Portuguese channel of general content.

He wants the channel to have two specific tabs: one tab that contains the objective of the channel and it has to appear in the homepage, another tab that contains the location of the company's headquarters.
• **Scenario B: Insert resources**

The producer starts adding resources to the companies database. These resources are both from local disk and referenced from remote machines over the web.

• **Scenario C: Create program “Experience”**

This scenario comprises the creation of a program called “Experience” which starts at the end of one day and ends in the day after. For that to happen, the producer creates two playlists. The first playlist will only pass this program which starts at 23:50:00 and the second playlist has the ending of the program at 00:10:00 on the next day.

• **Scenario D: Visualize program “Experience”**

After creating everything that's needed to broadcast a program, the producer wants to see that program in WebChannel.

### 5.2 Execution of the business scenarios

• **Scenario A: Create a channel**

Firstly, the producer clicks on the blue plus link, concerning the Channel module. After that channel's edition page appears and he can fill all the fields: name, genre and language. After choosing an appropriate logo, the producer clicks on the corresponding link to create a channel. Next, he goes to the tabs edition section (in the same page) and creates the two wanted tabs and, of the tab with the objectives, checks the checkbox which says for the tab to show in homepage.

Finally, he clicks on the back button to return to the edition's (WebComfort) tab.

• **Scenario B: Insert resources**

Using the same manner as before, the producer clicks on the plus sign in the ResourceAdd module. It will open the resource's edition page.

The producer fills the box concerning the title of the resource and browses an FLV movie from his local disk. After that, he selects the option to create a resource. Below appears two new options which allow to associate to the resource both topics and keywords. He adds some topics and keywords to this resources and does the same to the next ones (he gives different topics and keywords accordingly to the resources content).

In order to add other resources without leaving this page, the producer clicks on the links which says to add a new resource. Again, he puts a name to the resource but, this time, he writes a remote link of an FLV movie and clicks on the option to add resource.

Now the database has two topics both from the local disk and from a remote location.

For the database to be more complete, these steps are repeated so that there are, at least, ten different resources in the database.

At the end, the producer goes to the previous page.

• **Scenario C: Create program “Experience”**

In order to create two playlists, the producer clicks two times in the blue plus sign link on the PlaylistSelect module. This will create two sequential playlists.

After that, he edits the first playlist by checking the first playlist and by clicking on the pencil next to the blue plus sign. This will open the playlist's edition page.

The producer starts by putting the title “Experience” in the adequate box. He changes the start time to 23:50:00 and filters the list of resources so that only resources with topic “Documentary” and keywords “Egypt” and “Pyramid” appear.
He starts by adding the resources one by one, adding also a hyperlink to each entry of the playlist. When the last resource reaches 23:59:59 in its ending time, the producer goes back and edits the second playlist. The same is done but, this time, the producer wants to repeat the resources already added in the first playlist. For that to happen he has to import the resources from program “Experience” from the first playlist. So, selects the channel, playlist and program, with this order, in the section of importing program, and clicks on the import button so that it appears all the resources already added in the beginning of the program in the previous day.

Finally the producer adds again the same resources and goes back to the initial page.

- **Scenario D: Visualize program “Experience”**

This final scenario consist in clicking on the WebComfort tab which is called “WebChannel”. It was automatically created by the time the producer created a channel with that name.

To test the channel, the producer waits (or changes his computer's clock) until 23:50:00. When the emission starts, he watches the broadcasted resources, clicking in each one to open the correspondent associated hyperlink. Meanwhile he also navigates in the channel to see the tab that shows information about the location of the company's headquarters.

5.3 Discussion

These four business scenarios tested the whole project's features and all four tests were successful. This means that the implemented solution was adequate to the given reference model.

This validation did not take into account some implemented aspects, such as the ability of changing language, because it was not part of the reference model, although it was present in the studied scenarios. The aim in this assessment was to see if the proposed solution fulfilled the main requirements of the iTV paradigm.

The implementation of each entity – channel, playlist and resource – wasn't enough to prove the system's usefulness and correctness as each module had to cooperate with the other. That's why the validation section is of the utmost importance.

Should the validation have failed, the solution would not be prove to be useful in the iTV paradigm.

6 Conclusions

Despite being an “invention” of this century, the iTV paradigm already has a great number of systems that meet the requirements of a producer/viewer.

The next step would be the creation of standards for the iTV, specifically in terms of playlist creation and resource management. Several online sources (including organizations and companies related to technology and telecommunications) stated that the standards are being discussed, but there are still no published articles about them.

Throughout this study, it became evident that iTV comprises the possibility of a person to publish and visualize television content on a personal computer considering there’s an internet access point. It’s the attempt to transform the viewer into the producer. This tendency already exists as many of existing systems stimulate the user to create its own channel and create its own resources.

It was also lightly studied how the video interactivity paradigm is growing and creating new possibilities both for the producers and users/viewers.

6.1 Future work

This project can be used for further work. Some iTV features could be completed and other paradigm could be fully developed over this solution: Interactive TV.

An example of a possible feature to be inserted is the ability to search videos from portals and insert them into the playlists (like WorldTV). Other possible feature to be developed could be the ability to record movies from devices and put them online directly from this framework. These devices could be not only a webcam, as it already exists, but also a cell phone (that would have the ability of sending its videos through SMS to the website [5]; WorldTV already allows it). A direct consequence of this implementation is the possibility of sending live content through these devices (instead of only storing them, it would be better to be allowed to broadcast the content immediately).
In this project, the Interactive TV was almost cast aside, only developing a simple case of Interactive Video. But to take advantage of the full potential of the iTV, it should be possible to have other kinds of interaction. Examples of other interacting features are the following:

- Create questionnaires in which people could vote and see the results until that moment;
- Choose the winner of a contest;
- Participate in contests;
- See some events in different angles or perspectives (such as sports events);

Finally, it could be implemented an encoder that would allow a producer to add other kinds of movies other than FLV flash movies.

7 References


http://www.webcomfort.org/portal/Home@1.aspx

http://www.webcomfort.org/portal/Home@1.aspx