Context management in IPTV

Rai, M.B.

Award date:
2009

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Department of Mathematics and Computer Science

CONTEXT MANAGEMENT IN IPTV

Mehaal Bellipady Rai

(0666518)

Master's Thesis

Supervisors:

dr. Kees Huizing

dr. Andrew Tokmakoff

dr. Daniel Goergen

Referee:

dr. M. A. Westenberg

dr. J.C.M Baeten

Eindhoven, The Netherlands

August 2009

Abstract

Most modern interactive software applications use well-defined user input or parameters to function and provide service to users. With increasingly interconnected environments, there is a need for software applications to automate, react better and to provide service according to real time events or activities. The era of desktop computing is coming to an end and moving towards ubiquitous computing. Ubiquitous computing makes use of real-time information also referred to as "context" which keeps changing. Context information is not currently utilized in many user applications right now. There are ways for this information to be obtained and passed as context to applications, so that the applications could react to it and provide better services to the user. These systems are of referred to as Context Management Framework and are a form of Middleware.

Such middleware enables applications to become context aware. There is an exchange of context information between context source(s) (which produce context) and context sink(s) (which subscribe to, and collect context information). Context Sinks are utilized by Applications to become context-aware.. Applications use context to provide meaningful services; they can also act as a context source, providing contextual information to other context-aware applications. Designing context-aware applications is still an area of ongoing research and the applications which can benefit from context-awareness remains to be fully explored.

This thesis describes investigations into the use of context information in the IPTV domain, provides an overview of means for managing context in the IPTV domain, and also outlines the design and development of context-aware IPTV widgets. The context information received from context sources is communicated via a Context Management Framework. Context-Aware widgets subscribe to this context and adapt their behaviour according to the available context. We identify a set of context parameters which are suitable for use in the IPTV domain. We propose the most relevant higher level context information for the television domain some of which can be derived from other context information. We have developed a set of IPTV widgets which are linked to popular web 2.0 applications. The main achievement is the creation of Context-Aware widgets which enhance the users experience in his television session through the use of context information. A user test was carried out to check its relevance. Statistical evaluation was carried out to show its significance.

Acknowledgments

This thesis would not have been possible without the support of many exceptional people. I owe a lot to my advisors at Philips Research, dr. Andrew Tokmakoff and dr. Daniel Goergen who assisted me during my time there. I was guided and supported from the beginning to the end of this project. They were willingly subjected to several versions of this report and provided me with insights, comments and suggestions to make the report what it is right now. I would like to express my sincere gratitude to dr. Kees Huizing, who supervised me from the university. He believed in me and motivated me in every meeting we had. He provided me with timely and constructive criticism. He continuously monitored my progress in terms of development and spent ample time in correcting my earlier drafts of my report.

I am thankful to Prof. Preetham Kumar for being my project coordinate from Manipal University and for his evaluation. I am grateful to dr. J.C.M Baeten and Senior Director of Philips Research Jos Van Haaren without whom i would not have been doing my internship at Philips Research in the first place. I would like to thank Research Scientist Michael Verschoor for helping me with my user test analysis. This project would not have been possible without the support of dr. Manohar M. Pai and dr. Radhika M. Pai without whom I would not have been in the Netherlands. Finally, I must express my appreciation to my family and friends who have stood by my decision taking me one step closer towards a bright future.

Mehaal Bellipady Rai
Eindhoven
August 14, 2009

List of Figures

Table of Contents

Chapter1: Introduction

1.1 What Is Context?

It is important to realize what difference context can make to an application. Identifying context which makes the application more useful, more effective is itself a task. Context was previously defined in many ways. The term ‘context’ was first named by Schilit et al. (Schilit & Theimer et al, 1994) and referred it as location, identities of nearby people and objects and changes to those objects. Ryan et al. (Ryan, 1997) defined context as the user’s location, environment, identity and time.

The problems with these definitions are that these are based on examples of what they intended to achieve. Therefore most of the definitions of context are example specific. Hull et al. (Hull,1997) included the environment into context by defining it to be the aspects of the current situation. Schilit et al. (Adams, 1994) referred to a general category by claiming that the important aspects of context are where you are, whom you are with and what resources are nearby. Context is any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves. Schmidt et al. (Schmidt, 1998) defined context as a “the situation and the environment a device or the user is in. The appropriate definition which we came up with, which could fit for a TV domain is,

Definition 1: Context is information from the user, environment, device or the application itself that can influence or change the working of the application, making it more aware.

Aware is defined by having or showing realization or perception. As the platform we are using in this project is the television domain, the user, the environment around the user, the devices around or the device itself and the application usage becomes context information. Context information is also got from sensors and termed as context sources.

1.2 What Is A Context Aware Application?

Context-aware systems now play a very important role in modern software systems. The term ‘context-aware’ (Adams, 1994) has been defined earlier as “systems that adapt according to the location of user, the collection of nearby people, hosts and accessible devices, as well as to changes to such things over time”. Hull et al. (Hull, 1997) defined context-aware computing to be the ability of computing devices to detect, sense, interpret and respond to aspects of a user's local environment and the computing devices. Our definition which is suitable for the TV domain is,

Definition 2: A context aware IPTV application is an application which runs in an IPTV environment and which makes use of context information to provide an improved IPTV user experience.

Context information has to be considered optional. An application should be able to provide all the services without context information. On it being context aware, the service has to be improved. But there is need to acquire this context information and the need to develop applications which can use this information and react accordingly. In the below figure, we see how context information is acquired and used. It is important to note that a context binding (Broens T. Q., 2008) occurs when there is an exchange of context information between the application and the source.

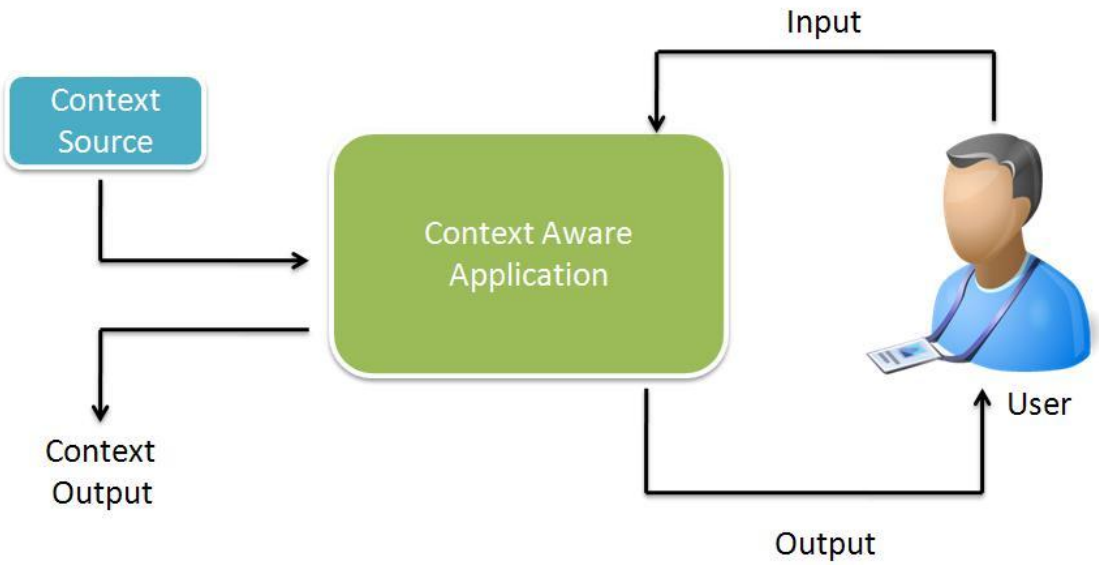


Figure 1.1: *Context Aware Application*

The functionalities and services provided by the application are used by the user. The way he uses the application itself can be context information. The state of information about the user can be context. All this information is an input to this application. There can also be other context sources providing information to this context aware application. On receiving all this context information, this application reacts to these changes or information provided. On getting this information, if it improves its services provided to the user then the application is context aware. Context information which does not improve services are not considered or used.

Example of such an application is the context aware tourist guide application (Park, 2007) which takes the current location of the user to provide vital information to the user. It uses the co-ordinates of the user to track the user's precise location and provides the user with the nearest places of interest or famous locations and it uses the internet to track the most visited locations in the given area. It uses current time and time schedules to react differently.

One of the issues (Broens, 2004) in acquiring the context information is that it can change every single second or abruptly which makes it very difficult to develop an application for it or in other words it makes these bindings hard to maintain. Some context information can keep changing like current bandwidth and signal strength. As the context information's quality and lifespan is arbitrary comparing to the entire lifespan of the application, it makes the binding difficult to maintain.

Most of the context information received is of low level and there is a need for high level context information. To build a context aware system or an application involves different stages like acquiring context, handling of context information, followed by context modeling and context reasoning etc.

1.3 Abstract Architecture of a Context Aware System

The architecture of a context aware system is designed in such a way that all the necessary context information required by any context aware application is present in the context management system at all times. There exists a mechanism of subscription for any context changes by an application thus making the application more responsive and reactive. This context information can be acquired from various sources. These sources include sensors in the environment, user information and the properties or functioning of the device itself. Context information got from these various sources/sensors can be combined to derive higher level context information which could be useful for the majority of the applications. The aim of a context aware system is to automate the system in a way that it acts according to the current state, thus enhancing his overall experience.

A user profile can also be stored, which is a continuous learning process of the user's activities thus building a profile of himself which consists of his likes or dislikes, interaction history etc. Such kind of information derived from a user without his actual input helps in recommendations. It is possible to identify the user's schedule by monitoring the context information such as time of usage for a longer period of time. In recent years, the main reasons for the emphasis on the development (Broens, 2004) of context aware applications were to learn by looking at the previous history of context information and to improve services without any user intervention. Context is dynamic, thus making the experience more dynamic.

Figure 1.2: *Abstract architecture of a context aware system*

The environment can contain many context sources. For example, to measure light (ultraviolet sensors, IR sensors), face detection (camera's), objects around (positioning system) and traffic (GPS) were used. For the users, there can be important context information such as emotions (voice sensors, camera's), mood (communication sensors), voice (Microphones, amplifiers) etc. Context information can be got from devices such as location (GPS, mobile network), size, connectivity etc. Context is gathered and used to create an associated model which we then use to assist applications such as IPTV to become context-aware.

Once such context information is gathered, it is important for context to be managed. Examples are: on a change of context state, a specific action takes place, a specific application should launch, a recommendation is made, an action is carried out etc. All this information is controlled by the context management framework which is connected to the context aware application. These applications react depending on the context information available.

Here we concentrate on IPTV (Wisegeek, 2003). IPTV is the transmission of television signals over internet. It is a system where service is delivered to the user using internet protocol. We use the information from the internet to build and test applications relevant for television. As we will be looking into the TV domain, by identifying the user's remote usage based on context inputs such as the response during ads, time at a channel, certain context aware applications will be displayed on the TV. It can be either done for a single session or for multiple sessions. An example of this would be that if the TV is able to identify that the user is very happy watching the TV (comparing emotion, pulse), the session is stored which can be viewed later on. Another

example would be the case of multiple people present in the room (face recognition, RFID badge), the IPTV provides recommendations which match the likes of all people in the room. A major concern is the privacy or the preferences set. The important issue is to define what drives the IPTV application or the recommendation to appear, is it because the user is not looking at the TV for a long period of time or is it because the user is doing some other activity in front of the TV or is it because the user is engaged in a conversation with another user in the same room etc. This is what context management (Hesselman, 2008) is about, identifying when the context change is relevant and the difference it makes to the application which has subscribed to these context changes. The usage of the IPTV itself is another context source as a lot of information can be derived from it such as usage channel being watched, program details, genre, ratings and navigation pattern. Such information helps in building the user profile because it identifies the user's personality or user's interests and his knowledge of the overall system.

1.4 Objective and Research Questions

The goal of this thesis is to develop context aware applications which can be used on television and verify if it enhances the user experience or not.

Some of the objectives and questions of this research are as follows,

- To understand context and identify its possible usage in television domain
- Which context information can be useful with respect to television?
- Is it possible to create higher level or rich context information gathered from different sensors?
- Implementing one or more context aware applications for television.
- Do the necessary implementation for the iNem4u European project. (iNEM4u, 2008)
- Conduct User test to analyze the acceptance of context aware applications on TV.
- Perform statistical analysis to check its significance.

These context aware widgets were developed for research purposes to look into how to enhance the existing social TV framework which would provide additional services to Philips Television users. This entire setup of widgets on TV is going to be evaluated through user demo with interviews to realize its potential and to identify the user's acceptance to widgets on TV. This research assumes that the context sensors or context sources are widely available and can be combined to generate higher level information. This research focuses on building at least one or two widgets for television which can be context aware. A part of this project is to develop widgets and connect it to a user interface using a framework called as Context Management Framework (Kranenburg, 2006). It is also referred to as CMF. The CMF manages all the context information from the context sources. The scope of this project is to understand the significance of using context information by developing context aware applications and evaluate the use by analyzing user's feedback.

1.5 Thesis Outline

In this chapter, we introduced briefly the concepts we will be looking into and are the most commonly used terms, so a precise understanding on these concepts is necessary. In chapter 2, we discuss about the technologies, applications needed for the widget application and information regarding the European project. In Chapter 3, we discuss about the research involved in this project. We speak about the different kinds of context information available with examples and sensors required to identify these information. In Chapter 4, we discuss about the design of the IPTV system and the implementation aspect in this project. We discuss about the conducted user test and the statistical analysis of the developed widgets in chapter 5. Finally, we conclude with the significant results obtained from the contribution we have done to the project.

Chapter 2: Background and Related work

2.1 What is IPTV?

IPTV stands for Internet Protocol TV. It delivers content to watch over TV through the internet. Philips offered users access to internet from the television using Net TV. (Philips Net TV, 2009) Earlier, receiving television content was done through dish network or cable wires. In this case, the television is directly connected to a broadband router and receives signals over the internet. IPTV usually operate over a private IP network but not the public internet. The advantages of having a private IP network is that Quality of Service can be guaranteed to the users of this service. For an IPTV network, TV signals are given the most priority.

The advantage of it being broadcast over the internet is that it solves the problem of integration with other services such as VoIP and broadband internet for the home computer. For all these services a single provider can be used thus making it cheaper and more efficient.

Advantages of IPTV can be divided into three areas (BSF, 2007),

- Content – More content is made available with easier access
- Convergence – Allows single applications to run over multiple end user devices
- Interactivity - Interaction among service providers and subscribers can be done both sides.

Below is the architecture (BSF, 2007) of an IPTV system,

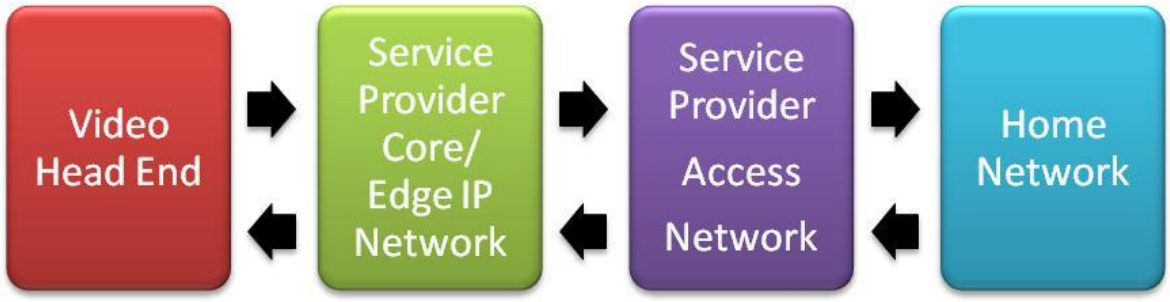


Figure 2.1: *Key Elements of IPTV system*

Video Head End is the point from which linear and on demand movies in sent for distribution over the IP network. This head end encodes each channel into MPEG2. MPEG4 is slowly being used by a few providers as it is easier to encode the HD signals. It is sent into the network in

multicast streams (MulticastStreaming, 2003) as many users can tune into the same broadcast channel at the same time.

The encoded video streams are sent via the IP network. Every network is unique to the service providers. At the network edge, the network connects to the access network. The access network signifies the link between the service provider and the house domain. DSL is the technology used here. There is a necessity of a DSL modem to deliver the Ethernet connection to the home network. The home network distributes the IPTV service to the device. It is connected to the set-top box and from that to television. The IPTV middleware is used to describe the software packages associated with delivering IPTV service. Service such as Video on Demand (VOD) and electronic guide are all controlled by the middleware.

If the television is connected to the home network, it can be used to seamlessly integrate with other devices on the network like a personal computer. It is possible to play video files and music from the other devices. With IPTV came a lot of other services such as VOD where programs and movies could be watched during the user's desired time. A list of programs and movies list is provided by the service provider and a request to it can be made. The quality compared to the other mechanisms is very good as it uses a much better standard of compression (MPEG-2).

The advantage of IPTV (Wisegeek, 2003) is that the quality of broadcast is much better and a lot of information can be added. When it comes to a satellite TV or a broadcasted line, each of the television signals is sent every time. It leads to usage of a lot of bandwidth. But in the case of IPTV, the data to be sent is stored in the server and only the channel currently being watched by the user is broadcast to the television thus saving a lot of bandwidth. With the addition of internet on TV, interactivity is possible. IPTV is the next generation TV provider because of its two way communication for interactivity. It enables users to participate in live shows thus helping in real time shared experiences. Information can be easily obtained from the internet and can be displayed. People can take part in quiz games hosted and thus the TV session can be more interactive. Another advantage of the IPTV is the ability of recording multiple broadcasts into digital video recorders. Recent developments in IPTV include the introduction of gigabit Ethernet, higher security, high performance IP routers, Ethernet switches and advanced middleware applications. This project concentrates on generating real time information from the internet on to the television sets.

2.2 TV widgets

Widgets are tiny applications that provide information obtained from the internet or local resources. These widgets usually show information using a textbox or a small GUI window. Widgets are usually placed in computers, blogs or websites. Widgets only contain information the user wants to see unlike the ads placed in different websites. Widgets can be made in an instant using some websites giving information exactly what the user wants. Some of the websites have their own widgets. Examples of such widgets can be CNN news or Weather

updates. The user goes to the websites offering these widgets, chooses his preferences and downloads these widgets which show real time information according to his needs. In the following chapters, importance is given to find out what a TV widget is, the technology behind it and also how it can be used effectively

.

2.2.1 Widget definition

The definitions of "widget" stated are different based on the kinds of widgets. According to W3C (W3C, 2009), "A widget is an interactive single purpose application for displaying and/or updating local data or data on the Web, packaged in a way to allow a single download and installation on a user's machine or mobile device. A widget may run as a standalone application, or may be embedded into a Web document." According to Google, "Gadgets or Widgets are simple HTML and JavaScript applications that can be embedded in WebPages and other apps." (Google, 2006) According to Nokia, "Widget is a thin application that resides on the device and does one task very well." (Mobile Widgets, 2006)

Most of the times widgets are usually static and sometimes with little interaction. Widgets are stand alone applications having certain functions. The more the functionalities, the more closer it gets towards being a fully fledged application which should not be the case. Hence, widgets are usually small but can be varied to get different kind of results and the look and feel plays an important role too. Widgets are usually programmed in HTML/XML/JavaScript/VBScript (Wiki Widgets, 2009). Examples of the personal widgets are sticky notes and calendar. These widgets need not necessarily contain information from the internet. A widget engine (Wiki Widgets, 2009) is a software service available to users for displaying widgets on a GUI. Some of the most common widget engines are Yahoo! Widgets (WikiYahoo!, 2007), Google gadgets and Dashboard widgets.

2.2.2 Types of Widgets

The Widgets are mostly differentiated into the following four categories (Mobigets, 2009) depending on the platform they are running on.

- Desktop widget – Targeted for PC's, stand alone applications. [Fig 2.2]
- Mobile widgets - Displayed on Mobile, easily movable on screen [Fig 2.3]
- Web widgets - Included in a web document
- TV widgets - On a Television Set

Figure 2.2: *Example of Desktop Widgets*

In this project, we will be working on TV widgets. The resultant information generated from the internet will be displayed on TV using small stand alone graphic user interface applications using flash development. (Flash Developer Center, 2006)

Widgets are also classified based on their functionalities.

The different classes of widgets are

- Information Widgets – Specific Content from a webpage
- Application widgets – Depends on other applications
- Utility Widgets – Uses the devices resources
- Fun Widgets – More emphasis on entertainment or games.

In this project, we will be concentrating on information and application widgets.

The advantage of having widgets in a system is that content can be directly shown to people. The user does not need to go through the process of visiting the website to find out information. Another advantage of widgets is that they can be personalized. Widgets can be created to display specific information or the category of information the user is interested in. It shows exactly what the user wants.

With the increased usage in Internet and with the user's shifting towards social sites or real time information, this project concentrates on Web 2.0 applications like YouTube, Twitter and Blogs (Web2.0, 2004).

Figure 2.3: *Example of Mobile Widgets*

Thus as time progressed, widgets became handy and widely spread. The iPhoneSDK enabled users to build widgets for their mobiles. Therefore the widgets have moved from one domain to another and the next generation of widgets could be on TV. Yahoo! officially announced its plans on working on TV widgets.

In the TV platform, TV widgets are small web applications that give the user the contents he requests for and that content can be dynamic. Neither are the widgets highly interactive, nor have a lot of functionalities but they are meant to deliver the requested information. These widgets to be designed will be accessible with a single button and shown as an overlay which can be closed anytime or positioned anywhere. They will save you the time spent online or in different channels by giving specific filtered information.

An example of an interesting widget for TV would be the weather widget from CNN. This widget would come in handy if the user is interested to know the weather updates. Instead of watching the channel and waiting for the weather update it can be graphically displayed as a widget for that specific location on a single click.

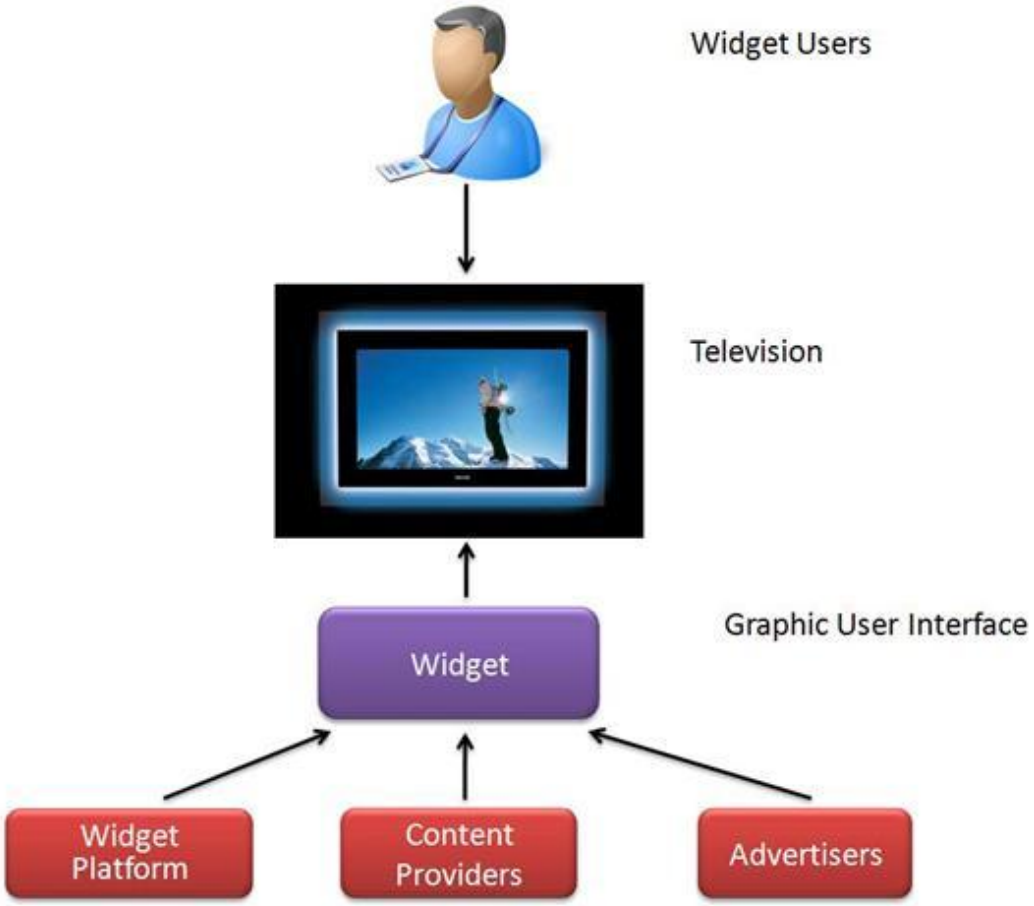


***Figure 2.4**: Abstract Architecture of Widgets*

In reference to this project, some of the widgets made will be context aware. The widget will be displayed on TV, depending on the context information obtained. It searches the content provider like YouTube, Twitter for content which can be displayed to the user. Widgets will be linked to the TV content providing real time information from social sites. We will be implementing a few personal widgets which will be displayed during context events such as appearance of advertisements.

2.3 What is a Web Feed?

A Web feed is a data format used for providing users with frequently updated content. There are two main types of feeds one being RSS and the other being ATOM. RSS stands for Really Simple Syndication (Wiki RSS, 1999). A feed link displays the contents or updates of a certain page. It can also be configured to display information according to the user's specific choices. RSS is a "mini database that contains headlines and descriptions of your web content, including hyperlinks that enable users to link back to the full article of their choice. RSS is a Web content

syndication format. RSS is a dialect of XML. The RSS feed is actually contained in a small XML file that contains the items which are subscribed. Nowadays, many sites provide RSS feed.

2.3.1 Type of Feed – RSS 2.0

The first implementation of RSS was the 0.90 (Wiki RSS, 1999). It was developed by Netscape for displaying the headlines of the main news websites. The development was complex for a simple feature. On repeated calls for an easier structure, Userland Software came up with RSS 2.0 which is the current one in use. Below is the format of a sample RSS 2.0 document (XML RSS 2.0, 1999).

```
<rss version="2.0" xmlns:dc="http://purl.org/dc/elements/1.1/">
  <channel>
    <title>XML.com</title>
    <link>http://www.xml.com/</link>
    <description>Description of the information to be displayed in the items.</description>
    <language>en-us</language>
      <item>
        <title>Normalizing XML, Part 2</title>
        <link>http://www.xml.com/pub/a/2002/12/04/normalizing.html</link>
        <description>In this second and final look at applying
         relational normalization techniques to W3C XML Schema data modeling,
         Will Provost discusses when not to normalize, the scope of uniqueness
         and the fourth and fifth normal forms.
         </description>
         <dc:creator>Will Provost</dc:creator>
         <dc:date>2002-12-04</dc:date>
      </item>
    </channel>
</rss>
```

The first line of the code indicates the Document element. It specifies the kind of RSS being used. This element contains all the other elements within it. Within the <rss> element is the <channel> element. It provides information about the information being displayed also referred to as feed. Within this channel element are tags which provide information about the feeds to be displayed like <title>, <link>, <description>. This element gives a description of what all the items represent. Every bit of information that is shown in a feed is placed within a tag called <item>. The element <item> contains the elements <title>, <description>, <pubdate> and <link>.

```
<item>
      <title>Mehaal Rai Blog</title>
      <description><![CDATA[Link of me singing Aadat]]></description>
```

```
    <pubDate>Thu, 08 Dec 2008 08:39:51 -0500</pubDate>
    <link>http://www.mehaal.blogspot.com/steve</link>
</item>
```

Each <item> is different and constitutes to different information within a feed. The information within an item can be the main title of the feed, the description in a line or two, the time the content was published in the site and the link to the site. The link is provided in case the user wants to read more about this feed.

2.3.2 Type of Feed – ATOM

A few years ago, some developers offered another format performing similar functions to RSS and termed it as Atom (Wiki Atom, 1999). It is also based on XML and relatively complex. Most of the blog sites usually have an ATOM feed along with an RSS 2.0 feed. For example, if we consider Blogspot by appending /atom.xml to the URL we get the blog's XML feed.

Example: *http://blogname.blogspot.com/atom.xml*

Below is a sample of an Atom feed.

```
<?xml version="1.0" encoding="utf-8"?>
<feed version="0.3" xmlns="http://purl.org/atom/ns#">
        <title>Testing Blog</title>
        <link rel="alternate" type="text/html"
        href="http://example.com/blog/" />
        <tagline>This is a testing blog!</tagline>
        <modified>2005-01-13T12:21:01Z</modified>
        <author>
                <name>John Doe</name>
                <email>john@example.com</email>
        </author>
        <entry>
                <title>Test #1</title>
                <link rel="alternate" type="text/html"
                        href="http://example.com/blog/2005/01/01/foo.html" />
                <issued>2005-01-01T09:39:21Z</issued>
                <modified>2005-01-01T09:39:21Z</modified>
                <id>tag:example.com,2005-01-01:example.001</id>
                <summary type="text/html" mode="escaped">
                        This is an example blog posting. &lt;a href="http://www.
                        Example.com/foobarbaz.html"&gt;Foo Bar Baz&lt;/a&gt;.
                </summary>
```

```
        </entry>
</feed>
```

An Atom feed, starts with the tag <feed>, followed with the <title>, <link>, <description> and the <author>. The link points back to the webpage the feeds are actually being displayed from. The <author> tag usually holds the name and the contact information of the author. The <entry> section is pretty much the same as an item field with some slight changes in the elements. It consists of <title>, <link>, <issued>, <modified>, <id>, and <summary>. Issued and modified specify when the specify entry was added or changed. The id tag is a permanent unique identifier for an entry.

A site which has web feeds displays one of these icons shown below.

The advantage of a feed is that the content to be seen by the user is handpicked thus giving information only which is required. It helps by providing a lot of information to the user at the same time. It acts as a good platform for advertisments.

In this project, we will be entering search queries to feed generating links based on some input. The input to the links are the context information acquired from the sensors. The output of this is an XML file and we will develop widgets that can parse these XML results acquired and display them on TV in a orderly and timely fashion.

2.4 The Context Management Framework

The Context Management Framework (Kranenburg, 2006) is also referred to as CMF. (Hesselman, 2008) It is used by applications to discover context information. There is a CMF container running on a server or a system of an user. It consists of a ContextSourceRegistry in which all the context sources are published. The ContextBroker which uses the ContextSourceRegistry and ContextSource to discover context brokers offers an interface (ContextAccessInterface) that enables other parties to discover the relevant context sources. It wraps information retrieved from sensors or domains managed by a single ContextBroker. A context wrapper hides platform and device specific details from the generic infrastructure. CMF offers discovery capabilities based on identities and context types. Context sources are registered to context agents. The application chooses which context information it wants to retrieve. When choices are made of which sources are going to be used, the CMF does the connection using a proxy which takes care of privacy rules.Values taken up by the source on change can also be logged. A context source provides context information. Context sources use ontology to describe the types of context information they provide. Context sinks use a discovery mechanism to find context sources. Every domain has at least one context broker, but it is usually the case that all the devices that support CMF run a context broker and the context brokers exchange information within the domain. A context agent is a service that is obligated to a specific entity. Context

agent keeps track of the context sources that can currently provide context information about the entity that the agent represents. A domain will publish a context agent for each entity about which it wants to provide context information. Since a context agent is the single point of access for the context information about a particular entity, it needs to enforce the entity's privacy policies. CMF follows the Policy Core Information Model which consists of Policy Decision Points(PDP) and Policy Enforcement Points(PEP). PDP is used for evaluation policies and PEP requests a policy decision from a PDP and enforces its decision. PEP's in this architecture are Context Agents and proxy context sources. In this, the context agent decides which information can be sent to the application and the proxy context sources make sure that the application gets this value. CMF is implemented as a Java component. It runs in a distributed run time environment which makes it easy to work over all operating systems. Context sources use OWL ontology to describe the types of information they produce. An example of the deployment is as follows. A CMF enabled PC is set up at an office and in the three different houses. Office is the home domain and the three other houses are the foreign domains. Sensors are placed to identify the user's presence at home like Bluetooth dongles. If the user is present in his house, the employee's context agent is informed.

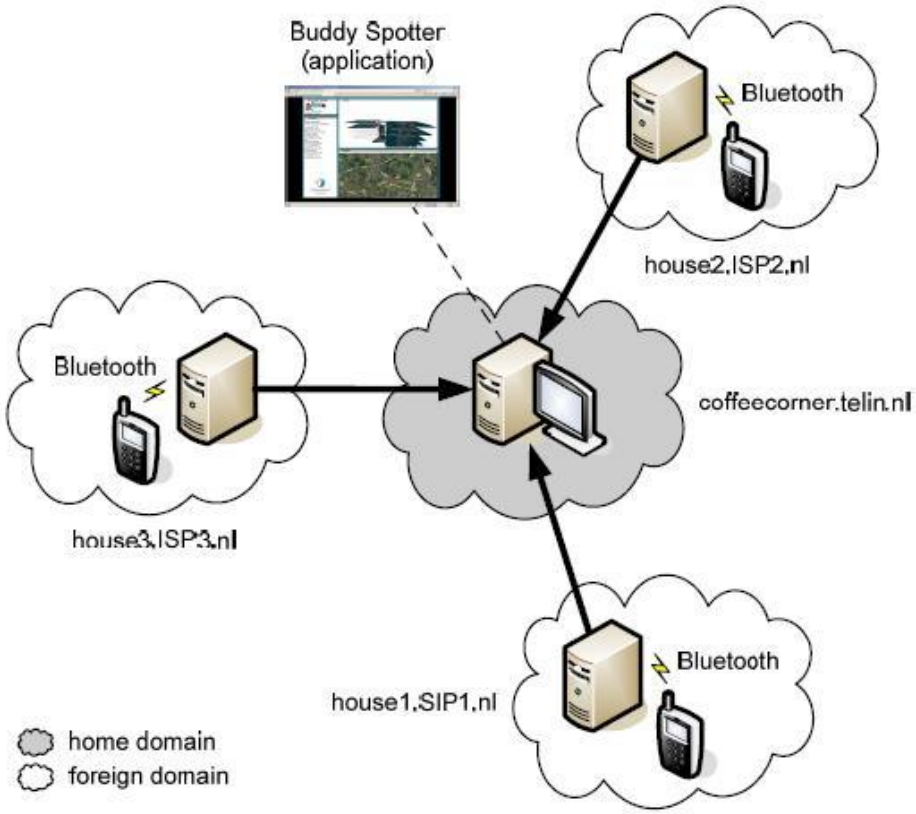


***Figure 2.5**: Deployment of CMF*

In this project, the home domain would consist of a many context sources. Some of the context sources come from various sensors and some come from the television. The CMF might be

hosted outside the home, probably in some kind of a cloud. A cloud is a dynamically scalable, virtualized resource which is provided to user as a service over the internet. There exists an in-home context agent and there can be many context sources registering to provide information for that agent. It is enough that the broker lies in this cloud. The context agent represents the home and registers themselves to the broker lying outside the home domain. All the context information remains local unless it is made visible by the user. The context management framework has been used in different projects like Freeband AWARENESS (Freeband, 2006) and Amigo project (Amigo, 2008).

2.5 Related Work in Context and IPTV

The upcoming web technologies that help in standardization in the IPTV field are W3C, HTML-5, CE-HTML (Fokus, 2009). These new technologies give us a platform in which flexibility is possible with applications working on mobile devices and television systems. These new technologies help in making the TV as the main sharing terminal and the mobile phone's interoperability with TV as a personalized terminal. FOKUS group (Fokus, 2009) which is the leading group into IPTV domain has conducted research in area of web based applications which can be used over three main devices, the desktop, TV and mobile devices.

The advantages of the FOKUS widget platform is it helps in coming to an agreement with different web concepts also considering the features of the device it is to be used in. Another advantage is the seamless integration of applications over the different platforms available. Earlier the most common fear of context-aware services involved is recording context information which shows many aspects of a user's life and thus threatening his or her privacy. This is solved by using context only within a personal network (Yaiz, 2006). This network handles all the privacy and security needs of the user. These services react to the user's movement around the personal network space and personalize the devices within the network based on the context information. The user can add new sensors he wants and can install new context processing software that provide new services based on the newly derived context information. One such software which should be installable on a mobile and IPTV device is an application that checks the user's health conditions and identifies emergencies if any (Mobihealth, 2007).

Whenever a personal device is being accessed at a new location in this case the IPTV is being accessed by a user elsewhere, service that has already been established continues to be maintained. Context information can also be exchanged between personal networks or devices so that services can be continued to be used. Context information can be used in application or service environments for enhanced service creation and management and in network environments towards optimized end-to-end network selection and routing if interoperability is desired.

Context awareness in Interactive Television applications can be used as a powerful tool to deliver most relevant and personalized ad based on user's context (Thawani, 2004). Below is the summary of the context used for this system which takes into context the user's current activity and the past activities.

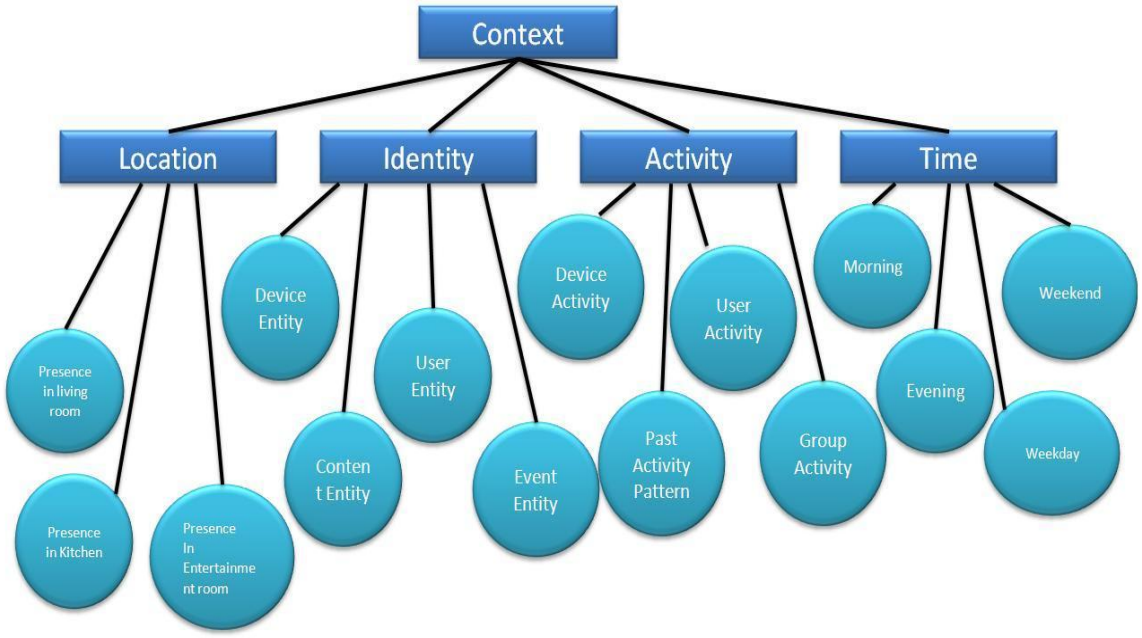


***Figure 2.6**: Selected Context for ad insertion*

This approach considers the context information in the past and relates it to the current activity, thus determining the relevant advertisement. The rise in the usage of PVR's and VOD is seen as a threat to the advertiser's as user's skipped the advertisements. Thus came the need for delivery of personalized advertisements by considering the likes and activities.

In this above architecture context for the TV is derived under the four main categories which is Location, Identity, Activity and Time. Location context can be such as "user is the bedroom" or "user is in office". Identity context information can be the user information or the device or content identity. User identity can be got from the home information system stored in the Set Top Box which usually includes information such as age, occupation, gender. Device identity can be resolution, features etc.

2.5 Summary

In this section we understood the connection between the information given in the previous two chapters with respect to this project. In this project, we built context aware applications. The context aware applications were based on popular web 2.0 applications. The applications were

designed and developed as widgets on IPTV. The widgets show information as feeds. These feeds are either in RSS 2.0 or ATOM format. We use these feeds to develop context aware widgets to be displayed on TV. The entire IPTV system is designed with modules having their own functions. CMF manages the current context information from sensors and the widgets use this context information to improve its services. CMF is a middleware for the current context. Applications request for information in the context sink and display on its conditions being met. The widgets developed are context aware as the information they provide change according to the current TV session.

Chapter 3: Determining IPTV Context Information

In this chapter, we will describe the context parameters valid for the television domain. The main aim was to identify this information and select the most important context information.

With respect to IPTV, we divided the context information into three main categories,

- Device
- User
- Environment

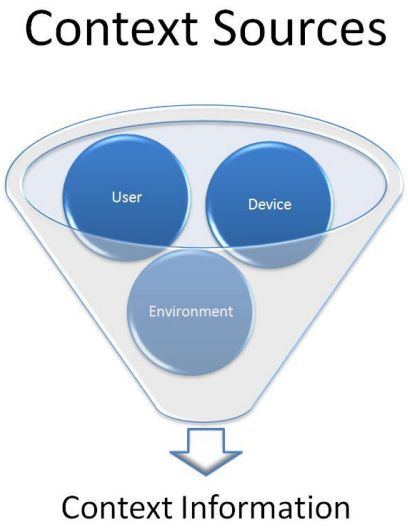


Figure 3.1: *Categories of context information.*

The three context categories are again sub divided into various context information types which can be used for any IPTV application to improve its services. Context information can be combined to derive higher level context information which can prove to be very useful in the application point of view. This combination can be done using various reasoning engines or different kinds of context information. For each of these context sources, sensors are assumed to exist which provide the application with the information it needs to form a context aware system overall.

3.1 Device Context Information

The most commonly used devices such as Mobiles and TV can provide context information to an IPTV application. Information such as current location and connectivity (type of connectivity, current bandwidth) are the important information required to be known to some applications. Example: Mobiles are handheld devices that can provide information such as the user's location. The current location of the mobile can be determined via the stations providing the range for it. One time context information of the mobile can be determined such as size (varies from mobile to mobile), connectivity (which determines the speed of the internet connectivity like GPRS, 3G, Wi-Fi), hardware (supported technologies like Bluetooth, Infrared which adds more value from the application point of view), Software (Operating System – Windows, OS X iPhone), RAM (shows the mobile's processing power), volume of the device or from the device(TV can adjust if it notices a mobile being used or on a incoming call). This information is required by the applications to check if its services are supported by the device. Status or profile of the mobile also provides context information. Device status has to be ON for any request to be sent to another user. If the mobile is synchronized to the calendar and the user has a meeting or enters a meeting room, it automatically changes the profile to Silent or Meeting. The caller will be notified of the meeting the user is in or the mobile will ring with just a vibration.

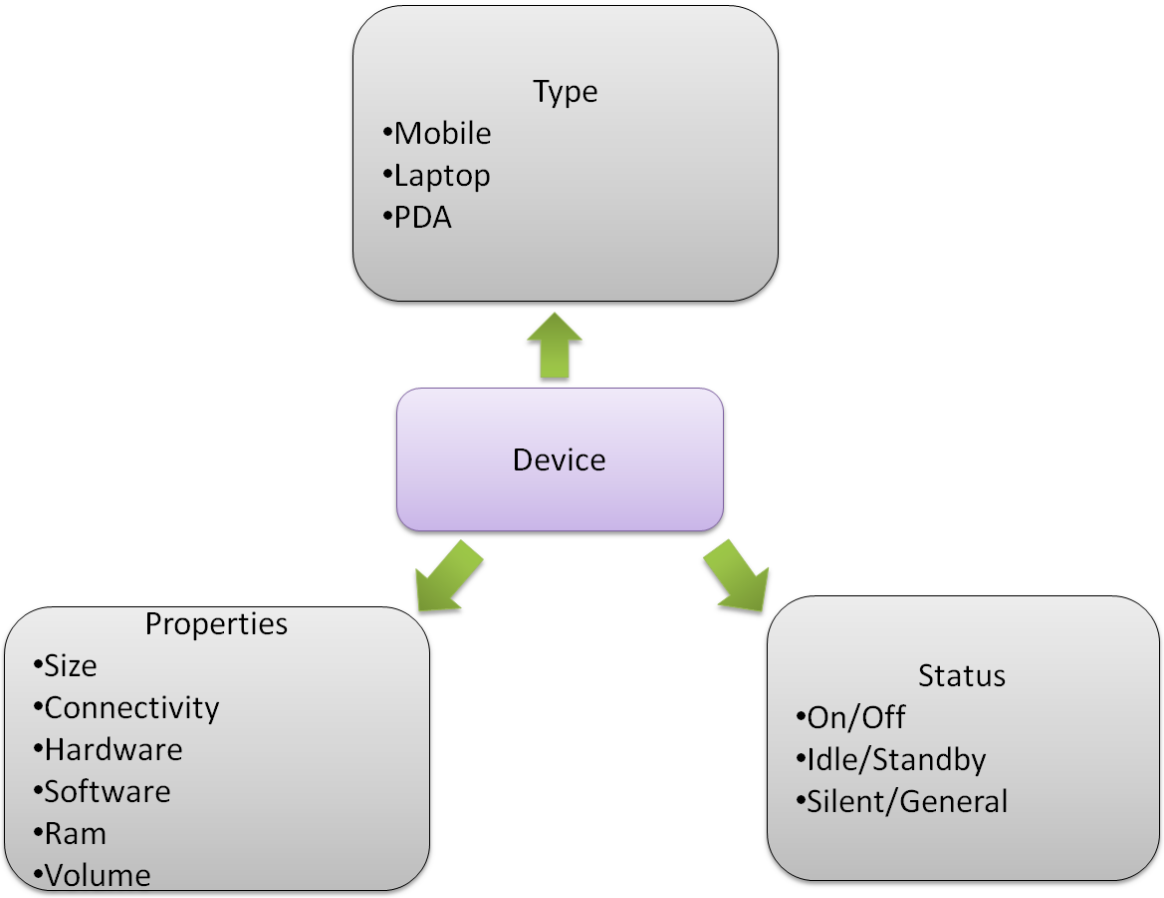


Figure 3.2: *List of device context information*

Other devices such as laptops, music systems, digital camera's can also be connected to a TV thereby increasing the number of applications possible. An example could be a photo sharing application on the TV, which connects all the devices to the TV and copies all the photos from these devices into a private space (only for the user's usage) and a public space (photos can be dragged from the private space to public space, so that friends can see it, thus ensuring privacy) (Lowet D., 2007). There are devices which can be wired into the home network like gaming consoles. The devices connected to the home network have to be monitored every time and on being plugged in, its availability in the network has to be made known to the other devices.

3.1.1 How to collect device context:

Device context can be acquired by monitoring the network or devices around the TV system. Information about the TV can be determined and based on configuration. Every other device has a set of services it can function and this information is made known to the TV system. It depends on the hardware and functionalities the device supports.

3.2 User Context Information

From the IPTV point of view, a user provides context information which can be useful to the application. However, IPTV applications use only few of the available context information. It is vital to choose information which can be used in many applications or context information which can be used to make the system context aware. Mentioned below is some of the context information derived from the user which can be used by the system under study. Some of the context information which can be got from a user via sensors is pulse, body temperature etc. As these change often, it is calculated over a longer period of time. Some context information does not change for a long time. Example: Age, Nationality.

Context information related to user's state such as emotions (excited, sad, happy, serious etc), intentions and habits (clapping on watching something funny, no communication while watching something serious) is more program dependent. An example of the usage of the above context sources is an application which records the TV program when the user is very happy while watching TV or when the user claps. The recorded session can be shown to the user when he is in a sad mood or as a recommendation when he is watching the TV in a browsing mode, so that he could relive his experience. The user can also view the recorded videos and make a video out of it and send a recommendation to his friend.

Decisions can also be made by looking at the user's calendar information. TV experience can be enhanced by personalizing it during special occasions like birthdays and public holidays. It is not required to provide a recommendation of watching a movie which is three hours long when there is meeting scheduled for the user in one hour from that time. Therefore recommendation uses context information and uses it to provide its services to the user.

Applications have to be interoperable between various devices to make it a good experience while using IPTV. If there were ways to identify what the user plans to do, different applications could be developed for it. The best example here would be looking the calendar of the user to give reminders. The other futuristic example is, if the user is going to have dinner, then an advertisement pops out showing the offers for the day from the nearest supermarket. This might be the future of IPTV in years to come.

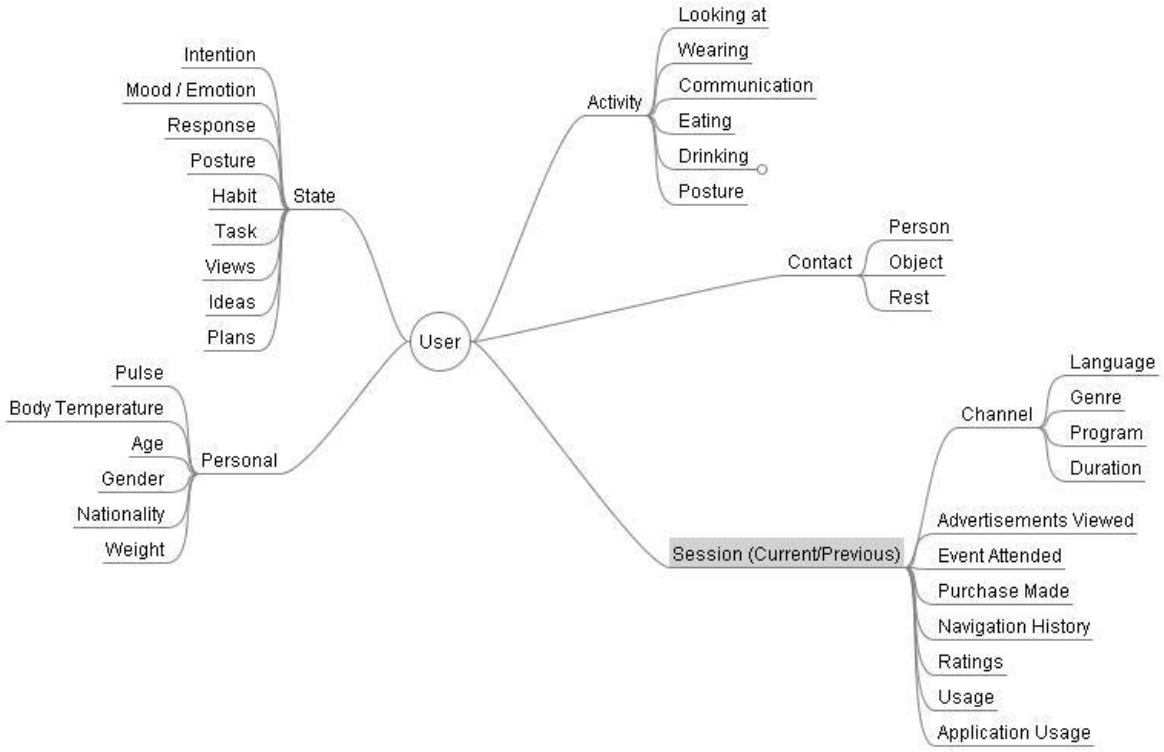


Figure 3.3: *User context information*

Knowing the activities of the user when sitting in front of the TV can help in targeted advertising or automation of the TV. If the TV can recognize when the user is not looking at it for a specific period of time, it can show some recommendations evaluating his likes in the personal profile that is being automatically built with his daily sessions on TV. Applications can be displayed when the user is changing channels at a very fast rate and not sticking to one channel.

If user presence can be determined, and the user is not looking at the IPTV or not communicating with the person in the room, the recommendation manager once again shows a recommendation which matches the likes of both the people in the room. Posture (Kamiya, 2008) can be a form of context information as it shows the mood the user is in or the plans he has for time to come. If the user is in a sleeping posture, it means that he has no work for some time or that he is very

tired (which can be determined by referring to the schedule he has had for the day), a new action movie trailer pops out (if his interests are action movies). If for example, motion (Deniz, 2007) of the user which is context information can be detected and the IPTV knows when the user wakes up (context can be time also), IPTV can switch ON automatically and an application can show his recent unread mails or RSS Feed of his favorite news sites.

Experience can be enriched if the activity of the user can be identified. For example, if the user is ironing his clothes in front of the TV, an advertisement pops up on his screen regarding dress sales in the area he stays by figuring out his location information which is context information. Communication can also be an important context and can convey information regarding the user's comfort with the other people in the room. If the IPTV can recognize this context information, it can help in having a better shared experience between users in the room. If the relationship between the people in the room can be found out, experience is much better as privacy is still a major concern and can be set appropriately by the user. It will definitely enrich the experience a context aware IPTV can provide.

Even contact or distance between the users in the room shows the comfort level between them. If there is no communication between the users and if they are sitting far away, the privacy of the session has to be controlled. A user would not want his private photos to be viewable when he is sitting with someone with whom he is not comfortable sharing his personal life. Other context information can also be on what the user is resting on, what are the objects he is in contact with. If for example the user is holding a gaming device and using Wi-Fi to play with his friend online, a popup can be displayed on the TV after the game. It can show the current review or the rating of the game he was playing or the new games available.

The current session is another important context parameter available for the IPTV application. Information such as current channel the user is watching, language being spoken in the program and genre of the program are vital information. All this information can be obtained from the EPG or the web (XMLTV, 2009). Information such as the advertisements the user viewed or skipped, events the user is attending, purchases made helps in building a profile of the user. The profile consists of the user's likes/dislikes and other information. Context such as time of usage can also play a very important role. On checking the previous log or session of a user, if a conclusion can be reached that the user watches National Geography at 8:30 PM on all Fridays, it can be automatically recorded so that he could come and view it later on.

Context information can keep changing per session and per user. It depends on what mood the user is in or what he watches. Sometimes the ratings the user gives for a movie or a program he just watched helps in building up the profile which the recommendation manager requires to enhance the experience of the user. Every session counts in building up the profile and with time, a pattern of viewing comes into existence. It only changes in time as the viewing nature of the

user changes. This can also be termed as the navigation history. Navigation history can include information such as the channel the user keeps watching and his reaction during advertisements (if the user continues watching or changes the channels). Looking at the navigation history, higher levels of information can be found out like the user's personality, interaction, type of usage (if the user watches intentionally or in a browsing mode), likes/dislikes (compare the genre in the EPG) and the ratings.

3.2.1 Sensors which can be used:

Emotion can easily (Reynolds, 2009) be identified if the IPTV has a camera attached to it by using face recognition. Also voice sensors can help identifying emotions. Sensors can determine the mood of the user by using the tone of the spoken voice as input. Voice can be easily recorded using microphones or amplifiers. The posture (Biswas, 2008) of a user can be found out by positioning sensors or wearable sensor networks. Positioning sensors can determine the objects in the room. To determine which objects are being held or closeness of the objects to the TV or user, capacitive or proximity sensors (Wiki proximity, 2009) can be used. There are different sensors available for checking if the user is looking at the TV or not by using vision sensors or cameras. Touch and weight can be easily found out as well. Location can be found out by GPS or cellular base stations information. Force applied on objects can be found out by using force sensitive sensors and can be done on conductive surfaces.

Weight of a user and objects can be easily found out by using load sensors. They can also help in identifying some movements of the user like standing up, sitting etc. Motion is usually detected using Passive Infra Red sensor (Wiki PIR, 2009). For common activities such as eating or drinking, there are sensors called as Gas sensors which determine heat in the room or show any change in temperature in the room compared to the overall temperature in the room. These can determine for example if the user is drinking coffee or beer and react accordingly. Heart rate can be found using Non-Invasive sensors, which can also determine higher level context information such as emotions (excitement, sadness etc). User mood can also be found by using skin resistance sensors. To determine any physical activities by the user there are sensors to determine muscle tension. There might be no application which could use multiple context information at the same time. But there is higher level context information which can be used in most of the applications such as emotions, recent history or current IPTV session. We are assuming the availability of context information and checking if it can benefit applications or IPTV as a whole.

3.3 Environment Context Information

Environment can be categorized into where the user is or what is around the device. Noise in or around the TV can be crucial information to identify the number of the people in the room, the volume of the TV compared to the communication in the room and also the noise in the room compared to the noise outside the room. Noise frequencies can be categorized into communication or music. It is also important to find the number of people in the room and how they are related to the user. Objects around the place can also be context information. Some

objects may be movable others being immovable. If the objects can be identified, recommendations can be made. Example: Paintings hung on the wall. Appliances or objects can have some status of it, for example, if the air conditioner, heating system is ON or OFF.

Distance of the user from the TV is an important context parameter. Height of the objects/TV can also be a parameter. It does not make sense to show a user's information on a TV which he himself finds it difficult to view from the place he is sitting. Current lighting condition within the room can be altered to enhance the mood of the user.

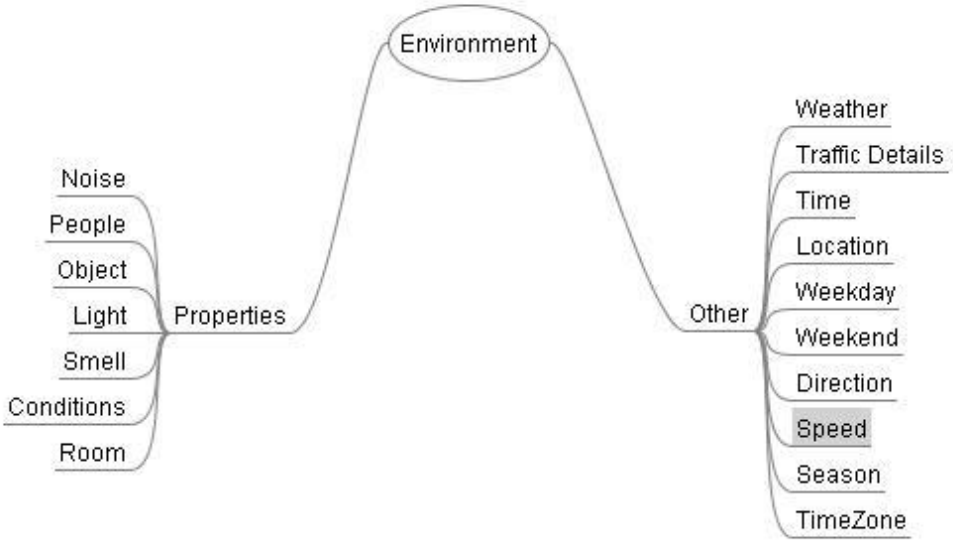


Figure 3.4: *Environment context information*

Shared experience can be enhanced by varying the lighting to the same conditions at the homes of the respective users. Properties of the light can also be tested such as intensity, density, reflection, color etc. The other properties of the room include smell, and various conditions such as humidity, brightness and temperature. The TV can be personalized depending on the type of room which can be either public/ private, business/pleasure, clean/dirty and spacious/small. No user would want his personal information being popped on a public TV. Other context information which cannot be generalized is the current weather details or any sudden weather changes.

If the IPTV application is synched with the Traffic details and the calendar, it can help in time approximation. Current Time and Time Zone is another context parameter which can be important from the user point of view. It is very important that the time of the user's be compared before sending a request to one another. It is not a good idea to send a request of watching a movie together at 8 PM with a friend who stays in the other part of the world who is

just about to sleep. Location information can be used by some applications like market place or local event applications etc. Activities over the weekdays are more consistent to the activities on weekends. There can be applications developed which use past information and derive rules like, it is mostly the case that user is not present from 9 AM to 5 PM at home on Weekdays.

It is also possible to see the direction or speed the user is travelling at and can help gather information. Season information sometimes leads to change in user activities. There is no point giving information on recommendations of activities like going to a beach on a rainy day or giving advertisements on buying skiing accessories during summer.

3.3.1 Sensors which can be used:

To identify the objects around, positioning sensors can be used. There are algorithms to recognize shapes. A camera mounted on the TV or in the room can identify the proximity between users and also recognize face. Noise from within the room can be identified using microphones. Smaller level context information such as pitch can be derived from Noise which helps in distinguishing between music and communication. Light inside and outside the room can be found out by using sensors such as UV and IR sensors. Air pressure changes when the door/window is opened, humidity increases with more number of people inside the room.

3.4: Summary

In this chapter, we identified the context parameters valid and gave examples to most of them. We mentioned the sensors needed to identify context information. We clustered and filtered the context parameters based on selection criteria such as level of abstraction, re-usability, functional grouping etc. We classified it into three main categories - device, user and environment. We researched on possible applications for television and backtracked to investigate on the valid context information which could enhance the services of the applications.

Chapter 4: Design and Modeling

It is not trivial (Keivanloo, 2008) to build a context aware system. Therefore we had to investigate the design process for building these applications. The design process we used is as follows,

1. Specification
   - Specify the problem being addressed
   - Identification of context information available in the IPTV domain [Chapter 3]
   - Evaluation of the most important context information for IPTV domain.
2. Acquisition and Reception
   - Identify the data the sensors provide for the selected context information[Chapter 3]
   - Store this gathered context [Input from GUI to Context a context sink using the Context Management Framework]
3. Delivery
   - Provide Methods to deliver this context information to applications [iNEM4U]
4. Action
   - Receive context and use it to adapt the applications behavior

4.1 Selected High Level Context Information

After conducting a few brainstorming sessions with a group of selected specialists in this domain, we came up with the chosen high level context information or useful sources of information. Illustrated below is the list of context sources which could be useful for the television domain. The location of the system was chosen to be one of important context information. The service provided by any application can be drastically improved if this information is known. One example which uses location as context information is news widget which provides location based news, for that area etc.

The current session in the device was considered as very important information. Information from the system such as the program being watched, the remote usage and occurrence of ad breaks is useful to the context aware applications to provide context aware services. Information available from the user such as, the intention, mood and current activity could be used effectively to improve services.

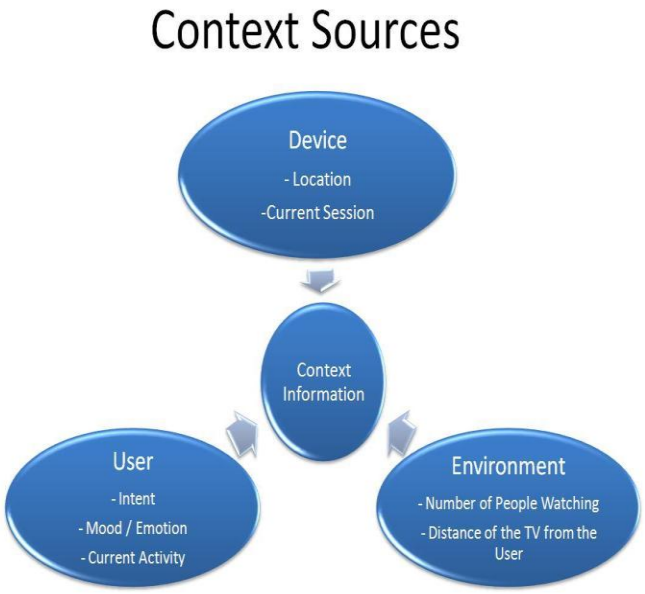


Figure 4.1: *Selected Context Information*

The information stating the number of people watching can determine the privacy levels to be set and also provide joint recommendations looking at the user profiles. The distance from the TV is another important piece of information to show the size of the content the widget displays. Some of these selected context information will be used in the widgets we created. The higher level of context information would include location (if mobile devices are taken into consideration while providing services), intent (can be synched with calendar to check the schedule of user), emotion or mood (from multiple sensors such as voice and face recognition, time) and current session.

4.2 Architecture of a Context Aware Widget

Below is the architecture we used to develop the widgets. The inputs can be from various sensors which act as context sources and provide context input to the application. Context events such as remote zapping, ad breaks are all monitored such that various personal applications can be displayed on the screen. Information from the television such as the title of the program being watched, genre of the current watched program can be the input to the applications. On context events occurring or valid context information, the widget application provides service to the TV.

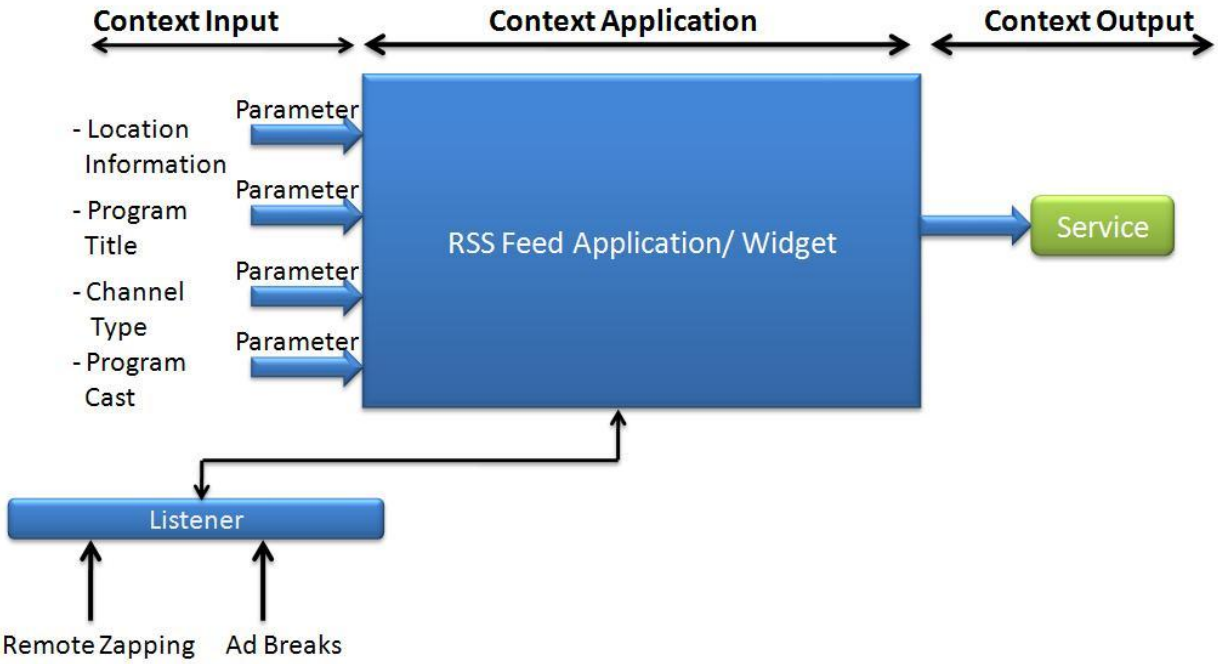


Figure 4.2: *Architecture of context aware IPTV widget*

4.3 Architecture of IPTV with Context Information

Building Blocks of the Context Aware IPTV System are as follows,

1) Context Sensors/Sources

Sensor is a device which converts a signal into a machine readable form which can be given as input to the TV. Sensors can be present on the user, in the environment or in the device which retrieve context information from them. These sensors can be placed on the user such as microphones or can be placed in the environment like noise sensors. Once obtained, this information represents a context source. Different context information can be derived from these sensor readings such as Pitch and Volume from Noise. Such information helps in distinguishing type of noises like music or communication. There are different types of sensors for Proximity, lighting, density, environment etc. Only sensors which obtain information important for an IPTV application are used. These sensors detect the values and the context source reports this value to the applications which have registered interest in it.

2) Environmental Model

This module collects all the information about the environment which is represented in XML in CMF. It obtains signals from the sensors, analyses it and converts it into a machine readable form. Information obtained from sensors can be combined to form higher level context

information. This module receives information continuously. It monitors changes in the input and sends the information to the context management module if required.

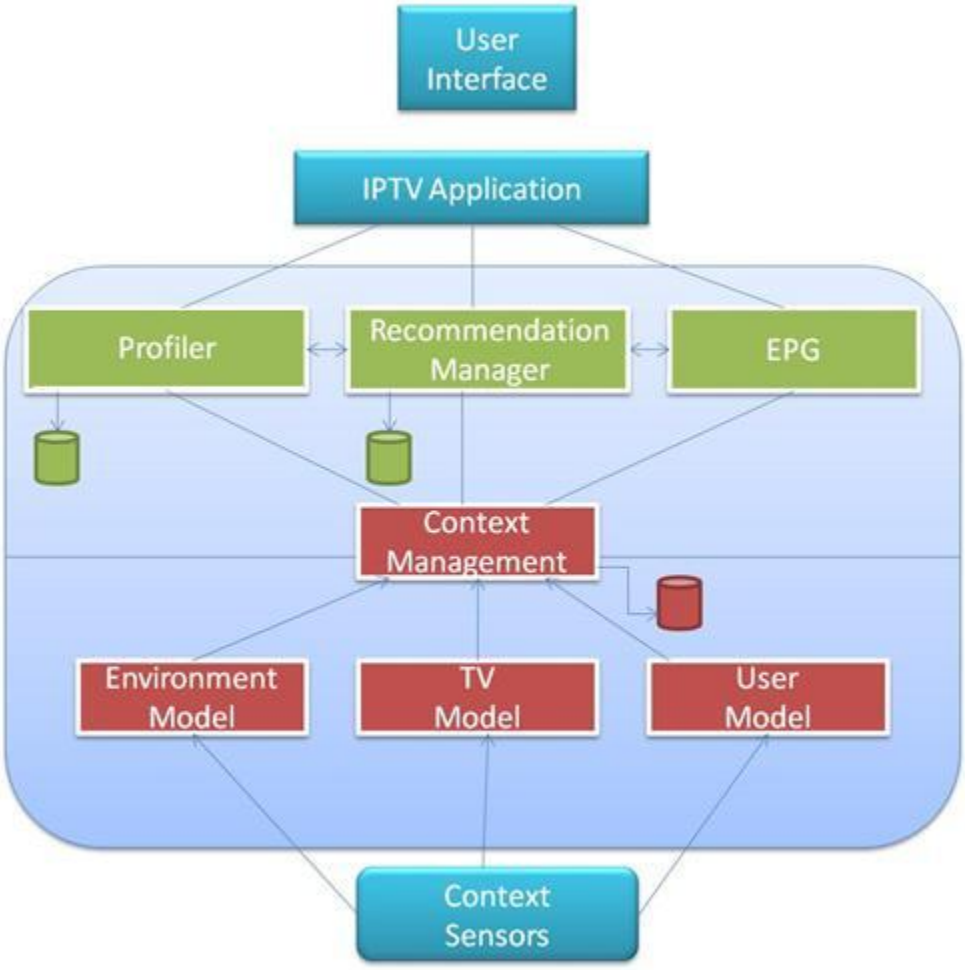


Figure 4.3: *Design of the Context Aware IPTV system*

3) User Model

This module collects information from the sensors placed on the user or around the user to determine the user's current activities or user's state. Context information such as what the user is in contact with, what the user is doing, what is his current pulse and body temperature etc is kept in this model. This module holds all the necessary sensor outputs and analyzes it and produces higher level context information if necessary. Example: For determining Mood or Emotion of a user, more than one context sources could be used to determine like a microphone to determine voice and also camera for face recognition and also a skin resistance sensor. All relevant information from the user is placed in this model. There is no interaction between the three models here. The context information from the other modules is shared at the context

management level. The User model holds only the current context information and sends the necessary information to the context management if needed.

4) TV Model

We also refer it to Device model. This module consists of information related to the current session of the user. The context information is derived from the user's interaction with the system. It also involves his usage with the other devices. It is more of logging user's activities with the TV so that it can help in predicting the user's behavior or intent or can be used as context for this session or the next. The user's navigation plays an important role in decision making. The device's such as mobile etc the user possesses also plays an important role. Devices provide context to this TV model such as location's visited and sometime's context information which does not change for a long time such as screen size, software, connectivity etc. All this information is sent to the context management module. This information can be easily got from the web. There are a set of utilities like the XMLTV (XMLTV, 2009) which stores the XML listings for the country chosen as per the current time. Below is the XML example of one of the channel listings,

```
<programme start="20090721063000 +0200" stop="20090721082000 +0200"
channel="24" >
    <title lang="nl">Cassandra's Dream</title>
    <desc lang="nl"> Thriller The brothers Ian and Terry from London urgently
need money. The one in his gokverslaving expenses and another for his dream
as a businessman to achieve. When their rich uncle Howard is proposing to
help them, they both have their future positive attitude. The brothers are
not just the money: there is an unusual request opposite. It conscience of
Ian and Terry, this is put to the test and they both react differently to
the desc>
    <credits>
       <actor>Colin Farrell</actor>
       <actor>Ewan McGregor</actor>
       <actor>Sally Hawkins</actor>
       <actor>Hayley Atwell</actor>
    </credits>
    <date>2007</date>
    <category>Crime/Mystery</category>
  </programme>
```

It provides information like title of the channel, description, starting time, ending time and other information like categories. Category and ratings can help in generating recommendation if required. Title information and category displayed in the above example is the information required for us to implement our context aware applications. These listings can be synched with the channel being currently watched and information about the program is readily available to

the context aware applications. Such kind of information has to be acquired from the television which is the current context of the TV.

5) Context Management

Context management framework is also referred to as CMF. This module acts like the central system which works in a publish/subscribe mechanism. The Electronic Program Guide, Recommendation manager and Profiler subscribe for information from this module. Recommendation manager is used for providing recommendations for the user based on the current contest stored in the CMF. Context Management subscribes for context information from the 3 models environment, user and TV. It uses this information and publishes it to any of the other modules requesting for it. It saves the context information for a period of time in a database for applications which need earlier context information by using the logging technique.

Context Management helps in building up a profile of the user by using logging technique. It stores all the context information obtained from all the models and stores it in a log file.

CMF supports the use of reasoning engines within this module. The context information comes to this module as context. This engine helps in transforming lower level context information to a higher level context. Context reasoning is needed to validate the sensed data which lead to actions. Some of the reasoning techniques are ontological, rule based reasoning, probabilistic reasoning etc.

6) Profiler

This module looks at the information present in the context management module. It builds up a profile of the user looking at the context information in the context management module. It analyses context over a long period of time and creates a profile which the recommendation manager uses to provide recommendations. Example: If a user watches action movies all the time, he has a liking towards the genre action. This information will be saved in the profile. This profile information is stored locally and can also be stored in a server because it should be allowed to be accessed from any IPTV device.

7) Recommendation Manager

This is a predictive content recommendation system which looks at the profiler information of the user and compares it with the EPG and gives user some recommendations. It provides recommendation only when certain context information (user is bored, user is eating) is valid. It can give recommendation to use the IPTV services or run applications. It uses the profile of the user that is being built up by the profiler. It helps the user to select the content which matches his needs and interests. It helps in generating recommendations based on implicit/explicit ratings, content metadata and context information. It has a database to store the feedback of the user for

the recommendation it provided which acts as context information again. It can compares profiles of multiple users.

8) EPG

This is a digital guide to the broadcasted channels and programs on the TV. The recommendation manager uses the EPG to give recommendations to the user. It compares the already built up profile to the EPG information and ranks it accordingly. The one with the highest match is gives out as a recommendation. It is one of the services the recommendation manager uses for its working.

9) IPTV application & User Interface

This is the application which runs on top of the IPTV framework in which a digital television service is provided using IP over a network. These applications require context information. These applications adapt to the current situation like user status, user mood, location etc. It uses the services of IPTV to get information needed for its functioning. Some applications need hi-speed data to be transferred to function properly. Most of the applications are related to education, finance and leisure. The applications are JS/CE-HTML (Lowet, 2008) based and the JavaScript objects are recommendation Manager, Profiler and EPG. Methods have to be invoked on these objects and callback handlers have to be defined.

4.3 Flow of Context in IPTV system

Example: Consider a user who just arrived home from his work. He switches on the TV. By using face recognition, the user is identified. Sensors such as voice sensors installed in the room picks up the user's voice and analyses it to detect emotions and also compares it with the emotion detection sensors (camera). Remote usage can also determine user's emotion. Camera in the environment can also identify the number of people in the room. Information such as current location is the location of the device which the user is using which is the TV here and stored in the device module.

Context Information such as distance of the user from the TV and number of people on the room can be stored in the environmental module (if required). Emotion information is sent to the user module which is the information acquired from noise sensors. The Context Management module uses the information from noise sensors and face recognition from the camera. It can apply reasoning techniques and come to a conclusion that the current emotion of the user is "bored".

The context management framework also has the ability to build a profile looking at the log file of the current context sources. However, a profiler can be built separately such that more services can be provided as the profiler option of CMF is very limited. Context information such as location and emotion which is consistent over a period of time is saved in the profiler. On identifying the user in a bored state, the recommendation checks the profiler for the likes/dislikes

in the user's profile or events in the area and compares it with the EPG or some applications and gives some recommendations accordingly.

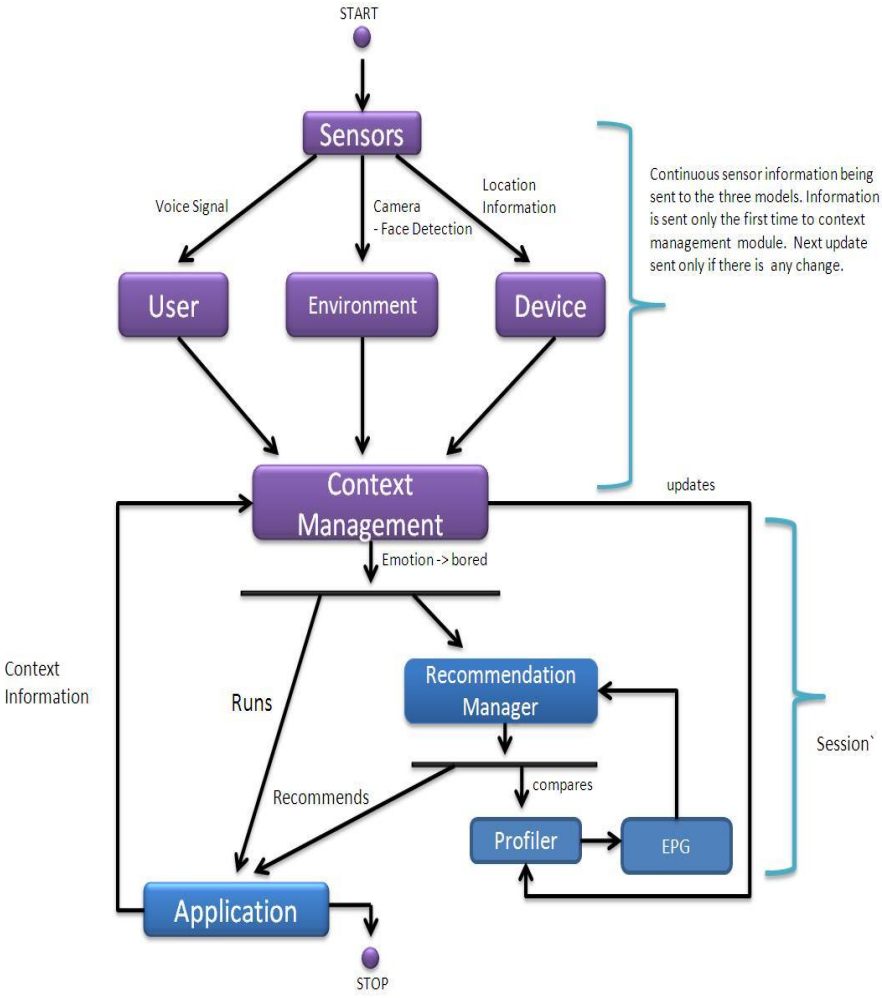


Figure 4.4: *Flow of context information Emotion in the IPTV system*

The Recommendation manager also compares the online friends and suggests for applications that can be used. Every channel viewed, skipped in a TV session etc is monitored and acts as a context information. If the user is on a casual browsing mode on the TV, randomly changing channels, this information is also stored in the user profile as the next time the user does this, a recommendation can be used to enhance the experience of the user.

There are digital signal processing units within the sensors which generate values to this context source in human readable form depending on values assigned to it. This information is managed by the context management framework. If the emotion is set to happy while watching a certain program, applications such as real time information generating context aware applications can be triggered. Implementation of a reasoning engine is a possibility to help trigger certain widget at

appropriate times. It is important to note that any kind of interaction with the system can be triggered as context information, but it is up to the system designer to decide on which is the most appropriate.

Applications request context information via proxy to the CMF and get triggered on changes in context information. Some of the widgets, we developed needs information from the EPG or from the web as it provides information regarding the channel to provide better real time results in the context aware applications we built.

4.4 Summary

In this chapter we looked into the architecture of our context aware widgets. These widgets request information from the context sink through the context management framework. The context information provided into the context aware widgets were the context information we derived by looking into each of its use, re-usability and other criteria's. The list of selected context information is mentioned in the beginning of the chapter. We also mentioned the design procedure we followed in this project. We designed the high level architecture of the context aware IPTV system and mentioned functionalities for each of the modules.

Chapter 5: Implementation of Context Aware IPTV Widgets

5.1 Context source development

The CMF [Section 2.4] contains a lot of functionality and base classes to facilitate the writing of context sources. The first step is to create new context sources in CMF depending on the number of the selected context sources for that system or application. Since we are working on a TV platform, a few of the high level context sources useful for the television has been selected prior and will be implemented in this CMF. This section we will look into the initial steps of creating a context source, how to publish the context sources and to retrieve values from it. The implementation of the GUI which is the simulated context source values are actually sensor values from the environment or user or the system itself. A CMF container does not publish data by default thus it has to be specified which context sources have to be published. CMF requires Java Runtime Environment 1.4.2 or higher. The first step in creating a context source is by creating a new source class below this directory (nl.telin.cmf.impl.wrapper). There are separate packages for operation system specific sources (in nl.telin.cmf.impl.wrapper.<os>) such as Windows, Symbian, etc. It is necessary to make the source extend a base class that implements the core wrapper functionality. The first choice is to use the BaseWrapper source class. It provides ContextSource interface through which context information can be obtained or ContextSubscription interface which deals with context source registration and notifications. The other option is to use ThreadedWrapper source class which is a basic context source that periodically checks for context updates. The newly created context sources are made to extend the BaseWrapper class as the use of periodically checking the context updates was not required. The source has to be called in the container which shows its availability to the system.

Below are the steps involved in creating a context source in CMF.

1. **Implementation of source specific initialization**

First the source has to be initialized. It is done when the component exists in a container.

2. **Implementation source specific configuration**

This is called just before starting a component, after reading settings either from a configuration file or configuration graphical user interface.

3. **Override the getProperties method**

We first create a properties bean, which contains all source specific properties. This is done by extending the appropriate PropertiesBean subclass. In our case we extend it to BaseWrapper.

If the context source is subscription based, it should be extended to Threaded Wrapper. As all the context sources we are going to create will not have subscriptions, it will just be displayed on polled or requested. Since there are no source specific properties we can use an instance of the appropriate PropertiesBean subclass.

The settings of the newly created source will automatically be stored in the Context Source Registry. It uses the file publishedcontextsources.xml for that. Every created context source will have its own settings, sometimes with common attributes. Properties can be added into this and its validity can also be checked.

4. **Addition of Source specific attributes**

If the context source requires any specific attributes, it is published into the context source registry. On calling the super method, the attribute is made common to all the context sources.For example, attrTable.put("user", new NameValuePair("user", "Philips")), this would add the value as Philips being the user of the system for all the context sources.

5. **Implementation of the source specific cleanup**

The cleanup is called just before restarting or shutting down a component. This is to free any resources claimed by the component.

6. **Make the source return its context.**

In reality the values of these context sources comes from sensors. Now as we have assumed the existence of sensors, we are replacing it to take values from a GUI. This source becomes a client to the CMF.

Usually first the context is created based on the parameters, and then the query is executed on that context to produce the final result. The returned string here is the value of the current context source. If for example the created context source is currently watching, then the value of the returned context string from the context source is the title of the program being watched which can be got easily from the web or from the EPG. If for example, there is sensor which indicates whether the user is watching TV or not, if the value of the context source is requested when the user is watching the TV, the returned string is “yes”. If the sensor detects that the user is not watching TV then the value of this context source returns “no”. All of this input is given from a GUI which can be continuously changed and the values will be changed within the CMF. The GUI updates the context sink which is accessible to the CMF with the values matching that of the context sources created. The context sources created are Emotion, No of People, Watching, BrowsingMode, and Channel.

7. Using the source

For clients to be able to use the source we have to:

1. Specify the ontology of the context that the wrapper produces. We have used string as the output of context sources.
2. Specify the dependencies of the source in a new dependency file.
   Add the newly created dependencies to the list of dependencies.

8. Dependency files

These files determine which components are included during the build and register the available context sources. The syntax of the dependency files are

```
<dependency>
        <include>..</include>
        <jar>...</jar>
        <files>...</files>
        <entry>..</entry>
</dependency>
```

The entry tag shows that the description within it describes that of a context source. For the context source being included during runtime it has to be defined within the dependency file. Below is the how a context source is made to run along with other components.

```
<entry>
        <name>Emotion</name>
        <component>eu.inem4u.cmf.tutorial.Emotion</component>
        <description>Example context source from the TV.</description>
        <settings>
        </settings>
        <autostart>true</autostart>
</entry>
```

Here the name element displays the name of the context source. In this project, we created 5 context sources namely Emotion, WatchingTv, Channel, BrowsingMode, and NoOfPeople. Component refers to the java file created to make a new context source which was explained in

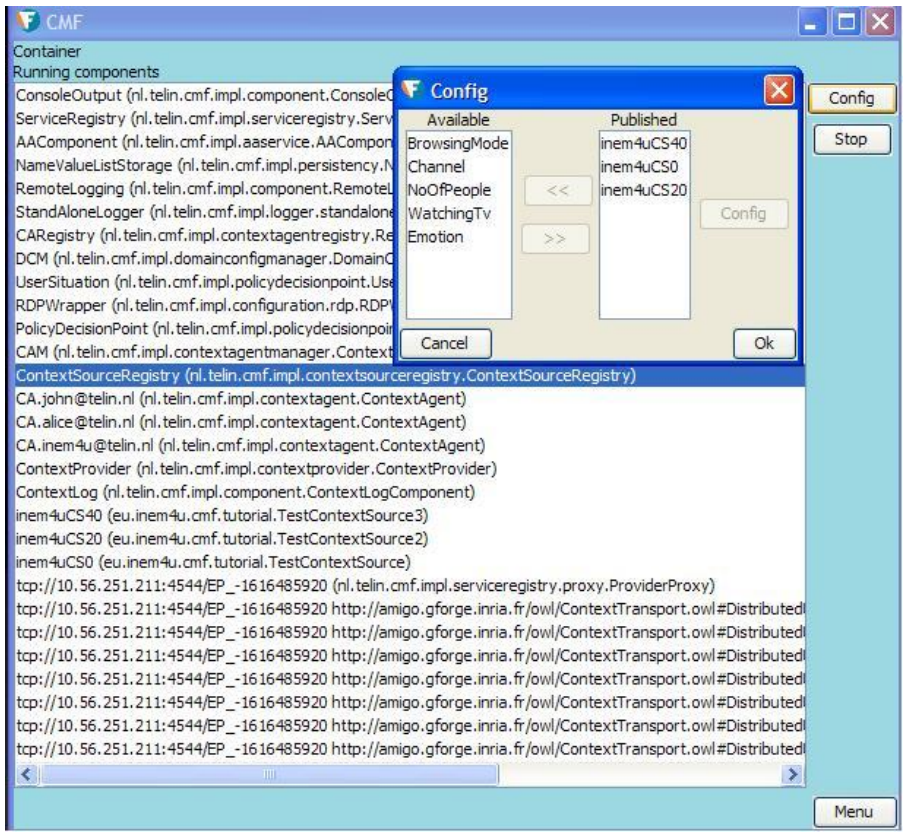


Figure 5.1: *Publish context sources*

the sections earlier. Description settings are just for human readable information. On creating the context sources, as they do not get published automatically, it has to be done manually once. ContextSourceRegistry displays all the available context sources in our case is the five new context sources we have created.

As all the five sources are required for us, we need to publish the context sources just created and move them from available to publish list. This list can always be modified and the main control lies in this container. At any moment, the already published context sources can be removed from the available list.

5.2 Context Source Tester

Context Source registry handles most of the work for a context needing system. From the figure below, the components are as follows,

- Service Registry – Used to connect to the service registry.
- Context Brokers – Shows all the context brokers registered at the service registry.

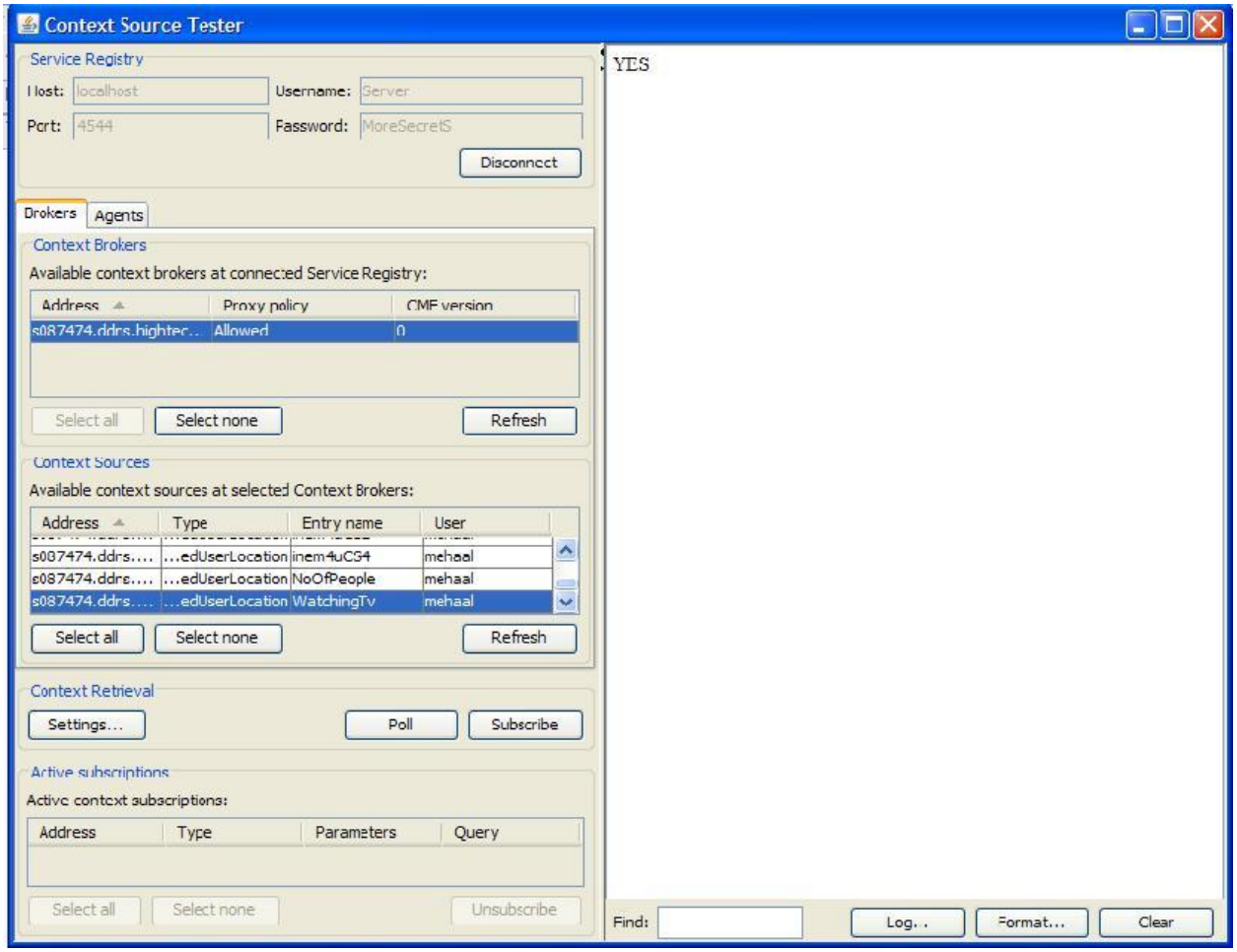


Figure 5.2: *Context tester GUI*

- Context Sources – Lists all the context sources registered at context brokers.
- Context Retrieval – Retrieves the value of the context sensors by polling
- Active subscriptions – Is not within the scope of this project
- Output Panel – Shows the chosen context source's values.

We connect to the service registry by entering the host name and the port on which the service registry is running. We also need to provide the username and password to connect to this registry. Context broker shows the name of the computer on which the broker is running with the proxy policy and the CMF version.

The context sources have values address, the type of context source, currently named to the default one as the type is not created, the entry name which shows the name of the context source and the user using the system. In context retrieval, chosen context source from the available context sources is polled to get the value of it which is displayed in the output panel. In this example, the WatchingTv context source is polled and the value is got as "yes". These are actually values "yes" or "no" from the face recognition cameras which lets the system know if the user is looking at the camera or not but in this case taken from the GUI.

5.3 Context Aware RSS TV widgets

We conducted various brainstorming sessions to decide on applications that could be used on a TV platform. We narrowed down to the use of the most common web 2.0 applications (Webware, 2008) among users based on popularity. These applications could be used on some occurrence of a context event or applications which could be made context aware. The frontend of the Widget applications have been developed in flash and we have used Actionscript 3.0 to parse the XML result from the feed generated. The parsed result is displayed in the applications using lists or text area on the front end. We shall first talk about context aware widgets.

5.3.1 Context Aware Twitter Search

We first implemented the Twitter Search RSS application. It displays contents on the twitter search application matching the current content information being watched on television which is the context information. The idea behind this is to generate a keyword of any of the following with respect to the program being watched i.e. the actor in the current programme, the title of the programme or the actress of the programme or the genre.. If the application knows the current program the user is watching, all we need to do is use one of the available information and use it to generate RSS feeds.

Twitter Search is a real time search website which means it return backs the content which matches the input in real time. The resultant information will be the recent most update by users of twitter on the internet. It returns back the tweets that match the specified input. To make it context aware, we use the twitter search API to search the chosen attribute of the currently watched channel. It returns back with the recent most tweet matching the input query.

Figure 5.3: *Context aware twitter search widget*

The search API we used is,
http://search.twitter.com/search.atom?q=twitter

which gives an Atom XML feed as the result. We parsed it using the technique of XML parsing in Actionscript 3.0 to generate the feeds in the display box using flash components.

As the channel being watched is a comedy central channel, it is playing the stand up comedy of a well known comedian. On the system realising that the user is interested in this channel and has been watching this from a long time, it shows the widget.

From the example, the search term is the actor in the current watched programme who is "Russell Peters". The results generated are formatted and placed to separate different results. If integrated with the already existing TV-browser framework, the tweets with links within the text area of the twitter search widget can be opened in a browser with the video background being muted if necessary. The best results are generated if the title is usually put as a query.

5.3.2 Context Aware YouTube Search

We wanted to implement an application that provides video content rather than just text. We decided to use the YouTube API to perform query based search to generate a video list. YouTube has many RSS links but the most important we decided for the TV platform is the one that can search based on the recent most tagged or uploaded title matching the keyword. The keyword is obtained from the current channel being watched and used as a search query. It generates results in XML format and we use XML parsing technique in Actionscript to generate the result shown on the GUI. If implemented in the SocialTV framework (Lowet, 2008), on clicking these videos, the video in the background can be muted and the chosen YouTube video could be played.

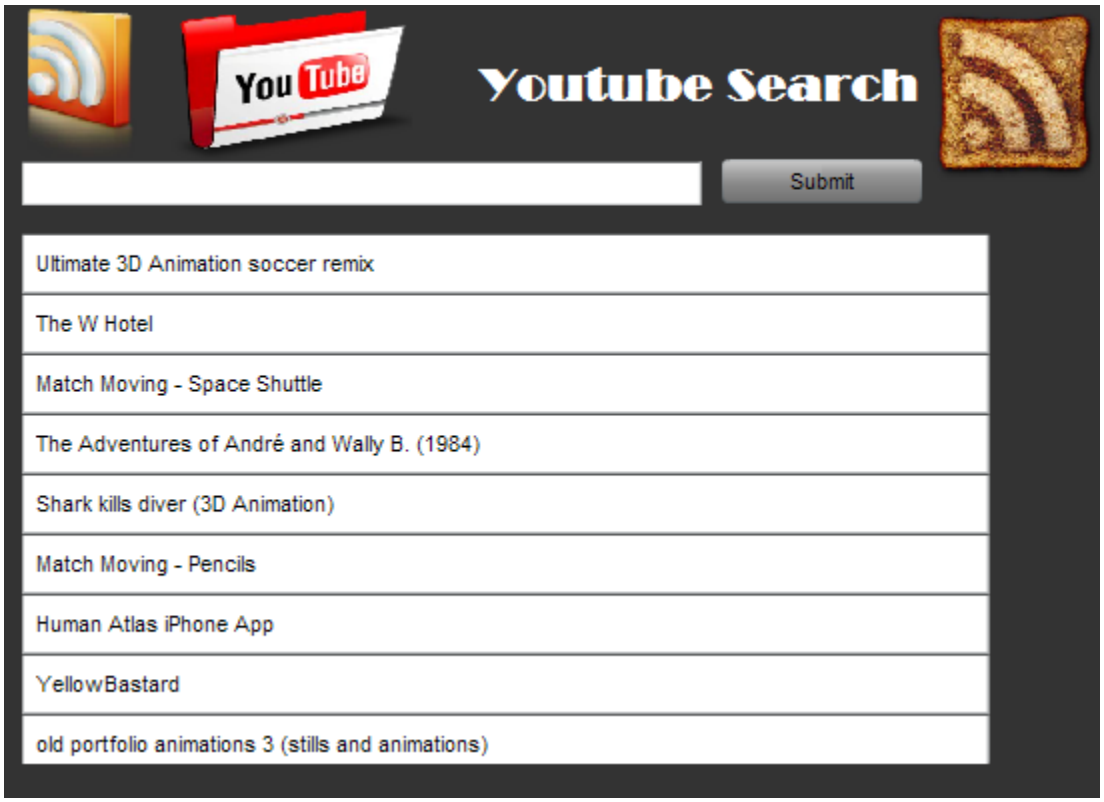


Figure 5.4: *Context Aware YouTube Search widget*

In this above figure, as the channel being watched is a 3D animation movie and is of the genre "animation", it searches in YouTube for the most recent tagged and uploaded title matching animation and provides result in a RSS 2.0 format. Additional functionalities such as generation of thumbnails on click have been implemented too. Our emphasis was given towards the content displayed on the Television. We used the YouTube search API and generated RSS results, *http://www.youtube.com/rss/tag/keyword.rss*

YouTube uses the RSS 2.0 format to show the feeds and the widget above parses the XML to build a context aware application. These widgets are meant to only appear when the concentration level is high. This can be done by checking if the user is watching the TV or evaluating the remote usage.

The other widgets we have developed is more for personal use but appear only on context events such as user not watching TV or on appearance of commercial breaks or on continuous zapping.

5.3.3 Yahoo Unread Mail

Yahoo Unread Mail feed shows the recent most unread emails in the inbox. This application should be displayed only when there is a single user in the room. Authentication can be done using RFID badges or camera recognition. Yahoo feed supports feeds in both ATOM and RSS 2.0 format. We used a system called Ymail Feed to convert the inbox into an XML format and parsed it using the Actionscript. We used RSS 2.0 to develop this widget. For a configured account, the Atom feed of an Yahoo mail unread email inbox looks like this,

Figure 5.5: *Yahoo unread mail widget*

ATOM Link:
*http://ymailfeed.appspot.com/atom/3481f98b9ebb99556c72c5977bb33e5f/47e93d9edddc591207c0c7877627ee72*The RSS feed of an configured Yahoo mail unread inbox looks like this,

RSS 2.0 Link:
http://ymailfeed.appspot.com/rss/3481f98b9ebb99556c72c5977bb33e5f/47e93d9edddc591207c0c7877627ee72

It shows the name of the user who sent the message along with the subject of the email. On clicking the retrieved mail, it shows the user, the received time of the email. A lot of additional features can be implemented in these widgets. But the aim of this project was to show how context makes a difference and how it can enrich and user's experience.

The link to generate the inbox of a gmail inbox is,
http://gmailusername:password@gmail.com/gmail/feed/atom/

Gmail always requests the user to enter the username and password which does not make sense to use as an RSS feed because instead of logging to just get the RSS info, it would be better to log into Gmail itself. We implemented Yahoo Mail widget as no authentication is required, therefore making it possible to display during a short advertisement break or any of the other valid context event without the trouble of login each time.

5.3.4 Subscribed Blog Widget

As bloggers and blog followers are increasing at a fast pace (Blog, 2008), we decided to develop a widget able to read blog updates. We decided to build a widget which can access both the highest used blog sites over the internet. These are meant to be personal widgets that appear on events such as advertisements or if the user is bored. The most widely used over the internet are *www.blogspot.com* and *www.wordpress.com* constituting more than 80 % of the blogs used on the web.

As Blogspot has a RSS 2.0 feed and Wordpress has a ATOM format, we parsed both using XML parsing in Actionscript. As we were the users of the TV system, it was configured to view the blogs we were subscribed too.

Figure 5.6: *Subscribed blog widget*

Below are the links to generate the feeds of the two major blog services,

http://www.blogaddress.blogspot.com/feeds/posts/default?alt=rss
http://www.blogaddress.wordpress.com/feed

The term blog address is the web address which is configurable by the user of the system. On executing this link with the appropriate address, it returns a list in XML format which can be displayed using Flash components like lists, text etc. We designed such that the last five posts in each of the subscribed blog addresses were shown.

5.3.5 YouTube Highest Viewed and Discussed widgets

These widgets were developed to appear during context events such as advertisements and context information such as user being bored. YouTube is a social media platform in which the user can view what he wants. So it is obvious that whatever the user chooses to view in the TV is content that he is interested in. It enlightens the user's mood and this is what context aware system is meant for, to display the services automatically.

YouTube's most used RSS information on different sites is that of the highest viewed and the most discussed for the day. It is no point displaying the highest viewed since all time, as it does not change that often.

Figure 5.7: *YouTube highest discussed widget*

Figure 5.8: *YouTube highest viewed widget*

We used flash development components and ActionScript to build these widgets. We used the icons of web 2.0 applications to make it look appealing and attractive on TV. Below are the RSS 2.0 links of the XML list we have parsed to display as list items in the GUI of the widgets.

http://www.YouTube.com/rss/global/top_viewed_today.rss
http://www.YouTube.com/rss/global/most_discussed_today.rss

5.3.6 Twitter Status Updates

As twitter is being widely used, we felt the necessity to implement it as a widget in Television. Since this is a micro blogging application, it acts as a way to share experiences about current activities. It can help in enhancing experiences on television as users might tweet about the programs they liked and current activities on TV. Some of the features in this can only be understood if the user has been using twitter application. The twitter search widget is different from the twitter status widget. Twitter site is used to update the status and twitter search is used to search for a specific keyword and generate results in real time from all the users using twitter.
It is been designed in a way to show the last four updates of the current user of the system. These personal widgets appear only during context events such as continuous zapping of channels or during the advertisements.

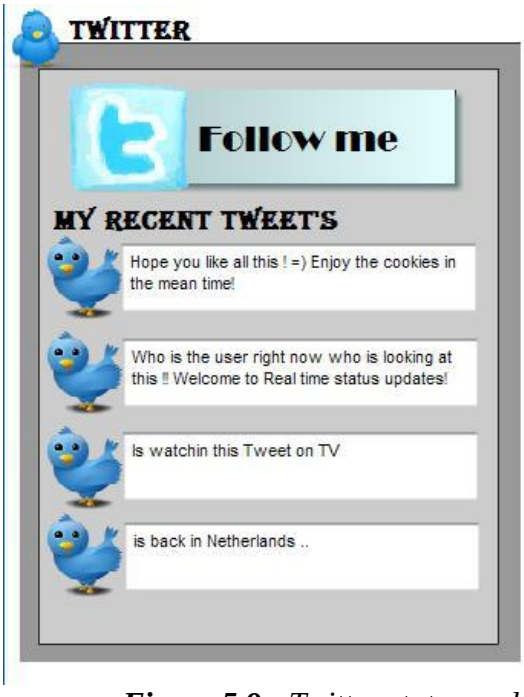


Figure 5.9: *Twitter status update widget*

This widget has been implemented with some features like on the click of a button (follow me) will take you to the twitter site where the user can update his status on login and the change is shown on the twitter widget if it appears again. If another user is using the TV, the user can click on follow me button and can follow the user to view his update thereafter. The format twitter allows for parsing status updates are JSON, RSS and Atom. We used the RSS technique to generate the list of status updates as a XML list to view them on the GUI of this widget.
Below is the link that generates the RSS of a specific user.

http://twitter.com/statuses/user_timeline.xml?screen_name=mehaalrai

5.4 GUI, CMF, Browser – Integration

We used a context sink to hold all values between CMF and context widgets. Any changes in the context information are updated in a context sink. We use this context sink to analyze the current context which is retrieved by the CMF. Now based on the current values in CMF, the appropriate widgets were displayed.

Figure 5.10: *Context Information GUI*

For example, if users emotion is sad and there are either many or one person, and not interested in watching TV, the widget of YouTube search highest viewed appears. Another example is if user's emotion is sad and there is one person and he is interested in watching TV, then the widget of twitter is displayed. This selection of widgets is done by reasoning of available context information within the CMF.

5.5 Summary

In this section, we showed the implementation of context sources in CMF and the integration with browser. We described the use of a context sink to store the current context. We described the technologies used in the developed context aware widgets of web2.0 applications. We developed two widgets which make use of context information and the others occur on context events.

Chapter 6: User Test and Analysis

A user test was conducted to get feedback on the implemented widgets. A total number of 17 users were invited to participate for this test. A pilot test was conducted to identify the problems if any and corrected before the actually user tests started. The first two tests were considered as Pilot Tests. The participants involved in the initial pilot tests were colleagues who had knowledge in the IPTV domain. The experiment lasted for about 25 minutes with each user having a 10 minute session viewing the widgets on TV.

The users were told to look at the widgets appearing on TV and investigate to see if the content appearing on the widgets matched the content on the TV. The users had choices between channels. The setting of the room was like a living room so that the user felt like he was using the television at his home. The users were asked to fill a questionnaire after the TV watching session and had to undergo a 5 minute interview to get precise feedback and to get information on the scope for improvements.

6.1 Steps involved in the Experiment

6.1.1 Selection of Context Widgets

We had to make a choice of the widgets to be shown on the television. The widgets shown were displayed for about ten seconds and also appeared only if the user was interested in the channel. The widgets were created based on popularity on the internet and only if it could be made context aware. So, mostly the users were being shown widgets like Twitter Search, YouTube Search because we were concentrating on getting content matching to what the user was watching on television. In the end of the experiment, all the developed widgets were shown, so that the importance of each application could be documented.

6.1.2 Participants and Test room setup

The number of participants in total was 15 with the first 2 considered as pilot tests. Most of the participants were aged between 20 – 26 years. All the participants were from Philips but none of them from the TV domain. We wanted to invite participants who were not aware of this technology on television. The tests were held in the AmbiLab which is an integral part of Philips Research where some of the experiments take place. The AmbiLab has a home scene setting meant to feel like a living room.

6.1.3 Wizard of OZ

As this prototype was required to give an interactive feedback during the TV session, the chosen experiment technique was Wizard of OZ (Wiki WOZ, 2009). Wizard of Oz experiment is a research experiment in which users interact with system that they believe to be autonomous, but which is actually being operated or partially operated by an unseen human being. There were some parts of the experiment were widgets had to be loaded according to the current content on television. For example, on appearance of an advertisement, a personal widget was to be loaded, this could only be done by human intervention without the test user being aware of it.

Reasons of usage of Wizard of OZ, was that there was no technology to determine appearance of advertisements during breaks.

6.1.4 Procedure

Before the test started, we explained the procedure of this test to the users. The experiment was subdivided into phases.

- 5 minutes of Introduction
- 10 minutes of TV session
- 5 minutes of questionnaire
- 5 minutes of interview

Phase 1 was informing the user regarding the purpose of the experiment and the user's role in the experiment. Phase 2 was where the user plays his role by watching TV and analyzing the widgets appearing on the TV. The user was asked to keep an eye on the time of the widgets and the content mostly. Phase 3 lasted for five minutes to check the user's feedback on this system and the phase 4 involved 5 minutes of interview with the instructor.

6.2 Methods of Analysis

6.2.1 Pearson's Product Moment Correlation Coefficient

Correlation indicates the strength and direction of a linear relationship between two random variables. The correlation is defined only if both of the standard deviations are finite and both of them are nonzero.

Pearson's Product Moment Correlation Coefficient (Wiki Correlation, 2009) is the most widely used measure of correlation. It is also referred to as Pearson's correlation or as the correlation coefficient. The values can range between 1 and -1. A value of -1 indicates perfect negative linear relationship whereas a value of 1 indicates positive linear relationship. For a value closer to 0, there is no linear relationship between variables. If the value is between 0.5 to 1 it is said to have strong positive correlation. It is said to have medium positive correlation if it lies between 0.3 and 0.5. It is done by summing up the products of the deviations from the mean. We calculate

the numerator by summing up the products of the deviations of a subject's X score from the mean of the Xs and the deviation of the Y's score from the mean of Y's. This is to be divided by the denominator which is the number of entries multiplied by the number of the standard deviation of the X variable multiplied by the standard deviation of the Y variable.

$$r = \frac{N \sum xy - \sum(x)(y)}{\sqrt{[N \sum x^2 - \sum(x^2)][N \sum y^2 - \sum(y^2)]}}$$

Where,

r = Pearson R correlation coefficient

N= Number of value in each data set

$\sum xy$ = sum of the product of the paired scores.

$\sum x$ = sum of the x scores

$\sum y$ = sum of the y scores

$\sum x^2$ = sum of the square x scores

$\sum y^2$ = sum of the square y scores

6.2.2 Wilcoxon Signed-Rank Test

The Wilcoxon signed-rank (Wiki Wilcoxon, 2009) test is a non-parametric statistical hypothesis test for the case of two related samples or repeated measurements on a single sample. The calculation is towards how far is each value from the hypothetical median which is the value 0. The Wilcoxon signed rank test works as follows.

- For a given sample, the paired values X_a and X_b is calculated.
- It takes the absolute difference of |Xa-Xb| for each pair
- It omits the cases where the absolute difference is equal to 0
- It ranks the other absolute differences in ascending order and gives shared ranks if the rank is tied
- If |Xa-Xb|>0, then a positive sign is given and if |Xa-Xb|<0, a negative sign is given to the rank.
- Calculate the value of W which is equal to the sum of all the signed ranks. $n_{s/r}$ is the number of signed ranks which is equal to the number of X_aX_b pairs we have in the beginning of the test minus the number of pairs for which $|X_a-X_b|=0$. If there are no tied values, then $n_{s/r}$ remains the same as the number of pairs in the initial given sample.

In our user test analysis we will compare the sample to be the widget application and compare it to how much the user liked it in television to that of the user's usage on the internet.

6.3 Results of User Test

6.3.1 Questionnaire Analysis

The questionnaire was designed to assess the user's opinion on different issues like the overall television watching experience, the use of widgets on television, and the perceived match between the TV content and the content that appeared in the context aware application. The questionnaire had a pair consisting of a positive and a negative question for each of the above criteria. Pearson's Product Moment Correlation Coefficient was used to calculate the correlation between the each of the pairs of user's responses. A correlation coefficient between 0.5 and 1 indicates there is a strong positive correlation. The correlation between the 1st and the 2nd question, which both intended to measure if the overall experience was interesting or not, was calculated. The value of the correlation was 0.56 implying a strong correlation (probability is 0.0296). A probability of less than 0.05 is considered as ideal or significant.

For the analysis of the matching content we used Pearson's technique again and the correlation coefficient was 0.51, which again indicates a strong positive correlation. The probability corresponding to this correlation coefficient is 0. 048, thus indicating a significant correlation.

Analysis of the use of widgets on TV gave a correlation coefficient of 0.56 which is a strong positive correlation and with the probability of 0.029 which is less than 0.05, thus making the result significant.

These user responses were therefore averaged per pair and subsequently considered as a single response. These values were aggregated to find the mean value of acceptance for the maximum value of 7. The mean value for watching experience was 5.63 and with 100% of the users rating it above 4. The mean value for use of widget was 5.56 with 100% of the users rating it above the mean value. The mean value for the matching content between TV and context aware widget was 5.6 with 100% of the users rating it above 4.

To analyze the difference in user responses regarding the perceived usefulness/value of the context aware twitter search application on the TV respectively PC we used the Wilcoxon Signed Ranks method. The value of the signed rank W was 87, which corresponds to a probability of 0.0025. If the value of probability p is less than 0.05 it is considered as significant. Therefore the users really preferred to use the application on the TV rather than on the PC.

Most of the users stated they were not even aware of the real time search application and were amazed by the matching results it generated. 93.3 % of the users had values above the neutral value of 4 towards the need of context aware YouTube search. These context aware applications however gave the best results when the searched query matched the title of the show but gave average results when it was searched based on actor or genre query. As for the distraction level,

the users rated it to about 4.1 mean value out of 7 which means it was close to neutral. This can be improved by reducing the time span of the widgets appearing. These widgets were displayed on the screen only if the users watched the same TV content for 3 minutes. This level of distraction can be decreased to a much more comfortable level if it is displayed once in half an hour or more. As for the other personal widgets that came during the advertisements, the participants clearly did not like the email widget as they did not want their private information to be displayed on the television. This fact is also proved with the interview results conducted earlier. This can be done using a reasoning engine which displays widgets only if certain conditions are met over a longer period of time. Clearly the other personal applications signed ranks are well about the hypothetical value of 0 which means most of the users liked it. The ratings of these applications differed as the applications used by the participants were related to the user's internet usage.

Please refer to appendix A.1 for the used questionnaire.

6.3.2 Interview Analysis

The 1st question was regarding the widgets content if it was too much or if it was just right. 53.4% of the participants found the amount of information in the widget appropriate. The other remaining participants said it was too much. Two of the participants found that the information was a little small and the only way to make it fine would be to decrease the content and increase the size of the text information.

The 2nd question was regarding the amount of time it was shown on TV. 74% of the participants found the time displayed on the TV to be just right which was kept for ten seconds. Two of the participants preferred it to appear only on a click of a button and the other participants preferred it to come for at least 20 seconds as the content was too much to read.

The 3rd question was to check if this demo showed the user the future of TV. 100% of the participants said they think this would definitely appear in the future and also wished more interaction was possible. As our aim was to make a context aware application but not concentrate much on interaction, it could be the future of this work.

The 4th question was towards the experience of this demo to check if it was a interesting experience or not and also to state the reason for the experience. 93.3% of the participants liked the idea of widgets showing matching content. Some of the participants also said it would bring youngsters back to television. One participant said that it was really nice that the voice of the social media could be got on to the television, let alone the internet. One of the users was not happy with the widgets appearing without being called as it would really distract him if he was watching something interesting. The user mentioned that it would be okay if there would be settings to control the appearance of widgets and not displaying without any human intervention. One of the user also mentioned of the "on click" appearance of these real time widgets which

would be really good if the user wants to know some information on the program he has never watched before.

The 5th question was to name any application they would want to see on TV and to also name any application that could generate real time results like the ones displayed in the demo. 66% of the participants wanted to see widgets about social sites like Facebook and chat options like use of Messenger or Skype. One of the other commonly mentioned application by the users was the IMDB widget to get relevant information. The only application the participants named for real time application is Google. Google is still in the process of developing a fully functional real time search as it is still based on page rank.

Question 6 was to find if the user would like to see widgets in the TV in the coming future. 100 % of the participants said that they really like the widget concept on TV. However some of them still insisted on it coming on a click and also in a much controlled fashion. This can also be the future work of the project for which a reasoning engine can be built depending on the sensor information to display widgets in a timely manner such that the user does not get annoyed.

The 7th question was to check if the user gets annoyed or if the user enjoyed this experience as a whole. 74% of the users found it enjoyable, however some users found it annoying because it was not interactive. Some users also said that it can be improved to be less annoying if it takes lesser display space, for example, show it as a roller in the bottom of the screen. Most of the users however agreed to the fact that since they are used to a television never reacting, it might have seemed like that, but once the user is used to it, it really would not be much of a concern.

The 8th question was framed to identify if the user thinks that widgets like these would make a television session different each time and make it more interesting. 93.3% said it would definitely make television watching more interesting and would bring back the younger crowd into television. Some of the participants said interaction would make it great.

The 9th question was regarding privacy issues. 100 % of the users said mails and YouTube subscriptions should never be shown when there is someone present in the room. Most of the users spoke about the need for a preference setting for the widgets shown.

The 10th question was for the users to give feedback on improvements in the user test. Most of the users said that the widget showed too much information and it would have been better with a clearer font size. Some of the users also pointed out the need for “on click” context aware widgets so that the user does not get annoyed. Some users also pointed out an option to close the widget “on click” if the user is concentrating on something important on TV. Some of the users said about 4-5 items of content was enough but not more than that. We displayed about 8 – 10 items in most of the widgets. Some of the participants said that it would be really nice to make it more interactive. 14 of the participants were delighted with the position of widget. One user said that it is better to make it like a roller or a ticker to make it less annoying.

6.4 Summary

In this chapter, we spoke about the relevance of context in IPTV. We used statistical techniques to prove it. We showed the importance of context aware applications and the user acceptance towards using context information in TV. We acquired user feedback for improvements by conducting an interview. We were provided with good insights on the necessary applications and its usefulness on TV.

Chapter 7: Conclusion and Future Work

7.1 Conclusion

In this thesis we did literature study in the area of context information and researched on valid context in television domain. The initial list of context parameters came to about 100 and we conducted brain storming sessions with experts to identify high level context information. We identified tiny applications which could be used in television and developed widgets on new web technologies. We learnt how context information could be managed within a context management framework and the services it could provide to make this system more aware.

Context is needed and we understood it by looking at the analysis of the user tests. It helps in providing a user a better experience. Certain context aware applications help in enhancing experiences and portraying these experiences into the social web. It helps in automating TV sessions and thus ensures that no session is alike. It brings the element of uncertainty into each session. It brings real time content into television thus making the user aware of real time events. It can help in linking advertisements in the television to the web thus bringing in more relevant content. It helps in changing the way people look into media. It makes the television system more intelligent and more responsive.

7.2 Future Work

As the domain of context is new to television systems more work can be done in the following directions,

- Reasoning Engine
  Use a reasoning engine which uses low level context information to develop higher level context information. It combines the facts and rules to generate higher level information.
  There is a need to decrease the level of distraction and can be done using a reasoning engine.
- Device interoperability to gather additional context
  Indentify how context information can be acquired from other devices and be used to improve services on television.
- Database Profiler
  Develop an individual profile for each user depending on the television sessions and come up with a long term profile. Profile can be made for likes in terms of genre or channels being watched.
- Social Site Widgets
  Develop simple television widgets for social platforms like Facebook and Orkut displaying sufficient information.

REFERENCES

Adams, N., Schilit, B.N., Adams,N. and **Want,R.** (1994) Context-Aware Computing Applications. *International Workshopon Mobile Computing Systems and Applications* , 85-90.

Amigo. (2008) Retrieved from Hi-Tech Projects: http://www.hitech-projects.com/euprojects/amigo/

Biswas, S. Q. and **Quwainder, M.** (2008) Body Posture Identification using Hidden Markov Model with Wearable Sensor Networks. *Third International Conference on Body Area Network.*

Blog. (2008) Retrieved from BlogHerald: http://www.blogherald.com/2008/02/11/how-many-blogs-are-there-is-someone-still-counting/

Broens, T.H.F. (2004) *Context-Aware, Ontology-based, Semantic Service Discovery.* The Netherlands.

Broens, T.H.F., **Quartel, D.** and **Sinderen, V.M.** (2008) *Towards a Context Binding Transparency.*

BSF. (2007) Retrieved from Broadband Services Forum: http://www.broadbandservicesforum.org

Deniz, U. and **Oral,M.** (2007) Centre of mass model - A novel approach to background modelling for segmentation of moving objects. *Image and Vision Computing* , pp. 1365-1376.

Flash Developer Center. (2006) Retrieved from Adobe: http://www.adobe.com/devnet/flash/

Fokus. (2009) Retrieved from Fraunhofer Institute for Open Communication Systems: http://www.fokus.fraunhofer.de/en/fokus/index.html

Freeband. (2006) Retrieved from Freeband Awareness: http://www.freeband.nl/project.cfm?id=494&language=en

Google. (2006) Retrieved from Google Gadgets: http://www.google.com/webmasters/gadgets/

Hesselman, C. E., Eertink, H., Wibbels, M., Sheikh, K. and **Tokmakoff,A.** (2008) Controlled Disclosure of Context Information across Ubiquitous Computing Domains. *IEEE Computer Society*, (pp. 98-105).

Hull, R., **Reid, J.** and **Geelhoed,E.** (1997) Towards Situated Computing. *International Symposiumon Wearable Computers* , 146-153.

iNEM4u. (2008) *iNEM4u.* Retrieved from Interactive Networked Experiences in Multimedia for You: http://old.telin.nl/inem4u/

Kamiya, K. K., Kudo, M., Nonaka, H., and **Toyama,J.** (2008) *Sitting Posture Analysis by Pressure Sensors.*

Keivanloo, L. A. and Abolhassini, H. (2008) An Architecture for Context-Aware Semantic Web Services. *IEEE International Conference on Web Services*, (pp. 779-780).

Kranenburg, V. , Bargh, M.S, Lacob, S. and **Peddemors, A.** (2006) A context management framework for supporting context-aware distributed applications. *Communications Magazine, IEEE* , 67-74.

Lowet, D. and **Shrubsole, P.** (2007) *Content sharing and experience sharing with Web4CE*. TICSP Adjunct Proceedings of EuroITV 2007.

Lowet, D. and **Khmelinskaya, E.** (2008) *Social API for CE-HTML devices.* EuroITV.

Mobigets. (2009) Retrieved from Widget Blog: http://mobigets.blogspot.com/2009/03/mobile-widget-development.html

Mobihealth. (2007) Retrieved from Putting care in motion : http://www.mobihealth.com/services/en/mh_mobile.php

Mobile Widgets. (2006) Retrieved from Plusmo: http://www.plusmo.com/blog/?paged=2

MulticastStreaming. (2003) Retrieved from Microsoft TechNet: http://technet.microsoft.com/en-us/library/cc753242(WS.10).aspx

Park, D. H., Hwang, S., Kim, A. and **Chang,B.** (2007) A Context-Aware Smart Tourist Guide Application for an Old Palace. *Convergence Information Technology*, (pp. 89 - 94).

Philips Net TV. (2009) Retrieved from Philips: http://www.consumer.philips.com/consumer/en/gb/consumer/cc/_categoryid_TELEVISIONS_CA_GB_CONSUMER/?tab=cinema%2021:9

Physical Context Detection . (2008) Retrieved from EGR: http://www.egr.msu.edu/~sbiswas/Research/WearNet.htm

Reynolds, C. and **Picard, R.** (2009) *Affective Sensors, Privacy, and Ethical Contracts.*

Ryan, N., Pascal, J. and **Morse, D.** (1997) Enhanced Reality Fieldwork: the Context-Aware Archaeological Assistant. *Computer Applications in Archaeology* .

Schilit, M. and **Theimer, M.** (1994) Disseminating Active Map Information to Mobile Hosts. *IEEE Network* , pp. 22-32.

Schmidt, A., Beigl, M. and **Gellersen, H.** (1998) *There is more to Context than Location.*

Thawani, A., Gopalan, S. and **V, Sridhar.** (2004) Context Aware Personalized Ad Insertion in an Interactive TV Environment. *the 4th Workshop on Personalization.*

W3C. (2009). Retrieved from World Wide WEb Consortium: http://www.w3.org/TR/widgets/

Web2.0. (2004) Retrieved from Wikipedia - The free Encylopedia: http://en.wikipedia.org/wiki/Web_2.0

Webware. (2008) Retrieved from Cool WEB 2.0 Apps For Everyone: http://www.webware.com/html/ww/100/2008/winners.html

Wiki Atom. (1999) Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Atom_(standard)

Wiki Correlation. (2009) Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Correlation

Wiki PIR. (2009) Retrieved from Wikepedia: http://en.wikipedia.org/wiki/Passive_infrared_sensor

Wiki proximity. (2009) Retrieved 2009, from Wikipedia: http://en.wikipedia.org/wiki/Proximity_sensor

Wiki RSS. (2009) Retrieved from Wikepedia: http://en.wikipedia.org/wiki/RSS

Wiki Widgets. (2009) Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Widget_engine

Wiki Wilcoxon. (2009) Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Wilcoxon_signed-rank_test

Wiki WOZ. (2009) Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Wizard_of_Oz_experiment

WikiYahoo! (2007) Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Yahoo!_Widgets

Wisegeek. (2003) Retrieved from Wisegeek - Research Team: http://www.wisegeek.com/what-is-iptv.htm

XML RSS 2.0. (1999) Retrieved from XML from the inside out: http://www.xml.com/lpt/a/1080

XMLTV. (2009) Retrieved from WIki XMLTV: http://wiki.xmltv.org/index.php/Main_Page

Yaiz, R., Selgert, R., Hartof,F. and **TNO,F.** (2006) On the definition and relevance of context-awareness in Personal Networks. *Mobile and Ubiquitous Systems*, (pp. 1-6).

Appendix

A-1 Questionnaire

Please tick the box you most agree with,

1. My TV watching experience was pleasant.

2. My TV watching experience was not interesting.

3. The Widgets appearing distracted me from the TV watching experience.

4 I find Widgets on TV a nice concept.

5 The Widget's information matched the content currently being watched.

6 The use of widgets on TV is not interesting.

7 Rate the Widgets according to the NECESSITY on TV. (Very Important -> 7 , Not Important -> 1)

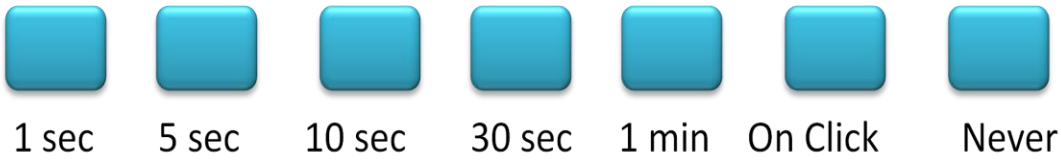
- Email -> / 7
- News -> / 7
- Twitter Search -> / 7
- Blog -> / 7
- YouTube Search -> / 7
- YouTube List of the day – Top Discussed/Commented/Views -> / 7
- Twitter Status Update -> / 7

8 Rate the application according to your usage on the Internet. (Very Important -> 7 , Not Important -> 1)

- Email -> / 7
- News -> / 7
- Twitter Search -> / 7
- Blog -> / 7
- YouTube Search -> / 7
- YouTube List of the day – Top Discussed/Commented/Views -> / 7
- Twitter Status Update -> / 7
-

9 These widgets appeared at appropriate times.

10 What is the appropriate duration, the widget should be shown.



11 The contents on the widget were inappropriate.

A-2 Interview Questions

1. Was the Widget content shown enough or was it too much?
2. How long would you want these widgets to be shown on the TV?
3. Does this demo show you a nice glimpse into the future of TV?
4. What did you like in this viewing experience?
5. If there is some other widget application that you want to see on the TV, which would it be? Is there any other application that could generate real time results according to the channel you are watching?
6. Is this something you would like to see in your TV in the near future?
7. Was this experience more enjoyable or more annoying than a "normal" TV viewing experience?
8. Do you think widgets like this on TV, would make TV watching more interesting?
9. Would you mind that your mail or your YouTube subscriptions show up in front of other people?
10. Would you like to see some changes in how the widget information was shown? (I.e. the widget's position on the screen or the information that was shown?

A-3 Wizard of OZ lab setup