André Filipe dos Santos Pinto Fidalgo

Licenciado em Engenharia Informática

IPTV Data Reduction Strategy to Measure Real Users' Behaviours

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática

Orientador : João Moura Pires, Professor Auxiliar,
Universidade Nova de Lisboa

Júri:

Presidente: Prof. Doutor Pedro Abílio Duarte de Medeiros

Arguente: Prof. Doutor Salvador Luís Bettencout Pinto de Abreu

Vogal: Prof. Doutor João Carlos Gomes Moura Pires

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Novembro, 2012

IPTV Data Reduction Strategy to Measure Real Users' Behaviours

Copyright © André Filipe dos Santos Pinto Fidalgo, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

To my parents

Acknowledgements

First, I would like to thank my dissertation supervisor, Prof. Dr. João Moura Pires, by the opportunity to work with him over the last year. It was intellectually rewarding and fulfilling. Every meetings and feedbacks, helped me to improve and enhance all the work done so far. I also thank to Novabase, in particular to Rui Pedro Alves and João Barateiro for all the support they gave me, mainly in earlier stages of my dissertation work. They provide me valuable contributions, regarding the provision of data and also, their insightful suggestions and expertise.

A special thanks go to Bruno Filipe Faustino whose friendship I deeply appreciate. To him, and all my colleagues, thank you not only for the support, but for all the great moments passed along this academic journey.

The last words of thanks go to my family. First to my parents Grácia e Fernando, for the commitment, effort and trust they placed in my graduate course. Last but not least, a very special thank to my girlfriend Ana, for all support, motivation and affection.

Abstract

The digital IPTV service has evolved in terms of features, technology and accessibility of their contents. However, the rapid evolution of features and services has brought a more complex offering to customers, which often are not enjoyed or even perceived.

Therefore, it is important to measure the real advantage of those features and understand how they are used by customers. In this work, we present a strategy that deals directly with the real IPTV data, which result from the interaction actions with the set-top boxes by customers. But this data has a very low granularity level, which is complex and difficult to interpret. The approach is to transform the clicking actions to a more conceptual and representative level of the running activities. Furthermore, there is a significant reduction in the data cardinality, enhanced in terms of information quality. More than a transformation, this approach aims to be iterative, where at each level, we achieve a more accurate information, in order to characterize a particular behaviour.

As experimental results, we present some application areas regarding the main offered features in this digital service. In particular, is made a study about zapping behaviour, and also an evaluation about DVR service usage. It is also discussed the possibility to integrate the strategy devised in a particular carrier, aiming to analyse the consumption rate of their services, in order to adjust them to customer real usage profile, and also to study the feasibility of new services introduction.

Keywords: IPTV, Data Stream, Complex Event Processing, Data Analysis

Resumo

O serviço digital televisivo IPTV tem evoluído em termos de funcionalidades, tecnologia e acessibilidade dos seus conteúdos. Contudo, a rápida evolução de funcionalidades e serviços tem trazido uma oferta mais complexa aos clientes, as quais muitas vezes, nem são usufruídas ou mesmo percebidas.

Nesse sentido, é importante medir o real proveito das funcionalidades e compreender como essas são utilizadas pelos clientes. Neste trabalho, apresentamos uma estratégia que lida directamente com os dados reais de IPTV, que resultam das acções de interação com as *set-top boxes* por parte dos clientes. Mas, esses dados têm um nível de granularidade muito baixo, complexo e de difícil interpretação. A abordagem passa por transformar os eventos ao nível das acções de *click* para um nível conceptual mais genérico e representativo das actividades a decorrer. Além disso, há uma redução significativa na cardinalidade de dados, acrescida dum ganho na qualidade de informação. Mais que uma transformação, esta abordagem pretende ser iterativa, ao ponto de a cada nível, se obter um maior rigor de informação e adequado à caracterização dum comportamento em particular.

Como resultados experimentais, apresentamos algumas áreas de aplicação referentes às principais funcionalidades digitais oferecidas neste serviço. Em particular, é feito um estudo do comportamento de *zapping*, e também uma avaliação de uso do serviço de DVR. É também discutida, a possibilidade de integração da estratégia idealizada numa operadora em particular, com o intuito de analisar a taxa de consumo dos seus serviços, de forma a ajustá-los ao real perfil de utilização do cliente, e também para estudo da viabilidade da introdução de novos serviços.

Palavras-chave: *IPTV*, *Data Stream*, Processamento de Eventos Complexos, Análise de Dados

Contents

List of Figures

Listings

Glossary

Activity Stream a stream of IPTV data representing the conceptual user activity level. 7, 18, 24, 25, 59

Audiometry is a tool that has a key role in the way of planning means of communication and extent the performance of it among consumers, with particular intent to inform the media the quantification and qualification of its audience. 4, 7, 15, 16, 60

Base Transceiver Station is a piece of equipment that holds an antenna used to enable a wireless connection between a user equipment and a network. A geographical region is served by a set of BTSs, each characterized by the coordinates: latitude and longitude. 10

Billing System is a solution in telecommunication industry that enables common management of all users and all services for operators through a powerful process that collect usage service data to raise invoices for the customers. 3, 4

Business Intelligence is a set of methodologies, processes, architectures, and technologies that transform raw data into meaningful and useful information used to enable more effective strategic, tactical, and operational insights and decision-making. 4

Call Detail Record is the record produced by a telephone connection containing details of calls that used this connection. 10

Click Recording an IPTV event record associated to a particular action performed by the user using a remote control, for instance, the channel tune action. 5, 13

Click Stream a stream of IPTV raw data regarding users click actions, i.e., the finest granularity available. 6, 7, 18, 24, 25, 27, 28, 33, 59

Complex Event Processing is a concept that is characterized by the analysis and processing of large volumes of data streams, on which, by applying filters, correlations, aggregates or patterns it is possible to extract meaningful information. 25, 60

Customer churn is a business term used to describe loss of clients or customers, for instance in switching to another company or service provider. 11

Digital Video Recorder is a major IPTV functionality which allows the users to easily record their favourite show or series and watch them whenever they want as well as pause and rewind live TV. 5, 48

Domain Specific Language is a type of programming language or specification language in software development and domain engineering dedicated to a particular problem domain, a particular problem representation technique, and/or a particular solution technique. 26

Esper is a component for Complex Event Processing (CEP), available for Java as Esper, and for .NET as NEsper.. 26, 30, 60

Event Correlation Engine is an engine able to process large number of events and pinpointing the few events that are really important in that mass of information. 26

Event Driven Architecture is a software architecture pattern promoting the production, detection, consumption of, and reaction to events. 26

Event Stream Processing is a related concept to *Complex Event Processing*, which deals with the task of processing multiple streams of event data with the goal of identifying the meaningful events within those streams, employing techniques such as detection of complex patterns of many events, event correlation and abstraction, event hierarchies, and relationships between events such as causality, membership, and timing, and event-driven processes. 26

Interactive Channel a special channel that offers exclusive interactive contents, for instance, follow news contents organized by areas of interest. 2, 5

Internet Protocol Television is a system through which television services are delivered using the Internet protocol suite over a packet-switched network such as the Internet, instead of being delivered through traditional terrestrial, satellite signal, and cable television formats. These services may include, for example, Live TV, Video On Demand and Interactive TV. 2

JavaBean is a Plain Old Java Object that is serializable, has a no-argument constructor, and allows access to properties using getter and setter methods that follow a simple naming convention. 27

Online Analytical Processing is a multi-dimensional data model where the information is conceptually organized into cubes that store values or quantitative measures. 16

Pay-per-View provides a service by which a television audience can purchase events to view via private telecast. The broadcaster shows the event at the same time to everyone ordering it (as opposed to video-on-demand systems, which allow viewers to see recorded broadcasts at any time). Events can be purchased using an on-screen guide, an automated telephone system, or through a live customer service representative. Events often include feature films, sporting events and entertainment. 20

Plain Old Java Object is a Java object not bound by any restriction other than those forced by the Java Language Specification. 27

Remote Access an emerging feature which allows user to watch TV contents or schedule recording of contents anywhere, just using a mobile device such a tablet or smart phone. 2

Remote Desktop Protocol enables subscribers to interact with applications with their remote control. Such applications can be in form of Web Applications or standalone Windows applications, that ultimately can interact with remote resources: web servers and databases. 20

Services Management the ability to personalize subscription services such as paid channels through a remote control. 2, 23, 60

Set-top Box is an information appliance device that generally contains a tuner and connects to a television set and an external source of signal, turning the source signal into content in a form that can then be displayed on the television screen or other display device. It's used in currently IPTV systems and users can interact with it through a specific remote control. 4, 12, 59

Television Widgets an interactive application regarding a particular subject, such news, traffic, weather forecast, utilities and others. 2, 23

Triple Play is a service that combines voice, data and multimedia services under a single communication channel bandwidth. 1, 15

Video on Demand is a service which allows users to select watch/listen to video or audio content on demand. Those contents can be viewed in real time or downloaded into a Set-Top Box for viewing any time. The majority of Triple play providers offer the ability to rent contents (films from a digital catalogue), or to watch them freely (programs that have already been broadcasted). 2, 5

Visualization is the act of user view a given content in a certain channel. 5, 17, 23, 28, 33–38, 41, 44–47

Visualization Moment is a particular period of visualization time. 30, 34–36, 38, 39, 41, 44, 48

Visualization Session is a sequence of contiguous Visualization Moments. 34, 35, 41, 43, 44

Zapping is a TV user behaviour which is characterized by the practice of quickly scanning through different television channels within a short time interval, owing to the search of a channel or specific content with interest enough to fix the user to view it. 7, 8, 14, 15, 23, 31, 33–39, 41, 43, 44, 46–48, 60

Zapping Moment is a TV watching period in which the user only watch a particular content for a very short time. 34–40, 44–48

Zapping Session is a sequence of Zapping moments which contain a watching duration less than a predefined threshold. 34–36, 38–41, 46

Acronyms

ADSL Asymmetric Digital Subscriber Line. 20

BTS Base Transceiver Station. 10, 11

CDR Call Detail Record. 10, 11

CEP Complex Event Processing. 7, 25, 26, 29, 60, 61

DBMS Database Management System. 17

DVR Digital Video Recorder. 6–8, 13, 20, 23, 24, 28, 29, 31, 33, 48–52, 54, 56, 57, 60

EPL Event Processing Language. 26, 27

FTTH Fiber to the Home. 20

IPTV Internet Protocol Television. 2, 4–7, 9, 12–21, 23–26, 28, 31, 33, 48, 50, 52, 54, 56, 59–61, 67

KDD Knowledge Discovery in Databases. 17

QoS Quality of Service. 11

STB Set-top box. 5, 6, 12, 13, 18–20, 22–24, 28, 33, 36, 39, 41, 50–52, 56, 59–61

VOD Video on Demand. 20, 21, 23, 24, 28, 29

1

Introduction

The hereto introductory chapter aims at presenting both the context and the motivation that have spurred the present work as well as the problem we will be focusing on. The approach to be followed and the expected outcome will also be briefly stated. Finally, the layout of this document will be presented.

1.1 Context and Motivation

The way Telecommunications market offerings reach customers has deeply changed over the last few years. In the past, each telecommunication company had a specific target and only focused on a single service. However, market stiff competition urged these operators to expand their offerings which led to a cross-selling increase of multiple distinct services and thus turning actual telecommunication companies into N-players operators, simply because many of them render customers a wide set of different services: Phone, TV, Internet, Mobile Voice, Mobile Data and Mobile TV. Take Vodafone (typically associated with mobile telecommunications service), for example, that embarked on a Triple Play service rendering. As for Zon Multimedia (typically associated with television subscription services), it has undertaken a new enterprise in the field of mobile communications (Zon Mobile). Competition in this business world has therefore become tougher and tougher, since each operator combines its services in promotional packages in order to attract customers from other carriers. We are therefore witnessing a phenomenon in which services offerings start to be complex. Instead of the traditional plain service rendering, we have a full package of services, which given their many combining possibilities due to its own peculiarities has led to some misunderstandings among customers. Moreover, these complex offerings are often designed to sell product X on the one hand and get product Y on

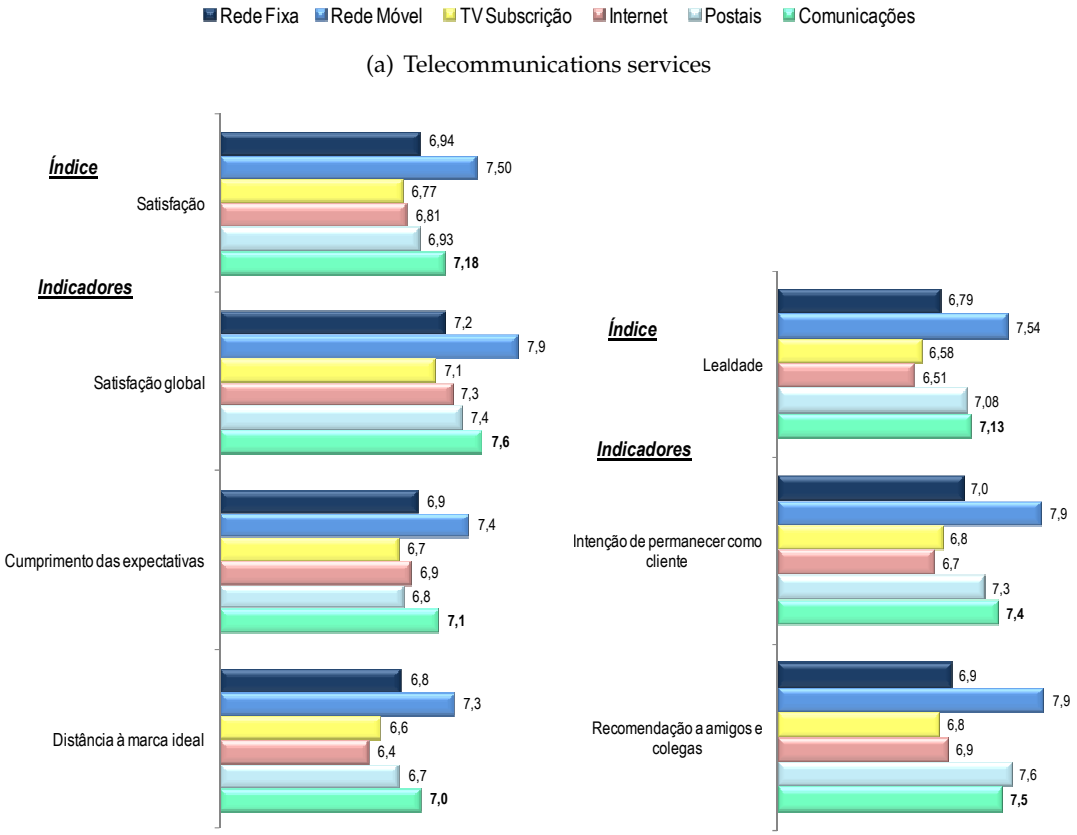
the other.

Along with the market offering developments, telecommunications services have become more functional, more accessible and user friendlier tools. Somehow, we can say that customers are aware of it, except for television service, where the technological factor does not always meet customers' interests, enjoyment needs and understanding. In the past, the analogical signal only allowed national television broadcasting; television signal was transmitted over the air by radio waves and received by a television antenna attached to a TV set. This scenario was rather limited in what concerned to channels offering and features: unidirectional and without users' interaction. Then, cable television allowed television programs to be broadcasted via radio frequency through coaxial cables signals. These cables not only carried bi-directional signals but allowed the transmission of large amounts of data as well, which paved the path to the appearance of other digital services such as cable internet and cable telephony. Furthermore, customers were able to receive multiple television channels, most of them international, which was a commercial turning point. However, users were still unable to interact with television contents. Recently, the digital television service Internet Protocol Television, which is based on the Internet as support, has been brought about and offers several digital services, that have allowed each operator to include in its offerings services like: Video on Demand, Television Widgets, Services Management, Remote Access and ability to control live broadcasts and hence allowing customers to be in full control and personalize the way they watch TV. These services gave rise to a deep change concerning to consumer-content relationship. Previously, customers depended on the content: a given content was broadcasted in a scheduled day, at a scheduled time. Nowadays, clients can watch the contents whenever they want. This means the Internet Protocol Television (IPTV) service offers the customer the chance to pause live broadcasts, replay the current TV content from its beginning forward, rewind a program one has been watching, schedule recordings of programs, rent films, use Interactive Channel, among others. As a result, we can classify actual IPTV service as a particular case of complex service rendering, where the technological factor hasn't caught up with the proper introduction of certain services in the market. Take for example the Mobile TV (Remote Access) service, which is still at a very disturbed phase because neither customers nor the operators themselves are adjusted to the turmoil within this service.

In a survey published by ANACOM[1][ANA09] in 2009 about customer satisfaction in the ECSI-Portugal model (European Customer Satisfaction Index - Portugal) in the telecommunications market (regarding several telecommunication services 1.1(a)), seven customer satisfaction variables were identified: Image, Expectations, Perceived Quality, Perceived Value, Satisfaction, Complaints and Loyalty). The results showed that the television and Internet services are those with a general low index of satisfaction (in a 1 to 10 scale) compared to other services (see Figure 1.1(b)). This satisfaction index is based

[1] *Autoridade Nacional de Comunicações* (ANACOM) regulates and supervises the electronic and postal communications sector in Portugal, providing national representation at various international forums.

on three indicators: overall satisfaction with the service, performance of expectations for the service provider, and the brand distance from the ideal service provider. In the same study, it was also pointed out that concerning the index of customer loyalty, subscription television and internet services are also the worse assessed ones in comparison with the other services (see Figure 1.1(c)).



(a) Telecommunications services

(b) Satisfaction index

(c) Loyalty index

Figure 1.1: Satisfaction Indexes about Communication Services

Due to every business operator's desire to be unique, the customer is placed in a position where one purchases a set of services from which some might be barely used or even known. According to a study published in 2011 by ANACOM[ANA11], 16% of Portuguese households reported that having a package service subscription translates itself into the provision of services that are not necessarily required. In the same study, and referring to the global satisfaction of TV subscription service comparing to the previous year (2010), the proportion of clients who gave a high positive grade decreased about 12 pp.

Despite the developments in this market, there is still the need to understand if the offer is suitable for customers' proper consumption. In mobile services there is a powerful Billing System (see Figure 1.2) which is a process of collecting usage data to raise invoices for the customers. The DWH module, present in this architecture, is a downstream system for the Billing System and usually keeps tons of historical data related to

the customers. This Billing System dumps several customer's information into the DWH system. This information includes service usage, invoices, payments, discounts, adjustments, etc. All this information is used to create different types of management reports, for Business Intelligence and forecast. Thus, mobile billing systems enable a detailed record of the consumption/usage of services either by billing needs or the need to know how the offerings reach customers. On this scope, it is an utterly discussed and explored domain.

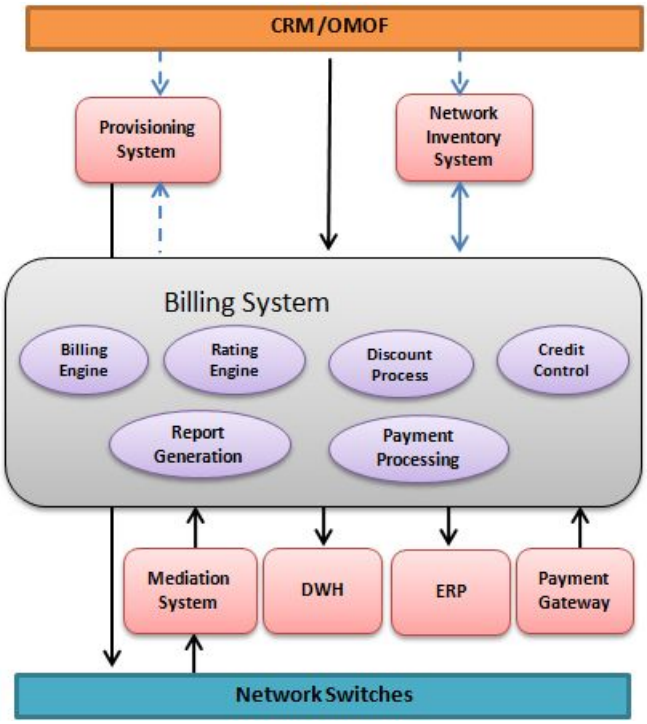


Figure 1.2: Mobile Billing System Architecture

Yet, similar analyses are not currently possible in television service. In spite of the key role that Audiometry analyses plays in means of communication planning and the range of its performance among consumers, it is very limited regarding to real usage. On the one hand, the analyses focus on a representative sample of population - called an audience panel, but the sample application to the universe does not depict neither the reality of a particular carrier, nor the reality of a particular customer, since it describes common general features. Furthermore, this kind of studies is strictly target to market share with an underlying intent to inform the media about the amount and quality of its audience. In this sense, we suggest a different way to deal with the existing operators need to measure the real usage of their television services. Current IPTV systems can generate amounts of information that are associated with user's click events through remote control, such as on/off Set-top boxes, tune channels, watch live TV, schedule programs or menu navigation to just mention some examples. This clearly gives us a most exquisite level of detail, since we can trace all actions performed by user. Moreover, we are not constrained to live broadcast content analyses, due to the ability to recognize

several actions regarding digital services such as Digital Video Recorder and Video on Demand actions, which meets our motivations to analyse the real usage of such features offering.

In sight of the recent technological advances in the telecommunications market, the focus of this thesis is to work with real IPTV data. This choice is due to several factors:

- The IPTV service is the one that has recently suffered major changes in the way it is introduced to the customer. It has even radically changed the way customers watch TV and how they interact with the service;
- The exclusive offerings of contents have fallen especially in this service, where for instance, there has been an increase of a wide range of services at the customer's disposal: Video on Demand, Interactive Channels , stop live broadcasts, schedule recordings of contents, unique widgets, Mobile TV. However, there is currently no monitoring/detection of activities or behaviours on these features;
- Telecommunications companies providing this service have the ability to store information that is associated to user events, particularly, all operations the customer performs using remote control, trigger event logs in Set-top box (STB). Actually, according to main Portuguese IPTV service providers, the way they measure the use or entry of a particular service is through questionnaires, focus group or percentage of users adherence to campaigns, which results in its actual non usage for service monitoring;
- In addition, this work relies on the involvement of a major Portuguese market consultant (Novabase[2]), who have provided us not only the necessary support in terms of some feedback prior to the devising scope of this thesis but the provision of data we will work with as well.

1.2 Problem

Besides the basic functionality of watching contents, the IPTV service enables the introduction of other services, as a means of accessing to those contents, as previously described. The integration speed and volume of these services do not always have to do with the customer perception and need, leading to mismatches between what is needed and what is used. Following our motivations, the problem to face is then to work out IPTV data from Click Recording, in order to find Visualization and usage profiles to measure the real advantage of those features and understand how they are used by customers.

Current IPTV systems can produce events associated to each click action by customers, so it is easy to imagine that thousands of scattered events are generated at the

[2]Novabase is a major ICT consultant in Portugal, and have specialized products and services for the Telecoms & Media.

same time, resulting in millions of events in a very short time. The problem we want to address is how to extract information from these records and thereby characterize activities related to services usage.

Since the raw data has a very fine granularity level it does not allow an immediate identification of a particular activity as for instance: watching live TV, accessing to video club programming, scheduling a program, etc. In other words, so that we can say that a given user could have taken advantage of a particular service, we may have to look for data as a sequence of clicks which has denoted a recent particular activity. Therefore, the core challenge is then how to turn the information straightforward in order to make it more noticeable both in conceptual and representative way about activities taking place in a STB. However, the activity level did not always allows an immediate behaviour analysis. For instance, although we can detect Digital Video Recorder (DVR) activities that occur in a given STB (start record, delete record), they are not directly linked. So, if we want to recognize a particular users' behaviour, such as whether users who scheduled a particular content really watched and deleted it later, we need another data transformation that simplify the information aiming at depicting such behaviour. In other words, we are talking about a iterative reduction process, where we could have the accurate information detail at each level.

Besides its very low granularity, this data is created at a high rate per second, which gives rise to millions of daily records. Thus, one problem is also to devise how to lower this amount of information. So, alongside the need to generalize the information in order to make it more perceptible, it will contribute to a substantial reduction in the volume of data and hence enriching the quality of information.

Of course, we have to deal with some problems relating the data. Although each activity can be represented in several distinct manners, there is no guarantee that the arrival order of those events it's always the same. Last but not least, the errors in data may occur, such events that come loose without realizing their meaning since do not fit in the context of the current activity.

1.3 Approach

The main objective is to characterize IPTV services customers' usage profiles, taking into account the customers' usage behaviours. To reach this goal, we must examine each STB available data as well as consider a framework that can transform the raw data into meaningful information related to current IPTV services, which could allow us to analyse them later.

The nature of this data enables us to consider it as a stream of complex events we called Click Stream. A means to lower the amount of information and improve the ability to understand the activities that occur in a given STB is to work with the notion of aggregates, take into account not the click activity but rather the user activity, which is

characterized at a higher conceptual level, reducing the cardinality and with a more representative meaning in terms of analysis. So, we suggest a transformation process that will be in charge of compressing information, turning the original **Click Stream** to **Activity Stream**. Moreover, this transformation process intends to be as much iterative as possible; each transformation level could by further applied, reducing data cardinality and enhancing information quality more and more. Each level will be the most suitable for the desired scope of analysis.

First, our approach focus on launching a comprehensive survey on the existing IPTV services, targeting them into activities likely to be further analysed and detailed as much as possible. The second phase involves the integration of the *Complex Event Processing (CEP)* concept in IPTV context, which allows the activities extraction through a CEP tool able to map the previous activities. This step could be repeated depending on the required level of detail for analytical purposes. In particular, this iterative process resulted in some application areas, which focus on users' behaviours analyses: Zapping and DVR usage.

1.4 Contributions

The main expected contribution is to make a general overview about current issues related to telecommunications market offerings, in particular of IPTV services. However, we do not intent to address matters relating to prices or specific offers, but instead create the basis for it. That overview is achieved based on the proposed framework which deals and transforms real IPTV data into meaningful information regarding users' behaviours. As we are talking about a framework, it allows its extensibility to other IPTV features, being only necessary to make an accurate survey of sequences of events that characterize a given feature, and map it in the CEP tool used for that.

Driving our motivations ahead, we suggest models that define some IPTV users' key behaviours, particularly Zapping and DVR behaviours. Based on the resulting data from our generalization strategy, those behaviours were instantiated and analysed.

The outcome of the hereto work intends to demonstrate the feasibility of using this data which enables a particular company to know how its customers are taking advantage from the services rendered.

1.5 Document Layout

The document will be organized as follows: Chapter 2 will be dealing with issues related to this thesis theme, starting with an overview about the Telecom data, especially IPTV. Some works done on this type of data will also be presented. Then, we will present Audiometry which is the most used tool to measure customers' preferences and trends as well as these tools current limitations. Finally, we will refer to some aspects related to

Data Stream Mining, which is a concept that characterizes the nature of the data we have to deal with.

Chapter 3 presents the followed approach in as much detail as possible. We will start by mentioning some important aspects of the data set we will focus on, and then we drill down along the approach designed to achieve our objectives.

In chapter 4, we will describe some application areas where we have explored the information retrieved from the designed approach. We will start with Zapping, which is an inherent behaviour to live viewing activity. Thereafter, we will present a detailed study about DVR feature in terms of its usage.

To conclude, in chapter 5 we will summarize of the work done so far and hold a critical discussion of both the approach followed and the results obtained.

2

Related Work

In the first part we discuss a little about the Telecom data, explaining how they are generated, with particular emphasis in the IPTV subsection, where it is made a more detailed presentation of the IPTV system architecture, in order to better understand the information that we'll work.

In the audiometry section is given an overview of what it is, its purpose and existing audience measurement studies about television service. Consequently, we'll discuss its limitations and justify why it is inappropriate to deal with IPTV data, namely to study the customers usage behaviours.

Finally we present a Data Stream Mining section which covers some issues related to data we have to deal, starting with its occurrence method - as a stream, and also analytical problems with its monitoring: data reduction, outliers identification and anomalies.

2.1 Telecom Data

Today, telecommunications companies offer to their customers a wide range of services, being known as *N-players* providers. In addition to services offering, telecommunications companies have the ability to record the activity that is associated with each of the services they provide. The most common is the telephone bill, where it is discriminated the records of calls and voice services used by the customer.

However, there has been research work that attempts to analyse the information from the activity records and from there determine customers' usage patterns, in order to understand their behaviours and trends. Some of these works will be referenced in this chapter.

We can then characterize the activity records in three different data types:

- **Voice**: records from the calls or services like SMS and MMS;
- **Data**: records from data traffic, associated to web access;
- **Television**: records from TV activity, as reported in previous chapter.

The following subsections are related to each of one of these types, with some important characteristics and references to works/approaches produced in those areas.

2.1.1 Call Detail Records

A Call Detail Record contains some basic attributes that characterize a call or service (SMS, MMS):

- The number making the call;
- The number receiving the call;
- Identification of Base Transceiver Station both origin and destination;
- Date and time;
- Duration;
- Type of service.

Besides these, the Call Detail Records (CDRs) may contain more attributes, which each telecommunication carrier shall adopt by whether or not according to their needs and characteristics.

There are several areas in witch CDRs could be used:

- **Classification of urban areas**

  In [SF11] the authors proposed a technique to identify and classify city urban areas based on the call recordings. The records were grouped around the Base Transceiver Stations (BTSs) and studied at several aggregation levels: total, weekday-weekend and daily. Applying the k-means clustering algorithm, it was possible to classify the characteristics of each cluster and areas of its occurrence, giving rise to the following urban areas: industrial parks & office; commercial; night-life; leisure and residential. This data mining approach overcomes the typical problems with questionnaires and costs, leading to new capabilities like tracing the evolution land usage and focus in a particular social background.

- **Segmentation of customers**

  The growing interest in analysing the social context in which the users belong, allows to understand the structure and dynamics of social networks in order to recognize patterns or trends in a given group. In this sense, in [DSV+08] is proposed

a model that is based on call records analysis, with intend to understand the role of social relations in the formation and growth of groups or communities in mobile networks. In particular, create graphs of connections through the call records, and forecast the probability of a customer churn, when other clients to whom he relates, already done it before. In the same line, in [PXQ+11] the authors propose a system that focuses on the mobility patterns analysis. In addition to detecting patterns within a cell, they intend to realize how people move between different cells. This type of analysis is beneficial not only in urban planning and traffic forecasting, but also in social behaviour.

- **Fraud Detection**

  The analysis of usage patterns may also be useful for detecting subscription fraud[1], or superimposed fraud[2]. Both are important for telecommunications companies because they are undesirable negative behaviours. The low occurrence rate of these behaviours, turns existing data mining algorithms to not perform well in its detection, as reported in [Wei]. So, new techniques and methodologies are needed to make their detection efficiently.

- **Qualification of services**

  In addition to analysing users behaviours, the CDRs also have applications in areas of monitoring the Quality of Service (QoS). In this area, one example is the monitoring of network configuration, as exemplified in [CHD00]. The proposed framework takes advantage of the volume of calls recorded in the BTSs in order to: *(i)* maximize nodes on the network; *(ii)* avoid traffic jams; *(iii)* find overloading reasons; and *(iv)* improve QoS.

As can be seen, there are several analysis areas in which the call records have direct application. The preceding examples are interesting so we can have an idea of what kind of applications we can obtain, which results can be expected, and not least, the current limitations. However, note that such applications are highly context dependent and designed to address a particular problem.

The CDRs also register the data traffic records, since they are captured in the same way as call services, because the BTSs also give coverage to 3G network. Thus, their content is similar, basically varying only the attribute that indicates the destination, which in this case is no more than the access point used, and the volume of information consumed, expressed in kilobytes (kB). Actually, there is currently no information that could support the analysis of the content that is passed over the network, and this is why we did not find related work regarding this topic. Nonetheless, it is important to point out that these analyses could be an interesting area. For instance, to study patterns related to the kind of applications used via mobile, and where they occur geographically.

[1] Clients who opens an account with intention of never paying it.
[2] Illegitimate activities by the customer.

2.1.2 IPTV

The current television service provides the user with a unique television experience, regarding the wide range of contents and quantity of available features. These features are supported by a digital service called IPTV differing from the traditional television formats because live TV streams are encoded in a series of IP packets and delivered to users through residential broadband access network.

Basically, the IPTV system architecture is composed by:

- **Super Head-end Office**: primary source of television which digitally encodes satellite signal, encrypt it and transmits to several Video Head-end Offices (VHO);
- **Video Head-end Office**: Each VHO is allocated to a metropolitan area and adds some content like advertisement before transmit it to a residential home;
- **Residential Gateway**: Device hosted in client home that connects to a modem and one or more STBs;
- **Set-top Box**: The device that connects directly to a client TV. The STB is controlled by a remote control, with several buttons to interact with IPTV service.

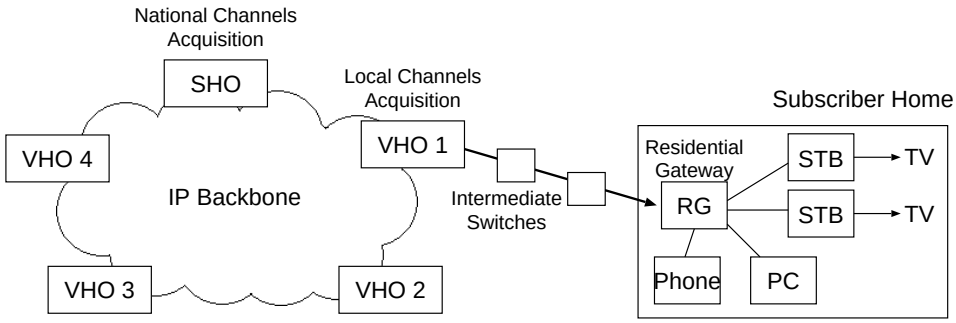


Figure 2.1: IPTV System Architecture

More precisely, one client can be assigned to more than one subscription plan (for instance an enterprise customer). Moreover, for each subscription there are one or more STBs assigned. This representative scenario is depicted in Figure 2.2.

Despite all advantages of the IPTV service, a major challenge for this carriers is to realize the extent their customers take advantage of the wide range of contents and services available. The IPTV service has the ability to generate volumes of information that are associated with events from the customers. Events such as: on/off STBs; tune channels, watch live TV, watch recorded programs, browse the menu are just some examples.

Thus, an IPTV event is characterized by several attributes, mainly:

- STB identifier;
- Date and Time (moment);

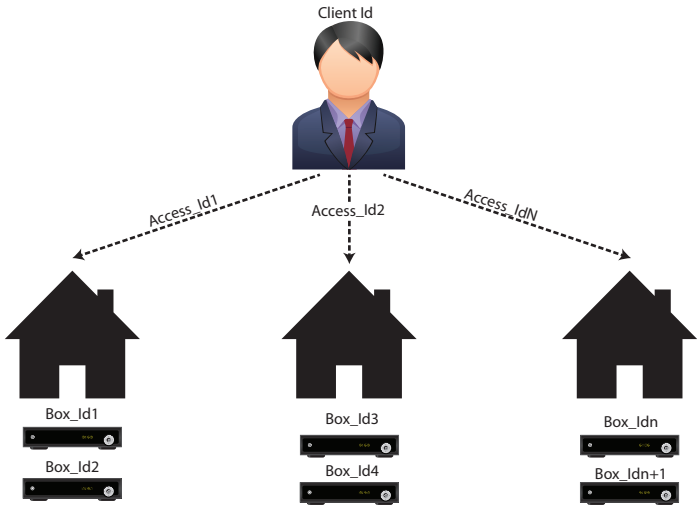


Figure 2.2: IPTV Client Scenario

- Action (Click Recording activity such: a program being watched, a menu selection, a DVR play, etc.);

- Content accessed.

In a simplistic way, when a user clicks on a command button of the remote control, triggers a new event and it is easy to imagine that thousands of scattered events are generated in the same temporal instance, resulting in millions of events in a very short time. So, there are conditions to measure the usage profile of clients over what is watched and used.

Recently approaches have emerged regarding the information usage from the IPTV service, which can be grouped in the following application areas:

- **Modelling User Activities**

  In [QGL+09a] is proposed a simulation tool of user activities, such as turning STBs on and off, switching channels and channel popularity. The data was studied exhaustively and for each user activity was defined a mathematical model that simulates its behaviour. Then, was created a simulation tool in order to recreate the workload of an IPTV system that allows to analyse its performance. The data was used just for the definition of models, having no direct use in the simulator. This tool aims to predict the behaviour of a given IPTV system and improve it according to its performance.

- **Modelling Channel Popularity**

  In [QGL+09b] it is made a detailed study about the popularity of TV channels on their temporal distribution. Like the previous example, were created mathematical models that represent the system activity. In addition, a method is used to identify groups of users with different preferences, allowing to detect temporal patterns at different times and aggregation scales (ranging from minutes to days).

- **Improving Channel Selection Problem**

  In [MC08] the authors argue that understanding the users' channel selection behaviour can be used to improve channel selection experience in live IPTV systems. Actual IPTV providers, offer several channels to users, where it is expected that people browse through a set of channels until they find something interesting (Zapping). So, the authors evaluate this behaviour according to: *(i)* how long do people sample a channel before deciding whether to continue or stop watching; and *(ii)* how many channels do people sample prior to viewing. In this sense, they propose a system that reduces unnecessary sampling, by creating a "hotlist" based on the content real-time popularity and users' past streaming history. The hotlist can be enhanced for each geographical location to reflect the users' locality and viewing preferences. Besides the desired improvements in user experience, they believe this effort allows a better performance in networking perspective.

- **Exploiting Social User Journeys**

  The social context in which the user is placed it may reveal more than what the records indicate. In this sense, in [JMR10] is proposed a IPTV system that integrates connections to social networks. Thus, they intend to explore the interaction client patterns, and realize how important it is, in order to understand his behaviours and trends. In addition to the metrics that contribute to the popularity of a program, others proposals have been associated with the social context of the client: total interface configurations, number of Facebook friends who are using the service, compare the list of favourites with friends, etc.

  And with that, the aim is to determine correlations between the metrics to evaluate the relevance of social context. It turned out that these correlations may be useful, for instance, in predicting which programs would be most popular.

- **Performance Diagnosis**

  Last but not least, in [MGS+09] it is proposed a diagnostic tool adapted to the IPTV network structure analysis, in order to characterize performance problems, using information from error logs, activity logs, alerts and requests for troubleshooting.

  Given the amount of information, it takes advantage of multi resolution techniques with spatial aggregations at various levels to expedite the process of identifying problems. One practical use it's the correlations analysis to identify events that

occurred at the same time and that may have significant impact on symptoms of the previous detected events.

It is clear the effort that has been made around the IPTV data. However, the first referenced approaches basically focus on the activities analyses with highly statistical purposes, a bit similar to the Audiometry analyses 2.2. On the other hand, [MC08] it is close to our motivations, by addressing a clear problem about users television experience. In fact, this approach will be further referenced in chapter 4.1, where we present a more formal definition about Zapping, detailing some interesting issues and discuss the results obtained in order to characterize the impact that this behaviour has on IPTV service.

Another aspect that should be noted, is that the above approaches do not have as main objective to address other IPTV features. They analyse only the visualizing television context, where there is no approach made to the determine profiles based on users' behaviours in terms of applications/features used, as was pointed out in our motivation. Furthermore, actual Portuguese telecommunications carriers recognize their ability to generate IPTV records, but there are no usage of such data, coupled with little interest shown, which results in a closure and low prospect in the analysis of such data. According to one of the top Triple Play provider, measuring the usage of provided features is actually done with online and phone surveys, and analyses of the clients adherence rate to campaigns.

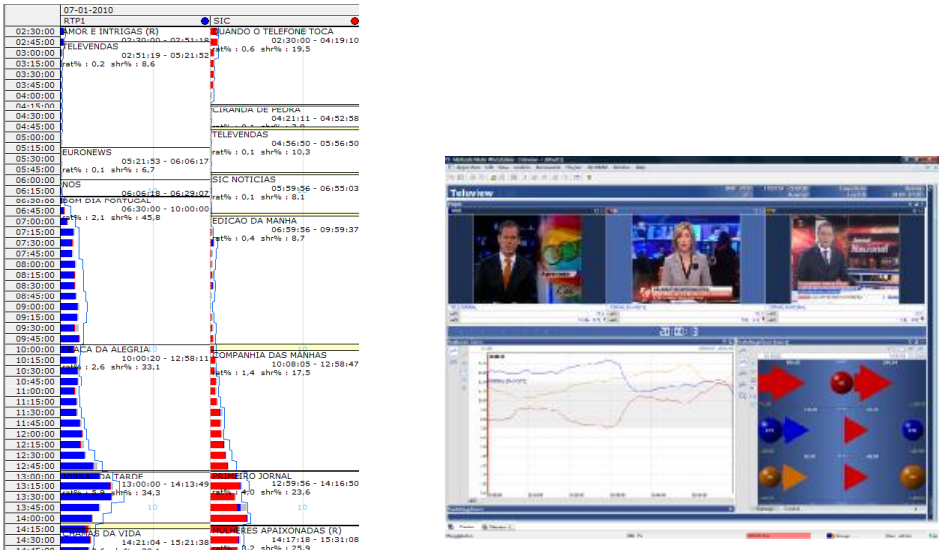
2.2 Audiometry

Nowadays, the Audiometry is widely used as audience measurement to analyse data on television audiences, in terms of watched channels, moment and duration. Also, it is used to manage television content and to schedule TV listings, in order to promote the hearings, as well as, to be used by media advertising to put its consumer products through the profiled target. For this analysis, a representative population sample (ensured by the Census Data) is carefully selected, stratified by region, class socio economic and pay TV holding, which is called a audience panel[3]. Subsequently, the results are extrapolated to the universe, where the pair universe/sample is a common denominator in audience measurement processes.

The data are obtained using a electronic meter system, placed at each panel viewer's home, allowing the determination of what has been seen and for whom. The output of these tests are reports that include several indicators such as average audience, TV ratings, share, average time viewing, and some viewing indexes. These indicators have a statistical purpose only. Currently, there are some tools used to evaluate the audiometry

[3]Methodology usage by Marktest in the audience panel definition. They use a service called Audipanel which provides television audience data for time periods and programs received in Portugal. Audiences are produced from a sample of 1,000 households - designated Panel - that represent the television behaviour of individuals at least with 4 years old, living in Portugal. According to the final data of Census 2001, this sample represents the behaviour of televised 9,459,181 individuals.

data about Portuguese television like Telereport[4] and Televiewer[5].



(a) Graphical analyse in Telereport. (b) Multiple channels simultaneously in Televiewer.

Figure 2.3: Examples of Audiometry Analysis Tools

Some studies have been done on audience measurement processes, which innovate in an attempt to detect regular patterns in these data[DMPCP05], with emphasis in time and content terms, taking advantage of clustering techniques to determine weekday patterns. Other approaches have focused on modelling the viewrs' behaviours in order to predict future audiences [DPL02]. More recently, [Dat06] addressed the contributions that traditional Online Analytical Processing and data mining techniques can bring to the audience measurement in Portugal, given the inherent characteristics of Audiometry information.

The Audiometry tools are actually quite important for audiences studies, with relative accuracy on defining several television watching indexes. But, technological advancements in television service, particularly with the entrance of digital services such as IPTV, launched new challenges to these analysis tools, which currently are still evolving and learning how to address this new type of information. Digital era overturned: what it's seen, how it's seen and how it's measured. So, actually Audiometry systems are outdated due to the new digital systems proliferation. Moreover, these tools are really intended for statistical purposes and are only based for broadcast content. In this sense, Audiometry analyses are not enough to address the new coming challenges, and hence we are suggesting a new approach to this emerging problem.

[4]Television audiences analysis tool (Fig. 2.3(a)), property of MediaMonitor, presented in tables, charts and grids.

[5]Television audiences analysis tool (Fig. 2.3(b)), property of MediaMonitor, that present in simultaneous image, sound, charts and audience values of various channels.

2.3 Data Stream Mining

We can define Data Mining as the process of extracting knowledge from a set of information typically stored in a database. This phase is part of a complex process that has the name Knowledge Discovery in Databases (KDD) which also involves other phases such as: selection, preprocessing, transformation, data mining and interpretation/evaluation [FPsS96]. Actually the use of KDD is spread over several business areas, such as marketing, finance, fraud detection, manufacturing, internet agents and with special interest for us, telecommunications.

This thesis scope deals with large volumes of information, and its main objective is the detection of Visualization and usage profiles from IPTV activity data. Current data are scattered and singular, not representing a particular behaviour. Thus, it is necessary to synthesize the large volume of data and transform scattered events into representative information, favouring what is generic and not what is specific.

Today, many applications have the ability to generate information that is associated with events occurring in a given system. These events have the particularity to be unpredictable when they occur, due to its pervasiveness. Given the increasing number of such applications and agents to act on them, we got to the point where the volume of information being generated is beyond the paradigm of the common database model[6]. Naturally, the information occurs in a sequence of data, or also known as stream. More precisely, the data is modelled best not to persistent relations but rather the transient data streams. It is recognized that this type of information does not fit the conventional DBMS and the queries are not effective [GO03, BBD+02].

Several applications already deal with data stream notion in its information model, for example, financial applications, internet network monitoring, security, web applications, sensor networks, and of course, telecommunications.

This information model has brought new problems in relation to its organization, as well as its analytical perception. Eamonn Keogh et al. [WKL+05] stated:

> *"Recent advancements in sensor technology have made it possible to collect enormous amounts of data in real-time. However, because of the sheer volume of data most of it will never be inspected by an algorithm, much less a human being."*

The need to understand the enormous amount of data being generated every day in a timely fashion has given rise to a new data processing model - data stream mining. These data streams bring unique challenges that make them interesting from a data mining perspective: size, rate of accumulation and data transience. Hence, a wide variety of analytical problems associated with mining and monitoring data streams has emerged, such as: Data reduction; Characterizing constantly changing distributions and detecting

[6] Insertion, updates and deletions operations occurs less frequently than queries. Besides, traditional Database Management System (DBMS) are not designed for rapid and continuous loading of individual data items.

changes in these distributions; Identifying outliers, tracking rare events and anomalies; "Correlating" multiple data streams; Building predictive models; Clustering and classifying data streams; and Visualization [DW08].

As already mentioned, the events resulting from the IPTV activity have a high rate of occurrence per unit time. For instance, imagine the number of customers that are at once interacting with the service, resulting in large volumes of information in a short time. The IPTV activity is a concrete application example where information has a data stream model, given its continuous generation in multiple, rapid and time-varying intervals.

So it makes sense to refer this topic due to its information nature. However, the resulting data stream of television activity can not be used directly for behaviours determination, because the data are too refined that makes it difficult to understand given its dispersion. So, we need to turn the information into a more generic level, that represents an activity running in the STB, i.e. we want to convert a Click Stream to an Activity Stream.

3

Click Stream to Activity Stream

In this chapter we detail the approach presented in section 1.3. Initially, we mention some important aspects that characterize the data to be used, in terms of size, granularity and content. The initial phase focus on a comprehensive survey of the existing IPTV services, targeting them into activities liable to be analysed. Next, we present the approach designed to deal with that information and how to turn it into a higher conceptual level.

3.1 Data Set

We have access to real IPTV activity information service from a Portuguese telecommunications carrier. In this subject we have the support of a market consultant (Novabase) which provided us the necessary support with regard to the provision of data. So far, we only have access to a sample of nine days of activity: 4 contiguous days from February 2012 and 5 contiguous days from April 2012. Regarding the volume of information at hand, those days of activity result in a data set of about 14 million records relating to approximately 53.000 STBs. There are situations where a client is assigned to more than one STB and it is expected that the number of customers is relatively lower than the number of STBs.

Each record has 24 attributes, which are illustrated in the Figure 3.1, grouped by types of information:

- **Who** - STB identifier;
- **When** - the event occurrence moment;
- **Channel** - the channel where a given event occurs;

- **Content** - what content was accessed;
- **Action** - identification of the action that triggered the event;
- **Duration** - event duration.

The IPTV service is based on Microsoft Mediaroom platform[1], which provides a framework that enables developers to build differentiated TV applications that extend and augment the Microsoft Mediaroom TV experience. So, it is expected that the kind of data available in this system we are working on it's similar to other ones in different carriers. Apart from that, there are some different STBs versions currently in use in the same operator, where it's possible that data is generated quite different among them. In particular, STBs without some features (DVR), cannot generate the some events as a STB with that feature.

It is possible to highlight the presence of the basic attributes already mentioned in 2.1.2: Identifier of the STB as CLIENT_ID; Date and Time as COD_DATE_GP; Action as EVENT_TYPE (for instance: Box Power), which is detailed in ACTION as (On or Off); Content accessed as CONTENT_ID. The remainder attributes are particularities of this carrier.

The events detected in this IPTV service are: channel tune, box power, menu selection, program watched, DVR recording operations (start, abort, playback, schedule, delete and cancel), and Video on Demand (VOD) purchases. However, there are other events documented by this operator that are not actually used: Remote Desktop Protocol (RDP) and Pay-per-View (PPV) purchase, which gives emphasis on the fact that our approach should be generic and not limited to match a predefined set of feasible actions able to detect/analyse, in order to enable the integration of those future features. At this stage, the events considered are the ones highlighted in Figure 3.2. Each event type is segmented in actions that detail a little more each one. For instance, the event type "Box Power" could have two actions associated: On and Off; the event type "DVR Playback" have several actions associated as: Play, Pause, Fast Forward, Replay, Rewind, Skip and Stop. Another interesting attribute is the SERVICE_TYPE which defines the service being used (Live, VOD or DVR) as a promising indicator of the kind of activity occurring.

There are some attributes which denotes only the identifier, not bringing significant value, for example: CHANNEL_ID and CONTENT_ID. So, we need additional information to map those identifiers into something more meaningful. In this sense, we also have access to some external entities: Access (information about client access: Asymmetric Digital Subscriber Line (ADSL) or Fiber to the Home (FTTH), activation date, and some commercial aspects); Asset (information about purchased contents - VOD operations); Channel (information related to a particular channel: name, number, type); and Content (here we just have content name). However, all the information that was provided were

[1]Microsoft Mediaroom is the world's number one IPTV platform, used by several companies. More details available at: http://www.microsoft.com/mediaroom/

Name	Description
CLIENT_ID	Set-top box identifier
CLIENT_TYPE	Set-top box model
COD_DATE	Event source date in YYYYMMDD format
COD_DATE_GP	Event source date and time in YYYYMMDDHH24MI format
COD_START_GP	Event source time in HH24MI format
SOURCE_TIMESTAMP	Event source timestamp
EXPIRATION_DATE	Date and time that the subscriber's access to the rented VoD asset expires
ACTION_TIMESTAMP	Date and universal time that the event occurred
FREQUENCY	Frequency of the recording
CHANNEL_ID	Channel Id for event type: Program Watched or DVRs events
CHANNEL_NBR	Channel number
CONTENT_ID	Id of the asset, program or channel depending on the event type
EVENT_TYPE	Mediaroom event type (Channel Tune, Box Power, VOD Purshace, Program Watched, etc.)
SERVICE_TYPE	Service type ('LIVE', 'DVR', 'VOD', etc.)
VIEW_MODE	Event type of stream (full-screen or PIP) and service type (primary or secondary) of the media session
ACTION	Action invoked by the subscriber
ACTION_STATE	State of the action (up or down)
MENU_ID	Id of the menu
DYNAMIC	Value indicating a dynamic or a manual recording
RECURRING	Value indicating a one-time or a recurring recording
INSTANCE_OF_RECURRING	One (1) means that the subscriber canceled a single instance of a recurring program
MANUAL_DELETION	Indication if the recording was deleted by the subscriber
TUNE_ID	Unique Id that links channel tune events with program transitions
DURATION	Event duration in seconds

Who Channel Action/Activity
When Content Duration

Figure 3.1: IPTV Data Description

scattered in various files and different formats, and was needed to assemble it, in order to achieve a model more consistent and easy to understand, which resulted in the proposed data model depicted in Figure 3.3. The main entity of this model is the Record table, referring to the IPTV events. At this stage, we took advantage of some preprocessing to change some attributes mainly on data type level and attribute filtering. All the external entities mentioned above are also present in this model. Just a reminder to the relation between Asset and Content: all Assets are included in Contents, because some contents are specific to VOD service, where for each Asset we have more detailed information as depicted on the model. Finally, the natural organization of some "commercial entities" (access, client) are depicted as a hierarchical relationship in this model: one IPTV record

Event type	Description
100	Channel Tune
101	Box Power
102	VOD Purchase
103	RDP Purchase
104	Trick State
105	Browse Panel
106	Application
107	Menu Selection
108	RDP Application Launch
109	RDP Application Disconnect
110	RDP Application Navigate Away
111	RDP Application MCE Tune
113	PPV Purchase
114	Program Watched
115	DVR Start Recording
116	DVR Abort Recording
117	DVR Playback Recording
118	DVR Schedule Recording
119	DVR Delete Recording
120	DVR Cancel Recording

Figure 3.2: IPTV Events Detected

is associated to a single STB, each STB in turn relates to an access, which it is final related to a customer.

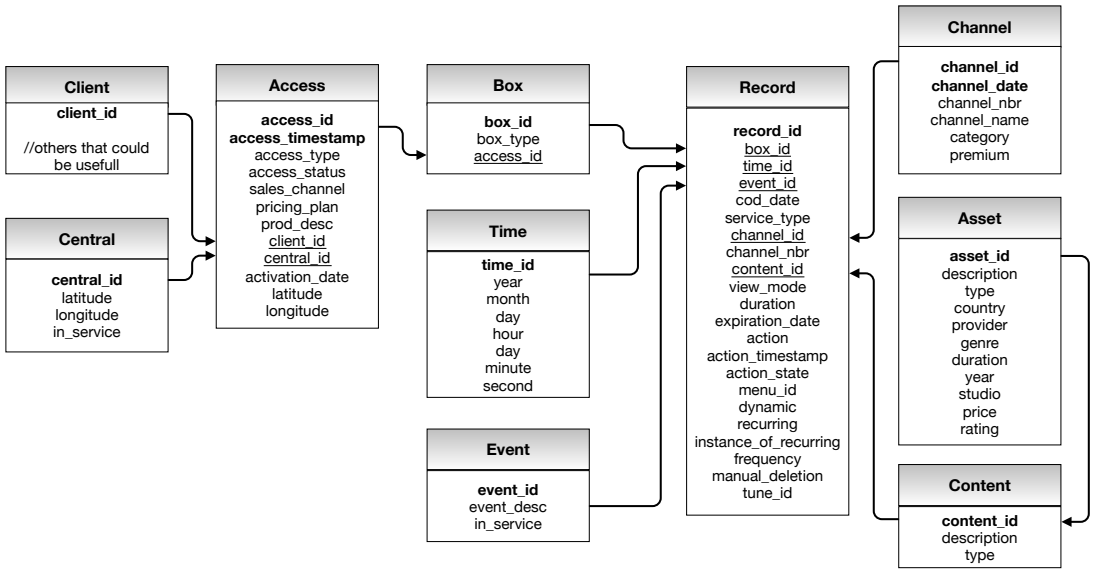


Figure 3.3: IPTV Data Model

One problem in the data is the existence of errors. As we are talking about a system that generates events for a data repository, there is no guarantee that the arrival order of those events is always the same. In particular, on activities characterized by a sequence of events (described in 3.2), they have an identical time of occurrence, influencing the

definition of event patterns (3.3). For instance, the live Visualization activity is characterized by a sequence of two events: event id = 100 followed by a event id = 114. But it's usual that those sequences arise in reverse order. Another recurrent situation is the lack of events, or events that come loose without realizing their meaning, because do not fit in the context of the activity taking place. Sometimes, some events occur like a menu selection (event id = 107), which we cannot ensure that are actually relevant to the activity taking place. A final issue is related with the temporal spectrum in which some of the activities take place, where only makes sense to consider an activity if it occurs within a certain time limit. Every time that a particular content ends, automatically a new record regarding the next one starting (when a user remains in the same channel) is generated. But sometimes, we catch Visualizations whose duration lasted for more than 8 hours, which is, at least, strange. For some reason the STB stopped to generate events, causing these situations. For simplicity and consistency in the definition of activities, these situations were ignored.

3.2 Survey of Activities

Based on captured events by the IPTV system, we followed a top-down approach in order to make a collection of feasible activities. This approach is defined by the characterization of the expected activities from a TV user, based on the offered features. This way, we can incorporate activities into three main classes: *(i)* **visualization**; *(ii)* **recording**; and *(iii)* **video on demand**. With respect to other existing features such as interactive applications (Television Widgets) or STB Services Management, will not be considered because actually those events are not captured. In future, they can be easily incorporated.

The visualization class includes all activities related to contents Visualization: live broadcasting, DVR playbacks (view of contents previously scheduled) and VOD playbacks (view of contents previously purchased). The distinction is made through SERVICE TYPE attribute. In fact, this class is the most important, because it is the one that represents the primary television activity. Generally, each Visualization activity is characterized by two events: one channel tune (event type: 100), and one program watch (event type: 114). However, following this pair of events may occur: *(i)* channel tune event - indicating that the user has finished the previous Visualization, and start a new one in another channel; *(ii)* program watch event - when a new content start, remaining on the same channel; *(iii)* another events that might indicate the end of Visualization activity (turn off STB), or *(iv)* parallel events like the auto-start of DVR recordings which do not compromise the end of current Visualization activity. Thus we intent to understand how each visualization type (Live, DVR and VOD) is characterized, relating the time spent in it, what kind of channels/contents have greater interest to users, and inherent behaviours for each type of Visualization. In particular, in section 4.1, we explore an expected behaviour on the live visualization: Zapping, from which we draw some interesting results.

Regarding the recording class, the identification of user actions is much simpler, since there is a corresponding event type associated to each of DVR operations (see Figure 3.2 - events 115-120). In addition, these events contain interesting attributes that allow to understand: if the DVR recording has started automatic or manual (DYNAMIC attribute); or for instance, if the scheduled DVR recording is just for one episode or all season (RECURRING attribute). Although a start DVR recording activity is independent from a delete DVR recording activity, we can combine them with a playback DVR recording, and try to figure out some users' behaviours. In particular, if we represent a DVR activity as a sequence (schedule-record-watch-delete, for instance), it is possible to analyse if the scheduled DVR recordings are effectively watched later, or even, if the users care about DVR recording managements (delete after watched it). In section 4.2 we focus our attention on these behaviours analysis.

Finally, in the video on demand class there are two visible behaviours: *(i)* VOD purchase (event type: 102); and *(ii)* watch trailer (videoclub activity) - represented by a sequence of events: tune to videoclub channel, followed by a menu action, ending with a program watch event (the content itself). Over this class, we could measure the usage rate of VOD, in terms of number of TV contents purchased, average amount spent on rentals, and so forth.

3.3 Transformation Process

The first problem that we faced was how to work the existing information in IPTV records. The information is generated at a very fine granularity, since each click in the command given by the customer leads to a new event. As you can imagine, this causes that many records have no meaning. What we are interested here is to recognize activities taking place in a given STB: watch a film, schedule a recording, accessing the video club, etc. And for each of these activities, there may be many events dispersed that in essence boil down to an activity taking place. For example, remember the watch trailer activity, which requires that customer follow some steps (clicks) in STB command. And instead of having these events, we want something more generic and meaningful, which summarizes the activity: Watch trailer. This leads to the need to compress the information, generalizing it to a higher conceptual level, in order to capture activities that occur in a given STB.

This need to transform **Click Streams** on **Activity Streams** brings some advantages in analytical terms, standing out:

- First, as each customer click action gives rise to a new event, it is easy to imagine the sheer volume of data we have to work. And so, by aggregating information to activity level, we will substantially reduce the final data size. In the limit, a sequence of N events that characterize an activity, it's reduced to just one event.
- Some of the events generated may not be relevant to the type of activities that we

want, so this transformation also allows filtering only data that in fact have interest;

- Singular events by itself may mean nothing, but when viewed in an sorted time sequence can be significant in the recognition of a given activity;
- A given event is only associated with the time of its occurrence, where is not possible to infer anything about duration, i.e. to know how long a particular activity lasted;
- A certain activity can be characterized by distinct sequences, which in essence, represent the same activity.

For these reasons, we believe that this transformation of clicks to activities is essential to have another notion and knowledge that the raw data can not give. Basically, we would like to turn data into information.

3.3.1 Complex Event Processing

The type of information that we are dealing, presents a characteristic very common in current systems, that is the ability to generate large volumes of data in a short time period, and in a continuous rate. This feature may be viewed as a scattered and complex data stream. Hence, the suitable way to handle this information is to interpret it in Complex Event Processing concept. Although this concept has a strong relation to a "real time" component, its purpose is mostly to produce information in the "right time".

The main difference and advantage of these CEP tools over relational databases is the ability to deal with information of temporal nature, since they require a specific query to return an information set. That is, store the data on which a query is executed. One the other hand, the CEP tools store queries where it is possible to specify a time window, over which the data is passed as the time progresses. This stream is submitted to filtering, typically by event pattern detection.

As you can imagine, the application of a CEP system is applicable to diverse areas such as Business Process Management, Financial Services Industry, Time Series Databases, among others. Therefore, it is easy to make the bridge between the CEP systems for our need to transform Click Streams into Activity Streams, since we have a large volume of events assigned to the user clicks on a short unit of time (seconds) over which we want to extract information representative of user's activity.

Although these tools have the main purpose of application in real time data and detect events, our approach takes advantage of the CEP concept from a slightly different way. Basically, current IPTV systems only dump the data at the end of a day, which prevent us to use it at real time. However, the approach designed is easily applicable to real time context, so that if the events logs delivery were instantaneous there would be no problem at all.

In short, we want to take advantage of the capabilities of these tools and adapt these principles in an environment where the time factor is controlled by us (due to data dump

strategy), where in the end we have a repository of activities captured by the tool, and then use that richer information set, in a most general and relevant way, for analyses purposes.

3.3.2 Esper

Actually the are some emerging CEP tools, either commercial: TIBCO BusinessEvents™[2], Streambase[3] and Feedzai Pulse[4], or open source solutions: Drools Fusion[5] and Esper[esp]. For obvious reasons, commercial solutions were not considered. On the other hand, both open source solutions present features in common very similar, but the events declarative language of Drools Fusion is not so user friendly like the Esper. Furthermore, the documentation and support are most exquisite and complete in Esper.

So, Esper was the chosen CEP tool to address the problem. It allows easy integration into any Java application, as well as its features extension with our own code. This component is known as an Event Stream Processing and Event Correlation Engine intended for real-time Event Driven Architectures capable to trigger actions when certain events conditions occur on those events streams. It is especially designed to cope with very large data volumes (millions). To be able to express conditions on such events, it has a Domain Specific Language (Event Processing Language (EPL)), very similar to SQL style, allowing to express richer conditions, correlations, declare sliding windows, among others.

The operation concept of this tool is simple. The events we are interested to detect must be represented by any of the underlying Java objects depicted in Figure 3.4. Another important aspect is that Esper offers the possibility to configure the engine to use or the internal CPU clock - when you want to work in real time, or external CPU clock - which only requires that the object representing the event has an attribute with timestamp data type, enabling the time factor to be controlled by us. This is a feature that has particular interest in this work, since we use data from different times (February and April 2012), where is the event timestamp that makes time to go forward.

So, let us state some important keywords:

- **Event Type** - Represents the Java object that encapsulate the event we are working. In our case, that event is only the IPTV record, as we described in section 3.1;
- **Statement** - The processing engine is based on a set of continuous queries. Each activity we want to detect (the ones referred in section 3.2), must be expressed in

[2]TIBCO BusinessEvents uses a model-driven approach to collect, filter and correlate events with respective business processes, and to deliver real-time operational insight enabling timely, well-informed decisions. More details at: http://www.tibco.com/products/event-processing/complex-event-processing/businessevents/default.jsp

[3]Streambase Event Processing Platform™ is a commercial solution which has an integrated development environment (IDE) and uses a graphical workflow model. More details at: http://www.streambase.com/

[4]Feedzai Pulse is another commercial solution, that works over the Esper engine with special focus in real-time dashboards. More details at: http://feedzai.com/

[5]Drools Fusion is part of Business Logic Integration Platform from JBoss Community project. It has a native rule language which requires some additional learning. More details available at: http://www.jboss.org/drools/drools-fusion.html

EPL as a Statement. This Statement is set as a string which specifies the attributes of the Event Type we want to work, and also the filtering operation to be applied using regular expressions or patterns. Finally, this Statement is added to a Listener.

- **Listener** - For each Statement defined, it is expected to have a Listener that is invoked each time a given sequence of processed events falls under the conditions specified in the Statement.

Java Class	Description
`java.lang.Object`	Any Java POJO (plain-old java object) with getter methods following JavaBean conventions; Legacy Java classes not following JavaBean conventions can also serve as events .
`java.util.Map`	Map events are implementations of the `java.util.Map` interface where each map entry is a propery value.
`Object[] (array of object)`	Object-array events are arrays of objects (type `Object[]`) where each array element is a property value.
`org.w3c.dom.Node`	XML document object model (DOM).
`org.apache.axiom.om.OMDocument or OMElement`	XML - Streaming API for XML (StAX) - Apache Axiom (provided by EsperIO package).
Application classes	Plug-in event representation via the extension API.

Figure 3.4: Event Underlying Java Objects

3.3.3 CEP + ESPER + IPTV

Since we have already chosen the approach and the tool to use, it's time to put all things together. From the data repository shown in Figure 3.3, the table *Record* contains all Click Stream events from users actions. So, the first step is to map this entity to a Java object in agreement with Esper requirements. In this sense, the entity *Record* was represented as Plain Old Java Object (POJO) (following JavaBean convections, see Appendix A.1.1). Then, for each activity identified in section 3.2, we created a Statement that expresses the conditions which define the pattern or sequence of events that characterize the desired activity (see Appendix A.2). Each Statement created has a corresponding Listener which is invoked each time a given sequence matches the conditions specified. Strictly speaking, each Listener has a List, and each time that it is triggered, adds the actual captured information to the List. At this stage, the final step is send all raw data to Esper engine.

The captured activities sequences are dumped to a Database, in three distinct tables, according the class of activity: visualization, recording and video on demand. Those tables contains only the attributes needed to characterize the corresponding class of activity

(DVR, VOD and Visualization). The transformation process is depicted in Figure 3.5.

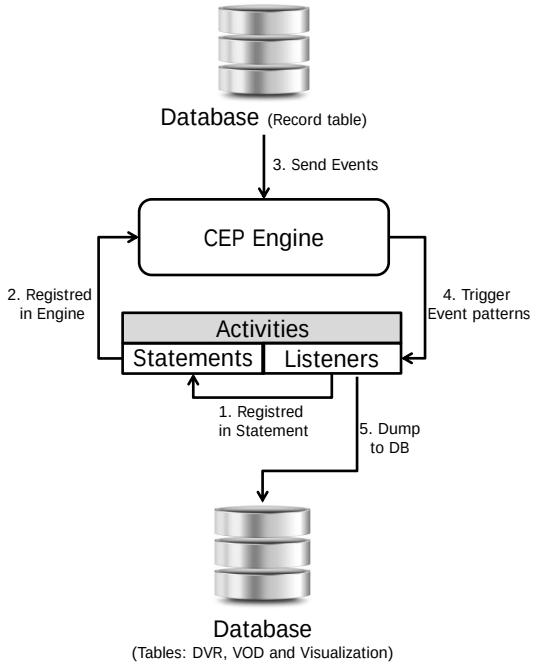


Figure 3.5: Click Stream to Activity Stream Processing

So the extractor proposed above is able to, for each STB, identify sequences of events, in order to generalize them in the respective activity. The result obtained is a listing of activities that occur in that STB. More than getting a richer and valuable conceptual level of information, this approach responds to the need to reduce the volume of data at hand. And so, by aggregating information to activity level, we substantially reduce the final data size.

As can be seen from Figure 3.6, there was a noticeable reduction of data cardinality. Of about approximately 14 million unique and disperse records, we got slightly more than 4 million records, assigned to classes of activities representing the majority of users' behaviours.

Clearly, the most representative activity is the Visualization of live broadcasting (the main functionality of IPTV service), and the one that suffered more cardinality reduction. Remember that, this activity is represented as a sequence of events (see section 3.2), and now is mapped only in a single record. Furthermore, many events that do not fit in the context of an activity are ignored like: menu navigation, content navigation (play, pause, rewind, skip). Thereby, the representation became more simplified and valuable.

VOD operations activities also suffer a slight reduction with the previous transformation. Remember, for instance, that watch trailer activity is characterized by a sequence of three events, and now we just have only one, i.e., at least $\frac{1}{3}$ of original data was reduced. Similar, the VOD purchase activity was originally characterized by a sequence of two events, and now we only have just one. Here, there was a reduction of 50%.

Regarding the processing time, this Click Stream transformation took on average $\approx$

Activity Class	Activity Stream	
	Activity	#Records
Visualization	Live broadcasting	3.600.000
	DVR Playback	128.000
	VOD Playback	5.200
Recording	Start Record	279.000
	Delete record	253.000
Videoclub	Rent a film	3.000
	Watch Trailer	2.500

Figure 3.6: Click Stream to Activity Stream Transformation Result

150 seconds per 24 hours of television day, which is not a bad result, since we are processing several millions of events, relating to nine days, with multiple Statements threading in the CEP engine, regarding distinct class of activities. So far, our approach enables a transformation phase from original data (raw data) to a conceptual level of activities at several areas: Visualization, DVR and VOD. Figure 3.7 sumps up the transformation done.

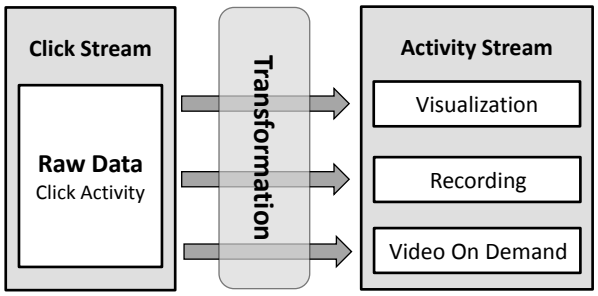


Figure 3.7: Click Stream to Activity Stream Transformation

However, the amount of data assigned to visualization class activity can be further reduced without any loss of information. But first of all, it is important to define some aspects that help to understand and clarify terms related to this topic. **Visualization** is the act of user views a given content in a certain channel, i.e., a viewing period of time which we call a visualization moment M. Each moment M is defined by the following proprieties:

- **Start** - timestamp related to starting instant of a given moment - which can be represented as function $start(M)$;
- **End** - timestamp related to ending instant of a given moment - which can be represented as function $end(M)$;
- **Duration** - how long a given moment takes - which can be represented as function $d(M)$, i.e., $end(M) - start(M)$;

- **Channel** - the channel where a given moment occurs - which can be represented as function $ch(M)$;
- **Content** - the content associated to that moment - which can be represented as function $ct(M)$.

A contiguous period spent by the user viewing TV, consists of several moments M, resulting in a contiguous sequence of Visualization Moments. Clearly, a given user is characterized by a set of sequences scattered throughout a day. Note that, each visualization record represents one single Visualization Moment. So, if we group contiguous Visualization Moments in one new single record (a sequence), we will substantially shrink again the total size of data, without any loss of information. In fact, the data become richer in information since we can add some interesting additive metrics, like: *(i)* total moments by sequence; *(ii)* total time spent by sequence; *(iii)* average daily visualization sequences, among others, which allows a better knowledge about users behaviours.

So, in order to accomplish this, we make another transformation process, similar to the one already done. Generally, the output from the transformation process serves as input to the next one. For this new transformation, we used the Visualization table as input. Again, we created a new Statement and a corresponding Listener, but this time, in a much simpler form, because we used the Esper engine just to receive events and dispatch to a Listener, but all the hard process to build the sequences was made by us. Then we just incorporate the code within the Listener. This new transformation phase, led us to reduce the data in more than 35%, which is a significant result. The figure 3.8 sums up the key features of this new transformation phase.

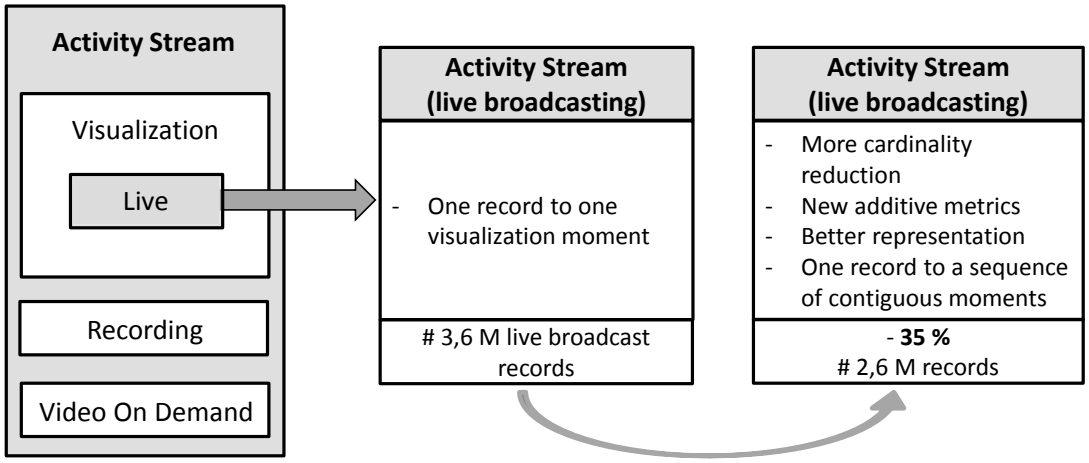


Figure 3.8: Summary of Live Broadcasting Iterative Transformation Process

In this sense, it is possible to generalize that process and turn it in an iterative transformation process, where at each transformation we can reduce even more the total amount of data, as enriching the information level. The new version of transformation process is represented in Figure 3.9. Note that, at each transformation level we can apply data analysis techniques (for instance: data mining, statistical analysis) depending on the desired

context of analysis. This iterative transformation process allows us to increase the level of detail as needed.

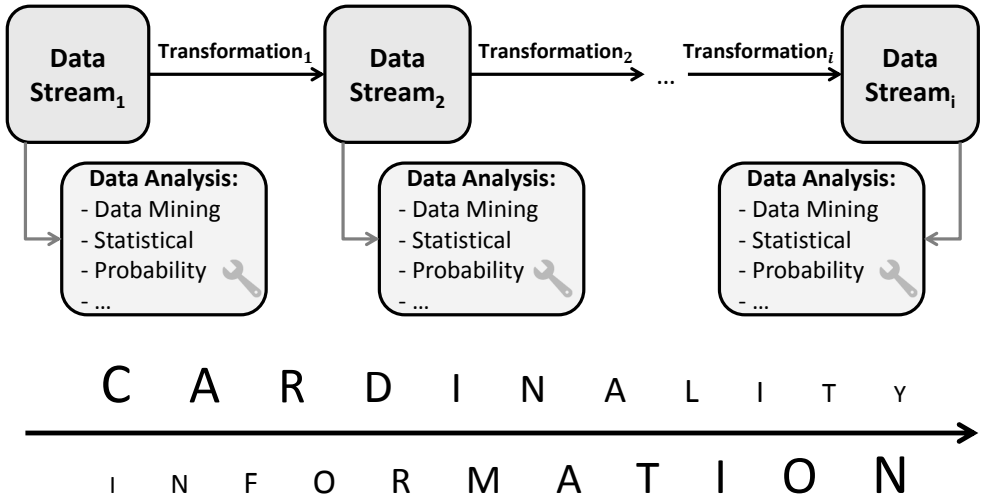


Figure 3.9: Click Stream to Activity Stream Iterative Transformation Process

The approach described here marks the first phase of work. However, the approach conceived to information generalization does not relate exclusively to the IPTV context, but also to other ones, like click web context: where we can measure user navigation and qualify a web content through user click activity. Clearly, the click level has a finer granularity, but when seen as a particular activity (reducing the amount of data, and with a higher conceptual level), we substantially improve the perception ability.

Following this, the next phase will take the output obtained earlier, where each user is now well characterized in terms of kind of activities performed by him. And the aim is then to analyse consumption behaviours of the IPTV service on a customers' universe. In particular, the next chapter presents some application areas, that take the output obtained earlier to further apply the iterative transformation process. We start with Zapping, using the output from the last transformation process iteration (Figure 3.8). Then we draw our attention to DVR service. There, using the data derived from the first iteration (Figure 3.7), we merge Visualization and Recording classes to depict some DVR usage behaviours.

4

Application Areas

The work so far aims to generalize the Click Stream events to a more conceptual and valuable level in terms of analysis. In addition, it allows reducing the data cardinality substantially. Following this, the next phase will take the output obtained earlier, where each STB is characterized by a set of well defined activities.

In this sense, this chapter presents some application areas in which we can explore the information retrieved so far. First, we explore an expected behaviour inherent to live Visualization activity, which is Zapping. The problem of quickly finding the right channel becomes harder as the number of channels offering grows in actual IPTV systems. So we try to figure out this behaviour in daily live broadcasting. Afterwards, we present a detailed study about DVR activity, in order to understand how people is taking advantage of that feature, which is one of the most attractive in current IPTV service offering.

4.1 Zapping

Although there are several features in IPTV service, it appears that the predominant activity is the live broadcasting. Thus, we intend to explore in greater detail this activity, trying to understand what the users' preferences are, and behaviours involved. One of them, is the channel surfing, also known as Zapping, which it is expected in services that offer multiple channels, where many do not fix the users' attention. In this sense, we will present a more formal definition about Zapping, detailing some interesting issues and discuss the results obtained in order to characterize the impact that this behaviour has on IPTV service.

4.1.1 Zapping Definition

Remember the last iteration of the transformation process presented in the previous chapter (section 3.3), where we grouped contiguous Visualization Moments in one single record, which we designated a sequence. In this chapter, we refer to those sequences as **visualization sessions**. So, a given Visualization Session can be represented as:

$$S = \{M_1, M_2, \ldots, M_m, M_{m+1}\}$$

And the only assurance we have about those sequences is that every moment M_{i+1} is contiguous to M_i. Adapting the formulation outlined in previous chapter, in order to analyse the Zapping behaviour, each **Zapping Session** is therefore defined as a sequence of moments that contain a duration less than a predefined threshold (*minVisualizationTime* - which symbolizes the minimum watching time that is considered acceptable to infer that the user is actually interested in that content). As Zapping is a hopping behaviour between channels, which involve more than one moment, the transition between two channels is called: $Hop\langle M_i|M_{i+1}\rangle$ = hop from $ch(M_i)$ to $ch(M_{i+1})$. Thus, $d(M_i)$ represents the Visualization time for that channel until a hop to another channel is made.

So, the session is being built until rich a Visualization with a duration that: *(i)* exceeds that limit; *(ii)* and is K times[1] superior the minimum between the average of the previous Visualizations and *minVisualizationTime*. In other words, a Zapping Session is defined by a sequence of m Zapping Moments, ending with a Visualization Moment M_{m+1} (the one that fixed the user). However, if the user never reach a Visualization Moment that fixes him, the session ends up with only Zapping Moments. Any moment M_{m+1} that is not immediately contiguous to M_m, will stay in another distinct session.

Formally, based on the Visualization Session definition, a given Zapping Session is defined as:

$$S = \{M_1, M_2, \ldots, M_m, M_{m+1}\}, \text{ with the following proprieties:}$$

1. Adjacent Moments have different channels:

$$\forall M_i \text{ with } i \in \{1, \ldots, m+1\}, ch(M_i) \neq ch(M_{i+1})$$

2. A Zapping Session must have, at least two Zapping Moments:

   We think that is incorrect to infer that a particular session with only one Zapping Moment means Zapping behaviour, because when people turn on TV, it is expectable that they tune, at least one channel, i.e., $m \geq 2$.

[1] The K factor will be detailed in next section, where its use allows a greater accuracy in Zapping definition.

3. Every Zapping Moments must have a duration less then:

   If we are starting a new session ($m = 1$):

$$d(M_1) \leq minVisualizationTime$$

   If the session contains at least one Zapping Moment ($m \geq 2$):

$$\forall M_i \text{ with } i \in \{1, \ldots, m\} :$$
$$d(M_i) \leq minVisualizationTime$$
$$\vee$$
$$d(M_i) \leq Min\left(K * \frac{\sum_{i=1}^{m-1} d(M_i)}{m-1}, K * minVisualizationTime\right)$$

Note that, if the user reach a Visualization Moment that fixes him, the session ends up with a Visualization Moment that:

$$M_{m+1} \text{ with:}$$
$$d(M_{m+1}) > minVisualizationTime$$
$$\wedge$$
$$d(M_{m+1}) > Min\left(K * \frac{\sum_{i=1}^{m} d(M_i)}{m}, K * minVisualizationTime\right)$$

In order to analyse inherent characteristics about Zapping, it is quite important to have some indicators. So, on each sequence we added the following metrics:

- **Zapping Time** - total time spent in a given Zapping session: $ZT = \sum_{i=1}^{m} d(M_i)$;
- **Number of Hops** - total of Zapping moments in a given Zapping session: $H = m$;
- **Average Zapping Time** - average zapping moment duration: $\frac{ZT}{H}$;
- **Sequential Zapping** - number of hops between adjacent channels (increasing or decreasing order);
- **Random Zapping** - number of hops between non adjacent channels (direct or random tune).

Beyond this metrics, it is also possible to add information like: *(i)* the hopped channels; *(ii)* which channel originated the current session; *(iii)* and which channel/content fixed the user's attention.

Despite this Zapping Session formulation, it is important to highlight that the previous Visualization Sessions are also depicted in this formulation. Remember that, Zapping Sessions are a special case of Visualization Sessions in which we just imposed some constraints to distinguish what is Zapping and what is Visualization. In practice, a given session is assigned as Zapping if it has at least two Zapping Moments, otherwise, it is a normal contiguous Visualization Session.

4.1.2 Zapping Evaluation

With the formulations presented earlier, our attempt is to evaluate how the Visualization activity performs on each single STB, and try to characterize it with the metrics we previously presented. In this sense, we focus the Zapping analyses answering the following questions:

1. Average daily Zapping Moments by session?
2. Average Zapping time by session?
3. Daily Zapping Sessions?
4. How is the Zapping behaviour along different channels categories?
5. What is the correlation between Visualization time and the number of Zapping Moments?

In our study, some evaluations were made with different parametrizations of *minVisualizationTime* value: 30, 60, 90 and 120 seconds. In order to have a greater reliability in the results, we use a K factor. The K value is related with the average Visualization time concerning the *minVisualizationTime* specified, where the K factor multiplied by that average time matches the duration considered (see Figure 4.1). In this way, Visualization Moments will be considered more accurately, since we force that it must have a duration K times superior the average Zapping time, and in this manner we better distinguish the two types of moments. The use of this K factor has especially impact when we are dealing with a duration which it is very close to the threshold predefined. We must emphasize that our goal is not to define the values that define the accurate Zapping values, but rather to understand the impact they have on this expected behaviour, over Visualization activity.

$minVisualizationTime$ (seconds)	$AverageVisualization$ Time (seconds)	K Factor
30	14	$K * 14 > 30 = \mathbf{3}$
60	26	$K * 26 > 60 = \mathbf{3}$
90	34	$K * 34 > 90 = \mathbf{3}$
120	41	$K * 41 > 120 = \mathbf{3}$

Figure 4.1: Parametrizations chosen to Evaluate Zapping Behaviour

The first observation was to realize what is the distribution of Zapping Sessions versus non Zapping Sessions (Figure 4.2). We can see that the volume of Zapping Sessions increases with the value of *minVisualizationTime* parameter. This result is understandable, since as we are dealing with an experimental procedure with parameter variation,

the increase in Visualization time duration allows a greater number of moments to be considered as Zapping.

However, it should be noted that, although 120 seconds be a short period of Visualization time in a television context, about 17% Visualizations are assigned to Zapping activity.

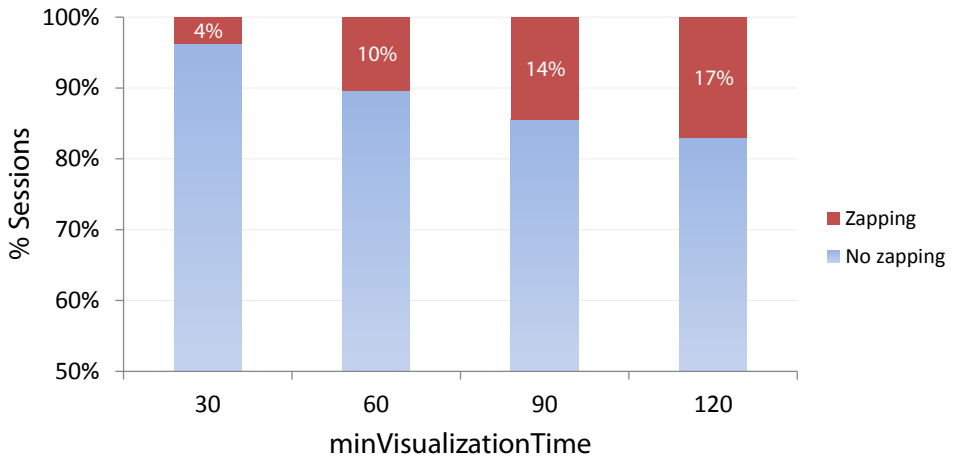


Figure 4.2: Percentual distribution of Visualization Sessions to different *minVisualizationTime*

So, given the representativeness of Zapping, it is important to understand other features underlying this behaviour, like:

Average daily Zapping Moments by session?

Figure 4.3 shows[2] the distribution of the average number of daily Zapping Moments, prior to watching a particular channel for longer than *minVisualizationTime* predefined.

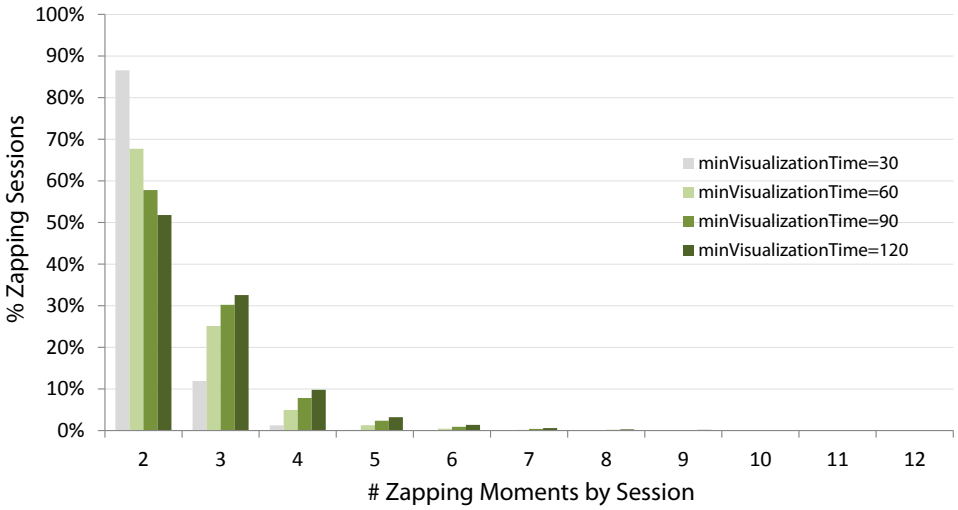


Figure 4.3: Average Daily Distribution of Zapping Moments by Session

[2]For better representation, the figure only shows the distribution up to 12 moments, since that for greater values, the percentage is too low.

Mostly, users sample 2 channels on average before a Visualization that fix them. This distribution is quite similar to a negative exponential distribution, so although there are few sessions with many moments (up to 27 with *minVisualizationTime*=120), they are singular cases and practically no representative. More precisely, the cumulative distribution up to five Zapping Moments covers almost the entire distribution (Figure 4.4), which depicts that Zapping Sessions are short. Moreover, independently of *minVisualizationTime* value, the average of Zapping Moments is almost the same and mode = 2, as depicted in figure 4.5.

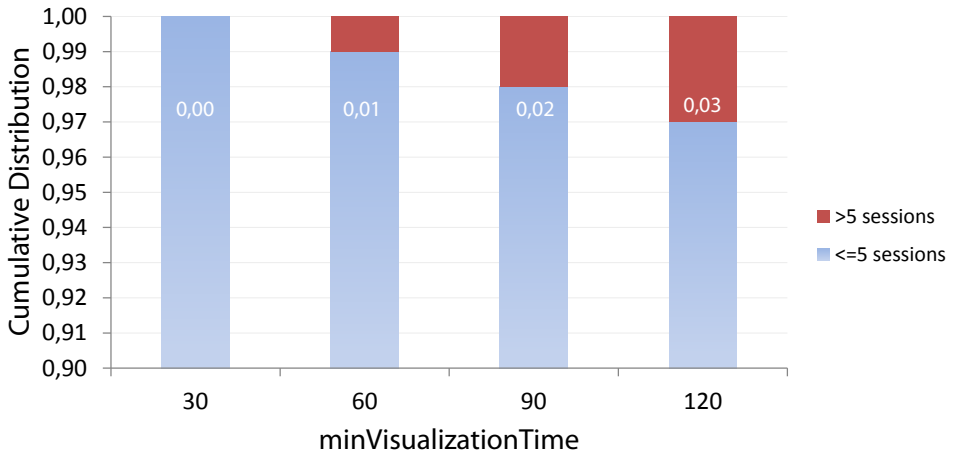


Figure 4.4: Cumulative Distribution up to Five Zapping Moments by Session

Parameter	**Number of Moments**				**% Records**
minVisualizationTime	Minimum	Maximum	Mode	Mean	(at most frequent)
30	2	8	2	$\approx 2,15$	$\approx 87\%$
60	2	23	2	$\approx 2,43$	$\approx 68\%$
90	2	26	2	$\approx 2,62$	$\approx 58\%$
120	2	27	2	$\approx 2,75$	$\approx 52\%$

Figure 4.5: Statistical Comparison for Zapping Moments Distribution

Yet, it seems there are different behaviours in this distribution. On the one hand, the distribution of two Zapping Moments decreases the greater the *minVisualizationTime* value; On the other, the distribution for the rest of Zapping Moments increases the greater the *minVisualizationTime* value. So, let us point out some aspects regarding this behaviour:

- The higher the value of *minVisualizationTime*, the greater the number of moments considered as Zapping. This means, that a significant part of Visualization Moments for low *minVisualizationTime* values, will now be considered as Zapping ones.
- In particular, the average Zapping time by session for *minVisualizationTime=120* is approximately 41 seconds, which is 3 times less that the limit (120). So, there shall be a great number of Zapping Moments. As those moments are contiguous, makes

the number of moments by session greater than the previously registered for *minVisualizationTime=30*. This justifies the difference between the various instantiations of *minVisTime* threshold.

- With this, we can not say that there is a different behaviour among number of moments by session (as it appears in Figure 4.3), since it is just a consequence of the chosen values for the instantiation of the *minVisualizationTime* parameter.

Average zapping time by session?

Besides the total of Zapping Moments, the time spent per session is also a characterizing metric of this behaviour. So, totalling the average Zapping duration for all sessions from a given STB, and grouping the results for all STBs, we have the distribution of the average Zapping time.

In this part, we also compared if the average Zapping time across sessions is consistent with the average time in sessions with only two moments (since those are more prevalent). Thus, we analysed the distribution of the average Zapping time, grouping the results by the total of sessions with only two moments regarding that average.

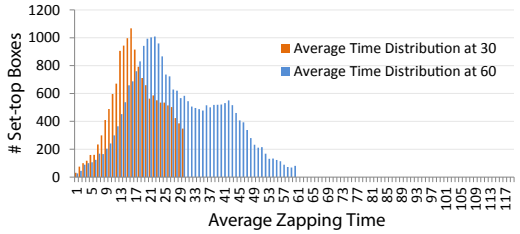
In order to better understand the influence of *minVisualizationTime* value, the results are presented with two adjacent instantiations of this parameter in same chart, in such a way we can grasp what changed among different parametrizations (Figure 4.6).

Analysing the distributions of average Zapping time per STB (Figures 4.6(a), 4.6(c) and 4.6(e)), both average and mode suffer slight increases with the growth of *minVisualizationTime*. One interesting aspect is the bimodal distribution behaviour for *minVisualizationTime* values from 60 seconds, seeming to show that there are two Zapping modes: *(i)* those with short moments, and *(ii)* others with longer ones.

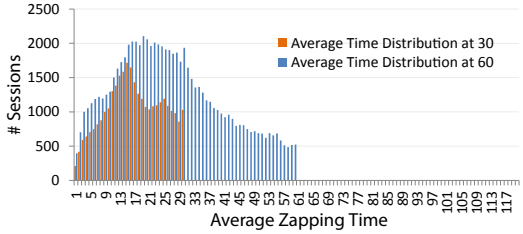
Regarding the distributions of average Zapping time per session (Figures 4.6(b), 4.6(d) and 4.6(f)), it seems to have a positively skewed normal distribution (asymmetric). From figure 4.7, we can see that the average Zapping time is relatively close to the one registered in previous distributions (Figures 4.6(a), 4.6(c) and 4.6(e)). On the other hand, 19 seconds appears to be the value most frequent along different instantiations (60, 90 and 120), being perhaps, the best value that define the average Zapping time, since we're talking about the sessions with most representativeness in the records (those with two Zapping Moments).

Daily zapping sessions?

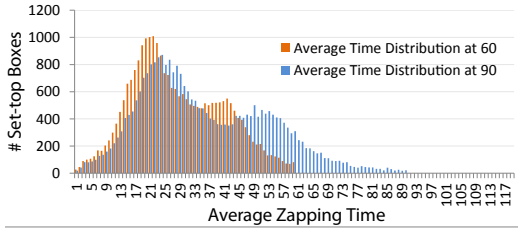
Each STB has a total of daily Zapping sessions. So, aggregating those totals over all STBs, we have a distribution of total daily Zapping sessions. This result is shown in Figure 4.8. This distribution is quite similar to the average Zapping Moments by session (Figure 4.3). Again, a lower *minVisualizationTime* value causes many moments to be considered as Visualization Moments (i.e., few Zapping Moments), yielding a low number of daily Zapping Sessions - 70% of STBs with *minVisualizationTime=30* have only one daily



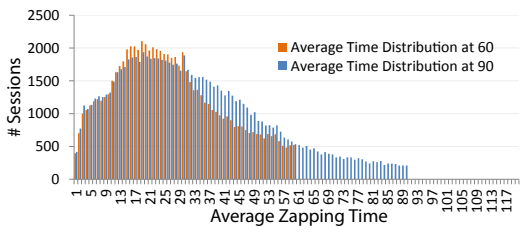
(a) Average zapping time distribution per *STB* with *minVisualizationTime* at 30 and 60



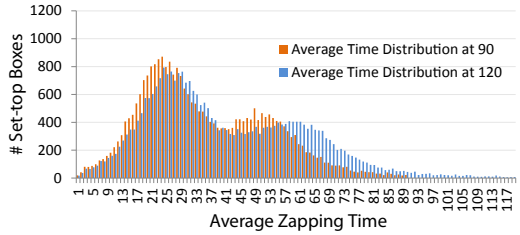
(b) Average zapping time distribution per session with two zapping moments with *minVisualizationTime* at 30 and 60



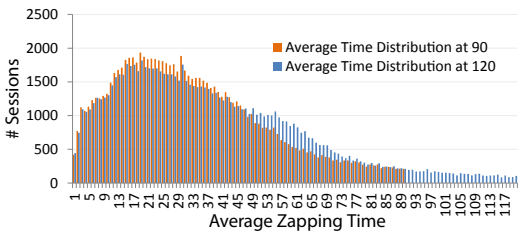
(c) Average zapping time distribution per *STB* with *minVisualizationTime* at 60 and 90



(d) Average zapping time distribution per session with two zapping moments with *minVisualizationTime* at 60 and 90



(e) Average zapping time distribution per *STB* with *minVisualizationTime* at 90 and 120



(f) Average zapping time distribution per session with two zapping moments with *minVisualizationTime* at 90 and 120

Figure 4.6: Average Zapping Time per *STB* against Average Zapping Time per session for different *minVisualizationTime* instantiations

minVisualizationTime	**Average Zapping Time**			
	Mean		**Mode**	
	STB[a]	**S2M**[b]	**STB**	**S2M**
30	17,51	16,68	16	15
60	28,37	26,41	22	19
90	35,64	32,97	24	19
120	41,05	37,92	24	19

[a] Average Zapping Time distribution per STB

[b] Average Zapping Time distribution per session with two zapping moments

Figure 4.7: Average Zapping Time Distribution Analysis

zapping session. On the other hand, a wider *minVisualizationTime* value allows more Zapping Moments, and consequently, a higher total of daily Zapping Sessions. In the same

way as has already happened in 4.3, this behaviour is caused by *minVisualizationTime* variation.

Mostly, there is a large proportion of only one daily Zapping Session. However, there are considerable records with more than one session per day (up to 5-6), although with a much lower occurrence rate. So, the Zapping behaviour is really insignificant in everyday life?

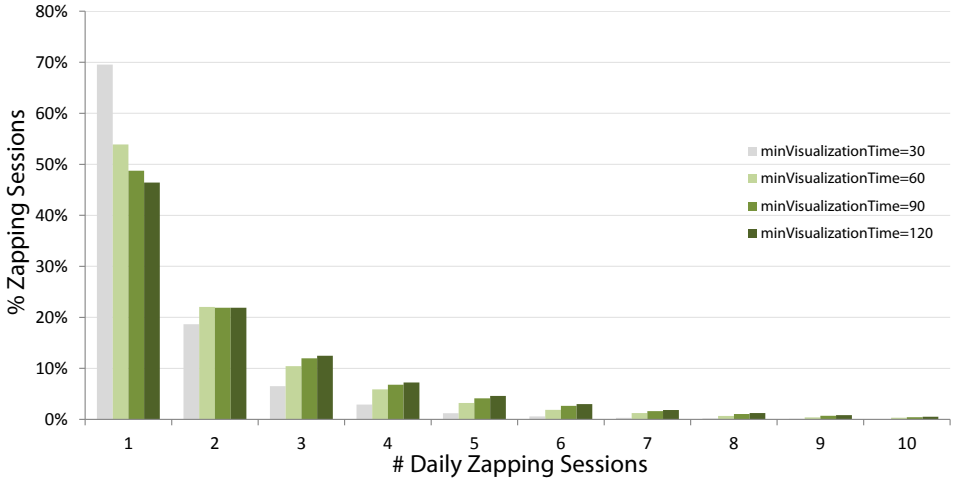
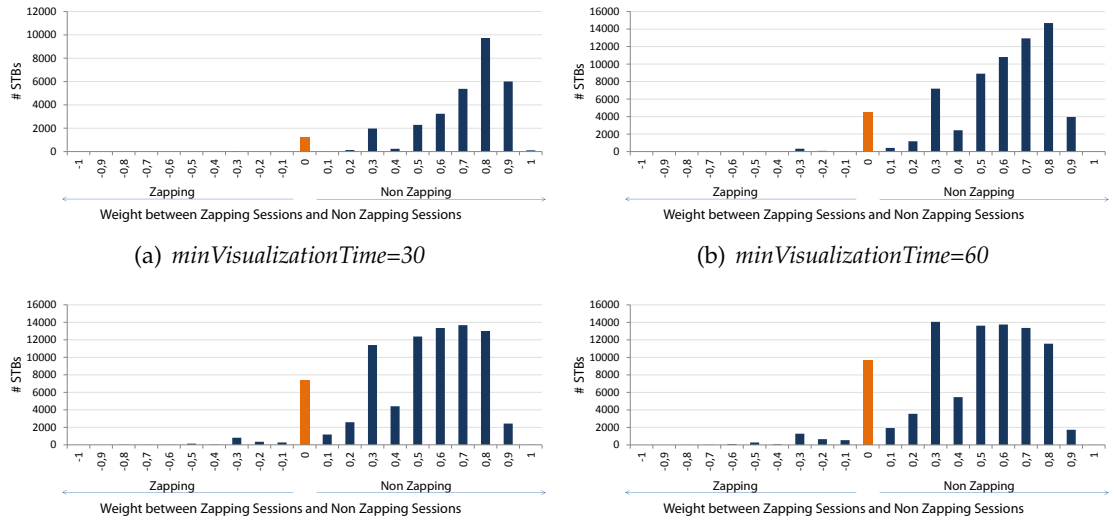


Figure 4.8: Daily Zapping Sessions Distribution

In order to understand if only one Zapping Session is indeed the most plausible behaviour, it is important to observe the weight between the number of Zapping Sessions versus Visualization Sessions for a given STB. So, for each STB we counted the daily number of Zapping Sessions (represented as ZS) and Visualization Sessions (represented as VS), expressing the difference between both in a range $weight \in [-1, .., 1]$. Considering the visualization ratio as: $VS_{ratio} = \frac{VS}{VS+ZS}$, $weight = VS_{ratio} - (1 - VS_{ratio})$ is the balance between VS and ZS, where: *(i)* $weight = -1$ indicates only Zapping Sessions; *(ii)* $weight = 1$ indicates only Visualization Sessions; and *(iii)* $weight = 0$ indicates that $VS = ZS$. In the figure 4.9 we can see that balancing, where there is a noticeable tendency for the number of daily Visualization Sessions be greater than the number of Zapping Sessions. However, there is also a large number of STBs with equal daily number of the two types of sessions, i.e., $weight = 0$ (orange bar in charts).

Moreover, if we take a look at the cumulative distribution of total daily Visualization Sessions (Figure 4.10(b)), more than 60% of STBs have five or less Visualization Sessions in a single day, in which only a single daily session is the most frequent (Figure 4.10(a)), whatever the *minVisualizationTime* value. We can also measure the distribution of the average number of contiguous moments by Visualization Session (see Figure 4.11(a)). In this case, the distribution presents a opposite behaviour than in 4.8, which is expectable since we are now dealing with Visualization Sessions. In fact, from that cumulative distribution (4.11(b)), more than 80% of records present three or less Visualization Moments by session, which denote that users have no large moments of contiguous Visualizations,

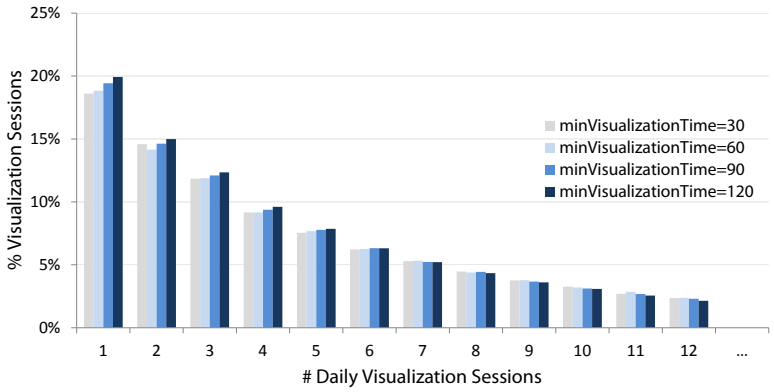


(a) *minVisualizationTime=30* (b) *minVisualizationTime=60*

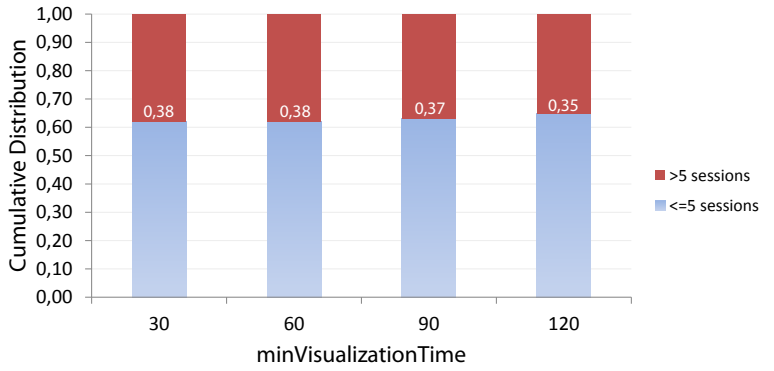
(c) *minVisualizationTime=90* (d) *minVisualizationTime=120*

Figure 4.9: Balance between Sessions type for different *minVisualizationTime* values

and a significant part of them tune to a different channel after the end of a given content.

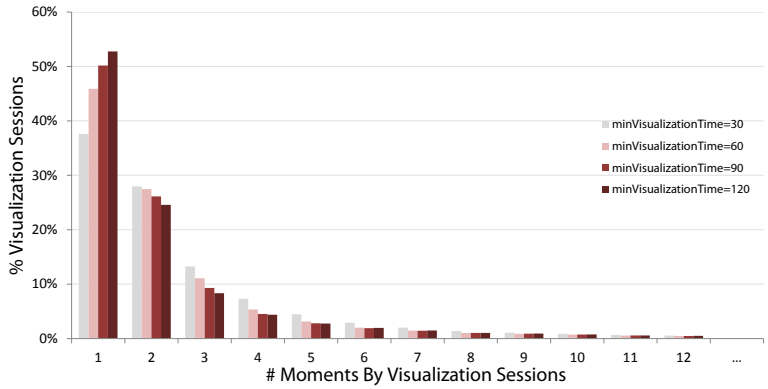


(a) Daily Non Zapping Sessions Distribution

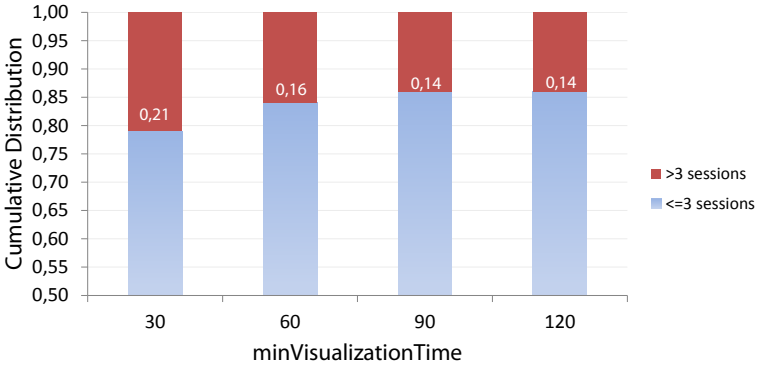


(b) Cumulative Distribution up to Five Non Zapping Sessions

Figure 4.10: Daily Non Zapping Distribution



(a) Average Daily Distribution of Non Zapping Moments by Session



(b) Cumulative Distribution up to Three Moments by Non Zapping Session

Figure 4.11: Average Daily Moments of Non Zapping by Session

Therefore, we can not underestimate the Zapping behaviour because its reduced number of daily sessions, since in general, people also have a very few Visualization Sessions along a television day, both in the number of sessions, as well as in contiguous moments per session, resulting in a not very intense daily activity. Thus, the Zapping actions during a day, makes this behaviour an integral part of television activity.

How is the zapping behaviour along different channels categories?

The reason for the existence of Zapping behaviour is due to the wide range of channels. However, people do not watch all the offered channels, fixing their attention on those that they most like. Thus, we present more detailed results, which focus on the influence that channels have on Zapping. In this section, the experimental procedure was done with only one instantiation of the parameter *minVisualizationTime* (=120), since it was only used to understand whether there were differences both in the number of moments, and number of sessions. As we have seen, the behaviour is similar to any of the tested values.

Both Zapping and Visualization Sessions consist in a sequence of moments. In turn, each moment is assigned to a particular channel, and we know the corresponding category of it. So, we can measure how Zapping and Visualization behave in terms of channels categories. Totalling each moment occurrence, we can analyse the distribution of moments by channel category. Also, for each moment, we know its duration, and therefore, we are able to measure the average Zapping time by channel category. Figure 4.12 presents for each channel category: *(i)* the average Zapping time; and *(ii)* the percentage of both Zapping and Visualization Moments, relating to that particular category.

Category	Average Zapping Time	% Moments	
		Zapping	Visualization
Adultos	57s	$0,5\%$	$0,1\%$
Desporto	43s	$8,2\%$	$4,0\%$
Documentários	41s	$5,7\%$	$3,6\%$
Filmes e séries	39s	$19,6\%$	$18,1\%$
Generalista	44s	$6,8\%$	$4,8\%$
Infantil	38s	$5,7\%$	$19,3\%$
Música	40s	$4,7\%$	$2,1\%$
Nacional	41s	$46,9\%$	$47,1\%$
Notícias	46s	$1,9\%$	$0,8\%$

Figure 4.12: Zapping behaviour by channel category

The average Zapping time is 43 seconds, and almost all categories have a similar mean value, with the exception of *Adultos* category (57 seconds). Concerning the total Zapping Moments, *Filmes e séries* and *Nacional* categories are the ones with most Zapping actions ($46,9\%$ and $19,6\%$ respectively). But when comparing the percentage of Visualization Moments (no Zapping) by category (Zapping and Visualization columns), the values are quite similar. Therefore, the Zapping rate is in agreement with the volume of Visualizations from its corresponding category. The only case with significant changes, is the *Infantil* category, where the percentage of Zapping is lower than Visualization (6% to 19%). This is explained by the fact that this category has a specific target audience (families with children) where it is expected to have a significant amount of views. The low relevance of Zapping in this channel category, symbolizes that people tune directly to these type of channels, and do not need to find something that interest them, because they already know what they want and choose it quickly and easily. But in general, the amount of Zapping is proportionally related with the amount of Visualization.

What is the correlation between visualization time and the number of zapping moments?

It becomes obvious that not all channels have the same rate of Visualization, and in some of them it is very low. So, for those cases, if a particular channel has a lot of Zapping actions, may reveal that it does not express interest enough to the users. Thus, for each channel, we calculated the total hours spent in Visualization and the total of

Zapping Moments for all days. Figure 4.13 shows the list of channels sorted by channel number line up, in terms of: *(i)* Visualization hours and Zapping Moments for each one; and *(ii)* average visualization hours and average Zapping Moments over all channels. This sorting allows to check if the channel position has influence both on the volume of Visualizations, and also on Zapping Moments.

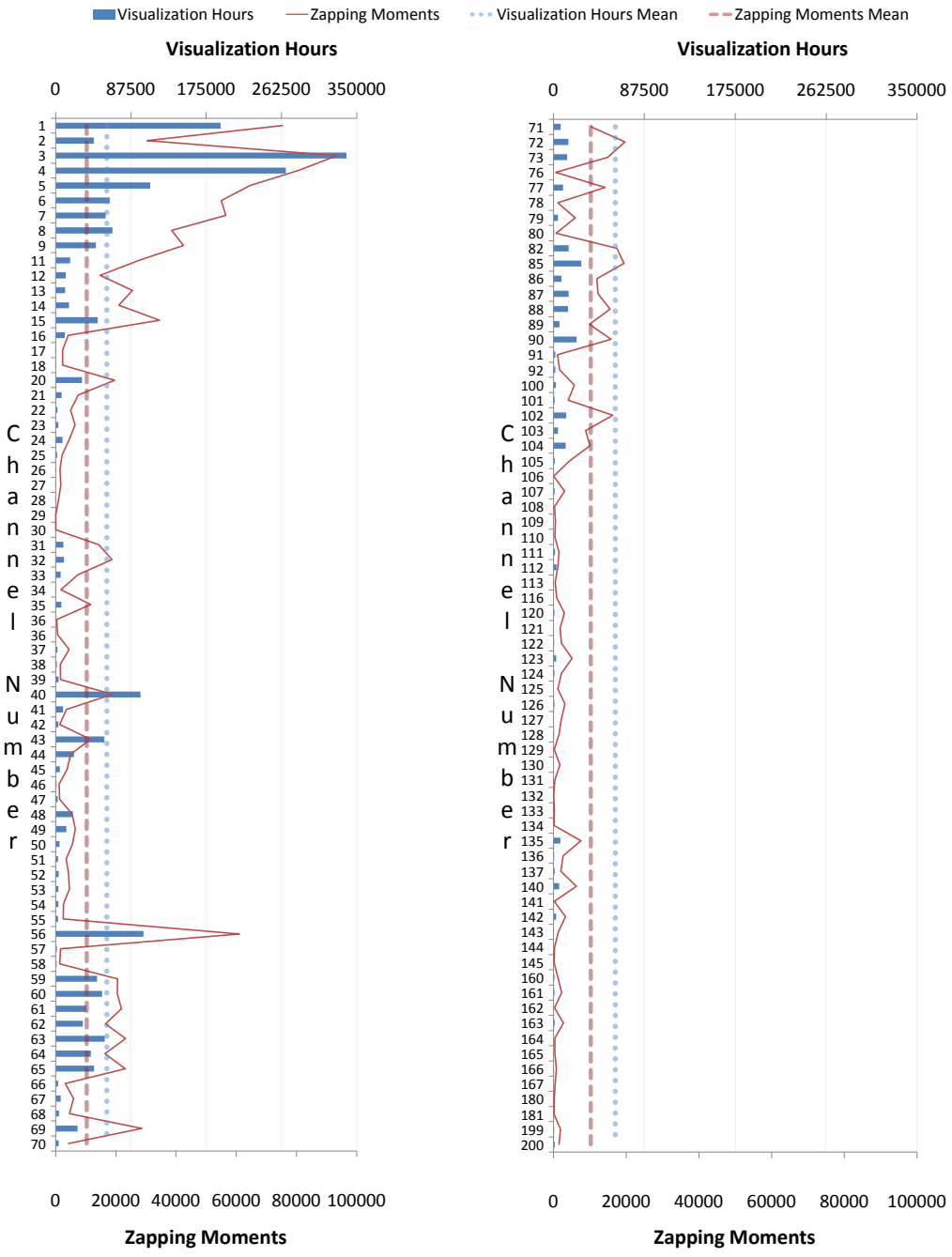


Figure 4.13: Channel-by-Channel Visualization Hours versus Zapping Moments

Generally, the volume of Zapping Moments is related with the volume of Visualization for a given channel, i.e., the larger the volume of Visualizations, the greater the number of Zapping Moments. Thus, we can not say that Zapping behaviour is independent of TV Visualization. What we have noticed is that people have a channel oriented profile, fixing their attention on a set of favourite channels, on which falls the whole Visualization activity, including Zapping.

This profile is notorious in Figure 4.13, where only a few channels express interest towards the others. On the contrary, from the 137 channels available, 94 ($\frac{94}{137} \approx 69\%$) have a total of Visualization hours 50% below from the mean value (about 17.000 hours). From these 94, 80 have a total of Zapping Moments 50% below from the mean value (about 10.200 moments). With this, we risk stating that about 60% ($\frac{80}{137} \approx 58\%$) of the offered television channels does not raise interest enough for most users.

In addition, remember that each Zapping Session has also two additive metrics indicating the kind of hop that was performed: sequential hop and random hop, counting how many of each occurred. So, we measured the weight between them, trying to understand what is the most that happens. For each Zapping Session, we counted the number of sequential (represented as SH) and random (represented as RH) hops, expressing the difference between both in a range $weigh \in [-1, \ldots, 1]$. Considering the random hops ratio as: $RH_{ratio} = \frac{RH}{RH+SH}$, $weight = RH_{ratio} - (1 - RH_{ratio})$ is the balance between RH and SH, where: *(i)* $weight = -1$ indicates only sequential hops; *(ii)* $weight = 1$ indicates only random hops; and *(iii)* $weight = 0$ indicates that $RH = SH$.

The results showed that 82% of the Zapping Sessions (at least, with two Zapping Moments) has only random hops, i.e., each channel tuned within a session is not adjacent to the previous, according to the channel line up. In particular, we also calculated the average weight between sequential and random hops, grouped by number of moments by session. In Figure 4.14 is depicted this balancing, where each line represents the average balance between sequential and random hops for all sessions with that number of moments. Whatever the Zapping Session length, the weight of random hops is very superior than the sequential ones.

This contradicts the existing idea of Zapping behaviour, in which people press only the next and previous buttons to channel tune. But after all, most people perform direct channel tuning, that in our view, is consistent with the results discussed earlier. Since the average number of moments per Zapping Session is low and there are very few channels that stand out in television preferences, it is understandable that the Zapping actions focus more on those few channels, in which people tune directly when they are searching for a specific channel or content that express enough interest to fix them.

4.1.3 Zapping Remarks

The Zapping is an expected behaviour during the watching television activity. As such, we proposed a model that defines what is Zapping, which could be instantiated and

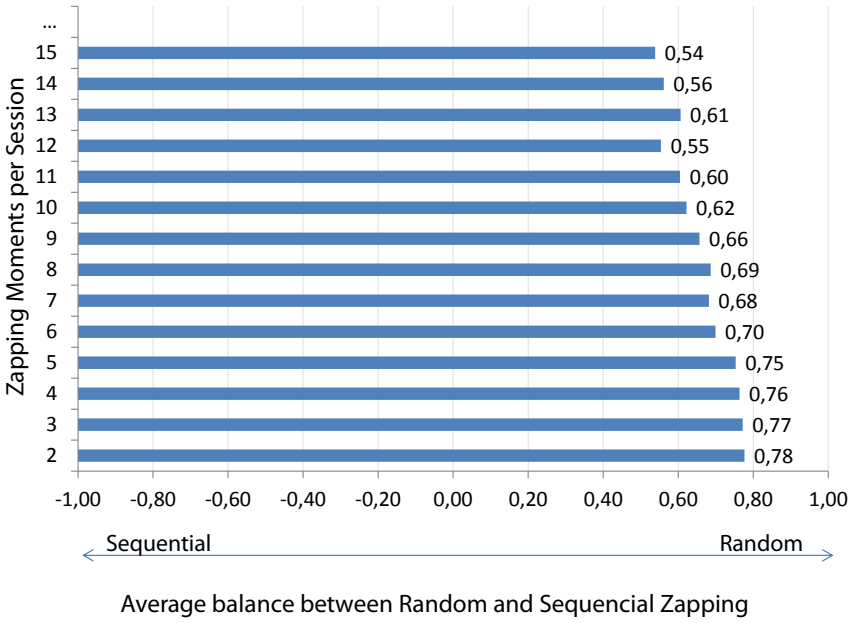


Figure 4.14: Average weight between zapping hops

tested. This model was based on the approach presented in section 3.3, where it was applied a new transformation phase, in order to model the existing information until then, to be consistent with our zapping formulation.

Although the data are limited in time span, inside what was possible, we focused our attention on the general analysis of the Zapping behaviour, in terms of the number of daily sessions, average time spent, relation with the Visualization activity, and Zapping behaviour along different channel categories.

Globally, we found that there are few daily sessions. However, the Zapping behaviour is proportional to the Visualization activity, wherein the longer a channel is watched, more Zapping actions it will have. In this line, stands out a channel oriented profile, where users focus their attention in a small group of channels, yielding, at least 60%, of the offered channels to have watching rates greatly reduced. This justifies the fact that Zapping is essentially direct, since users hop between non contiguous channels, focusing on the ones that interest them most. Regarding the channels categories, the overall behaviour is homogeneous, except for *Infantil* category, in which Zapping activity is much lower than the Visualization activity. Also, the *Adultos* category presents an average Zapping time superior than the general average Zapping time. These exceptions are due to the own characteristics of the channels involved in these categories.

The different parametrizations showed similar results both in number of sessions, moments per session and Zapping time mode for sessions with two Zapping Moments (which is the most prevalent). In [MC08], the authors evaluated some metrics related to channel selection problem. In their study, 60% of users hop to another channel within 10 seconds for sojourn times of channels switchings that occurred within one minute. But, in our experimental results, we observed that, mostly, users hop to another channel within

15-20 seconds, for different sojourn times considered (60, 90 and 120 seconds as depicted in Figure 4.6). They also totalled the average moments until users fix into a interesting channel, and came to the conclusion that users sample 4 channels on average. As we have already mentioned, our results shown that, mostly, users sample 2 channels before a fixing moment. So, in the context of this Portuguese carrier, users sample less channels until fix in a particular channel, but take longer until they decide. It is also should be noted that, in [MC08] there was no formulation for Zapping, neither the use of factors (such as K factor used) to distinguish better from Zapping Moments of Visualization Moments with durations very close to the threshold imposed, which may have an impact on the results.

4.2 DVR

Digital Video Recorder is one of the functionalities that contributed to change the client-content relationship, since it allows the users to schedule recordings of contents autonomously, so they can view them when they wish. In this sense, we studied this service in terms of its usage. We start with a brief description of the DVR features, and also with a formulation about DVR behaviour. Further, we detail along this section, several results obtained that help us to characterize and measure this IPTV functionality.

4.2.1 DVR Overview

Actual DVR service, provides the users with a unique ability to control TV contents. In particular, it allows that they record the contents that most prefer in order to view (and review) when they want. Apart from that, the users can manage their recordings like: cancel scheduled recordings or delete recordings already made. Nowadays there are three types of recordings available:

- **Manual** - recordings started with manual user intervention, particularly on live broadcast contents;
- **Dynamic one time** - recordings of an particular episode that was previously scheduled;
- **Dynamic recurring** - recordings of recurring episodes (a season) that were also previously scheduled. This recording mode also allows the users to specify some parameters such episode repetition, which allows them to record all broadcasts, or only the first edition of each episode.

As we have already explained in section 3.3.3, the proposed transformation process enabled us to recognize DVR activities (start and delete recordings), storing such information in a specific table (DVR table). Also recall that, visualization class activity includes DVR Playback visualizations. However, these singular activities, do not represent if the

user takes advantage of this service. More than understand if people are used to schedule DVR recordings, we need to check if they really watch them later. Or, we can try to figure out if users care about DVR managements. Following this, we can describe users' DVR behaviours like a decision tree (see Figure 4.15), regarding the three DVR actions we detected earlier: **R** - Record; **W** - Watch; and **D** - Delete. So, for each DVR recording action, a given user could watch it or not, and delete it or not. For instance, if we say: R=4, RW=3 and RWD=1, means that user recorded 4 contents, 3 of them were watched, and from those 3, he just deleted 1.

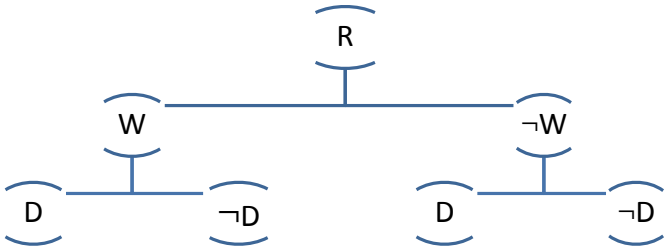


Figure 4.15: Users' Behaviours regarding DVR actions

So, in this section we explore the DVR usage combining these two activity classes in order to better understand the real usage behaviour. To achieve that, we propose a new transformation phase, in which, we can reduce even more the cardinality of data, and consequently, simplify it. Since we have, at most, three records relating to a particular DVR content: Record, Watch and Delete, it is possible to assemble them in one new record which contains all information needed. Besides further developing of our approach, this new transformation enables to recognize DVR behaviours, since for each DVR schedule, we know if they are watched or deleted later, which enhance further analyses. Figure 4.16 sums up this new transformation, regarding the DVR behaviour.

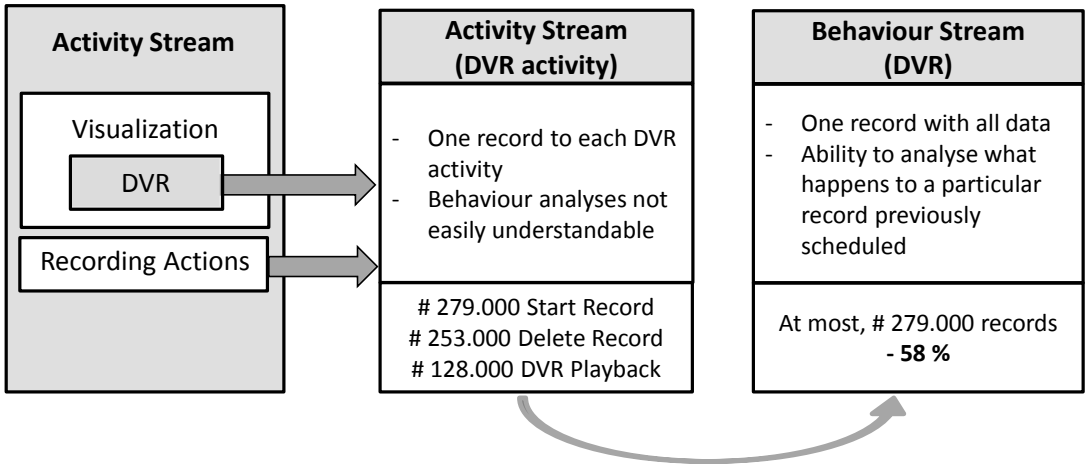


Figure 4.16: Summary of DVR Tranformation Process

An important aspect relating this DVR behaviour analyses is the weakness of current data. We only have access to few days of IPTV activity (no more than five contiguous days), which affect the evaluation, since we have no ability to measure in medium term, when people watch the recorded contents. Our attempt, focuses on a subsequent period in which we are able to detect the occurrence of these behaviours. In particular, we found evidences that a considerable number of DVR recordings are watched in the first 24 hours after they have been recorded. The next subsection details several results we obtained in our analyses.

4.2.2 DVR Evaluation

Since we already have all activities traced, relating to DVR behaviours for each STB, we intend to understand how people take advantage of this IPTV functionality. Our study contemplated some main issues:

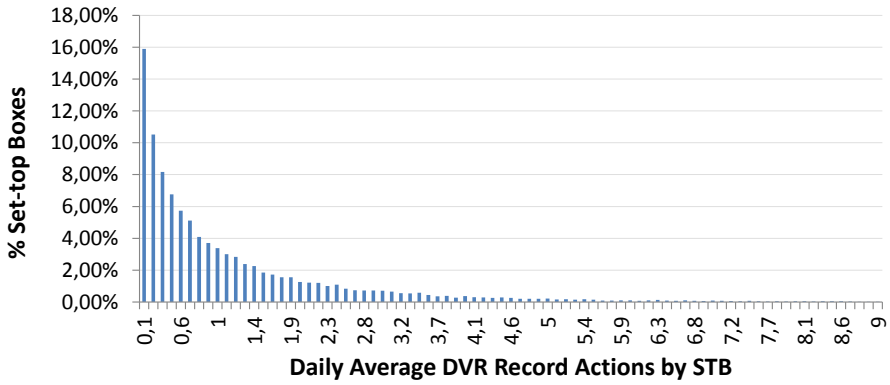
What is the volume of DVR recordings?

Of all about 52.000 STBs, the results shown that approximately 47% are active in start recording actions, which is a very reasonable percentage of this activity usage. However, such percentage may not mean that users make a rigorous use of this feature. Following this, it is possible to analyse the daily average number of recordings by STB, which results in the distribution depicted in Figure 4.17(a). These results are related to all nine days of activity we have in hands, where we counted the total of DVR recordings of a particular STB, and calculated the daily average. From this distribution we verify that the predominant daily number of start recordings by STB is just $\frac{1}{10}$, i.e., only one record in 9 days. Furthermore, note that the more than 60% of STBs (see Figure 4.17(b)) have no more than 1 daily recording.

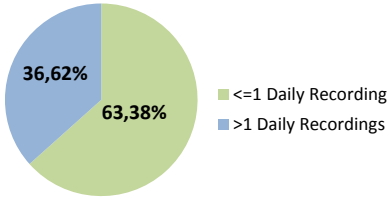
What is the most widely used type of recordings?

We explored the start DVR recordings distribution by its type (manual, dynamic: one time or recurring - see Figure 4.18), where we noticed that all registered recordings are dynamic (i.e., scheduled in advance by the user - 99%), where we can already conclude that the manual recordings are rarely used. Yet, the dynamic recurring recordings occur in about 80% of the cases. This result, led us to anticipate that the type of contents most recorded are the ones scheduled in several episodes.

Since a given STB may have more than one DVR recording, and even of different types (one time versus recurring), we noticed that about 50% of STBs have recordings from the two types. So, we counted the daily average number of: *(i)* **recurring recordings** (represented as R); and *(ii)* **one time recordings** (represented as O), expressing the difference between both in a range $weight \in [-1, .., 1]$. Considering the recurring ratio as: $R_{ratio} = \frac{R}{R+O}$, $weight = R_{ratio} - (1 - R_{ratio})$ is the balance between R and O, where: *(i)* $weight = -1$ indicates only one time recordings; *(ii)* $weight = 1$ indicates only recurring recordings; and *(iii)* $weight = 0$ indicates that the number of recurring recordings is equal to one recurring recordings.



(a) Percentual distribution of recordings number by *STB*



(b) Cumulative Distribution of Daily Average Recordings

Figure 4.17: Daily Average Number of DVR Recordings

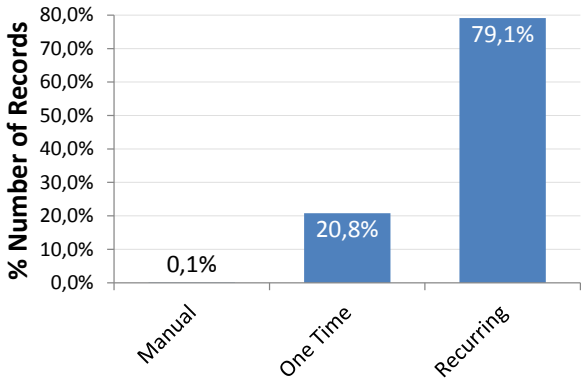


Figure 4.18: DVR Records Type distribution

Figure 4.19 shows the distribution of that balancing over all STBs. The horizontal axis contains the values between $[-1, \ldots, 1]$, and for each one there's a corresponding percentage of STBs with that balancing. It is noticeable, that users have a major tendency to schedule more recurring recordings against one time recordings, which is coherent with the result discussed previously (Figure 4.18).

From all recordings made, how many are watched?

Following our initial statement, users only take advantage of DVR service if they really watch the previously scheduled recordings. So, we try to figure out the percentage of those watched recordings. Of course, along these few days of activity, there is no ability to measure all of them, since people can only watch the recordings, for instance, at the

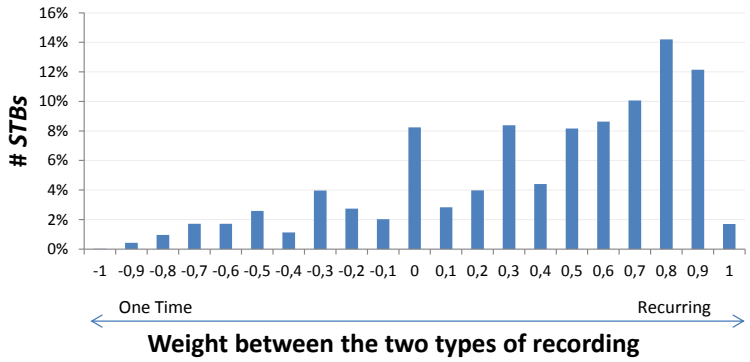
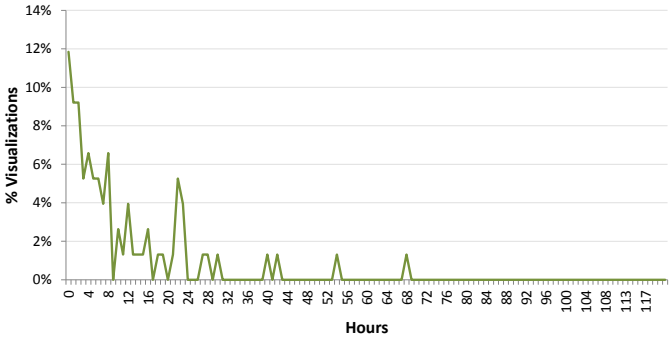


Figure 4.19: Balance between Dynamic Recordings Types

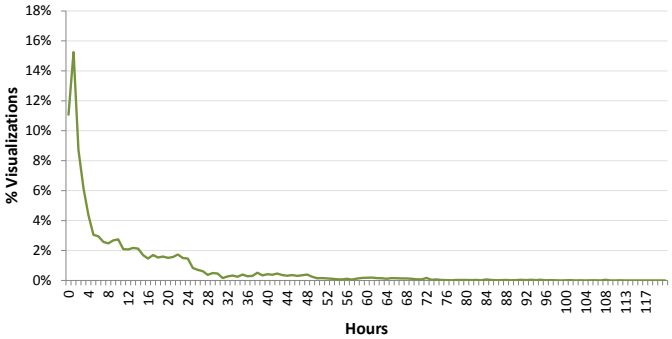
end of the next week. In this sense, it is very important to analyse the temporal difference between the recording time and watching time, and then realize how the percentage of watched recordings evolved throughout the days.

For each type of recording, we measured the number of hours needed to watch a particular DVR recording, whose result is depicted in Figure 4.20. The watching rate decreases day by day, in which most of the entire watching activity occurs in the first 24 hours, more specifically, within the first 4 hours after the recording time. Therefore, we calculated the percentage of watched records within 24 hours, since, most of them are watched in this period. Also, we are sure that the results we are taking in account are truthful, because a 24 hours period is perfectly controlled by us. Figure 4.21 shows the percentage of daily average watched recordings for each type (manual, one time and recurring). Only 8% of the recurring recordings are daily watched. But, in this kind of recording we believe that people tend to gather a certain number of episodes in order to watch them later, which in a few contiguous days is impossible to check. Although we analysed that most of the recordings were watched in the first 4 hours, we don't know the DVR behaviour in the remainder month, so that, a wider period of IPTV activity, may have reflex in a higher percentage of DVR visualizations.

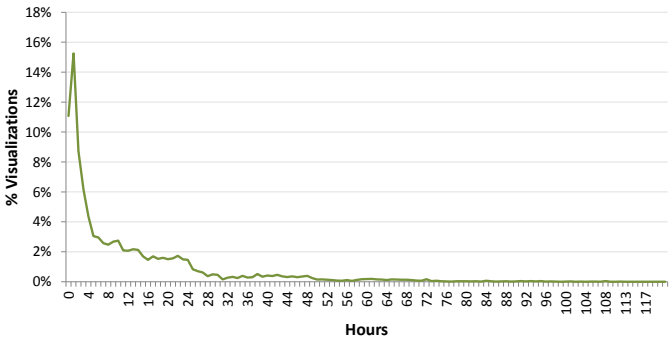
Finally, we can also measure the ratio between watched recordings against DVR recordings, per each STB within 24 hours. Figure 4.22 shows the distribution of that ratio overall STBs, in which most of users watch a low percentage of all recordings made during a particular day.



(a) Manual Recordings



(b) One Time Recordings



(c) Recurring Recordings

Figure 4.20: Distribution of Hours Needed to Watch a DVR Recording

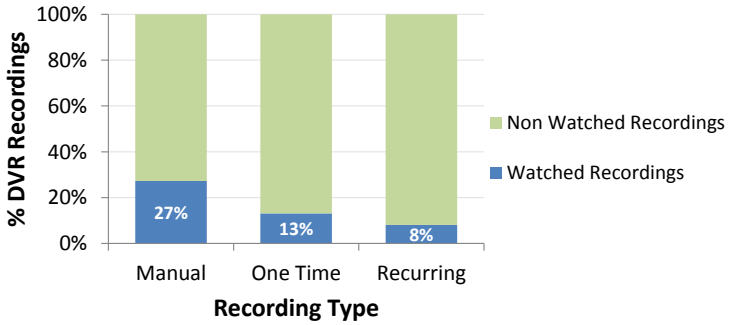


Figure 4.21: Total of visualized DVR Recordings

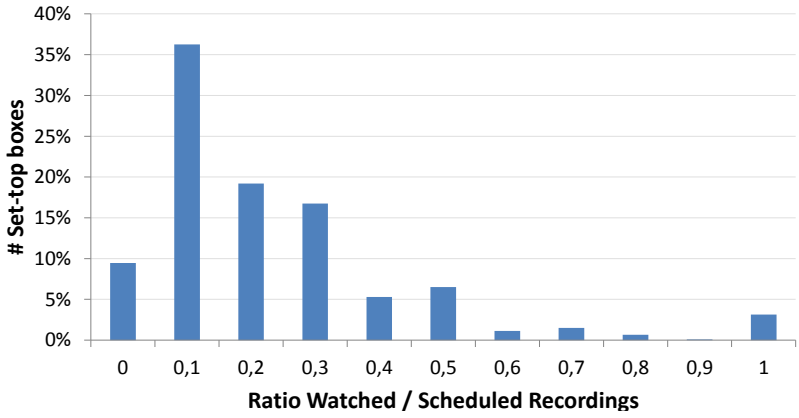


Figure 4.22: Ratio of DVR Visualizations

The users care about DVR recordings management?

Actual IPTV systems include contents self management, in order to auto erase recordings to free up space on the hard drive. Users could set this option, by locking the contents they want. However, this option is not the default one, being expected that most of users even know about it. So, from a total of 253.000 delete actions, about 32% are auto erase actions.

Yet, IPTV systems also allow their users to manage the contents that they schedule. Since we have already studied the DVR usage in terms of total recordings made and corresponding visualization, in this part we try to figure out if the users really care about DVR managements. Remember the users' DVR behaviours decision tree (Subsection 4.2.1, Figure 4.15), where we also included the Delete Actions. Now, we are interested to understand if users delete the recordings that they scheduled previously. Despite trying to know if users delete or not the recordings made, we can find two behaviours: *(i)* users that delete recordings after watch them (RWD); and *(ii)* users that delete recordings without even watch them (R¬WD). In general, from all 279.000 DVR recordings, almost 14% were deleted.

Beginning with the first point (RWD), and considering only the 14% of deleted recordings, about 35% of them, relate to recordings which were watched. Just analysing those 35%, there are considerable values: about 50% of dynamic recordings were deleted (52% for One Time; 47% for Recurring), and about 80% for manual ones (see Figure 4.23). In the same way as, we measured how long after the users watched the corresponding recordings, we can make a similar analysis over deleted recordings. For each type of recording, we measured the number of hours needed to delete a particular DVR recording, after its start watching time, whose result is depicted in Figure 4.24. Whatever the recording type, the majority of deleted recordings are performed until the first 2 hours.

Focusing only on those deleted records within 2 hours, we observed that about 20% of them occur in the first two minutes after the visualization beginning, i.e., the user recorded the entire program, but when he was decided to watch it, he just erased it after 2 minutes. This means that, a significant part of the recordings are nowhere to be watched

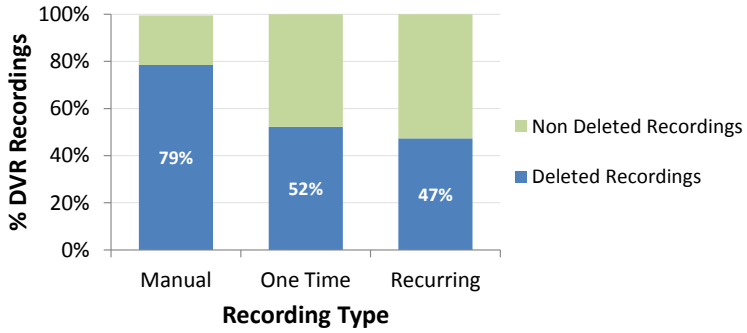
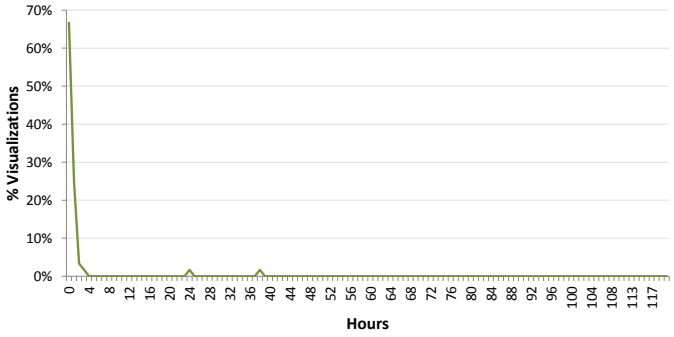
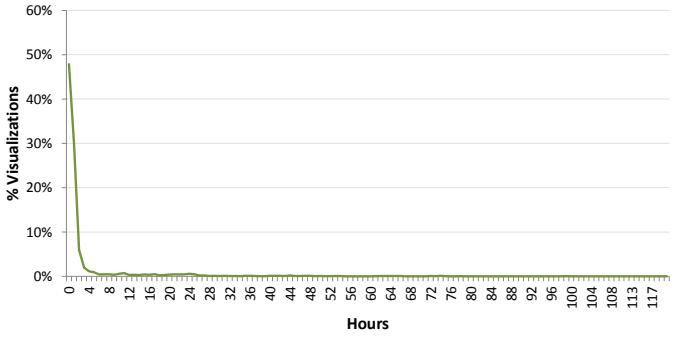


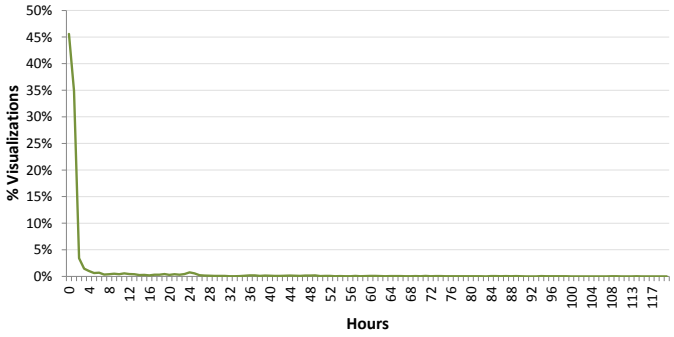
Figure 4.23: Total of deleted DVR Recordings after have been watched



(a) Manual Recordings



(b) One Time Recordings



(c) Recurring Recordings

Figure 4.24: Distribution of Hours Needed to Delete a Watched DVR Recording

in its entirety.

In respect to the second point, regarding all deleted recordings without ever been watched (R¬WD), about 65% of all deleted recordings, are in this situation. Figure 4.25 depicts the distribution of that ratio over all STBs in terms of recording type (Manual, One Time and Recurring), where we note that Manual and One Time recordings present a significant percentage of recordings with ratio=1, i.e., every recordings were deleted without ever been watched. In general, if we consider the cumulative distribution of this ratio (up to 0,5), there is a meaningful recordings that are deleted, i.e., there are a significant part of users, which delete up to half of the recordings made. That is representative, since it is always a volume of recordings that are ultimately deleted without being watched.

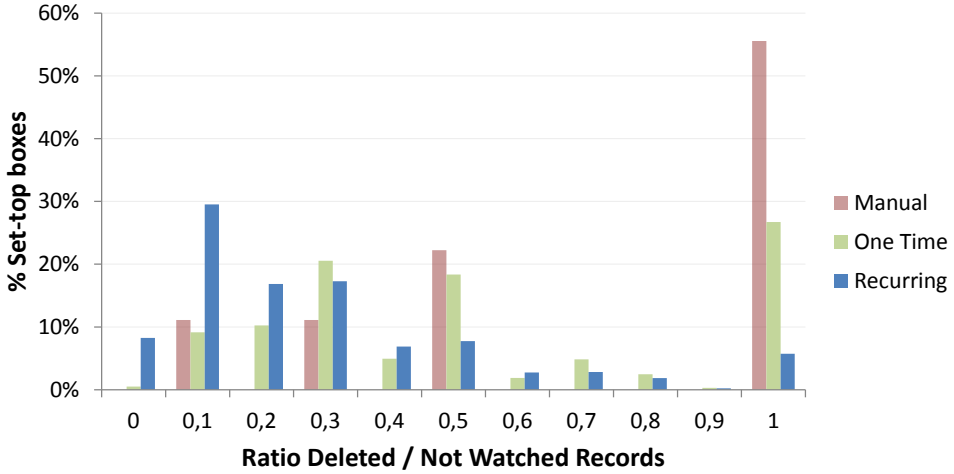
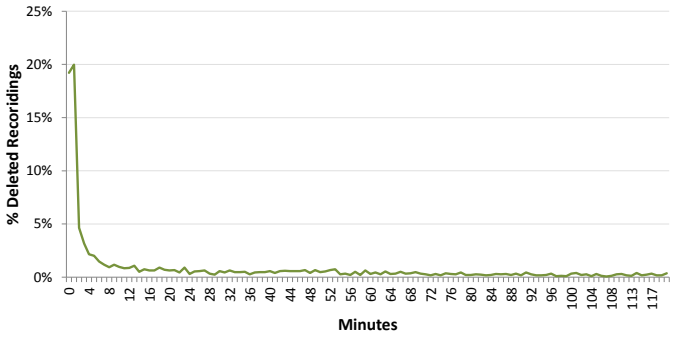


Figure 4.25: Ratio of Deleted Records that have never been watched

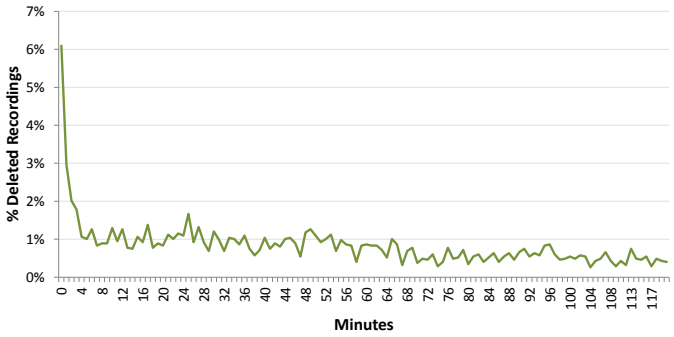
Finally, if we analyse the temporal difference between recording moment and erasing moment (for dynamic ones, since manual are very rarely), we observed that in One Time recordings (Figure 4.26(a)), there is a reasonable percentage of them that are deleted mainly within the first 2 minutes (40%). In recurring ones, a similar behaviour exists, but with a lower percentage, about 10% (Figure 4.26(b)). Users mistakes may be the reason for these values, since they schedule a particular STB recording (one time or recurring), erasing it after a very short period.

4.2.3 DVR Remarks

The DVR service is one of the most attractive feature in IPTV offer, since it allows users to schedule the contents they most like, contributing to the change between client and content. Thence, came our motivation to analyse how people really take advantage of this service. Following our approach, which transforms a set of click actions to a meaningful activity actions, we were able to recognize some important user's actions regarding DVR recordings, like: **record**, **watch** and **delete**. In this line, we proposed a formulation that, more than understand those activities, we can represent the users' behaviour regarding



(a) One Time Recordings



(b) Recurring Recordings

Figure 4.26: Distribution of Minutes Needed to Delete a Non Watched DVR Recording

this functionality. Namely, figure out how users benefit from their DVR recordings: if they watch them; how long after they watch the recordings; if they care about DVR managements, and others. To achieve that, we applied a new transformation phase, which allowed us to reduce the several records into a single one, representing such behaviours in a clearer and simpler way.

However, the biggest limitation we had to face was the weakness of data, in terms of time span, which did not enable to study those DVR behaviours with proper accuracy, since we didn't know how long after the users watch or manage the scheduled recordings. Therefore, we checked that most of the DVR actions, happen at the first 24 hours after the recording operation had been performed. So, we focused our analyses over this period, because we could measure what happen during that period, allowing to show some truthful results.

Generally, most of the users do not perform more than daily recording. The manual recordings are rarely used $(0,1\%)$, where 80% of them are dynamic recurring recordings and the remaining 20% are dynamic one time recordings. Along a television day, about 10% of the scheduled recordings are indeed watched. Yet, about 50% of DVR recordings are erased without ever been watched, which is a significant result.

5

Conclusion

In this chapter, we present the final conclusions regarding the work done so far. First, we start with a summary of the approach taken and how the proposed framework was used to achieve our main goals. Then we make an evaluation of it, and how we compare it with other proposals. Also, we refer some aspects related to data limitations, experimental procedure results and finally, some guidelines to future work.

5.1 Summary

The main goal of this work, was to propose a framework able to deal directly with real IPTV data, in order to recognize IPTV users' behaviours. Remember that, original data presents a very low granularity, which it is complex and hard to interpret. So, our approach focus on a reduction strategy, which enables to compress the original data and enhance the information quality, resulting in a transformation of clicking actions into a more conceptual and representative level of the running activities in a Set-top Box. This transformation process aims to be iterative, where we can apply further reductions until reach a desired level of detail. In scope of this work, this iterative process started with a Click Stream to Activity Stream transformation, with the purpose to detect the main activities that occur in a given STB. At this stage, we achieved a considerable cardinality reduction and also, the representation became more simplified and valuable. However, more than those users' actions, in order to understand how they take advantage of the services at them disposal, we need to aggregate a set of activities to represent them as a particular behaviour. In this line, we applied a new transformation phase to agglomerate singular activity records into one single record regarding that behaviour. As a result of

that transformation, we explored two of the most expected users' behaviours: *(i)* **Zapping** (regarding to live broadcasting) and *(ii)* **DVR usage**, on which we made several statistical analyses to study how users exploit the features available on the IPTV service offering.

As we have large volumes of complex and scattered daily data, the suitable way to handle this information was to interpret it in a Complex Event Processing concept, since we can process such data, by applying filters, correlations, aggregates or patterns to extract meaningful information. The CEP tool chosen was Esper, which allowed us to build the iterative transformation process, previously described. The use of this tool is simple, where we just need to map the sequence of events that represent a particular activity or behaviour, using such processing operations to detect those occurrences in a data stream. The matching sequences are then dumped to a database, where each record represents the desire activity or behaviour at the accurate analysis level.

5.2 Evaluation

Despite the increasing interest in IPTV data, the analysed works (Section 2.1.2) focus, essentially in the broadcasting television context, where there is no approach made to determine profiles based on users' behaviours, in terms of the used features and services offering, as was pointed out in our motivation. The Audiometry area, deals with similar matters but different from our context, since it is used as audiences measurement, to manage television content and to schedule TV listings for media advertising purposes. Currently, the Audiometry tools are still evolving to a new digital era, and hence they are are not enough to address the new coming challenges.

So, we are proposing a different approach from what we found in literature, which was thought to embrace the existing need in measure how users take advantage of the vast service offering. Thence, we submitted our approach with an assessment to real IPTV data, which resulted in two application areas regarding users' behaviours: Zapping and DVR usage. For each of those behaviours, we proposed a formulation able to represent and analyse their usage.

However, the biggest limitation we had to face was the weakness of data. Only few contiguous days, do not enable to measure such behaviours in an accurate way, and consequently, to study how they change over the time. So, we limited our scope of analyses, just testing those behaviours in an activity controlled time fraction. This fact, imposed some limitations in terms of possible further analyses, such as data mining techniques. Also, all the activities detected resulted from a survey made, following a top-down approach in order to make a collection of feasible activities. With respect to other existing features such as interactive applications (widgets web) or STB Services Management, they were not considered because actually those events are not captured. But, as long as we have such features detectable, they can be integrated, just following the same principle made for the features studied along this work.

Based on the data available, we proposed models that define real users' behaviours, which were instantiated, yielding to significant results. More than getting results, those models allowed to give relevance to the proposed transformation process, which together led to achieve the objectives outlined.

5.3 Future Work

We know that this work does not end here. So far, we idealized and built the supporting bases aiming the use of this data, in order to understand how the IPTV service is exploited by its users.

Our approach has the potential to move forward, namely with the enrichment of activities liable to be analysed, and also with a wider period of IPTV activity. That is possible, since the approach designed allows such extensibility, being enough a rigorous definition of the activities running on a STB, and map them according the CEP tool standards. Also, the introduction of more data, may enhance the analyses about users' behaviours, by the application of clustering techniques, in order to detect users' patterns, regarding the offered features.

Although the actual IPTV systems dump the data at the end of the day, an interesting point to explore, will be the use of this framework at real time, where the same activities could be detectable as soon as they occur. That, brings new challenges in terms of analyses, for instance, in forecasting modelling or to correlate users with similar behaviours in a particular moment.

Last but not least, would be the chance to apply such framework on a particular carrier to analyse the consumption rate of its services. As we worked with real IPTV data, from a real carrier, and achieved some interesting results, we are trying to show the feasibility on usage of such data, which is more accurate, truthful and reliable than the actual inquiries made to users.

Bibliography

[ANA09] ANACOM. Índice nacional de satisfação do cliente, 2009. http://www.anacom.pt/streaming/Relatorio_satisfacao_cliente_comunicacoes2008.pdf?contentId=969967&field=ATTACHED_FILE.

[ANA11] ANACOM. Situação das comunicações 2011, 2011. http://www.anacom.pt/streaming/situacaocomunicacoes2011072012.pdf?contentId=1127288&field=ATTACHED_FILE.

[BBD+02] Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom. Models and issues in data stream systems. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, PODS '02, pages 1–16, New York, NY, USA, 2002. ACM.

[CHD00] Qiming Chen, Meichun Hsu, and Umesh Dayal. A Data-Warehouse/OLAP framework for scalable telecommunication tandem traffic analysis. In *Data Engineering, International Conference on*, volume 0, page 201, Los Alamitos, CA, USA, 2000. IEEE Computer Society.

[Dat06] Nuno Datia. Aplicação de técnicas de apoio à decisão a dados de audimetria. Master's thesis, FCT-UNL, 2006.

[DMPCP05] N. Datia, J. Moura-Pires, M. Cardoso, and H. Pita. Temporal patterns of tv watching for portuguese viewers. In *Artificial intelligence, 2005. epia 2005. portuguese conference on*, pages 151 –158, 2005.

[DPL02] Nuno Datia, Helder Pita, and Carlos Leandro. Análise da dados sobre audimetria da televisão em portugal. In *CCTE 2002 - Conferência Científica e Tecnológica em Engenharia*, 2002.

[DSV+08] Koustuv Dasgupta, Rahul Singh, Balaji Viswanathan, Dipanjan Chakraborty, Sougata Mukherjea, Amit A. Nanavati, and Anupam

Joshi. Social ties and their relevance to churn in mobile telecom networks. page 668. ACM Press, 2008.

[DW08] T. Dasu and G. M. Weiss. Mining data streams. In *Encyclopedia of Data Warehousing and Mining, second edition*, pages 1248–1256. J. Wang Ed. Kluwer Academic Publishers, 2008.

[esp] Esper - complex event processing. http://esper.codehaus.org/.

[FPsS96] Usama Fayyad, Gregory Piatetsky-shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, 17:37–54, 1996.

[GO03] Lukasz Golab and M. Tamer Özsu. Issues in data stream management. *SIGMOD Rec.*, 32:5–14, June 2003.

[JMR10] Andrew Jones, Keith Mitchell, and Nicholas J.P. Race. TriggerTV: exploiting social user journeys within an interactive TV system. In *Proceedings of the 20th international workshop on Network and operating systems support for digital audio and video*, NOSSDAV '10, page 51–56, New York, NY, USA, 2010. ACM.

[MC08] Pablo Rodriguez Meeyoung Cha, Krishna P. Gummadi. Channel selection problem in live iptv systems. 2008.

[MGS+09] Ajay Anil Mahimkar, Zihui Ge, Aman Shaikh, Jia Wang, Jennifer Yates, Yin Zhang, and Qi Zhao. Towards automated performance diagnosis in a large IPTV network. In *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, SIGCOMM '09, page 231–242, New York, NY, USA, 2009. ACM.

[PXQ+11] Jiansu Pu, Panpan Xu, Huamin Qu, Weiwei Cui, Siyuan Liu, and Lionel Ni. Visual analysis of people's mobility pattern from mobile phone data. pages 1–10. ACM Press, 2011.

[QGL+09a] Tongqing Qiu, Zihui Ge, Seungjoon Lee, Jia Wang, Jun Xu, and Qi Zhao. Modeling user activities in a large IPTV system. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, IMC '09, page 430–441, New York, NY, USA, 2009. ACM.

[QGL+09b] Tongqing Qiu, Zihui Ge, Seungjoon Lee, Jia Wang, Qi Zhao, and Jun Xu. Modeling channel popularity dynamics in a large IPTV system. In *Proceedings of the eleventh international joint conference on Measurement and modeling of computer systems*, SIGMETRICS '09, page 275–286, New York, NY, USA, 2009. ACM.

[SF11] Víctor Soto and Enrique Frías-Martínez. Automated land use identification using cell-phone records. page 17. ACM Press, 2011.

[Wei] Gary M. Weiss. Data mining in telecommunications. In Oded Maimon and Lior Rokach, editors, *Data Mining and Knowledge Discovery Handbook*, pages 1189–1201. Springer-Verlag, New York.

[WKL+05] Li Wei, Nitin Kumar, Venkata Lolla, Eamonn Keogh, Stefano Lonardi, and Chotirat Ann Ratanamahatana. Assumption-free anomaly detection in time series. In *Proceedings of the 17th International Conference on Scientific and Statistical Database Management (SSDBM'05)*, pages 237–242, 2005.

Implementation Details

A.1 Event Type

This section presents a sample code of the Event Type Object that represent the IPTV events: IPTV Record class.

A.1.1 IPTV Record

Listing A.1: IPTV Record

```
/**
 * The Java Object that maps the entity Record
 */
public class Record {

  private String box_ID;
  private Timestamp time_ID;
  private Integer event_ID;
  private String channel_ID;
  private String content_ID;
  //others as depicted at Figure 3.1

  //constructor

  public String getBox_ID() {
    return box_ID;
  }
  public void setBox_ID(String box_ID) {
    this.box_ID = box_ID;
```

```
  }
  //others getters and setters
}
```

A.2 Statements

This section presents a sample code of the Statements declared for each activity identified in section 3.2.

A.2.1 Live Visualization

Listing A.2: Live Visualization

```
/**
 * The Statement that represents the Live Visualization activity.
 */
public class LiveVisualizationStmt{
  private EPStatement statement;

    public LiveVisualizationStmt(EPAdministrator epAdmin){
      String stmt = "select␣a.box_ID␣as␣box_ID,␣a.time_ID␣as␣startTime,"+
        "a.content_ID␣as␣channel_ID,␣a.channel_nbr␣as␣channel_nbr,␣"+
        "b␣as␣contents,␣e.time_ID␣as␣endTime,␣e.event_ID␣as␣stopEvent,"+
        "e.channel_nbr␣as␣switchTo␣" +
      "from␣pattern␣[every␣a=Record(event_ID=100␣and␣channel_nbr!=201␣"+
      "and␣channel_nbr!=202)" +
      "->␣[1:]␣b=Record(box_ID=a.box_ID␣and␣event_ID=114␣"+
      "and␣channel_ID=a.content_ID)␣"+
      "until␣e=Record(box_ID=a.box_ID␣"+
      "and␣(event_ID=100␣or␣event_ID=101␣or␣event_ID=107))]";

      statement = (epAdmin.createEPL(stmt);
  }
  //...
}
```

A.2.2 DVR Visualization

Listing A.3: DVR Visualization

```
/**
 * The Statement that represents the DVR Visualization activity.
 */
public class DVRVisualizationStmt{
  private EPStatement statement;

    public DVRVisualizationStmt(EPAdministrator epAdmin){
      String stmt = "select␣b.box_ID␣as␣box_ID,␣b.time_ID␣as␣startTime,␣"+
```

```
      "b.channel_ID␣as␣channel_ID,␣a.channel_nbr␣as␣channel_nbr,␣"+
      "b.content_ID␣as␣content_ID,␣b.duration␣as␣contentDuration,␣"+
      "b.view_mode␣as␣view_mode,␣e.time_ID␣as␣endTime␣" +
      "from␣pattern␣[every␣a=Record(event_ID=100␣and␣channel_nbr=202␣"+
      "and␣service_type='DVR')␣" +
      "->␣b=Record(box_ID=a.box_ID␣and␣event_ID=114␣and␣service_type='DVR'␣"+
      "and␣util.TimestampDifference.getDifference(time_ID.getTime(),␣"+
      "a.time_ID.getTime())<=3)␣" +
      "->␣e=Record(box_ID=a.box_ID␣and␣(event_ID=100␣or␣event_ID=101␣"+
      "or␣event_ID=107␣or␣event_ID=114␣or␣event_ID=119))␣"+
      "where␣timer:within(4␣hours)];"

      statement = (epAdmin.createEPL(stmt);
  }
  //...
}
```

A.2.3 VOD Visualization

Listing A.4: VOD Visualization

```
/**
 * The Statement that represents the VOD Visualization activity.
 */
public class VODVisualizationStmt{
  private EPStatement statement;

    public VODVisualizationStmt(EPAdministrator epAdmin){
    String stmt = "select␣a.box_ID␣as␣box_ID,␣a.time_ID␣as␣startTime,␣"+
      "a.content_ID␣as␣channel_ID,␣a.channel_nbr␣as␣channel_nbr,␣"+
      "b.content_ID␣as␣content_ID,␣b.view_mode␣as␣view_mode,␣"+
      "b.duration␣as␣contentDuration,␣e.time_ID␣as␣endTime␣" +
      "from␣pattern␣[every␣a=Record(event_ID=100␣and␣channel_nbr=201␣"+
      "and␣service_type='VOD'␣and␣view_mode='FULLSCREEN_PRIMARY')␣" +
      "->␣b=Record(box_ID=a.box_ID␣and␣event_ID=114␣and␣service_type='VOD'␣"+
      "and␣util.TimestampDifference.getDifference(time_ID.getTime(),␣"+
      "a.time_ID.getTime())<=3)␣" +
      "->␣e=Record(box_ID=a.box_ID␣and␣(event_ID=100␣or␣event_ID=101␣"+
      "or␣event_ID=107␣or␣event_ID=114))];"

      statement = (epAdmin.createEPL(stmt);
  }
  //...
}
```

A.2.4 DVR Start Operation

Listing A.5: DVR Start

```
/**
 * The Statement that represents the DVR Start activity.
 */
public class DVRStartStmt{
  private EPStatement statement;

    public DVRStartStmt(EPAdministrator epAdmin)
    String stmt = "select␣a.box_ID␣as␣box_ID,␣a.time_ID␣as␣time,␣"+
      "a.content_ID␣as␣content,␣a.duration␣as␣duration,␣"+
      "a.dynamic␣as␣dynamic,␣a.recurring␣as␣recurring␣" +
      "from␣pattern␣[every␣a=Record(event_ID=115)]";

      statement = (epAdmin.createEPL(stmt);
  }
  //...
}
```

A.2.5 DVR Delete Operation

Listing A.6: DVR Delete

```
/**
 * The Statement that represents the DVR Delete activity.
 */
public DVRDeleteStmt(EPAdministrator epAdmin){
      private EPStatement statement;

      String stmt = "select␣a.box_ID␣as␣box_ID,␣a.time_ID␣as␣time,␣"+
      "a.content_ID␣as␣content,␣a.manual_deletion␣as␣manual_deletion␣" +
      "from␣pattern␣[every␣a=Record(event_ID=119)]";

      statements.add(epAdmin.createEPL(stmt));
  }
  //...
}
```

A.2.6 VOD Operations

Listing A.7: VOD Operations

```
/**
 * The Statement that represents the VOD Operations activities.
 */
public class VODOperationsStmt{
    private List<EPStatement> statements;

    public VODOperationsStmt(EPAdministrator epAdmin){
        statements = new LinkedList<EPStatement>();
      //WATCH TRAILER
      String stmt = "select␣a.box_ID␣as␣box_ID,␣a.time_ID␣as␣moment,␣"+
      "b.content_ID␣as␣content_ID,␣c.content_ID␣as␣type␣" +
      "from␣pattern␣[every␣a=Record(event_ID=100␣and␣channel_nbr=201␣"+
      "and␣view_mode='FULLSCREEN_SECONDARY')␣" +
      "->␣c=Record(box_ID=a.box_ID␣and␣event_ID=107␣"+
      "and␣content_ID='VER␣TRAILER'␣"+
      "and␣util.TimestampDifference.getDifference("+
        "time_ID.getTime(),a.time_ID.getTime())<=5)" +
      "->␣b=Record(box_ID=a.box_ID␣and␣event_ID=114␣"+
      "and␣util.TimestampDifference.getDifference("+
        "time_ID.getTime(),a.time_ID.getTime())<=5)]";

      statements.add(epAdmin.createEPL(stmt));

      stmt = "select␣a.box_ID␣as␣box_ID,␣a.time_ID␣as␣moment,␣"+
      "c.content_ID␣as␣content_ID,␣a.content_ID␣as␣type␣" +
      "from␣pattern␣["+
      "every␣a=Record(event_ID=107␣and␣content_ID='VER␣TRAILER')␣" +
      "->␣b=Record(box_ID=a.box_ID␣and␣event_ID=100␣and␣channel_nbr=201␣and␣"+
      "view_mode='FULLSCREEN_SECONDARY'␣"+
      "and␣util.TimestampDifference.getDifference("+
        "time_ID.getTime(),a.time_ID.getTime())<=5)␣" +
      "->␣c=Record(box_ID=a.box_ID␣and␣event_ID=114␣"+
      "and␣util.TimestampDifference.getDifference("+
        "time_ID.getTime(),a.time_ID.getTime())<=3)]"

      statements.add(epAdmin.createEPL(stmt));

      //RENT
      stmt = "select␣a.box_ID␣as␣box_ID,␣a.time_ID␣as␣moment,␣"+
      "a.content_ID␣as␣content_ID,␣\"PURCHASE\"␣as␣type␣" +
      "from␣pattern␣[every␣a=Record(event_ID=102)]";

    statements.add(epAdmin.createEPL(stmt));
  }
  //...
}
```