



Escola d'Enginyeria de Telecomunicació i
Aeroespacial de Castellet

UNIVERSITAT POLITÈCNICA DE CATALUNYA

MASTER THESIS

TITLE: Business Intelligence module with optimized access for mobile devices

MASTER DEGREE: Master in Science in Telecommunication Engineering & Management

AUTHOR: Ignacio Fernández Hernández

DIRECTOR: José Manuel Yúfera Gómez

SUPERVISOR: Jordi Alborch

DATE: September, 25 2012

TITLE: Business Intelligence module with optimized access for mobile devices

AUTHOR: Ignacio Fernández Hernández

DIRECTOR: José Manuel Yúfera Gómez

SUPERVISOR: Jordi Alborch

DATE: September, 25 2012

OVERVIEW

Throughout the years, decisions making in business has been linked to the quantity and quality of information available to the executive boards to carry out their objectives.

Technology advances have let new strategies and tools so this task, so crucial in business, can be carried out more accurately and efficiently. The Business Intelligence implementation on the company is one of these mechanisms.

The main goal of this final degree project is the Business Intelligence system implementation, called BI MODULE, on Onebox, a company specialized on tickets sales industry. Overall, this module will provide the necessary information related to sales made by customers of the company so the provided information will be useful for them and for the company.

Moreover, another goal has been defined: the system optimization to be consulted through via mobile devices such as smartphones or tablets.

Index

CHAPTER 1. ONEBOX. TICKETING SERVICES	10
CHAPTER 2. BUSINESS INTELLIGENCE	12
2.1 DESCRIPTION. BASIC CONCEPTS	12
2.2 GLOBAL ARQUITECTURE.....	12
2.2.1. Data Sources	13
2.2.2. ETL (Extract, Transform and Load).....	13
2.2.3 Data WareHouse	14
2.2.4. DataMart.....	15
2.2.5. Information use	17
CHAPTER 3. OLAP (ONLINE ANALYTICAL PROCESSING)	18
3.1 DESCRIPTION.....	18
3.2 OLAP CUBES	18
3.2.1 WHAT IS AN OLAP CUBE?	18
3.2.2 CUBE DIMENSIONS. LOGICAL MODEL.....	19
3.2.3 OPERATIONS	19
3.3 ROLAP SYSTEMS (RELATIONAL-OLAP SYSTEMS).....	20
3.3.1. ROLAP SYSTEMS. RDBMS DEFINITION	20
3.4 MDX (MULTIDIMENSIONAL XPRESSIONS) LANGUAGE	22
3.5 XMLA (XML for Analysis)	22
CHAPTER 4. BI MODULE. DATA EXTRACTION	23
4.1 DESCRIPTION.....	23
4.2 MAIN TECHNOLOGIES.....	23
4.3 DATASOURCES.....	23
4.4 DATA WAREHOUSE CREATION.....	24
4.3.1. ETL process. Description	24
4.3.2. ETL process.Kettle jobs and transformations	24
4.5 SCHEDULING ETL PROCESS.....	27
CHAPTER 5. BI MODULE. DATAMART OLAP	28
5.1 DESCRIPTION.....	28
5.2 OLAP ENGINE. MONDRIAN	28
5.3 SALES CUBE. DESING AND CREATION	28
5.3.1. Dimensions	29
5.3.2. Measures and Calculated Members	32
5.4. CACHE MANAGEMENT	33
CHAPTER 6. BI MODULE. DATA VISUALIZATION	34
6.1 DESCRIPTION.....	34

6.2	REPORT ENGINE. JASPER REPORTS	34
6.3	REPORT MODEL ARQUITECTURE	34
6.4	REPORTS LIST	35
6.4.1.	Operator Reports	35
6.4.2.	Events manager reports.....	35
6.4.2.	Channel manager reports	36
6.5	USERS/ROLES ADMINISTRATION	37
CHAPTER 7. BI MODULE. MOBILE ACCESS		38
7.1	DESCRIPTION.....	38
7.2	MAIN TECHNOLOGIES.....	38
7.3	MULTIDEVICE ACCESS. HOW IT WORKS?	38
7.4	SMARTPHONES/TABLETS USER INTERFACE	41
7.4.1.	PC Browser.....	41
7.4.2.	Mobile-based browser.....	43
CONCLUSIONS		45
BIBLIOGRAPHY.....		46
ANNEX A. BI MODULE DATABASE SCHEMA.....		47

FIGURE INDEX

Fig. 1 Ticketing managers relationship.....	10
Fig. 2 Onebox managers relationship.....	11
Fig. 3 Business Intelligence process	12
Fig. 4 OLAP dimension definition	19
Fig. 5 Start schema definition	21
Fig. 6 Snowflake schema definition	22
Fig. 7 BI Module's ETL process.....	24
Fig. 8 Transformation icon.....	24
Fig. 9 job_carga_bi Kettle's Job	25
Fig. 10 Quartz's sheduling process.....	27
Fig. 11 Sale cube components	28
Fig. 12 Mondrian's dimension example	29
Fig. 13 Dimension mapping example	29
Fig. 14 Measure definition example.....	32
Fig. 15 Measure definition based on SQL query	32
Fig. 16 Calculated Member definition	33
Fig. 17 Repot model arquitecture	34
Fig. 18 SiteMesh arquitecture.....	39
Fig. 19 AgentDecoratorMapper usage	40
Fig. 20 SiteMesh multi-device operation	40
Fig. 21 PC browser login screen	41
Fig. 22 Reports list screen.....	42
Fig. 23 Report viewer	42
Fig. 24 Mobile-based login page	43
Fig. 25 Mobile-based reports list page	43
Fig. 26 Mobile-based report form	44
Fig. 27 Mobile-based report display	44

TABLE INDEX

Table 1 OLAP and OLPT comparison	17
Table 2 Event Manager Dimension	30
Table 3 Channel Manager Dimension	30
Table 4 Enclosure Dimension.....	30
Table 5 Sales Channel Dimension	31
Table 6 Group Dimension.....	31
Table 7 Product Type Dimension	31
Table 8 Sale Type Dimension	31
Table 9 Price Zone Dimension	32
Table 10 Promotion Dimension	32
Table 11 Report visibility by entity type	37

INTRODUCTION

The technology advances and especially the Internet proliferation have provided an opportunity to increase their market share to many economic sectors. On the ticketing industry, the Internet use as a sales channel has been a revolution in the sector, allowing access to more potential customers.

In this area we find Onebox. Onebox is a company that develops technology for selling shows and leisure activities. It is launching an innovative ticket sale and distribution system on to the market, offering the available capacity for a show in a centralized way (without quotas) on all connected channels, in order to increase client sales, ensure maximum publicity for their shows and provide the best service to spectators.

This final degree project is structured into seven chapters to give a working solution to Onebox for creating BI MODULE, providing the necessary information about the sales to its clients and to the company itself. The first three chapters put the reader into the situation to understand and become familiar with the concepts and components used on the final technological solution. The last four chapters explain each development solution stage.

Chapter one presents Onebox and how acts on the ticketing industry, and the main objectives of the sales module that give meaning to this project.

On chapter two is made an explanation of the Business Intelligence concept and the meaning it has for the business sector. Business Intelligence stages are also shown.

Chapter three explains the On-Line Analytical Processing (OLAP) concept, its main characteristics, its basic components and the various aspects in the data mining system.

Chapter four describes the first development phase of the BI MODULE, defining the data extraction process and the technologies and tools used.

Chapter five explains how to use the provided data of the previous phase to create OLAP infrastructure.

Chapter six describes how it will perform the operation of the OLAP data infrastructure by Onebox and their customers through sales reports, describing each one of them and their availability according to the user.

Chapter seven shows how will be displayed BI MODULE if accessed through a traditional web browser or from a smartphone/tablet.

Finally, on the annexes section appears technical information related to the implementation of some key points of the BI MODULE.

As an important point, the success of this project is not only determinate to get a good technological solution, insomuch as the customer satisfaction with the delivered product makes the difference between success or failure.

CHAPTER 1. ONEBOX. TICKETING SERVICES

The Internet access that millions of homes around the world have nowadays has been a great advance for lots of economic sectors. Focusing on the business of selling tickets for events, this platform has been a great advance for the ticketing world: clients can buy and access easily because they do not have to go before the exhibition to buy their ticket, saving unnecessary time. The companies have increased their results exponentially.

The two basic entities in this market are the following:

- **Event Managers:** entities in charge of organizing the event itself (where, when and how the event will take place).
- **Channel managers:** entities in charge of ticket sales.

Fig. 1 shows the relationship between them:

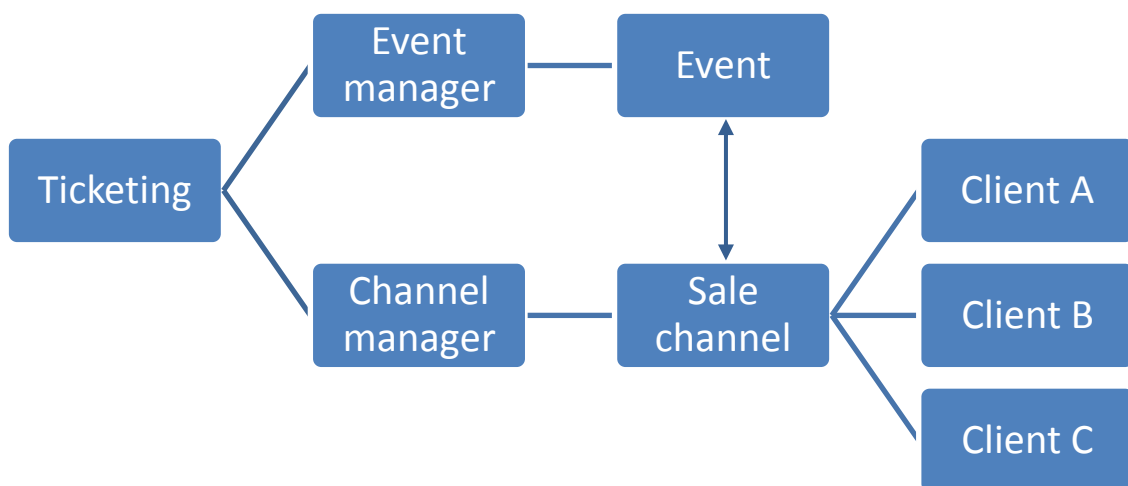


Fig. 1 Ticketing managers relationship

Each entity must to assign a specific capacity to each channel through it wants to sell, with the risk that these sales capacity allocations can not be shared between different sales channels.

This generates a functional problem: a channel could sell all their assigned locations and other, however, could keep unsold locations, not maximizing profits per event.

In order to solve this problem the company OneBox, creating a link between the event managers and channel managers, makes that the location are not exclusive per channel and are shared by all of them, as shown in **Fig.2**:

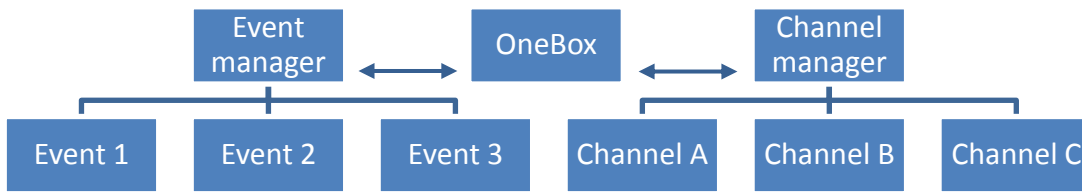


Fig. 2 Onebox managers relationship

Another problem of the sector was that the sales information was unilateral for the sales channels, it means, event managers did not have access to these data at any time, they only receive part of the information from the channels.

For this reason, the main objective of BI Module is based on offering to Onebox customers all information related to their sales volume, creating a centralized information point without contacting each other.

The project provides a platform that allows obtaining data about sales of event managers: number of locations sold per event, turnover per performance... and reports (in html, xls and pdf) that allow cross-checking by dates chosen by the event manager. In this way, they can analyze the results and decide on a reliable base.

Channel managers also have their own platform that allows them to analyze data from events which are in charge, also taking their own decisions, through the information they get from the reports that are provided.

In addition, this module should provide to Onebox as useful as its customers but from the point of view of operator. So Onebox should be able to obtain information concerning sales from a global perspective.

The project has gone one step further, allowing accessibility to the platform through any mobile based device.

CHAPTER 2. BUSINESS INTELLIGENCE

2.1 DESCRIPTION. BASIC CONCEPTS

Business Intelligence is defined as the set of techniques and tools that let the executive board to manage, to administrate and to have the enough knowledge through the information provided by the data analysis in an organization or a company.

The main goal of this process is having competitive advantage in the hard task of making decisions providing information to the company.

The **Fig. 3** shows a Business Intelligence process:



Fig. 3 Business Intelligence process

These concepts could be defined as follows:

- **Data:** minimum semantic unit that represents an attribute or a characteristic about an item. Data has not senses without a context.
- **Information:** processed data set with sense and value through a context, measure unit, with a calculation.
- **Knowledge:** process obtained thanks to the experience, values and comprehension of an issue or subject of the reality. These elements let the incorporation of new information and experiences and getting the last step of Business Intelligence procedure.
- **Decisions:** final stage of the process in which, following a subjective way, a set of determinate actions. In the Business Intelligence context, these decisions are obtained thanks to the information due to the experience and the knowledge obtained on the previous steps.

2.2 GLOBAL ARQUITECTURE

Lots of solutions could be found depending on the use and the specific application of the Business Intelligence system.

Normally, this factor is defined principally by the main objective of the system and displays those results.

There are several components that a Business Intelligence need to have in order to get the expected results.

2.2.1. Data Sources

All BI system must be supplied by the necessary data to carry out the process of decision making in the most efficient and optimal way. The source of this data can be very different. Some cases of data source can be:

- Internal data at corporate level that may be related to any activity of the company (sales, expenses, information about customers...)
- External data used as a complement or internal data support (stock quotes, foreign exchange, public data from competitors...)

From the point of view of the support, this data can be stocked in multiple formats, such as:

- Database servers
- Text files
- Web services
- Spreadsheets

2.2.2. ETL (Extract, Transform and Load)

Once data sources that are capable of being analyzed are detected, it is necessary to perform a treatment process of these data, due to in most cases, the heterogeneity of different origins and the initial structure of these data, is not the better one to be analyzed correctly.

To solve this problem, ETL processes are used.

ETL processes allow us to manipulate data from different sources in order to be able to store other data sources to analyze and thus be able to achieve the purpose of BI.

ETL process stages

An ETL process consists of three main stages, which are as follows:

- **Data extraction:** where the data from different sources are obtained. The main task of this first stage will be select and verify the extracted data to validate them, then go to the next stage, in addition to homogenize the various formats of various origins.
- **Data Transformation:** the transformation stage applies a series of business rules or functions on the extracted data to convert them on the data that will be load. Some of the sources of data require some minor manipulation of the data.
- **Load Data:** the loading phase is the moment in which data of the previous stage are loaded in the target system. Depending on the requirements of the organization, this process can have a wide variety of actions.

2.2.3 Data Warehouse

A data warehouse is a corporate database that is characterized by integrating and debugging information from one or more than one different source, and then process it allowing analysis from many perspectives and with high speed response. The creation of a data warehouse is in most cases the first step, from the technical point of view, to implement a comprehensive and a reliable Business Intelligence solution.

The main characteristics of a DataWarehouse are:

- **Integrated:** the stored data in the data warehouse should be integrated into a consistent structure, so inconsistencies between the various operational systems must be eliminated. The information also is usually structured in different levels of detail to suit the different needs of the users.
- **Thematic:** only the necessary data for the business knowledge generation process are integrated from the operational environment. Data are organized by topic to create an easy and understandable access for users. For example, all data of customers can be consolidated into a single table of the data warehouse. Thus, customers' information requests will be easier to answer because all information resides in the same place.
- **Historical:** time is an implicit part of the information contained in a data warehouse. Operational systems always show the actual state of business activity. On the contrary, the information stored in the data warehouse is used, for example, to perform trend analyzes. Therefore, the data warehouse is loaded with different values of a variable in time to allow comparisons.
- **Non-volatile:** the information store of a data warehouse exists to be read but not modified. The information is permanent meaning that data warehouse update incorporates the latest values of the variables contained in it without any action on what already existed.

The main advantage of such databases is the structures in which information is stored (star tables models, snowflake, relational cubes...). This kind of persistence of the information is consistent and reliable and allows for consultation and treatment hierarchized itself (always in a different environment to the operational systems).

As additional advantages, a DataWarehouse gives us a tool for decision making in any functional area, based on integrated and global business information. It also facilitates the application of statistical techniques and modeling analysis to find hidden relationships between different data. It also provides the ability to learn from past data and predict future situations in various scenarios.

2.2.4. DataMart

A **Datamart** is a departmental data base specialized on the data store of a determinate business area.

It is characterized by having the **optimal data structure** to analyze the information in detail from all perspectives that affect the department processes. A datamart can be powered from the data in a data warehouse, or integrating itself a compilation of different sources of information.

Therefore, to create a functional area datamart of the company is necessary to find the optimal structure for the analysis of its information, structure that can be created with an OLTP(On-Line Transactional Processing) database (with a data warehouse) or with an OLAP (On-Line Analytical Processing) database. The choice of one or another depends on the data requirements and the specific characteristics of each department. Knowing this, two types of data marts can be created:

Datamart OLAP

OLAP systems are databases for analytical procedures. This analysis typically involves reading lots of data amounts to extract some useful information: sales trends, patterns of consumer behavior, complex reporting elaboration... This is the typical datamart system and its main characteristics are:

- Access to data is usually read-only. The most common action is to consult, with very few inserts, updates or deletes.
- The data are structured according to business areas, and data formats are integrated uniformly on the whole organization.
- The data record is long term, usually from two to five years.
- The OLAP databases are usually fed information from existing operational systems, through a process of extraction, transformation and loading (ETL).

Therefore, these data marts based on OLAP cubes, explained in detail in the next chapter, which are constructed by adding, as required by each area or department, the dimensions and indicators required for each relational cube. The creation, operation and maintenance mode of OLAP cubes is very heterogeneous because depends on the final tool that is used.

Datamart OLPT

OLTP systems are database used for transaction processing. A transaction generates an atomic process (which should be validated with a *commit* or invalidated with a *rollback*) and could include insert, update and delete data operations. The transaction process is typical of the operational databases. Its main characteristics are:

- Data access is optimized with frequent tasks of reading and writing (for example, banks or hypermarkets databases have to handle lots of transactions daily).

- Data are structured according to the application level (custom program management, departmental information system...).
- Data formats are not necessarily uniform on different departments (no compatibility, data islands).
- Data historical is often limited to current or recent data.

The datamarts used on these systems may be based on a simple extract from the data warehouse, however, it is common to introduce performance improvements (aggregations and the filtrates are the most common operations) using the particular characteristics of each company area. The most common structures are the report tables that are reduced fact-tables (adding the appropriate dimensions) and materialized views.

Table 1 shows the most important differences between OLAP and OLPT systems:

	OLPT	OLAP
Data source	Operational data; OLTPs are the original source of the data.	Consolidation data; OLAP data comes from the various OLTP Databases
Data purpose	To control and run fundamental business tasks	To help with planning, problem solving, and decision support
Kind of data	Reveals a snapshot of ongoing business processes	Multi-dimensional views of various kinds of business activities
Inserts and updates	Short and fast inserts and updates initiated by end users	Periodic long-running batch jobs refresh the data
Queries	Relatively standardized and simple queries returning relatively few records	Often complex queries involving aggregations
Processing speed	Typically very fast	Depends on the amount of data involved; batch data refreshes and complex queries may take many hours; query speed can be improved by creating indexes
Space requirements	Can be relatively small if historical data is archived	Larger due to the existence of aggregation structures and history data; requires more indexes than OLTP
Database design	Highly normalized with many tables	Typically de-normalized with fewer tables; use of star and/or snowflake schemas

Backup and recovery	Backup religiously; operational data is critical to run the business, data loss is likely to entail significant monetary loss and legal liability	Instead of regular backups, some environments may consider simply reloading the OLTP data as a recovery method
----------------------------	---	--

Table 1 OLAP and OLPT comparison

2.2.5. Information use

The data hosted on the data warehouse or datamart are exploited using business analysis tools, reporting tools, alerts... Business Intelligence products are also based on these built tools. There are many systems to manage data on BI solutions, like the followings:

- Decision support systems (DSS)
- Executive Information Systems (EIS)
- Balanced Scorecard (BSC).

CHAPTER 3. OLAP (ONLINE ANALYTICAL PROCESSING)

3.1 DESCRIPTION

As described in the previous chapters, OLAP systems allow us to handle lots of data in a more efficient way from the analytical point of view because are optimized for reading tasks. This is the main feature that allows to use an olap system in the technical solution. The cubes uses OLAP system core and will be explained in more detail in the next section.

There are two kinds of OLAP systems:

- MOLAP systems (Multidimensional-OLAP systems): based on the data warehouse multidimensional tables.
- ROLAP systems (Relational-OLAP systems): based on the data warehouse relational tables.

This chapter will focus on the description of ROLAP systems because are used as technology solution that will be detailed in the next chapter.

3.2 OLAP CUBES

3.2.1 WHAT IS AN OLAP CUBE?

An OLAP cube is a multidimensional database which the physical storage of data is performed in a multidimensional vector, so these data are represented by a set of:

- **Dimensions:** parameters used to analyze the data. Examples of dimensions are time, city, gender, type of product ...
- **Measures:** numeric values, data to be analyzed. Examples of measures are devolutions, turnover ...

Therefore, the OLAP cube consists of numeric facts called measures which are categorized by dimensions. The measures are placed at the intersections of the cube, which is spanned by the dimensions as a Vector space. An OLAP cube is also called hypercube, because of the amount number of dimension that could be contain.

3.2.2 CUBE DIMENSIONS. LOGICAL MODEL

On a logic level, each of the cube dimensions could be defined as **Fig. 4** shows:

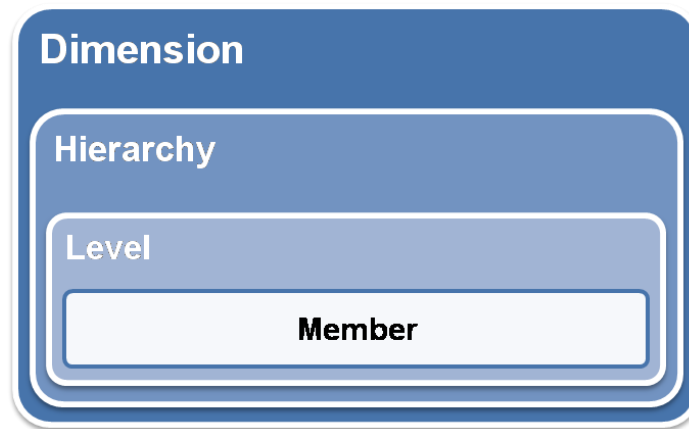


Fig. 4 OLAP dimension definition

Where:

- A **member** is a point within a dimension determined by a particular set of attribute values.
- A **level** is a collection of members which have the same distance from the root of the hierarchy.
- A **hierarchy** is a set of members organized into a structure for convenient analysis.

This data hierarchical structure to define the dimensions of a cube allows a general and a specific analysis level, depending on the level of detail in which they descend in the hierarchy.

At the same time, OLAP also displays information directly related to the specific members of each level. If you have a level that includes the promoter events within a hierarchy of tours, it is possible to be displayed values as its capacity, its start/end date... These values are the properties and can be used as many as necessary for each level.

3.2.3 OPERATIONS

- **Slice:** a slice is a subset of a multi-dimensional array corresponding to a single value for one or more members of the dimensions not in the subset.
- **Dice:** the dice operation is a slice on more than two dimensions of a data cube (or more than two consecutive slices).
- **Drill Down/Up:** drilling down or up is a specific analytical technique whereby the user navigates among levels of data ranging from the most summarized (up) to the most detailed (down).
- **Roll-up:** a roll-up involves computing all of the data relationships for one or more dimensions. To do this, a computational relationship or formula might be defined.

- **Pivot:** this operation is also called rotate operation. It rotates the data in order to provide an alternative presentation of data - the report or page display takes a different dimensional orientation.

3.3 ROLAP SYSTEMS (RELATIONAL-OLAP SYSTEMS)

ROLAP system access to data stored in a data warehouse to provide OLAP analyzes. The ROLAP systems premise is that OLAP capabilities are supported better against relational databases.

The ROLAP system uses three levels architecture:

- **Relational Data Warehouse:** this level uses RDBMS (relational database management) to manage, to access and to obtain the data. In this case, could be a specific data mart or a data warehouse
- **ROLAP Engine:** this engine provides the analytic functionality
- **Layer ROLAP engine access:** level to execute the multidimensional queries from users

The ROLAP engine integrates with presentation layers, through which users perform OLAP analyzes. After the data model for the data warehouse has been defined, data is loaded from the operational system. If is required by the data model, database routines are executed to add the data. Then the indexes are created to optimize queries response times.

End users running their multi-dimensional analysis, through the ROLAP engine, which dynamically transforms multi dimensional queries to SQL queries. These SQL queries are executed in relational databases, and their results are related by cross-boards and multidimensional arrays to return the results to users.

ROLAP architecture is able to use pre-calculated data if these are available, or dynamically generated results from the elemental data if necessary. This architecture directly accesses the data warehouse data, and supports access optimization techniques to speed up queries. These optimizations are, among others, partitioning the data at the application level.

3.3.1. ROLAP SYSTEMS. RDBMS DEFINITION

As described above, ROLAP systems make use of a relational database for storing necessary data, so that the OLAP engine, based on the cubes defined, is able to translate multidimensional expressions, which are explained in detail in the next section, to SQL statements.

At RDBMS structure level, there are two main concepts to define the tables:

- **Fact table:** This tables defines all measures and dimension relationships.
- **Dimension table:** Defines a dimension structure.

In order to mine efficiently data from an OLAP cube in a ROLAP, it is necessary that the relational model of data mart / data warehouse presents one of the following two types of topology:

- **Star topology**

In this centralized topology, the fact table is the only link between the different measures and dimensions, that are defined in separated tables. The following figure, **Fig. 5**, shows a star topology:

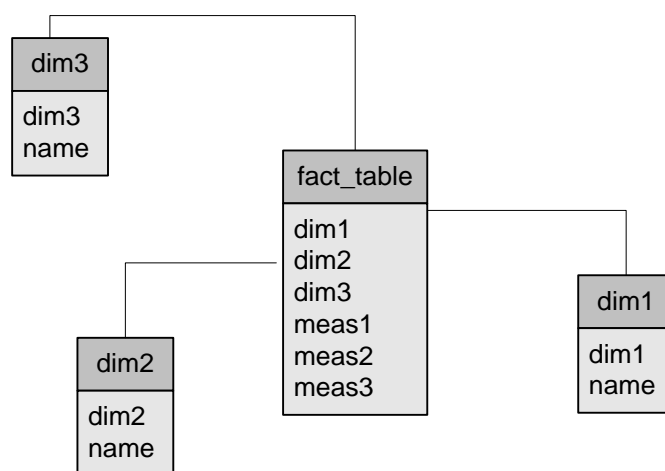


Fig. 5 Star schema definition

- **Snowflake topology**

Unlike the star topology, where one dimension is defined in a single table, in a snowflake topology a dimension can be defined in more than one table. The **Fig. 6** shows a snowflake topology:

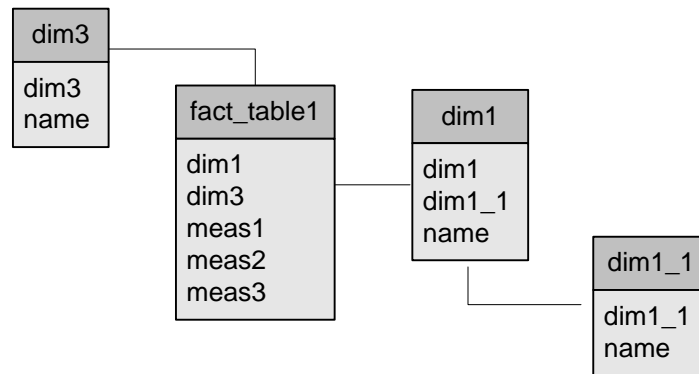


Fig. 6 Snowflake schema definition

3.4 MDX (MULTIDIMENSIONAL XPRESSIONS) LANGUAGE

Multidimensional Expressions (MDX) is a query language for OLAP databases, much like SQL is a query language for relational databases. It is also a calculation language, with syntax similar to spreadsheet formulas.

Provides a specialized syntax for querying and manipulating the multidimensional data stored in OLAP cubes. While it is possible to translate some of these into traditional SQL, it would frequently require the synthesis of clumsy SQL expressions even for very simple MDX expressions. MDX has been embraced by a wide majority of OLAP vendors and has become the standard for OLAP systems.

MDX was introduced by Microsoft with Microsoft SQL Server OLAP Services in around 1998, as the language component of the OLE DB for OLAP API. More recently, MDX has appeared as part of the XML for Analysis API. Microsoft proposed MDX as a standard, and its adoption among application writers and other OLAP providers is steadily increasing.

Since MDX is a standard language, this documentation don't cover its syntax here¹.

3.5 XMLA (XML for Analysis)

XML for Analysis is a standard that allows client applications to talk to multi-dimensional or OLAP data sources. The communication of messages back and forth is done using web standards – HTTP, SOAP, and XML. The query language used is MDX, which is the most commonly used multi-dimensional expression language today.

¹ <http://msdn.microsoft.com/en-us/library/ms145506>

CHAPTER 4. BI MODULE. DATA EXTRACTION

4.1 DESCRIPTION

This first chapter about business intelligence solution done it will include the details of the practical process of creating the data warehouse and which have been the source of the data to supply it and the corresponding ETL process.

4.2 MAIN TECHNOLOGIES

Kettle

Kettle is an open source app developed by Pentaho to create ETL processes. It has a graphical interface to create the different stages of the process in a simple and intuitive way thanks to its design tool spoon.

It is written in Java and has an integrated API that, once the ETL process is created, can be integrated easily with any Java application.

Quartz

Quartz is a task scheduler based on Java that lets you execute processes on a regular basis. In addition, it integrates Spring Framework, simplifying the deployment and the programming tasks.

4.3 DATASOURCES

The first step to create BI module is based on detecting the corresponding data sources to extract the data needed in order to create the corresponding data warehouse to analyze sales.

Initially, the data can be founded on the following MySQL database schemes:

- **ob_cpanel:** data base that contains information about business entities, events, sessions, tours, sale channels...
- **ob_compras:** data base that contains information about purchases made by clients.
- **tickets :** data base that contains information about the locations of the enclouser.

Obviously, the data analysis, as nowadays are distributed, present several difficulties to be performed in an efficiently way. The first and basic difficulty is that the data analysis and the data crosses are referenced on the schemes, creating difficulties on reading and increasing development times in case of future implementations. Also, mySQL shards difficulty is included as it would be necessary to do all the corresponding nodes consultations to obtain the expected results.

4.4 DATA WAREHOUSE CREATION

Once all the data sources are known to extract the data, have to be defined in a clear way the data model structure of the data warehouse in order to create later the OLAP datamart. So this data model have to be totally related with the dimensions and the measures needed to elaborate the sales OLAP cube that it will be defined into detail in the next chapter.

4.3.1. ETL process. Description

On the datawarehouse generation and maintenance process, Kettle will be the element that it will concentrate all the necessary data of the three schemes that has been described before in only one schema, ob_bi, as is described in **Fig. 7**:

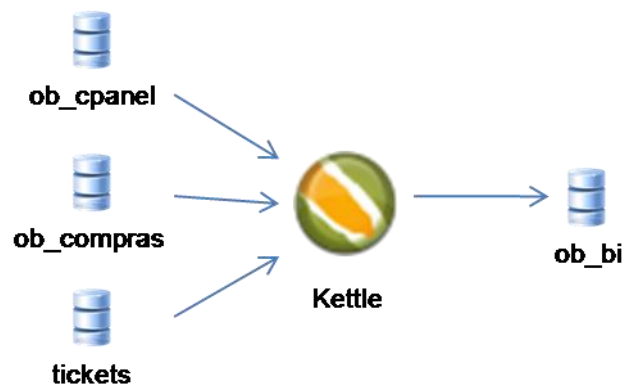


Fig. 7 BI Module's ETL process

This let to have all the necessary data for future sales analysis on one scheme independent of the rest where the users only will do reading operations because Kettle is the only one that can update the data.

4.3.2. ETL process.Kettle jobs and transformations

Kettle has two basic concepts to implement an ETL process:

- **Transformation:** define an ETL process. It is represented in **Fig 8**:



Fig. 8 Transformation icon

- **Job:** define a sequence of one or more than one transformations.

Knowing this, the transformations allow connecting to corresponding data sources and extract the necessary data in order to adapt to the requirements of the target source, and finally dump them on it.

On the solution for the BI Module, each transformation represents a process for creating a dimension, auxiliary tables or the fact table, and the job will define the sequence of these processes in sequentially way. This job called job_carga_bi is represented in **Fig. 9**:

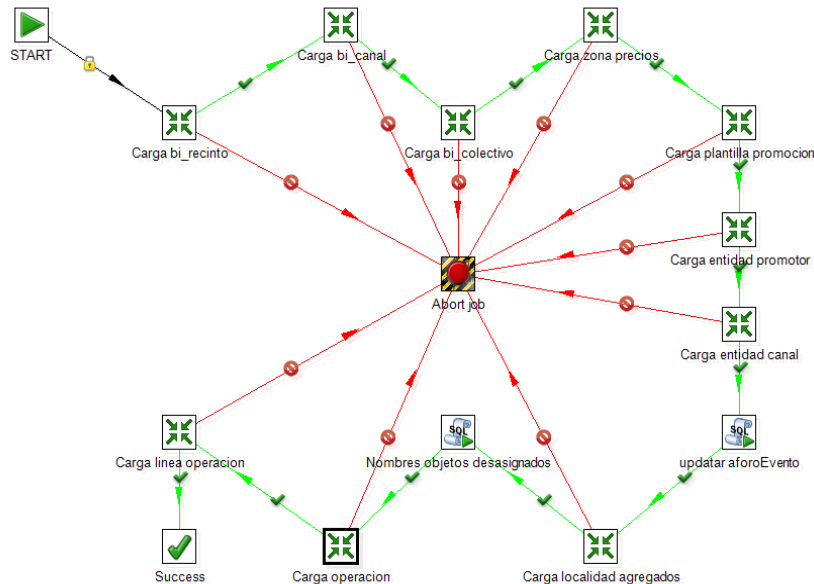


Fig. 9 job_carga_bi Kettle's Job

As shown, job runs every transformation sequentially, aborting the whole process if an error occurs in one step. In general, the process of loading the data warehouse does in each transformation an insert/delete of the data to be dumped, so content is never deleted in order to have the historic. The last step is to load transform of the fact table, since it requires that all dimension tables are loaded and updated in order to establish relations with them and the measures.

The ETL process transformations used, that it could be found on the execution order are the followings (**annex A** shows the whole database schema):

Carga bi_recinto

- Destination table: bi_recinto
- Main function: collects all the data about the configurated encloures and collects the relation with the events

Carga bi_canal

- Destination table: bi_canal
- Main function: collects all the data about the sales channels

Carga bi_colectivo

- Destination table: bi_colectivo
- Main function: collects all the data about the groups of people

Carga zona precios

- Destination table: bi_zona_precios
- Main function: collects all the data about the different prices zones of the enclosures

Carga plantilla_promocion

- Destination table: bi_plantilla_promocion
- Main function: collects all the data about the different sale promotions

Carga entidad_promotor

- Destination table: bi_entidad_promotor
- Main function: collects all the data about the event managers. These data include the events, the sessions and the operator that has the event

Carga entidad_canal

- Destination table: bi_entidad_canal
- Main function: collects all the data about the channel manager, the events, the sessions that can be bought and the channels to be bought, and the operator that has the event

Carga localidad_agregados

- Destination table: bi_localidad_agregados
- Main function: this table collects all the available tickets for each session per price zone

Carga operación

- Destination table: bi_operacion
- Main function: collects the data about the different price zone of the enclosures

Carga linea_operacion

- Destination table: bi_linea_operacion
- Main function: does the fact table function. This table collects all the measures related with the different dimensions

4.5 SCHEDULING ETL PROCESS

Finally, it is necessary the updating of the data kept on the data warehouse to do the necessities analyzes. To achieve this, it is necessary that ETL process explained above has to be executed periodically.

In this case, quartz, the task scheduler used to do this function, executes each 30 minutes the Kettle process as shown in **Fig. 10**:



Fig. 10 Quartz's scheduling process

CHAPTER 5. BI MODULE. DATAMART OLAP

5.1 DESCRIPTION

Once data warehouse data model is determined and how the ETL process is done to provide it, the OLAP datamart could be done. This datamart creation is basically the OLAP cube design and implementation where the data could be found.

As explained above, a datamart can be a part of the data contained on the data warehouse or the whole data warehouse. In the BI module implemented case, the datamart represents all the content hosted in the data warehouse. Therefore, the BI datamart MODULE will use all the data model of the data warehouse for OLAP cube creation (it is called sales cube) by Mondrian OLAP engine.

5.2 OLAP ENGINE. MONDRIAN

Mondrian is an OLAP engine, based on ROLAP, written by Java. It is one of the open source OLAP systems developed by Pentaho community more popular and extended. The Mondrian choice as OLAP engine for this BI module is based on providing the following features:

- It is based on ROLAP so it could store the data on SQL data base.
- It has a cache system to store the calculations made by the MDX statements made to reuse as necessary, thus improving response times.
- Allows MDX queries using XMLA, providing, through HTTP remote access from every platform, everywhere.
- The OLAP cubes definition and configuration is done through XML files, allowing to make changes in a simple and intuitive way

5.3 SALES CUBE. DESING AND CREATION

Sales cube creation have been done following the below procedure that shows Fig. 11:

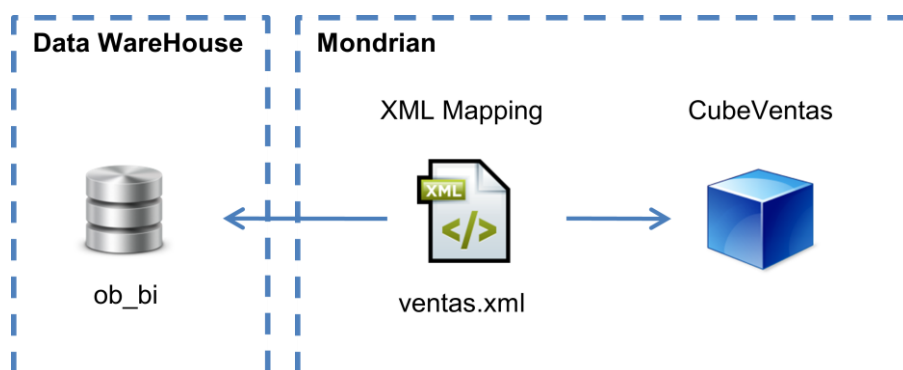


Fig. 11 Sale cube components

As shown, Mondrian will use ventas.xml file (see Annex B for full definition) to create the cube structure based on the schema ob_bi, previously created and loaded by the ETL process. For example, to create the dimension to analyze sales according to price areas of the enclosures, the XML definition is as described in **Fig. 12**:

```
<Dimension name="ZonaPrecio">
  <Hierarchy hasAll="true" primaryKey="id">
    <Table name="bi_zona_precios"/>
    <Level name="zonaPrecio" column="id" nameColumn="descripcion" ordinalColumn="descripcion"/>
  </Hierarchy>
</Dimension>
```

Fig. 12 Mondrian's dimension example

Area Price dimension consists of a single hierarchy and on one level called zonaPrecio. This definition indicates that this dimension, called ZonaPrecio, is based on bi_zona_precio table data and linking the fact table through the PrimaryKey 'id' expressed in <Hierarchy> tag. The level, called zonaPrecio, is expressed in the <level> tag, indicating its unique identifier using the attribute 'column'. The nameColumn attribute lets you show (on a user-level) a field in the table, in this case 'descripcion' and ordinalColumn attribute lets you use a table field to apply a sort order, in this case also the column 'descripcion'. To complete the dimension mapping is necessary to relate the fact table in the cube section as described in **Fig. 13**:

```
<DimensionUsage name="ZonaPrecio" source="ZonaPrecio" foreignKey="idZonaPrecio"/>
```

Fig. 13 Dimension mapping example

In this case, the 'foreignkey' attribute indicates the fact_table field that will act as indicator to link with the 'primary key' attribute defined on the previous definition.

5.3.1. Dimensions

The dimensions defined for the creation of the sales cube and its hierarchies, levels and properties are as follows:

Tiempo

This dimension, which is outside of the ETL process, it is used to show the analysis per temporary magnitudes.

EntidadPromotor

This dimension represents the event managers, as shown in **Table 2**.

Hierarchy	Level	Level Properties
-	Operadora	-
	entidadPromotor	-
	Gira	-
	Evento	Aforo FechaInicioEvento ObjetivoVentas
	Sesion	Aforo FechaInicioSesion Abono

Table 2 Event Manager Dimension

EntidadCanal

This dimension, described in **Table 3**, represents the channel managers.

Hierarchy	Level	Level Properties
-	Operadora	-
	entidadCanal	-
	Gira	-
	Evento	Aforo FechaInicioEvento ObjetivoVentas
	Sesion	Aforo FechaInicioSesion Abono
	Canal	Aforo

Table 3 Channel Manager Dimension

Recinto

This dimension, represented in **Table 4**, shows the enclosure where the show will take place.

Hierarchy	Level	Level Properties
-	Operadora	-
	entidad	-
	evento	-
	configrecinto	-
	zonaprecio	-

Table 4 Enclosure Dimension

Canal

Table 5 represents sales channel dimension.

Hierarchy	Level	Level Properties
-	canal	-

Table 5 Sales Channel Dimension

Colectivo

This dimension, as shown in **Table 6**, represents the different people groups.

Hierarchy	Level	Level Properties
-	colectivo	-

Table 6 Group Dimension

TipoProducto

This dimension, as shown in **Table 7**, represents the different products.

Hierarchy	Level	Level Properties
-	tipoProducto	-

Table 7 Product Type Dimension

TipoVenta

This dimension represents the different types of sale.

Hierarchy	Level	Level Properties
-	tipoVenta	-

Table 8 Sale Type Dimension

ZonaPrecio

This dimension represents the different price areas of the enclosures. This dimension, represented in **Table 9**, only expresses a list of them all, without a hierarchy.

Hierarchy	Level	Level Properties
-	zonaPrecio	-

Table 9 Price Zone Dimension

Promocion

Table 10 represents the different promotions.

Hierarchy	Level	Level Properties
-	promocion	PromoMan PromoAuto DesCliente

Table 10 Promotion Dimension

5.3.2. Measures and Calculated Members

Measures are described on Mondrian as shown in **Fig. 14**:

```
<Measure name="PrecioBase" column="precioBase" formatString="#,##0.00" aggregator="sum"/>
```

Fig. 14 Measure definition example

Where:

- **name:** indicates the measures name
- **column:** fact table field that contains the measure value
- **formatString:** pattern to be applied to convert to String
- **aggregator:** criteria applied when performing aggregation of values

Another way to define a measure on Mondrian is doing an SQL statement, so you can have a measure that depends on several fields of the fact table. The **Fig. 15** is an example of this:

```
<Measure name="Ventaa" formatString="#,##0" aggregator="sum">
  <MeasureExpression>
    <SQL dialect="mysql">
      (coalesce(case when `bi_linea_operacion`.`tipoEntrada` = 1 AND `bi_linea_operacion`.`idTipoOperacion` = 1 then 1 else 0 end,0))
    </SQL>
  </MeasureExpression>
</Measure>
```

Fig. 15 Measure definition based on SQL query

Another way to define measures on Mondrian is doing calculated members, thus creating through other measures with other declared measures using MDX. The next figure is a calculated member from reported measures:


```
<CalculatedMember name="Entradas" dimension="Measures">
  <Formula><![CDATA[CoalesceEmpty([Measures].[Ventas]-[Measures].[Devolucion],0)]]></Formula>
  <CalculatedMemberProperty name="FORMAT_STRING" value="#,##0"/>
</CalculatedMember>
```

Fig. 16 Calculated Member definition

All the measures definitions and calculated member could be found on Annex B.

5.4. CACHE MANAGEMENT

Mondrian's cache is one of the crucial points to provide good performance to the BI system. This cache acts so that when a user makes a query for the first time, Mondrian will search the data on MySQL, will perform the necessary data aggregation and will serve in multidimensional way. However, again made the same request, Mondrian won't need to go to MySQL because the data will be saved on cache.

Otherwise, it is necessary to do a flush of the cache every time all the data are updated in the data warehouse, because otherwise Mondrian won't detect changes on a data level. For this reason, a flush of the cache is done after completing the Kettle process and the updated data could be consulted each 30 minutes.

CHAPTER 6. BI MODULE. DATA VISUALIZATION

6.1 DESCRIPTION

Once Mondrian's OLAP cube is ready to answer queries, the next BI MODULE phase need to determinate how data will be delivered to end users.

The display OLAP data mart will be done through predefined reports. Thus, the user will have a list of reports where each provides sales data from different points of view.

6.2 REPORT ENGINE. JASPER REPORTS

For the realization of them, the report engine that has been chosen is JasperReports. JasperReports is a reporting tool that has the ability to deliver rich content to the monitor, printer or PDF files, HTML, XLS, CSV and XML. It reads its instructions from an XML or .jasper file.

It is written entirely in Java and can be used in a variety of Java applications, including J2EE or Web applications, to generate dynamic content. JasperReports reports are defined in an XML file format, called JRXML, which can be hand-coded, generated, or designed using a tool.

The choice of Jaspers Reports as the reporting engine for the BI Module has been determined by the ability to allow the report creating using XMLA datasources, allowing access to data cube sales through Mondrian. It also has a graphical tool, called iReport, which allows to create report templates.

6.3 REPORT MODEL ARQUITECTURE

Report generation is determined by two main components:

- **Static components:** The templates that define the reports, layout, design and basic MDX query that is used by XMLA against Mondrian.
- **Dynamic components:** Dependent parameters of the user. These parameters vary depending on the requested report.

The sequence with which it builds a report requested by the user through the components described above is shown in **Fig. 17**:

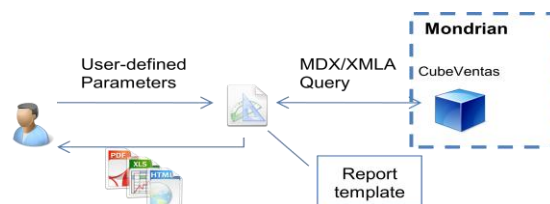


Fig. 17 Report model architecture

And it's composed by the following steps:

- The user requests a report type based on parameters that chooses through the application.
- This report, which is associated with a Jasper Reports template, interprets the received parameters and executes the MDX query associated against Mondrian via XMLA.
- After receive Mondrian's response, generate the report, showing data as defined in the template.
- Finally, the reports are rendered in a file format (PDX, XLS or HTML) defined by the user and sended to receive it.

6.4 REPORTS LIST

There are three main groups of reports. Each of these groups, which are explained below, represents a set of reports that display information related to sales about operator entities, channel managers and event managers.

6.4.1. Operator Reports

This group of reports is only accessible to operator entities. It has the following report:

Overall Domain Sales

Displays information related to the sales of the operator, according to event managers and channel managers. It has two sections:

- Sales by event manager
- Sales by cannel manager

6.4.2. Events manager reports

This group has three reports that shows the information related wit the event manager sales. This reports are the following:

Events Sales

Displays sales and refunds made in an events. It has seven sections:

- Sales by product
- Sale per channel
- Sales by Price zones
- Sales by group
- Sales by ticket type
- Refund by product
- Refunds per channel

Overall Sales

Displays the sales and refunds made by an event manager related with channel managers. It has 3 sections:

- Sales by tour
- Sales per channel
- Refunds by tour

Performance Sales

Display the information about the event managers sessions sales. It has 2 sections:

- Sales by price zone
- Sales per channel

6.4.2. Channel manager reports

This group, that contains two reports, shows the information about the sales related with channel managers

Channel sales by event

Shows the information related with event channel managers sales and refunds. It has 5 sections:

- Sales by product
- Sales per channel
- Sales per group
- Sales by discount/promotion
- Refunds by product

Overall channel sales

Shows the information about sales and refunds of a channel manager, related with event managers. It has three sections:

- Sales by event manager
- Sales per channel
- Refunds per channel

6.5 USERS/ROLES ADMINISTRATION

As described above, reports are disaggregated by entity type in three different groups. Therefore, it is necessary to create an access control for the users that wants to consult a report, filtering report groups and available parameters in function of them.

To do this, two available attributes are used to define a user. One is the entity to which it belongs, so it is possible to know which of the entity groups belongs. The second attribute is the user role. User roles determine if it has access to the report visualization. User roles affecting BI module are:

- **Operator Manager:** Enables full access to the entire system, including viewing reports.
- **Operator Analyst:** Allows only viewing reports from the point of view of an operator.
- **Channel Manager:** Enables full management of an entity channel managers, including viewing reports.
- **Event Manager:** Enables full management of an event manager entity, including viewing reports.
- **Entity Analyst:** Allows only viewing reports to a user who belongs to a event or channel manager entity.

Therefore, only those users who belong to one of the above roles can view reports. Finally, as shown in **Table 11**, depending on the type of institution to which the user belongs, the lists to be displayed are:

Entity type	Visible groups
Operator	Operator reports Event manager reports Channel manager reports
Event managers	Event manager reports
Channel managers	Channel manager reports

Table 11 Report visibility by entity type

CHAPTER 7. BI MODULE. MOBILE ACCESS

7.1 DESCRIPTION

Finally, once all the reports and the creation process is defined, the next stage defines the UI-level access.

Two different access types has been defined to carry out. The first one, from a traditional PC or laptop and the second one from a smartphone or tablet. The main reason for this difference is that they have big differences each other. It is also important from business point of view that customers can access their sales data from anywhere using a simple procedure. And finally, as important reason, the web application is developed in Flex, so that not all mobile devices allow the use of Flash.

7.2 MAIN TECHNOLOGIES

SiteMesh

SiteMesh un framework web, escrito en java, creado por OpenSymphony. Provides a web-page layout and decoration framework to achieve a consistent look and feel across multiple web applications.

JQuery Mobile

JQuery Mobile es un framework javascript basado en jQuery que permite crear sitios web optimizados para los dispositivos moviles actuales.

7.3 MULTIDEVICE ACCESS. HOW IT WORKS?

Para poder definir una interfaz de usuario en función del tipo de dispositivo que accede a BI MODULE es necesario encontrar un parámetro a nivel de comunicación HTTP que permita distinguirlos, en este caso, el campo de la cabecera User-Agent, permitiendo así conocer el navegador y el navegador Web y el dispositivo que quiere acceder al sistema.

Este proceso de detección del User-Agent del navegador y servir las vistas correspondientes se ha llevado a cabo mediante SiteMesh.

SiteMesh. Architecture

The SiteMesh architecture is based on Filters, parsers and decorators, based on **Fig. 18**:

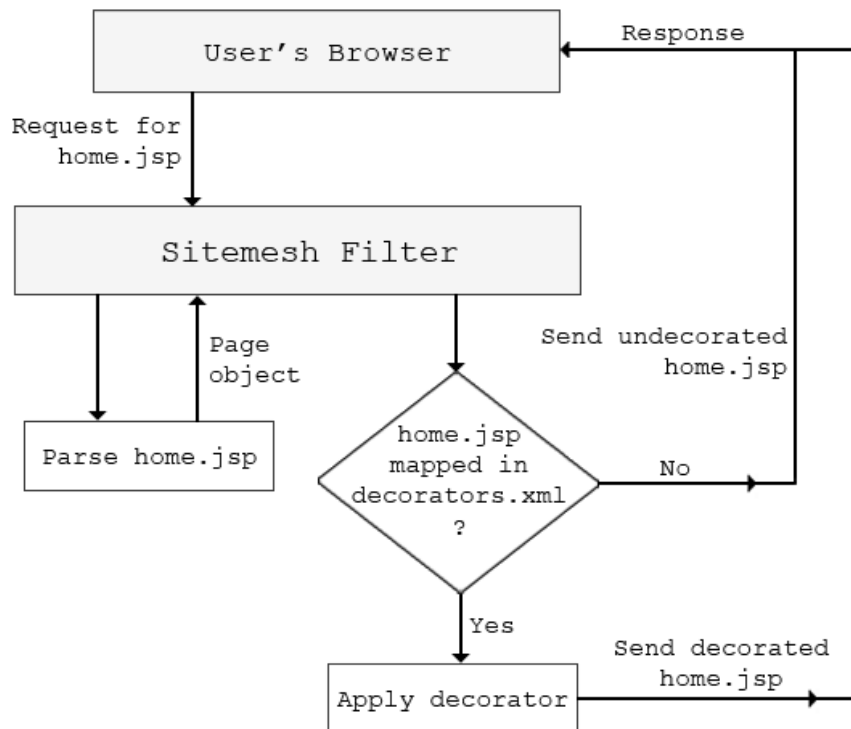


Fig. 18 SiteMesh architecture

As can be seen, when a container receives a request for a page, the filter will create a custom response object for collecting the application's response, and pass it to the web application along with the request. Then, the web application will write the requested resource in the custom response object and will return it back to the filter. Next, it will examine the content type of the response generated by the web application. Depending on the content type, the filter will determine which parser class to use for parsing the response. It will create an object of that parser class and will pass the response to it. SiteMesh determines if the requested page is mapped on the decorators file to apply it.

To apply the previous steps, two XML files are used to configure them:

- **sitemesh.xml**: This file determines the filter configuration, including the parsers and the decorator mappers. In BI MODULE, the mapper used, named AgentMapperDecorator, detects the user-agent of the http request header, appending -mobi to the name of the decorator file if matches with the following configuration described in **Fig. 19**:

```

<mapper class="com.opensymphony.module.sitemesh.mapper.AgentDecoratorMapper">
  <param name="match.iPhone" value="mobi" />
  <param name="match.iPad" value="mobi" />
  <param name="match.Android" value="mobi" />
  <param name="match.BlackBerry" value="mobi" />
  <param name="match.blackberry" value="mobi" />
  <param name="match.webOS" value="mobi" />
  <param name="match.palm" value="mobi" />
  <param name="match.Opera Mobi" value="mobi" />
  <param name="match.IEMobile" value="mobi" />
</mapper>

```

Fig. 19 AgentDecoratorMapper usage

As can be seen, this decorator mapper is only applied if the user-agent matches with a mobile-platform browser.

- **decorators.xml:** This file is used to declare the decorators that will be applied to your application pages. In this file, each `<decorator>` tag defines one decorator. The name attribute is used to give the decorator a name, while the page attribute maps the JSP page template that should be used for decorating. The `<pattern>` child element limits the scope of the decorator and specifies when it should be applied.

To resume, **Fig. 20** describes how SiteMesh works the same page requested for a mobile browser and a PC browser.

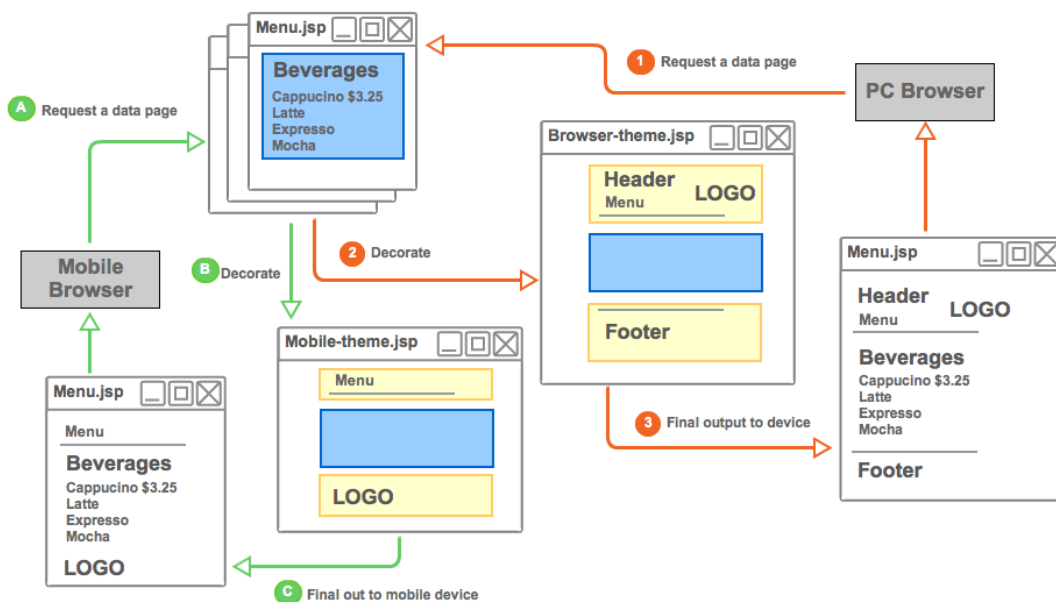


Fig. 20 SiteMesh multi-device operation

7.4 SMARTPHONES/TABLETS USER INTERFACE

Once defined how Sitemesh works, the next step is to show the different views that are used for the user interface in mobile-based browsers and pc browsers.

7.4.1. PC Browser

The user interface defined for pc browsers are the following three pages.

Login Page

Once the user is authenticated against the web application. It has the following aspect, described in **Fig. 21**:

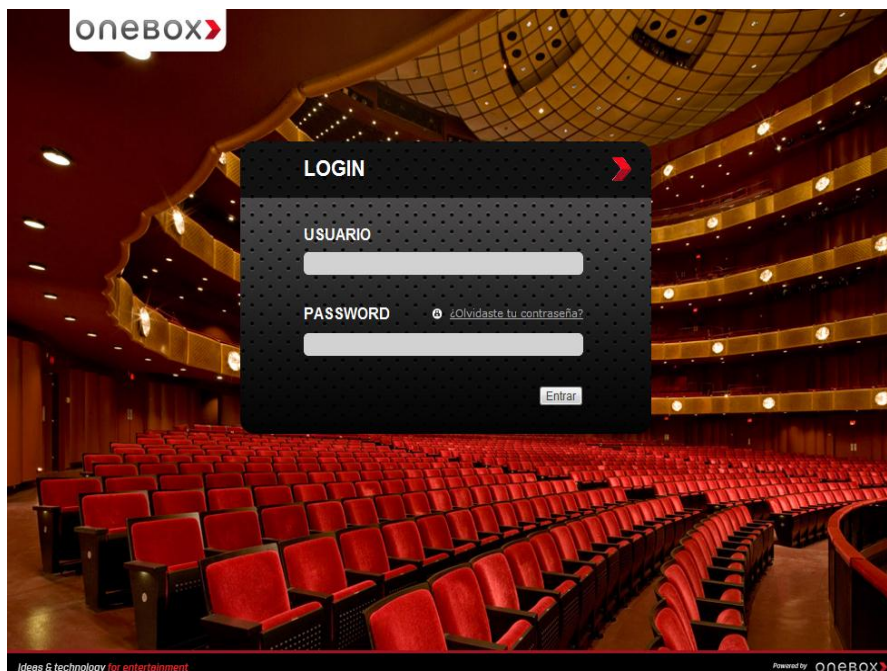


Fig. 21 PC browser login screen

Reports list

Once the user is authenticated, it is able to access to the BI MODULE, viewing the available reports list to view them.

This page has two main components. The first one is the accordion on the left side, displaying the different reports grouped by entity. The second component that appears on the right side and display the form with the available parameters for each report, changing dynamically each time a report is selected.

The **Fig. 21** shows this page:

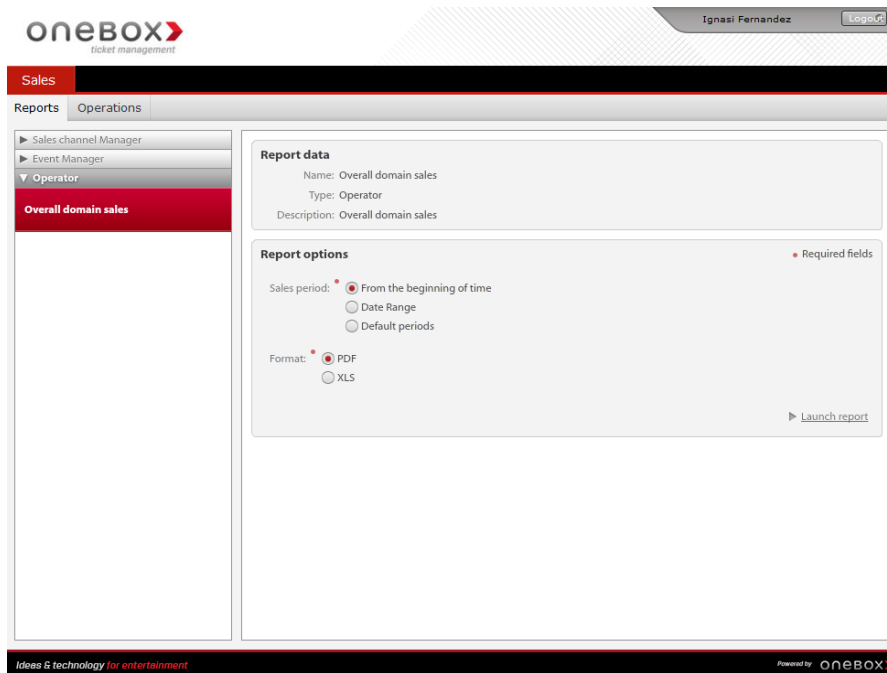


Fig. 22 Reports list screen

Report viewer

This page shows the report requested. Once the user selects the desired parameters and clicks on “Launch Report”, the report viewer appears, displaying the requested information. **Fig. 23** shows the report’s viewer aspect:

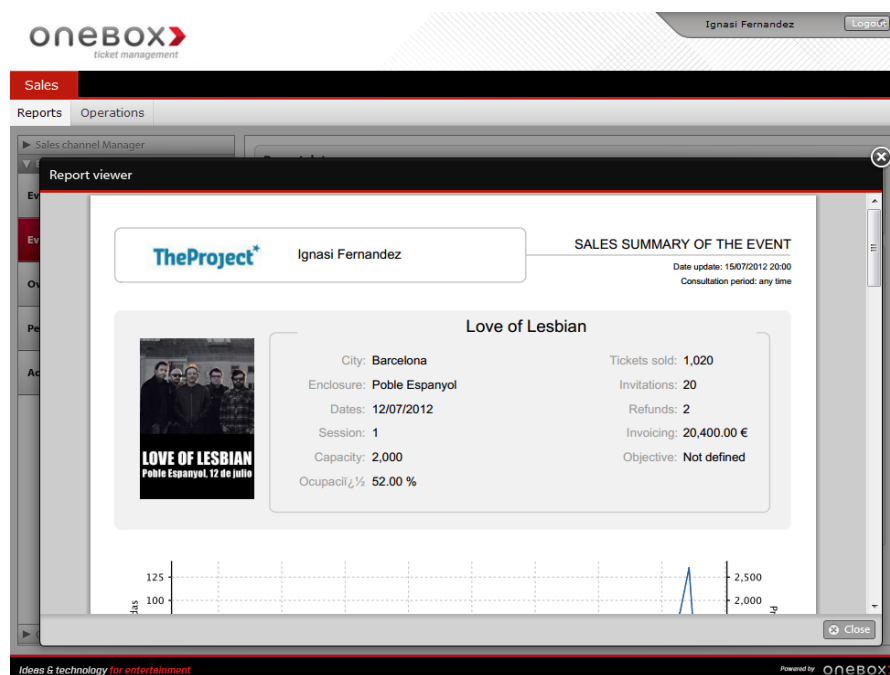


Fig. 23 Report viewer

7.4.2. Mobile-based browser

The user interface defined for mobile browsers are the following four pages.

Login Page

The page where user login to the platform. This page looks like **Fig. 24**:

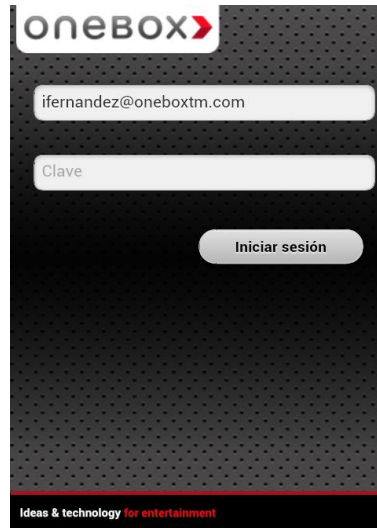


Fig. 24 Mobile-based login page

Reports list

Once the user is authenticated, the next page shows de report list grouped by entities. The next figure, **Fig. 25** shows the mobile-based reports list:

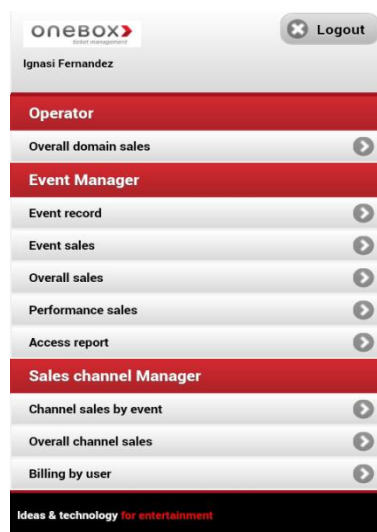


Fig. 25 Mobile-based reports list page

Report form

Once the user selected a report from the previous page, the next page shows the form associated with the available parameters. This page looks like **Fig. 26**:

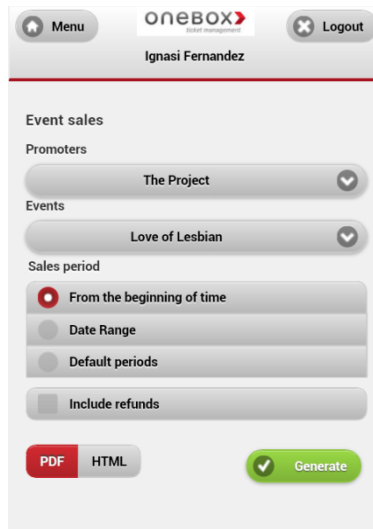


Fig. 26 Mobile-based report form

Report display

Finally, when the user selects the parameters and clicks “Generate” button, the report is displayed like **Fig. 27**:



Fig. 27 Mobile-based report display

CONCLUSIONS

Then, it's time to evaluate the final impact of this product at technological and commercial level; because of the main goal is to feedback the customers satisfaction level.

At technological level, the main goal has been achieved, in other words, the creation of a tool that allows to obtain sales reports for classical PC and laptop and the newest mobile devices, offering an innovative and prepared product to present times. Also, the architecture implemented in the development phase allows a good maintenance and scalability for future updates.

At commercial level, this product is being used by clients like Proactiv, LetsBonus, Atrapalo and The Project. Therefore, it means that the goal about to create a system that allows the business intelligence process has been reached.

Accordingly, in a global view, the objective has been reached with more or less difficulty. One of the most critical tasks was the integration of the different BI MODULE components because of the different natures and architectures

Finally, emphasize that BI MODULE is a product that continues in constant evolution both for improve the technological infrastructure for satisfy customers needs.

BIBLIOGRAPHY

- [1] Mondrian. Main documentation. [online] < <http://mondrian.pentaho.com/documentation/> >
- [2] Mondrian. Technical guide. [online] < <http://jasperserver.sourceforge.net/docs/3-7-0/Mondrian-3.0-Technical-Guide.pdf> >
- [3] Bussiness Intelligence presentation. [online] < <http://www.slideshare.net/gruponkjr/business-intelligence-3608729> >
- [4] Bussiness Intelligence documentation. [online] < <http://datawarehouse4u.info/> >
- [5] Microsoft. MDX reference. [online] < <http://msdn.microsoft.com/en-us/library/ms145506> >
- [6] Mondrian, MDX specification. [online] < <http://mondrian.pentaho.com/documentation/mdx.php> >
- [7] XMLA. [online] < <http://news.xmlforanalysis.com/what-is-xmla.html> >
- [8] Keettle. Main documentation. [online] < <http://kettle.pentaho.com/> >
- [9] SiteMesh. Main documentation. [online] < <http://wiki.sitemesh.org/display/sitemesh/Home> >
- [10] JQuery Mobile. Main documentation. [online] < <http://jquerymobile.com/demos/1.1.1/> >

ANNEX A. BI MODULE DATABASE SCHEMA

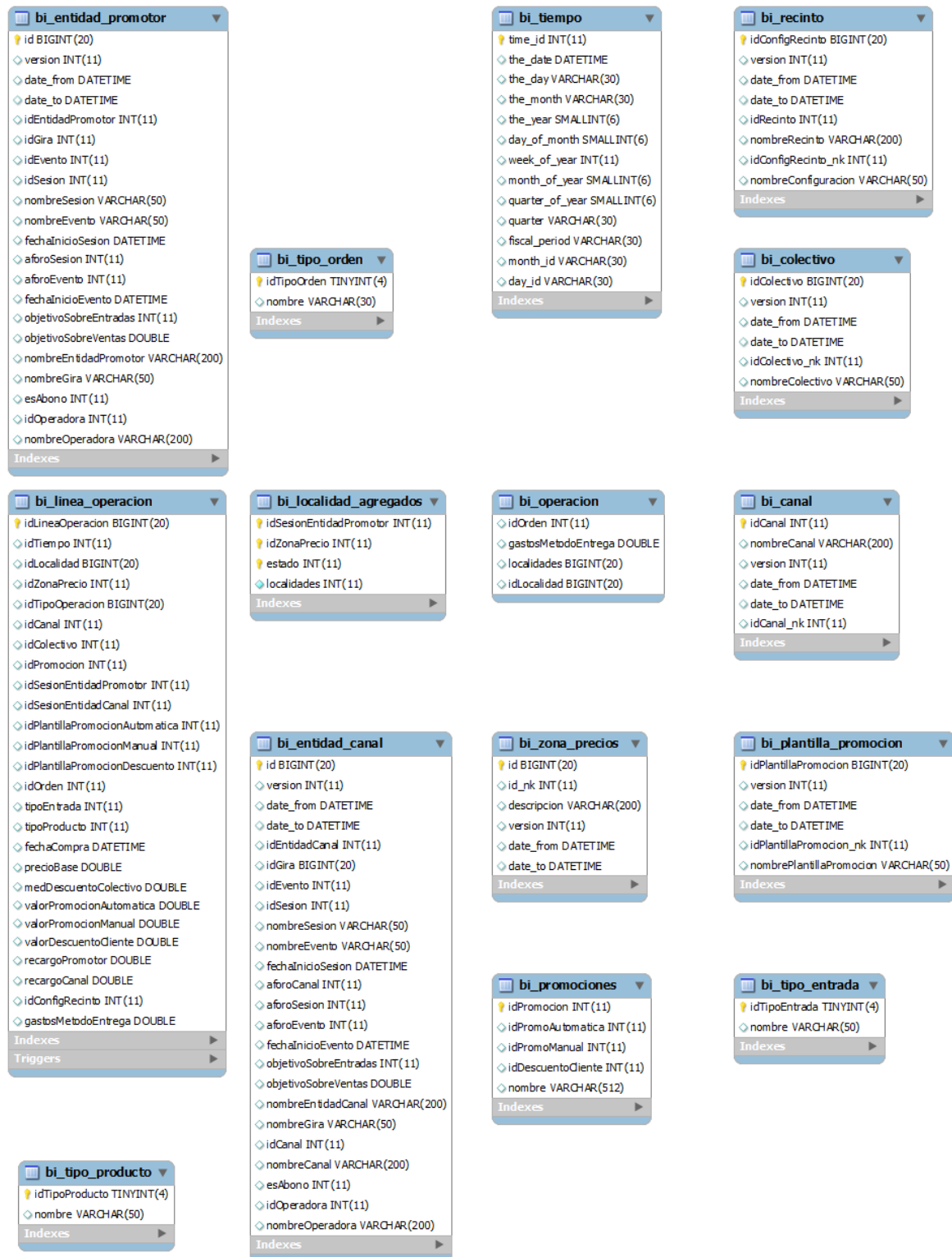


FIG A 1. BI MODULE Database Schema