

# Gesture-Based OLAP Navigation on Touchscreen Devices

---

*Master's Thesis*



Yong Hua Chan



---

# Gesture-Based OLAP Navigation on Touchscreen Devices

---

THESIS

submitted in partial fulfillment of the  
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

by

Yong Hua Chan  
born in Rotterdam



Web Information Systems  
Department of Software Technology  
Faculty EEMCS, Delft University of Technol-  
ogy  
Delft, the Netherlands  
<http://www.wis.ewi.tudelft.nl>



Incentro  
Rijnzathe 40  
De Meern, the Netherlands  
<http://www.incentro.com>



---

# Gesture-Based OLAP Navigation on Touchscreen Devices

---

Author: Yong Hua Chan  
Student id: 1153692  
Email: [yong.chan@incentro.com](mailto:yong.chan@incentro.com)

## Abstract

This report describes the design and development of a dashboard that visualizes OnLine Analytical Processing information. Users are able to navigate through the information using touch-based gestures. The source of the information is a Customer Relationship Management tool. This information is made accessible using a Data Warehouse and a Data Mart.

Before the design and implementation of the dashboard was started a few other steps were required. The first step in this process was to map out the situation at the start as well as the information needs of the end user. The design of the dashboard, comprised of a visualization and set of gesture interactions, is based on the information needs of the end user. The design was implemented using standardized web technologies. This approach has resulted in a web application usable on mobile as well as desktop devices with touchscreens.

Finally an end user validation of the system was performed to determine whether the application fulfills the needs determined at the beginning. Testing with the dashboard revealed that there were a few points on which the dashboard can be improved. Users want to explore the information with more detail. Another point is that the users required more axes to fulfill their information need. Users found the system easy to work with and the dashboard invited them to explore the information.

Keywords: Touch, Gestures, Interaction, OnLine Analytical Processing, OLAP, MDX

---

Thesis Committee:

Chair: Prof. dr. ir. G.J. Houben, Faculty EEMCS, TUDelft  
University supervisor: Dr. ir. A.J.H Hidders, Faculty EEMCS, TUDelft  
Company supervisor: Ing. C.W. van Bommel, Manager, Incentro  
Committee Member: Dr. ir. G. de Haan, Faculty EEMCS, TUDelft

---

# Preface

A few years ago I graduated with a Bachelor's degree in *Media en Kennistechnologie*. This was a Bachelor's program focused on human interaction with computers. It was during this period I decided that the business side of IT would be more suitable for me. Because of this I chose to follow the Information Architecture Track of the Computer Science Master's program, a decision I have not regretted since. I never looked back at my MKT years, at least not until my Master's Thesis Project.

A fellow student got me touch in with a company called VLC. They had an assignment for developing a touch-based dashboard for Enterprise Information Management solutions. This assignment allowed me to use my knowledge from both my Master's program as well as my Bachelor's. For me it was the ideal research project. It also turned out to be the ideal environment to carry out this project, because from the start of my internship at VLC I was treated and felt like one of their own.

Now a year later, many things have changed. The project I started has taken a massive detour to settle in its current form. And while my internship began at VLC, I am finishing it at a company called Incentro. Fortunately only the name is different and it is still the same company. I feel even more at home now than when I started here. During this period I have gained a lot of knowledge as well as making many new friends. I would like to thank all my colleagues at Incentro for their interest, advice and support, and for making me feel like a colleague instead of "the intern". I would especially like to thank Mark van de Ven, because without him I would have never even started this project.

I would also like to thank Kees van Bemmelen, John de Koning and Jan Pieter van Santen for their guidance and diligence. Without them I would still be finishing my prototype. Thanks also go out to Jan Hidders and Gerwin de Haan for their very critical views on my work. Because of their critique, the aforementioned detour was taken. But had I not strayed from the original path, the end result would have been as notable as it is now.

Last but not least I would like to thank my friends and family for their continued encouragement and support.

Yong Hua Chan  
Delft, the Netherlands  
September 7, 2012





---

# Contents

<b>Preface</b>	<b>iii</b>
<b>Contents</b>	<b>v</b>
<b>List of Figures</b>	<b>ix</b>
<b>Listings</b>	<b>x</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background of Incentro . . . . .	1
1.2 Motivation . . . . .	1
1.3 Research objectives . . . . .	2
1.4 Approach . . . . .	3
1.5 Outline . . . . .	3
<b>2 Related Work</b>	<b>5</b>
2.1 Multidimensional and OLAP visualizations . . . . .	5
2.2 Multi-Touch Interaction . . . . .	7
2.3 Touch-based BI interactions and visualizations . . . . .	8
2.4 Conclusion . . . . .	10
<b>3 Starting Situation</b>	<b>11</b>
3.1 Salesforce as information source . . . . .	11
3.2 Data Warehousing . . . . .	12
3.3 Data Marts . . . . .	12
3.4 Touchscreen . . . . .	13
3.5 Conclusion . . . . .	13
<b>4 Information need</b>	<b>15</b>
4.1 Description of the End-Users . . . . .	15
4.2 Interviews . . . . .	15
4.3 Use Cases . . . . .	16
4.4 Conclusion . . . . .	18

<b>5</b>	<b>OnLine Analytical Processing</b>	<b>19</b>
5.1	OnLine Analytical Processing Basics . . . . .	19
5.2	Serving OLAP data . . . . .	20
5.3	Getting OLAP data . . . . .	21
5.4	Limitation of the current solution . . . . .	24
5.5	Conclusion . . . . .	24
<b>6</b>	<b>Design of the Dashboard</b>	<b>27</b>
6.1	Visualization . . . . .	27
6.2	Touch Interaction Design . . . . .	30
6.3	Conclusion . . . . .	33
<b>7</b>	<b>Implementation</b>	<b>35</b>
7.1	Technical Design of the Dashboard . . . . .	35
7.2	Challenges for the Implementation . . . . .	36
7.3	Retrieving Information . . . . .	37
7.4	Visualization . . . . .	41
7.5	Touch Navigation . . . . .	43
7.6	Issues and additions to the original design . . . . .	47
7.7	The dashboard on the iPad . . . . .	48
7.8	Conclusion . . . . .	49
<b>8</b>	<b>Evaluation</b>	<b>51</b>
8.1	Testers . . . . .	51
8.2	Testing Methodology . . . . .	51
8.3	Results . . . . .	53
8.4	Possible Improvements for the Dashboard . . . . .	54
8.5	Conclusion . . . . .	55
<b>9</b>	<b>Revised Dashboard</b>	<b>57</b>
9.1	Improvements to the Visualization . . . . .	57
9.2	New Gestures . . . . .	57
9.3	Conclusion . . . . .	62
<b>10</b>	<b>Conclusions and Future Work</b>	<b>63</b>
10.1	Research Question . . . . .	63
10.2	Contributions . . . . .	63
10.3	Conclusions . . . . .	65
10.4	Recommendations and Future work . . . . .	65
	<b>Bibliography</b>	<b>67</b>
<b>A</b>	<b>Data mart</b>	<b>71</b>
<b>B</b>	<b>OLAP Schema</b>	<b>73</b>
<b>C</b>	<b>Excerpts from user tests</b>	<b>75</b>
C.1	User Test 1 . . . . .	75

---

C.2	User Test 2 . . . . .	77
C.3	User Test 3 . . . . .	78
C.4	User Test 4 . . . . .	79
C.5	User Test 5 . . . . .	80
<b>D</b>	<b>Evaluation Questionnaire</b>	<b>83</b>



---

## List of Figures

2.1	Example of a trellis plot [13]	5
2.2	An example of multiple views [35]	6
2.3	An early visualization of Parallel Coordinates	6
2.4	Parallel Sets	6
2.5	OLAP visualization designed by Techapichetvanich and Datta [29]	7
2.6	Tap	8
2.7	Swipe gesture	8
2.8	Unpinch gesture	8
2.9	Pinch gesture	8
2.10	Qlikview for the desktop and the iPad [23]	9
2.11	Two possible views of the Roambi iPad app [25]	9
3.1	Example of a Star Schema [36]	12
5.1	General structure of a schema	20
5.2	Example region hierarchy	22
5.3	Example time hierarchy	22
6.1	Examples of possible visualization channels [28]	28
6.2	Early sketch for the prototype	29
6.3	Sketch of a vertical bar chart	29
6.4	Gesture for slicing and the result	31
6.5	Gesture for dicing and the result	32
6.6	Gesture for drilling down	32
6.7	Gesture for drilling up	33
7.1	Overview of the technical software design	36
7.2	Example region hierarchy	38
7.3	Example time hierarchy	39
7.4	Simplified structure of a Cell after post-processing	42
7.5	Visualization prototype	43
7.6	Horizontal and vertical bars from an early prototype	43
7.7	Screen capture of the prototype used for the evaluation	48

9.1	Finished dashboard prototype . . . . .	58
9.2	The gesture for dicing remains the same . . . . .	58
9.3	New gesture for dicing . . . . .	59
9.4	New gesture for drilling up . . . . .	60
9.5	New gesture for drilling down . . . . .	60
9.6	New functionality has been added for removing members with this gesture	61
9.7	The gesture for comparing different members is shown on the left. The result is on the right . . . . .	61
A.1	Columns from the fact-table on the data mart . . . . .	71
A.2	Columns from the dimension table on the data mart . . . . .	72
B.1	Schema for the OLAP Server . . . . .	74

---

## Listings

5.1	Basic MDX Query . . . . .	23
5.2	Using a set in MDX . . . . .	23
5.3	Example of MDX functions . . . . .	23
7.1	Initial Query . . . . .	39
7.2	Slice . . . . .	40
7.3	MDX Dice . . . . .	40
7.4	MDX drill down . . . . .	40
7.5	MDX drills . . . . .	41
7.6	Referencing the multitouch library . . . . .	44
7.7	Adding the TUIO object to the page . . . . .	44

# Chapter 1

---

## Introduction

### 1.1 Background of Incentro

Incentro is a company based in the Netherlands specializing in Enterprise Information Management (EIM). EIM focuses on the fields of Business Intelligence, Enterprise Content Management and Web Content Management. Incentro uses Enterprise Search to bridge the gaps between those separate fields, creating a whole which is greater than the sum of its parts. But EIM is not just about the technical side of information. Incentro believes that people are the central piece to the EIM puzzle. The focus should be on providing people with information that is specific, relevant and understandable. When people are put first they can be inspired by information that is of value.

Incentro is also a company that believes in growth. By adopting an organizational structure called the *cell model*, Incentro has been expanding rapidly. During the last fifteen years, Incentro has grown from a company with only a few employees to a team of over one hundred and fifty people, active in three different countries.

### 1.2 Motivation

The rapid growth that Incentro has enjoyed over the years has brought many advantages, but unfortunately the increase of people and activities has also brought some challenges. One of these challenges is that it is harder and harder to keep track of all the sales activities across all the offices of Incentro. To address this issue Incentro has recently begun using a Customer Relationship Management (CRM) tool from Salesforce.com. The Salesforce tool provides a lot of functionality regarding the management of sales information, but some people at Incentro feel that this information could be of even greater value.

One of the suggested improvements is showing historic sales data, which is not possible with the current tool. To make the historic data available Incentro has chosen to utilize their in-house expertise. Using Business Intelligence (BI) techniques, a Data Warehouse has been created to keep track of the historic sales information. Currently, this data still needs to be made available for use by sales managers. This will be done by using a technique called OnLine Analytical Processing (OLAP), a BI technology which provides multidimensional information.

Next to in-house software expertise Incentro also possesses technology in the form of hardware. The company recently invested in a large multi-touch screen. This is used in conferences for presentations and for demonstrating the possibilities of their EIM solutions.

Incentro would like to combine the touchscreen with the OLAP information to create a touch-enabled dashboard for multidimensional data. The aim is to create something, which embodies the philosophy of Incentro: to provide their own sales managers with usable and understandable sales information so they can be inspired to take action.

Another reason for creating a touch enabled OLAP dashboard is the growing trend of mobile BI information access. Gartner predicts that by the year 2013 one third of all the BI functionally will be accessed on hand-held devices [12]. During the process of developing a touch enabled BI dashboard, lessons can be learned for developing BI application on these hand-held devices. Also by developing a touch-enabled dashboard, Incentro will promote themselves as a trend-setting company in the field of EIM.

### 1.3 Research objectives

The main goal of Incentro is to pursue their motto and enable the sales information to inspire their managers and customers. Incentro wants to achieve this by providing the resources to maximize their chance of gaining new insights into CRM data. One of these resources could be an information system with an intuitive and natural interface to explore the information. This will bring managers in touch with the data in a playful manner, inviting them to play around with the information and hopefully increase their knowledge about it.

This report will focus on the realization of the goals of Incentro regarding the touchscreen and the Salesforce.com information. These goals have been formalized into the main question of this thesis:

*"How can a gesture-based interface support the intuitive navigation through OnLine Analytical Processing information by end users, in a Customer Relationship Management domain?"*

This main question handles many different topics on a high level of abstraction. To make the question a bit more manageable, it has been broken down into smaller sub-questions, which can be found below.

1. What information needs does a sales person at Incentro have?

It is important that the information needs of the end user of the system are listed. This knowledge will be used for the design but also for validating the finished product.

2. How can a gesture based interface be designed to fulfill those information needs?

With greater insight into the information needs, an interface will be proposed to address those information needs. This interface will combine aspects from information visualization and touch interaction.



### 3. How can such an interface be built?

One of the goals of this thesis is to evaluate the design. In this case the evaluation will be done by creating a prototype and test it with users.

## 1.4 Approach

In order to come up with answers to the questions posed in the previous section, the following approach will be taken. First of all, literature related to the work in this thesis is introduced, after which the situation at the start of the project is discussed. This is done to provide a frame of reference for the rest of this report. The following step is mapping the information needs of sales managers using interviews. Based on the information gained from these interviews use cases are specified. Together with sources from literature these use cases will provide the basis for the software design. The design will be comprised of two parts, namely a visualization design and a proposed touch interaction model. After the design has been completed, it will be implemented. To show whether or not the design and the implementation fulfill the information needs of the end-users, the application will be evaluated through user test.

## 1.5 Outline

This thesis report will continue with a discussion on related work in Chapter 2, to give some more context to the topics described in this report. Chapter 3 will examine the situation at the start of the project as well as the preconditions of the project, which have been determined by Incentro. Afterwards the information needs of the end users will be mapped in Chapter 4. Following this the OnLine Analytical Processing technology is discussed in Chapter 5. In Chapter 6 a design proposal is presented, which is based on the information needs of the end-users, covered in Chapter 4. To evaluate the design a prototype will be implemented and this process of implementing the dashboard will be described in Chapter 7. To see whether the design and implementation fulfill the information need of the end-users, the dashboard will be subjected to user tests. This process and its result are presented in Chapter 8

Finally this report will end with a concluding chapter. In this chapter the contributions of this project are reiterated. The contributions are evaluated against the main research questions, to see to what extent they have been answered, and possible future work is discussed.



## Chapter 2

---

# Related Work

The main goal of this project is to design and implement a dashboard, which provides users with multidimensional Business Intelligence or OLAP information. Furthermore this dashboard should also allow users to navigate through the information using touch interactions. This project will combine aspects from Business Intelligence, information visualization as well as touch-based interactions. Because BI technologies will be discussed in following chapters, this chapter will focus on multidimensional information visualization and touch interactions. This is done to give some more context to the work of this thesis.

First works in the field of multidimensional data visualization are reviewed, after which a short summary is given on multi-touch computer interaction. After which the combination of the two areas discussed, followed by a concluding section.

### 2.1 Multidimensional and OLAP visualizations

Over the years many different approaches have been developed for visualization information. The same is also true for showing multi-dimensional data. In 1983 Tufte suggested the use of *small multiples* which are a collection of multiple graphs arranged in a grid [30]. This overview of grids enables users to quickly compare different data dimensions. An example of these small multiples - also known as trellis plots - is shown in Figure 2.1.



Figure 2.1: Example of a trellis plot [13]

A concept which builds further upon the idea of small multiples is that of multiple views. Instead of showing a grid of graphs, which are of the same type, the information is shown from different perspectives or views. Multiple views are also often used to display multi-dimensional data. Buja [6] was one of the first to work with this concept, which has evolved over the years. Weaver [35] mentioned more recent implementations, one of which can be found in Figure 2.2

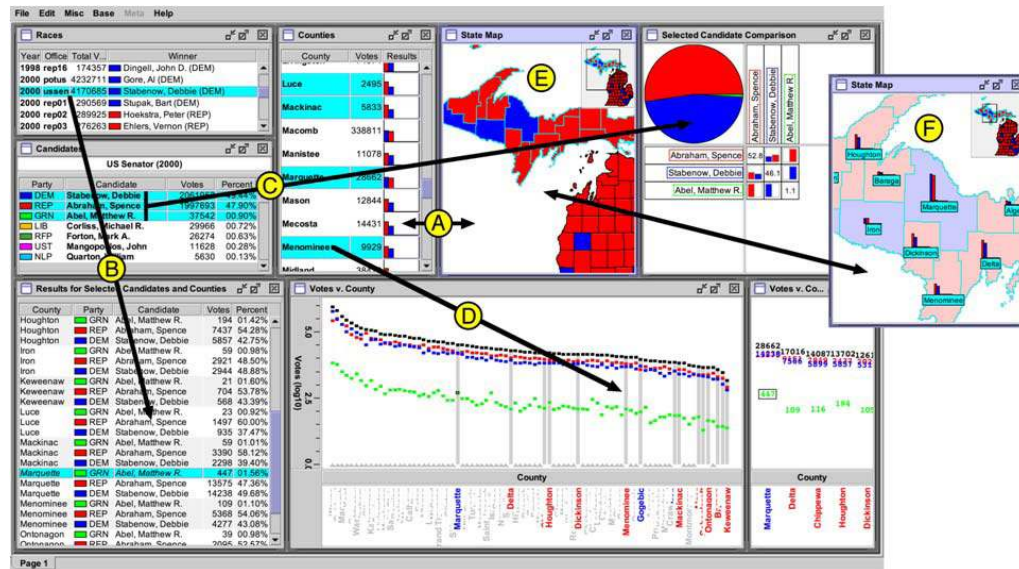


Figure 2.2: An example of multiple views [35]

Another popular visualization technique used for visualizing multidimensional data is Parallel Coordinates. It was proposed by Inselberg in 1980 [15] and has evolved over the years. Figure 2.3 shows a parallel coordinate system. Parallel coordinates show different axes parallel to each other, and lines intersecting with these axes. The axes represent different dimensions of the information and the lines represent entities of data.

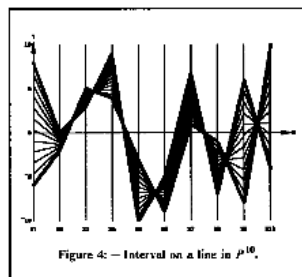


Figure 2.3: An early visualization of Parallel Coordinates

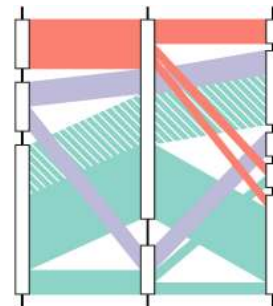


Figure 2.4: Parallel Sets

Unfortunately because the axes in multiple coordinates are continuous, they do not lend themselves well for dimensions which are categorical or discrete. In the case of categorical axes the lines representing the value entities will converge on a few points,

or in the worst case scenario of one value, all the lines will converge in single point. This makes the graph harder to interpret. To address this problem Bendix [2] developed the concept of parallel sets. In parallel sets the axes are not a continuous scale but are subdivided in the different categories of those axes. The size of the categories is proportional to the frequency of the number of occurrences. Also the (converging) lines have been removed and replaced by bands, of which the width is also determined by frequency. An example can be found in Figure 2.4.

The concept of multi-dimensional information has given rise to the idea of 3 techniques to show this information. An example is Eick [10], who developed a system visualizing OLAP information in a 3D environment.

Another visualization which has been specifically designed for OLAP information is described by Techapichetvanich and Datta [29]. As can be seen in Figure 2.5 this system uses vertical bars to represent different dimensions of the information. These bars are subdivided into the possible values and their widths are determined by frequency. The bars are also arranged in a hierarchical manner - this way it is possible to navigate to a more detailed information level.

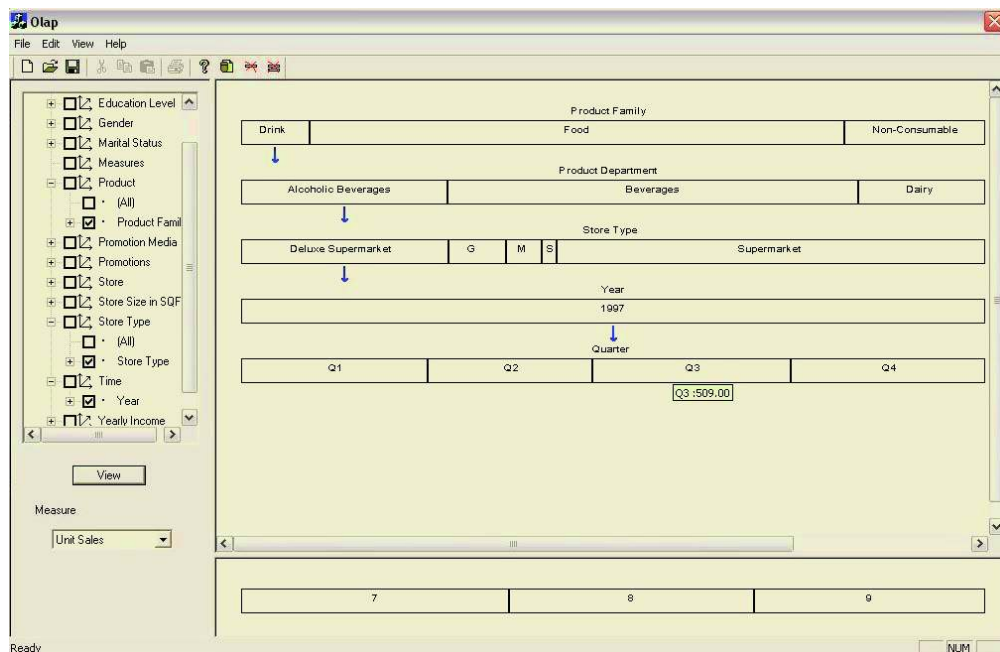


Figure 2.5: OLAP visualization designed by Techapichetvanich and Datta [29]

## 2.2 Multi-Touch Interaction

In recent years the number of mobile touch-enabled devices has seen a phenomenal growth. If analysts are right, the market for these devices will only be growing even faster. Not only has the usage of touchscreens become commonplace, the interactions used for navigating with them are also very similar. The most commonly used in-

interactions are illustrated below. These illustrations are based on images provided by GestureWorks® ([www.gestureworks.com](http://www.gestureworks.com)).

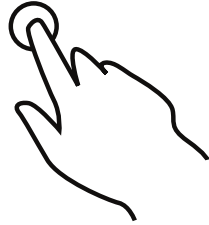


Figure 2.6: Tap

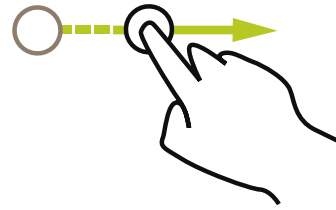


Figure 2.7: Swipe gesture

Figure 2.6 and 2.7 show some single touch interactions. The most basic interaction on touchscreen devices is a tap. It often provides the same interaction as a mouse click would allow for. Another single touch gesture is a swipe gesture, often used for scrolling to navigate through a page. The figures below show two multi-touch gestures which are commonly used in existing touchscreen devices. Zooming in is often done using an unpinching gesture, while the reverse zooming out is accomplished by performing a pinch.

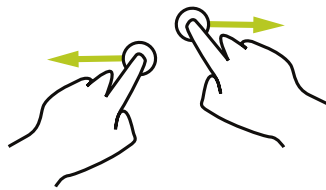


Figure 2.8: Unpinch gesture

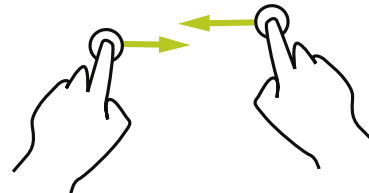


Figure 2.9: Pinch gesture

## 2.3 Touch-based BI interactions and visualizations

The increasing use of mobile-touch devices is not only due to consumers. The adoption of these devices is high among business users as well. Because of this many established BI companies have developed mobile versions of their existing desktop applications. Finley [11] lists several of these. These applications focus on Business Intelligence in general and not specifically on multidimensional information. In the section below some of these apps will be discussed.

### 2.3.1 Visualization

Often mobile BI applications are modeled after their desktop counterparts - if they have one. An example of this is QlikView, a data analysis tool for desktop environments. QlikView offers many differing visualizations to users e.g. charts, graphs, plots

and dashboards. Figure 2.10 shows multiple views on both the desktop and the iPad version. The similarities between the two versions can be clearly seen - the colors may be different, but the visualizations are very much alike. This has probably been done to present a consistent user interface across different devices to their customers.



Figure 2.10: Qlikview for the desktop and the iPad [23]

Another company developing mobile BI applications for the iPad is Roambi [25]. While Qlikview is an established company with existing desktop applications, the main focus of Roambi is mobile applications. This means the mobile application does not need to be modeled after an existing software package. Figure 2.11 shows two views on the information available in the Roambi application. The layers view is using the *multiple views* visualization discussed in previous sections. But the cardex view is using a different visualization technique altogether.



Figure 2.11: Two possible views of the Roambi iPad app [25]

### 2.3.2 Touch Interaction

Of the applications described by Finley, Incentro has experience with MicroStrategy, Pentaho and Qlikview. BI experts at Incentro have confirmed that most of the touch interaction on the mobile application is comprised of point-and-click actions. Some gestures are used, but these are limited to swipes, used to "flip" through the pages. So the interaction is similar to that on the desktop applications on which many of the mobile apps are based. The same is true for other iPad BI apps. Videos demonstrating interactions with other iPad BI applications [24] [39] also showed mostly point-and-click interaction. Other interactions such as swiping were mostly used for showing other pages or scrolling through them. These gestures were not used to navigate through the information.

## 2.4 Conclusion

This chapter has provided information on works related to visualizing OLAP information as well as touch interactions on mobile devices. Additionally the combination of the two has been discussed in the section regarding mobile BI application. This information has been provided to give some more context for the work done during this project.

There are several options for the visualization of multidimensional information. One example is using multiple views, an evolution of trellis plots. Another option is using *parallel coordinates* where parallel axes are used to represent the different dimensions, but for OLAP *parallel sets* are more suitable. An interesting idea is proposed by Techapichetvanich and Datta, using stacked bars to show the different dimensions.

In recent touch interaction has become very widespread. Many of the interactions are the same across different devices, such as tap, swipe and pinching. This means that users can switch easily from different devices, but this also means that it might confuse users if commonly used gestures are used for a different task.

Finally there are many different touch-enabled BI applications available, mostly for the iPad. However most of these applications only use point-and-click interaction and swipes for scrolling. There is little to no use of gestures to navigate through the information itself.



## Chapter 3

---

# Starting Situation

This chapter describes the situation at Incentro at the start of the project. The main reason for this is to give an overview of the preconditions and constraints of the assignment at Incentro.

The foundation for the dashboard is a Data Warehouse in which Customer Relationship Management information is stored. To give a better idea about what kind of information is stored there and why, this chapter will dive a bit deeper into the subject. But first the concept of data warehousing is explained. After this the data itself will be discussed. This will be followed by a short description of the touchscreen used for the dashboard. Finally a short summary will be given at the end of the chapter.

### 3.1 Salesforce as information source

The source of the data used for the dashboard is Salesforce. Salesforce is a company providing a cloud based Customer Relationship Management (CRM) tool. This tool allows users to manage information related to their sales activities. It also allows users to keep track of their customer *accounts* as well as sales *opportunities*, both of which are stored in the cloud. Using cloud technology users are able to use this information from any Internet connected device in real time [26]. It is used by Incentro to increase the efficiency and effectiveness of their sales efforts.

Salesforce provides a lot of functionality including the creation of dashboards, but Incentro feels that there are some features lacking. The first point is that the creation of dashboards is too complex for the actual users of the application. If there is demand for a new dashboard, this is usually implemented by a database expert. The second point is that these dashboards are lacking historic data. The dashboards provide a good overview of the sales activity and allows user to analyze the information, but the only data that is available is a snapshot of the current situation. It is not possible to get information over a longer period of time, for example to analyze trends. The final point of improvement is that Incentro would like to link the information from the Salesforce environment to other information. The most obvious option would be linking the data with information from their financial back office. To facilitate these improvements the decision was made to store the Salesforce data in a Data Warehouse.

## 3.2 Data Warehousing

A Data Warehouse (DWH) is a database on which data is stored for the purpose of reporting and analyzing. A DWH can combine information from multiple database sources, and provide a single point of access for high level information. Such information can be used to support decision making processes within an organization [33].

Data Warehouses are filled using a process called ETL. The three letters of the ETL acronym represent the steps required for filling a DWH: extract-transformation-load [16]. The first step is extracting data from the original sources. Often the sources are comprised of multiple databases spread across different systems. It is therefore possible that these databases have different structures or patterns, while more uniformity is required. This is where the transformation step comes in. Transforming the data is done to alter the extracted information so that it conforms to the needs of an organization or to technical requirements. After the transformation step is completed the final step is loading the information on to the DWH so that it can be accessed and queried by users.

## 3.3 Data Marts

A Data Warehouse database can be queried directly but usually the available information is disclosed using data marts. A data mart (DM) is a subset of the entire DWH, which is designed to represent a single process within an organization.

DMs are often stored in a multidimensional data structure called a star schema, to make the access to the information faster than in operational databases. Operational databases commonly use a normalized data structure to facilitate fast data storage. Unfortunately this also means that querying large amounts of data can be very slow, because of the potentially high amount of interdependencies. To increase the query speed of large data amounts, the data structure is denormalized into a star schema. A star schema is comprised of a single fact table and one or more dimension tables. The fact table contains numeric data for an event or transaction and the dimensional tables contain more descriptive data. An example of such a star schema can be found in Figure 3.1.

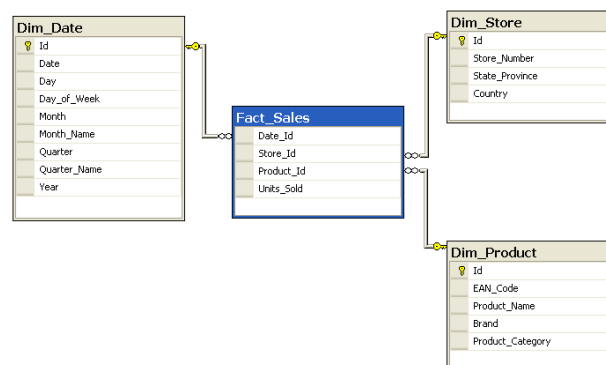


Figure 3.1: Example of a Star Schema [36]

Incentro has set up a DWH to dispose a few of the points discussed in section 3.1. To address the lack of historic data a script has been developed to renew the information. Every few days this script is run to check whether any *account* or *opportunity* data has been changed or if there is any new data. If there is any new data, this new information is copied to the DWH. When information has changed a few extra steps are needed. The edited data is copied to the DWH as well, however the existing entry in the DWH is invalidated but still kept in the DWH. This way all the historic data is saved and with this information it is possible to go back to a certain point in time and see what the situation was then.

Based on the information stored in the DWH, a DM has been created containing the information related to sales. This information is stored using a star schema and will be used as the information source for the dashboard. Currently there is one fact table and one dimensions table of which the column names can be found in Appendix A. Even though the DWH is updated regularly, the DM is still a test environment and is not updated as often. This means that it is possible that the information used by the dashboard has not been updated with the most recent information.

### 3.4 Touchscreen

Next to the software side there is also a hardware side. The dashboard will be designed for use on the touchscreen used by Incentro. This touchscreen is actually not a touchscreen as everyone is accustomed to now. It is rather a television set with a touch overlay placed in front of it. This overlay can be connected to a Windows, OS X or Linux computer. When the touch overlay is connected to the computer as an input device and the television is used as output device, the whole system will act just like a regular touchscreen.

### 3.5 Conclusion

The data used for the dashboard originates from the Salesforce CRM tool used by people at Incentro. However Salesforce can only provide the information as it is now and cannot show historic data. This is why the information is extracted from the Salesforce database and stored in a Data Warehouse. The data is then made accessible using a data mart, which stores the information using a star schema. This data mart will provide the data used for the dashboard.



## Chapter 4

---

# Information need

In the previous chapter the source of the information was discussed as well as how this data is made available. But just making information available is not enough. This information needs to be used by people who can take appropriate action. This chapter will discuss who the end-users of the dashboard are in Section 4.1, as well as what they will be using the information for. These goals have been mapped using interviews, which are discussed in Section 4.2. Based on these interviews, use cases have been formulated which will be discussed in Section 4.3. These use cases will be utilized in later chapters to determine how well the finished product fulfills the information needs of the end-users.

### 4.1 Description of the End-Users

The end-users of the finished product will be people at Incentro who are either concerned with sales or involved in the management of the company. Incentro is a relatively young company with young employees. In general the people using the dashboard are in their thirties or forties. They are also people who use the Salesforce CRM tool on a daily basis, so they are familiar with the information used for the dashboard. Despite the fact that they are involved in the sales of BI solutions, they have limited technological knowledge about those solutions.

### 4.2 Interviews

To determine the functional requirements of the dashboard, the information needs of the end-users have to be taken into account. To map those information needs a few of the intended end-users were interviewed. The conversations were about Salesforce, including the day-to-day use of the CRM tool and most importantly, what the interviewees expect to accomplish with a dashboard.

This round of interviews was composed of short informal conversations, to map the fundamental functionalities of the dashboard. It was not the intention to give an extensive list of desired functionality from the users. The goal is not to develop a fully functional dashboard, but to answer the questions posed in this thesis. That is why only the most basic functionalities will be implemented.

One of the most important points mentioned during these interviews was that the people at Incentro would like to see the sales information set out against time. Not only historic data is important but the trends for the future would be a great asset as well. This is closely related to the concept of the **sales funnel**. This sales funnel is a representation of the current sales activities. It is calculated from the information in Salesforce and is the sum of the weighted revenues of open opportunities. The weighted revenue of an opportunity is in turn calculated by multiplying the estimated revenue with the probability of the opportunity being won. As such, the sales funnel is an indicator of future sales. The interviewees indicated that this would be a very important functionality of a CRM dashboard.

Another desired functionality was the ability to compare the funnels of separate offices at Incentro as well as those of the sales managers. Moreover, the managers would like to see which business sectors the clients are from.

## 4.3 Use Cases

With the information gained from the interviews held with sales managers working for Incentro, a few use cases have been specified. These use cases have been based on a persona. This persona is a fictional character representing the end-users and will be discussed in the following section. Afterwards the use cases are discussed.

### 4.3.1 Persona

Dylan Peterson is a 32 year old sales person, who has been working at Incentro for two and a half years. Recently Incentro has been using Salesforce.com as a Customer Relationship Management tool. This tool allows users to gain insights into their sales figures, however it only gives a snapshot of the current situation. Using the new tool he hopes to get more insight into the sales information he and his colleagues have been putting into their CRM tool.

#### Use Case 1: Viewing sales funnels

1. When Dylan steps up to the touchscreen the system is already opened full screen.
2. He wants to see how his sales funnel has been progressing over the years.
3. On the screen he selects the following: Sales Funnel, His name, 2011
4. He presses the select button.
5. Now the screen shows the value of his funnel, over the course of 2011.
6. Because he is very competitive with his co-workers he wants to compare notes.
7. Dylan taps on his name after which a list appears
8. On that list he selects account (representing all accounts)
9. Now the view changes to show all amount of sales for his colleagues over 2011

**Use Case 2: Show sales amount over the last year**

1. Dylan steps up to the screen the system is already opened full screen. He wants to view his sales figures in the last year.
2. On the screen a few lists are shown which represent aspects of sales.
3. From the lists he selects a few items: sales amount, 2011 and field of expertise
4. Once he has chosen the criteria he is interested in, he clicks on the selection button, which will take him to the visualization.
5. Based on the selection Dylan has made the, information is displayed on the screen showing him the sales amount in 2011 set out against field of expertise

Note: The systems requires the selection of one measure and at least one dimension

**Use Case 3: Selecting sales amount in hospitals information**

Dylan sees a lot of information on the screen and something draws his attention: One of the values shown is very high compared to the others. To find out more about that specific value he selects it by tapping on it. The system now shows more detailed information about the selected item.

**Use Case 4: More detailed information about government clients**

1. This use case requires the steps of use case 1 to be completed
2. Dylan is viewing the selected information on the screen e.g. sales amount over 2011 with the customer type. He notices that the value associated with the government type, is very high compared to other values. He wants to know a bit more about this.
3. To get a more detailed view of the information he uses a pinch gesture
4. The interface now zooms into the item only if there is more detailed information available (e.g. more levels in the hierarchy)
5. Dylan now sees that in the fourth quarter of last year there was a large sale to a government agency

**Use Case 5: Changing the current view**

This use case requires use case 1 to be completed Dylan is viewing the selected information on the screen

1. There are now significant outliers regarding sales amounts amongst the customer types. He now wants to see his sales figures from 2011 but this time set out against region. maybe there are some outliers there
2. Dylan taps on the axis which represents the customer type

3. A list appears, with other values. Of these values he selects "region"
4. The display changes so the new information can be seen (sales amount from 2011 set out against region).

## 4.4 Conclusion

In this chapter the process of mapping the information needs of the envisioned end-users have been described. These end-users are sales managers working at Incentro. The first step in determining these needs was to analyze the intended end-users. These end-users are sales managers at Incentro. The next step was to map the information needs they have. This was done using informal interviews. The analysis and interviews have been materialized in the shape of use cases. These use cases only cover basic functionalities of the dashboard, because the focus of this project is on gesture interaction. Nevertheless these use cases will be used to evaluate the design and the implementation.



## Chapter 5

---

# OnLine Analytical Processing

Now that we have described the situation at the start of the project and the desired situation, the following chapters of this thesis report will be focused on bridging the gap between the two.

Part of the final product is the provision of data, which is stored in a Data Warehouse. However a data warehouse is optimized for data storage and these optimizations make a DWH less efficient for providing data. This is where OnLine Analytical Processing (OLAP) technology comes in. To give an insight into OLAP technology first a high level description will be given in Section 5.1. After this a more detailed description is given on the subjects the most relevant for this thesis. This will be a discussion on providing the information as well as the structure of the data provided. This is followed by an explanation of how OLAP information is retrieved, including a short introduction to the query language MDX. This is followed by a discussion on technical limitations in relation to the information need discussed in Chapter 4. This chapter will end with a concluding section.

### 5.1 OnLine Analytical Processing Basics

OLAP is a database technology which provides access to multidimensional data. It is a Business Intelligence technology used for analyses on high level information. These analyses are used for supporting decision making processes in organizations [20]. The term OLAP was derived from the concept of OnLine Transaction Processing (OLTP) which is a name used for traditional database technologies.

From a technical point of view OLAP technology provides a communication layer between front-end applications and the back-end BI storage. This back-end can contain information stored using multidimensional data structures or using relational databases [7]. If the back-end is a relational database the term *Relation OnLine Analytical Processing* (ROLAP) is often used. For this project the information source will be the relational Data Mart described in Section 3.2. This means that the *Relational* version of OLAP technologies will be employed. Because this project is mostly concerned with the front-end and not the storage, the more collective term OLAP will be used.

## 5.2 Serving OLAP data

The Salesforce information in the Incentro Data Warehouse, is made accessible by using a Data Mart. This DM is a relational database and to make this database available using OLAP technologies, a ROLAP server needs to be used. For this project the Mondrian server [22] will be used. Mondrian was chosen because it was one of the few OLAP servers which had both support for relational databases as well as being open-source and free. Another reason is that the Mondrian server is also used by Incentro, so if it is needed, this in-house expertise can be utilized.

Before the Mondrian Server can be used, a schema needs to be created. A schema defines a multidimensional database and contains a logical model, consisting of cubes, hierarchies, and members. This schema also maps the logical model onto the source of the data, which is often a star-schema[21]. In the case of the dashboard, the source is the DM described in Section 3.3. A general structure of a schema is shown in Figure 5.1.

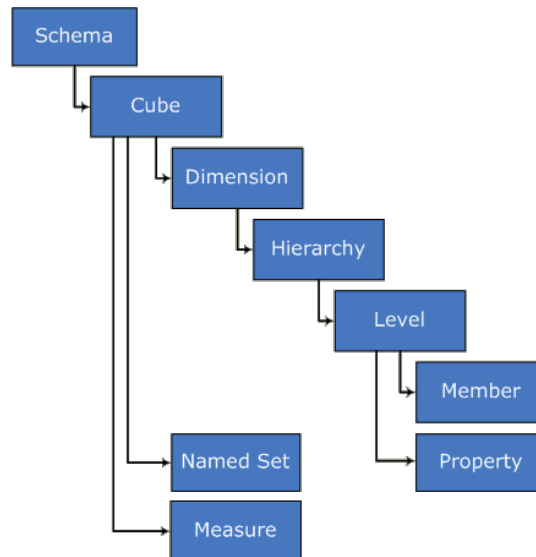


Figure 5.1: General structure of a schema

A schema is built up from one or more cubes. Usually one cube is made up of data related to a certain process. A cube can have two different types of information, namely measures and dimensions. Measures are numeric facts and dimensions are used to describe certain aspects of a measure. An example of a numeric fact would be the value of a sale. Additional information such as the time stamp and the person who closed the sale are examples of information which can be used in dimensions.

The dimensions within an OLAP cube are organized in a hierarchy. Every layer in this hierarchy is called a level, for example a time hierarchy would have the level **Year**, **Quarter** and **Month**. Another example would be a location hierarchy with **Region**, **Country**, **City** as levels. Depending on the demands of the users and available data, other levels could be left out or added. A single entity within the hierarchy is called a member and each member can have zero or more children. In the example of the time dimension, **2011** is a possible member of the level **Year**. And the **2011**

member will have 4 children members representing the four quarters in a year.

In this project the source of information for the Mondrian server will be the DM discussed in Section 3.3 of which more details can be found in Appendix A. The schema itself is an XML file, created using the Pentaho Schema Workbench. The resulting schema consists of one cube called Salesforce, with the following measures:

- Hoeveelheid Opportunities (Number of Opportunities)
- Som van gewogen omzet (Sum of weighted revenue)

These measures are the value which will be available to the end-users and are derived from the fact tables in the DM. However these measures are of little use if people cannot set them out against dimensions. These dimensions are derived from the dimension tables in the DM. The data available allows the following OLAP dimensions to be made:

- Fase (Stage)
- Leadbron (Lead source)
- Sales Manager
- Sluitingsdatum (Closing Date)
- Tijd (Time)
- Vakgebied (Area of Expertise)

All of these dimensions contain a single hierarchy. These hierarchies are comprised of one or more levels. Unfortunately the data on the DM does not allow for much depth in the hierarchies. For more information on the hierarchies and the available levels, a more detailed overview of the schema can be found in Appendix B.

Note that because all the dimensions are made up of one hierarchy these terms will be used interchangeably from now on.

## 5.3 Getting OLAP data

The previous section showed how OLAP information can be made accessible on a server. The remainder of this chapter will discuss how the provided data can be accessed from a client point of view.

### 5.3.1 MultiDimensional eXpression

MDX is a language used for querying OLAP. It was introduced in 1997 by Microsoft and since then has become the de facto standard. It resembles the SQL language, but it is aimed at multidimensional queries [19]. To give a better idea of the possibilities of MDX a few examples will be given. These examples will use the hierarchies shown in Figure 5.2 and 5.3. Figure 5.2 shows the Regions hierarchy from a Salesforce cube.

Only the top Level is displayed. In Figure 5.3 the time hierarchy belonging to the same cube is shown. The top two levels of this hierarchy representing the years and months are shown as well. To reduce clutter only the child members of the 2010 year member are shown.

Note that these hierarchies are only used as examples and have little to do with the sales information used for the dashboard.

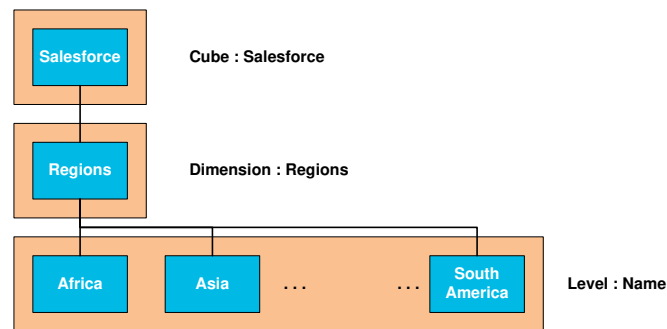


Figure 5.2: Example region hierarchy

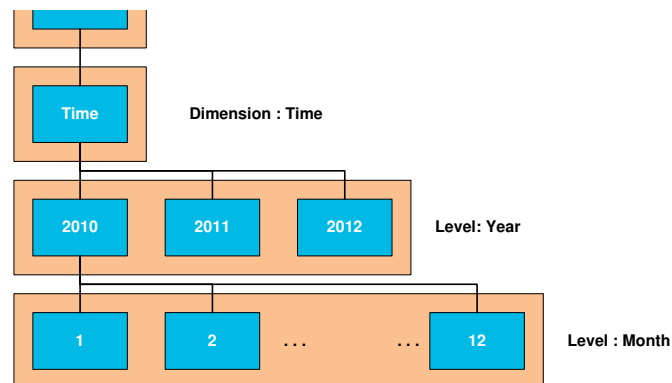


Figure 5.3: Example time hierarchy

With the hierarchies defined in the figures above, one of the more basic queries can be found below. Note that the reserved words COLUMNS, ROWS and PAGES in the query represent axes but they can be replaced by numbers.

This query will return one value namely the number of sales in the year 2010 in the Asia region. But the goal of OLAP and MDX is to get a good overview of the data. If only one value is displayed for every query it would be difficult for users to get that overview. Fortunately it is possible to use sets in MDX queries

The result of this query will be comprised of two values: the sales amount of 2010 in Asia, and the amount for the same year, but for the Europe region. It is possible

**Listing 5.1: Basic MDX Query**

```
SELECT
  [Measures].[Sales Amount] ON COLUMNS
  [Time].[2010] ON ROWS,
  [Regions].[Asia] ON PAGES
FROM Sales
```

**Listing 5.2: Using a set in MDX**

```
SELECT
  [Measures].[Sales Amount] ON COLUMNS
  [Time].[2010] ON ROWS,
  {[Regions].[Asia],[Regions].[Europe]} ON PAGES
FROM Sales
```

to use sets to query all the descendants of a given member, but MDX also provides functions which can do the same with less effort.

**MDX Functions**

Just like the SQL language, MDX provides functions for operations [17]. The most commonly used functions are those called *Members* and *Children*. The *Children* function returns the direct descending members of either a single member or those of a hierarchy. The *Members* function does something similar but instead of only returning direct descendants, this function returns all the descendants of a member (or hierarchy). These functions are appended to the end of a member as can be seen below.

**Listing 5.3: Example of MDX functions**

```
SELECT
  [Measures].[Sales Amount] ON 0
  [Time].[2010].Children ON 1,
  [Regions].Members ON 2
FROM Sales
```

This query will return all the children of the [2010] member on axis 1, namely months 1-12. On axis 2 all the members of the [Regions] hierarchy are returned : Africa - South America. For every combination the number of sales is returned but only if this information is available, which is not always the case. For example if a company has only been active in Asia since 2011, no sales records will be available for the year 2010 from that region.

**5.3.2 XML for Analysis**

Using a query language is only part of acquiring data. The data also needs to be communicated between different machines. This is where XMLA comes in. XML

for Analysis (XMLA) provides a communication layer between OLAP applications, which is based on web-standards such as XML, SOAP and HTTP [38]. XMLA consists of two SOAP methods : Discover and Execute. The Discover method allows the retrieval of meta-data about the OLAP cubes. The Execute method requires a query string and will return the requested OLAP data.

## 5.4 Limitation of the current solution

During the interviews conducted in Section 4.2, the subjects indicated they would like to be able to divide their customers into types (e.g. government). Unfortunately this information is not tracked by the CRM tool and is therefore not available for the dashboard. Another point the interviewees felt strongly about, was the ability to see historic trends as well as predictions for the future. However, as mentioned in Chapter 3 the data-set used for the dashboard may not be up to date with the most recent information. This means that prognoses for the future will be very unreliable. This reliability is only increased because there are issues with supplying the sales funnel information.

Chapter 4 discussed the concept of the sales funnel in the Salesforce CRM. From an end-user's perspective this is an important aspect of the sales, as it is an indicator of future sales. Unfortunately there are a few difficulties with making the sales funnel available. First of all, it is not directly available from fact tables and can therefore not be queried directly. This is because it is calculated from a fact and a dimension, respectively the sales amount and the probability. Furthermore the funnel is also a snapshot of a given moment. Because of these factors it is unknown whether the sales funnel can be queried using the MDX language. And if it is possible it would add a level of complexity to the dashboard, increasing the time frame for the project.

A possible workaround would be to update the DWH and DM every day, but even then there are some challenges. Continuously refreshing the information would provide an accurate sales funnel for every date on a time hierarchy but for higher levels it would need to be averaged out. It is unclear how this would affect the accuracy of the values. Another point is that updating the information is out of the scope of this project.

Due to time constraints and to avoid scope creep, the choice was made to remove the sales funnel from the requirements for the dashboard. But fortunately it can be approximated. According to one end-user, the funnel can be estimated by showing the weighted revenue against their closing date.

## 5.5 Conclusion

In this chapter the concept of OnLine Analytical Processing was introduced as a technology facilitating high level data analytics. This chapter also discussed the configuration of a Mondrian server to provide the sales information to the dashboard. This server interfaces with the Data Mart created by Incentro. The Mondrian server provides information in the form of OLAP cubes, which are made up from measures and dimensions. The measures are numeric facts and the dimensions describe properties of those facts.

This data can be queried by using the MDX language, which is the de facto query language for OLAP data. This language provides functions, some of which will be used for the implementation. To transmit the queries between the dashboard and the server XMLA will be used as a communication layer.

The setup of the OLAP server did bring some issues concerning the sales funnel. During the interviews described in Chapter 4, the end-users indicated that this was an important aspect of Salesforce, and therefore an important part of the dashboard. Unfortunately this funnel cannot be made available during the allotted time, but it can be approximated.





## Chapter 6

---

# Design of the Dashboard

In the previous chapters, the situation at the start of the project was mapped, as well as the goal of this project. This chapter will discuss the design of a dashboard application which will attempt to bridge the gap between the two. This design will be based on literature discussed in Chapter 2 as well as the requirements imposed by the use cases presented in Section 4.3 and the users themselves. These use cases only cover a small part of a fully functional dashboard and this will also be true for the design.

The design of the application will be comprised of two parts, which depend on each other: the visualization and the touch interaction. In Section 6.1 the visualization will be presented, which is the medium over which the OLAP information is shown to the user. The touch interaction will allow the end-user to navigate through this information and will be discussed in Section 6.2. This chapter will end with a concluding section.

### 6.1 Visualization

The following section will discuss the design of the visualization. First the requirements are discussed after which the proposal for the design will be presented.

#### 6.1.1 Requirements for Designing

In Chapter 2 many possibilities were discussed for visualizing multidimensional information. Arguably the most appropriate one for this project would be parallel sets. But the visualization should facilitate the use cases defined in Chapter 4. There are also a few other aspects which needs to be taken into consideration.

#### Users

For the visualization the users and their needs, require attention. As mentioned in Chapter 4 these users will be sales managers working at Incentro. Even though these users are very familiar with the information that is used, in general they do not have much experience with using Business Intelligence solutions or OLAP information. That is why the aim is to make things as intuitive as possible.

## Data

A second aspect to be considered is the data. It is comprised of different fact on different dimensions. These dimensions are hierarchical by nature. These aspects will affect the possibilities of the visualization.

### 6.1.2 Design

The literature discussed in the previous sections provides many different types of visualizations. These vary from very basic to rather complex ones. For the visualization, the end-users need to be taken into account. They will have little knowledge about OLAP or associated visualizations and because of that a very simple visualization is proposed.

Initially there was an idea of visualizing OLAP cubes, as real cubes in a three-dimensional environment. However using three-dimensional visualizations does have its drawbacks. One of the main issues is that of occlusion, which means some objects will be blocked from view by other objects [18]. That is why the decision was made to "flatten" the cube, which resulted in a grid.

Because a grid is two-dimensional, two positional values can be used as an information channel. These positional values of x and y, will be used to differentiate between members of a dimension. This will plot two dimensions against each other along an x and y axis. In general values along the x and y axes are used to show quantitative data, e.g. every centimeter x increases a certain value. However with OLAP the data is not quantitative because a dimension is made up from members representing entities. Therefore the names of the members should be shown along the axes.

Because the  $x$  and  $y$  positions are already chosen to convey information regarding dimensions and members, another information channel is required to represent values of the measures. Besides position, Munzner discusses the possibility of using other channels to visualize information such as color, shape, tilt, and area in [27]. Examples of these channels are provided by Stolte [28] and is shown in Figure 6.1.






property	marks	ordinal/nominal mapping	quantitative mapping
shape	glyph	○ □ + △ S U	
size	rectangle, circle, glyph, text		
orientation	rectangle, line, text	- / \   \ / -	
color	rectangle, circle, line, glyph, y-bar, x-bar, text, gantt bar		

Figure 6.1: Examples of possible visualization channels [28]

Shape is most often used to differentiate between the individual groups of categorical information. If shapes are used to show sequential information every different value will have to be assigned a separate shape. This will make it harder for users to see the difference between them. Size is also has its drawbacks, since it carries

the same risk as using a three-dimensional visualization, namely that of occlusion. It would be possible that smaller shapes are removed from view by bigger items. That is why color is chosen to encode the value information.

For the measure values a quantitative mapping is appropriate. When this is the case, Figure 6.1 suggest using saturation to encode the values. But Brewer [4] promotes using color lightness to show quantitative data or sequential data (as she calls it). For this design the color lightness will be used. A darker color will represent a higher value, while lighter colors will represent lower values. This principle will also be used in the grid resulting in something very similar to heat map. To give an idea of the design a paper sketch of the design is presented in Figure 6.2. On the left and bottom of this sketch hierarchies are drawn to illustrate where the members are in their hierarchies.

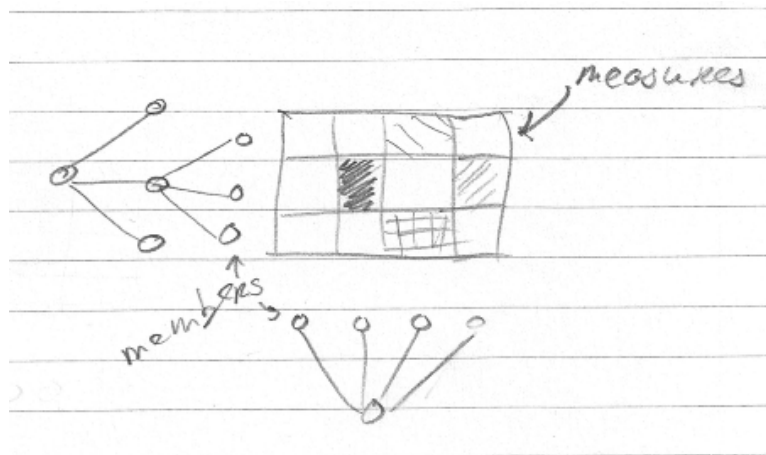


Figure 6.2: Early sketch for the prototype

The heat map design uses three different channels to convey information. These channels are the x-position and y-position and color lightness. But there will be some situations where the x or y axis will only hold one value or member e.g. after a slice operation. In this case one of the position channels will be carry no meaning and this position will be used to emphasize the value. This will result in a bar chart, either horizontally or vertically. A sketch of a vertical bar chart is shown in 6.3.

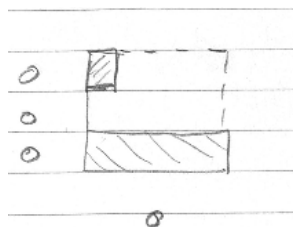


Figure 6.3: Sketch of a vertical bar chart

## 6.2 Touch Interaction Design

Now that the design for the visualization has been determined, the next step is to design a touch interaction for that visualization.

### 6.2.1 Hardware

The touchscreen which will be used for the development of the dashboard is made by PQLabs. It is able to detect up to thirty-two touch points. Multi-touch gestures should therefore not be a problem. Because of the size of the touchscreen it is possible to have multiple users at the same time, but this is outside of the scope of this project. The interaction will be designed for single-user interaction.

### 6.2.2 Touch Interactions

In Chapter 2 it was mentioned that there is a widespread use of touch enabled devices. These devices also have very similar touch interactions such as tap, swipe and (un)pinch. Because of these two factors, the assumption is made that most people (including the targeted end-users) have already worked with this gesture set. Because of their previous experience with this gesture set, people may have grown accustomed to these gestures. Therefore the navigation for the dashboard should not deviate too much from these existing gestures to allow easy adjustment to the navigation.

### 6.2.3 Desired navigation functionality

An important part of the navigation is being able to switch between the different dimensions of an OLAP cube. This will not be done by using gestures but by selecting options from a drop-down menu. The navigation through the hierarchies, contained in a dimension, will be done by the gestures. Which actions can or should be performed with gestures is determined by taking the most common OLAP operations and assigning gestures to them. These operations are called slicing, dicing, drilling up and drilling down [40]. With these operations it is possible to navigate through an entire hierarchy.

### 6.2.4 Assigned gestures

In the remainder of this section a proposal is discussed for gesture navigation through the OLAP data. The gestures are designed to work on the visualization described in Section 6.1. For the navigation through the information, the OLAP operations described in subsection 6.2.3 are used. Below the results of these OLAP operations are discussed as well as the gestures assigned to them. Only the gestures defined for the horizontal axis will be discussed. For the y-axis the gestures need to be rotated by 90 degrees.

#### Slicing

When an OLAP cube is sliced, only a single member within one of the hierarchies is chosen to be shown. For the dashboard the operation of slicing is done by performing

a tap on the desired member in one of the axes. On the left side of Figure 6.4 a slice gesture is performed on the fifth member on the x axis. This is done by tapping on the name of that member. The result is that only this member will be shown on the x-axis. Because there is only one member this visualization will be a horizontal bar-chart. The result can be found on the right side of the figure.

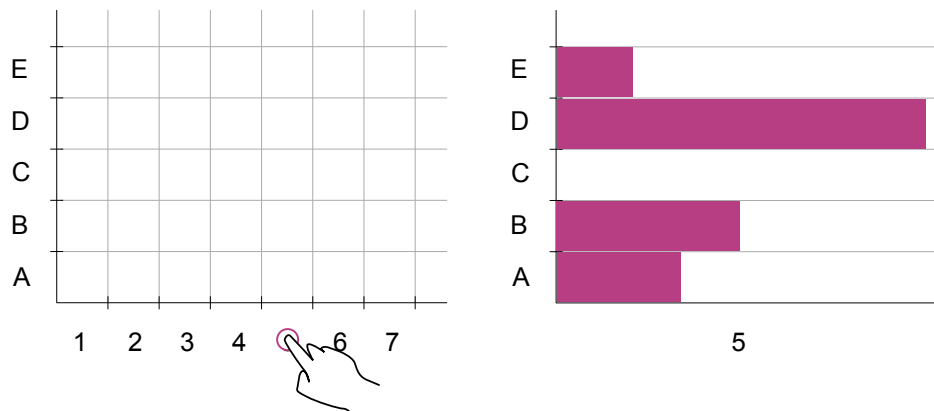


Figure 6.4: Gesture for slicing and the result

### Dicing

When an OLAP cube is diced, a sub-selection of one of the dimensions of the cube is made. This sub-selection can be defined by a range of members. To select the starting and ending member for a dice a multi-touch gesture is used. The gesture will be similar to the already established gestures discussed Section 2.2. Unpinching will "zoom in" on the selected data and pinching will zoom out of a dice. In Figure 6.5 a dicing gesture is performed on the visualization using an un-pinch gesture.

The members that were touched at the start of the swipes will be used to determine the range of the new visualization. In this case these were the third and fifth member along the x-axis. The right side of the figure shows this visualization defined by those two members.

### Drilling

Arguably the most important actions for navigation through an OLAP cube are those of drilling up and drilling down. Drilling down will move the visualization down a hierarchy and doing so expose more detailed information. Drilling up does the reverse: the visualization is moved up the hierarchy and will show more aggregated data.

The gesture for drilling down will be a swipe down along the target member, across the axis line. An example of this gesture can be found on the left side of Figure 6.6. This gesture is performed on the fourth member along that axis. This will move the visualization down the hierarchy and the result is shown on the right side of that Figure. Of course this is only the case if that member has child members.

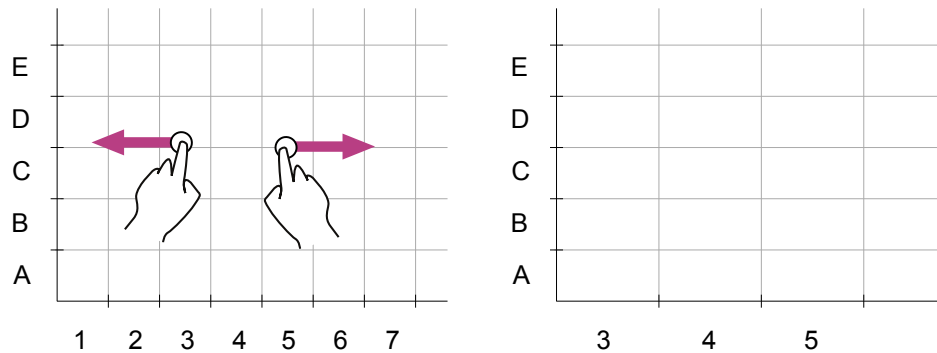


Figure 6.5: Gesture for dicing and the result

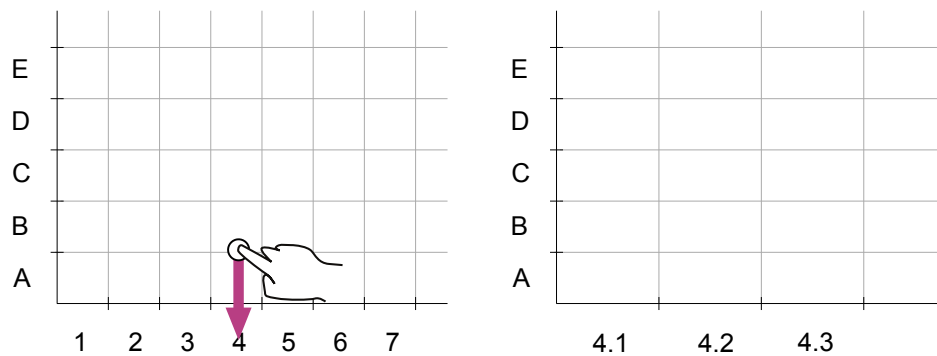


Figure 6.6: Gesture for drilling down

The gesture for drilling up will be the reverse: a swipe up across the axis line. This gesture is shown in Figure 6.7. The gesture is performed on the second member of the x-axis though performing this gesture on any member will result in same action. In this case the action is moving to the top level of the hierarchy. Because the top level has only one member a bar-chart will be shown. The resulting visualization is shown on the right side of the figure.

By defining the gestures for drilling up and down in such a manner, the movement away from the center is chosen as the gesture for "zooming in" on the data, and the movement towards the center will "zoom out" of the data. This is analogous to the pinch gestures discussed in Section 2.2.

The gestures described above correspond with the information shown along the x-axis. The gestures can also be applied to the y-axis, but then the gestures need to be

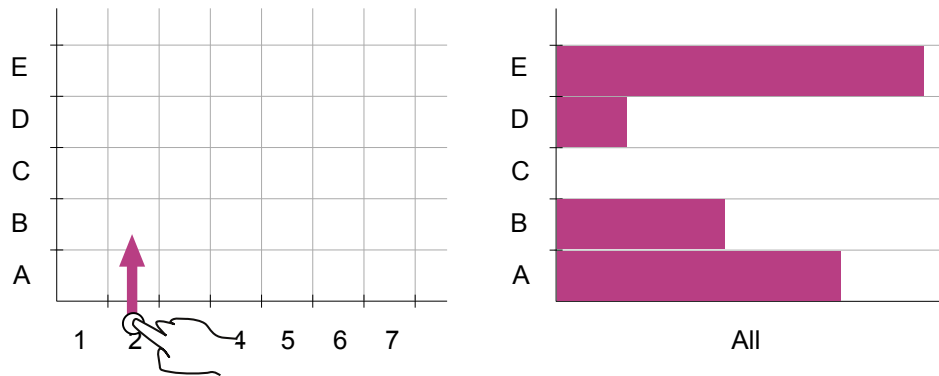


Figure 6.7: Gesture for drilling up

rotated 90 degrees clockwise.

### 6.3 Conclusion

This chapter presented the design of a touch enabled dashboard. This design was based on literature and use cases described in previous chapters. Because the use cases only covered a subset of functions, this is also true for the design. The design is comprised of a visualization combined with a touch interaction. In the visualization the x and y coordinates are used to plot two different dimensions. To convey the value, color lightness will be used, with white representing low values and darker color expressing higher values. The displayed dimensions will be selected using menus. But navigation through the hierarchies of dimensions will be done using touch gestures. These gestures are inspired by existing OLAP operations and will allow a user to navigation through the whole hierarchy of the selected dimensions.





## Chapter 7

---

# Implementation

The following chapter will discuss the implementation of the dashboard. This implementation is based on the design presented in the previous chapter. This design only covered limited functionalities of a dashboard, which is why the resulting implementation will lack some features. The resulting implementation should therefore be regarded as a prototype. However this should not be an issue since the goal of this prototype is to evaluate the touch interaction with the dashboard.

This chapter will begin with the technical design of the dashboard in Section 7.1. This will be followed by the challenges which will be faced during the implementation of the system. In Section 7.3 the retrieval of the OLAP information is discussed. How this information is visualized will be covered in the following section. Section 7.5 discusses the implementation of the touch navigation. After this some issues with the original design are described in Section 7.6. Some additions to the original design are presented as well. The chapter will end with a concluding summary.

### 7.1 Technical Design of the Dashboard

As mentioned in Chapter 3 there are a few constraints applying to the dashboard project. The use of a DWH, DM and Mondrian means that many aspects regarding the software design on the data side have already been realized. The assignment also requires the use of data visualization combined with touch interaction. This means the general software architecture is fixed and will look something like Figure 7.1.

#### Developing a web based dashboard

Many of the software components are already predefined, but there is also a hardware constraint which needs to be taken into account. Because the overlay only supports desktop platforms, the dashboard can only be developed on a Windows, OS X or Linux computer. An example would be to develop a desktop application for the Windows operating system, using .NET. But this would mean that this program would only run on Windows machines.

It is possible to use Java for the development of a dashboard which could be used for cross platform deployment. However, in recent years the W3C has been working on a Touch Events draft for the new HTML5 specification [34]. Using these touch events it would not only be possible to develop a dashboard which could work on

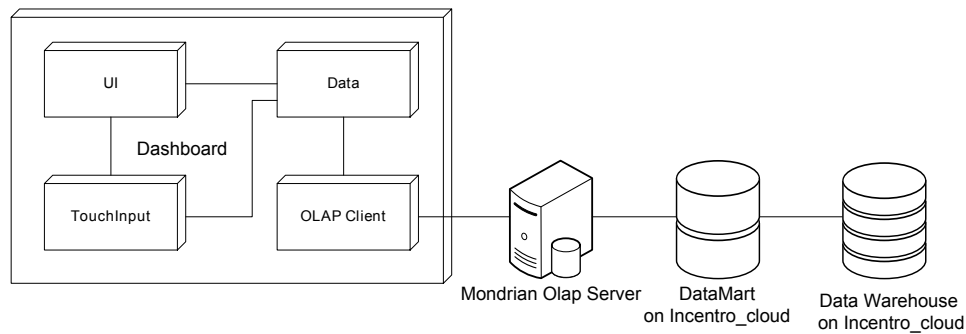


Figure 7.1: Overview of the technical software design

different desktop devices, but one that is usable on mobile touch devices supporting these events as well. That is why the choice was made to implement the dashboard as a touch enabled website. Creating such a website would have the following benefits:

- Touch enabled devices are becoming more and more ubiquitous but they probably will never fully replace desktops. If the dashboard is implemented as a web application the dashboard will serve as a single point of access for these different platforms. And developing the dashboard as such can reveal new insights.
- Making use of the latest HTML5 touch specification will make it an innovative product. Or at least the use of touch input will distinguish the dashboard from other applications.
- A web application is hosted on-line, so no client application is installed locally. This means that it is much easier to update the software for all users.

Now that platform for the dashboard has been decided upon, the next point is determining how to build for that platform. There is always the option of building the design from scratch but this will take too much time. Fortunately there are often tools and libraries available already capable of handling part of the desired functionality. To avoid reinventing the wheel the choice was made to utilize those tools and combining them into a whole greater than the sum of the separate parts.

## 7.2 Challenges for the Implementation

The advantage of using existing tools for the development of the dashboard is that time will be saved. The down side of this approach is that getting the different components to work well together also takes time. Often the amount of time saved is greater than the extra costs. Another possible issue is that the libraries could have critical bugs, depending on their maturity. This in turn, could hinder the development and the workings of the dashboard.

Another challenge for the implementation is the use of the MDX query language. This language is the de facto standard for querying OLAP information. Despite this, there are few resources available for learning the query language.

Using the draft touch event specification of HTML5 could also provide some obstacles for the implementation. This specification is not finalized and it is difficult to predict which browsers support this draft. And if browsers do support the draft specification, the implementation could still differ from it, because the draft has been revised a few times. This deviation from the specification could lead to unexpected behavior or - if there is no support - no behavior at all.

## 7.3 Retrieving Information

The first step of developing the dashboard is retrieving the information from the OLAP server. This requires two parts: the communication with the server and the generation of the queries. The first part is handled by using a JavaScript library called `xmlla4js`. To generate the MDX queries a custom JavaScript object has been developed.

### 7.3.1 `xmlla4js`

For the communication with the OLAP server the `xmlla4js` library [37] is used. This library is fairly new and still contains a few critical bugs, one which came to light during the implementation. This problem is complicated because `xmlla4js` has only two developers working on it. This would mean that it would have taken quite a long time before this bug would have been fixed. Fortunately it was possible to bypass the problem using a workaround.

`Xmlla4js` is a JavaScript library that takes care of all the XMLA calls, so developers can focus on working with the information itself. Because it is basically a wrapper API for XMLA it provides the same methods as the XMLA specification: `Discover` and `Execute`. These will be discussed in more detail below.

#### **Discover**

Before a query can be executed the user needs to know which information is available. This is where the *discover* function comes in. This function calls the Mondrian OLAP server and returns meta-data, which is data about the data stored on the server.

The user needs to be able to select which measures and dimensions he wishes to see. So the *discover* function needs to be called twice - once for the measures and once for the dimensions. The meta-data that will be returned by the Mondrian server is based on the schema shown in Appendix B. With this information the items in the drop-down menus are populated, so users can choose which dimensions they want to see. This information is also used to determine the number of levels in the hierarchies, which will be used for the generation of MDX queries. Because the information on the server does not change during normal operation, this needs to be done only at the start of the application.

## Execute

The *execute* function is used to query the OLAP server. The function requires a MDX query strings as a parameter, which is wrapped into a XMLA request. The Mondrian server will then return the result using XMLA containing the requested information. The *xmlla4js* library then converts the result into a JavaScript object called a *Xmlla.Rowset*.

As mentioned before the *execute* functions takes a MDX query string as a parameter. How this string is generated will be discussed in the following section.

### 7.3.2 MDX Query object

For the generation of MDX queries a custom JavaScript object has been created to keep the logic for the queries centralized. The most important function of this object is to generate the query. This generation is based on certain parameters, which are discussed below.

In Section 6.2.3 it is mentioned that the user will be able to select dimensions from a drop-down menu. The dimensions selected by the users are stored in the MDX object and will be used on the axes of the MDX query. The user is able to navigate through these axes using gestures and every time a new dimension is selected the corresponding axis is reset using the new dimension.

For every dimension the number of levels in the hierarchies is stored in the MDX object. It also keeps track of the depth of the current query and whether or not the *Children* function - discussed in Section 5.3.1 - should be used. These variables are used to generate a new MDX query, following a gesture input. They are also used to check whether or not certain actions are possible. How this generation is performed, will be discussed in the following section.

### 7.3.3 MDX examples

To give a better idea of how the MDX object works, a few examples are given. These examples will use the [Sales Amount] measure as if it were chosen by the user. Furthermore the example hierarchies used in Section 5.3.1 are also used here as the hierarchies selected by the user.

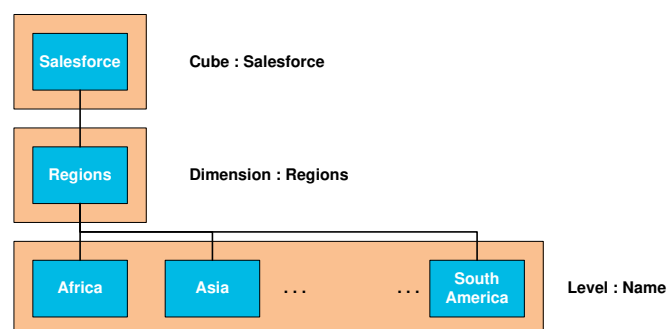


Figure 7.2: Example region hierarchy

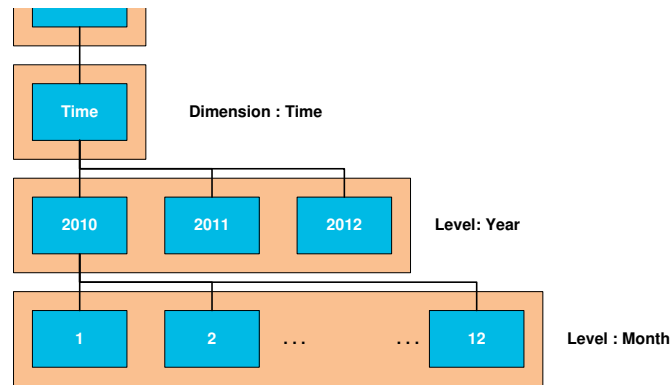


Figure 7.3: Example time hierarchy

When the dashboard is initialized the discover function is called using the xmla4js library. This will return meta-data concerning the information on this server. This information is used by the dashboard to determine the maximal depth of the different dimensions. For every dimension this is the same as the number of levels contained in the dimension. In this example the maximal depth for the regions and time dimensions would be 2 and 3 respectively. This is because the top level is counted as well.

After the user has selected a new dimension, the current depth is set to 1 and the Children value is set to true. Based on these values, the query shown in Listing 7.1 is executed. In the following sections some more MDX examples will be given to show what happens when the OLAP operations discussed in Section 6.2.3 are performed. Some of these examples will use Listing 7.1 as the starting situation.

#### Listing 7.1: Initial Query

```

SELECT
  [Measures].[Sales Amount] ON 0
  [Time].Children ON 1,
  [Regions].Children ON 2
FROM Sales
  
```

### Slice

Listing 7.2 shows what happens when a user slices along the time axes. For example the user only wants to see the information belonging to the year 2009. When he performs the required action the axes is changed. The flag for using the *Children* function will be negative and depth on the axis will be increased by 1. This result in the following query:

### Dice

A dice operation creates a sub-selection along a given axis. Again Listing 7.1 is used as a starting point. We are assuming that the children of the [Time] have the range

## Listing 7.2: Slice

```
SELECT
    [Measures].[Sales Amount] ON 0
    [Time].[2009] ON 1,
    [Regions].Children ON 2
FROM Sales
```

of [2007] up to [2013]. When the user dices along this axis and only wants to from [2009] to [2011], the MDX object is adjusted accordingly. The depth along the time axis stays the same but the *Children* flag is put to negative. The resulting query can be found in Listing 7.3.

## Listing 7.3: MDX Dice

```
SELECT
    [Measures].[Sales Amount] ON 0
    {[Time].[2009], [Time].[2010], [Time].[2011]} ON 1,
    [Regions].Children ON 2
FROM Sales
```

## Drilling

Drilling up and down is more complex than the slice and dice operations. The query generation by itself is not much more difficult, however there are a few cases requiring special attention. But first a regular situation is discussed. Again, for this example Listing 7.1 will be the precondition. Drilling down on the year [2010] the depth is incremented and the *Children* flag is activated (even though it already was). The resulting query is found in Listing 7.4.

## Listing 7.4: MDX drill down

```
SELECT
    [Measures].[Sales Amount] ON 0
    [Time].[2010].Children ON 1,
    [Regions].Children ON 2
FROM Sales
```

As mentioned in Section 6.2.4 drilling down is the reverse of drilling up. This is true for the MDX queries as well. A drill up action will decrease the depth of the code and activate the children flag. If Listing 7.4 is as the starting point, it will result in the situation described in Listing 7.1. So even though the algorithm is not the opposite, the result is.

There are a few cases in which the actions of drilling up and down require a different approach. This is the case when the level is either at the top or the bottom of a hierarchy.

If a drill up is done to the top of a hierarchy the depth will be 0 and *Children* flag will be set to false. If a drill down is performed to the maximally allowed depth this flag is also set to negative. Listing 7.5 shows what happens after a drill up on the Time dimension is done and a drill up on the Europe member of the Regions dimension is performed.

#### Listing 7.5: MDX drills

```
SELECT
  [Measures].[Sales Amount] ON 0
  [Time] ON 1,
  [Regions].[Europe] ON 2
FROM Sales
```

### 7.3.4 Post-processing

The result of a MDX query is a *Xmla.Rowset*. This is an object of the *xmla4js* library containing the result of the query. This structure of this object - and the information contained within - is defined by the *xmla4js* library. A *Xmla.Rowset* is an array containing *Cell* objects, comprised of a value and an index indicating the position within the result cube. This index is an ordinal number and needs to be translated into a multidimensional coordinate for the visualization. The only meta data included in a *Xmla.Rowset* is a string indicating the name of the queried measure. To make the data more usable additional information is added, such as the corresponding member names. The adjusted data is stored and will be the basis for the visualization.

## 7.4 Visualization

Now that it is clear how to retrieve the data from the OLAP server, the following step is visualizing that information. How this visualization will look like has already been defined in Chapter 6 and this section will discuss how this design has been implemented. First of all the rationale for the final choice of the library is given. After this a few technical details of the implementation will be described. This section will only mention a few points since most aspects of the visualization have already been presented in the previous chapter.

### 7.4.1 Choice for the visualization library

As mentioned before, the dashboard will be developed using tools which are capable of handling part of the functionality. There are many of visualization libraries available, but fortunately the choice for implementing the dashboard using web technologies has limited the possibilities. But even then many tools still remain viable.

The libraries considered included D3.js, Google Charts, Highcharts, Paper.js, Processing.js and Raphaël. Both Google Charts and Highcharts are focused on graphing like bar charts and pie charts. But because there is need for a more general visualization tool, these two are disqualified.

The four libraries left are very similar to each other considering their functionalities. Furthermore they all use web standards and work with different browsers which comply with those standards (so they might not be backwards compatible with some of them). To get a feel for the differences between them a few tutorials were done using those libraries. From those four libraries, d3.js stood out for its ease of using objects as a base for the visualization.

D3.js stands for Data-Driven Documents [9] which is a JavaScript library for manipulating web documents. It discloses the capabilities of standards such as HTML5, CSS3 and SVG. Additionally the d3.js library enables developers to easily and dynamically manipulate the content of a web page, which makes it suitable for rapid prototyping. D3.js is also designed to work with data as a base for the visualization. One example of this data orientation is the fact that objects can be used a data source. Another example is that there is very little need for iteration. It is possible to introduce an array of data to the library and the function calls are then executed on all the elements in the array. Because of this specialization with data, d3.js is chosen to draw the visualization.

#### 7.4.2 Technical Details

The visualization is based on information resulting from MDX queries that has undergone some post-processing. After post-processing the result is an array containing adjusted Cells. The structure of an adjusted Cell is shown in Figure 7.4. A Cell contains a value, the coordinates in multidimensional space and the member names associated with those coordinates. Note that this figure shows a simplified structure of the object.

Cell				
Value		5		
		coordinates		memberNames
	0	0	0	Opportunities
	1	3	1	2012
	2	5	2	Jan

Figure 7.4: Simplified structure of a Cell after post-processing

For every Cell in the array the d3.js is used to calculate the location in the visualization. On this location a rectangle is drawn with the color that represents the value of a Cell.

This color is determined using a function of the d3.js library that maps a domain of values onto a range of colors. The domain of numbers is derived from the lowest and highest value in the array of Cells. The range of the colors is a set by white and a darker color. This darker color has been provided by the ColorBrewer 2.0 [3] tool. Gray is used to represent cells with no values.



Because d3.js uses SVG to draw, the resulting visualization is a SVG image. SVG is based on XML and the shapes in the visualization are similar to XML elements. These elements can be used to add meta-data to the visualization. In this case the value of the measure and the member names of the Cell are stored in the XML elements. This meta-data can then be used to determine on which element gestures are started and ended. This information is in turn used to provide the parameters for OLAP operations. Figure 7.5 and 7.6 show a few screen captures of an early prototype.

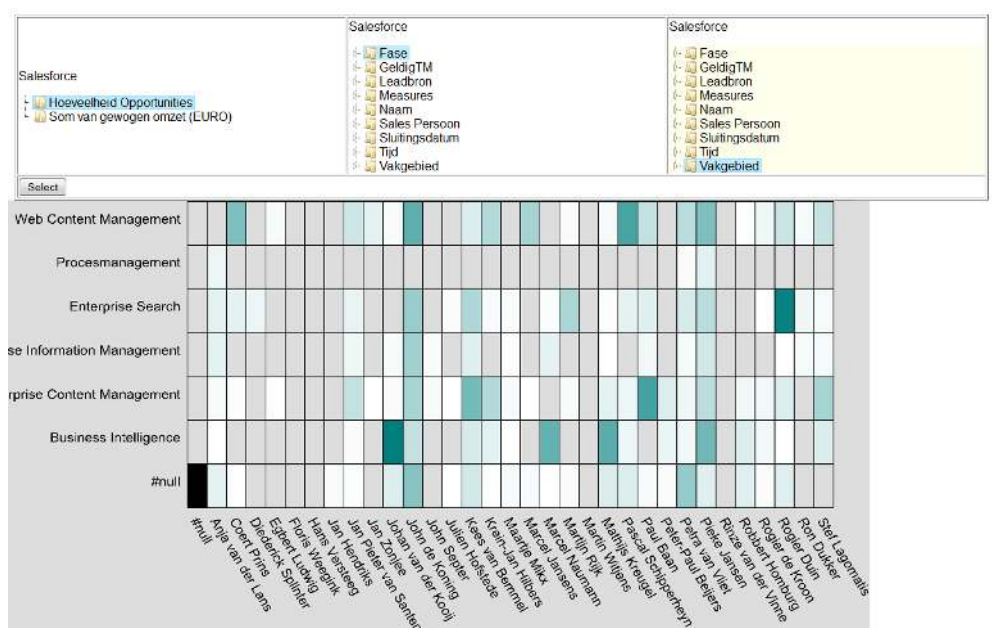


Figure 7.5: Visualization prototype



Figure 7.6: Horizontal and vertical bars from an early prototype

## 7.5 Touch Navigation

At the Incentro Utrecht office there is a multi-touch screen comprised of regular television with a PQLabs overlay. Though the overlay is not plug-and-play, it only requires the installation of drivers to get it working. With the drivers the overlay is compatible with Windows 7 native touch. For programs not supporting touches, windows 7 will emulate a regular mouse, but with software supporting touches (e.g. Google Earth)

the frame works without any extra effort, thanks to the native touch in the operating system.

### 7.5.1 Detecting touches in desktop browsers

Unfortunately there is very little support for detecting touches in desktop browsers. Initial testing with Firefox and Chrome only returned negative results for multi-touch detection and the browsers worked as if they were controlled by a mouse.

Next to compatibility with Windows native touch, PQLabs offers many different SDKs for the development of multi-touch application, including Java, C# and Adobe Flash. PQLabs also provides support for TUIO, an open framework which defines a protocol and an Application Programming Interface (API) for multi-touch surfaces, creating an abstraction layer for interactive surfaces. It allows touch information to be interchanged between detectors and software [31]. Of course this is only possible if the detector and software both are TUIO enabled. In this case the detector supports TUIO, but with a workaround it is possible for the software (i.e. browser) to support this as well.

The Mozilla Firefox does not support TUIO on its own, but by installing the *np-TuioClient* plugin, this support can be added to the browser. However there is still some more configuring required before the touch input can actually be used within a web page. To enable touch information to be passed through to the pages, there are a few JavaScript libraries which work with the npTuioClient. The two most commonly used ones are *magictouch.js* and *tuio.js*. These two libraries allow touch information to be passed on from the plug-in to the page in the browser. The *tuio.js* library has been written by the same author who created the npTuioClient plug-in. To prevent incompatibilities between the two it would seem logical to use this library. However the *magictouch.js* library has an advantage over *tuio.js*. *Magictouch.js* generates events based on a W3C proposal for touch events [34]. Even though it is only a draft at the time, it is likely that something similar to the current draft will be included in the final HTML5 specification. Also being a draft, the touch events may or may not be supported by different browser. For the purposed of this paper the plug in and JavaScript library both work in the Mozilla Firefox browser.

When the plug-in is installed in the browser, it only takes a few more steps to get the touches working on the page. In the header of a HTML page the *magictouch.js* library needs to be included. This can be done using the following line:

#### Listing 7.6: Referencing the multitouch library

```
<script type="text/javascript" src="magictouch/magictouch.js">
</script>
```

The final step is to add the following line to the body of the HTML file.

#### Listing 7.7: Adding the TUIO object to the page

```
<object id="tuio" type="application/x-tuio" style="width: 0px;
height: 0px;"/>
</object>
```

Using these steps it is possible to get touch information and pass it on to the browser and the page. How this information is handled there, will be discussed in the following section.

### 7.5.2 Touch recognition

The touch input from the touchscreen is passed through to the browser using the TUIO protocol. In the browser these touch inputs trigger events which can be linked to any desired functionality. The following section will describe the implementation of the gesture recognition.

#### Rationale for avoiding touch libraries

There are many different libraries available for gesture recognition on websites on touch enabled devices. The majority of these libraries also have support for gestures and multi-touch. That is why there should not be any need for custom implementation of touch detection for the dashboard. Unfortunately some difficulties arose during testing.

While testing several touch libraries including JQTouch, Sencha Touch, xui, and jQuery Mobile a few issues arose. The detection of taps went as planned, but when testing gesture recognition things did not go as expected. When using the libraries on a test web site touches were recognized but gestures such as swiping were not. When testing the same sites on an iPad both the touches and swipes were correctly recognized.

This could be explained by the fact that these libraries are focused on mobile devices or in devices running iOS. The *Safari* browser in the iOS operating system uses custom events for certain gestures interactions [14] and it could be that the tested libraries rely on those custom events. To avoid conflicts with libraries they were not used for the implementation of the dashboard and custom code was written for this functionality. How this was done will be described in the next sections.

#### Touch Events

The touch events that are used for registering the inputs of the user are currently being drafted by the W3C [34]. This means that these are not final, but because the draft has been around for a while, there are browsers do support them. Within the browser touch actions trigger events which are defined in the W3C recommendation. These events work exactly the same as the existing ones. There are four touch events which are:

- touchstart: fired when a new touch is detected.
- touchend: fired when a touch ends.
- touchmove: fired when a touch point moves.
- touchcancel: fired when a touch is interrupted.

In JavaScript a function can be bound to an event. When this is done and the event triggers, the associated function is called. The function is called with an argument

containing the data from the event. This data contains information about the event. But for the dashboard only the following information will be used:

- x: the x-coordinate of the touch.
- y: the y-coordinate of the touch.
- target: the target of the touch, meaning which element was touched.
- identifier: Every touch is given a unique id, stored in this variable.
- changedTouches: An array of the touches that triggered the event.
- touches: An array of all the touches, occurring at this moment.

The target of a touch is an element on the page. If this element is part of the visualization this target will also contain extra meta-data such as the measure value and member names. This data has been attached using D3.js

### Definitions

The goal is to use gestures to control the information that is shown. Before the technical details are discussed, a few definitions are given for clarification.

- Touch Path: A touch path is defined by two points: the start of a touch movement and the end of the start movement. For the sake of simplicity the real path of a touch movement is discarded.
- Gesture : For this project a gesture is comprised of one or more paths. To maintain simplicity while supporting multi-touch gestures, only one gesture can be registered at a time.

### 7.5.3 Gesture Algorithm

The algorithm for handling the gestures has been divided up into three subtasks. The first step is registering the touches, during which the touch information is stored in a gesture as a set of paths. After this the gesture is analyzed to see whether it matches one of the predefined gestures. If this is the case the corresponding action is performed.

#### Gesture registration

The registration of a new gesture only start when there are no touches on the screen. When, under that circumstance, a new touch is registered, a new gesture is initialized in the code. This gesture will then contain a new path, which has the starting position of the touch. When that same touch is removed from the screen, the coordinates of where the touch left the screen, are saved as the ending position of that path.

If a new touch enters the screen while a gesture is already in progress, a new path will be initialized. This path will then be added to the existing gesture. While there are still touch points on the screen, every new touch point will create a path which will consequently be added to the gesture, resulting in a multi-touch gesture.

A gesture is only completed when there are no more touches on the screen. This means that all the paths are stored in one object and are passed as one entity to be recognized. This approach means that "real time" gestures are not available, since gesture based events are only triggered when there are no more touches left.

### **Gesture Recognition and OLAP function**

A completed gesture is passed on to the recognition part, in which the all the paths contained in a gesture are analyzed. The input is compared with the gestures presented in Section 6.2. If a match is found, the corresponding function is called. These functions will change the parameters of the MDX object, which will result in a new query being executed. This, in turn, will return a new dataset that is used for a new visualization.

## **7.6 Issues and additions to the original design**

Software development is an iterative process and the first attempt is rarely a success. For the dashboard this was true as well. The early prototype shown in Figure 7.5 revealed several issues. Some of which were easier to correct than others.

One of the issues that arose was that of data quality. Often queries would return a *#null* member. This happens when certain fields are not available for entries in the DM. Because the information on the DM is outside the scope of this project this issue will not be fixed, but it is recommended that Incentro addresses this issue.

Another issue is that the dashboard only has a limited amount of space to visualize the information. When there are too many members along an axis, they will only receive a small portion of the available space. Too much information will be visible in a small space and the visualization will become overcrowded. Texts will become unreadable and the cells will become too small to see. This can be addressed by making the visualization scrollable or zoomable. The issue could also be resolved by structuring the hierarchies on the OLAP server in such a manner that the number of children below a member is limited. This problem has not been fixed due to time constraints.

A related concern is performance. When there is a large amount of members the visualization takes a long time to be rendered. Performance issues also occurred with the detection of touches. When multiple touches are used the system experiences a high load on its resources. This could be due to the number of events fired when the touches are detected. But this is not certain.

During the implementation a few extra parts of the visualization have been added that were not in the original design. The most important addition was a legend. With this legend the user is able to determine which values the different colors represent. Because the visualization can become either a horizontal or vertical bar chart, one legend has been added for the x-axis and one for the y-axis. To make the values even

clearer a pop-up has been added to every cell in the heat map. When a cell is touched a small window appears displaying the value that cell represents.

The rest of the additions are mostly cosmetic. An example of this is the inclusion of a start screen. Another functionality that has been added is a change of color hue when different measures are selected. This way the switch to another measure is more visible for the user. The final change is the removal of the selection list shown at the top of Figure 7.5. These have been replaced by drop down menus with a more mobile look and feel. The position of these menus has been changed so they are closer to their respective axis.

These changes have resulted in the dashboard of which screen captures can be found in Figure 7.7. In this figure the two legends can be seen as well as the drop down menus. Furthermore the pop has been enlarged for clarity. This version of the dashboard was used during the evaluation.

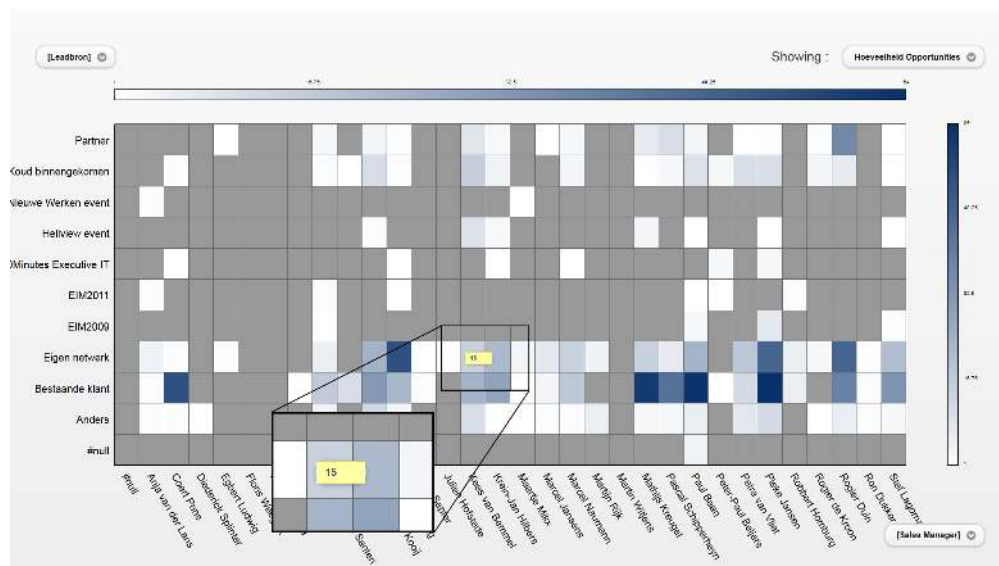


Figure 7.7: Screen capture of the prototype used for the evaluation

## 7.7 The dashboard on the iPad

At the start of the implementation the goal was set to implement the dashboard using web standards. One of the reasons was so it could possibly be used on mobile touch devices. Even though this is outside the scope of the project the prototype was tested with touch-enabled mobile devices. The first attempt was on mobile phones but the screens were too small to show the prototype correctly. However when the dashboard was opened on the iPad it rendered very closely to how the dashboard was rendered on the desktop.

Unfortunately there are a few bugs with the dashboard on mobile devices. For example not all the gestures work properly on iOS devices. This may have been caused by the custom codes for the gesture recognition that causes a conflict with the events

supported on the mobile Safari browser. Another discrepancy between the two platforms is that on the iPad the pop-ups do not seem to work.

## 7.8 Conclusion

In Chapter 6 a design for the dashboard was proposed. This chapter described the process of implementing a prototype dashboard based on this design. The choice was made to implement the dashboard as a touch enabled website, using the touch event specification for HTML5. This approach would enable cross-platform accessibility.

The implementation was done using existing libraries that already provided certain functionality. The retrieval of the OLAP data has been done using the xmla4js library and a custom MDX object which generates the queries. The results of these queries are then used by d3.js to generate the visualization.

The approach of using existing libraries was not possible for the touch detection. The tools available were mainly focused on touch events provided by iOS. Some of these events are not included in the HTML5 specification. To prevent conflicts a custom implementation was decided upon. This may have caused some of the gestures not to work on the iPad.

The choice to implement the touch-enabled dashboard has brought additional complexity to the development. Because the web application needed to work on a desktop environment, a desktop browser was needed with touch support. Because this was not provided by the major browsers, this functionality was added using a workaround.

With the implemented touch gestures users are able to edit the aforementioned MDX object. This in turn renews the displayed information. This is how users can navigate through the data from the OLAP server.

During the implementation of the dashboard, several technical issues surfaced that have not yet been resolved. The first concern is that of data quality. Often queries would return *#null* members indicating that some entries in the DM are missing fields. Another issue is that of performance. This concerns the rendering of the visualization as well as the detection of touches. Another concern is that the visualization has a limited amount of space to show all the information. When a large amount of members needs to be shown, the visualization can become overcrowded.

Because the choice was made to use the HTML5 Touch Event specification draft, the dashboard also works on some mobile touch enabled devices.





## Chapter 8

---

# Evaluation

In the previous chapters, the design and implementation of the dashboard have been described. In this chapter their evaluation will be discussed. The dashboard has been designed to fulfill requirements set up in Chapter 4. To see whether this is the case, the dashboard will be evaluated. Part of the evaluation will be based on the use cases described in that same chapter.

To give a better idea of the whole procedure first, in Section 8.1, the users who will test the dashboard are discussed. Secondly the testing methodology is described in section 8.2. Afterwards the results will be discussed of the user tests and questionnaires are discussed. Based on these results Section 8.4 will discuss options for improving the dashboard. This chapter will end with a concluding section.

### 8.1 Testers

The use cases described in Chapter 4 were inspired by the information needs of sales managers at Incentro. Because of that the best way to test the dashboard is using those same sales managers. These sales managers are already familiar with the information used for the dashboard.

### 8.2 Testing Methodology

The process of testing the dashboard is comprised of two different parts. The first part is a user test. This test follows a script, which is described in Section 8.2.1. During this test users are encouraged to think out loud. During the test the sound will be recorded. Following this procedure, the users are asked to fill in a questionnaire. This questionnaire is taken to get a quantitative measurement on how well the dashboard was received by the testers. This questionnaire is discussed in Section 8.2.2.

#### 8.2.1 User test script

The script for the user test is comprised of the following steps:

- In the first part of the test users are asked to use the dashboard without instructions while thinking out loud.

This is done to check whether design of the visualization and touch interaction is intuitive. This will also show what the opinions first time users will have of the dashboard.

- Explanation and Instructions

After the user has played around with the dashboard to their liking, an explanation is given about the dashboard and the hierarchical nature of the data. The user is also instructed on the use of the available gestures (slice, dice and drilling).

- Use Cases

With the provided knowledge users are then asked to perform a few use cases. The use cases are based on those discussed in previous chapters. Unfortunately not all the desired information was available in the Salesforce source data, which means there are some deviations from the use cases discussed in Chapter 4.

- Use Case 1

The user is asked to get information on the Sum of the weighted revenue, set out against the closing date and himself. If the user is not in the data ask him to pick a random user.

- Use Case 2

For this use case, the user asked to retrieve the following information. The number of opportunities which have the lead source of EIM2009 or EIM2011. This should be set out against the area of expertise of Enterprise Content Management.

- Use Case 3

The user is asked to get the number of opportunities which have been brought in by partners for all the areas of expertise.

- Finally the user is allowed to continue navigating the dashboard as he wishes. At the end the tester is also asked for any concluding remarks he might have.

With more knowledge about how the dashboard and the navigation work, the user is asked to use the dashboard. Hopefully the user will gain some new insights into the information. This also provides an opportunity to observe the tester interacting with the dashboard, when he has more experience with the gestures.

The duration on the user test varied with the different users, but in general the user tests took approximately thirty minutes.

### 8.2.2 Questionnaire

Filling in questionnaires after working with a system is a commonly used technique to assess usability of a software program. This method is tried and tested, which is why this approach is also used for evaluating the prototype dashboard.

Similar to the development of the prototype, it is possible to reuse existing tools for the evaluation of the dashboard as well. Using questionnaires for the evaluation of software is quite common and many existing questionnaires are available [1]. A few examples are the System Usability Scale (SUS) developed by Brooke [5] and the Questionnaire for User Interface Satisfaction (QUIS) presented by Chin et al. [8].

Unfortunately there are reasons why these questionnaires are not appropriate for testing the dashboard. The first concern is that the existing questionnaires are too general and do not handle aspects which are important for this project. Examples of such aspects are visualization and the gesture navigation. A second reason is that some questions are not applicable to the dashboard in its current form. Because it is still a prototype, features such as error and help messages are still missing. This is why a custom questionnaire was created for the testers of the dashboard.

Creating the custom questionnaire does not mean the established questionnaires should be ignored. The questions asked in those questionnaires can be used as a base for the questions posed about the dashboard. One example is intuitiveness of the gestures, which can be considered as a vague term. Instead of using this term the ease of learning will be asked. Other important aspects for this project the visualization, ease of use and of course the use cases. Different questions are posed regarding these features. These questions have been ordered in a chronological order similar to how the script for the user test was ordered. The choice was made to allow only four degrees of agreement, avoiding a neutral choice. The questionnaire - as well as all the answers - can be found in Appendix D.

## 8.3 Results

The following section will present matters that were discussed during the user tests. This will be followed by a selection of results from the questionnaire.

### 8.3.1 Results from the user tests

The user test discussed in the previous section, was performed by several sales managers of Incentro. The testers were encouraged to think out loud, and the audio was recorded. Excerpts of these recording can be found in Appendix C.

During the tests several issues were brought up by the users. Users noticed that the quality of the information could be improved. They also quickly realized that the information shown was either incomplete or outdated. The combination of these two issues could lead to users gaining insights that are not true. Furthermore the users seemed much more interested seeing future trends than historic data.

A feature that the users found lacking, was the ability to zoom to an even higher level of detail. Most users wanted to be able to see which opportunities are aggregated in a single Cell. Three of the testers also wanted to see summarized information along the edges, so they could get a total for all the members.

Another issue that was voiced by most the testers, was the use of gray in the visualization to indicate that there was no value. Furthermore the dashboard would sometimes crash while being handled by the users.

### Observations

There are also a few points which were not discussed with the testers but were observed during the evaluation. One notable point was that people sometimes did not know where they were in the hierarchies. This basically meant that they got lost in the visualization of the data. Another striking point that was noticeable during the user tests was the large variation in time it took to adapt to the use of gestures. This is especially interesting because of the uniformity of the user groups.

### 8.3.2 Results from the questionnaire

After the sales managers at Incentro had completed the user tests, they were asked to fill in the questionnaire found in Appendix D. This appendix also contains all the answers given by the testers, as well as the averages. The dataset is rather small and can therefore not be used to draw any significant conclusions [32]. However the given answers are indicative of how usable the prototype is. In the following section the most notable results are presented.

The users were critical of the level of information not being detailed enough. Unfortunately this is a limitation of OLAP technology, which is designed for high level analysis. Another point the testers commented on was that more dimensions would enhance the possibilities of the dashboard. This could be the result of the design which was focused on simplicity and doing so, sacrificed some functionality.

On a positive note, the testers found the use cases very easy to perform. The users also agreed on the fact that the dashboard invited them to play around with the information. In the end the average score given to the dashboard was 7.2 on a 1 to 10 scale.

## 8.4 Possible Improvements for the Dashboard

The user tests combined with the questionnaire revealed the following major points on which the dashboard can be improved upon:

- The information is not up to date and the data quality can be improved.

The information used for the dashboard is provided by a Data Mart set up by Incentro. This Data Mart is only a test environment and is rarely updated, as was mentioned in Section 3.3. A finished Data Mart will be updated more often and outdated information should not be a problem anymore. The issue of data quality still remains though and it is recommended that Incentro address this point of concern.

- The level of detail of the information was too high.

As mentioned before the detail level of the information is a direct result of the use of OLAP technology. Perhaps it would be possible to combine the relational database and ROLAP techniques to supply this information. But this is outside the scope of this project.

- There were not enough dimensions available.

The availability of the amount of dimensions is inherent to the design of the dashboard, which was made with simplicity in mind. Perhaps the use of parallel sets would have increased the amount of information shown, but this also would have increased the complexity of the dashboard.

- Users got "lost" in the information.

Preventing the users of getting lost can be accomplished by using several options. First of all, providing more information on where the displayed data is in the hierarchies. This could be done using breadcrumbs. Another option would be using animations so that users could see the visualization zooming in and out with the information.

- There is a large variation in the time spent learning the gestures.

The large variation in the learning time for the gestures, could be caused by the fact that the gestures are not intuitive. This also could have contributed to the problem of people getting lost in the information. Adjusting the gesture could make them easier to learn. This could also prevent users from getting lost. But this is something which requires more research.

## 8.5 Conclusion

With the implementation of the dashboard design completed, the next step is to evaluate the implementation. The evaluation has been done through user tests based on the use cases and requirements determined in Chapter 4. These tests were recorded and excerpts of these recordings can be found in Appendix C. To quantify the results the user were asked to fill in the questionnaire in Appendix D.

In general the opinions were positive but there were also points of critique. Users quickly discovered that the information was either outdated or incomplete. A second issue was that the testers were not satisfied with the level of detail of the information shown by the prototype. Another issue was that more axes or dimensions would be desirable for the visualization. Other points that became apparent during the tests, were that users would sometimes get lost in the information and that there was a large variation in learning time for the gestures. To end on a positive note, the users found the use cases easy to perform and they indicated that the dashboard invited them to play around and explore the information.



## Chapter 9

---

# Revised Dashboard

In the previous chapter the evaluation of the prototype is discussed. Based on this evaluation the dashboard has been revised by improving some aspects of the prototype. This chapter will discuss these improvements.

In section 9.1 the improvements to the visualization are discussed. After this new gestures will be discussed in Section 9.2. In the concluding section this chapter will be summarized.

### 9.1 Improvements to the Visualization

The evaluation of the prototype has presented several possible improvements for the dashboard. In this section improvements to the visualization will be discussed.

The first issue that has been fixed was a small bug pointed out by one of the testers. The scale along the legends showed decimal numbers, when this should not happen. Fortunately this was easily fixed.

The other improvement to the visualization was changing the color that represents "no value". The color used for this was gray, but users found it to be distracting. That color has been replaced by white. However because white is also used to represent the lowest of the value range, users will no longer be able to differentiate between them. Because of this the domain of possible colors has been changed. ColorBrewer 2.0 [3] was used to select darker colors for the minimal value.

The final adjustment that has been made is the reduction of the visualization size. This will benefit the rendering on tablet devices. A screen capture of the final prototype can be found in Figure 9.1.

### 9.2 New Gestures

One of the concerns that came to light during the evaluation was the gesture interaction. There was a large variation in learning time amongst the different testers. This could indicate that the gestures are not intuitive. In the following section a new gesture set for the navigation is proposed.

The number of functions has also been expanded. The new set includes the already existing capabilities and offers two extra options. In addition to slicing, dicing and

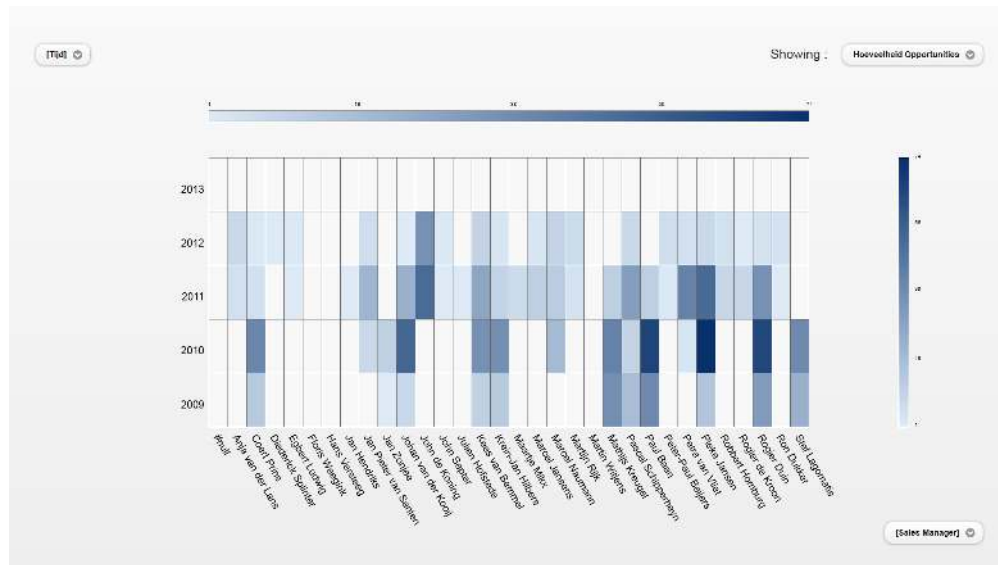


Figure 9.1: Finished dashboard prototype

drilling up and down, the new gesture set allows users to remove a member or to compare different ones.

### Slicing

The slicing gesture has not changed - it is still a tap on the desired member. Figure shows the visualization being sliced on the third member on the left. The right side shows the result.

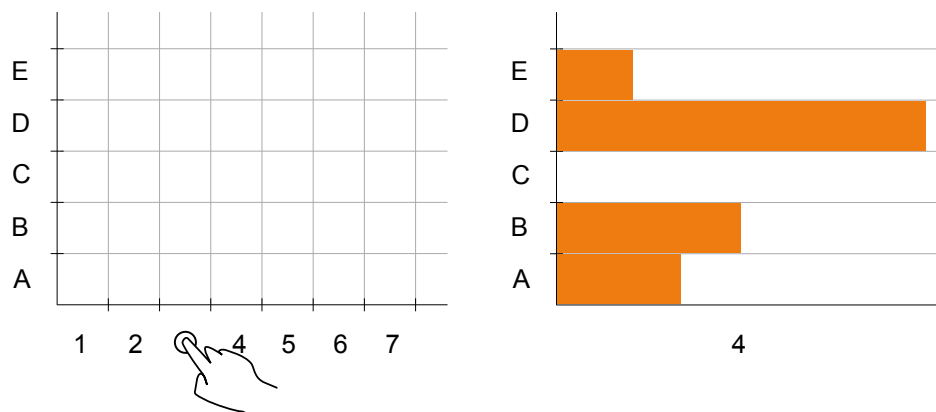


Figure 9.2: The gesture for dicing remains the same



### Dicing

The gesture for dicing has changed from the initial design. It has been changed from an un-pinch gesture to two parallel swipes. If the swipes are vertical, the visualization will be sliced on the x-axis. If they are horizontal the visualization will be diced on the y-axis. The starting points for the swipes will be used as a cut off point for the dice.

The left side of Figure 9.3 shows the slice gesture performed on the third and fifth member along the x-axis. This means the new visualization will start at the third and end at the fifth member. This situation is shown on the right side of the figure.

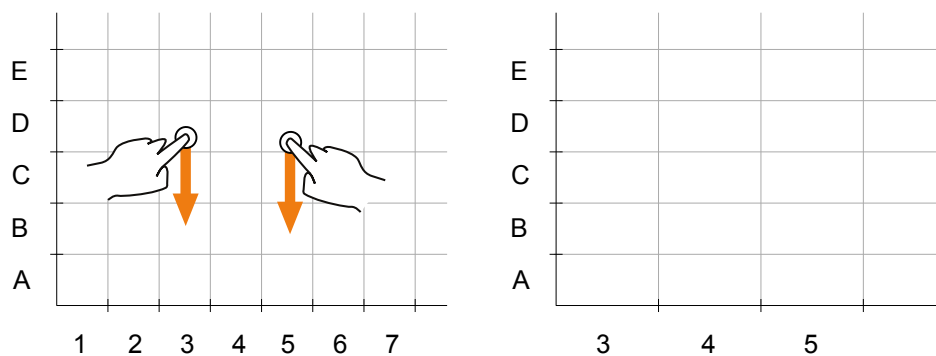


Figure 9.3: New gesture for dicing

### Drilling up and down

In Chapter 8 it was mentioned that testers got lost in the hierarchies. This could be caused by a lack of animation, but could also be partly caused by the gestures used for drilling up and down. The new gestures for drilling up and down are pinching and un-pinching gestures respectively. These gestures are possibly more intuitive than the previously assigned gestures, but this requires more investigation.

Figure 9.4 shows the drilling up action being performed. It does not matter on which members this action is done. The result will be a visualization that is moved one level up in the hierarchy. The right side of Figure 9.4 shows a visualization at the top level of a hierarchy.

The gesture for drilling down is shown on the left side of Figure 9.5. It is an un-pinch gesture similar to the one used to zoom in on a map on other devices. The un-pinch gesture is comprised of two swipes. It is important that the starting points of those swipes are the same member or the action will not be performed. This is the case in this figure, where both of the swipes start at the fourth member on the x-axis. The resulting visualization will show the children of this member. This is shown on the right side of the figure.

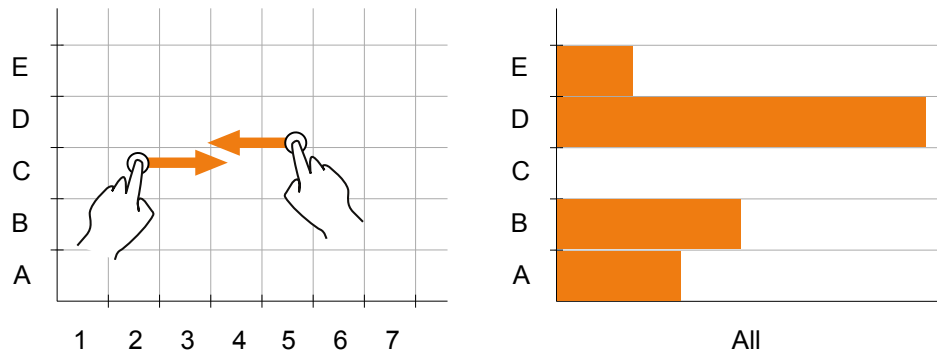


Figure 9.4: New gesture for drilling up

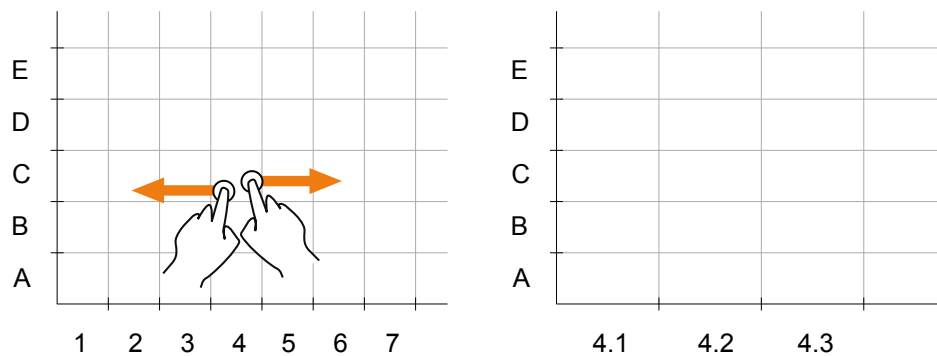


Figure 9.5: New gesture for drilling down

### Remove

One of the concerns that were raised during the implementation was that often *#null* members would be present in the visualization. To address this concern an option for removing members has been introduced. The remove gesture is a simple swipe. When this swipe is performed horizontally a member is removed from the y-axis. If the swipe is vertical, a member on the x-axis is dropped. Which member is removed is determined by the starting point of the swipe.

The gesture shown on the left side of Figure 9.6 will remove the third member on the x-axis. The result is shown on the right side where the third member is missing.

### Compare

The final gesture in the new set, is used for comparing different members. The gesture is comprised of two or more taps. The gesture starts when a member is selected with a

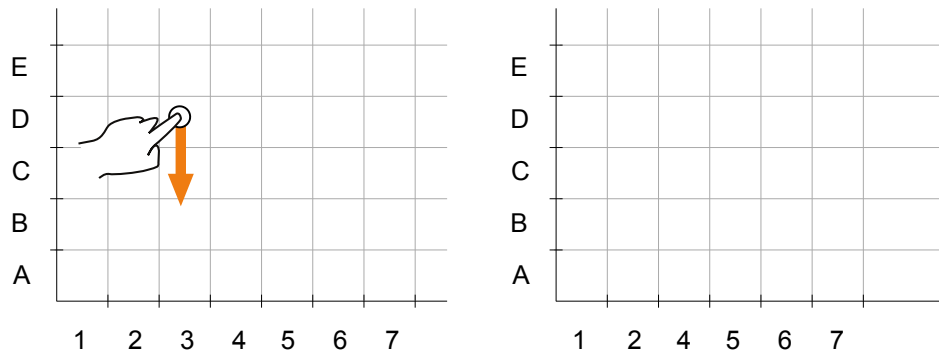


Figure 9.6: New functionality has been added for removing members with this gesture

tap that is held. Any other members that are selected while the first tap is active, will be added to the comparison.

Figure 9.7 shows the selection starting at the second member, while the third and sixth member are added to the comparison. The result can be found on the right side of the figure.

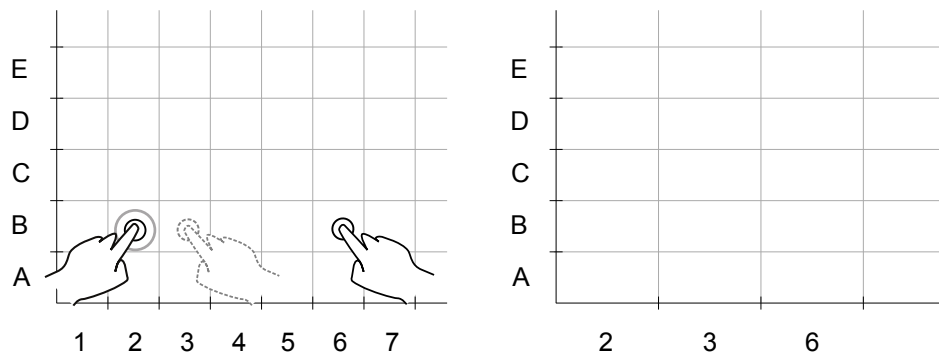


Figure 9.7: The gesture for comparing different members is shown on the left. The result is on the right

In Section 8.4 it was suggested that adjusting the gestures could have two benefits. It would make them easier to learn and it would reduce the chances of users getting "lost" in the information. To see if these new gestures would have this effect, more user tests are required. Unfortunately due to time constraints user tests have not been performed with these new gestures.

### 9.3 Conclusion

The evaluation of the prototype dashboard has resulted in several areas for improvements. Based on these result from the evaluation the dashboard has been improved.

The changes made to the visualization have given the dashboard a more polished look. Next to the look of the prototype, the navigation has been revised as well. A proposal is given for a new gesture set for the navigation. This has been done to make the navigation more intuitive and preventing users getting "lost" in the hierarchies. Unfortunately these new gesture have not been tested for their effects.

## Chapter 10

---

# Conclusions and Future Work

This chapter concludes and summarizes the work carried out during this thesis project. This chapter will start with reiterating the research question posed at the start of this project. This will be followed by the contributions that have been made during this project in Section 10.2. This will be followed by conclusions in Section 10.3. This chapter will end with recommendations and future work.

### 10.1 Research Question

This project focused on the development of a gesture based interface for OLAP information from the Salesforce CRM tool. This development has taken place at Incentro who wanted to allow managers to gain insights into its sales activities. In order to do so Incentro had utilized a Data Warehouse to store the information. This was described in Chapter 3. Given this situation this thesis set out to answer the following research question:

*"How can a gesture-based interface support the intuitive navigation through OnLine Analytical Processing information by end users, in a Customer Relationship Management domain?"*

This is a question with a high level of abstraction, and in order to make the whole more manageable, the question was split into smaller ones. These questions will be discussed in the following section.

### 10.2 Contributions

This section discusses the contributions of this thesis, structured using the sub-questions posed in Chapter 1.

1. What information needs does a sales person at Incentro have?

One important step for the development of the dashboard was indexing the information needs of the end-users. This was done by interviewing the targeted end-users. The rest of the results of these interviews were materialized into use-cases. The resulting use cases only cover the basic functionality, because a fully

functional dashboard was not the goal of this project. These use cases were used for the development of the system as well as its evaluation.

One concept that was left out of the use cases was the Sales Funnel. To the end-users this was one of the most important points, because it is an indication of future sales. Unfortunately it is unknown if it is possible to provide this information using OLAP technologies and because of time limits this issue was never addressed. Fortunately the funnel can be approximated using the weighted revenue and the closing dates of opportunities.

2. How can a gesture based interface be designed to fulfill those information needs?

With the basic information needs of the end-users mapped out, the next step was designing a system to fulfill those needs. In this report a proposition for a design has been described, comprised of two parts: a design for the visualization of OLAP information, and a design of a gesture set used for the navigation.

The visualization proposed in this report allows users to select two OLAP dimensions and set them out against each other along two axes. The values or measures are shown using color lightness. While the design of the visualization is a minor contribution, the main contribution of this project is the set of gestures used for the navigation. These gestures are inspired by the touch interaction on existing touch devices and allow users to navigate through an entire hierarchy of a dimension.

3. How can such an interface be built?

For the design to be evaluated by the end-users, it has to be implemented. The decision was made to implement the dashboard as a touch-enabled website. The development of the prototype was done using web standards so that it can be used across different platform. The platform was implemented by partly using existing libraries, as well as custom code for the touch interaction and the MDX query generation.

The custom code for the touch interaction was required because existing touch libraries were mainly focused on touch events provided iOS. However some of these events are not defined in the HTML touch specification. To avoid conflicts custom code was written. The chosen approach has resulted in a dashboard that works, not only on the development environment, but also on the iPad. However some gestures do not work correctly on the iPad. This may have been caused by a conflict between the custom code and the events supported on the iOS browser.

The use of touch interaction on a web application further increased the complexity of the project. Developing and testing the dashboard required a desktop browser supporting touch interaction. Unfortunately this functionality was not provided by the major browsers. This is why a workaround was required to get touch support in a desktop browser.

During the development of the dashboard several issues surfaced. These will be discussed in following sections.

## 10.3 Conclusions

At the start of this project there were many different BI applications available for touch-enabled devices. But all or almost all of these programs only use point-and-click interaction, not making full use of the capabilities of the touchscreens. That is why for this project a dashboard was designed and implemented to see whether touch gestures could be used to navigate through BI information on touchscreen devices using CRM information.

The design and implementation of the dashboard have been evaluated by conducting user tests, based on use cases developed with the help of some of those same users. After the tests the users were asked to fill in a questionnaire to acquire some quantifiable results. Even though the size of the dataset is rather small, the results can be used as an indication. The prototype was received positively by the testers. These testers also found the use cases easy to perform, and they indicated the dashboard invited them to play around and explore the information.

However the results also indicated that there are some points which could be improved. These points will be discussed in the section below.

## 10.4 Recommendations and Future work

The current dashboard is still only a proof of concept, but it has been well received by the intended end-users. However, the evaluation has revealed several points which can be improved. In the following section future work in this area is presented.

### Data Quality

A concern first became apparent during the implementation phase was data quality. Often the result of queries would contain *#null* members, indicating that some fields were not filled in on the Data Mart. It is recommended that Incentro address this issue, for example by setting guidelines and enforcing them.

### Comparing gesture sets

Based on the evaluation of the prototype, the dashboard was developed further. Additions to this version of the dashboard include a new gesture set for navigating the visualization. This could possibly make the dashboard more intuitive to use. It could also prevent users from getting lost. To research this, a new round of user testing is recommended, comparing the new gesture with the old one.

### Zoomable web applications

For the development of the dashboard, custom code was written for the multi-touch functionality. This code used the new HTML5 touch specification. This has resulted in a web application that works on a desktop as well as tablet. However not all the gestures seem to work on mobile devices. This warrants further investigation.

**Zoomable web applications**

Even on a large multi-touch screen the dashboard runs out of space when a large amount of data is displayed. This problem is even greater on devices with smaller screens such as mobile phones. This can be addressed by making the visualization scrollable or zoomable. There are few zoomable web applications available so new lessons can be learned if that avenue is taken.

**Changing the visualization**

The evaluation showed that users felt that they had too few dimensions at their disposal. The fact that there are only two axes available, is because of the choices made during the design. Based on the analysis of the end-users, the decision was made to focus on simplicity. This choice may have impacted the functionality of the dashboard. The current design only allows for two axes to be displayed and navigated through. To facilitate more axes, this design will have to be revised to allow multiple axes. An alternative to the proposed visualization, which could provide this functionality is parallel sets. This would make the visualization more expressive, but would also increase the complexity.

**Combining OLAP and relational databases**

The issues described above can be improved on the short term, but there is an issue which will require some time. One of the main concerns that appeared during testing, is the level of detail of the information. Most of the users wanted to see individual items at a more granular level. But this wish is not compatible with OLAP technology, which is aimed at providing aggregated data. To see more detailed information a regular database needs to be accessed.

It would be ideal if the high level aggregated data of OLAP could somehow be combined with granular databases. If there is a way to do this ROLAP technology could play an important role, because the multidimensional data provided from a ROLAP server is extracted from a relational database.

If combining the two technologies is not an option, perhaps another possibility would be developing an intermediate layer providing access to both the OLAP server as well as the relational source information.



---

# Bibliography

- [1] Aaron Bangor, Philip T. Kortum, and James T. Miller. An Empirical Evaluation of the System Usability Scale. International Journal of Human-Computer Interaction, 24(6):574–594, July 2008.
- [2] F. Bendix, R. Kosara, and H. Hauser. Parallel sets: visual analysis of categorical data. In Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on, pages 133 – 140, oct. 2005.
- [3] Cindy Brewer and Mark Harrower. Colorbrewer. <http://colorbrewer2.org/>, 2012. [Online; accessed 14-May-2012].
- [4] Cynthia A. Brewer. Color use guidelines for data representation. In Proceedings of the Section on Statistical Graphics, American Statistical Association, pages 55–60, 1999.
- [5] John Brooke. Sus- a quick and dirty usability scale. In P. W. Jordan, B. Weerdmeester, A. Thomas, and I. L. Mclelland, editors, Usability evaluation in industry. Taylor and Francis, London, 1996.
- [6] Andreas Buja, Dianne Cook, and Deborah F. Swayne. Interactive high-dimensional data visualization. Journal of Computational and Graphical Statistics, 5(1):pp. 78–99, 1996.
- [7] Surajit Chaudhuri and Umeshwar Dayal. An overview of data warehousing and olap technology. SIGMOD Rec., 26(1):65–74, March 1997.
- [8] John P. Chin, Virginia A. Diehl, and Kent L. Norman. Development of an instrument measuring user satisfaction of the human-computer interface. In Proceedings of the SIGCHI conference on Human factors in computing systems, CHI '88, pages 213–218, New York, NY, USA, 1988. ACM.
- [9] d3.js. d3.js. <http://mbostock.github.com/d3/>, 2012. [Online; accessed 13-April-2012].
- [10] Stephen G Eick. Visualizing multi-dimensional data. SIGGRAPH Computer Graphics, 34:61–67, 2000.

- [11] Klint Finley. 10 business intelligence tools for the ipad. <http://www.readwriteweb.com/enterprise/2011/02/10-ipad-business-intelligence-tools.php>, February 2011. [Online; accessed 22-February-2012].
- [12] Gartner. Gartner says new relationships will change business intelligence and analytics. <http://www.gartner.com/it/page.jsp?id=1513714>, January 2011. [Online; accessed 22-February-2012].
- [13] Jeffrey Heer and Ben Schneiderman. Interactive dynamics for visual analysis. ACM Queue, 2012.
- [14] Apple Inc. Safari web content guide. <http://developer.apple.com/library/ios/#DOCUMENTATION/AppleApplications/Reference/SafariWebContent/HandlingEvents/HandlingEvents.html>, 2012. [Online; accessed 16-Jan-2012].
- [15] Alfred Inselberg. N-dimensional coordinates. In Picture Data Description & Management, IEEE Pattern Analysis & Machine Intelligence (PAMI), 1980.
- [16] R. Kimball and M. Ross. The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling. John Wiley & Sons, 2011.
- [17] MSDN. Mdx language reference (mdx). <http://msdn.microsoft.com/en-us/library/ms145514.aspx>.
- [18] Tamara Munzner. Visualization principles. In VIZBI 2011. University of British Columbia, 03 2011.
- [19] Carl Nolan. Manipulate and query olap data using adomd and multidimensional expressions. <http://www.microsoft.com/msj/0899/mdx/mdx.aspx>, 1999. [Online; accessed 7-October-2011].
- [20] OLAP.com. Olap software and education wiki. [http://www.olap.com/w/index.php/OLAP\\_Education\\_Wiki](http://www.olap.com/w/index.php/OLAP_Education_Wiki), 2012. [Online; accessed 20-March-2012].
- [21] Pentaho. Mondrian documentation. <http://mondrian.pentaho.com/documentation/schema.php>, 2011. [Online; accessed 13-December-2011].
- [22] Pentaho. Pentaho mondrian project. <http://mondrian.pentaho.com>, 2011. [Online; accessed 6-December-2011].
- [23] QlikView. Explore qlikview. <http://www.qlikview.com>, 2012. [Online; accessed 23-August-2012].
- [24] Roambi. How roambi works. <http://www.youtube.com/watch?v=3spOHNnk1ac>, 2009. [Online; accessed 7-May-2012].
- [25] Roambi. Mobile business intelligence, analytics and dashboards | roambi. <http://www.roambi.com>, 2012. [Online; accessed 23-August-2012].

- [26] Salesforce. Sales cloud. <http://www.salesforce.com/eu/crm/sales-force-automation/>, 2012. [Online; accessed 14-Februari-2012].
- [27] Peter Shirley, Steve Marshner, Michael Ashikhmin, Michael Gleicher, Naty Hoffman, Garrett Johnson, Tamara Munzner, Erik Reinhard, Kelvin Sung, William B Thompson, Peter Willemsen, and Brian Wyvill. Fundamentals of Computer Graphics. AK Peters, third edition, June 2009.
- [28] C. Stolte, D. Tang, and P. Hanrahan. Polaris: a system for query, analysis, and visualization of multidimensional relational databases. Visualization and Computer Graphics, IEEE Transactions on, 8(1):52–65, jan/mar 2002.
- [29] Kesaraporn Techapichetvanich and Amitava Datta. Interactive visualization for olap. In Osvaldo Gervasi, Marina Gavrilova, Vipin Kumar, Antonio LaganÃ, Heow Lee, Youngsong Mun, David Taniar, and Chih Tan, editors, Computational Science and Its Applications, volume 3482 of Lecture Notes in Computer Science, pages 293–304. Springer Berlin / Heidelberg, 2005.
- [30] Edward Tufte. The Visual Display of Quantitative Information. CT: Graphics Press, 1983.
- [31] TUIO. Tuio.org. <http://www.tuio.org/>, 2011. [Online; accessed 20-September-2011].
- [32] T. S. Tullis and J. N. Stetson. A comparison of questionnaires for assessing website usability. In Proceedings of the Usability Professionals Association (UPA) 2004 Conference, pages 7–11, June 2004.
- [33] Panos Vassiliadis and Timos Sellis. A survey of logical models for olap databases. SIGMOD Rec., 28(4):64–69, December 1999.
- [34] W3C. Touch events version 2. <https://dvcs.w3.org/hg/webevents/raw-file/tip/touchevents.html>, 2012. [Online; accessed 3-April-2012].
- [35] Chris Weaver. Building highly-coordinated visualizations in improvise. In IEEE Symposium on Information Visualization, pages 159–166. IEEE, October 2004.
- [36] Wikipedia. Star schema — wikipedia, the free encyclopedia. [http://en.wikipedia.org/w/index.php?title=Star\\_schema&oldid=449167595](http://en.wikipedia.org/w/index.php?title=Star_schema&oldid=449167595), 2011. [Online; accessed 16-September-2011].
- [37] xmla4js. xmla4js - xml/a for javascript: Olap for client-side web applications. <http://code.google.com/p/xmla4js/>, 2012. [Online; accessed 29-March-2012].
- [38] xmlforanalysis.com. What is xml for analysis (xmla)? <http://news.xmlforanalysis.com/what-is-xmla.html>, 2012. [Online; accessed 20-March-2012].
- [39] Yellowfin. Yellowfin mobile bi for the ipad. [http://www.youtube.com/watch?v=trjqCY3bL\\_k](http://www.youtube.com/watch?v=trjqCY3bL_k), 2011. [Online; accessed 7-May-2012].

- [40] O.R. Zaiane, Man Xin, and Jiawei Han. Discovering web access patterns and trends by applying olap and data mining technology on web logs. In Research and Technology Advances in Digital Libraries, 1998. ADL 98. Proceedings. IEEE International Forum on, pages 19 –29, apr 1998.

## Appendix A

### Data mart


#	Name	Datatype	Length/Set	U
1	account-id	VARCHAR	18	
 2	<b>opportunity-id</b>	VARCHAR	18	
3	SluitingsDtm_yr	INT	4	
4	SluitingsDtm_qtr	INT	1	
5	SluitingsDtm_mth	INT	2	
6	SluitingsDtm	DATETIME		
7	Opportunity_cnt	BIGINT	21	
8	Gew_Omzet_sum	DOUBLE		

Figure A.1: Columns from the fact-table on the data mart

#	Name	Datatype	Length/Set	Ui
1	opportunity_id	VARCHAR	255	
2	Account_id	VARCHAR	255	
3	Verwijderd	VARCHAR	255	
4	Naam	VARCHAR	255	
5	Beschrijving	TEXT		
6	Fase	VARCHAR	255	
7	Bedrag	VARCHAR	255	
8	Waarschijnlijkheid	VARCHAR	255	
9	Sluitingsdatum	VARCHAR	255	
10	Gesloten	VARCHAR	255	
11	Gewonnen	VARCHAR	255	
12	Prognosecategorie	VARCHAR	255	
13	Prognosecategorie_1	VARCHAR	255	
14	Heeft_regelitem	VARCHAR	255	
15	Prijslijst	VARCHAR	255	
16	Eigenaar_id	VARCHAR	255	
17	Aanmaakdatum	VARCHAR	255	
18	Gemaakt_door_id	VARCHAR	255	
19	datum_laatste_wijziging	VARCHAR	255	
20	Laatst_gewijzigd	VARCHAR	255	
21	Systeemwijzigingsstempel	VARCHAR	255	
22	Laatste_activiteit	VARCHAR	255	
23	Fiscaal_kwartaal	VARCHAR	255	
24	Fiscaal_jaar	VARCHAR	255	
25	Fiscale_periode	VARCHAR	255	
26	Sales	VARCHAR	255	
27	Kans	VARCHAR	255	
28	Vakgebied	VARCHAR	255	
29	Leadbron	VARCHAR	255	
30	Gewogenomzet	VARCHAR	255	
31	Spanje_component	VARCHAR	255	
32	Contactpersoon	VARCHAR	255	
33	Fase_check	VARCHAR	255	
34	Advisory_actief	VARCHAR	255	
35	GELDIG_VA	DATE		
36	GELDIG_TM	DATE		

Figure A.2: Columns from the dimension table on the data mart

# Appendix B

## OLAP Schema



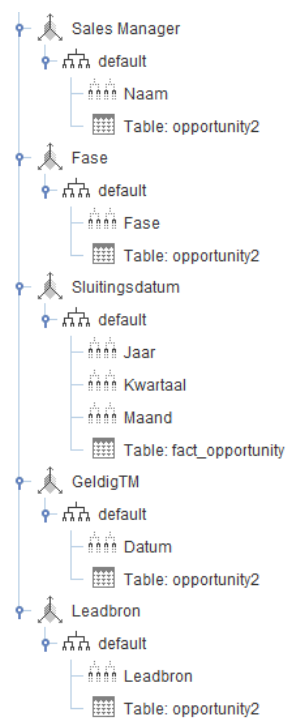


Figure B.1: Schema for the OLAP Server



## Appendix C

---

# Excerpts from user tests

This appendix summarizes the conversations during the user tests. During these tests the users were encouraged to think out loud and they were free to ask questions. These conversations have been recorded. Of these recordings the most notable points have been transcribed. A lot of information has also been left out because it was part of the script described in Section 8.2.1. The user tests were conducted in Dutch so the remarks have been translated into English.

The user tests were divided in different parts, as described in Section 8.2.1. Users were asked to explore the dashboard by themselves. This was followed by an explanation of the information and instructions on the gestures. After this, the users were asked to perform use cases. And finally the testers are asked for concluding remarks. The excerpts will be marked using these different phases - exploration, instructions, use cases and concluding thoughts. This is done to show in which phase of testing certain remarks are made. A T will indicate that the tester has said something and a C is a remark the person conducting the test has made.

### C.1 User Test 1

#### Exploration

- T: These are revenue forecast numbers?
- C: These are the sums of all the opportunities ever
- T: I don't get it! These are the number of opportunities? I now see that existing customers have resulted into eighteen opportunities? Is this current?
- C: No the information has not been updated recently.
- T: So this is historic data...
- C: Yes
- T: This is the current situation including the history. because this is history. There are eight proposals at existing customers. That is not a lot.. my conclusion

- T: Oh now I have to go back to the previous screen... how do I go back?
- C: You can do this by clicking on this.
- T: The darker the color the higher the value... and gray is no value? Is there no value or is the information not available
- C: There is no value
- T: These two are the most important. The other information is too old.
- C: So you don't care about historic data?
- T: Well actually I do, but this one is not important anymore
- T: Our funnel is filled by own network and existing customers and the input from partners has become minimal.
- T: You cannot drill down further?
- C: No
- T: I would like to be able to do that

### **Instructions**

- C: Funnel was not possible but I have included the closing date. You said this was a good approximation
- T: That is correct

### **Use Cases**

- T: How can you see that this is 2012? I cannot see it anymore
- T: I would prefer to see the values of the won leads and discard the lost ones.
- T: I would like to go further into the cell and show the separate opportunities
- C: Is it possible to link this to actual revenues?

### **Concluding thoughts**

- C: So, what do you think?
- T: This is cool, I can show this to colleagues, and compare them
- T: Can I set this out against year? (regarding a full visualization)
- C: Unfortunately no
- T: This is polluted data. It would be nice if you could know which one they are and edit them. By playing around I gained some new insights and some suspicions have been confirmed. It would be nice if I was able to click through the information. It is noticeable that the information is not up to date. It would also be nice if the information was summarized.

## **C.2 User Test 2**

### **Exploration**

- T: Starting at the Y-axis I want to see myself. I want to see how many things I have in a certain phase.
- T: So now I can see this not for just myself, but also others. And now I want to see how much I have won. And I want to see the same for this person.
- T: It is weird that this person has won nineteen and this other person has won fifty four.
- C: Well this information is all of the opportunities found in Salesforce.
- T: But even then I expected different values.

### **Instructions**

- T: I now want to zoom in on 2012... I think I broke it.
- C: Selecting the same hierarchy on both axes breaks it.
- T: Does it let me know I broke it?
- C: No...
- T: You could gray it out. Now I want to see something from EIM2011. Hmm we have not won anything from this. And we have won one from EIM2009.

**Use Cases**

- T: I would put something here like a reset button (Lower left corner) , and something of a help button. And showing the total amount would be nice.

**C.3 User Test 3****Exploration**

- T: Hmm what do I see here? So these are the number of opportunities that everyone has?
- C: Yes
- T: This is not right this number should be higher.
- C: That is true, this is because the information is only updated until the start of this year. It is lagging behind a bit.
- T: It would be nice to see the sum of the information along the edges.

**Explanation**

- T: Can I click down through this?
- C: No, the technology I used does not support this.
- T: Too bad, I would like to be able to see which opportunities make up this cell. And maybe even edit them from here
- C: Interesting to see that fewer opportunities have been entered in 2011 than 2010.

**Concluding thoughts**

- T: It would be nice if you could see more axes. For example it would be nice to be able to show only opportunities that have been won for a certain rendering. It would also be nice to see multiple measurements next to each other.

## **C.4 User Test 4**

### **Exploration**

- T: I see a lot of gray is that zero?
- C: That is no value
- T: What a drama.
- C: The fact that there is no value is not bad but if there are low values could indicate problems.
- T: And it is that the maximum is determined by this one, which is ridiculously high. So if you remove this one the rest will be more spread apart (according to color)
- T: This is strange. This value is shown for a year that is closed.
- C: These are all the opportunities even those that are closed. So you would rather see only open opportunities?
- T: Yes, otherwise I would like to see the sum of the revenues instead of weighted revenues.
- T: I now notice that the data is not correct. It works well, but then you get insights which are not true.
- T: It would be nice if you were able to determine which scale is used, so you can filter information out. I notice if there is a very high value, the other colors are very similar.

### **Explanation**

- T: I would like to remove this one, is that possible?
- C: Unfortunately not, but it would be useful if you want to filter outliers.
- T: Exactly, because I would like to remove this one.
- C: Would it be useful if you could select different people so they can be compared?
- T: Definitely
- T: I would like to be able to filter members, for example for each office and only active opportunities
- C: Well it would be possible to categorize closed and open opportunities.

- T: Yes, but WON is the option I always want to see.
- C: That would be one level lower than open opportunities.
- T: Yes then you have active and inactive opportunities.
- T: What I would do is... you now have white for zero - or close to zero - and gray for no value. If someone has close to zero opportunities he has a small value. For me that is almost the same as no opp. This gray makes it more complicated for me. There is a huge difference between zero and one. And this is more interesting. The difference between zero and five is more important than zero and one. Everything that is dark blue is good. And the gray distorts it for me.
- C: And if it was lighter gray?
- T: For me white is better.

### Use Cases

- T: Lead source is not important because it is not properly maintained or updated.

### Concluding

- C: Are there things you are missing?
- T: Yes what I mentioned before, office name, filtering. And I would like to be able to remove some of the items.
- T: The outliers cause that the rest are more difficult to differentiate. If you could adjust the scale, so that I can determine which values are dark or light. I would also like to have another filter.

## C.5 User Test 5

### Exploration

- T: Hey this is not possible. You either have an opportunity or not. You cannot have half opportunities.

T: What do these colors represents?

T: I would like to see the accounts

### **Instructions**

T: Nice and how will this work on a laptop?

C: It won't

T: So you cannot go through to the opportunity itself?

C: No the technology used, will not allow it.

T: Do you have any other renderings? The gray is nothing, the white have low values. Why have you chosen gray?

C: To indicate that there is no value.

T: Why are these gray? This would mean that there are opportunities that have no sales manager That should not be possible.

T: Do you have any other renderings?

C:

### **Use Cases**

T: I should have more weighted revenue here.

C: This information is not up to date.

### **Concluding thoughts**

T: I would adjust the legend. You can easily get an overview of information using this.

T: Check the source. The source information is important. We have only started using Salesforce and you can notice this. Also a lot of things are changing.

C: Are there things that are missing or things that you would change?

- T: I would remove the gray, it can be shown differently. And again the scale should not contain half numbers.
- T: I would also like to see a number showing the total of all the values displayed.
- T: The last thing I wanted to say. These are the quarters, you can see that they are filled nicely. But you want to see the months across the years. All twelve of them and not for a quarter.
- C: I configured the server in this way. But you would like to see all the months next to each other, to get an overview?
- T: At the start of the year there are two quarters which are important. Followed by a quarter in which not a lot is done. Then there is a quarter in which a lot happens. If I look at the quarters I don't have the right overview.



## Appendix D

# Evaluation Questionnaire

This questionnaire is used to evaluate the Incentro Salesforce dashboard.

For most questions there are only four possibilities

1. Strongly Disagree
2. Disagree
3. Agree
4. Strongly Agree

(so a neutral option is intentionally left out)

Questions	Tester					Average
	1	2	3	4	5	
Getting Started						
When viewing the starting screen it is clear what needs to be done	3	3	3	4	3	3,2
The items in the three drop down menus are clear to me	4	3	4	3	3	3,4
Without instructions I quickly got a hang of the gesture navigation	3	3	3	1	4	2,8
Without instructions I understood what information I could see	3	2	3	4	3	3
Use cases						
The explanations and instructions given were clear	3	3	4	4	4	3,6
The gestures are easy to learn	3	2	4	3	3	3
After the explanation, the gestures were easy to use	4	3	4	3	3	3,4

After the explanation, the navigation through the data is easy	2 3 4 4 2	3
After the explanation, the displayed information is clear to me	3 3 4 2 4	3,2
All the elements of the dashboard are placed logically	3 3 3 4 3	3,2
The values displayed are clear to me	3 3 3 2 3	2,8
The use of colors gave me a good overview	2 2 3 4 3	2,8
I find the User Interface visually pleasing	2 2 3 4 3	2,8
The texts along the axes are clear to me	3 3 3 2 4	3
The legend helps me understand the data	3 4 1 3 3	2,8
The tooltip/popup with the value helps me understand the data	3 2 3 2 3	2,6
The use cases are easy to perform	4 4 4 4 4	4
I can now use the dashboard without additional help / instructions	3 3 4 1 4	3
<b>Information</b>		
While using the dashboard I gained new insights about the information	4 4 3 2 3	3,2
The level of information is too detailed	2 2 2 1 1	1,6
The level of the information is not detailed enough	3 2 4 1 4	2,8
I would like to use the dashboard with other information	3 4 4 4 4	3,8
The two available axes were enough	2 2 3 1 3	2,2
<b>General</b>		
I find the dashboard easy to use	3 3 4 3 3	3,2
I find the dashboard awkward to use	2 3 2 1 4	2,4
I find the dashboard complex	2 2 3 1 1	1,8
The ability to use gestures for navigation, gives the dashboard added value	3 1 4 4 3	3
The dashboard invites me to play around with the information	4 4 4 4 4	4

### *Evaluation Questionnaire*

---

I would use the dashboard on the Incentro multi-touch stand	4	2	4	4	4	3,6
I would use the dashboard on a touch enabled mobile device	4	3	3	4	4	3,6
I give the dashboard the following score	7	8	8	5	8	7,2