



**POLYTECHNIC
INSTITUTE OF VIANA DO
CASTELO**



**INTEGRATION OF BUSINESS INTELLIGENCE SYSTEM
AND MANAGEMENT INFORMATION SYSTEMS:
AN OPEN-SOURCE APPROACH**

Baghman Behbudov

Master in Technology and Management of Information Systems

School of Technology and Management

June 2010

**INTEGRATION OF BUSINESS INTELLIGENCE SYSTEM
AND MANAGEMENT INFORMATION SYSTEMS:
AN OPEN-SOURCE APPROACH**

Baghman Behbudov

Master

in

Technology and Management of Information Systems

Supervisor: Pedro Castro

Co-supervisor: Miguel Bento Alves

Program was supported by ERASMUS MUNDUS External Cooperation

I dedicate this master project to my Parents.

Acknowledgement

I would like to thank to my supervisor Professor Pedro Castro and co-supervisor Mr. Miguel Bento Alves for their support in the preparation of this thesis. In addition their comments were very important to improve the thesis. My other thanks to Professor Rui Gomes and Jorge Ribeiro for their some orientations. I also thank to Mr. Gasham Zeynalov, Associate Professor of Khazar University, Azerbaijan for his helpful advice.

Abstract

If we look at over past years, Business Intelligence (BI) and Management Information System (MIS) became unavoidable technological advantages that bring financial benefits to large companies which could purchase, implement and maintain the MIS and BI solutions. But small companies have limited budgets that can make unachievable the implementation of such systems because they cannot support, in general, the total cost of these solutions. In this context, open source software has the capability to reduce costs, providing the software industry with an alternative and competitive way of doing business, contributing to reduce the competitive disparity between large and small size enterprises.

This document explores an open source (OS) approach to integration of BI and MIS, taking into account that open source software can be an alternative option to the commercial solutions where the most important factor for this purpose is cost saving. In other hand, we present a research indicating that the integration of these systems can be a promising and desirable solution for small sized enterprises where the cost saving is an issue.

Keywords: Open Source Software, Business Intelligence (BI), Enterprise Resource Planning (ERP), Small Enterprises.

Table of Contents

1.	Introduction	1
1.1.	Background	1
1.2.	Research Problem	2
1.3.	Research objective	4
1.4.	Chapter overview	5
2.	Open Source Initiative	7
2.1.	Open Source Software	7
2.2.	Open Source Selection Requirements.....	11
3.	ERP Systems	17
3.1.	Commercial ERP vs. Open Source ERP.....	17
3.2.	Requirements for Open Source ERP	19
4.	Business Intelligence Systems.....	23
4.1.	Open Source BI vs. Commercial BI	25
4.2.	Requirements for Open Source BI Solution	28
5.	Work Developed.....	31
5.1.	Software selection.....	32
5.1.1.	Dolibarr ERP/CRM	32
5.1.2.	Pentaho BI	34
5.2.	Implementation	37
5.3.	Practical samples.....	49
5.4.	Summary	53
6.	Conclusion and Future Work.....	55
6.1.	Conclusion	55
6.2.	Future Work.....	57
7.	Bibliography	59
Appendices		
	Appendix A. GNU GENERAL PUBLIC LICENSE.....	i
	Appendix B. List of Top Open Source Business Intelligence Solutions.....	xvii

List of figures

Figure 1 - Average cost per user, comparing traditional and open source across scenarios	4
Figure 2 - Open Source Project ROI Cumulative Costs	16
Figure 3 - The global structure of the BI Systems	24
Figure 4 - Dolibarr ERP/CRM.....	34
Figure 5 - Pentaho Administration Console	36
Figure 6 - System model.....	37
Figure 7 - Dolibarr ERP/CRM Modules.....	38
Figure 8 - Sub-model of Dolibarr Database	39
Figure 9 - Data warehouse schema.....	40
Figure 10 - Tools and data flow of the implemented BI	40
Figure 11 - Pentaho Data Integration Tool.....	41
Figure 12 - Pentaho Data Integration Table Input step	41
Figure 13 - Pentaho Data Integration Value Mapper step	42
Figure 14 - Pentaho Data Integration Select Values step	42
Figure 15 - PDI Combination lookup-update step for product dimension	43
Figure 16 - Pentaho Data Integration Table Output step.....	43
Figure 17 - Star model of sales analysis	44
Figure 18 - Schema Workbench /ROLAP Cube	45
Figure 19 - New Analysis View	45
Figure 20 - Drill down method example	46
Figure 21 - Sales by year in Vertical Bar 3D chart type.....	46
Figure 22 - Creating an action sequence within Design Studio	48
Figure 23 - Dashboard example.....	48
Figure 24 - Modified source code	49
Figure 25 - Cross Table of Sales by Years	50
Figure 26 - Sales by Country.....	50
Figure 27 - The most important customers.....	51
Figure 28 - Sales by Year	51
Figure 29 - Sales in 2009 by month.....	52
Figure 30 - Drill Down by products	52
Figure 31 - Number of sales in May 2010.....	53

List of Tables

Table 1 - Open Source BI customers.....	27
Table 2 - Dolibarr ERP/CRM selection criteria	33
Table 3 - Pentaho BI selection criteria	35

1. Introduction

1.1. Background

Computer software is a revolutionary factor for businesses in the world economy. Software is one of the key elements of information and communication technology (ICT) that has a tremendous role in industry. If we look at early years of the computing, software was mainly free and it was evolved among researchers and software developers. However that tendency changed when computer technology started to be more common, and where the commercial and proprietary software became more suitable for many enterprises. In last several years the idea of Open Source Software (OSS) and free sharing of software products have become a base and perspective interest both for the software development and for economic concepts. Some organizations began to define that Open Source Software (OSS) can have benefits and profits. The most outstanding example of this is IBM, “which continues to reap high returns from its approximately one billion-dollar investment in Linux” [1].

Open source software is getting to be one of the most interesting experiences of the whole information technology scenery and it increases the amount of attention by different organisations. The impact of open source technology is extremely perceptible in the software industry, as well in society as a whole. It grants for innovative development models, which have already been established to be especially compatible to competently take advantage of the work of developers which are across all corners of the over the world. It also gives totally new business models that are creating a network of groups and companies based on open source software development. Commonly OSS has a very progressive influence to the creation of new software markets and business opportunities. Therefore open source projects can be viewed and examined since their existence is based on Internet.

The list of open source software increasingly changes every day. Already there is a lot of open source software for different business areas. Considering Management Information Systems (MIS) and Business Intelligence (BI), open source creates a playing field for many enterprises especially for small enterprises that in case of commercial software they have never had budget to pursue it. Small enterprises can be considered as economically and socially

important player in this field. In the context of Europe Union (EU) small-sized enterprises share is 7.4% of all small and medium-sized enterprises (SMEs) which are 23 million and represent 99% of all enterprises [2]. Europe Commission (EC) [3] reports that only 10% of all small firms used specific ICT solutions for marketing, sales and procurement compared with 20% of medium-sized and almost 30% of large firms. The same report presents that: “SMEs (Small and Medium Enterprises) still suffer from limited understanding of ICTs and their potential, limited budget for ICT investments and difficulty in recruiting ICT professionals”. However, if some years ago the only choice for business was to legalize or make illegal version of commercial software, now open source software is a new choice that can be realized on the business. Using open source technologies information management can intensely increase the complement of services and improve the effective organizational development through low cost and reasonable software solutions [4].

As the source codes are available for modification without any proprietary licensing issues organizations get noticeable opportunity and the freedom to customize the application for their own choice. So, considerably distinctive from proprietary solutions, the organizations have the full freedom to modify and customize the software product to their requirements which in case of traditional software the users are bound by copyright agreements.

If we notice to the past years the use area of open source software was narrow while recently it is already applying to the management level solutions. Thus all of these give us a reason to find new alternatives in the market that can be used and bring MIS and BI to small businesses. In this context, the software offers that appear on the global market as open source can constitute an opportunity to have low cost solutions for enterprises, especially for small businesses that have low IT budgets.

1.2. Research Problem

Management Information Systems (MIS) are unavoidable and competitive technological advantage of the companies which can buy, implement and maintain these solutions. Management and decision makers need up-to-date information to run their businesses. In this case MIS are integral and leading factors of every modern organization. Today’s competitive business environment makes it obligatory and important for organizations to develop these

systems and to compete in rapidly changing economy. Enterprise Resource Planning (ERP) implementations have changed the work landscape of companies and it is necessary for the business drivers because the ERP software packages provide the required work functions of MIS in an organisation. These systems are created for efficient data entry and storage in real time. But from time to time this growing data requires new tools and methods to manage and analyze, because ERPs are not oriented to design analytical reports. Business Intelligence (BI) fulfills this gap and can change strategic face of the company. In other words BI is simply a facilitator of these ERP systems. Furthermore, the integrated framework of BI and ERP adds another business value to enterprise systems. However in case of small sized enterprises there are some obstacles especially cost factor that prevents them to invest in the purchasing of commercial solutions in their business. Large businesses spend millions of dollars to implement powerful ERP and BI solutions and it makes possible to manage their businesses and still they can bear this load on their shoulders because of their budget capabilities. But all of these are financially out of reach of small enterprises. Small businesses are often faced with limited access of finance to support purchasing of commercial business intelligence [5]. It is undoubtedly true that there are a lot of small enterprises which are not able to buy expensive ERP and BI solutions from software vendors. Considering the cost of SAP ERP [6] licence for 25 users is about 140000\$ or Oracle BI [7] costs for 25 users are about 17000\$ this is unattainable for many small enterprises. If such kind of strength differences between big and small enterprises is increasing, small enterprises can lose their competitive development. Another way to see differences between open source and commercial vendors is the cost scales increasing with the number of users [7]. Figure 1 shows the average per-user cost over three years between open source and traditional BI. First of all considering small businesses there is a significant difference between open source and traditional BI. Changing from small environment to a larger one actual average costs increase the per-user cost with traditional BI while decreasing with open source.

So these points give us a reason that it is very important to define the opportunities of open source solutions for Management Information System and Business Intelligence that can be desirable solution for small businesses and startup companies.

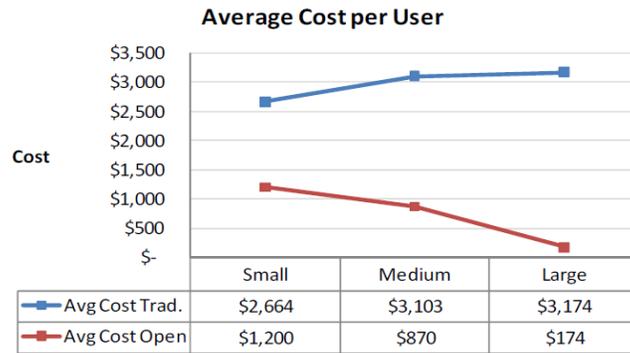


Figure 1: Average cost per user, comparing traditional and open source across scenarios [7].

1.3. Research objective

The purpose of this research is an experimental project to deepen knowledge on open-source software, namely an open source BI suite and open source ERP, and to develop an integrated system of these tools which can facilitate for small enterprises to increase the competitiveness. We will consider that OS BI and ERP can be an alternative for small enterprises. Within the research we are going to support the following positive expectations:

- The integration of open source management information system and business intelligence software is viable in order to create a business value and it can bring benefits to small enterprises.
- There is enough level of maturity, functionalities and potentialities that can be achieved with such a solution and the integration can provide enough business values and functionalities.
- The integration of OS systems is a solution that can provide cost savings for small enterprises.
- Another important aspect is that considering our individual requirements, selected tools are modifiable and customizable.

This document provides to research Open Source Environments, namely to study and choose necessary open source ERP software and open source Business Intelligence Tool, to integrate them and implement a Business Intelligence System and to define the potential benefits of this integration. We will define an architecture that can be adopted by companies with a lower cost of implementation. This proposition can be useful either to small companies or start-up companies that don't have enough budgets to acquire and implement commercial solutions.

Taking into account the objectives of this work and due to time limitations, the experimental system implemented in this project has been limited to the integration of two largely used open-source software and to a functional area of a business to be analysed with BI tools. However, we consider that, despite these limitations, the project is a contribution in an effort to give insight of the advantages and benefits to develop integrated open-source solutions that can be used by small enterprises.

1.4. Chapter overview

This thesis is composed of six chapters.

Chapter 1 we present the introduction to this project and the purpose of the document.

Chapter 2 describes the research background of Open Source Software Initiative. This chapter also elaborates the concept of open source software, open source development and evaluation of open source maturity.

Chapter 3 describes the important requirements for an open source ERP.

Chapter 4 explores Open Source Tools for Business Intelligence, the advantages of Open Source BI and the requirements for Open Source BI selection process.

Chapter 5 describes the work developed, the backbone of the thesis. This chapter describes the selection of OS software, the implementation of the experimental solution and presents some of the functionalities that can be achieved with such a solution.

Chapter 6 presents the conclusion that summarizes the key points of the project and future work.

2. Open Source Initiative

The Open Source Initiative (OSI) [8] is a non-profit organization that gives insight about open source software, supports for the advantages of open source and creates bridges among different people in the open-source community. The most important activity of OSI is to maintain the Open Source Definition for the gain of the community. The Open Source Initiative has an approved license trademark and a concrete program that creates a confidence for the developers, users, community members, corporations and governments who can organize open-source cooperation.

Open-Source software (OSS) has been carried out the activity since the 1980s, it was not ready for the general use but in general it has been viewed as a domain IT types. As an organization, Open Source Initiative was founded in February 1998, by Bruce Perens and Eric S. Raymond. Nowadays more companies, organisations and large government departments such as the US Department of Defence are one of proponents of open-source software solutions [8]. They state that open-source software has a lower total cost of ownership, has a more rapid development process and is more secure than proprietary software. Their reasoning for greater security is that, while the source code of products are open for all to view, security weakness are found and resolved before they are exploited.

2.1. Open Source Software

Open source software (OSS) is a computer software for which the source code is available and some rights usually reserved for copyright holders are provided under an open source software license that meets the Open Source Definition or that is available for the public domain [9]. This condition allows users to use, modify, extent the software, and redistribute it in modified or unmodified forms. The most important and noticeable difference between open source software created for freely using and proprietary software offered by vendors is that open source software is published under license which elucidates that the source code is available for everyone to examine, download, change and explore as they wish. The basic meaning of open source is that the source code which is created in the OS programming language (PHP, Python, Ruby) can be accessed modify and extend by anyone with the suitable skills.

Open source is not only about accessing to the source code and modify it. There are several distribution terms that have been defined by the Open Source Definition (OSD) and these terms must comply with 10 criteria [8]:

1. Free Redistribution

The license doesn't constrain and forbid any people for using, modifying and selling as a part of comprehensive software distribution that includes programs from different sources. The license also doesn't require a royalty or fee for such kind of sale.

2. Source Code

The program should contain the source code well-defined and must permit to distribute the source code in a good compiled form. Open source product should be distributed with source code that programmer can improve and customize it, and normally should be available to download via Internet without any charge. "Deliberately obfuscated source code is not allowed".

3. Derived Works

Licence must permit not only modification of source code but also a new generated and obtained source code to be distributed and undertaken with the same licence of original software.

4. Integrity of the Author's Source Code

"The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time". The license may control new generated program in order to hold new release name or version number comparing with original software.

5. No Discrimination against Persons or Groups

"The license must not discriminate against any person or group of persons."

6. No Discrimination against Fields of Endeavour

The scope of licence should be large. The license must not discriminate anyone while using the program in a particular field of science. For instance, the licence must allow program to be used in different fields of economy, research and so on.

7. Distribution of License

“The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.”

8. License Must Not Be Specific to a Product

“The rights attached to the program must not depend on the program's being part of a particular software distribution”. The rights which program is holding don't depend on a specific software distribution. If the program is the extraction of redistributed software the terms of licence and rights will be the same as in original version.

9. License Must Not Restrict Other Software

Open source licence can't make any limitation and restrictions for already distributed and licensed software. For instance, licence must not put any condition that other programs should be open source.

10. License Must Be Technology-Neutral

“No provision of the license may be predicated on any individual technology or style of interface”. The act of supplying or fitting out of the license may not be particular technology but should be neutral.

Recently it is true that open source development models are studied, while these new methods and models are being successfully tested. One of the first efforts which emphasized some of the characteristics of these new open source development models was by Eric Raymond's “The Cathedral and the Bazaar” [10]. In his book the author states how successful open software projects are created and managed. Eric Raymond analyses two different types of software development models: “cathedral” and “bazaar”.

“Cathedral development model” (like in Middle Ages cathedrals were built) is distinguished and indicated as proprietary software development. It has relatively strong control on designing and implementation of models. The cathedral model is the traditional style of software development. In this model there is a strict control while attaching patches to the source code. This model has a well-defined development structure but there is a strict control on releasing of codes. So no one can access source code and improve it.

“Bazaar development model” is based on informal exchange of knowledge and improving source code between the developers, and several programming styles can be chosen while building the software projects like in a bazaar where anyone can choose what they wish. There is no control on release code. This model as stated in the book may be characterized with an expression: “release early and often”. As author explained if the objective is to have fewer bugs, why to release a version in every six month, it is better to “work like a dog on debugging between releases”. This method allows users to be informed regularly on how the program is developed and it works like an indicator for developers. Seeing this “indicator”, informal developers or users extend the functionality of software “scratching the itch”. While this open source software can be improved, commercial or proprietary projects cannot get these benefits.

Considering “The Cathedral & Bazaar” we will enumerate Raymond's some points that we defined which are very important and useful to understand open source software development [10]:

- “Every good work of software starts by scratching developer’s personal itch.” While fixing the bugs in same time we improve the software.
- “Linus Torvalds' style of development is release early and often, delegate everything you can, be open to the point of promiscuity.”
- “Users are wonderful things to have, and not just because they demonstrate that you're serving a need, that you've done something right. Properly cultivated, they can become co-developers.”
- “If you have the right attitude, interesting problems will find you.”
- “Good programmers know what to write. Great ones know what to rewrite.”
- “When you lose interest in a Program, your last duty it is to hand it off to a competent successors.”
- “The Open Source world behaves in many respects like a free market or an ecology, a collection of selfish agents attempting to maximize utility which in the process produces a self-correcting spontaneous order more elaborate and efficient than any amount of central planning could have achieved.”
- “A happy programmer is one who is neither underutilized nor weighed down with ill-formulated goals and stressful process friction. Enjoyment predicts efficiency.”

- “It may well turn out that one of the most important effects of open source's success will be to teach us that play is the most economically efficient mode of creative work.”

2.2. Open Source Selection Requirements

Many enterprises are gradually using and adopting open source projects because of their effective cost savings and free license policies. All open source systems grant to the users continuous and permanent rights and freedom that users can use, study, modify, and redistribute the product. Despite of this, when end users chose open source products there can be a lot of discrepancy in terms of the product quality, functions, performance and the available support. So while selecting open source elements for an enterprise, it is important to make several initial checks to be sure that chosen open source product is compatible with our enterprise business requirements and IT standards. According to Chamindra de Silva there are 10 questions that we can ask to evaluate open source maturity [11]:

1. Are the open source license terms compatible with my business requirements?

Basically open source product is protected by copyright law, and the product license provides the fundamentals and principles for the rights that we hold on the selected software.

Considering as any open source product we need to have some main freedoms on it: freely download, use, modify and redistribute. But on the other hand, these licenses can have other terms that are incompatible with specific business models. For instance, the well-known General Public License (GPL) (See Appendix A) license asks that any changes, enhancements and derivative works also should be released under the same terms of license.

Although it sounds affordable condition and will not be problematic for general usage, it can be inappropriate to business models of proprietary product company that desires a GPL element in its software product. So it is important to understand open source definition and terms of licences before select them for our enterprise.

2. What is the strength of the community?

Open source projects are not only about the source codes, also about the open communities that extend and support the project. “These communities can range from the lone teen developer to a single organization to complex diverse multi-entity communities, such as found in Linux and

Firefox“[11]. In open source communities a wide range of users and developers test, maintain and fix the codes, security problems and bugs contributing to decrease the problematic source codes. That is why community can always be characterized as a key indicator of endurance of an open source project.

3. How well is the product adopted by users?

The users of open source projects are like pointers that give a feedback and show the adaptation level of product to specific requirements. This gives us enough assurances how well the project will be successful especially when there are users similar to our enterprise. It can be more confident if the user is still using the product. In successful experience, it usual to have a mailing list of users where feedbacks from users can be a good guide.

4. Can I get a warranty or commercial support if I need it?

All open source projects licenses are free and allow any modification on it but they don't give any warranty. In this case we can get support from community developers and before selecting the product we can notice the number of users that gives logical responsiveness to improve our confidence. There are particular users that are looking for open source software with fee-based or commercial support. But the “ideal open source project has multiple vendors involved in the product that provide different types of support from customization to maintenance to 24×7 production assistance” [11].

5. What quality assurance processes exist?

Quality assurance process is a systematic process of controlling if the open source product is developed and meet our specified requirements. We can check this process following the project releases. If we see the quality assurance process is not appropriate to our business standards we have to improve this process in our enterprise internally or spend money to realize it by vendors.

6. How good is the documentation?

One of the most important factors of open source projects are theirs documentation that can be clear and rich or confused and poor. Sometimes developers focus on the development of the project but they don't pay attention to the documentation process. At least all products need user manual and administrator guide that we can get online in suitable forums.

7. How easily can the system be customized to my exact requirements?

Open source, unlike proprietary products, has source code available for public which can be customized. But the cost of the customization differs depending on its system architecture and how well source code is written. Open source product should have flexible architecture that allows us to customize the product, or another choice, is to rely on a vendor that can do it for a fee.

8. How is this project governed and how easily can I influence the road map?

There can be several open source projects that are based on different models and governed in different ways. But the most successful governance model can be created if all developers, users and other contributors participate in the development of the open source system. So depending on the influence of users to community developers, open source projects can be oriented to our requirements.

9. Will the product scale to my enterprise's requirements?

Scalability is the ability of a computer software or product to keep the continuity of functioning even if its size, system environments and other elements are changed, for instance, changing the software to a new operating system. For example, more scalable open source software can be created on the base of WAMP (Windows, Apache, MySQL), LAMP (Linux, Apache, MySQL), J2EE technology and many of application's source code are written in PHP/Perl/Python.

10. Are there regular security patches?

The release of security patches is significant not only for open source but also for proprietary project. However in case of open source project it is more efficient, because, first of all, we can internally fix them and we don't need to wait answer from community developers.

Open source projects are developed under a free policy, containing many contributors from all over the world that have never met formally but still are sharing informal commitments. Gartner predicts [12] a significant diagnosis in all sectors: "By 2012, 80 percent of all commercial software will include elements of open source technology. Many open source

technologies are mature, stable and well supported. They provide significant opportunities for vendors and users to lower their total cost of ownership and increase returns on investments. Ignoring this will put companies at a serious competitive disadvantage. Embedded open source strategies will become the minimal level of investment that most large software vendors will find necessary to maintain competitive advantages during the next five years”.

In order to earn a profitable income from the investment, companies evaluate ongoing project performance and effectiveness of their projects assessing their Return on Investment (ROI) accurately. The same scenario concerns to open source software. When we present an open source system to our company the real costs and commitments of new system become clear [13]. In finance the return on investment (ROI) or just return is a calculation used to determine whether a proposed investment is wise, and how well it will repay the investor. It is calculated as the ratio of the amount gained (taken as positive), or lost (taken as negative), relative to the basis [14].

Return On Investment (ROI) is the economic benefits comes from spending money in the creation and development of a system. ROI is a performance measure that evaluates the productivity of the investment or compares the efficiency of several investments. ROI is an important measure to be sure about sustainability of our business. Return on investment is equal the amount of benefit or a return that derived from the investment divided by the cost of the investment. We can get derived ROI number as percentage or just a ratio:

$$\text{ROI} = (\text{Gain from Investment} - \text{Cost of investment}) / \text{Cost of investment}$$

Return on investment is a very popular indicator that is, if the investment does not have a positive ROI or if a higher ROI is possible, then the investment shouldn't be undertaken [15].

Managers sometimes are annoyed while analysing ROI for several reasons [16]:

- It is not easy to create a comprehensive approach. The expected results of any ROI can be changed by adjusting assumptions.
- Some intangible benefits such as time, flexibility or building simpler system architecture are frequently not taken into consideration.
- Managers have difficulty to make efficient estimation about how the new technology will increase profit.

While we analyse the specific costs of calculating ROI of using open source, these costs can be classified in three main categories [17]:

- License/support/maintenance/subscription costs,
- IT/Service provider costs,
- Organizational costs.

Each of these categories has different type of costs that allows analyse the ROI:

1. License/Support/Maintenance/Subscription Costs.

These costs characterize the payments made to developers in order to get access for using software. License fees are advance payment for a vendor's delivering software, while Support/Maintenance refers to the annual payments required to get access to patches for software that has already been licensed. Subscription may be a less familiar term, but it is often used for both commercial and open source products. Subscription is a yearly fee paid to maintain access to the regular releases and enables limited time for using products.

2. IT/Service Provider Costs.

These costs can also be called technical costs that include the work of evaluation, installation, configuration, customization and operating the software. The expected costs for all these elements depend on how much programming, integration and customization are required for modification of product. Also these costs can be comparatively different depending on technical personnel that can be internal or external. That is why it is important to be careful to define the amount of work needed to implement and hold an open source system.

3. Organizational Costs

Organizational costs are those derived by the end users of a new system. These costs reflect the learning of new software. When end users start to use new software productivity decreases till they are more familiar with the new system. The all cost experienced during this period is accumulated in organizational costs.

Figure 2 illustrates the effect of the project time duration can have on the relationship of the different cost elements of a project. While license fees during the first year of the project

represent a smaller percentage of project costs than IT/Services costs, after this, percentage begins to increase because of their recurring maintenance fees and they ultimately become the majority of total project cost.

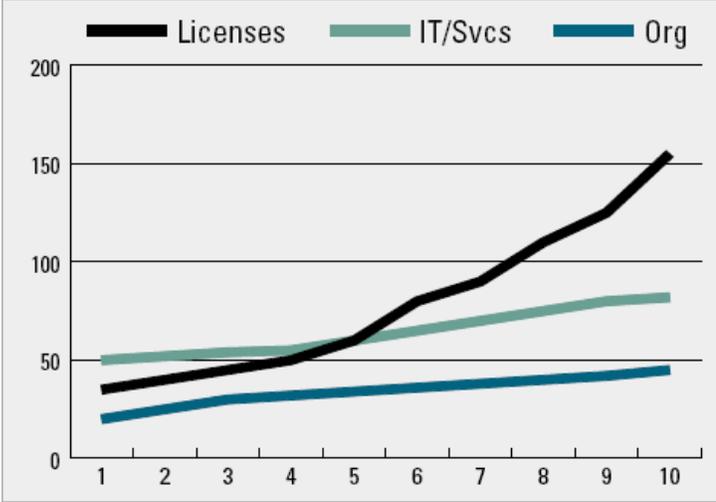


Figure 2: Open Source Project ROI Cumulative Costs [17]

Consequently when we make an open source ROI calculation, it is essential to use a realistic project till this will make a more accurate ROI assessment [17]. Then it is recommended to create a table with ROI calculations based on several years’ project durations. This allows to assists managers in the decision process based on ROI information.

3. ERP Systems

ERP systems are important strategic tools in today's competitive business and IT environment. ERP - Enterprise Resource Planning is an information system which integrates and controls all the business processes in the enterprise. According to Nah and Lau [18] ERP is "a packaged business software system that enables a company to manage the efficient and effective use of resources (materials, human resources, finance, etc.) by providing a total, integrated solution for the organization's information-processing needs". ERPs are powerful systems for the comprehensive management of the company and the integration of all the functional information, enabling operations on the collected data, based on common database. ERPs allow easy and immediate access to information about inventory, product, customer and sales.

3.1. Commercial ERP vs. Open Source ERP

Enterprise resource planning exists in two forms. One of them is standing as commercial while another is called open source ERP. Commercial and Open Source ERPs have many differences [19]:

- **Pricing**

First of all commercial ERP software is possessed and managed by an owner company and the package is normally expensive and is developed for corporations. The price depends on the size of the enterprise and business. Commercial ERP packages are not flexible and modifiable. Any modification and customization can cause troubles. These packages are not subject to flexibility and moulding. Their usage modalities are rarely liberal and cause troubles when they are modified.

The biggest advantage of open source ERP packages is that they are available as free of charge in internet. This is a desired factor to companies since it decreases the number of costs in the table of expenses. Also licenses are available with source code and any customization process is allowed without any expense.

- **Flexibility**

The flexibility criteria is a technical concept that is related with the platform independence of the package. Comparing open source ERP with commercial one, this important feature can be classified under some sub-criterias:

- Customization: depending upon the skills of enterprise it is allowed to customize for our business requirements. In contrast, the use of commercial ERP can have the consequence to change the way of the business.
- Database independence: the database has high influence on the scalability of the system. We can easily access to the relational model of the system in OS ERP, while commercial ERP makes this access difficult.
- Programming language: the programming languages usually used in open source ERP Systems are open source scripting languages like PHP, Python, Perl and Java. We can use these languages and make our modifications.

However when it comes to commercial ERP systems, above processes are complicated. These major differences between commercial and open source ERP show that open source ERPs are more flexible.

- **Dependence and Results**

Open source ERP is built by different informal developers and as the source codes are open, the companies rely less on vendors and they take care of their ERP system themselves. So the future of the ERP system depends on the activity of the company and they can control the productivity of the system. This condition can affect and decrease the failure rates of ERP system and this can be characterized as a way of relegating the level of dependence on the ERP vendor.

- **Security**

As open source enables to access source code and everything is transparent and available, anyone can examine the software to identify security insufficiencies. So the users and volunteer developers can make the elimination of security defects that can be missed, that improve the security level of the software. But in case of commercial ERPs they are more risk, since they are vulnerable to traps and pitfalls of hackers [19].

Open source ERP systems are often used by companies whose requirements are not met by standard software since they can make adaptive software modifications in order to adapt to changing process and requirements.

“For a long time, smaller companies purchased individual accounting and payroll packages and then migrated to expensive ERP solutions as they grew” [20]. Now, with much Open Source software there are many possibilities for Small and Medium-Sized Enterprises (SMEs) to choose ERPs. There are several Open Source ERPs in the market including Compiere, Dolibarr, ERP5 and Openbravo. However, while selecting ERPs companies have to carefully define the individual requirements and test the software to determine fitness.

3.2. Requirements for Open Source ERP

One of the most important aspects in ERP selection is the process of evaluating and defining requirements for an open source ERP System and these aspects are never the same for any two companies. Only specific internal conditions will show exactly what work should be done. Selecting the right ERP software for our company can increase productivity, improve operational superiority and take customer satisfactions to new levels. There are a lot of ERP applications to choose from, but basically, we should choose an ERP offering that best fits our company's requirements. While choosing open source ERP, several unique factors must be considered. Considering the project the most important requirements for open source ERP should be as following [21]:

1. Customization Complexity

Even if it is considered that ERP solution is complete and full, customization of the software is never avoidable. As the source code is freely available open source ERP solutions are easy to customize. In addition these systems are coded in popular, widely used computer programming languages, for instance, PHP and Java. One more important issue is in case of software has been customized internally, how this new code can be integrated with software upgrades.

2. Functionality

In general all ERP solutions have enough procedural functions that can support main business processes, like functionalities to create customer, vendor and products, maintain customer orders.

However, ERP solutions can differ largely on functionalities that are not the core business process. There are, for example, solutions oriented to manufacturing and others to accounting. So it is important and necessary to define the strength and advantage of every open source ERP solution in order to make a decision which one meets our business requirements better.

3. User interface

Nowadays ERPs use technologies that make possible to build friendly and satisfying user interfaces that allow users to easily find information and connect the ERP to the business process. User interface must be clear, comprehensive, with context based support and should facilitate the execution of daily tasks efficiently.

4. Development Activity

In competitive IT market sometimes small companies are bought by big companies or they can go bankrupt with the consequence that software solutions can disappear. However in the case of products that are open source and available for public, we can develop the product from remaining condition till they meet our requirements even if the formal development doesn't continue. So a good development activity of ERP software makes a significant role in our future business decisions.

5. Knowledge

One of the important and success factors of information system is knowledge about the product developed. Sometimes some software products cannot be successful, because there is an insufficient number of professional that know how to implement the software product. This also concerns to ERP software where it is important to have enough experienced consultants that can help customers in implementing ERP solutions.

6. Integration

All ERP project needs integration. While using open source ERP we need to communicate with other software systems like mail servers, the software of ordering system, shipping companies. This kind of integration in commercial ERP systems is created by the vendor using several integration components. In open source ERP this level of integration is not supported. Generally,

they only provide the standards that enable the integration with other software, for instance, using XML files we can import and export data or using web services.

7. Migration

This is one of the most important factors of ERP systems. Migration provides functionalities to ERP systems that allow the migration of great amount of data from older ERP system to the new one. This migration depends on the complexity of accumulated data. Inefficient migration can cause failure of ERP system. Usually ERP systems provide export and import sources between several formats (text files ,excel file, XML files and so on)

8. Total Cost of Ownership (TCO)

This is a financial factor of ERP software. We know that open source ERP is free for downloading and using. However we generally we will need to customize it and there is always an interesting question – what is the real cost of open source software? Without the licensing cost, we can spend money only for making customization, consulting, support and maintenance and in this way we can save money.

9. Community

The open source existence is all related with community. There is a group of developers that always work on the project and creates a wide range of source code without being paid. All well-organized open source projects should have a community providing everything needed to the project, including development but also documentation, as fixing and support forums. An open source project with a large community of developers and users has a greater probability of success.

While choosing any open source ERP for the project these factors will be the most important factors.

4. Business Intelligence Systems

In almost all companies the operational tasks are based on operational systems supported by Enterprise Resource Planning (ERP) systems. This kind of systems provides the main functionalities for their daily tasks to deal with customers, suppliers and some internal operations. However, the data stored on those systems during the time is growing and with the competitiveness of the markets the companies need to create new mechanisms to analyse the stored information and decide different kind of strategies in order to face with the concurrency. Unfortunately, the operational systems are designed to store normalized information and are not oriented to provide analytical reports. To solve in part this limitation, the data of these operating systems is stored in a database designed for this purpose. This kind of databases is called “Data Warehouses” [22] [23], from which the analytical tools consume the data. Business intelligence (BI) is a group of software programs, applications, and technologies that support to gather an increasing data from business operations, store them and analyse them to get better business decisions. BI systems generate several activities of decision support systems, query and analytical reports, and online analytical process (OLAP). The transformation of data and all of these activities are accomplished through the data lifecycle of Business Intelligence illustrated on Figure 3.

Traditionally, considering the amount of data, the need to analyse the historical analytical data from different point of views, the OLAP [22] operate on the information from the Data Warehouses, pre-calculating and processing all combinations of the group-by operator and materializing them in Multidimensional Structures or Data Cubes. Based on these Multidimensional Structures the Business Intelligence (BI) tools provides a simple and easy way to give analytical and graphical reports to decision makers in order to provide different mechanisms to get better business decisions

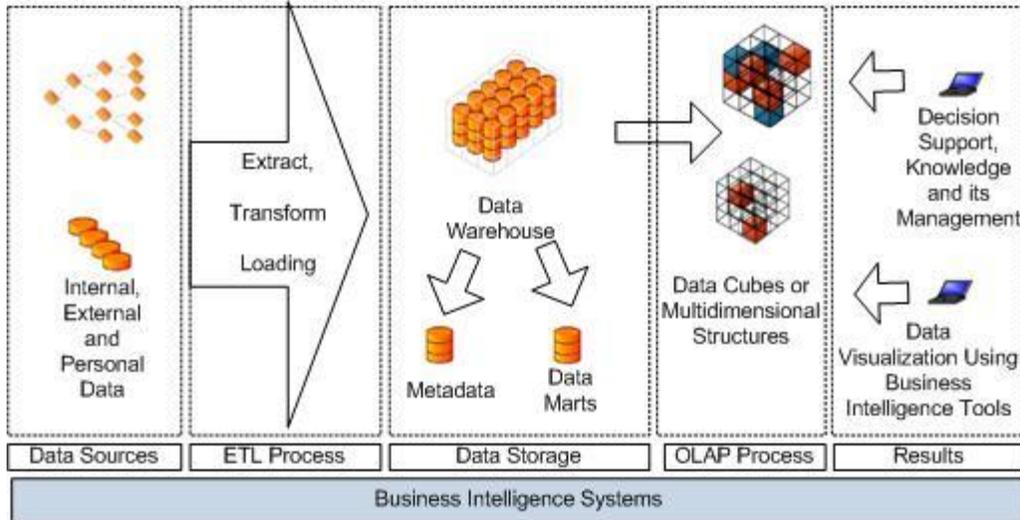


Figure 3: The global structure of the BI Systems

In the figure 3 we present the global view of the BI Systems. In order to provide information to present to decision makers using the BI tools, the process starts from the data stored in the operational systems and transformed into a new formats (using the multidimensional Model) [23] [24] and stored in the Data Marts of the Data Warehouse. This type of integration is traditionally realized by the ETL (Extract, Transforming and Load) process [23]. Based on the data stored in the Data Warehouse the OLAP tools generate all the combinations of the group-by operation, creating the Data Cubes (or Multidimensional Structures). The BI tools provide a simple and easy way for visualize this kind of structures. With BI tools the decision makers are able to visualize the data in a different and better perspective in order to get better decisions for their business.

In the corporations and organizations internal staff use a huge volume of data extracted from the ERP systems, databases and other data sources and now business intelligence is a fundamental and vital opportunity for strategic and tactic information management for decision makers. “Business intelligence is becoming a strategic tool to help people lead, measure, optimize, discover, and innovate in order to change the landscape of their organization” [25].

Gartner [26] forecasts: “Regardless of the impact of the recent economic downturn on organizations, there is a general consensus that the business and economic landscape will not return to the pre-crisis state — but will eventually stabilize into something different. New business patterns will emerge new ways of doing business will be exploited by early adopters to gain competitive advantage. Business Intelligence (BI) will play a significant role in helping

your organization identify and understand the potential impact of these new patterns, providing the vital insight that business leaders will find invaluable. New ways of using BI, coupled with the increasing sophistication of analytic applications and the emergence of mash-ups, offer huge potential for IT to help build the business in this new world". All of these allow us to conclude that Business Intelligence Solutions are getting to be very important tool to analyze information and make on time decisions for successful business strategy. Hence Business Intelligence technology is really calling world economy to make innovative changes in business decisions and more companies are trying to use this technology to be competitive.

4.1. Open Source BI vs. Commercial BI

As they are applied in different information management fields, open source tools are applied to business intelligence solutions. Open source BI is becoming well known and an important component for business level applications. Open source BI solutions such as Pentaho [36], Spago[41] and JasperSoft [42] have been developed by community-driven tools with professional support for enterprise adoption and growing demand. Moreover open source solutions give more organizational flexibility, in this way organizations are not limited and restricted with proprietary vendor's platforms [27]. Indeed the main attractive reason to organizations select open source business intelligence solution is that flexibility. It allows developers in organizations to enlarge business intelligence tools. That flexibility doesn't give this advantage only to IT staff and developers, but also to different users from the company to use the reporting tools that are available in open source BI packet.

According to Manoy Jasra [28] "if you're interested in bringing business intelligence tools to your company, but are concerned about both the cost and the complexity of it, consider open source BI. It will provide you with the flexibility you need and won't bust your budget. Make your life a little easier, and your company more successful, with open source business intelligence software". If we consider that open source tools are customizable open source BI is usually easier than most commercial software to customize. Depending upon our internal company resources, we can make our own changes, can ask for help to the open source community for improvements, hire a developer, or can do all these considering the situation.

According Gartner's research [29] many commercial business intelligence (BI) vendors can pride with their long history and large marketing budgets. But open source software that are less popular, such as Pentaho, JasperSoft, Actuate BIRT, or Spago have already started to strengthen their condition in the market. Now not only "report writers" for small businesses but also bigger enterprises are getting aware of available and possible open-source BI options [29]. Selecting the right Open Source Business Intelligence tool for new, small business just takes a little research. This research can be made by just someone in the company who has enough knowledge about requirements of the company to find the suitable open source software. Knowledge required by this process for Open Source BI can be such as traditional BI skills, enough knowledge on working with BI concepts, dimensions, metadata, metrics, open source system skills, etc.

Open source business intelligence software depends on the requirements of the company. Big companies usually have enough IT professionals to handle the complexity and broadness of business intelligence software, but small companies, which have small IT budgets and fewer employees, don't have that advantage. However, these small companies have chances to get and profits from the BI tools by the flexibility of open source BI software. Certainly the advantage of this flexibility is that open source software is freely available, which allows companies to customize it for business solutions and for this step big investment is not needed. However, Davy Nys [30] shows that there are five myths of Open Source BI which customers fear to choose OS BI tools:

- "Nobody's using it."
- "Open source licenses are risky."
- "Open source BI will cost me more if I'm 'standardized'."
- "Open source BI is an all-or-none decision"
- "Only Innovators and forward-looking organizations are using open source BI."

Answering to these opinions we elaborate a brief research of some open source business intelligence customers (Table 1). As we can analyze we conclude that open source BI software is been used by some large companies, which demonstrate that these solutions strengthen their market position.

Open source BI Customers	Description	Open source BI Software
Sun Microsystems	High Technology	Pentaho
Swissport	Global Aviation Ground Services	Pentaho
	Company	
NAVAIR	U.S. Naval Air Systems Command	Pentaho
University of Montreal	Canada's Second Largest University	Pentaho
	National Real Estate Brokerage	
ZipRealty	Leading European Health Care	Pentaho
Otto International	Organization	Pentaho
	Leading Multimedia Company	
Lifetime Networks	Digital Media Leader	Pentaho
DivX	Public university, USA	Pentaho
University of Nebraska	Private research university, USA	JasperSoft
Vanderbilt University	Financial Services	JasperSoft
Zurich Financial Services	Healthcare services	JasperSoft
McKesson Corporation	Global IT solutions and business	JasperSoft
Telvent	Information services provider	JasperSoft
Siemens	Electronics	RapidMiner
Philips	Electronics	RapidMiner
Honda	Automobile	RapidMiner
Lufthansa	Airlines	RapidMiner
T-Mobile	Mobile Operator	RapidMiner

Note: for more information about OS BI solution see Appendix B

Table 1: Open Source BI customers

4.2. Requirements for Open Source BI Solution

One of the most important factors that should be considered for open source Business Intelligence is the total cost of ownership for software [27]. The TCO concept calculates the costs related with the acquisition, deployment, operation and control of a BI solution. Though TCO measurement methods differ for different BI solutions, they are based on requirements and business needs of the company. Generally TCO is the well-known financial calculation for evaluating the BI solution. According to Gartner [31], we must “know our mission and use TCO to prioritize, guide and test progress toward meeting enterprise goals”. Sunil Mistri [27] states that the TCO factors that contribute to most BI application’s cost are such as staffing (50%), software (20%) and hardware (8%). So if we consider that the software costs are the 20% of total costs of BI applications, using open source BI software we can save this money.

There are several factors like user requirements, complexity of development and scalability of the solution that can be analysed to measure the total cost of ownership. According to Sunil Mistri [27] there are five factors that affect open source BI solution are:

- BI product selection and user requirements,
- Complexity of development,
- BI project timelines,
- Product support and third party support,
- Performance and scalability.

BI product selection and user requirements

Today in every company IT departments can identify and gather business needs and requirements for their purpose that they don’t need to contact to the product owner because organization can define their requirements without purchasing BI tool. Depending on these requirements companies can choose internally accepted open source BI solution.

Complexity of development

While developing a BI solution the extension and development is dependent not only on the user’s requirements, but also on the product features and flexibility of the platform. Comparing with commercial BI, open source BI products are based and built on the open source technologies that are available in the public domain. The resources and opportunities for

development and maintenance of the software can be easily found. Most open source BI platforms permit to make desirable modifications getting feedback from BI users. So the flexibility of the product allows decreasing the complexity of development and this factor can be a good criteria for selection open source BI.

BI project timelines

Almost every BI solution requires organized and coordinated team endeavor to develop and complete the solution on time. Selection of proprietary and open source technology influences the human cost. In case of the open source tool, companies should consider and attract all developers and supporting people of the solution such as database administrators and testers to get more knowledge and learn the selected technology. Sunil Mistri [27] states that “open source BI products have simplified the use of tools and added features that can reduce the development timelines”.

Product support and third party support

Support is a vital factor for open source projects. If we research a little we can define that comparing with proprietary vendors many open source companies make support for the products at very low subscription prices. But we can realize this wish with the support of community which the level of this support depends on the community activity.

5. Work Developed

In rapidly changing IT industry, Business Intelligence (BI) and Management Information Systems have reached a new level of importance for decision makers. Information Systems play progressive role in improving the competitiveness of the business. They handle repetitive tasks, guide and advance business processes. In order to increase the efficiency of accountants and different people of functional departments we have to understand linkage issues to be able to implement the communication between architectures. In this context ERP system makes the ideal platform for this communication and integrates different activities from product planning, purchasing, inventory control, distribution to order tracking. However it is important to notice that every enterprise has a specific list of requirements for Management Information System software application and it is impossible to define a single list of requirements appropriate for every enterprise. Enterprises differ for its IT infrastructure, required level of security, cash position, capabilities, etc. All these factors affect a software solution. On the other hand the number of BI users is constantly increasing. But depending on IT budgets and considering BI's high TCO, some new small and medium sized businesses meet actual problems to implement BI solutions which are related with the cost of software.

Considering that solutions of Business intelligence systems and Management information systems are expensive, the purpose of this project is to investigate low cost solutions which are available as open-source. Open source software (OSS) is more than an opportunity for small and medium sized businesses to make cost savings and to bring in advanced technologies that in the past were available only to large organizations through expensive, proprietary packages.

Concretely the objective is to use suitable environments free of charge in order to integrate Business Intelligence environment with a Management Information System. Under the realization of the project there are some objectives that are outlined in the following points:

- To choose open-source ERP and BI software;

The choice of software has to meet some requirements. ERP software should fit to our needs and objectives, for example, it should be possible to access the relational model of the database. Open source BI software should provide us with the necessary tools to create a business intelligence environment.

- To implement the system integration of the chosen ERP and BI software:

- Access to the relational model of ERP that will allow to identify data to be used;
- Create dimensional Data Warehouse. It is undoubtedly true that each data warehouse is unique because it must adapt to the needs of business users in different functional areas, whose companies face different business conditions and competitive pressures. So before building data warehouse we have to specify our business needs.
- Implementing business intelligence over open-source management information system. This task is the target goal of entire thesis.
- OLAP analysing and reporting. Reporting is a broad BI category and we will get subject oriented reports and analyses.

5.1. Software selection

One of the first project steps is the selection of the open source software that we will use to develop the integrated system. Searching on internet, we can find that there are a lot of available open source solutions that we can use for ERP and BI (See Appendix B). Consulting the SITE – For Open Source ERP [32] we can verify that there are tens of different solutions available on the market. The large number of available OS ERP and OS BI ready to use on the market makes the choice of a solution, a process can be time consuming

Considering that it was not the objective of the present project to experiment and analyse the different OS solutions existing on the market, we choose two solutions that meet the needs and requirements to develop the project and have an active community of users and a large number of downloads. While downloads are not an indicator of the project quality, a large number of downloads shows the level of user community adoption of the project. This option was due to time limitation for the execution of the project and considering that it was more important to focus on the integration of tools to create a system than in testing capabilities of different solutions available on the market.

5.1.1. Dolibarr ERP/CRM

From the large number of open source ERP/CRM solutions available today we decided to choose Dolibarr ERP/CRM since it provides the characteristics needed for the project (Table 2), namely it is based on an open source database (MySQL), as well as a significant user community

and a large number of downloads (more than 1000 weekly downloads) [33]. Beyond that, Dolibarr ERP/CRM is one of those ERP systems where everything is centralized in a single, simple and flexible enterprise management system.

CHARACTERISTICS	DOLIBARR ERP/CRM
-Architecture	Apache,MySQL,PHP
-Size	Small and Medium
-Customization Complexity	Simple to develop and customize
-Functionality and modules	Rich
-Products and services management	Yes
-Stock management	Yes
-Bank account managements	Yes
-Invoices managements	Yes
-Orders managements	Yes
-Payments managements	Yes
-Shipping management	Yes
-Emailings	Yes
-User Interface	Simple, clear
-Programming Language	PHP
-Scalability	Yes
-DB independence	Yes
-Development Activity	Yes
-Development status	Stable
-Integration	Yes
-Migration	Yes
-Localizable	Yes
-Community forum	Yes

Table 2: Dolibarr ERP/CRM characteristics

Dolibarr ERP/CRM - Enterprise Resource Planning and Customer Relationship management (Figure 4) is software built by modules addition that allows to enable the needed features that can be used to manage small or medium companies, freelancers or foundations [34]. We can say Dolibarr is an ERP or CRM, depending on which modules we want to activate.

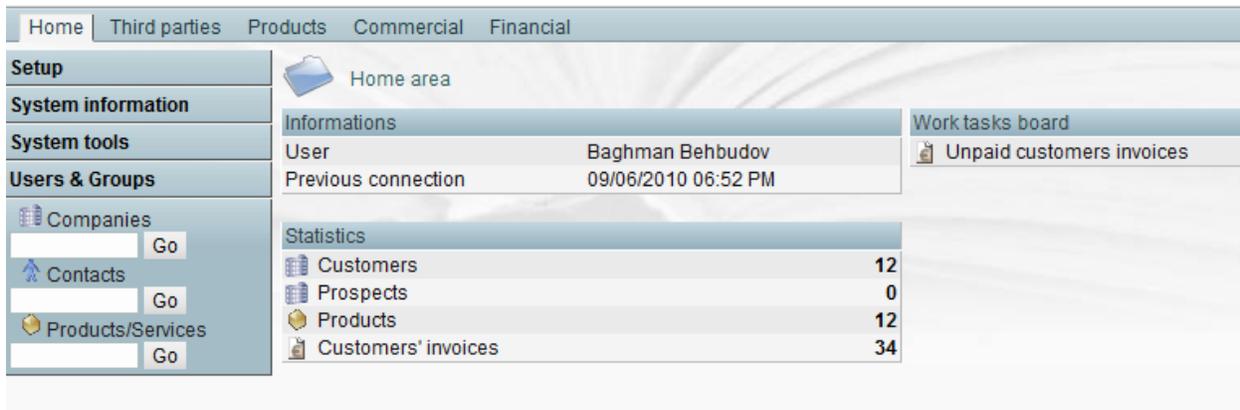


Figure 4: Dolibarr ERP/CRM

Dolibarr is an Open Source project built on a WAMP, MAMP or LAMP server (Apache, Mysql, and PHP for all Operating Systems). According to [34] the main difference between Dolibarr and other ERP or CRM software (such as Tiny ERP, ERP5, OpenBravo, OpenERP, Neogia, Compiere, etc) is that full packet was created to be simpler in all steps being simple to install, to use and to develop. Dolibarr is a full package with auto-installers. Any new users can install Dolibarr and its core architecture (Apache, Mysql, PHP) without any technical knowledge. So it makes exploitation easier. Dolibarr is designed for users looking to find software to manage activity of their company or foundation, easily and everywhere. Modularity of Dolibarr allows having software that fits our needs.

As indicated above, Dolibarr ERP/CRM has several modules which can be activated that include: products and services catalog, stock management, bank account managements, customers, prospects and suppliers management, invoices managements, orders managements, payments managements, shipping management among others.

5.1.2. Pentaho BI

The search we made of the available OS BI on the market has identified a group of alternatives like Pentaho BI, Japersoft, Openi, SpagoBI. From these available solutions we decide to choose Pentaho BI as it offers a full set of functionalities to implement our project (Table 3). However, this software has a large community of users and a high number of downloads.

CHARACTERISTICS	PENTAHO BI
License	Several components: GPL 2, LGPL, MPL (Mozilla Public License)
Technology	J2EE, CAS, ACEGI, JBoss Portal
Database Environment	JDBC, MySQL , Firebird/InterBase, IBM DB2, Microsoft SQL Server, Oracle, PostgreSQL (pgsql)
Programming Language	Java, JavaScript, JSP, XSL (XSLT/XPath/XSL-FO)
Reporting	Yes
Dashboard	Yes
Reporting	Yes
OLAP Analysis	Yes
Data Mining	Yes
Data Integration	Yes
ETL	Yes
Alerts	Yes
Repository	Yes
Localized & internationalized	Yes
Ease of use	Medium
Export formats	PDF, HTML, XLS, RTF, text
Web capabilities	Admin, report retrieval, parameter support, drill-through, hypertext links
Report designing	Visual report designer
Modifiable system	Yes
Desirable platform	Yes
Installation	Easy
Scalability	Very Scalable, easy to embed into our architecture
Community activity	Rich
Downloads	6700 downloads a week

Note: Information is taken from the Pentaho's official web site and Source Forge

Table 3: Pentaho BI selection criteria

Pentaho BI is open source BI suite that can be installed on Windows or Linux. Pentaho BI combines applications for reporting, analysis, dashboards, data mining and business intelligence platform. Pentaho BI includes various components that can be used as a suite or as a stand-alone application [36]:

- Pentaho BI platform provides the architecture and infrastructure required to build BI solutions, providing core services like authentication, logging, auditing, workflow, web services, and rules engines. It is the platform that integrates reporting, analysis, dashboards and data mining components to form a sophisticated and complete BI platform;
- Pentaho Data Integration (Kettle): provides the tool for the Extract, Transform, and Load (ETL) processes and it is generally used in data warehouses environments;
- Mondrian: an OLAP server that enables to analyze datasets stored in SQL databases using a multidimensional format to present results.
- Pentaho Reporting: provides tools for the creation, generation and distribution of reports;
- Weka: provides a set of tools for machine learning and data mining.

Pentaho also includes a collection of editors, viewers and administration modules that provides a graphical user interface to facilitate the interaction with Pentaho components, like for example, the Pentaho Administration Console that provides a centralized administration that allows Pentaho deployments. This console holds many administrative tasks like managing databases connections (Figure 5).

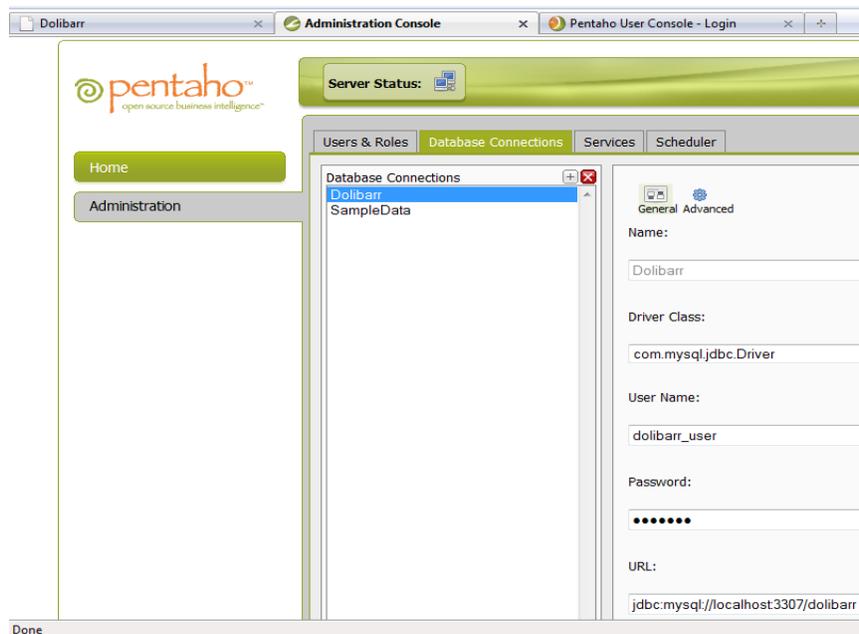


Figure 5: Pentaho Administration Console

5.2. Implementation

As previously described we have chosen open source Dolibarr ERP/CRM and open source Pentaho Business Intelligence to make an integrated system of these tools. The architecture of the implemented system is presented in figure 6. In this architecture, the relational database of Dolibarr ERP/CRM is accessed and transformed into a new format to be stored in the data warehouse through ETL process. Based on the data stored in the Data Warehouse, the OLAP tools are used to generate reports and analyses.

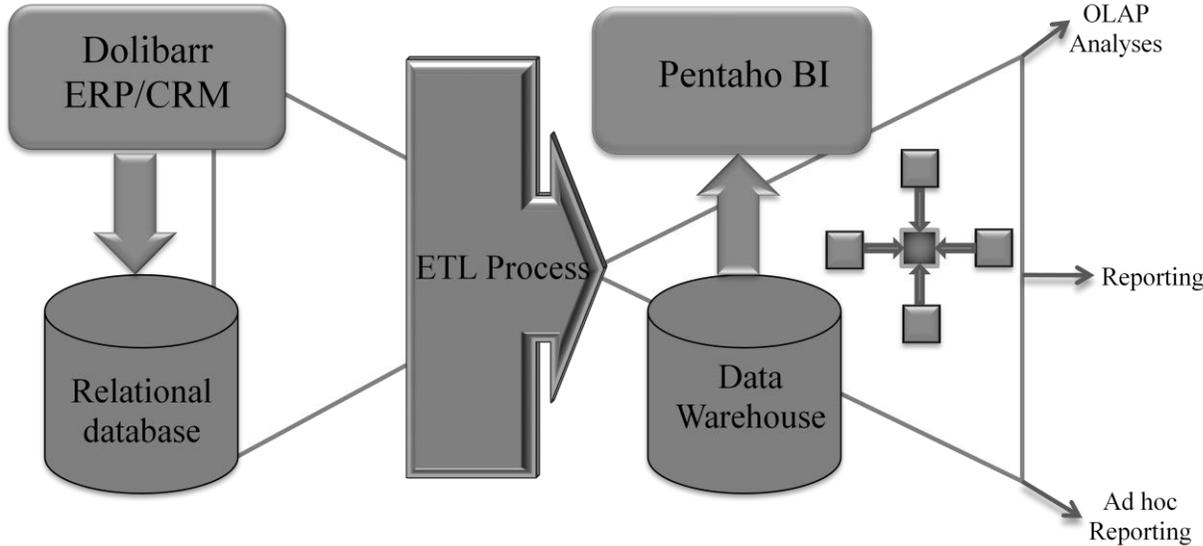


Figure 6: System model

Dolibarr ERP/CRM software has a single database that contains all data for various modules available like products, stocks, customer orders and invoices among others. Different modules of Dolibarr ERP/CRM can be activated to define which functionality is enabled in software (Figure 7).



Figure 7: Dolibarr ERP/CRM Modules

For the purpose of the present work, we consider the case of an enterprise that wants to analyze its sales to different organizations of different countries and uses Dolibarr ERP/CRM system to register sales information. Considering this circumstance we have activated and integrated some Dolibarr Modules:

- Third parties module from CRM that contains companies and contacts' management;
- Invoices from Financial Modules family which contains sales, invoices and credit notes' management for customers;
- Products module from Product Management;

Each of these modules corresponds to a main functional area of the company. With these activated modules we can register the information about the sales and general information like customers, products, countries, cities.

This example will be the base to experiment the integration of ERP with the BI suite, in order to validate the practicability to integrate these two software.

Based on sales information, we considered the following requirements for the business intelligence:

1. Sales statistics
2. Ability to generate reports like total sales by product, by location, by customer and by date
3. Top customers, top products and more.
4. Generating ad-hoc reports and another OLAP analysis.

After defined the requirements, we need to access and identify the data sources that will be used by Pentaho BI. Using Dolibarr ERP/CRM we can get the sales information of the company, since it uses an open source database and it is possible to access to the data. In order to identify data source for the data warehouse, we had to explore deeply and study the relational model of Dolibarr ERP/CRM which has more than 200 tables. Taking into account the activated modules we had to analyze the product sales of the company. From this analyse, we have identified the tables that store sales data and have selected the sub-model of whole model in order to get the data (Figure 8).

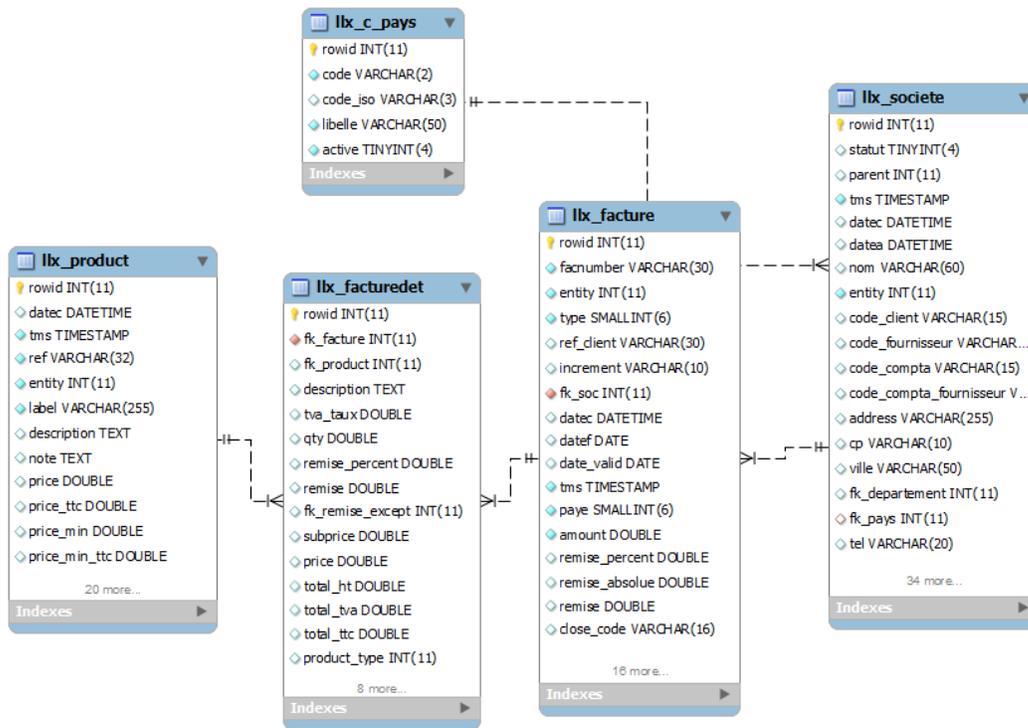


Figure 8: Sub-model of Dolibarr Database

The sub-model consists of 5 tables with data about sales. The main data for our work are:

- **ilx_societe**: data about customers, namely name (company or individual), address, zip code, country, city, phone, e-mail, capital and type;
- **ilx_c_pays**: stores data about countries, namely, code and name;
- **lx_product**: this table keeps the information about product, namely, name, price and tax;

- llx_facture: stores data about invoice, namely, invoice number, date of sales, date of payment and total sales;
- llx_facturedet: stores data about details of invoice, namely, VAT, the quantity of product, prices, sales and sales value.

Based on this relational model, we define a target schema that has six dimension tables and one fact table. The fact table should contain all sales by product, date, customer, country, postcode and city (Figure 9).

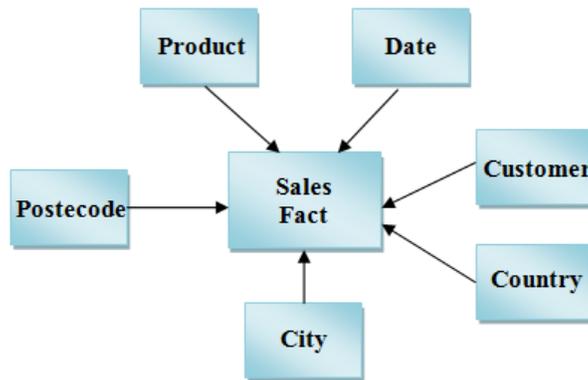


Figure 9: Data warehouse schema

After identifying sources of necessary data in MySQL database of Dolibarr ERP to be used to create BI solution, we use different Pentaho BI tools to create the star model, dimensional data warehouse, ETL (Extract Transform Loading) process, publish to BI server repository and make some OLAP analysis and reporting. Figure 10 present a schematic representation of tools and data flow of the implemented BI.

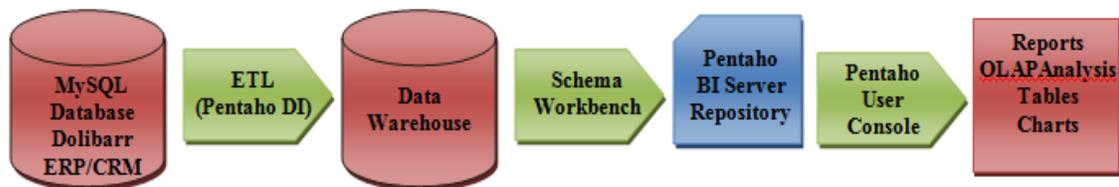


Figure 10: Tools and data flow of the implemented BI

The first phase of the process involves the creation of data warehouse and the loading of data to it from tables of Dolibarr ERP/CRM relational database (ETL process). These tasks can be made using Pentaho Data Integration Tool where this process can be done through a graphical

interface. Pentaho Data Integration (PDI), also called as Kettle, is the element of Pentaho BI which provides the Extracting data, Transform and Loading (ETL) processes. PDI facilitates the creation of each process with a graphical tool without the need to write code.

For the purpose of our work, we define the ETL process shown on figure 11. This process involves several transformations.

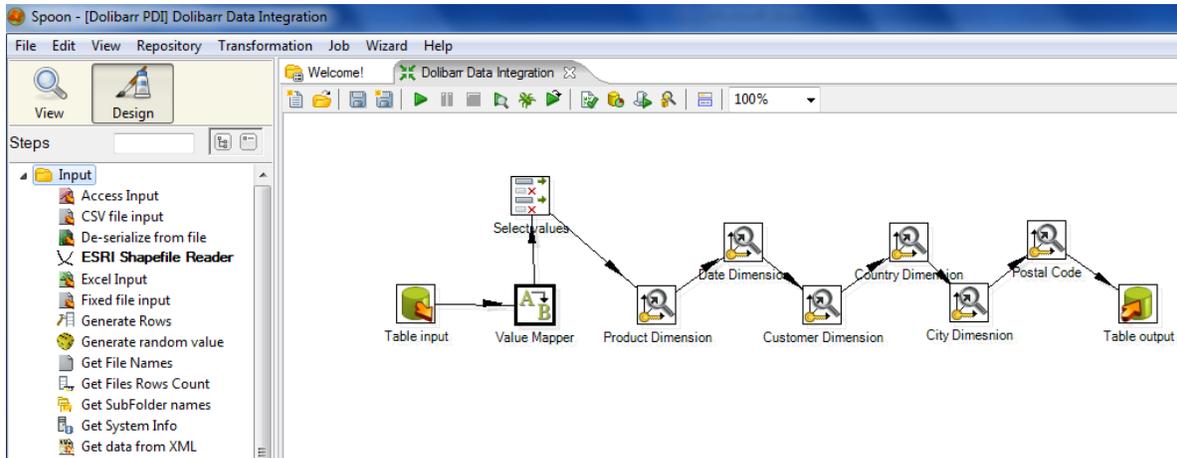


Figure 11: Pentaho Data Integration Tool

The first step defined in PDI graphical interface, identified as “Table Input”, is used to read information from data source, using a connection to a database. The user interface allows the definition of the SQL command that will be used to read data from external data source (Figure 12).

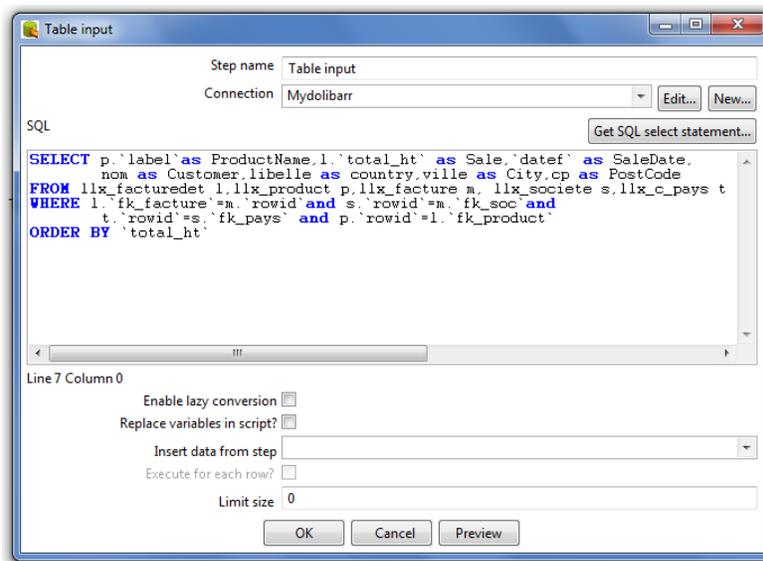


Figure 12: Pentaho Data Integration Table Input step

The second step “Value Mapper” is used to change string values from one value to another. We used it to translate the name of countries that were in French in ERP database (Figure 13).

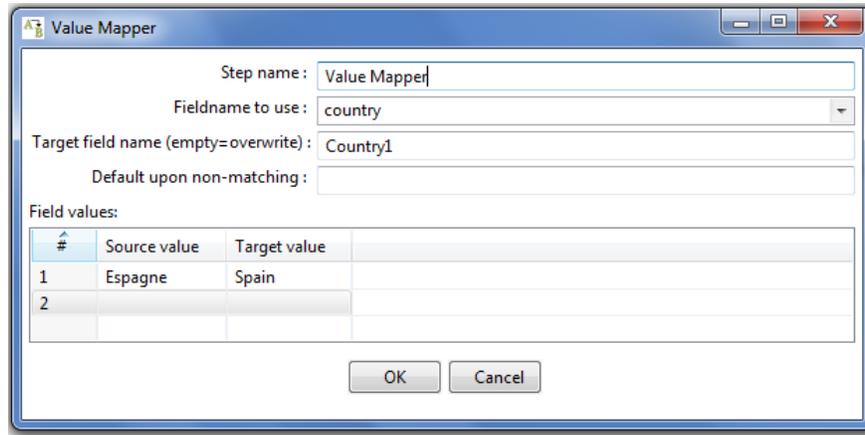


Figure 13: Pentaho Data Integration Value Mapper step

The next step is “Select Values” which provides us selecting, renaming fieldname and changing data types (Figure 14). In our work we used this step to change the date format.

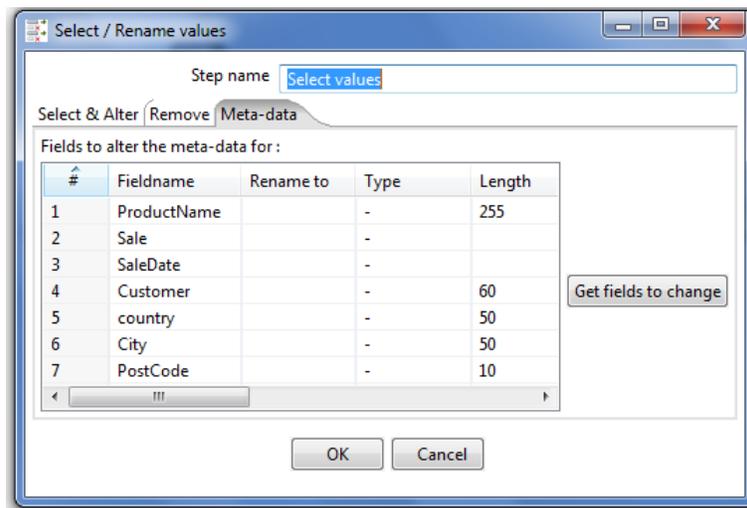


Figure 14: Pentaho Data Integration Select Values step

The next six steps are used to create dimensions to store information using the “Combination Lookup/update”. These steps allow to create/maintain a technical key out of data with business keys. Using this step we define six dimensions in our experimental project: product (Figure 15), customer, date, country, city, and postcode. In order to create each dimension, it is necessary to indicate the dimension name, database connection, the target table

name and technical key for this table. This step can also be used to create dimension tables in the data warehouse database. There exists an option that can be used to create SQL script to build the dimension table and provides us to execute this SQL script.

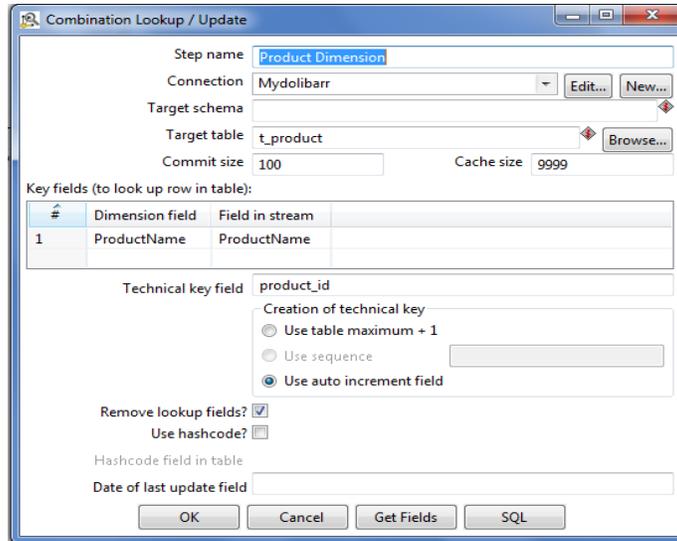


Figure 15: PDI Combination lookup-update step for product dimension

The last step is “Output Table” that allows to load data into the data fact table (Figure 16). In this step we have to indicate the database connection where we want to store data and the name of the fact table where we want to save data.

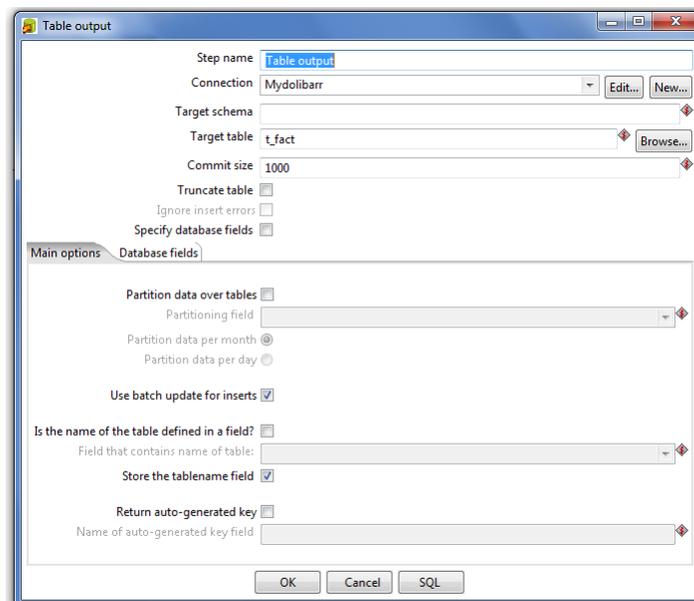


Figure 16: Pentaho Data Integration Table Output step

After creating this graphical view we execute the defined process and it creates a star model and executes ETL process, loading data to data warehouse from Dolibarr ERP/CRM database. Figure 17 shows the dimensional model of sales that have been created by PDI.

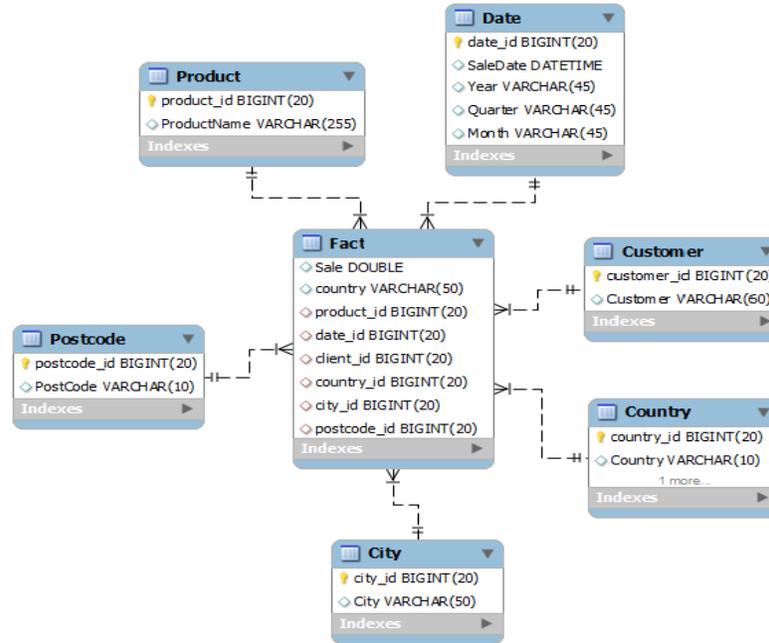


Figure 17: Star model of sales analysis

After creating the data warehouse and defined the ETL process, the next phase is about the creation of OLAP cube to publish to BI server. We can make this process by another Pentaho design tool that is called Pentaho Schema workbench.

Pentaho Analysis provides Mondrian OLAP tool. The Pentaho BI Platform presents this Mondrian engine as a part of system architecture. This tool also creates MDX (MultiDimensional eXpressions) to make ROLAP queries. These schema files are XML metadata models that are created in a specific structure used by the Mondrian engine. These XML models can be considered cube-like structures which utilize existing fact and dimension. It does not require that an actual physical cube is built or maintained only that the metadata model is created. This XML based metadata structure can be created using a text editor or the Schema Workbench [37]. So using Schema Workbench we create our ROLAP Cube (Figure 18).

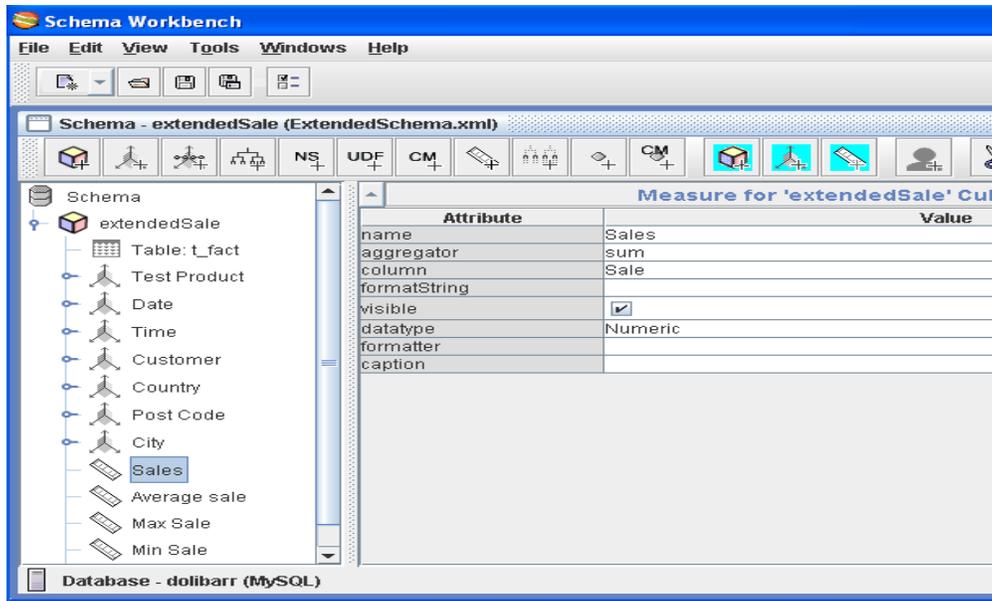


Figure 18: Schema Workbench /ROLAP Cube

After the schema is created and published to the Pentaho BI Server, we can begin to define the reports and OLAP analysis using Pentaho Analysis View. For this part of process we have to use Pentaho User Console. The Pentaho User Console is the standard Web interface for the Pentaho BI Platform, and includes interactive elements of Pentaho Reporting and Analysis. The User Console is customizable through manual editing of configuration files. Using Pentaho User Console and after choosing appropriate scheme and cube, Pentaho shows default page with a JPivot report (Figure 19).

					Measures			
Products	All Years	All Customers	All Countries	All Cities	• Sales	• Average sale	• Max Sale	• Min Sale
[-] All Products	[+] All Years	[+] All Customers	[+] All Countries	[+] All Cities	9,777,450	5,492.949	18,000	2,160
Aquamarine	[+] All Years	[+] All Customers	[+] All Countries	[+] All Cities	537,120	4,439.008	7,200	2,160
Bronze	[+] All Years	[+] All Customers	[+] All Countries	[+] All Cities	657,800	4,698.571	8,800	2,640
Coral	[+] All Years	[+] All Customers	[+] All Countries	[+] All Cities	604,080	4,027.2	6,480	2,160
Diamond	[+] All Years	[+] All Customers	[+] All Countries	[+] All Cities	1,109,500	6,488.304	12,600	4,200
Emerald	[+] All Years	[+] All Customers	[+] All Countries	[+] All Cities	761,300	5,399.291	13,800	2,760
Gold	[+] All Years	[+] All Customers	[+] All Countries	[+] All Cities	1,716,600	8,135.545	18,000	3,600
Jade	[+] All Years	[+] All Customers	[+] All Countries	[+] All Cities	538,000	4,446.281	10,000	2,400
Jasper	[+] All Years	[+] All Customers	[+] All Countries	[+] All Cities	667,850	4,736.525	9,500	2,280
Sapphires	[+] All Years	[+] All Customers	[+] All Countries	[+] All Cities	820,000	5,430.464	10,000	3,000
Silver	[+] All Years	[+] All Customers	[+] All Countries	[+] All Cities	699,300	4,631.126	12,600	2,520
Tanzanite	[+] All Years	[+] All Customers	[+] All Countries	[+] All Cities	632,700	4,487.234	9,500	2,280
White Gold	[+] All Years	[+] All Customers	[+] All Countries	[+] All Cities	1,033,200	7,327.66	11,200	3,360

Figure 19: New Analysis View

This type of report has some tools to analyse the sale such as drilling into detail from summary (Figure 20), navigating hierarchies, sorting, ranking, pivoting, filtering, charting, exporting are completely dynamic:



Figure 20: Drill down method example

The Pentaho platform also supports a number of different charts through the charting engine API that wraps JFreeChart. Pentaho offers a set of different output charts that can be used to represent data from pivot table. Figure 21 shows an example of a vertical 3D bar chart of yearly sales.

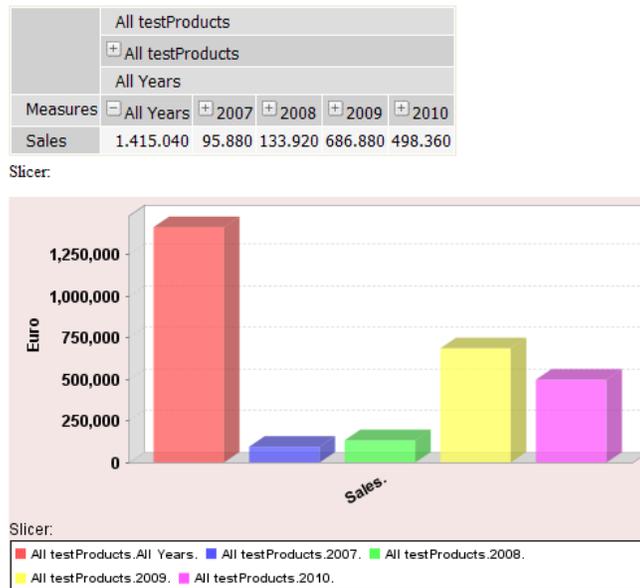


Figure 21: Sales by year in Vertical Bar 3D chart type

As we can see the Pentaho User Console gives the ability to create reports that allows the decision makers of a company to analyze data in different forms in order to support decisions. This type of report gives a set of dynamic tools to analyse data, like drilling into detail from summary, navigating hierarchies, sorting, ranking, pivoting, filtering, charting and exporting. This tool offers great freedom in the way users interact with information and derive insights from data.

A functionality that has been included in BI solutions is the capacity to create dashboards for decision makers. BI dashboards are intended to provide decision makers with an overview of business performance at a glance, using attractive and intuitive visual interface and a few simple clicks of the mouse. From a decision maker's perspective, dashboards provide a useful way to view data and information such as single metrics, graphical trend analysis, capacity gauges, geographical maps, percentage share, stoplights and variance comparisons, which are presented to the user in a format that is easy to understand and to assimilate by time pressured managers [38].

Pentaho BI Suite offers a module that let users create personalized dashboards using a graphical interface called dashboard designer [37]. However, this product is only available in the Pentaho Enterprise Edition, the commercial open source version.

Despite of this circumstance, we have verified that the community edition has some tools to create dashboards that require the creation of action sequences files. Action sequences allow developers and business users to perform advanced tasks that cannot easily be accomplished through Pentaho's design tools and user interface functions [39]. An action sequence is a XML document with the definition of an ordered set of actions to perform a task. In order to facilitate the creation and management of action sequences, Pentaho provides the Design Studio, a graphical interface to create, edit and publish action sequences. In this project, we decided to create an experimental example of a dashboard. Figure 22 shows the definition of an action file to create a chart with cumulative year sales for a given product using the Open Flash Chart Component. This interface facilitates the creation of XML document that will be used by the Pentaho framework to generate the reports.

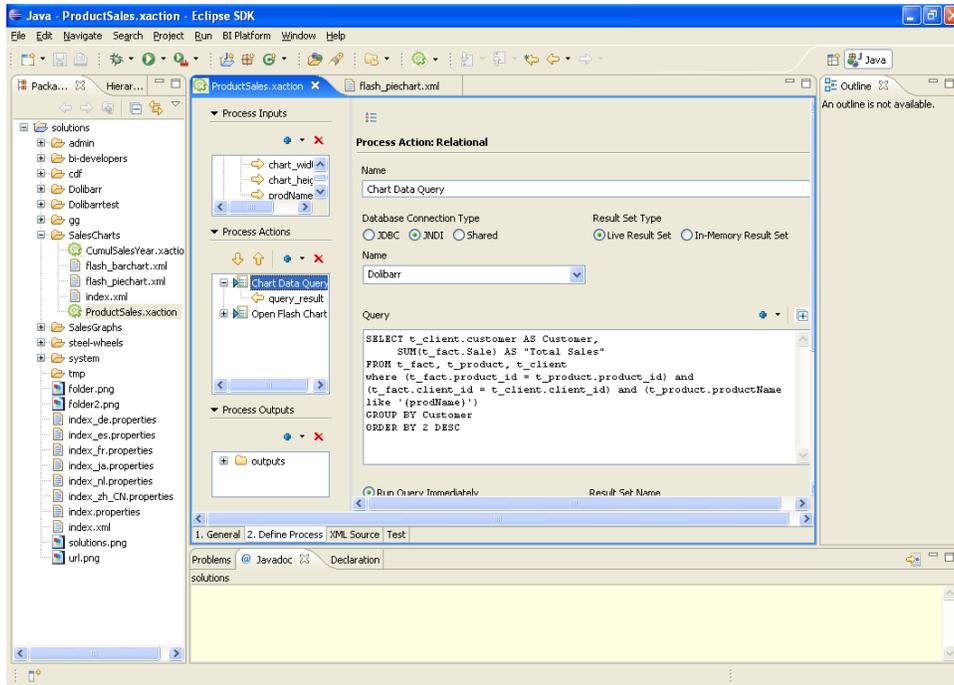


Figure 22: Creating an action sequence within Design Studio

Using action sequences files, it is possible to create parameterized reports and charts that can be integrated in a web page to create dashboards for decision makers. Figure 23 shows an example developed in this project that presents total sales by customers and cumulative sales by year for a given type of product.

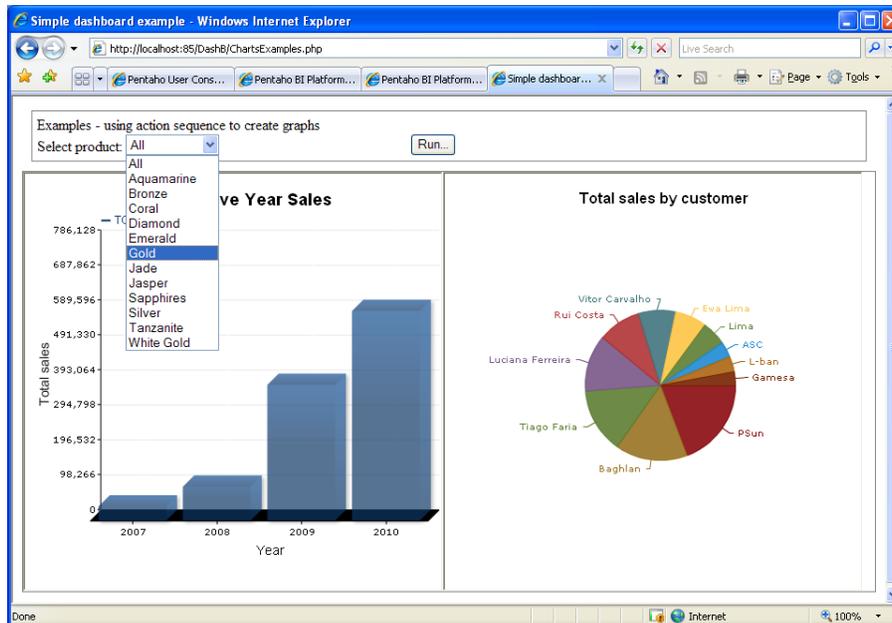


Figure 23: Dashboard example

As known, one of the advantages of OS software is the possibility to correct errors that can occur, if we have enough knowledge to do it. We faced this kind of problem in our project. While creating ROLAP cube, we have detected that the Schema Workbench interface had an execution error since it was not possible to use the option to add a key expression in order to create SQL script. In an effort to eliminate this bug we researched appropriate forums and explored the source code of this tool. After having identified the problem, we modified some lines of code (Figure 24) and recompiled the source code.

```
else if (pathSelected instanceof MondrianGuiDef.Level) {
    jPopupMenu.add(addProperty);
    jPopupMenu.add(addKeyExp);
    if (((MondrianGuiDef.Level) pathSelected).keyExp == null) {
        addKeyExp.setEnabled(true);
    } else {
        addKeyExp.setEnabled(false);
    }
    //...
    jPopupMenu.add(jSeparator1);
    jPopupMenu.add(delete);
    //...
}
```

Figure 24: Modified source code.

This kind of modifications proves that software is flexible and it allows any customization if we have a sufficient expertise to do it. However, in the same time this factor is a drawback of using community open source, since usually we rely on a community of developers and our skills to solve the problem.

5.3. Practical samples

The Pentaho User Console offers a valuable set of functionalities with a friendly and interactive interface that can be used in the analysis for the sales of a company. This tool can be used by a manager to make business decisions. We will present an example in order to demonstrate the business value of the presented solution. For this purpose, we consider a hypothetical company that sells precious stones and consider that the executive manager of the company is interested in some questions in order to analyse sales. Answering to these questions we can make some analysis and reports:

1. Analyze products sales by year:

Products	Measures				
	Sales				
	All Years				
	• All Years	• 2007	• 2008	• 2009	• 2010
All Products	9,777,450	479,400	669,600	3,434,400	5,194,050
Aquamarine	537,120	43,200		182,520	311,400
Bronze	657,800	39,600		184,800	433,400
Coral	604,080	43,200		157,680	403,200
Diamond	1,109,500	42,000	126,000	386,400	555,100
Emerald	761,300			276,000	485,300
Gold	1,716,600	72,000	144,000	633,600	867,000
Jade	538,000	36,000	72,000	168,000	262,000
Jasper	667,850	45,600		216,600	405,650
Sapphires	820,000	45,000	60,000	360,000	355,000
Silver	699,300		75,600	226,800	396,900
Tanzanite	632,700	45,600	91,200	205,200	290,700
White Gold	1,033,200	67,200	100,800	436,800	428,400

Figure 25: Cross Table of Sales by Years

2. Get the sale of each product by countries:

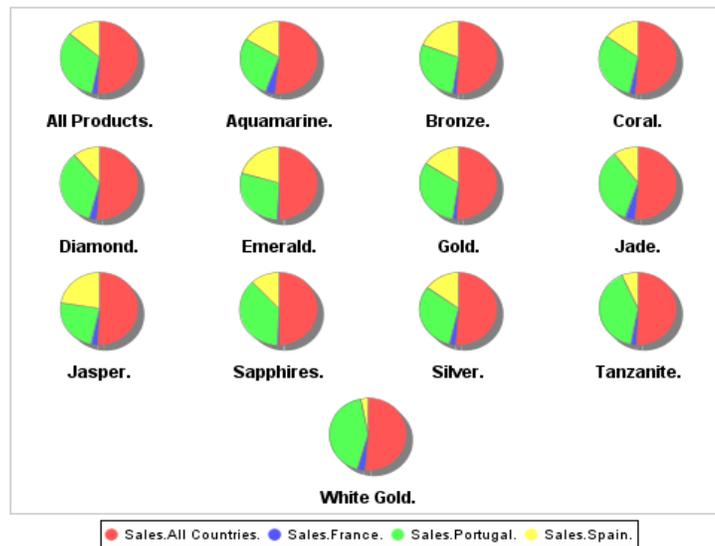


Figure 26: Sales by Country

3. Identify the most important customers:

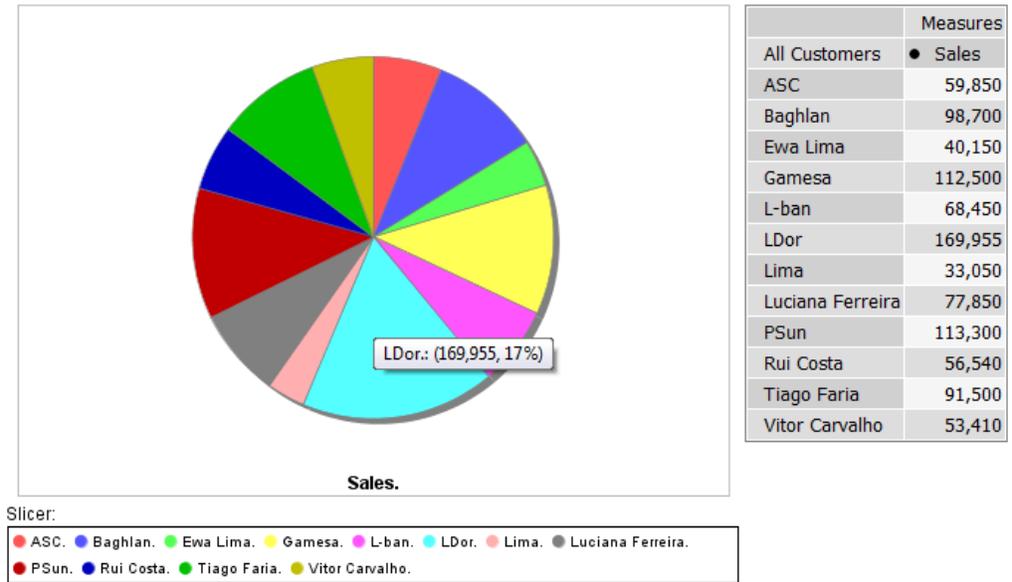


Figure 27: The most important customers

In figure 27 this tool allows decision maker to identify easily the most important customers. As we see the most 3 important customers are: LDor, PSun, and Gamesa.

4. Identify in which year sales are the highest:

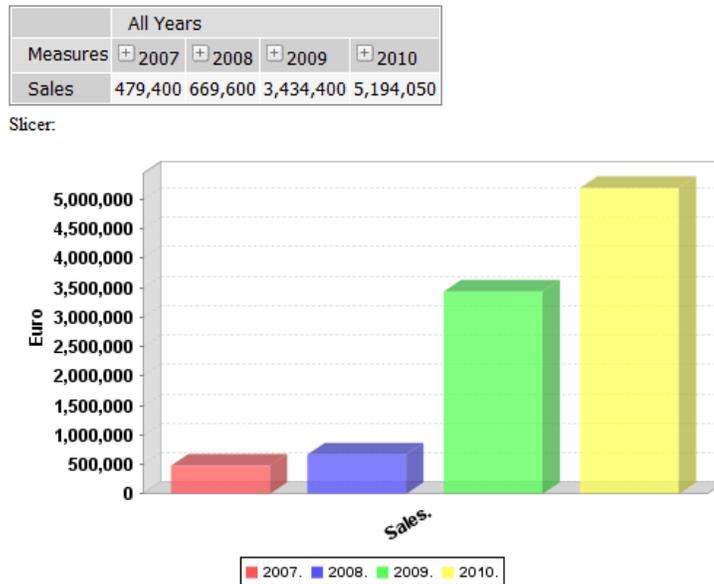


Figure 28: Sales by Year

5. Visualize all sales in 2009 by month:

	All Years						
Measures	6	7	8	9	10	11	12
Sales	326,400	327,600	595,800	641,280	741,720	319,800	481,800

Slicer:

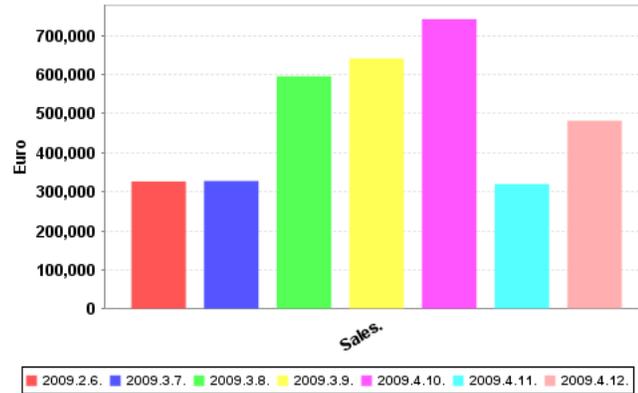


Figure 29: Sales in 2009 by month

As we see in Figure 29 the lowest sales are in November. So we can make drill down by products to see which products were sold:

Drill Through Table for Sales						
Products	Years	Month	Customers	Countries	Cities	Sales
Diamond	2009	11	Ewa Lima	Portugal	Guimarães	10,500.00
Diamond	2009	11	Ewa Lima	Portugal	Guimarães	10,500.00
Diamond	2009	11	Ewa Lima	Portugal	Guimarães	10,500.00
Diamond	2009	11	Ewa Lima	Portugal	Guimarães	10,500.00
Diamond	2009	11	Ewa Lima	Portugal	Guimarães	10,500.00
Diamond	2009	11	Ewa Lima	Portugal	Guimarães	10,500.00
Diamond	2009	11	Ewa Lima	Portugal	Guimarães	10,500.00
Diamond	2009	11	Ewa Lima	Portugal	Guimarães	10,500.00
Sapphires	2009	11	Ewa Lima	Portugal	Guimarães	6,000.00
Sapphires	2009	11	Ewa Lima	Portugal	Guimarães	6,000.00
Sapphires	2009	11	Ewa Lima	Portugal	Guimarães	6,000.00
Sapphires	2009	11	Ewa Lima	Portugal	Guimarães	6,000.00

Page 2/4 Goto Page 2 Rows/page 12

Figure 30: Drill Down by products

6. The number of sales by product in May 2010

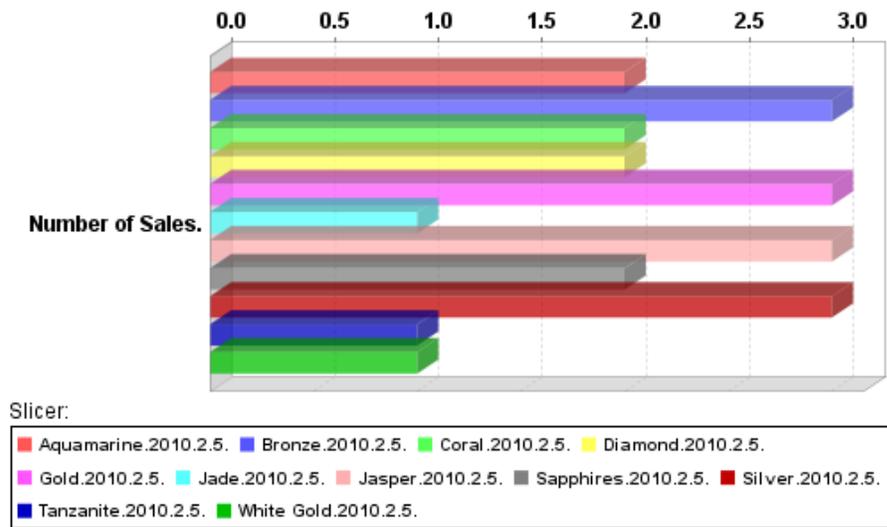


Figure 31: Number of sales in May 2010

5.4. Summary

The experimental system implemented in this project focused on developing a business intelligence system over an open source management information system, namely an open source ERP system. After selecting two largely used open-source software, an integration of these software have been developed that involves several tasks including the access to the relational database of the ERP system, the implementation of ETL process to load data in the data warehouse, publish to the BI server, the use reporting tools and the creation of dashboards. Furthermore, using this integrated system we “open the door” to small enterprises to extend implementations in the future and using this open source technology model we got the following results:

- As an experimental system we could realize successfully a system that can be very useful to small enterprises and developers. The integrated solution offers an initial set of functionalities and potentialities that provide business value to a small enterprise.

- The used tools provide an interactive graphical user interface which can be used easily but there are some significant skills that are important while integrating this system:
 - to know how to analyze the relational model
 - enough knowledge on working with BI concepts
 - knowledge of defining the data warehouse and ETL process
 - knowledge of dimensions, fact table and metrics
 - using of OLAP tools
 - programming skills to implement specific functionalities like dashboards;
- **Low cost.** There are no license fees for this combination, reducing annual license fees cost to zero. There is zero cost of scale, because as these tools are open source, also this integrated system didn't require additional licenses.
- **Accessibility.** This is another important factor. We could access to each of these tools (Dolibarr ERP/CRM and Pentaho BI) and could make modifications for our own goal.
- **Flexibility.** This integrated system meets our experimental requirements for small enterprise. If there's something we want it to do we can either modify the product ourselves or pay someone to do it for us.
- **Support.** We obtained good support from community forums and that support is often provided by the same people that wrote the original code but we should have the technical knowledge and skills to assimilate it. Furthermore, there is no compromise to get an answer in useful time and we cannot guarantee a quality service. However, we can hire professional support from companies.

6. Conclusion and Future Work

6.1. Conclusion

In general, the use of open source software offers various advantages, such as the ability to reduce costs or to avoid being dependent on a single vendor. It is therefore to be expected that more and more companies and institutions will start using open source software. Open Source Software is indeed the start of a fundamental change in the software infrastructure marketplace. Up to this point, this research has focused primarily on how to develop a business intelligence system over an open source management information system. The proposed project tends to be a lower cost implementation and focus on the delivery of well-organized management information system with an integrated business intelligence system. In order to realize expected result we selected open source tools (Dolibarr, Pentaho), integrated them and implemented a business intelligence system. During the realization of this project we have done several tasks, such as the study an open-source ERP including its database model, the definition and creation of the Data Warehouse database in order to support the information for the Business Intelligence environment. Finally, we defined and implemented the business intelligence tool over an open-source management information system.

One of the goals of our project was to show that open source software system can be an alternative solution to commercial software solutions. More concretely, we presented a solution whose target is small companies or start-up companies which can get real benefits with a lower investment on IT. Our solution can bring a business value to its target companies that do not have enough budgets to invest in commercial suites. In this way, we can provide the main functionalities that can bring business values to the companies that adopt this propose. Of course, we can consider that one limitation of this propose is that the business is not faithfully represented in the system.

However, this limitation is, at the same time, the strength of this propose since allows the target companies to realize the implementation of an ERP and BI integrated system with a lower cost. Besides, tailoring an information system to their business is not a problem for the most of small companies and start-up companies since they know that they do not have financial resources for that.

We demonstrated that open-source ERP and BI bring the main functionalities that are necessary to a business. So, we consider our approach a “lower-cost” since we save the money from tailoring the information systems to the business. Researching and analyzing of open source tools has shown that OS BI and OS MIS meet basic needs of the businesses. Besides, the greatest value of OS lays in minimizing risk of proprietary vendor lock-in and maximizing flexibility of solution through the support of open standards and source code availability.

Something that deserves to be noted is that there are also some potential drawbacks when using open source software in enterprises. Despite the existence of a community that give support, it can be difficult to find or obtain a reply to some specific questions or to understand answers. For example, when some bug or error is detected, it is usual to find answers that give instructions aimed at developers to solve the problem. However, if there is no staff with technical skills or outsource consultants, it may have to wait indefinitely for the next release with possible consequences on the performance of business. There is another factor that sometimes documentation is poor or difficult to understand since it is produced by different developers.

However, considering all these facts, using open source is still useful when companies need to have low cost solutions for their information systems. If we consider pricing for small enterprises BI solution of IBM (about 72000\$) or Oracle (about 53000\$) [30], the presented project provide a significant cost saving. In fact, if we look at the TCO factors [27], we can see that what contributes to most BI application’s cost are costs such as staffing (50%), software (20%) and hardware (8%). So with our proposed solution companies can have a business intelligence system saving 20% of software cost and part of 50% of staffing. Related with staffing, this propose have its own cost, since it had costs in production of the system. Even taking into account the production costs, we consider that this solution will have a lower cost since it is a system that can be sold to several companies taking advantage of scale factor. With this approach, the companies that adopt this system will only have the functionalities and analysis foreseen in the system. Although we expect that these functionalities cover most requirements of the business that are general by nature. However if the companies desire to tailor the system and implement specific requirements of their business, savings will be 20%, corresponding to the software cost.

Finally, we would like to mention that, due to time restrictions, the experimental system implemented in this project has been limited to the integration of two specific open-source

software and applied to only one simple business process of a business and that it is necessary to test the integration of other solutions in a real context of small enterprises. However, we consider this research demonstrates that the integration of open source management information systems and BI systems is a strategy for, at least, implementing solutions for small enterprises that can have a low cost.

This document “opened the door” to extend the implementations in future and can make a contribution to the developing body of open-source management information system that can be a good guideline to new small companies to implement a business intelligence environment.

6.2. Future Work

With this work we presented a study in order to demonstrate the viability and advantages of the integration of the ERP and BI open source software. However, as future work we consider that is important to realize the following tasks:

- To extend the system to all functional areas of business, analyzing deeply the requirements in a global way and implement it using the chosen environments and make available a suite “ready-to-use” that can be used by companies.
- To assess a real case of ROI and TCO of open source software considering real enterprise environment and conditions.
- To use the selected open source tools in order to extend this implementation for the case of any concrete enterprise

7. Bibliography

- [1] Linux Information Project, 2005. <http://www.linfo.org/proprietary.html>,
- [2] European Commission Enterprise and Industry, “Europe is good for SMEs, SMEs are good for Europe”, 2008 edition,
http://ec.europa.eu/enterprise/policies/sme/files/docs/sme_pack_en_2008_full_en.pdf
- [3] EC. European Charter for Small Enterprises. European Commission, 2006.
http://ec.europa.eu/enterprise/policies/sme/files/charter/2008_charter_docs/gp2008en_en.pdf
- [4] Wikiversity “ Information management using open source”,
http://en.wikiversity.org/wiki/Information_management_using_open_source
- [5] Gibson, Marcus, and David Arnott. "BI for Small Business: Assessment, Framework & Agenda." 7th Pacific Asia Conference on Information Systems. 2003. 743-759.
- [6] SAP® Support Practices Survey ,2009,
http://hosteddocs.ittoolbox.com/support_costs_survey.pdf
- [7] Third Nature, “Lowering the Cost of Business Intelligence With Open Source”, 2010
http://www.pentaho.com/download/asset_container.php?durl=lower_costs_with_osbi.pdf&furl=lower_bi_costs
- [8] OpenSource Initiative, www.opensource.org
- [9] Wikipedia, Open Source Software, http://en.wikipedia.org/wiki/Open-source_software
- [10] Eric S. Raymond, “The cathedral and the bazaar“,2001
- [11] Chamindra de Silva, “10 questions to ask when selecting open source products for your enterprise”, December 20, 2009, <http://blogs.techrepublic.com.com/10things/?p=1232>
- [12] Josep Curto Díaz “Adoption and Usage Survey: Open Source Business Intelligence and Reporting “ June 10, 2009, BeyerNetwork, <http://www.beyerresearch.com/study/10501>
- [13] ActiveState Software Inc “Open Source: What is the Total Cost of Ownership?”.
WhitePaper, March 2008
- [14] Wikipedia, Rate of Return, http://en.wikipedia.org/wiki/Rate_of_return
- [15] Investopedia ®, Return On Investment,
<http://www.investopedia.com/terms/r/returnoninvestment.asp>

- [16] Dan Woods, Gautam Guliani “Open source for the enterprise: managing risks, reaping rewards” , 2005
- [17] Navica and OpenLogic “Open Source Return on Investment: Achieving the Financial Promise of Open Source Software” A Joint Whitepaper OpenLogic, Inc., December 2006, http://go.openlogic.com/pages/start/download-whitepapers/index.html?Campaign_Id=1301&Activity_Id=2861&rsc=Open.Source.ROI.Whitepaper.pdf
- [18] Nah, F. and Lau, J. 2001. ‘Critical factors for successful implementation of enterprise systems’. *Business Process Management Journal*, 7 (3): 285-296.
- [19] ERPwire, “What distinguishes an Open Source ERP from a commercial one?” <http://www.erpwire.com/erp-articles/commercial-and-open-source-erp.htm>
- [20] “SME Open Source ERP” , January 12, 2006 http://www.erpsoftware-news.com/open_source_erp/
- [21] Technojos, January 8, 2008, <http://opensourceerp guru.com/2008/01/08/10-evaluation-criteria-for-open-source-erp/#more-9>
- [22] Berson, A., Smith, S. "Data Warehousing, Data Mining, and OLAP (Data Warehousing/Data Management)", Computing McGraw-Hill, 1997.
- [23] Kimball, R., Ross, M. "The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling", Wiley, 2 edition, 2002.
- [24] Inmon, W., "Building the Data Warehouse", Wiley; 2nd edition, 1996.
- [25] Microsoft Dynamics™ NAV “Business Intelligence- Driving better business performance for companies with changing needs” White Paper, January 2007 download.microsoft.com/.../nav_business_intelligence_whitepaper.docx
- [26] The 11th Annual Gartner Business Intelligence Summit 2010, http://www.gartner.com/it/content/927900/927913/bi_summit_brochure_2010.pdf
- [27] Sunil Mistri ”Open Source BI Solutions: a Low TCO Prospect”, Information Management Special Reports, June 9, 2009, http://www.information-management.com/specialreports/2009_145/open_source_business_intelligence_bi_management_decision_making-10015500-1.html
- [28] Web Analytics World by Manoj Jasra, March 06,2009 <http://www.webanalyticsworld.net/2009/03/do-you-need-open-source-business.html>

- [29] Andreas Bitterer, “Who's Who in Open-Source Business Intelligence”, Gartner Research, 16 April 2008,
http://www.stratebi.es/todobi/may08/whos_who_in_opensource_busin_156326.pdf
- [30] Davy Nys “Open Source BI Overview” presentation, Pentaho Corporation,
<http://www.bi-community.org/Bijlage/MicrosoftPowerPoint-20200803TDWIBeneluxChapterOpenSource.pdf>
- [31] Gartner Research, “Management Update: Total Cost of Ownership Analysis Provides Many Benefits”, 27 August 2003, ID Number: ICG08272003-01.
- [32] List of Open Source ERP Softwares Available. SITE – For Open Source ERP.
<http://open-source-erp-site.com/list-of-open-source-erps.html>
- [33] Dolibarr ERP/CRM, Sourceforge,
http://sourceforge.net/search/?type_of_search=soft&words=dolibarr&search=Search
- [34] Dolibarr ERP/CRM, http://wiki.dolibarr.org/index.php/What_Dolibarr_Do
- [35] Open Source Business Intelligence. Squidoo. <http://www.squidoo.com/osbi>
- [36] Pentaho Community, <http://community.pentaho.com>
- [37] Pentaho Corporation ”Introduction to the BI Platform
”<http://wiki.pentaho.com/display/ServerDoc2x/Introduction+to+the+BI+Platform>
- [38] Hall, O. P., Using Dashboard-Based Business Intelligence Systems - An approach to improving business performance, Graziadio Bussiness Report – A Journal of Relevant Information and Analysis, 2003, Volume 6, Issue 4,
<http://gbr.pepperdine.edu/034/bis.html>
- [39] Pentaho, Creating Action Sequences, Pentaho Corporation,
http://wiki.pentaho.com/download/attachments/11862386/creating_action_sequences.pdf
- [40] GNU General Public Licence, <http://www.gnu.org/licenses/gpl.html>.
- [41] SpagoBI, <http://www.spagoworld.org>
- [42] JasperSoft, <http://www.jaspersoft.com>

Appendix A

GNU GENERAL PUBLIC LICENSE [40]

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The GNU General Public License is a free, copy left license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

“This License” refers to version 3 of the GNU General Public License.

“Copyright” also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

“The Program” refers to any copyrightable work licensed under this License. Each licensee is addressed as “you”. “Licensees” and “recipients” may be individuals or organizations.

To “modify” a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a “modified version” of the earlier work or a work “based on” the earlier work.

A “covered work” means either the unmodified Program or a work based on the Program.

To “propagate” a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays “Appropriate Legal Notices” to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a

major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.

- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to “keep intact all notices”.
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an “aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
- d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
- e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically

terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that

transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's “contributor version”.

A contributor's “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, “control” includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a “patent license” is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To “grant” such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of

this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. “Knowingly relying” means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your

obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <year> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author>

This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an “about box”.

You should also get your employer (if you work as a programmer) or school, if any, to sign a “copyright disclaimer” for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.

Appendix C

List of Top Open Source Business Intelligence Solutions



[Pentaho BI Suite](#) is one of the world's most popular open source BI suite which has won many awards.

Key capabilities: Reporting, analysis, dashboards, data integration, data mining

Runs on: Java runtime version 1.5 or higher and tested on Sun Hotspot Client VM



[JasperSoft BI Suite](#) is an easy to deploy and cost-effective enterprise strength BI suite which specializes in reporting tools suitable for organization departments, SMBs and ISVs.

Key capabilities: Reporting, dashboards, analysis, data integration

Runs on: 100% pure Java. JDK 1.5 and above. Supports any RDBMS with JDBC 2.0 driver.



[OpenI](#) is an open source BI application for on demand web-based SAAS deployment. Its web-based application allows you to connect to OLAP data sources, relational databases, data mining models for the purpose of publishing web-based reports.

Key capabilities: Reporting, analysis, dashboards

Runs on : J2EE web application environment with connectors to OLAP servers, RDBMS servers and Data Mining servers



[SpagoBI](#) is a unified BI solution built on top of a free and open source platform specifically for enterprises.

Key capabilities: Reporting, multidimensional analysis, dashboards, KPI models, data mining

Runs on: Java based web application, supported on Tomcat, JBoss and WebSphere. Also runs on portal applications such as eXo WebOS and Liferay.



[Palo BI Suite](#) was originally seen as essentially an add-on for Microsoft Excel but was eventually developed to a BI Suite called Palo Suite.

Key capabilities: OLAP server for real-time aggregation, Worksheet server for reporting, planning and analysis, and ETL server for transforming and loading data.

Runs on: Desktop application for Windows or Linux environment. Even has deep integration for Excel users.

BIRT Project

[Eclipse BIRT](#) is an open source Eclipse-based reporting system that integrates with your Java/J2EE application to produce compelling reports. Also well-known as a reporting plug-in for Java and J2EE.

Key capabilities: Reporting features such as data access, report layout and scripting

Runs on: Eclipse platform, GEF, EMF, DTP and WTP packages



[RapidMiner](#) is a comprehensive data mining and analysis tool which is excellent for forecasts and predictive analysis.

Key capabilities: Distributed data mining, knowledge discovery, multi layer data view, machine learning library

Runs on: 100% pure Java