Evaluating Business Intelligence Platforms: a case study

Carlo DELL'AQUILA, Francesco DI TRIA, Ezio LEFONS, and Filippo TANGORRA
Dipartimento di Informatica
Università di Bari
via Orabona 4, 70125 Bari
ITALY

Abstract: The paper examines some common platforms supporting Business Intelligence activities in order to state evaluation criteria for the system choice. The evaluation considers a software measurement method based on the analysis of the functional complexity of the platforms. The study has been performed on an academic warehouse that uses historical data available in legacy databases. Experimental results are reported which show the advantages and the drawbacks of each considered system.

Key-Words: - Data warehouse, Data mart, OLAP system, Functional size measurement.

1. Introduction

Modern Business Intelligence (BI) technologies are tightly integrated, easily, and widely deployed and usable for they are based on prepackaged application solutions [1]. For these reasons, Business Intelligence has become so much easy to justify relevant investments and the cost for developing and maintaining a data warehouse has significantly decreased. Traditional users of data warehouses are banks, financial services, or chains of supermarkets; instead, Institutional Organizations (e.g. Academies) in the past were not interested to collect and store large amount of data to use for strategic decision making. Now the trend is reversed: nowadays, we can consider the management of a University as critical as the management of a big business company, because the factors affecting an optimal management of a University are the same involved in the business processes [2]. It is clear that the development of an academic data warehouse can provide a lot of benefits, as these databases represent the source of knowledge for the researchers and for the Academic Decision Makers. However, without an effective Business Intelligence System that allows users to extract vital information, the data often go underutilized; in this case, if there is a very large collection of data to manage and there is an effective and competitive IT-competence, a Business Intelligence solution can help academic staff to ask questions that are impractical in a traditional way [3]. In order to implement a Business Intelligence solution in different business contexts and to maximize the benefits that end users can obtain, its technologies must be organized. The techonology must be deployed within an infrastructure with the capabilities to implement the Business Intelligence process that has been described in this paper and to support the range of applications best suited to every user of every type [4]; this infrastructure is called Business Intelligence Platform. These tools are software designed to support access to all forms of business information, not only the data stored in the data warehouse [1]. In fact, an effective business intelligence tool must be able to access quality information from a variety of sources stored in different forms, even in unstructured forms; in these cases, vertical collection-building and metasearching methods are necessary [5].

It is evident that BI platforms are very different among themselves as concerns performances and features; in spite of everything that, only in the April 2007, Gartner [6] has formalized standard criteria for executing an evaluation and a comparison of the technical and functional characteristics that must be owned by these software tools. The platforms are the client side components of the Business Intelligence Architecture; the server side is called OLAP Server and it is always a subsystem embedded in modern DBMSs, that usually are able to integrate MOLAP and ROLAP technologies [7, 8]. It is possible to find a complete checklist for evaluating OLAP Servers in [9, 10], since this topic is outside the scope of this paper. In particular, the aims of this paper are: (a) performing an effective comparison of the platforms to get force and weakness points of these software and (b) developing a Business Intelligence Application that uses real data stored in an Academic data warehouse.

2. Evaluation criteria

As stated in Section 1, the evaluation criteria adopted in this paper are the ones introduced by Gartner in [6], which defines twelve capabilities, grouped in three main areas. For each capability, we have established a series of tasks that can be object of evaluation.

I. Information delivery

a. *Reporting*. This capability comprises the task of creating and formatting interactive reports, by performing on-line analytical queries on both relational and multidimensional data sources, hiding the complexity of the warehouse's logical schema. The capacities of scheduling and sharing a report between end users are comprised too.

b. *Dashboards*. This capability is logically linked with the previous one and it concerns the ability to build, to publish and to update a set of meaningful and interactive charts to a web-based application.

c. *Ad hoc queries*. This capability allows users to create their own queries. In this case, users need to know the data warehouse's logical schema and SQL programming language.

d. *Microsoft Office integration*. A lot of users are used to create their own report with Microsoft Excel. This capability comprises the tasks that a user have to do for creating a report using Excel as a OLAP client and the BI Platform as a middleware.

II. Integration

a. *BI infrastructure*. In this capability we insert all the tasks regarding the implementation of the political rules for the security administration.

b. *Metadata management*. The process of metadata creation is the first and the most important task to carry out the integration of the BI Platform with the OLAP Server.

c. *Development environment*. A BI Platform must be equipped with a set of reusable component to integrate in a BI Application.

d. *Workflow and collaboration*. In this capability they are included all the tasks that allow users to share information, to communicate each other in a public way, or to implement business rules to generate information by trigger-driven events.

III. Analysis

a. *OLAP*. This capability comprises all the tasks that allow users to execute traditional OLAP queries (as drilling) and to define their own functions.

b. *Visualization*. In some cases, users need to visualize a report containing multi-dimensional data so as to get an optimal view even in a two-dimensional screen; as an example, this effect for example can be obtained defining the graphics details of the tool.

c. *Predictive modeling and data mining*. This capability comprises the tasks that allow users to manage a predictive modeling environment.

d. *Scorecarding*. This capability regards the tasks needed for designing strategy maps that align key performance metrics with the achievement of strategic objectives.

2.1. Software evaluation metric

Software measurement is a field of the Software Engineering and it consists in a quantitative evaluation of a tool. In the experiment illustrated in this paper, we have used the Function Point metric [11], that has been the most utilized metric for the functional size measurement of a software in the last years. In fact, its main feature is to be platform-independent, not only from hardware technology but also from the programming language used for the development. Moreover, a function point analysis is carried out from the user's point of view, not the developer's one. The Function Point analysis measures the features whose an application is composed of, by listing all the real elements that are enumerable by the end user. A key-factor is that the Function Point metric provides a normalization technique that allows the comparison among systems of different vendors. In fact, this metric measures an application on the basis of two evaluation areas: the first is based on the Unadjusted Function Point value, that reflects the features provided to the user by the application; the second provides the Adjustment Factors values, that emphasize the complexity of the general features provided to the user. The final value of this metric is get by adding the first value to the second one. The first step consists of determining the type of functional counting, in reference to the state of development of the application; the second step establishes the counting context; this context is determined by the scope of the counting and it identifies the tasks that must be evaluated. In this paper, we have applied the metric at the Application Counting level, that measures an application already installed. This counting is a baseline metric and estimates the features actually provided to the user. At last, the context of the counting is the one stated in previous section 2.

3. Academic Business Intelligence System

The Academic Business Intelligence System faced by this paper is represented in Figure 1 and it has the following principal components:

A. **Didactics Data Mart**. The Didactics Data Mart is part of the Academic Data Warehouse, developed with Oracle Database 9i and explained in [12]. The Didactics Data Mart's logical model can be thought as a cube, whose axes represent analysis dimensions; the principal dimensions are: time, student, and course of study; these are the base dimensions, because they represent the minimum of information to express *who*, *where* and *when* aggregation levels for business analysis; according to these three coordinates, it is possible to find data; such data is stored in a cell of the cube that represents the value of a measure; a measure is the quantitative description of a fact; in a business context, a fact is a meaningful event to be analyzed. In general, the Didactics Data Mart has got 4 cubes: tax, examination, bachelor's degree, and enrolment.

B. **MicroStrategy 8**. *MicroStrategy Desktop* [13] is a Business Intelligence Platform that enables users to develop applications using a simple graphical interface. MicroStrategy Desktop is the tool that allows users to define metadata (called *schema objects*), such as attributes, facts, tables, and hierarchies. These metadata are mapped to the data warehouse structures and they are stored by MicroStrategy in a relational database in a proprietary format; the schema objects are used to convert user requests into SQL queries. With this tool, users can develop application objects too; these objects, as metrics, prompts and filters, are the building blocks for creating reports and documents and they are shared among applications.

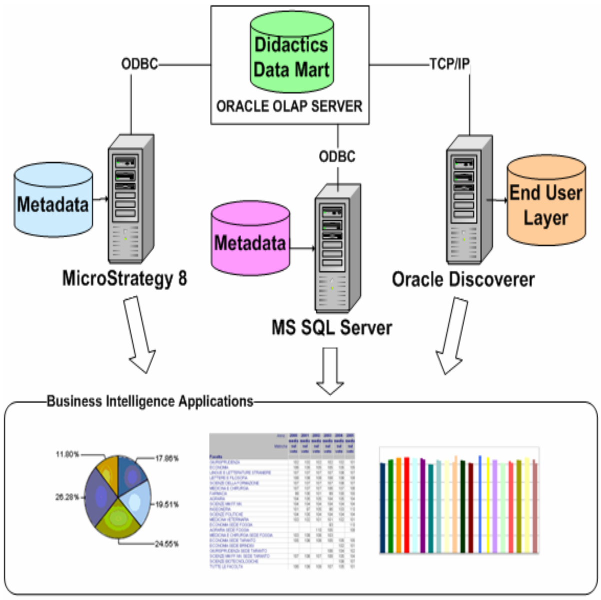

Fig. 1. Academic BI Architecture.

C. **Oracle Discoverer**. Oracle Business Intelligence Discoverer [14] is a Business Intelligence Platform that gives business users the ability to access information stored in a data warehouse, providing a business view that hides the complexity of the underlying data structures. *Oracle Discoverer* is composed of two components: *Desktop* and *Administrator*. *Discoverer Desktop* is a Windows-only application that enables end users to build new worksheets to analyze relational data. *Discoverer Administration* is a Windows-only application used by the Discoverer manager to create and maintain a business oriented view of relational data. The *Discoverer End User Layer* component is a repository for storing and retrieving definitions of objects used when querying relational data sources.

D. **Microsoft SQL Server 2005**. This is a software that comprises a set of powerful Business Intelligence tools [15]; *Analysis Services* is the tool that provides a unified and integrated view of all business data and provides algorithms for data mining with which it is possible to identify rules and patterns in business data; *Integration Services* is a component that can integrate data coming from any source; finally, *Reporting Services* is a server-based reporting tool, designed to help end users to manage interactive Web-based reports.

3.1. Business Intelligence Application

We have developed a Business Intelligence Application to evaluate the Academic Information System. The evaluation of the university information system is fundamental for the total university improvement. In fact, it stimulates the introduction in every Athenaeum of excellent quality systems, as the universities have to behave like business companies. The major difference is that in the business environment the omnipresent hard metric of prices or amount is used. Such hard metric is not applicable to educational environment for most of the activities. It is fundamental to develop an application that enables universities to measure the success or failure of teaching activities [16]. The aim of a business intelligence application developed on the Didactics Business Area is to allow analysis of the student status relative to the single registration centres, for discovering eventual problems affecting a particular course of study or a teaching matter; such a Business Intelligence application has got some metrics that allow the analysis of the academic career and the Didactics.

Typical Academic business applications are:

- Monitoring the incoming and the outgoing flows of the student in the University;
- Monitoring the didactic workload of the teaching staff;
- Monitoring the payment of the taxes.

The application we have chosen to create is the one relative to the payment of the student's taxes. In particular, the report calculates the sum of the taxes paid by the students grouped by degree course.

The same Business Intelligence Application has been developed with the three tools showed in the Figure 1. The development of such an application allowed us to obtain an evaluation of the Business Intelligence Platforms. To obtain a quantitative and not experience-based evaluation, we have used the metric illustrated in section 2.1 to the tasks listed in section 2. The experimental results are described below.

4. Experimental data

Here, we report the comparison carried out on the following Business Intelligence Platforms: Oracle Discoverer, MicroStrategy 8, and MS SQL Server.

The first step to obtain the complexity of a task is to list all the components of that task. The components can be of two types: Data Functions and Transactional Functions. Data Functions are: (a) Internal Logical File (ILF) and (b) External Interface File (EIF). The ILF is a logical, persistent entity about which data will be maintained. ILFs are based on logical user requirements and they are independent of the physical implementation or storage (tables or databases). The EIF is a logical, persistent entity which is required for reference or validation by the task being counted, but which is maintained by another task.

The complexity of ILFs/EIFs is evaluated by counting the non-recursive user data fields (Data Element Type - DET) and the logical record element types (Record Element Type - RET) it contains, according to the following Tab. 1.

ILF/EIF	DET		
RET	1-19	20- 50	>50
<2	Low	Low	Medium
2-5	Low	Medium	High
>5	Medium	High	High

Table 1. ILF/EIF complexity.

Transactional Functions are: External Input (EI), External Output (EO), and External Inquiry (EQ). The EI is a logical business process that maintains the data within one or more ILFs. The EO is a logical business process that generates data to a user or other application outside of the software. The EQ consists of a trigger and response (or question and answer) pair whereby the question or request for data comes into the application from outside and data are retrieved to answer the request and sent out. The complexity of EIs/EOs is evaluated by counting the user data fields involved (Data Element Type) and the sum of the ILFs and EIFs involved in the process (File Type Referenced). The complexity is given by the next Tab. 2 and Tab. 3.

EI	DET		
FTR	1- 4	5-15	>15
<2	Low	Low	Medium
2	Low	Medium	High
>2	Medium	High	High

Table 2. EI complexity.

The complexity of EQs is the maximum between the complexity of the EI and the EO components, by using the their own tables.

EO	DET		
FTR	**1- 5**	**6-19**	**>19**
<2	Low	Low	Medium
2- 3	Low	Medium	High
>3	Medium	High	High

Table 3. EO complexity.

The function point value of each task component is assigned according to the own function type and the evaluated complexity. Tab. 4 shows the values suggested in IFPUG [11].

Function type	Complexity		
	Low	**Medium**	**High**
ILF	7	10	15
EIF	5	7	10
EI	3	4	6
EO	4	5	7
EQ	3	4	6

Table 4. Function point value for each function type.

Once calculated the values for each task component, the sum of these values provides the Unadjusted Function Point value (UFP):

$$UFP = \sum d_i$$

where d_i is the value of complexity assigned to each task component ($1 \leq i \leq n$, and n is the number of the identified task components).

The final step in function point counting involves adjusting the function point count by a Value Adjustment Factor (VAF), which assesses additional business constraints of the software that are not addressed by the five function types.

The VAF is determined by evaluating the following 14 parameters: (a) data communications, (b) distributed data processing, (c) performance, (d) heavily used configuration, (e) transaction rate, (f) on-line data entry, (g) end-user efficiency, (h) on-line update, (i) complex processing, (j) reusability, (k) installation ease, (l) operational ease, (m) multiple sites, (n) facilitate change. Each parameter must be evaluated according to its degree of influence, whose range is based on a scale from 0 to 5, from "no influence" to "strong influence". The formula to evaluate the VAF is the following:

$$VAF = (\sum c_i \times 0.01) + 0.65$$

where c_i ($1 \leq i \leq 14$) is the degree of influence assigned to each parameter.

The final Function Point value is given by the following expression:

$$FP = UFP \times VAF .$$

Now, we show an example of our measurement on the drilling task of MicroStrategy.

Task component	Function type		DET	FTR	Complex. value
drill down / roll up	EI		7	2	4
report	EO		5	1	4
item selection	EQ	Input	1	1	4
		Output	1	1	
			DET	**RET**	
attributes list	EIF		3	1	5
parameters	ILF		5	1	7
UFP					24

Table 5. UFP for the drilling task.

Tab. 5 shows the value assigned to the five identified components. The total UFP means that Microstrategy has obtained the score 24 as complexity value on the drilling task.

Once counted the UFP, we must evaluate the VAF, assigning a degree of influence to the fourteen parameters that represent the general system characteristics (Tab. 6).

Adjustment parameter		Degree of influence	Degree of influence × 0.01
a	data communications	5	0.05
b	distributed data processing	4	0.04
c	performance	5	0.05
d	heavily used configuration	0	0
e	transaction rate	1	0.01
f	on-line data entry	5	0.05
g	end-user efficiency	5	0.05
h	on-line update	0	0
i	complex processing	5	0.05
j	reusability	5	0.05
k	installation ease	0	0
l	operational ease	1	0.01
m	multiple sites	5	0.05
n	facilitate change	1	0.01
Total			0.42

Table 6. Weights to evaluate VAF for the drilling task.

The total 0.42 is used in the final expression to count the VAF:

$$VAF = 0.42 + 0.65 = 1.07 .$$

Finally, the counted UFP and VAF are used as data to count the FP:

$$FP = 24 \times 1.07 = 25.68 .$$

The value 25.68 reached by MicroStrategy in drilling task represents the functional complexity value of the task, considering the adjustment factor introduced by VAF.

To obtain a significant evaluation of the three software, we must count the FP for at least one task for each capability belonging to the three areas reported in the evaluation criteria of Section 2.

The next Tab. 7 shows the score obtained by each Business Intelligence Platforms, according to the tasks executed during the development of the Business Intelligence Application.

Benchmark			Score		
Areas	**Capabilities**	**Tasks**	**Oracle Disc**	**MS SQL Server**	**Micro-Str**
Information Delivery	reporting	creating reports	91.67	68.40	136.40
	dashboards	creating charts	10.32	21.85	44.52
	ad-hoc queries	defining ad-hoc queries	4.62	36.90	20.02
	Subtotal		106.61	127.15	200.94
Integration	bi infrastructure	security and privileges	64.24	45.58	84.39
	metadata management	metadata creation	123.30	61.60	149.60
	workflow & collaboration	sharing info.	17.38	10.92	15.33
	Subtotal		204.92	118.10	249.32
Analysis	OLAP	drilling	23.69	9.81	25.68
Total			335.22	255.06	475.94

Table 7. Functional Complexity of BI Platforms.

Experimental data show that MS SQL Server, with the value 255.06, has the lowest functional complexity; its force points consist in the integration's capabilities that allow the BI Platform Administrators to manage users, metadata and information in a simple and immediate way. By contrast, the high functional complexity of MicroStrategy, with the value 475.94, is due to the management of object-oriented metadata. These metadata are used as building block to create more complex objects; reports are designed simply assembling all the necessary components. This method is in antithesis with the Oracle's approach that provides a wizard for the execution of all the tasks and then, for these reasons, forces users to repetitive steps.

5. Conclusion

In this paper we have showed an evaluation of three important Business Intelligence Platforms: MS SQL Server, MicroStrategy and Oracle Discoverer. The evaluation has been executed using a software measurement method that consists of the analysis of the functional complexity. The experimental data allow us to say what is already known in literature: an object oriented approach leads more complex tasks to be executed but it favours the reuse of the objects, ensuring consistency across business objects and minimizing the number of objects to maintain during the development of a Business Application.

Future works have the aims to extend this benchmark in breadth and depth; in breadth, it is possible to add others columns relative to different Business Intelligence Platforms; in depth, it is possible to insert new rows relative to the capabilities not considered in this paper, also including more than one task for each capability.

References:

[1] Colin J. White, *The IBM Business Intelligence Software Solution*, *version 3*, DataBase Associates International, Inc., March 1999.

[2] D. Wierschem, J. McMillen, and R. McBroom, What Academia Can Gain from Building a Data Warehouse, *EDUCAUSE Quarterly Magazine*, Vol. 26, No. 1, 2003, pp. 41-46.

[3] Debra Kelly, Business Intelligence: the smart way to track Academic Collections, *EDUCAUSE Quarterly Magazine*, Vol. 28, No. 4, 2005, pp. 48-53.

[4] Mitch Kramer, *A Comparison of Business Intelligence Strategies and Platforms. Comparing Microsoft, Oracle, IBM, and Hyperion*, Green Hill Analysis, November 2002.

[5] Byron Marshall, Daniel McDonald, Hsinchun Chen, and Wingyan Chung, EBizPort: Collecting and Analyzing Business

Intelligence Information, *Journal of the American Society for Information Science and Technology*, Vol. 55, No. 10, August 2004, pp. 873-891.

[6] Kurt Schlegel, Bhavish Sood, *Business Intelligence Platform Capability Matrix*, Gartner, 2007, www.gartner.com.

[7] Steve B. Elkins, OLAP's Place in the Warehouse Architecture, *DBMS*, Vol. 11, No. 4, April 1998, pp. 34-45, www.dbmsmag.com.

[8] Steve B. Elkins, Open OLAP, *DBMS*, Vol. 11, No. 4, April 1998, pp. 34-45, www.dbmsmag.com.

[9] Dan Bulos, How to evaluate OLAP servers - A Comprehensive checklist for evaluating the latest OLAP server products, *DBMS,* Vol. 8, No. 9, August 1995, pp. 96-103, www.dbmsmag.com.

[10] *OLAP Council APB-1 OLAP Benchmark Release II,* OLAP Council, November 1998, www.olapcouncil.org.

[11] *Function Point Counting Practices Manual, Release 4.1*, International Function Point Users Group, IFPUG, Westerwille, Ohio, 1999.

[12] C. dell'Aquila, F. Di Tria, E. Lefons, and F. Tangorra, An Academic Data Warehouse, *Proceedings of the 7th WSEAS International Conference on Applied Informatics And Communications* (AIC'07), Vouliagmeni, Athens, Greece, August 24-26, 2007, pp. 229-235.

[13] *MicroStrategy 8 Basic Reporting Guide*, MicroStrategy, www.microstrategy.com.

[14] *Oracle® Business Intelligence Discoverer Administration Guide 10g Release 2 (10.1.2.1) B13916-04*, Oracle, July 2005, www.oracle.com.

[15] Microsoft SQL Server, www.microsoft.com/sql/default.mspx.

[16] P. Čech, and V. Bureš, Utilisation of Business Intelligence in an Education Environment, Current Developments in Technology-Assisted Education (2006), *Fourth International Conference on Multimedia and Information and Communication Technologies in Education*, , Seville, Spain, November 22-25, 2006, pp. 210-214.