Business Intelligence Systems: a Comparative Analysis

Carlo DELL'AQUILA, Francesco DI TRIA, Ezio LEFONS, and Filippo TANGORRA
Dipartimento di Informatica
Università di Bari
via Orabona 4, I-70125 Bari
ITALY
{dellaquila, lefons, tangorra, francescoditria}@di.uniba.it

Abstract: - A set of evaluation criteria is described and considered for comparing some popular OLAP systems that support Business Intelligence. These criteria involve critical aspects such as: information delivery, system and user administration, and OLAP queries. The measurement method is based on the functional complexity analysis. Experimental results have been carried out using a data warehouse in academic environment and they allow to evidence the weaknesses and the points of force of each compared system.

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

1 Introduction

Data warehouse is the central core of a Business Intelligence (BI) system and it is usually the most expensive component to manage. The cost of the data warehouse depends on (a) the process of development and design, (b) the decisions they support, and (c) the existing technology. As concerns the last aspect, the costs are considerably reduced with time. In fact, during the last years, the market is offering several systems supporting Business Intelligence tightly integrated, easily and widely deployed and usable at acceptable costs [1]. Therefore, Business Intelligence has become so much easy to justify relevant investments and the cost for developing and maintaining a data warehouse has significantly decreased.

In order to implement a Business Intelligence solution in different business contexts and to maximize the benefits that end users can obtain, the technological components must be organized. The technology must be deployed within an infrastructure with the capabilities to implement the Business Intelligence process and to support the range of applications best suited to every user of every type [2]. This infrastructure is the so-called Business Intelligence platform.

A Business Intelligence platform is a software tool designed to support the 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 [3].

BI platforms are very different among themselves as concerns performances and features. Recently, the research has pointed out the attention on the definition of formalized standard criteria for evaluating and comparing the technical and functional characteristics that these software tools must own [4]. The platforms are the client side components of the Business Intelligence Architecture, and the server side, or the so-called OLAP Server, is always a subsystem embedded in modern DBMSs, that usually are able to integrate MOLAP and ROLAP technologies [5, 6]. We defer to [7, 8] for a complete checklist to evaluate OLAP Servers, since this topic is outside the scope of this paper.

In particular, the aim of this paper is to perform an effective comparison among the most popular platforms on the market, by means of the development of a Business Intelligence application that extracts information from a data warehouse designed and used in the Academic context.

In fact, traditional users of data warehouses have been banks, financial services, and chains of supermarkets; whereas, Institutional Organizations (*e.g.*, Academies and Universities) were not interested in the past to collect and store large amount of data to use for strategic decision making. Nowadays the trend is reversed: the management of a University can be considered as critical as the management of a big business company, because the factors affecting an optimal management of the University are the same as those involved in business processes [9]. 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 the academic decision makers. However, with no effective Business Intelligence system that allows users to extract vital information, the data are often underutilized; in this case, if there are a very large collection of data to manage and an effective and competitive IT-competence, a Business Intelligence solution can help academic staff to ask questions difficult to express in a traditional way [10].

The original and main contribution of this paper consists of a non-experience based evaluation of Business Intelligence platforms, that can serve as a landmark for the strategic choices of software tools to integrate in an Information System.

This paper has the following structure. Section 2 introduces the criteria and the measurement technique used for the evaluation, and Section 3 illustrates the Business Intelligence platforms that we have chosen for the comparison. Section 4 presents the Academic Data Warehouse used as environment in which we developed a Business Intelligence application using the software tools introduced in Section 3. Section 5 explains the functional complexity measurement process and the values assigned to each evaluated software tool. At last, in Section 6, we discuss the results of the experiment carried out.

2 Evaluation criteria

The evaluation criteria [4] define twelve capabilities, grouped in the following three main areas. For each capability, we established a set of tasks on which to perform the comparison of the software tools investigated.

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, while hiding the complexity of the warehouse's logical schema. The ability of scheduling and sharing reports among end users is also considered.
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 utilize Microsoft Excel to create their own reports. The MS Office integration 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

e. *BI infrastructure*. In this capability, we include all the tasks regarding the implementation of rules for the security administration.
f. *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.
g. *Development environment*. A BI platform must be equipped with a set of reusable components to be integrated in a BI application.
h. *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

i. *OLAP*. This capability comprises all the tasks that allow users to execute traditional OLAP queries (as drilling) and to define their own functions.
j. *Visualization*. In some cases, users need to visualize a report containing multi-dimensional data so as to get an optimal view even on a two-dimensional screen; as an example, this effect can be obtained defining the graphics details of the tool.
k. *Predictive modeling and data mining*. This capability comprises the tasks that allow users to manage a predictive modeling environment.
l. *Scorecarding*. This capability regards the tasks needed for designing strategy maps that align key performance metrics with the achievement of strategic objectives.

2.1. Metric for the Software evaluation

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 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 which an application is composed of, by listing all the real elements that are enumerable by the end user.

The key-factor is that the Function Point metric provides a normalization technique that allows to compare systems of different vendors. In fact, this metric measures the 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 value, that emphasizes the complexity of the general features provided to the user. The final value of this metric depends on both the first and the second values.

The first step consists of determining the type of functional counting, in reference to the state of the 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 apply 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, referring to the tasks presented at the beginning of this Section.

3 Business Intelligence Platforms

Business Intelligence is an activity that faces the problem to extract information from synthetic data, by using different software technologies. Information is obtained by the development of Business Intelligence applications, that consist of (a) executing analytical queries on large amount of historical data, and (b) showing the results of these queries by means of tables and charts. A Business Intelligence platform is a software that allows business users to develop Business Intelligence applications.

Once extracted, the information is used for decision making, with the aim to improve the process of an Information System. The core of the typical Business Intelligence architecture consists of a Data Warehouse, that represents the repository of historical data, built up with a process of integration of data coming from different and heterogeneous database sources.

Now, we briefly describe the three Business Intelligence platforms, that we have selected for the evaluation with the metrics introduced in Section 2.1.

The evaluated Business Intelligence platforms are the following (see, Figure 1):

a) **MicroStrategy 8**. *MicroStrategy Desktop* [12] 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 (or *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 proprietary format. The schema objects are then involved to obtain SQL queries from user requests.

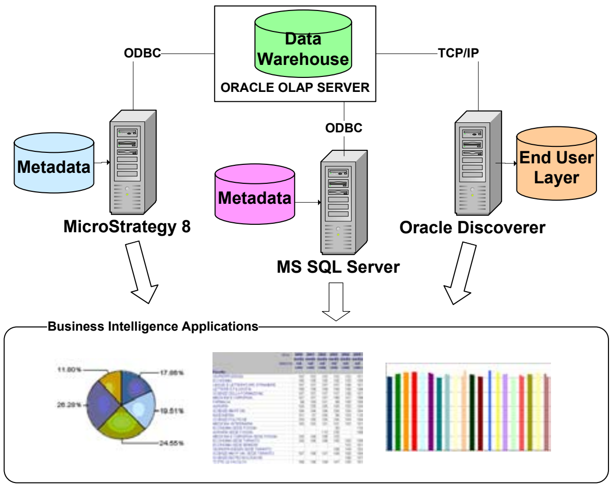

Fig. 1. Typical BI architecture.

With this tool, users can also develop application objects. These objects, such as metrics, prompts and filters, are the building blocks for creating reports and documents and they are shared among applications.

b) **Oracle Discoverer**. *Oracle Business Intelligence Discoverer* [13] 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 Windows-only components: *Desktop* and *Administrator*. *Discoverer Desktop* enables end users to build new worksheets to analyze relational data. *Discoverer Administrator* is used by the Discoverer manager to create and to 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.

c) **Microsoft SQL Server 2005**. This software comprises a set of powerful Business Intelligence tools [14]. *Analysis Services* is the tool that provides a unified and integrated view of all business data and provides algorithms for data mining (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.

4 Case Study: Didactics Data Mart

In order to have a real case study to apply the metrics, we have chosen a Data Mart of the Academic Data Warehouse as environment for developing an application with the three Business Intelligence platforms object of the present evaluation.

In particular, the Didactic Data Mart [15] contains data about the students enrolled to the Faculties of the University of Bari.

This Data Mart has been designed through the integration of the logical schemas of two transactional databases: (a) ESSE3 (Secretary and Services for Students), that is the current database that supports all the didactic curricula, and the administrative processes and services to the students in accordance with the didactic autonomy of the University; and (b) NOGE (NOt ManaGEd), that is a secondary database that stores residual historical data about students enrolled before the ESSE3 introduction.

NOGE and ESSE3 represent also the repository of data used to feed the Data Mart, after the *Extraction Transformation Loading* (ETL) process. There are significant differences between the two databases, making very difficult the process of data integration. These differences regard not only the cardinality of the tables but even the representation of the information.

ESSE3 is a database where the cardinality of tables is of the order of millions of tuples. In NOGE, there are tables with thousands of records, but nevertheless the database contains very dirty data.

The Didactics Data Mart's logical model can be thought of as a set of data cubes, whose main dimensions are: time, student, and course of study; these are the base dimensions, because they represent the minimum information to express *who*, *where* and *when* aggregation levels for business analysis. According to these coordinates, it is possible to find data; a single datum is stored in a cell of the cube and it represents the value of a measure; a measure is the quantitative description of a fact; and, 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, university degree, and enrolment.

The *tax cube* is represented by a fact table whose function is to control the payment of the taxes by the student; it has only the student, time and degree course dimensions and the amount field as measure. Figure 2 shows the conceptual schema representing the *tax cube*, according to the data warehouse Dimensional Fact Model [16, 17]. The *examination cube* is represented by a fact table that allows analyses of the university career of the student; this fact table has four dimension tables; the additional dimension is represented by the teaching course, that allows didactic aggregation. In this fact table there are two fields: the mark field, and the *cum laude* boolean field. The *university degree cube* consists of a fact table useful to obtain information about the graduated students; it has the same four dimension tables, measure, and fields, as the examination fact table. The *enrolment cube* is the fact table that stores information about the enrolment of the student to a course of study; it has five dimension tables and no measure; the additional dimensions are the residence, that allows demographic and geographic aggregations, and the kind of enrolment, that allows administrative aggregations.

Moreover, the Didactics data mart contains further fact tables to allow analyses and statistics on the didactic offer of the University.

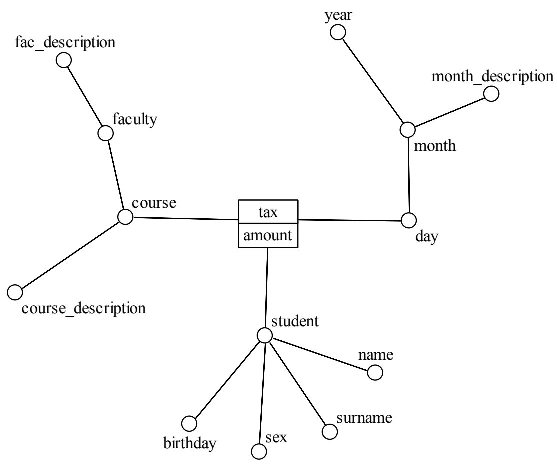

Fig. 2. The Tax Cube.

The *distribution cube* is represented by a fact table that indicates the teaching courses for each degree course of study. The information includes the number of teaching hours, the number of university formative credit (UFC), the kind of lesson, the kind of examination, and also the teacher of each teaching course.

The *cost cube* is represented by a fact table relative to the annual costs supported by the University for the management of each teaching course, and totally for each degree course per academic year. It contains also information on the cost of teachers not enrolled in the University teacher's staff.

In order to obtain aggregate results at different levels of granularity, the degree course is organized in a two-level dimensional hierarchy: course and faculty, for allowing to aggregate measures (for example, the count of graduate students) at the degree study level or the faculty level. The residence is a four-level dimensional hierarchy for aggregating measures at the city, province, region, and nation levels. This hierarchy allows users to analyze data referring to different geographic contexts. Finally, the time is a three-level dimensional hierarchy, including the day, month, and year levels. All other dimensions of the Didactics Data Mart are one-level hierarchies.

The complete logical schema of the Didactics Data Mart is shown in Figure 3. For example, residence hierarchy is composed of a set of four tables, where city represents a child dimension and province its own father dimension. In this case, the city table has a foreign key constraint, referencing the primary key field of the province table.

The enrolment fact table was traduced to a table whose primary key is composed of the set of the foreign keys, referencing the primary keys of kind, city, day, student, and degree course tables.

4.1. Business Intelligence Application

We developed a Business Intelligence application to evaluate the Academic Information System. The evaluation of the university Information System is a difficult task with respect to a business one. The major difference is that in the business environment hard metrics, such as price or amount, are used. Such hard metrics are not applicable to the educational environment for the most activities. It is fundamental to develop an application that enables universities to measure the success or

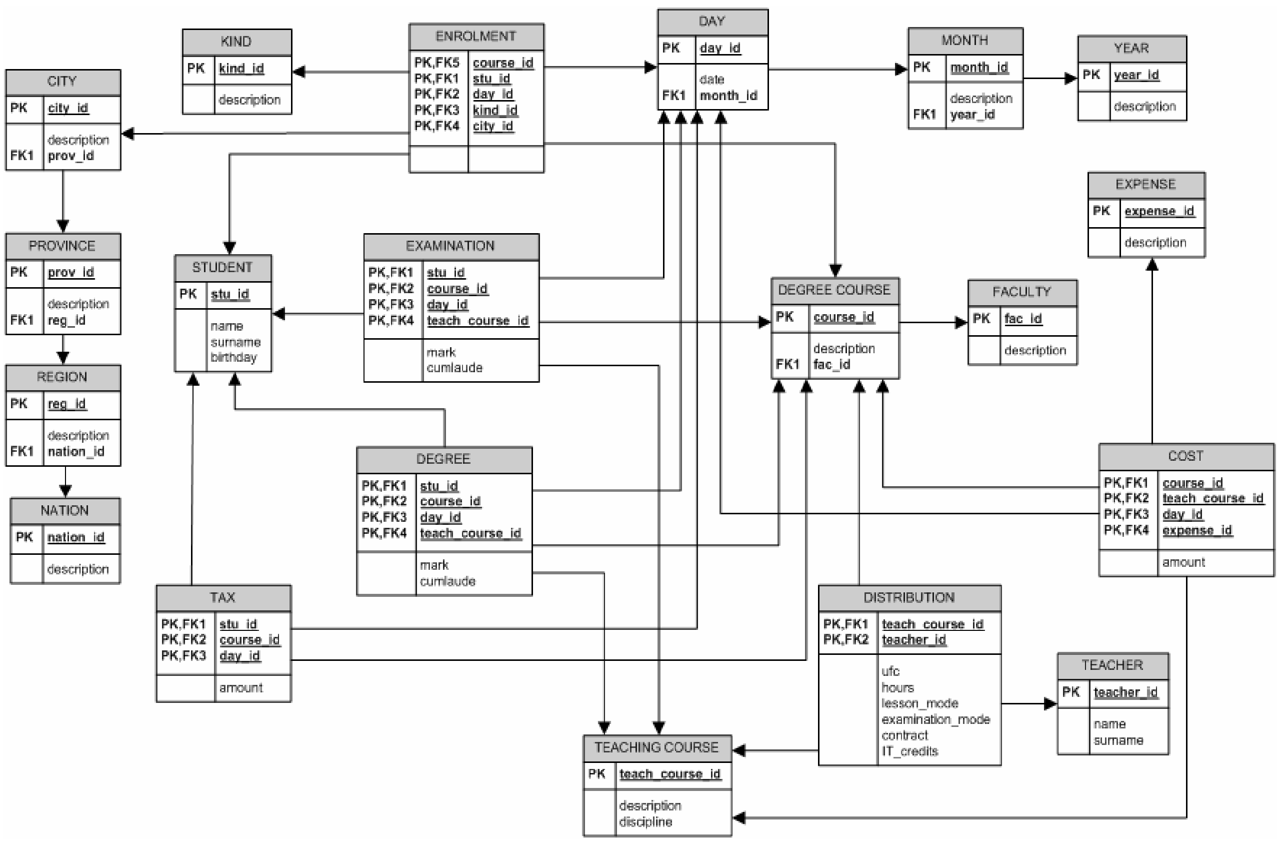

Fig. 3. The Didactics data mart.

failure of teaching activities [18]. The aim of a Business Intelligence application developed on the Didactics Business Area is to allow analyses of the student status relative to the single registration centres, for discovering 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 student's careers 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; and
- monitoring the payment of the taxes.

The application we chose to create is the one relative to the enrolments of the students to the courses of study. In particular, the report calculates the percentage of students enrolled at the University of Bari from 2000 to 2005, grouped by Faculty and academic year. The report is shown in Table 1.

FACULTY	2000	2001	2002	2003	2004	2005
Edu. Sci.	12.88	14.23	14.91	16.85	20.26	18.22
Law	21.99	18.98	16.84	15.57	13.36	13.74
Med. and Surgery	8.17	7.87	8.46	9.66	11.57	12.30
Economics	14.35	14.20	13.64	12.63	11.62	12.04
Math., Phys. and Nat. Sci.	9.83	12.36	13.09	12.16	10.54	10.08
Arts and Philosophy	7.81	6.95	7.56	7.22	6.99	7.49
Pharmacy	5.47	6.15	4.73	5.05	6.18	6.75
Foreign Lang. and Lit.	4.79	5.12	5.74	5.96	5.96	5.84
Political Sciences	6.84	6.36	6.13	5.83	4.79	5.20
Law (Taranto city)	2.62	3.11	3.74	3.91	3.53	3.30
Veterinary Medicine	2.07	1.93	2.12	2.38	2.85	2.74
Agricultural Sciences	1.87	1.56	2.04	1.84	1.45	1.25
Economics (Taranto city)	1.30	1.16	0.99	0.94	0.91	1.06

Table 1. Percentage of university students, grouped by Faculty and Year (2000 – 2005).

This report shows the most populated Faculties of the University of Bari and the branch university town (Taranto city). In particular, it is possible to extract the information that in 2005, the last considered year, the Educational Science Faculty had the highest flow of students (over 18 per cent), while the Economics Faculty, localized in Taranto city, had an irrelevant number of students (about 1 per cent).

Traditionally, a large number of students always chose the Law Faculty, that was, for social and historical reasons, the most populated Faculty of the University. Since 2004, this Faculty has definitively loosed almost the 43% of its students, that are migrated to Educational Science and Medicine and Surgery Faculties. The rest of the Faculties does not exhibit significant difference along the five years.

The same Business Intelligence application was developed with the three tools shown in Figure 1. The development of this application allowed us to evaluate the Business Intelligence platforms. In order to obtain a quantitative, not empirical evaluation, we applied the metric discussed in Section 2.1 to the tasks listed in Section 2. The experimental results are described in Section 5.

5 Experimental data

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

The first step to measure the complexity of a task is to list all the components of that task. The components are of two types: Data Functions and Transactional Functions. Data Functions are: Internal Logical File (ILF) and External Interface File (EIF). The ILF is a logical entity that determines what data will be managed by the task being counted. ILFs are based on user requirements and they are independent of the physical implementation or storage (tables, databases). The EIF is a logical entity that is used by the task being counted, but it is managed 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 Table 2.

Transactional Functions are: External Input (EI), External Output (EO), and External Inquiry (EQ).

The EI is a logical process that stores external data into one or more ILFs. The EO is a logical process that generates data to a user or external

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

Table 2. ILF/EIF complexity.

applications. The EQ consists of a question-answer pair whereby the question is triggered by an external event, as a user input, and the data that satisfy the request are retrieved and sent to the user.

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 - FTR). The complexity is given by the next Tables 3 and 4.

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

Table 3. EI complexity.

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

Table 4. EO complexity.

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

The function point value of each task component is assigned according to the own function type and the evaluated complexity. Table 5 reports the suggested values in IFPUG [11].

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

$$UFP = \sum_i d_i$$

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

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 5. Function point value for each function type.

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, and (n) facilitate change. Each parameter must be evaluated according to its degree of influence, whose range is based on a scale from 0 ("no influence") to 5 ("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 the corresponding 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.

Table 6 shows the value assigned to the five identified components. The total UFP means that Microstrategy has obtained the score 24 as complexity value of the drilling task.

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	3	1	
			DET	**RET**	
attributes list	EIF		3	1	5
parameters	ILF		5	1	7
UFP					24

Table 6. UFP for the drilling task.

In detail, to execute the drill down task, we started from the report shown in Table 1, and then we created a new report, where the grouping fields were *Degree Course* and *Year*.

For the "drill down/up" task component, we counted the value 7 for the DET. This value is given by the following parameters: (1) choice of the function, (2) item selection, (3) drilling direction selection (up or down), (4) father selection, (5) thresholds selection, (6) filter setting, and (7) confirm of the parameters. The value 2 for FTR is given because this task component depends on two files: database file, and metadata file.

For the "report" task component, that is an output component, we counted the value 5 for the DET. This valued was obtained because the fields involved in the report are: (1) Degree Course, (2) Year, (3) the percentage of enrolled students, (4) the filter on the year, and (5) the ordering on the values of the percentage. This task component generates only one file, that is the value for FTR.

"Item selection" task component is composed of two parts, because it is a question-answer component. It consists of selecting a starting item and then choosing an item (among those connected by a father-child relation to the starting item) on which executing the drilling. So, for the input part, we obtained the value 1 for DET, because there is one logical field to fill, and the value 1 for FTR, because it depends only on the metadata file. For the output part, we obtained the value 3 for DET, because this process returns three logical fields: items on which executing drill down, those for drill up, and those for drill across. The value for FTR is 1 because it depends only on the metadata file.

For the "attributes list" task component, whose data are generated by an external component, we counted the value 3 for DET and the value 1 for RET, because the attributes are separated into three logical fields, belonging to the same logical record.

As for the "parameters" task component, we counted the value 5 for DET. These parameters are the input data managed by the "drill down/up" task component, and in particular they are the following: (1) item selected, (2) drilling direction selected, (3) father selected, (4) thresholds selected, and (5) filter set. These parameters are grouped into the same logical record.

Once the UFP is counted, we must evaluate the VAF, by assigning a degree of influence to the fourteen parameters that represent the general system characteristics (see, Table 7). The total 0.42 is used in the final expression to evaluate the VAF:

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

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	0.00
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	0.00
i	complex processing	5	0.05
j	Reusability	5	0.05
k	installation ease	0	0.00
l	operational ease	1	0.01
m	multiple sites	5	0.05
n	facilitate change	1	0.01
Total			0.42

Table 7. Weights to evaluate VAF for the drilling task.

Finally, the evaluated UFP and VAF are used as data to evaluate 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.

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

Table 8 shows the score obtained by each Business Intelligence platforms, according to the tasks executed during the development of the Business Intelligence application.

Experimental data show that MS SQL Server, by obtaining 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, that reaches 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

Benchmark			Score		
Area	**Capability**	**Task**	**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 8. Functional Complexity of BI platforms.

assembling all the necessary components. This method is in antithesis with Oracle's approach that provides a wizard for the execution of all tasks and, therefore, forces users to repetitive steps.

6 Conclusion

In this paper, we have shown the evaluation of three popular Business Intelligence platforms: MS SQL Server, MicroStrategy and Oracle Discoverer. The evaluation has been carried out using a software measurement method consisting of the analysis of the functional complexity. The experimental results allow us to confirm what is 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 be maintained during the development of a Business Intelligence application.

Future work is 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] C. J. White, *The IBM Business Intelligence Software Solution, version 3*, DataBase Associates International, Inc., March 1999.

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

[3] B. Marshall, D. McDonald, H. Chen, and W. 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.

[4] K. Schlegel, and B. Sood, *Business Intelligence Platform Capability Matrix*, Gartner, 2007, www.gartner.com.

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

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

[7] D. 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.

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

[9] 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.

[10] D. Kelly, Business Intelligence: the smart way to track academic collections, *EDUCAUSE Quarterly Magazine*, Vol. 28, No. 4, 2005, pp. 48-53.

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

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

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

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

[15] 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.

[16] M. Golfarelli, D. Maio, and S. Rizzi, The dimensional fact model: a conceptual model for data warehouses, *International Journal of Cooperative Information Systems*, Vol. 7, No. 2-3, 1998, pp. 215-247.

[17] M. Golfarelli, and S. Rizzi, A Methodological Framework for Data Warehouse Design, *Proceedings ACM First International Workshop on Data Warehousing and OLAP (DOLAP 98),* Washington, D.C., USA*,* November 7, 1998, pp. 3-9.

[18] 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.