Binding Data Mining to Final Business Users of Business Intelligence Systems

Ana Azevedo
CEISE/STI
ISCAP/IPP
Porto, Portugal
aazevedo@iscap.ipp.pt

Manuel Filipe Santos
Centro Algoritmi
University of Minho
Guimarães, Portugal
mfs@dsi.uminho.pt

Abstract— **Since Lunh first used the term Business Intelligence (BI) in 1958, major transformations happened in the field of information systems and technologies, especially in the area of decision support systems. Nowadays, BI systems are widely used in organizations and their strategic importance is clearly recognized. The dissemination of data mining (DM) tools is increasing in the BI field, as well as the acknowledgement of the relevance of its usage in enterprise BI systems. One of the problems noted in the use of DM in the field of BI is related to the fact that DM models are, generally, too complex in order to be directly manipulated by business users; as opposite to other BI tools. The main contribution of this paper is a new DM language for BI conceived and implemented in the context of an Inductive Data Warehouse. The novelty is that this language is, by nature, user-friendly, iterative and interactive; it presents the same characteristics as the usual BI tools allowing business users to directly manipulate DM models and, allowing through this, the access to the potential value of these models with all the advantages that may arise.**

Keywords – Data mining; DM language; Business Intelligence; BI system; Inductive database; Inductive data warehouse; Business user.

I. Introduction

Organizations compete in environments whose complexity increases in a daily basis. Consequently there are many demands that organizations must answer in time and adequately in order to survive and gain competitive advantage in those complex environments. In this context, computerized Decision Support Systems (DSS), in particular Business Intelligence (BI) systems, play an important role in order to improve decision making and thus conducting organizations' actions. BI systems are gaining momentum each day in organizations and have a fundamental role in these issues [1][2].

The usage of Data Mining (DM) tools in BI is increasing. BI and DM, despite having roughly the same age, have different roots and as a consequence have significantly different characteristics [3][4]. DM came up from scientific environments, thus it is not business oriented. DM tools still demand heavy work in order to obtain the intended results, hence needing the knowledge of DM specialists to explore its full potential value. On the contrary, BI is rooted in industry and business, thus it is business oriented. As a result, BI tools are user-friendly and can easily be accessed and manipulated by business users.

From the literature review, it is evident that the majority of BI tools are directly manipulated by business users, allowing them to explore their potential value in a more effective way. The reason for this is related with the fact that BI tools are user-friendly, iterative, interactive, business oriented, and oriented to business activities. DM is an exception [5][6]. Despite its usage in BI systems is increasing day by day, DM models are not directly manipulated by business users who depend on reports from DM specialists. This way, business users could be unable to extract the potential business value contained in DM models. The complexity of DM models, as opposite to other BI tools, has been identified as the key factor for this.

The importance of allowing final business users to access and manipulate DM models comes up from the need of allowing business users to be more autonomous, without the permanent necessity to depend on the presence of a DM specialist. Moreover, considering that DM specialists do not usually have a complete knowledge of the business issues, making DM directly available to business users is the key element that allows obtaining all the potential business value that could be hidden in DM models. Hereby, the authors state that this can be done by means of a DM language developed, above all, to accomplish the necessities of final business users of BI systems. Consequently, it is considered in the research hereby presented, the importance of developing DM languages for BI, which are oriented to business users and, moreover, to BI activities.

Realizing the importance of the aspects mentioned above, the recognition of this reality establishes the foundations for this research. Accordingly, and based in the literature review, the research problem has been identified as: Final business users do not directly access and manipulate DM models and consequently their full potential business value could be not completely explored. The presented problem arises from the business needs existing in environments where BI systems include DM usage. Binding DM to final business users of BI systems thus inducting them into data mining models is considered a pertinent contribution.

From the literature review, it is given evidence of the necessity to develop tools for DM that present the same characteristics of BI tools, namely being user-friendly, interactive, iterative, oriented to business users, and oriented to BI activities, and thus could be directly manipulated by business users. This is also aligned with the roots of DM and Knowledge Discovery in Databases (KDD) as stated in [7] where KDD is presented as an iterative and interactive process, with many decisions being made by the user.

The main contribution of this paper is a new DM language conceived and implemented in the context of an Inductive Data Warehouse (IDW). This new DM language will bind DM to final business users of BI systems, thus allowing them of being able to extract the potential business value hidden in DM models.

This paper presents the developed research and is organized as follows. It starts by presenting background and related work, in Section II. It follows with the research methodology and obtained results, in Section III. Next, in Section IV, limitations and future research design are brought in. The paper concludes in Section V.

II. BACKGROUND AND RELATED WORK

The term knowledge discovery in databases or KDD, for short, was coined in 1989 to refer to the broad process of finding knowledge in data, and to emphasize the "high-level" application of particular DM methods [8]. Fayyad considers DM as one of the phases of the KDD process. The DM phase concerns, mainly, the means by which the patterns are extracted and enumerated from data. As of the foundations of KDD and DM, several applications were developed in many diversified fields. The growth of the attention paid to the area emerged from the rising of big databases in an increasing and differentiated number of organizations. Nevertheless, there is the risk of wasting all the value and wealthy of information contained in these databases, unless the adequate techniques are used to extract useful knowledge [9][10][11]. The application of DM techniques with success can be found in a wide and diversified range of applications. One important application is in BI systems.

BI is the top level of a complex system. On its foundations lay several databases, usually based in the relational model for databases [12], that can be accessed and manipulated using specific database (DB) languages, such as SQL and Query-By-Example (QBE). On the next level, data warehouses (DW) can be manipulated using exactly the same sort of languages. Applying DM to data stored on both databases (DB) and data warehouses (DW), knowledge bases (KB) arise on the next level. KB store DM models and, traditionally, are not based on the relational model, unlike DB and DW. Nevertheless, using the framework of inductive databases (IDB), DM models can be stored in databases in the same way as data, thus DM models can be accessed and manipulated at the same level than data [13][14][15]. "Inductive databases tightly integrate databases with data mining. The key ideas are that data and patterns (or models) are handled in the same way, and that an inductive query language allows the user to query and manipulate the patterns (or models) of interest" [15, pp 69].

Using the framework of inductive databases, DM models can be obtained and manipulated through the use of DM languages, such as MineRule [16], DMQL [17], or MSQL [18]. Table I presents a comparison of the syntax of these SQL-based DM languages. The three languages are SQL extensions. The extensions are made through the implementation of a new operator that allows obtaining the DM models, namely "find classification rules" operator for DMQL, "MINE RULE" operator for Mine Rule, and "Get rules ... into ..." operator for MSQL.

TABLE I. COMPARISON OF SQL-BASED DM LANGUAGES SYNTAX [19]

Schema: student(id,gender,age,nenroll,grant,grade) Classification Rules for grade in consequent Having grade<10; support>0.1; confidence>0.2	
DMQL	use database school find classification rules as Classification Rules according to grade Related to gender, age, nenroll, grant From student Where student.grade<10 With support threshold > 0.1 With confidence threshold > 0.2
MineRule	MINE RULE ClassificationRules AS SELECT DISTINCT gender, age, nenroll, grant AS BODY, grade AS HEAD FROM student WHERE grade<10 EXTRACTING RULES WITH SUPPORT: 0.1, CONFIDENCE: 0.2
MSQL	GetRules (student) Into ClassificationRules Where consequent is {(grade<10)} and body in {(gender=*), (age=*), (nenroll=*), (grant=*)} and confidence > 0.2 and support > 0.1

These languages are very important. But, just like SQL, they are not business oriented, are not oriented to business users and are not oriented to BI activities. This is a crucial issue in organizations that is gaining momentum each day.

Codd's relational model for databases has been adopted long ago in organizations. Initially, two formal languages were defined for relational databases: relational algebra and relational calculus [20][12]. Since that time, several languages were developed in order that business users could access data stored in databases. Query-By-Example (QBE) languages [21] were developed with success. The use of QBE languages by business users in order to directly obtain answers to ad-hoc business questions is a usual practice in organizations nowadays. QBE languages are declarative, also called nonprocedural or very high level, languages. By using this type of languages the user defines "what s/he wants to do" instead of defining "how to do it", which is typical of imperative languages. According to Zloof, Query-by-Example is: "a high-level database management language that provides a convenient and unified style to query, update, define, and control a relational database. The philosophy of Query-by-Example is to require the user to know very little in order to get started and to minimize the number of concepts that s/he subsequently has to learn in order to understand and use the whole language." [22, pp 324]. QBE languages are business oriented; moreover they are oriented to business users and to BI activities.

III. Research Methodology and Obtained Results

In this research a BI system including DM was conceived and implemented. An architecture that allows an effective usage of DM with BI by business users in order to conduct to DM integration with BI, was envisaged. This architecture should bring DM into the front line business users, be iterative, visual, and understandable by front line business users, and work directly on data. Following these guidelines, an architecture for integration of DM with BI is presented in Figure 1. This architecture intends to conduct to an effective usage of DM in BI.

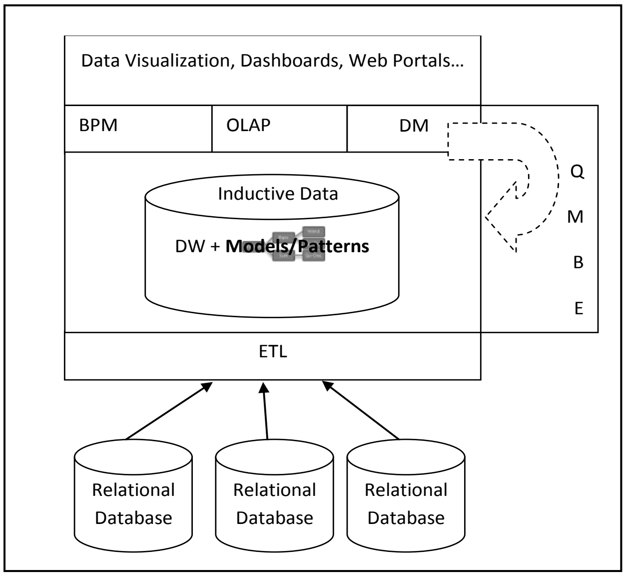

Figure 1. Architecture for integration of Data Mining with Business Intelligence.

The DM module extracts data from the DW, generates the DM models, and feeds the database with DM models. There is the possibility to include as many models as needed by the user, and new models can be included just by adding a new table.

This architecture implements the concept of Inductive Data Warehouse (IDW), which is a data warehouse storing data and data mining models at the same level, that is to say, both data and DM models are stored in data warehouse tables and can be accessed and manipulated in the same way.

An important aspect is the inductive language. Thus a new language, named as QMBE (Query Models By Example), was developed and implemented as an extension of a QBE language. Using QMBE the user is, thus, able to interact directly with the models, and to construct queries including different criteria. Table II presents several business questions commonly posed by business users involving DM models. All the business questions can be converted into queries to the system, defined in the QMBE language.

Since QMBE is an extension of QBE language, by nature it has two important characteristics, which are interactivity, and iterativeness. These characteristics are inherited from QBE languages upon which QMBE is extended. The novelty of the QMBE language is that it is oriented to business users and to BI processes. This kind of approach allows business users to directly access and manipulate data and models. This will bring DM to the front line business users, alike other BI tools, thus allowing DM integration with BI.

TABLE II. Business questions involving DM models

Queries on models	Queries on models and data
What are the characteristics of "good" students?	Select the actual students that can be "good" students.
What are the characteristics of "bad" students?	Select the actual students that can be "bad" students.
What are the characteristics of the students that do not conclude the grades according to initial schedule?	Select the actual students that cannot conclude the grades according to initial schedule.
Are there different types of students in the school?	…
…	…

Following, the concept of IDW, and QMBE language are presented.

A. Inductive Data Warehouse

In the context of BI there can be said that an IDB contains both the DW and the KB, that is to say, the DM models. This way, we can refer to this database as an Inductive Data Warehouse (IDW). Thus, an IDW is a DW which includes data and DM models, both stored in tables of the DW. This is an important concept in the realm of this research, since it focuses on making DM available to business users. In an IDW data and DM models can be accessed by business users in the same way as data. The DM models are stored in the DW in specific tables: the model tables. It is possible to include several model tables, one for each generated model.

In this research, the generated DM model corresponds to rules, since these were considered adequate for the problem under study. A rule is an IF-THEN expression of the form *IF antecedent THEN consequent*, written as:

$$antecedent \Rightarrow consequent$$

where antecedent and consequent are propositions of the form

$$V_1\ cond_1\ C_1\ \ AND\ \dots\ AND\ V_N\ cond_N\ C_N$$

where $V_1, \dots, V_N$ are variables; $C_1, \dots, C_N$ are constants; and $cond_1, \dots, cond_N$ stands for < or > or = or <= or >=.

In the case of classifications rules, the consequent is of the form:

$$V_i\ cond_i\ C_i$$

where V_i is the target variable; C_i is a constant; and $cond_i$ stands for < or > or = or <= or >=.

Usually BI systems are supported by special databases, namely DW. For the sake of generality, consider a DW with one fact table named FACT_TABLE, and N dimension tables named DIMENSION_1, DIMENSION_2, DIMENSION_3, ..., DIMENSION_N. The fact table has one ID column, and N columns, Dimension1, Dimension2, Dimension3, ..., DimensionN, each corresponding to one dimension table, and a column Fact. Each of the dimension tables has got several columns, each one corresponding to a variable that can be selected for DM. Consider for instance that DIMENSION_J has M_j variables, namely, IDJ, VarJ1, VarJ2, .., VarJI, ..., $VarJM_J$.

In an IDW, DM models are stored in the database in one, or more, specific table, or tables. Without losing generality, hereby only one table will be considered and named MODEL_TABLE. The first column of the model table, ID, corresponds to the rule identifier. The next two columns, confidence and support, stand respectively for the rule confidence and for the rule support. The following column corresponds to the selected DM target variable that corresponds to one of the columns of one of the dimension tables. The L variables selected for data mining, each one corresponding to a column of one of the dimension tables included in the DW, form the rest of the table columns, namely, DMVar1, DMVar2, ..., DMVarL. Keep in mind that DMVar1, DMVar2, … DMVarL of MODEL_TABLE are selected from all the columns of tables DIMENSION_1, or DIMENSION_2, …, or DIMENSION_N. Thus, all the columns of the MODEL_TABLE are the same as some column of the dimension tables. In this manner MODEL TABLE is connected to the DW tables.The IDW general schema is presented in Figure 2.

MODEL_TABLE (ID, confidence, support, DMTarget, DMVar1, DMVar2, …, DMVarL)
FACT_TABLE (ID, Dimension1, Dimension2, ..., DimensionN, Fact)
DIMENSION_1 (ID1, Var11, Var12, …, Var1I, …, $Var1M_1$)
DIMENSION_2 (ID2, Var21, Var22, …, Var2I, …, $Var2M_2$)
DIMENSION_3 (ID3, Var31, Var32, …, Var3I, …, $Var3M_3$)
…
DIMENSION_J (IDJ, VarJ1, VarJ2, …, VarJI, …, $VarJM_J$)
…
DIMENSION_N (IDN, VarN1, VarN2, …, VarNI, …, $VarNM_N$)

Figure 2. IDW General Schema.

Each rule is introduced in the MODEL_TABLE as a line of the table. Data is introduced in a cell of the table whenever there is a constraint in the rule for the correspondent variable, and is left blank (NULL) elsewhere. Consider, for instance, a general rule:

Rule I:
DMVar1 cond1 Value1 AND ... AND DMVarK condK ValueK AND ... => DMTarget condT ValueT; where cond1, ..., condK, condT stands for < or > or = or <= or >=.

Then the line (tuple) that corresponds to that rule is:
(I, valueC, ValueS, condT ValueT, cond1 Value1, ..., condK ValueK, ...).

New models can easily be added to the IDW by the simple introduction of model tables in the IDW, one for each model.

B. QMBE Language

In the research described in this paper, a new language, named Query Models by Example (QMBE) was developed as an extension of QBE languages existing in some Relational Database Management Systems (RDBMS). Similarly to QBE languages, upon which QMBE is based on, QMBE is a declarative, also called nonprocedural or very high level, language. By using this type of languages the user defines "what s/he wants to do" instead of defining "what to do", which is typical of imperative languages.

Business users are able to interact directly with the models, and to construct queries as a way to obtain answers to ad-hoc business questions. Business questions can be converted into queries to the system, defined in the QMBE language. Like in RDBMS QBE languages, the user will be able to define different criteria, considered significant to business. Business questions can be converted into queries in the QMBE language. To construct the query, the user will have to fill in a skeleton table (Figure 3).

Table →				
Column →				
Criteria →				

Figure 3. Skeleton table for the QMBE language.

The user will have to identify which are the tables, in the first line of the skeleton table; the corresponding columns that have the necessary data to answer the intended business question will have to be identified in the second line of the skeleton table. Specific criteria can be defined for each selected column, in the next lines of the skeleton table. More than one line can be considered for criteria. If criteria are defined in the same line, they are linked with AND. If criteria are defined in different lines, they are linked with OR. There can be considered three types of QMBE queries, namely:

- queries on data, corresponding to traditional QBE languages;
- queries on models, corresponding to QMBE extensions; and
- queries on models and data, corresponding also to QMBE extensions.

In all these three cases, examples of business questions will be presented based on the IDW schema from Figure 2. There will also be presented the correspondent queries in QMBE, as well as the relational calculus sentences that correspond to each one of those QMBE queries. Relational calculus is based in a branch of mathematical logic called predicate calculus [20]. QBE languages are connected with relational calculus and so is QMBE. Just like for traditional QBE queries, all QMBE queries can be written as relational calculus queries.

 ISBN: 978-1-61208-224-0

TRADITIONAL QBE: QUERIES ON DATA

Generally speaking, queries on data involve columns from any of the tables of the IDW, except the MODEL_TABLE, for instance DIMENSION_J and FACT_TABLE. Similarly, criteria can be defined for any column.

Business Question I

What are the data from Dimension J table, which corresponds to Fact (of FACT TABLE) equal to a certain value (value)?

QMBE query I

The query is presented in Figure 4.

Table →	FACT_TABLE	DIMENSION_J	…	DIMENSION_J
Column →	Fact	Var11	…	Var1N
Criteria →	value			

Figure 4. QMBE query I.

Relational Calculus Query 1

Q1 = {f.Fact, d | FACT_TABLE(f) AND DIMENSION.J(d) AND f.Fact = value}

QMBE EXTENSIONS: QUERIES ON MODELS

Generally speaking, queries on models may involve any of the columns of the MODEL_TABLE and criteria can be defined for any column.

Business Question J

What are the rules of MODEL TABLE which correspond to DM Target equal to a certain value (value)?

QMBE query J

The query is presented in Figure 5.

Table →	MODEL_TABLE	MODEL_TABLE	…	MODEL_TABLE
Column →	DMTarget	DMVar1	…	DMVarL
Criteria →	value			

Figure 5. QMBE query J.

Relational Calculus Query J:

QJ = {m | MODEL TABLE(m) AND m.DMTarget=value}

QMBE EXTENSIONS: QUERIES ON MODELS AND DATA

Queries on models and data may involve columns from all the tables of the IDW and criteria can be defined for any column.

Business Question K

What are the data from DIMENSION J which corresponds to a pre-selected rule from MODEL TABLE, for instance, rule I?

QMBE query K:

The query is presented in Figure 6.

Table →	DIMENSION_J	…	DIMENSION_J	…	DIMENSION_J	...
Column →	VarJ1		$VarJI_1$		$VarJI_K$	
Criteria →			cond1 Value1		condK ValueK	

Figure 6. QMBE query J.

Relational calculus Query K:

QK = {dJ | DIMENSION_J(dJ) AND VarJI1 cond1 value1 AND … AND VarJIK condK ValueK AND …}

IV. DISCUSSION, LIMITATIONS, AND FUTURE RESEARCH DIRECTIONS

As a consequence of being an extension of a QBE language, this new DM language is iterative and interactive in nature. It allows business users to answer to ad-hoc business questions through queries on data or/and on DM models. QMBE allows business users to directly access and manipulate DM models. The novelty of the QMBE language is that it is oriented to business users and to BI activities. This kind of approach allows business users to directly access and manipulate data and models. This will bring DM to the front line business users, like other BI tools, allowing them to completely exploring DM potential value.

The presented architecture is being implemented as a prototype. One limitation is that, at the moment, the system is not completely automated. Another limitation is that only rules are addressed at the moment. Nevertheless, rules will be followed by clustering. This is due to the application domain, which focuses on these two DM tasks.

User interface is also a concern.

The architecture, including IDW and QMBE language, was implemented as a prototype and used in different and controlled situations, proving that the concepts are viable and can be applied in the considered environments, of BI systems using DM. Only this conceptual evaluation has been made at the moment, but a questionnaire is planned in order to obtain business users opinions.

It is expected that when tests are finished the system will be integrated in a real situation. The authors hope that the implementation on a real situation could help to bring new useful insights.

V. CONCLUSION

The goal of the research presented in this paper is to allow business users to manipulate directly DM models, thus being able to explore completely their potential business value. This is achieved by means of the use of the IDB framework in the area of BI, presenting the concept of IDW. Also, a new data mining language for BI, named as QMBE,

 ISBN: 978-1-61208-224-0

which is oriented to BI activities as well as oriented to business users, was developed. QMBE is presented as an extension of traditional QBE languages, which are included in most of the RDBMS nowadays.

The authors have introduced a BI systems' architecture that allows final business users to directly access and manipulate DM models and consequently being able of extracting their full potential business value. Consequently, the business value contained in DM models could be completely explored in BI systems that incorporate DM tools. This was achieved through means of two new important concepts: the concept of Inductive Data Warehouse and a new DM language, QMBE, which is iterative, and interactive in nature. By using this language, business questions can be converted into queries in the QMBE language, thus it is oriented to BI activities and to BI business users. This will allow business users to directly manipulate DM models, as well as data, thus bringing DM into the front line business personnel, allowing to increase DM potential to attaining BI's high potential business value.

This new DM language is extensible and flexible, since several DM models could be added just by adding new model tables to the IDW, and it is context independent, since it can be applied to any DW.

The main contribution of this paper is to verify the viability of allowing business users of BI systems to directly manipulate DM models and thus providing the possibility of exploring the potential value of applying DM in the context of BI.

REFERENCES

[1] E. Turban, R. Sharda, J. E. Aroson, D. King, Business Intelligence: A Managerial Approach, Pearson Prentice Hall, Upper Sadle River, NJ, 2008.

[2] E. Turban, J. E. Aroson, T. Liang, R. Sharda, Decision Support and Business Intelligence Systems, Pearson Prentice Hall, Upper Sadle River, NJ, 2007.

[3] G. Piatetsky-Shapiro, Data Mining and Knowledge Discovery 1996 to 2005: Overcoming the Hype and Moving from "university" to "business" and "analytics", Data Mining and Knowledge Discovery 15(1), 2007, pp: 99-105.

[4] H. Kriegel, K. M. Borgwardt, P. Kröger, A. Pryakhin, M. Schubert, A. Zimek, Future Trends in Data Mining, Data Mining and Knowledge Discovery 15(1), 2007, pp: 87-97.

[5] W. McKnight, Bringing Data Mining to the Front Line, Part 2, Information Management magazine November(2002), 2003, pp: Retrieved on July, 16th 2009 at http://www.information-management.com/issues/20021101/5980-1.html.

[6] W. McKnight, Bringing Data Mining to the Front Line, Part 1, Information Management magazine November(2002), 2002, pp: Retrieved on July, 16th 2009 at http://www.information-management.com/issues/20021101/5980-1.html.

[7] R. J. Brachman, T. Anand, The Process of Knowledge Discovery in databases, In U. M. Fayyad, G. Piatetski-Shapiro, P. Smyth, R. Uthurusamy (Eds.), Advances in knowledge discovery and data mining, AAAI Press/The MIT Press, Menlo Park, CA, 1996, pp: 37-57.

[8] U. M. Fayyad, G. Piatetski-Shapiro, P. Smyth, From data mining to knowledge discovery: an overview, In U. M. Fayyad, G. Piatetski-Shapiro, P. Smyth, R. Uthurusamy (Eds.), Advances in knowledge discovery and data mining, AAAI Press/The MIT Press, Menlo Park, California, 1996, pp: 1-34.

[9] E. Simoudis, Reality Check for Data Mining, IEEE Expert 11(5), 1996, pp: 26-33.

[10] U. M. Fayyad, Data Mining and Knowledge Discovery: Making Sense Out of Data, IEEE Expert 11(5), 1996, pp: 20-25.

[11] M. Chen, J. Han, P. S. Yu, Data Mining: An Overview from a Database Perspective, IEEE transactions on Knowledge and Data Engineering 8(6), 1996, pp: 866-883.

[12] E. F. Codd, A Relational Model of Data for Large Shared Data Banks, Communications of the ACM 13(6), 1970, pp: 377-387.

[13] T. Imielinski, H. Mannila, A Database Perspective on Knowledge Discovery, Communications of the ACM 39(11), 1996, pp: 58-64.

[14] S. Dzeroski, Towards a General Framework for Data Mining, in S. Dzeroski, J. Struyf (Eds.), Knowledge Discovery in Inductive Databases - 5th International Workshop, KDID 2006, Lecture Notes in Computer Science: Vol. 4747, Springer-Verlag, Berlin, Heidelberg, 2007, pp: 259-300.

[15] L. De Raedt, A perspective on Inductive Databases, SIGKDD Explorations 4(2), 2003, pp: 69-77.

[16] R. Meo, G. Psaila, S. Ceri, An Extension to SQL for Mining Association Rules, Data Mining and Knowledge Discovery 2(2), 1998, pp: 195-224.

[17] J. Han, Y. Fu, W. Wang, K. Koperski, O. Zaiane, DMQL: A Data Mining Query Language for Relational Databases, (), 1996, pp: 27-34.

[18] T. Imielinski, A. Virmani, MSQL: A Query Language for Database Mining, Data Mining and Knowledge Discovery 3(4), 1999, pp: 373-408.

[19] A. Azevedo, M. F. Santos, A Perspective on Data Mining Integration with Business Intelligence, In A. Kumar (Ed.), Knowledge Discovery Practices and Emerging Applications of Data Mining: Trends and New Domains, IGI Publishing, Hershey, NY, 2011, pp: 109-129.

[20] E. F. Codd, A Data Base Sublanguage Founded on the Relational Calculus, (), 1971, pp: 35-68.

[21] M. M. Zloof, Query-by-Example: a data base language, IBM Systems Journal 16(4), 1977, pp: 324-343.

[22] M. M. Zloof, S. P. de Jong, The System for Business Automation (SBA): Programming Language, Communications of the ACM 20(6), 1977, pp: 385-396.

 ISBN: 978-1-61208-224-0