

A Survey of the State of the Art in Data Mining and Integration Query Languages

Sabri Pllana, Ivan Janciak, Peter Brezany, and Alexander Wöhrer

Research Group of Scientific Computing

University of Vienna

Nordbergstrasse 15/C/3, 1090 Vienna, Austria

{pllana,janciak,brezany,woehrer}@par.univie.ac.at

Abstract—The major aim of this survey is to identify the strengths and weaknesses of a representative set of Data-Mining and Integration (DMI) query languages. We describe a set of properties of DMI-related languages that we use for a systematic evaluation of these languages. In addition, we introduce a scoring system that we use to quantify our opinion on how well a DMI-related language supports a property. The languages surveyed in this paper include: DMQL, MineSQL, MSQL, M2MQL, dmFSQL, OLEDB for DM, MINE RULE, and Oracle Data Mining. This survey may help researchers to propose a DMI language that is beyond the state-of-the-art, or it may help practitioners to select an existing language that fits well a purpose.

Keywords-Data mining, data integration, query languages

I. INTRODUCTION

Data integration, the process of combining data from different sources, has been an active research topic for many years [13]. A perfectly integrated information system would give the illusion that users interact with only one central, locally running, homogeneous and consistent information system providing location-, schema- and language transparency [13]. Enabled by the network performance improvement, data integration research promises to have similar effects as data mining [12].

Data Mining (DM), also popularly referred to as Knowledge Discovery in Databases (KDD), is the automated or convenient extraction of patterns representing knowledge implicitly stored in large volumes of data [14], [24], [25]. In recent years there has been a rapid growth in data mining techniques [15]. Data mining has two main high-level goals: prediction and description. Commonly used methods for data mining are: association rules, sequential patterns, classification, regression, clustering, and change and deviation detection. From user's point of view, the execution of a data mining process and the discovery of a set of patterns can be considered either as an answer to a sophisticated database query or as a result of an execution of a data mining workflow. The first is called the *descriptive approach*, while the second is the *procedural approach*.

According to the CRoss Industry Standard Process for Data Mining (CRISP-DM) [21], the life-cycle of a data mining project consists of six phases. These phases include: *business understanding* (understanding the project objectives

and requirements); *data understanding* (get familiar with the data and identify data quality problems); *data preparation* (also known as data pre-processing, is about how to construct the final dataset); *modelling* (data mining techniques are selected and applied and their parameters are calibrated); *evaluation* (review the steps executed to construct the model to be certain it properly achieves the business objectives); *deployment* (what actions will need to be carried out in order to actually make use of the created models).

On-Line Analytical Processing (OLAP) is an approach for providing fast answers to analytical queries that are dimensional in nature. Systems configured for OLAP often employ a multidimensional data model, also called a cube, allowing for complex analytical and ad-hoc queries with a rapid execution time.

The DMI query languages developed so far support the above mentioned functionality at different level, as discussed in this survey.

A. Related Work

For obvious reasons commercially-available surveys are not discussed here. Academic surveys related to our work are scarce. A survey of data mining query languages for the domain of multimedia databases is presented in [3]. In [4] authors present a survey of 43 data mining tools. In contrast to the related work our survey is not limited to a particular domain (such as multimedia) or to data mining tools, but it considers both *data mining* and *integration* aspects of a representative set of languages.

B. Contributions

In this paper we systematically evaluate a representative set of state-of-the-art DMI-related languages: (1) DMQL, (2) MineSQL, (3) MSQL, (4) M2MQL, (5) dmFSQL, (6) OLEDB for DM, (7) MINE RULE, and (8) Oracle Data Mining. For each language we provide an overview and we state the interesting features and drawbacks that we have observed.

For the purpose of systematic language evaluation, we have developed the following list of properties (or criteria): data integration, data pre-processing, OLAP, presentation, DM methods, implementation, granularity, model processing, CRISP-DM compliance, background knowledge, and

extensibility. For each property we provide a single paragraph description on how well the language supports the property. We use our scoring system introduced in Section II to quantify our opinion on how well a DMI-related language supports selected properties: insufficient, poor, sufficient, good, very good, excellent.

The rest of this paper is structured as follows. The list of properties and the scoring system is described in Section II. In Section III we systematically evaluate the selected DMI-related languages. The results of the evaluation are summarised in Section IV. Section V concludes this paper.

II. EVALUATION APPROACH

In this section we describe a set of properties of DMI-related languages. We use these properties for our subsequent systematic evaluation of DMI-related languages. In addition, we introduce the scoring system that we use to quantify our opinion on how well a DMI-related language supports a property.

Our intention was to develop a comprehensive but manageable set of properties. The following properties are used in our survey:

- p1 **Data integration.** Indicates whether the language provides constructs for integration of data from various data sources that may be geographically distributed.
- p2 **Data pre-processing.** Indicates whether the language provides constructs for data pre-processing (such as selection, cleaning, or transformation).
- p3 **OLAP.** Indicates whether the language provides constructs for OLAP.
- p4 **Presentation.** Indicates whether the language supports visualization of results, and specification of the output format (or type).
- p5 **DM methods.** Indicates the DM methods that are supported.
- p6 **Implementation.** Indicates whether and how well the language was implemented. In addition, it indicates the degree of the implementation portability.
- p7 **Granularity.** Indicates the specification level (such as operator-level, algorithmic-level, or task level) and parameterization capabilities.
- p8 **Model processing.** Indicates the support for model verification, evaluation, validation, visualization, or reuse.
- p9 **CRISP-DM compliance.** Indicates the support for phases and tasks of the CRISP-DM reference model.
- p10 **Background knowledge.** Indicates the support for concept hierarchies and type mapping needed by various data mining techniques.
- p11 **Extensibility.** Indicates the easiness of extension with new language constructs.

The following grades (from 0 to 5) are used in our scoring system for quantifying how well a DMI-related language supports a property:

- **Insufficient** (0). Indicates that the property is not supported at all.
- **Poor** (1). Indicates that there is a little support for the property; it only scratches the surface.
- **Sufficient** (2). Indicates that there is a sufficient support for the property, but many relevant improvements are possible.
- **Good** (3). Indicates that there is a good support for the property, but some obvious improvements could be made.
- **Very good** (4). Indicates that there is a very good support for the property, but some small improvements are possible.
- **Excellent** (5). Indicates that there is an excellent support for the property. To the best of our knowledge, nothing relevant could be added.

III. THE SURVEY

In this section we systematically evaluate a set of state-of-the-art DMI-related languages. The following languages are considered in our survey: (1) DMQL, (2) MineSQL, (3) MSQL, (4) M2MQL, (5) dmFSQL, (6) OLE-DB for DM, (7) MINE RULE, and (8) Oracle Data Mining.

A. *LI: DMQL*

The DMQL language was designed within the DBMiner [2] project for mining several kinds of knowledge in relational databases. The language allows the user to specify templates of rules to be discovered, called metapatterns. These metapatterns can be used to impose strong syntactic constraints on rules to be discovered. Portions of the language have been implemented in the DBMiner system for interactive mining of multiple-level knowledge in relational databases [1].

The language consists of the specifications of four major primitives that should be specified in a data mining request:

- 1) Set of data in relevance to a data mining process
- 2) Kind of knowledge to be discovered
- 3) Background knowledge (e.g., concept hierarchy)
- 4) Justification of the interestingness of the knowledge (i.e., thresholds)

Additionally, the language provides syntax for five different data mining patterns: generalized relations, characteristic rules, discriminant rules, classification rules, and association rules.

Evaluation of Properties

- p1 *Data integration.* The language does not provide any constructs specific for data integration. Although the language is based on the SQL syntax, the constructs such as JOIN are supported only for local data-sets.
- p2 *Data pre-processing.* The language does not support any constructs specifically for data cleaning and transformation, but data selection is supported by the

FROM, WHERE and HAVING constructs that are allowed as constraints for data selection queries.

- p3 *OLAP*. OLAP is not supported. The OLAP specific functions like roll up and drill down are used in the language but have different meaning because they can be only applied on the mined rules to traverse attribute hierarchies.
- p4 *Presentation*. The language uses a display as a construct to specify presentation of the discovered patterns in forms including rules, tables, crosstabs, pie, bar chart, decision trees, cubes, curves, or surfaces.
- p5 *DM methods*. The language provides specifications for the following data mining methods: generalized relations, characteristic rules, discriminant rules, classification rules, and association rules.
- p6 *Implementation*. The language has been fully implemented in the DBMiner system.
- p7 *Granularity*. The language operates on the level of data mining methods and allows users only limited possibilities to specify some threshold parameters.
- p8 *Model processing*. The language provides some constructs for model visualization.
- p9 *CRISP DM compliance*. The language supports only the modelling phase of the CRISP-DM reference model.
- p10 *Background knowledge*. The language uses concept hierarchies in order to express mined knowledge at different levels.
- p11 *Extensibility*. DMQL can be easily extended by standard SQL constructs or by data mining methods.

p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11
1	2	0	4	2	5	2	1	1	2	4

Table I
SUMMARY OF EVALUATION OF PROPERTIES FOR DMQL.

B. L2: MineSQL

MineSQL is a declarative, SQL like language developed at the Institute of Computing Science, Poznan University of Technology. MineSQL is an extension of SQL and serves as a uniform Application Programming Interface (API) for building business applications dealing with knowledge discovery [5]. Association, characteristic, classification and discriminant rules are supported. Using these techniques, MineSQL enables the user to express rule queries. Moreover, MineSQL provides support for conceptual hierarchies and generalized and multiple level rules. Rule discovery, storage and manipulation are supported as well. For storing generated rules in relational databases it defines a new data type: the RULE. The rules can be integrated into existing databases and are searchable via a single query. The RULE data type supports various functions like support(), confidence(), body(), etc; which provide access to

the rule elements as well as facilitate the conversion of values between RULE and other data types. To generate rules, MineSQL introduces a data discovery statement called MINE. Moreover, the MINE statement can be used as a sub-query in another statement, e.g. SELECT, or INSERT.

Some of the interesting features on MineSQL are:

- Support for storing rules for further use in the same database.
- The ability to search the rules as well as the data in one query.
- The MINE statement can be used as a sub-query.

Limitations of MineSQL include:

- Support only for a limited set of data mining techniques.
- Rule discovery/querying issues are not adequately addressed.

Evaluation of Properties

- p1 *Data integration*. There is no evidence in the studied literature that MineSQL supports data integration, although the integration of local data-sets via JOIN is supported.
- p2 *Data pre-processing*. The data pre-processing steps like selection, cleaning and transformation are partially supported; MineSQL has strong support for data selection through the FROM and WHERE clauses of MINE statements but has little or no support for cleaning and transformation.
- p3 *OLAP*. MineSQL does not provide constructs for OLAP.
- p4 *Presentation*. MineSQL only displays the generated rules but has no support for visualization of results and/or specification of output format.
- p5 *DM methods*. A basic set of DM methods i.e. association rule, characteristic rule, classification rule, conceptual hierarchy and generalization rule; are supported and enable the user to express his/her queries.
- p6 *Implementation*. A research prototype system has been implemented.
- p7 *Granularity*. A user can specify requirements at the task level. MineSQL provides FOR and TO clauses of MINE statements through which the user can specify the body and head of rule he/she is interested in.
- p8 *Model processing*. Model processing is partially supported.
- p9 *CRISP DM compliance*. Data preparation and data modeling are supported.
- p10 *Background knowledge*. MineSQL allows the definition and construction of taxonomy and generalized multiple level and multi-attribute association rules.
- p11 *Extensibility*. MineSQL is developed from SQL, and therefore may be easily extensible with new constructs.

p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11
1	3	0	1	3	2	3	1	1	3	3

Table II
SUMMARY OF EVALUATION OF PROPERTIES FOR MINESQL.

C. L3: MSQL

MSQL was designed by Imielinski and Virmani in 1999 [6]. This language supports association rule-mining. The language has the following properties [8]:

- It is an extension of SQL.
- It supports the closure property in which saved results of the queries can be query.
- It supports cross-over operations between data and rules i.e. the rules can be mapped back to the source data from which these are first extracted.
- The language can generate rules and query those rules using the same syntax.

MSQL rule generation is based on descriptors [7]. A descriptor is basically a pair of attribute names and values. MSQL defines support and confidence thresholds with which to retrieve the rule. The resultant rule body can consist of multiple attributes, while its head will have only one attribute. The syntax of this language supports a set of new data mining operators. For instance, the operator GETRULES generates the association rules.

Evaluation of Properties

- p1 *Data integration*. There is no evidence in the studied literature that MSQL supports data integration.
- p2 *Data pre-processing*. It has partial support for the pre-processing, i.e. it can convert continuous data into discrete data, but it lacks other pre-processing tasks like sampling.
- p3 *OLAP*. There is no support for OLAP.
- p4 *Presentation*. MSQL offers some post processing primitives using the operator SelectRules.
- p5 *DM methods*. MSQL only supports association rule mining.
- p6 *Implementation*. MSQL was designed by Imielinski and Virmani in 1999 at the Rutgers University, New Jersey, USA. It is an academic project; we found no information on implementation.
- p7 *Granularity*. The specification level of MSQL is operator level. It supports operators such as GetRules or SelectRules.
- p8 *Model processing*. There is no support for model processing.
- p9 *CRISP DM compliance*. It supports data modelling.
- p10 *Background knowledge*. There is no support for background knowledge.
- p11 *Extensibility*. The language is designed only for association rules, so it is difficult (if not impossible) to accommodate other models for extending the language.

p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11
0	2	0	1	1	2	3	0	1	0	1

Table III
SUMMARY OF EVALUATION OF PROPERTIES FOR MSQL.

D. L4: M2MQL

M2MQL [9] is a declarative language and it is compatible with SQL (it is an extension of SQL). It focuses on nested relational algebra for integrating multi-step association rule mining. It is based on the formal semantics sustained by the nested relational algebraic operations. M2MQL supports association rule mining primitives, which provide users with the flexibility to reuse the mining results and to conduct mining tasks based on them.

M2MQL supports the following kinds of association rules:

- Simple item-based association rules, such as rules that satisfy the minimal thresholds.
- Generalized item-based association rules.
- Quantitative association rules.

The language syntax provides CREATE RULE PRIMITIVE statement that is used for rule type-declaration, rule schema-illustration and rule generation. The SELECT RULE PRIMITIVE statement provides interactive processing of generated rule sets and original data sets, thereby supporting the reuse of existing rule sets in other data mining queries.

Evaluation of Properties

- p1 *Data integration*. There is no evidence in the studied literature that M2MQL supports data integration.
- p2 *Data pre-processing*. There is no support for pre-processing.
- p3 *OLAP*. There is no support for OLAP.
- p4 *Presentation*. It provides select and project operators to view the result of mined rules.
- p5 *DM methods*. It only supports association rule mining.
- p6 *Implementation*. There is no implementation.
- p7 *Granularity*. The specification level of M2MQL is at operator level.
- p8 *Model processing*. There is no support.
- p9 *CRISP DM compliance*. It supports the data modelling phase.
- p10 *Background knowledge*. There is no support for background knowledge.
- p11 *Extensibility*. The language is designed only for association rules, so it is difficult to extend the language with other kinds of models.

p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11
0	0	0	1	1	0	3	0	1	0	1

Table IV
SUMMARY OF EVALUATION OF PROPERTIES FOR M2MQL.

E. L5: dmFSQL

dmFSQL (data mining Fuzzy Structured Query Language) [16] is an extension of FSQL for data mining. Since FSQL is an extension of the SQL language, then dmFSQL inherits all the properties of SQL. FSQL extends the SQL language with the capabilities to specify flexible queries to address tables that store vague information using fuzzy attributes. dmFSQL introduces a new type of object that does not exist in FSQL. This object holds the initial conditions, intermediate and end results of the DM process. These intermediate results are used to improve the performance of the iterative DM process.

Advantages of dmFSQL include: (1) similar syntax to SQL; (2) it is understandable; (3) it is possible to express fuzzy conditions in a query; (4) supports iterative DM; and (5) management of any type of data. A drawback of dmFSQL is that it supports only a limited number of DM techniques: clustering and classification. Furthermore, the extensibility of dmFSQL with other DM techniques is considered to be difficult.

Evaluation of Properties

- p1 *Data integration.* There is no evidence in the studied literature that M2MQL supports data integration.
- p2 *Data pre-processing.* Data pre-processing is unsupported, except for SQL data-set selection function.
- p3 *OLAP.* dmFSQL does not provide constructs for OLAP.
- p4 *Presentation.* Like FSQL, dmFSQL also returns a SQL sentence. They have the same way to present the visualization of results.
- p5 *DM methods.* dmFSQL includes two DM methods: clustering and classification. They are often used, but represent only a small part of possible DM methods.
- p6 *Implementation.* dmFSQL may be used via the dmFSQL Server that has been programmed mainly in PL/SQL, or a new version of a prototype called DAPHNE [18] that incorporates the dmFSQL Server.
- p7 *Granularity.* As far as data selection is concerned, because of the properties of SQL, dmFSQL supports operator-level granularity. The DM method in dmFSQL is at task-level.
- p8 *Model processing.* Concerning, model processing, dmFSQL can be reused through the internal invocation: its classification process always needs the prior execution of a clustering process. It does not support model verification, evaluation, validation, or visualization.
- p9 *CRISP DM compliance.* dmFSQL supports partially Data Preparation and Modelling.
- p10 *Background knowledge.* dmFSQL can not support concept hierarchies. Also there is no functionality for type mapping.
- p11 *Extensibility.* dmFSQL is already a fixed and previously designed extension of FSQL. It is hard to extend. But it is possible, that on the basis of the DM methods in

dmFSQL, other DM methods can be described. How to implement this extension is not clear yet.

p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11
0	1	1	1	2	3	2	1	2	0	2

Table V
SUMMARY OF EVALUATION OF PROPERTIES FOR DMFSQL.

F. L6: OLE-DB for DM

OLE DB is an API comprising SQL capabilities for accessing various data sources. OLE-DB for DM provides the possibility of data mining on OLE-DB providers [19]. A provider is responsible for the implementation of the OLE-DB interfaces, whereas a consumer accesses data via OLE DB interfaces. Basically, the provider encapsulates the access to data and exposes it to consumers. The interface of OLE-DB for DM that is used for exploring and manipulating the data mining models is similar to the interface that is used for exploring tables. OLE DB for DM represents data mining models as table objects.

Evaluation of Properties

- p1 *Data integration.* As OLE-DB for DM is based on OLE-DB, it provides a rich infrastructure for data integration.
- p2 *Data pre-processing.* Since OLE DB for DM is based on OLE DB, it provides strong capabilities in data selection. It also supports data transformation.
- p3 *OLAP.* OLE-DB for DM alone possesses no OLAP capabilities, but can fulfill OLAP requirements via collaboration with OLE-DB for OLAP. OLE-DB for OLAP enables users to perform data analysis via interactive access to various views of the data.
- p4 *Presentation.* The DM interface supports model browsing for some models.
- p5 *DM methods.* It supports Decision Trees and Clustering.
- p6 *Implementation.* Microsoft has implemented an OLE-DB provider for Data Mining based on the OLE-DB for DM Specification.
- p7 *Granularity.* It operates at algorithm level.
- p8 *Model processing.* No support for model verification and similar processing tasks; it cannot post-process a model. But, the model can be reused to do prediction.
- p9 *CRISP DM compliance.* OLE-DB for DM (together with OLE-DB) supports Data Understanding, Data Preparation and Modelling.
- p10 *Background knowledge.* There is no support for concept hierarchies. There is no support (also no need) for type mapping in OLE-DB for DM, since OLE-DB performs type mapping before the process enters DM phase.
- p11 *Extensibility.* Microsoft data mining provider is open as it is an OLE-DB component. Other DM algorithms can be plugged into the same platform.

p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11
3	4	4	3	2	4	2	2	2	3	3

Table VI
SUMMARY OF EVALUATION OF PROPERTIES FOR OLE-DB FOR DM.

G. L7: MINE RULE

The MINE RULE language is an extension of SQL, which enables the extraction of association rules in relational databases [20]. The user specifies MINE RULE constraints and the format of the association rules that should be extracted. The MINE RULE language may extract the association rules directly from a database. It is also possible to use MINE RULE for rule extraction from data-sets that have previously been prepared for DM. Data selection and integration functionality is provided by the inherent SQL capabilities of MINE RULE.

Based on MINE RULE, many algorithms and optimization schemas for extraction of association rules have been developed. A design for an association rule-extracting system has also been proposed, although no implementation has been done.

Evaluation of Properties

- p1 *Data integration.* As MINE RULE inherits all SQL capabilities, it provides infrastructure to facilitate data integration inside one database. No effort is made to support data integration between different databases.
- p2 *Data pre-processing.* As MINE RULE inherent all SQL capabilities, it provides the possibility of data selection. However, it supports no normalization, transformation or cleaning functions specific for DM input.
- p3 *OLAP.* There is no interface defined in MINE RULE to support any external OLAP engine. On the other hand, MINE RULE barely supports the features of OLAP without any extension. For example, it does not offer multidimensional views on data; it cannot do statistical aggregation on the data, nor any other complex statistical calculation to support users with multidimensional views on the data. It cannot do time analysis on the data.
- p4 *Presentation.* MINE RULE does not offer the user with any visualization facility. The results from data selection and DM cannot be shown by any means within the context of this language. The results of DM are described as a database table.
- p5 *DM methods.* Association rules are supported.
- p6 *Implementation.* No real implementation has been found, although an association rule extraction system has been designed conceptually.
- p7 *Granularity.* For dataset selection, MINE RULE reaches operator-level (inherited from SQL). For DM, it behaves at task-level, as it only specifies the task (association rule or cluster or both) but not the algorithm.

- p8 *Model processing.* There is no support for model verification, and cannot post-process a model.
- p9 *CRISP DM compliance.* MINE RULE supports only Data Preparation and Modelling.
- p10 *Background knowledge.* MINE RULE does not support or recognize concept hierarchies. It offers no means for type mapping in data selection phase or between data selection and DM. According to [22] MINE RULE provides limited capabilities for background knowledge specification via setting conditions on the values of BODY and HEAD.
- p11 *Extensibility.* It is initially designed specifically for the extraction of association rules. But other DM methods can be added by introducing new DM operators. This kind of extension is, however, only possible at a task level.

p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11
1	3	1	1	2	2	3	0	2	0	2

Table VII
SUMMARY OF EVALUATION OF PROPERTIES FOR MINE RULE.

H. L8: Oracle Data Mining

Oracle Data Mining (ODM) [23] offers a data mining development and deployment platform inside Oracle databases, along with a GUI named Oracle Data Miner. The user uses the GUI to navigate through the process of creating, testing, and applying models approximating the CRISP-DM methodology. ODM provides a PL/SQL package to create, destroy, describe, apply, test, export and import models. According to METAspectrum's evaluation report on data mining tools, ODM is in the leader segment together with SAS and SPSS. Some of the interesting features of ODM are:

- Automatic data preparation for each model.
- It supports the Java standard for data mining (JSR-73).
- Bioinformatics Sequence Search and Alignment (BLAST) available in the Oracle Database via table functions.

Evaluation of Properties

- p1 *Data integration.* Data integration is supported via Oracle Gateway technology.
- p2 *Data pre-processing.* ODM provides utilities for data preparation steps prior to model building such as outlier treatment, discretization, and normalization.
- p3 *OLAP.* Support for OLAP is available via another option for Oracle Enterprise Edition, namely Oracle OLAP DML.
- p4 *Presentation.* Presentation is supported via ODM and MS Excel Add-In.
- p5 *DM methods.* Since Oracle version 11g, ODM supports data transformation and model analysis, classification,

regression, clustering, association rules, feature extraction as well as text and spatial mining.

- p6 *Implementation.* The platform has been fully implemented in Oracle RDBMS.
- p7 *Granularity.* ODM provides coarse grained granularity on SQL function call level.
- p8 *Model processing.* Model processing is supported.
- p9 *CRISP DM compliance.* The following phases are supported via the platform inside the Oracle RDBMS and ODM: data understanding, data preparation, modelling, evaluation and deployment.
- p10 *Background knowledge.* We found no clear statement in the literature that background knowledge is supported. But, considering the provided support for other properties, we may infer that background knowledge should be supported; other properties could not be supported to this extent without the background knowledge.
- p11 *Extensibility.* The data mining package can be extended via additional methods.

p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11
4	3	4	4	4	5	1	4	3	3	3

Table VIII
SUMMARY OF EVALUATION OF PROPERTIES FOR ORACLE DATA MINING.

IV. EVALUATION SUMMARY

In Figure 1, dark areas represent lack of the adequate support, whereas bright areas indicate excellent support of evaluated languages for properties.

Properties where the average grade is less than 2.0 (that is less than sufficient) are: *data integration*, *OLAP*, *model processing*, *CRISP-DM compliance*, and *background knowledge*. For these properties most of the surveyed languages provide less than *sufficient* support. The average grade for a property p_i is calculated as follows:

$$\frac{\sum_{j=1}^8 G(p_i, L_j)}{8}$$

Table IX shows the ranking of the surveyed languages based on average grades. The average grade for a language L_j is calculated using this formula:

$$\frac{\sum_{i=1}^{11} G(p_i, L_j)}{11}$$

Oracle Data Mining received the highest average grade (that is 3.5), whereas M2MQL the lowest one (that is 0.6).

V. CONCLUSIONS AND FUTURE WORK

We have presented a survey of the state-of-the-art of DMI-related languages. For each surveyed language we provided an overview and stated the interesting features and drawbacks that we have observed.

Rank	Language	Average Grade
1	L8: Oracle DM	3.5
2	L6: OLE-DB for DM	2.9
3	L1: DMQL	2.2
4	L2: MinieSQL	1.9
5	L7 MINE RULE	1.5
6	L5: dmFSQL	1.4
7	L3: MSQL	1.0
8	L4: M2MQL	0.6

Table IX
LANGUAGE RANKING BASED ON AVERAGE GRADES.

For the purpose of systematic language evaluation, we have developed a list of properties (or criteria) and a scoring system. For each property we have provided a single paragraph description on how well the language under this study supports the property. We used our scoring system to quantify our opinion. We have observed that most of the surveyed languages provide less than sufficient support for data integration, OLAP, model processing, CRISP-DM compliance, and background knowledge.

In future, based on our observations made during this survey, we plan to propose a model and the corresponding language that will advance the state-of-the-art in DMI languages.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement 215024 (ADMIRE Project, www.admire-project.eu).

REFERENCES

- [1] J. Han, Y. Fu, W. Wang, K. Koperski, O. Zaiane. DMQL: A Data Mining Query Language for Relational Databases. SIGMOD'96 Workshop on Research Issues in Data Mining and Knowledge Discovery, 1996.
- [2] J. Han et al. DBMiner: A System for Mining Knowledge in Large Relational Databases. In Proc. 1996 Int'l Conf. on Data Mining and Knowledge Discovery (KDD'96), pp. 250-255, 1996.
- [3] A. El-Mustafa et al. A survey of data mining query language (DMQL) for multimedia databases. Asia J. Inform. Technol., 4: 466-470, 2005.
- [4] M. Goebel and L. Gruenwald. A Survey of Data Mining and Knowledge Discovery Software Tools. SIGKDD Explorations. Vol. 1, Num. 1, pp. 20-33, 1999.
- [5] M. Zakrewicz. Data mining within DBMS functionality. In book Databases and information system, pp. 85-96, ISBN 0-7923-6823-1, 2001, Kluwer Academic Publishers, Norwell, MA, USA.
- [6] T. Imieliski, A. Virmani. MSQL: A Query Language for Database Mining. Data Mining and Knowledge Discovery, Volume 3 Issue 4, Kluwer Academic Publishers, December 1999.

	<table border="1"> <tr><td>0</td><td>insufficient</td></tr> <tr><td>1</td><td>poor</td></tr> <tr><td>2</td><td>sufficient</td></tr> <tr><td>3</td><td>good</td></tr> <tr><td>4</td><td>very good</td></tr> <tr><td>5</td><td>excellent</td></tr> </table>											0	insufficient	1	poor	2	sufficient	3	good	4	very good	5	excellent
	0	insufficient																					
1	poor																						
2	sufficient																						
3	good																						
4	very good																						
5	excellent																						
	Data integration	Data pre-processing	OLAP	Presentation	DM methods	Implementation	Granularity	Model processing	CRISP DM compliance	Background knowledge	Extensibility												
	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11												
L1: DMQL	1	2	0	4	2	5	2	1	1	2	4												
L2: MineSQL	1	3	0	1	3	2	3	1	1	3	3												
L3: MSQL	0	2	0	1	1	2	3	0	1	0	1												
L4: M2MQL	0	0	0	1	1	0	3	0	1	0	1												
L5: dmFSQL	0	1	1	1	2	3	2	1	2	0	2												
L6: OLE-DB for DM	3	4	4	3	2	4	2	2	2	3	3												
L7: MINE RULE	1	3	1	1	2	2	3	0	2	0	2												
L8: Oracle D. Mining	4	3	4	4	4	5	1	4	3	3	3												
Av. Property Grade	<u>1.3</u>	2.3	<u>1.3</u>	2.0	2.1	2.9	2.4	<u>1.1</u>	<u>1.6</u>	<u>1.4</u>	2.4												

Figure 1. Evaluation summary. Most of the surveyed languages provide less than sufficient support for data integration, OLAP, model processing, CRISP-DM compliance, and background knowledge.

- [7] J. Boulicaut, C. Masson. Data Mining Query Languages. Data Mining and Knowledge Discovery Handbook, 2nd ed. Springer 2010: 655-664.
- [8] M. Botta, J. Boulicaut, C. Masson, and R. Meo. Query Languages Supporting Descriptive Rule Mining: A Comparative Study. LNCS 2682 Springer 2004: pp. 24-51.
- [9] X. Yuan. Data mining query language design and implementation. Master's thesis, December 2003.
- [10] D. Kossmann. The state of the art in distributed query processing. ACM Computing Surveys (CSUR), 32 (4), 2000.
- [11] J. Gray. Distributed computing economics. MSR-TR-2003-24, 2003.
- [12] I. Foster et al. Data integration in a bandwidth rich world. CACM, 2003.
- [13] S. Busse et al. Federated Information Systems: Concepts, Terminology and Architectures. TR, University of Berlin, 1999.
- [14] J. Han and M. Kamber. Data Mining. Concepts and Techniques. Morgan Kaufman Publishers, 2001.
- [15] B. Park and H. Kargupta, Distributed Data Mining: Algorithms, Systems, and Applications, Data Mining Handbook, 2002.
- [16] R.A. Carrasco et al. dmFSQL: a Language for Data Mining. XVI Database and Experts Systems Applications, DEXA, IEEE, pp. 440-444, 2006.
- [17] R.A. Carrasco et al. dmFSQL: a Server for Data Mining. Workshop on Data Mining and Business Intelligence (DMBI2007). April 15, 2007, IEEE. Istanbul, Turkey, 2007.
- [18] A. Salguero et al. Applying Fuzzy Data Mining for Soaring Area Selection. LNCS 4507, pp. 597-605, Springer 2007.
- [19] Z. Tang, J. MacLennan, P. Kim: Building Data Mining Solutions with OLE DB for DM and XML for Analysis. The Data Mining and Knowledge Discovery Handbook 2005: pp. 1331-1345.
- [20] R. Meo, G. Psaila, MINE RULE: Semantic Dimensions in Association Rule Mining, in John Wang (ed.), Encyclopedia of Data Warehousing and Mining, ISBN: 1-59140-557-2. Information Science Publishing (an imprint of Idea Group Inc.), 2006.
- [21] CRoss Industry Standard Process for Data Mining (CRISP-DM). <http://www.crisp-dm.org/>.
- [22] F. Bonchi, F. Giannotti, C. Lucchese, S. Orlando, R. Perego, R. Trasarti: On Interactive Pattern Mining from Relational Databases. KDID 2006: pp. 42-62.
- [23] Oracle Data Mining White Paper. <http://www.oracle.com/technology/products/bi/odm/>
- [24] F. Khan, Y. Han, S. Pillana, and P. Brezany. An Ant-Colony-Optimization Based Approach for Determination of Parameter Significance of Scientific Workflows. AINA 2010: 1241-1248.
- [25] P. Brezany, I. Janciak, and A. Tjoa. GridMiner: An Advanced Support for e-Science Analytics. Book chapter in "Data Mining Techniques in Grid Computing Environments", pp. 37-55, John Wiley & Sons Ltd, 2008.