

Comments on the Microsoft draft standard (specification) for Data Mining

April 23, 2000

Microsoft with support from Data Mining companies (ANGOSS Software, Appsource, Comshare, DB Miner Technology, Knosys, Magnify, Megaputer Intelligence, Maximal Innovative Intelligence, NCR, PolyVista and SPSS) developed a draft standard for Data Mining (OLE DB for Data Mining, DRAFT Specification):

<http://www.microsoft.com/presspass/press/2000/Mar00/DataMiningPR.asp>

<http://www.microsoft.com/data/oledb/>.

This draft (Version 0.9) is open for a public discussion until May 15, 2000.

From our viewpoint the main goals of these specifications are

- 1) to unify terminology,
- 2) to unify, simplify and speed up communications between databases, data mining tools (called data mining services), mined knowledge (in the form of data mining models) and data mining final output (in the form of forecasts, ranking, distributions, associations, correlations and so on for a particular data set), and
- 3) to help to select (automatically) the most appropriate DM services/algorithms for a specific data set.

To solve these tasks Microsoft specified metadata. These metadata describe each data column (target column, column used for forecasting the target, numeric data formats, contents of the column, type of the possible DM model and so on).

Similarly metadata are specified for DM services, characterizing an algorithm's capabilities.

Some flexibility is permitted. DM services can add provider-specific metadata.

Potentially these two sets of metadata (database metadata and DM service metadata) can be **matched automatically** for selecting an appropriate Data Mining service. This productive idea of matching probably was most clearly illuminated by Dhar and Stein in the concept of problem ID [Intelligent Decision Support Methods, Prentice Hall, 1997].

It is critical that in order for this matching to occur that specifications catch the really **important features of both data and algorithms** and are flexible enough to incorporate future algorithm developments and improvements. From our viewpoint, the most sensitive component for matching is the **type of contents of columns**. Microsoft suggests the following flags for types of data contents: key, discrete, continuous, cyclical, ordering, probability distribution and so on. For instance, the flag PROBABILITY permits matching services working with probability distributions with databases, which contain probability data. However, the matching for discrete, ordered, continuous and some other content data types is not so obvious.

There are two **difficulties**:

- 1) Terminology (equal terms should have the same meanings for DM consumers and providers)
- 2) OLE DB for DM Grammar (Microsoft draft, p.80) should permit adequate matching.

The MS draft provides the following description for data contents.

Type Flags

KEY	The column is discrete and is a key. Key columns will not have any other flags except in the case of a nested table with no attribute columns.
CONTINUOUS	The column contains values in a continuous range, such as Age or Salary.
DISCRETE	The column contains a discrete set of values, such as Gender.
DISCRETIZED	The column contains a continuous set of values that should be converted to buckets.
ORDERED	The column contains a discrete set of values that are ordered, such as Salary Level.
CYCLICAL	The column contains an ordered discrete set of values that are cyclical, such as Day of Week, or Month.
SEQUENCE TIME	The column contains time measurement units.

Specifically DISCRETE is described as follows: “Even if the values are numeric, NO ORDERING is implied by the values. (“Area Code” is a good example.)”

This means that DISCRETE could be used to represent ordered data (salary levels) or unordered data (gender). No specific flag is provided for DISCRETE and UNORDERED data types. Two flags should be set up for this case: DISCRETE and UNORDERED, but the grammar presented by Microsoft, does not provide the flag UNORDERED.

According to Microsoft a column definition is one of the following forms:

```
<column name> <type>          [<content flags>] [<column relation>] [<prediction flag>]
<column name> TABLE          [<prediction flag>] ( < non-table column definition list > )
```

The fields <content flags> can be selected as one of the words: continuous, discrete, discretized, sequence_time, ordered and cyclical.

```
<col_content>      -> DISCRETE
                   | CONTINUOUS
                   | DISCRETIZED( [<disc_method> [, <numeric_const>]] )
                   | SEQUENCE_TIME
```

```
<col_content_qual>-> ORDERED
                   | CYCLICAL
```

We suggest adding a new flag to identify DISCRETE and UNORDERED using the term **NOMINAL** (or classification):

```
<col_content>      -> NOMINAL
                   | DISCRETE
                   | CONTINUOUS
                   | DISCRETIZED( [<disc_method> [, <numeric_const>]] )
                   | SEQUENCE_TIME
```

This term has been used a standard term in measurement theory for more than 30 years [P. Suppes, J. Zinnes, Basic Measurement Theory, in: Handbook of Math. Psychology, v. 1, 1963, Wiley].

We think that it will be more efficient to add the NOMINAL type directly into the grammar rather than to rely on non-unified terms provided by an individual vendor.

Why is it important to add the NOMINAL data type to the grammar directly?

The Microsoft draft provides the following example (Tree Model to Predict Credit Risk):

```
<?xml version="1.0"?>
<pml>
<statements>
<statement type = "CREATE" value = "Create Mining Model CreditTree1
( ID long key,
  Credit text discrete predict,
  Education text discrete,
  Age text discrete,
  Pay text discrete
) using microsoft_decision_trees
"/>
<statement type = "TRAIN" value = "Insert Into CreditTree1
( ID, Credit, Education, Age, Pay)
OPENROWSET("Microsoft.Jet.OLEDB.4.0",
  "data source=w:\test\demozero\credit.mdb",
  "SELECT ID, Credit, Education, Age, Pay FROM CreditTraining"
)
"/>
```

This example uses four DISCRETE columns: Credit (bad, good), Education (Bachelor, Partial College, High School, Partial High School,...), Age (Young, Middle, Old) and Pay (Weekly pay, Monthly salary,...). Credit, Education and Age are clearly ORDERED. On the other hand when one considers Pay, it is not so obvious what kind of order makes sense. But all four columns are described as DISCRETE without any specifics for Pay. Similarly a new discrete column Occupation (professor, student, composer, artist,...) can be added in this example and coded as 1,2,3,4,... Again, there is no obvious order for occupation. Therefore, a DM service should not consider codes 1,2,3,4 as ordered numbers. They are just labels and any meaningful data mining algorithm should treat them in this way, i.e., avoiding knowledge discovery computations which include relation ">" or "<", because they are not defined for "Occupation".

DISCRETE and UNORDERED (NOMINAL) is just one example from the large set of contents data types not represented in the draft grammar.

Another example would be the grades, which millions of students get each year at the universities and colleges. Professors give letter grades such as A, A-, B+, B, B-, C+, C, C-, D+, D, D-, F, and I (incomplete). The I is a grade (cell value), it is not the mark that the cell is empty. Is letter grade discrete, ordered data contents type? Without the grade I (incomplete) it is, but the I grade makes this data type special. There is no term for this data type in the grammar. Ignoring the I grade we might match this ORDERED column with a Data Mining algorithm (service) such as decision trees, which work properly with ordered data. However, letter grades are converted by the University registrar office into numeric values to compute GPA, using mapping such as 4 for A. In this way, ORDERED column can become DISCRETIZED or even CONTINUOUS, completely changing the set of applicable Data Mining Algorithms. Now we can apply the whole spectrum of numeric statistical methods and discover trends in student's performance by comparing GPA probability distributions for different subjects, professors, groups of students and so on.

It is not clear how the current draft can address this issue. If a DM service provider will get letter grades generated by professors without mapping them into numbers, it can prevent this service from using some of the most suitable algorithms. If the computed numeric column is provided instead of letter grades then the problem can be solved, but in this case a DM service will actually work with a secondary database. The development of the secondary database requires extra effort. In both scenarios information about the mapping should be available. The most natural place to keep this mapping would be metadata, which should accompany the original column.

However, we oppose the unrestricted extending of the grammar with more and more terms. Instead we suggest adding a **special reference** to metadata similar to the suggestion by Microsoft for Model ID (Model Catalog, Model Schema, Model name). This will be a reference for an **application programmer interface (API)**, which will represent the contents of the column, e.g., letter grade. It can be a C++ **header file** with contents data type description and an **implementation file** for member functions. In particular, the

above mentioned mapping for letter grades can be naturally implemented in this **OOP approach** and supplied to a DM service provider by a DM consumer. It can include the alphabet of the column and all meaningful operations and relations over them.

The set of these APIs can be developed by industry vendors or the DM volunteer community (similar to LINUX) and made publicly available. If this idea will be implemented it will impact not only Microsoft products but also many others.

We would suggest an open discussion on the subject. This OOP approach is outlined in the recent book [Data Mining in Finance: Advances in Relational and Hybrid Methods, by B. Kovalerchuk and E. Vityaev, Kluwer, 2000, see pp. 164, 169-186]. The study is based on the concept of measurement scales and homomorphisms for scales pioneered by Professor P. Suppes at Stanford University [Foundation of measurement, by D. Krantz, R.Luce, P. Suppes and A.Tversky, Academic Press, v.1-3, 1971, 1989,1900].

We omit a specific analysis of other flags such as CONTINUOUS and CYCLICAL, but we want to provide a few comments showing that they also should be analyzed more closer.

The Microsoft draft suggests that CYCLICAL is discrete (p. 8).

- **CYCLICAL:** A set of values that have cyclical ordering. Day of the week is a good example, since day number one follows day number seven. Attributes with a type flag of CYCLICAL are also considered to be ORDERED and DISCRETE.

However AZIMUTH is CYCLICAL, but **not necessarily discrete** and **stronger** than simple ordering. It can be CONTINUOUS. Again the draft does not permit the simultaneous setting of flags as CONTINUOUS and CYCLICAL, without easing the requirement for cyclical to be discrete. Similarly, temperature or salary require a separate flag CONTINUOUS and NON-CYCLICAL. Algorithms appropriate for CYCLICAL can be inappropriate for NON-CYCLICAL and visa versa.

Summary of suggestions:

1. Add **NOMINAL** to <col_content> and consider adding some more flags (see #6 below).
2. Permit **negated flags**, such as NON-CYCLICAL.
3. Permit **combinations of flags**, such as CONTINUOUS and NON-CYCLICAL.
4. Add flag **COL_REFERENCE <reference ID>** for special column contents data types, such as "letter grade". This flag will be used in addition to common terms presented in the Grammar for <col_content> and <col_content_qual> directly.
5. Develop **APIs**, which will describe contents data types as C++ classes (**OOP approach**). Each API should be an **implementation** for <reference ID> in COL_REFERENCE <reference ID> statement. For more about this OOP approach see [Kovalerchuk B., Vityaev E., [Data Mining in Finance: Advanced in Relational and Hybrid Methods](#), Kluwer, 2000, pp. 164, 169-186].
6. Make terms used in <col_content> and <col_content_qual> **consistent** with terms already used in **measurement theory** [D. Krantz, R.Luce, P. Suppes and A.Tversky, Foundation of measurement, Academic Press, v.1-3, 1971, 1989,1900].

Discussion of Microsoft specification on Data Mining in KDNuggets can be an important input for further development of DM applications.

©

Boris Kovalerchuk, Ph.D.
Dept. of Computer Science, Central Washington University
Ellensburg, WA 98926-7520
ph. (509) 963-1438, fax (509) 963-1449
borisk@tahoma.cwu.edu
<http://www.cwu.edu/~borisk/finance>