Yugoslav Journal of Operations Research
15 (2005), Number 1, 125-145

DATA WAREHOUSING AND DATA MINING - A CASE STUDY

Milija SUKNOVIĆ, Milutin ČUPIĆ, Milan MARTIĆ

Faculty of Organizational Sciences,
University of Belgrade, Belgrade, Serbia and Montenegro
Milijas@fon.bg.ac.yu, Cupic@fon.bg.ac.yu, Milan@fon.bg.ac.yu

Darko KRULJ

Trizon Group, Belgrade, Serbia and Montenegro
KruljD@trizongroup.co.yu

Received: August 2004 / Accepted: February 2005

Abstract: This paper shows design and implementation of data warehouse as well as the use of data mining algorithms for the purpose of knowledge discovery as the basic resource of adequate business decision making process. The project is realized for the needs of Student's Service Department of the Faculty of Organizational Sciences (FOS), University of Belgrade, Serbia and Montenegro. This system represents a good base for analysis and predictions in the following time period for the purpose of quality business decision-making by top management.

Thus, the first part of the paper shows the steps in designing and development of data warehouse of the mentioned business system. The second part of the paper shows the implementation of data mining algorithms for the purpose of deducting rules, patterns and knowledge as a resource for support in the process of decision making.

Keywords: Decision support systems, data mining, data warehouse, MOLAP, regression trees, CART.

1. PREFACE

Permanently decreasing ability to react quickly and efficiently to new market trends is caused by increase in competition on the market. Companies become overcrowded with complicated data and if they are able to transform them into useful information, they will have the advantage of being competitive.

It is familiar that the strategic level of decision-making usually does not use business information on a daily basis but instead, cumulative and derivative data from specific time period. Since the problems being solved in strategic decision-making are mostly non-structural, it is necessary in decision-making process to consider the large amounts of data from elapsed period, so that the quality of decision-making is satisfied. Therefore, Data Warehouse and Data Mining concept are imposed as a good base for business decision-making.

Moreover, the strategic level of business decision-making is usually followed by unstructured problems, which is the reason for data warehouse to become a base for development of tools for business decision-making such as the systems for decision support.

Data warehouse as a modern technological concept, actually has the role to incorporate related data from vital functions of companies in the form that is appropriate for implementation of various analyses.

2. DATA WAREHOUSE IMPLEMENTATION PHASES

Basic data warehouse (DW) implementation phases are [1]:

- Current situation analysis
- Selecting data interesting for analysis, out of existing database
- Filtering and reducing data
- Extracting data into staging database
- Selecting fact table, dimensional tables and appropriate schemes
- Selecting measurements, percentages of aggregations and warehouse methods
- Creating and using the cube

The description and thorough explanation of the mentioned phases is to follow:

2.1. Current situation analysis

Computer system of FOS Student's Service Dept. was implemented at the beginning of nineties but it has been improved several times since then with the aim to adapt it to the up-to-date requests. This system fully satisfies the complex quality requests of OLTP system, but it also shows significant OLAP failures. Data are not adequately prepared for complex report forming. The system uses dBASE V database that cannot provide broad range of possibilities for creating complex reports. dBASE V does not have special tools for creating queries that are defined by the users. Design documentation is the most important in selecting of system information and data used for analysis. All vital information needed for warehouse implementation could often be found out from the design documentation of OLTP system. This phase is the most neglected one by the designers of OLTP system; therefore their solutions do not give possibilities of good data analysis to users.

Since at this phase the possibility of realization and solution of the problem can be seen, it represents a very important phase in warehouse design. Users often know

problems better than the system designers so that their opinion is often crucial for good warehouse implementation.

2.2. Selecting data interesting for analysis, out of existent database

It is truly rare that the entire OLTP database is used for warehouse implementation. More frequent case is choosing the data sub-set which includes all interesting data related to the subject of the analysis. The first step in data filtering is noticing incorrect, wrongly implanted and incomplete data. After such data are located they need to be corrected if possible or eliminated from further analysis.

2.3. Filtering data interesting for analysis, out of existent database

The next step is searching for inappropriately formatted data. If such data exist, they have to be corrected and given the appropriate form. Data analysis does not need all the data but only the ones related to a certain time period, or some specific area. That is why the data reducing practice is often used.

2.4. Extracting data in staging database

After the reducing and filtering of data, data are being extracted in staging database from which the data warehouse is being built (Figure 1). If OLAP database is designed to maintain OLAP solutions, this step can be skipped.

DTS package is written in Data Transformation Services SQL Server 2000. Package writing is very important in DW implementation because packages can be arranged to function automatically so that DW system users can get fresh and prompted data.

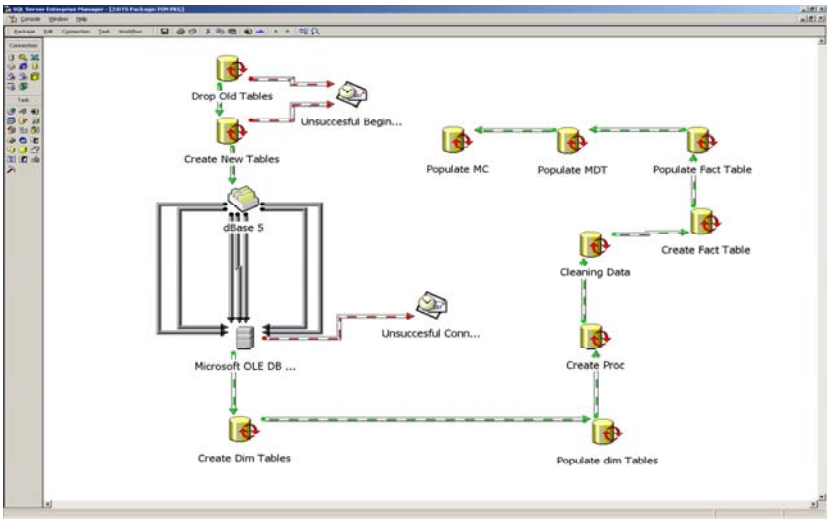


Figure 1: DTS package based on [12]

2.5. Selecting fact table, dimensional tables and appropriate schemas

The entity-relationship data model is commonly used in the design of relational databases, where a database schema consists of a set of entities and the relationships between them. Such a data model is appropriate for on-line transaction processing. A data warehouse, however, requires a concise, subject-oriented schema that facilitates on-line data analysis. Figure 2 shows the schemas that are used in implementation of Data warehouse system.

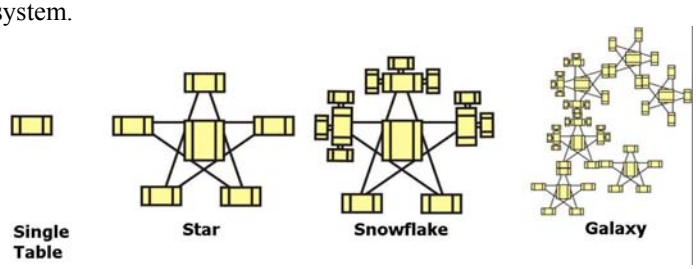


Figure 2: Data warehouse schema based on [20].

The simplest scheme is a single table scheme, which consists of redundant fact table. The most common modeling paradigm according to [10] is star schema, in which the data warehouse contains a large central fact table containing the bulk of data, with no redundancy, and a set of smaller attendant tables (dimension tables), one for each dimension. Snowflake schema is a variant of star schema model, where some dimension tables are normalized, causing thereby further splitting the data into additional tables. Galaxy schema is the most sophisticated one, which contains star and snowflake schemas.

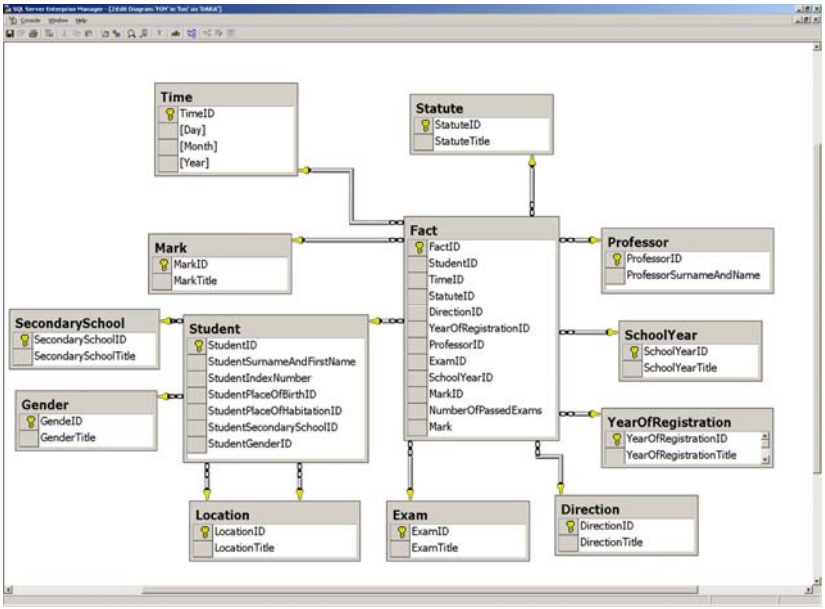


Figure 3: Snowflake scheme from Student's service based on [19]

Only the table that contains the most detailed data should be chosen for the fact table. The most detailed table in this project is the one with students' applications. Tables directly related to it, can be observed as dimensional tables. Because of the complex structure of the data warehouse, snowflake scheme represented at Figure 3 represents the best solution.

2.6. Selecting measurements, percent of aggregations and warehouse modes

The next step in designing data warehouse is selecting measurements. In this case, two measurements can be seen: total number of passed exams and average mark achieved in passed exams.

In the data warehouse implementation very often appears the need for calculated measurements that are attained from various arithmetic operations with other measurements. Furthermore, this system uses the average that has been calculated as the ratio of the total mark achieved on passed exams and the number of passed exams.

Data warehouse solutions use aggregations as already prepared results in user queries and through them they solve the queries very fast. The selection of an optimal percentage of aggregation is not simple for the designer of the OLAP system. The increasing of the percentage of aggregated data speeds up the user-defined queries, but it also increases also the memory space used.

From a Fig. 4 we can conclude that the optimal solution is 75% aggregation, which takes 50 MB of space.

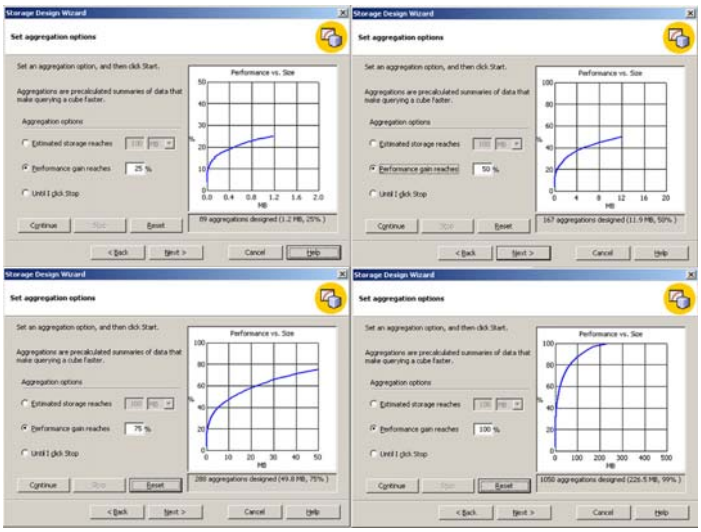


Figure 4: The selection of the optimal percentage of aggregation

The most important factors that can have an impact on the storing mode are:

The size of the OLAP base,
The capacity of the storage facilities and
The frequency of data accessing.

Manners of storing are:
ROLAP (RELATIONAL OLAP),
HOLAP (HYBRID OLAP) and
MOLAP (MULTIDIMENSIONAL OLAP).
wich is shown on fig 5.

ROLAP stores data and aggregation into a relational system and takes at least disc space, but has the worst performances. HOLAP stores the data into a relational system and the aggregations in a multidimensional cube. It takes a little more space then ROLAP does, but it has better performances. MOLAP stores data and aggregations in a multidimensional cube, takes a lot of space, but has the best performances since very complex queries will be used in analysis it is rational to use MOLAP.

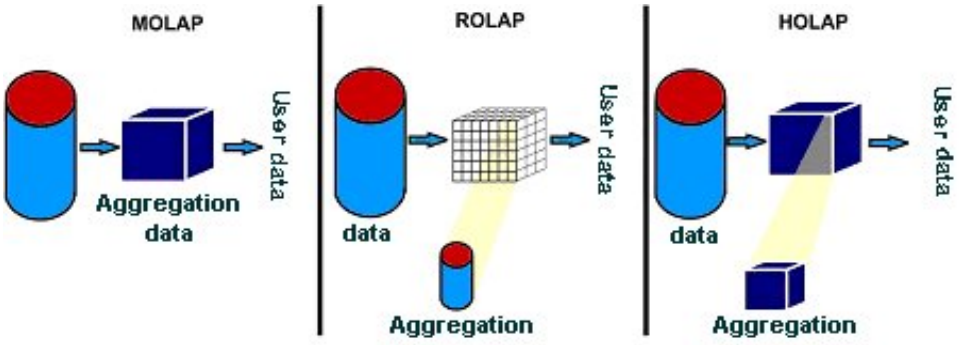


Figure 5: Modes of data storing based on [15].

2.7. Creating and using the cube

The cube is being created on either client or server computer. Fundamental factors that influence the choice of the place for cube's storehouse are: *size of the cube, number of the cube users, performances of the client's and server's computers and throughput of the system.* The created cube can be used by the support of various clients' tools.

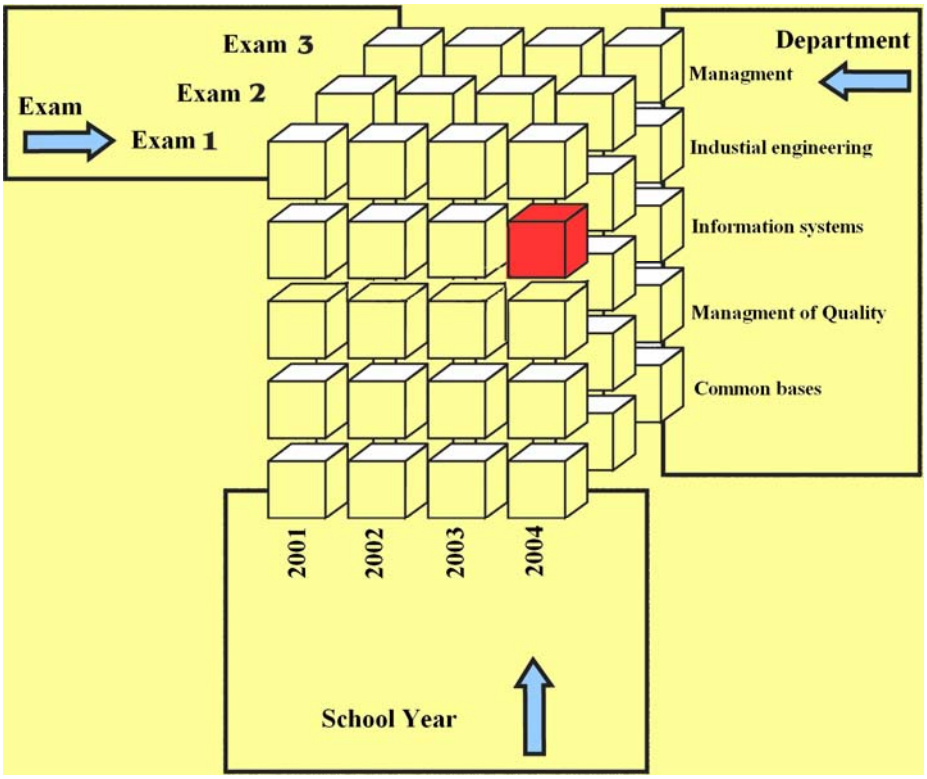


Figure 6: Multidimensional cube

Figure 6 shows a three-dimensional cube that represents average grades by department, Exam and school year. In comparison with OLTP systems that have to calculate the average grade on each request, data warehouse systems have results prepared in advance and stored in multidimensional format.

Figure 7 shows basic screen form, which offers possibility of creating complex reports with the aim of making appropriate prognosis. Application provides creation of various user reports. Functioning of the faculty's management is especially made easier in relation to the analyses of passing exams, e.g. by subjects, examination period, examiners, etc.

The example of the cube usage with MS Excel is shown on Fig. 8, where we can see the average grade and total number of passed exams by professor, Direction and Exam.

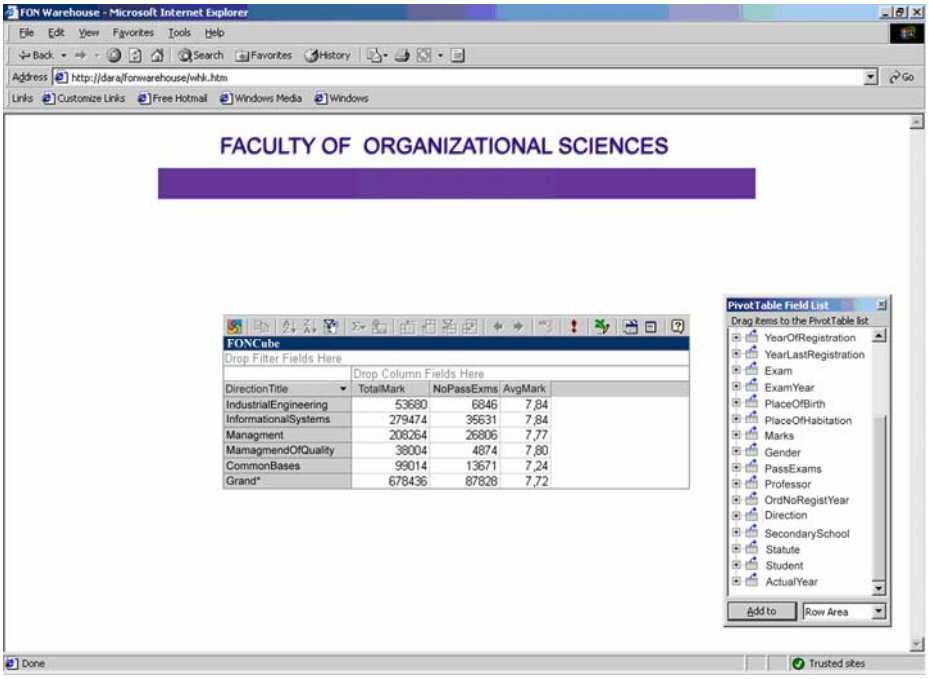
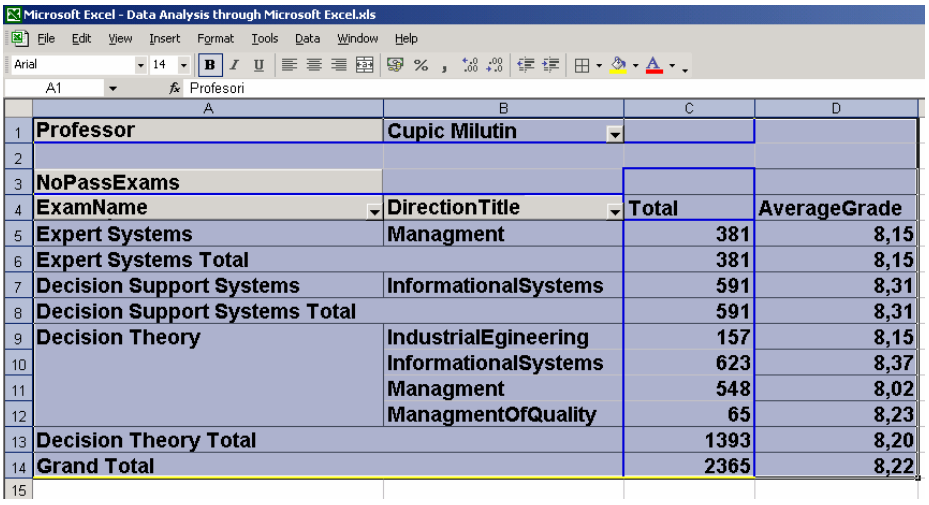


Figure 7: Data analysis through WEB application



	A	B	C	D
1	Professor	Cupic Milutin		
2				
3	NoPassExams			
4	ExamName	DirectionTitle	Total	AverageGrade
5	Expert Systems	Managment	381	8,15
6	Expert Systems Total		381	8,15
7	Decision Support Systems	InformationalSystems	591	8,31
8	Decision Support Systems Total		591	8,31
9	Decision Theory	IndustrialEgineering	157	8,15
10		InformationalSystems	623	8,37
11		Managment	548	8,02
12		ManagmentOfQuality	65	8,23
13	Decision Theory Total		1393	8,20
14	Grand Total		2365	8,22
15				

Figure 8: Data analysis through Microsoft Excel application

3. FROM DATA WAREHOUSE TO DATA MINING

The previous part of the paper elaborates the designing methodology and development of data warehouse on a certain business system. In order to make data warehouse more useful it is necessary to choose adequate data mining algorithms. Those algorithms are described further in the paper for the purpose of describing the procedure of transforming the data into business information i.e. into discovered patterns that improve decision making process.

DM is a set of methods for data analysis, created with the aim to find out specific dependence, relations and rules related to data and making them out in the new, higher-level quality information [2]. As distinguished from the data warehouse, which has unique data approach, DM gives results that show relations and interdependence of data. Mentioned dependences are mostly based on various mathematical and statistic relations [3]. Figure 9 represents the process of knowledge data discovery.

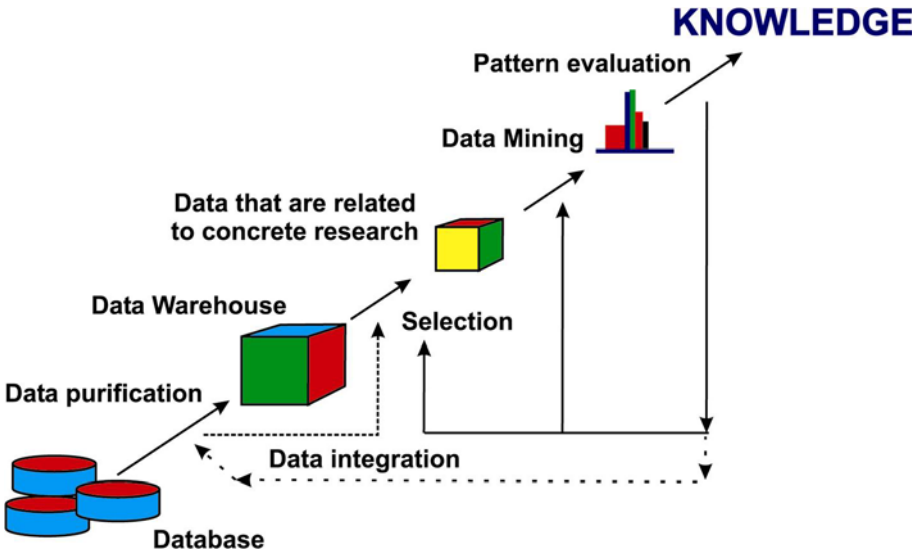


Figure 9: Process of knowledge data discovery based on [10]

Data for concrete research are collected from internal database system of Student's Service Dept., and external bases in the form of various technological documents, decisions, reports, lists, etc. After performed selection of various data for analysis a DM method is applied, leading to the appropriate rules of behavior and appropriate patterns. Knowledge of observed features is presented at the discovered pattern. DM is known in literature as the "extraction of knowledge", "pattern analysis", "data archaeology" [3].

3.1. Regression trees

A regression tree is based on [7] a nonparametric model which looks for the best local prediction, or explanation, of a continuous response through the recursive partitioning of the predictor variables' space. The fitted model is usually displayed in a graph which has the format of a binary decision tree which grows from the root node to the terminal nodes, also called leaves.

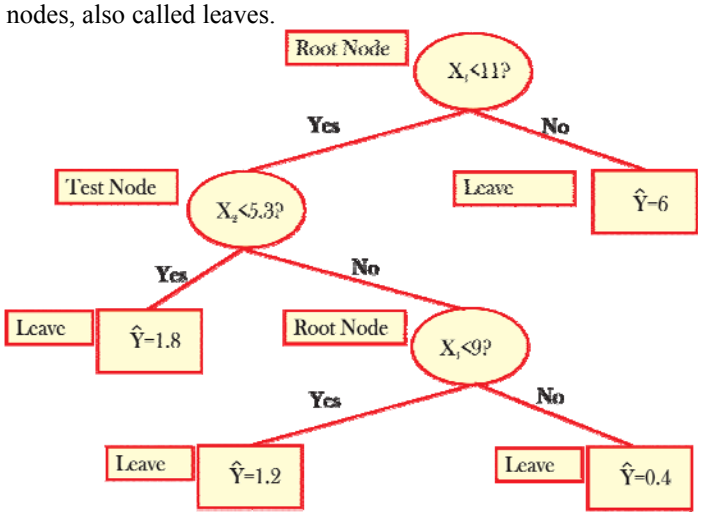


Figure 10: Graphical Display of a Regression tree based on [7]

Figure 10 illustrates the features of a model provided by a regression tree that explains the relationship between a response variable y and a set of two explanatory variables x_1 and x_2. In the example above, the predicted values for y are obtained through a chain of logical statements that split the data into four subsets.

Mathematical Formulation

Let $x_t = (x_{1t}, x_{2t}, ..., x_{mt})'$ be a vector which contains m explanatory variables for a continuous univariate response y_t. The relationship between y_t and x_t follows the regression model:

$$y_t = f(x_t) + \varepsilon_t \tag{3.1.}$$

where the functional form f is unknown and there are no assumptions about the random term ε_t. Following [14], a regression tree model with k leaves is a recursive partitioning model that approximates f by fitting functions in subregions of some domain $D \subset R^m$ so that:

$$\hat{f}(x_t) = \sum_{t=1}^{k} \hat{f}(x_t) I_i(x_t) \tag{3.2.}$$

where $I_j(x_t)$ indicates the membership of t^{th} observation to the j^{th} leave that constitute a subregion of D. The functional form of $\hat{f}_i$ is usually taken to be a constant and, conditionally to the knowledge of the subregions, the relationship between y and x in (3.1.) is approximated by a linear regression on a set of k dummy variables. In [5] discuss, upon evidences, that there is not much gain in choosing $\hat{f}_i$ to be a linear or a more complex function of x_t.

3.1.1. CART Regression Tree Approach

The most important reference in regression tree models is the CART (Classification and Regression Trees) approach in [6], thus the discussion from now on is entirely based on this work. The top-down method of the growing tree implies specifying at first a model associated with the simplest tree as in Figure 11.

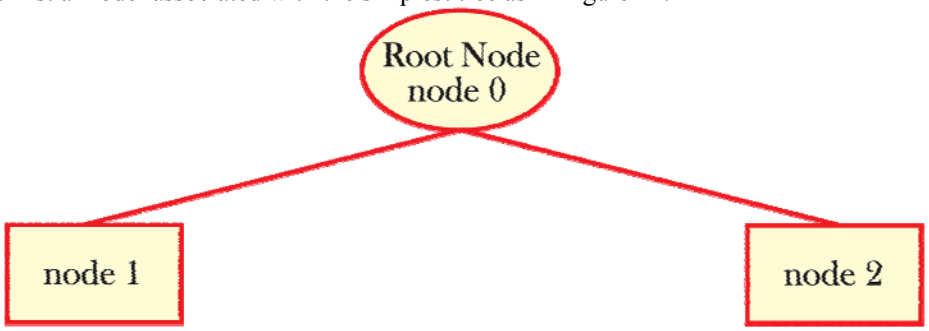


Figure 11: Simplest tree structure based on [7]

To locate the model parameters in the tree structure above, we adopt a labelling scheme which is similar to the one used in [8]. Here, the root node is at position 0 and a parent node at position i generates the left-child node and right-child node at positions $2i+1$ and $2i+2$, respectively. Therefore, a tree model equation for Figure 11, that fits a constant model in each leave, may be written as:

$$y_t = \beta_{01} I_{01}(x_t, w_0, c_0) + \beta_{02} I_{02}(x_t, w_0, c_0) + \varepsilon_t \tag{3.3.}$$

$$I_{01}(.) = \begin{cases} 1, & fw_0' x_t \le c_0 \\ 0, & fw_0' x_t > c_0 \end{cases} \tag{3.4.}$$

$$I_{02}(.) = 1 - I_{01}(.) \tag{3.5.}$$

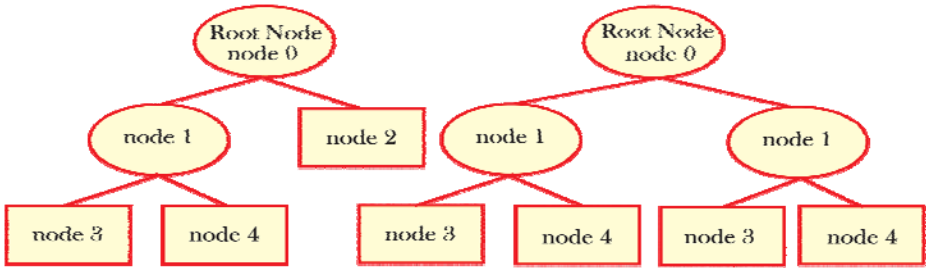
In (3.4.), the hyperplane $w_0' x_t = c_0$ which defines the binary partition of the variable space is obtained from a vector of weights $w_0 = (w_{01}, w_{02}, ...w_{0m})'$, and a scalar threshold c_0.

The constant β_{01} is the local model for the observations that are assigned to the left child node, that is, for those which $I_{01}(.)=1$. Analogous interpretation is given to the parameter β_{02}, hence the model in (3.3) approximates the unknown function f in (3.1.) by a non-smooth and discontinuous function. In the CART context, it is usual to obtain the recursive partitioning by using hyperplanes that are perpendicular to the predictor variables' axis, which means that the vector w_0 is composed of $m-1$ zeros and a unity element in the S_0^{th} position. This simplifies the specification of the model, allowing to perform a less expensive search in the continuous space of hyperplanes. For practical purposes, that means to replace the vector of parameters w_o in (3.3.) by a scalar parameter S_0 (index of the splitting variable) which takes values in the set of integers $S=\{1,2,.....,m\}$ so that:

$$I_{01}(.)=\begin{cases}1, & fx_{S_0t}\le c_0\\ 0, & fx_{S_0t}>c_0\end{cases} \tag{3.6.}$$

$$I_{02}(.)=1-I_{01}(.) \tag{3.7.}$$

By omitting the arguments of the indicator functions, more complex tree structures are represented in the equations that follow Figure 12:



$$y_t=\beta_{02}I_{02}+(\beta_{11}I_{11}+\beta_{12}I_{12})I_{01}+u_t \quad y_t=(\beta_{11}I_{11}+\beta_{12}I_{12})I_{01}+(\beta_{21}I_{21}+\beta_{22}I_{212})I_{02}+u_t$$

Figure 12: Regression trees with 3 and 4 terminal nodes based on [7]

β_{ij} and I_{ij} denotes the constant model respectively and indicator function for the subset of observations located on node at position i that are assigned to the child-node j ($j=1-$ left / $j=2-$ right).

3.1.2. Growing the tree by CART algorithm

The tree growing induction can be seen as an iterative and recursive nonparametric modeling cycle that specifies at each iteration a tree architecture by selecting a node to be split, a splitting variable, and a threshold. Then, local models are

estimated for the observations assigned to the generated nodes. This procedure is repeated recursively in the created nodes until it reaches a point where there is no gain in partitioning the observed data. The final model can be evaluated through cost-complexity measures and re-specified by cutting some branches of the tree. This cycle starts with the specification of the simplest model (Fig 9) by selecting a splitting variable x_{S_0} and a threshold c_0 and estimating the parameters β_{01} and β_{02} .The selection and estimation procedures are carried out simultaneously by searching exhaustively the pair (S_0, c_0) that minimizes the sum of squared errors or the sum of absolute deviations for the model in equation (3.3). The tree obtained in the former case is called LS (Least Squares) regression tree. Conditional on S_0, c_0 it is straightforward to show that the best estimators for β_{01} and β_{02} , in the least square sense, are the sample mean values of the response variable within the children nodes. The tree grown by minimizing of the sum of absolute deviations is called LAD (Least Absolute Deviation) regression tree and it can be shown that the sample median of the response variable within the terminal nodes is the best estimator in this case. More about absolute deviation regression can be found in [16] and [4]. After splitting the root node (node 0), and if we use the least square criterion, the tree model may be re-specified by selecting the tree architecture in the left side of Figure 12.

The recursive partitioning implies that all parameters which are not involved in the current splitting are kept fixed. Thus, in the equation on the left side of Figure 12, the estimates of β_{11} and β_{12} are found by selecting the pair (S_1, c_1) that minimizes the overall sum of squared errors conditionally on S_0, c_0 and β_{02}. The specification/estimation process will continue by maximizing the decrease in the sum of squared errors while adding a terminal node. This procedure naturally forces at each iteration the sum of squared errors to be lowered and thus it becomes necessary to establish a stopping rule in order to verify if a generated node shall be recursively split or shall be declared as terminal, otherwise it will lead to an overfitting. In [6] suggests that a generated node containing a number of observations equal or less than 5 shall be declared as terminal. To reduce the complexity of the tree, the last diagnostic checking can be performed through a pruning technique

We present in Figure 13 and Figure 14 a version of CART algorithm to fit a regression tree model. The tree growing algorithm is shown in Figure 13 while Figure 14 highlights the node splitting procedure. The algorithm executes in the node at the i^{th} position an exhaustive search to find the index S_i of the univariate splitting variable and the threshold c_i that provide the minimum value of a loss function L , usually the sum of squared errors or absolute deviations. A node is declared as terminal if its number of observations is equal or less than 5. According to the adopted labelling scheme, we control the status of the i^{th} node by assigning: 0 (test node) and 1 (terminal node). During the execution of the algorithm, the status of the node can change from "terminal" (1) to "test" (0) and if a node position has to be skipped, these nodes are labelled with -1 .

```
1 d=0, endtree=0
2 Node(0)=1,Node(1)=0,Node(2)=0
3 while endtree < 1
4       if
Node(2^d − 1)+Node(2^d)+...+Node(2^{d+1} − 2)=2 − 2^{d+1}
5               endtree=1
6       else
7               do i= 2^d − 1, 2^d ,..., 2^{d+1} − 2
8                       if Node(i) > -1
9                               Split tree
10                      else
11                              Node(2i+1)=−1
12                              Node(2i+2) =−1
13                      end if
14              end do
15      end if
16 d=d+1
17 end while
```

Figure 13: General form of CART algorithm based on [7]

```
Split tree
1 do  s_i =1,2,...,m
2       do c_i = x_{s_i,(1)} ,..., x_{s_i,(99)} (percentiles of x_{s_i})
3 Estimate β_{i1}, β_{i2}
4 Evaluate loss function  L(β_{i1}, β_{i2}, c_i, s_i)
5       End do
6 End do
7 (ŝ_i, ĉ_i, β̂_{i1}, β̂_{i2}) =argmin  L(β_{i1}, β_{i2}, c_i, s_i)
8 Node(i)=0
9  Split  Node (i) and create Node(2i+1) , Node(2i+2)
10 do j=1,2
11      if (# observations ∈ Node (2i + j) ) > 5
12              Node(2i + j) = 1
13              Node(2i + j) = 1
14      Else
15              Node(2i + j) = -1
16              Node(2i + j) = -1
17      End if
18End do
```

Figure 14: CART splitting algorithm based on [7]

3.2. Illustrated examples of DM

More than one hundred cluster models and decision tree models have been implemented in this project. For the sake of being practical this project will present only some representative examples. In implementation we used Microsoft Decision Trees (MDT) which is based on CART algorithm [17] described in the paragraph 3.1.

The first example is *evaluation of student's mark* based on the following criteria: exams, departments, and number of attempts and gender of students. MDT, which is represented at Figure 15, chooses number of attempts as the most significant attribute for the first level. After that the algorithm chooses department and gender as the next attributes by importance. At the end, there are evaluations of expected marks depending on the values from analyzed attributes.

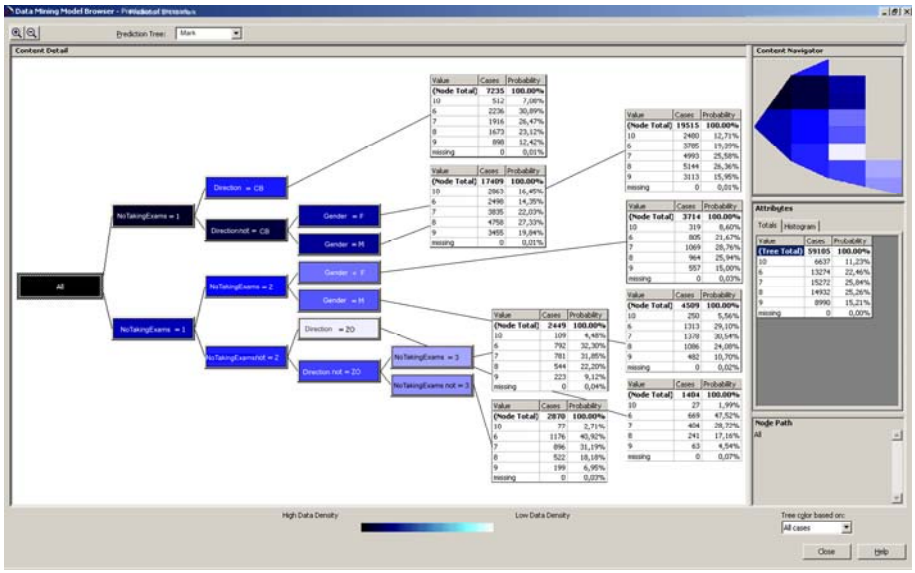


Figure 15: MDT for evaluation of marks

Based on the decision tree the following can be concluded: students that take exam in less number of attempts get better marks and students from the first and the second year of studies (Common Bases Department CB) get worse marks than students from the third and the forth year.

Figure 16 represents correlation between criteria for evaluation of attained mark. By moving cursor to the left-hand side of the Figures, the intensity of correlation between observed criterions is displayed. Therefore it is easily perceived that the obtained mark depends on student's gender the least, a bit more on student's department, the most on the number of attempts.

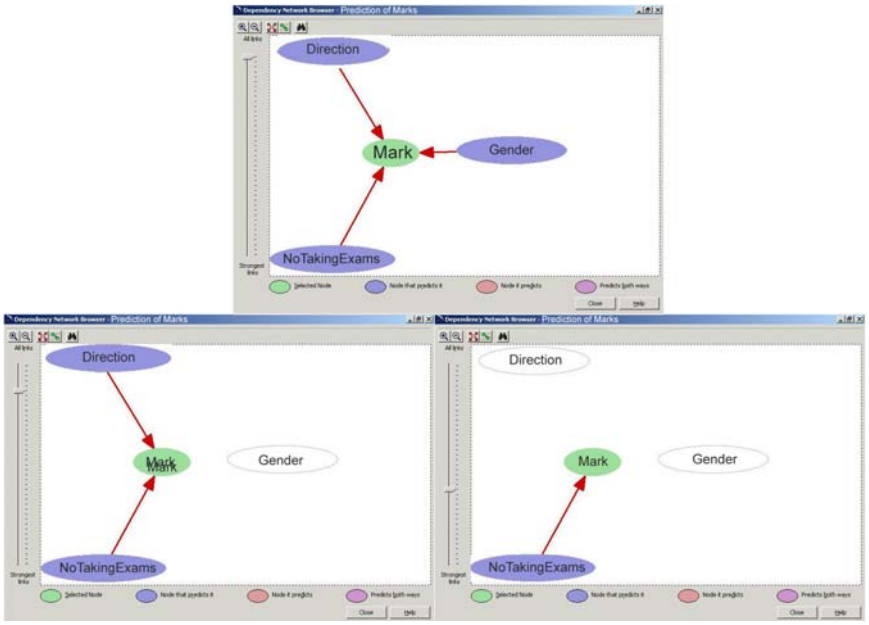


Figure 16: Analysis of correlation between criterions for evaluation of attained mark

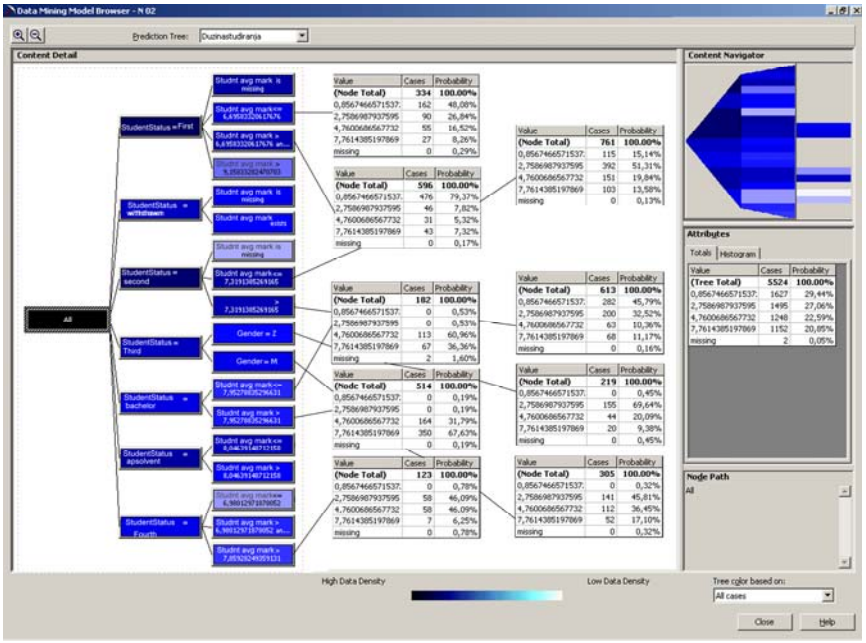


Figure 17: MDT for evaluation of length of studying period based on [13]

The second example represents MDT for evaluating the length of studies period. (Figure 17).

Relevant criterions for decision-making are: firstly, the year of studies observed, then secondly, observed average student's mark and then for one category, student's department. Total population, which is observed for this analysis, is 5524 students, (Figure 17).

Table 1: Interval's limits of time of studying [13]

Interval	I	II	III	IV	V
Span (Year)	till 0,856	0,86-2,758	2,76-4,760	4,761-7,761	Noise
Number of students	1627	1495	1248	1152	2
%	29,45	27,06	22,59	20,858	0,04

Accordingly, out of the total number of observed students, 1672 students or 29.45% belong to *the first interval,* 1495 students or 27.06% belong to *the second*, 1248 students or 22.59% belong to *the third*, 1152 students or 20.858% belong to *the forth* and 2 students and 0.04% belong to *the fifth interval*. Two students that have wrongly inscribed data, belong to the fifth interval.

For example, if it is necessary to make evaluation of the length of studies of a second year student, whose average mark is over 7.70, it will be claimed accordingly that his length of studies will be 2.758 years, with the possibility of 0.2592 or 25.92%.

Figure 17 shows correlation between criterions of evaluation of length of studies. Moving cursor to the left-hand side of the Figures, we come to the intensity of correlation of observed criteria. It is easily perceived that the length of studies depends on student's department and gender the least, while it depends more on average mark and current student's status.

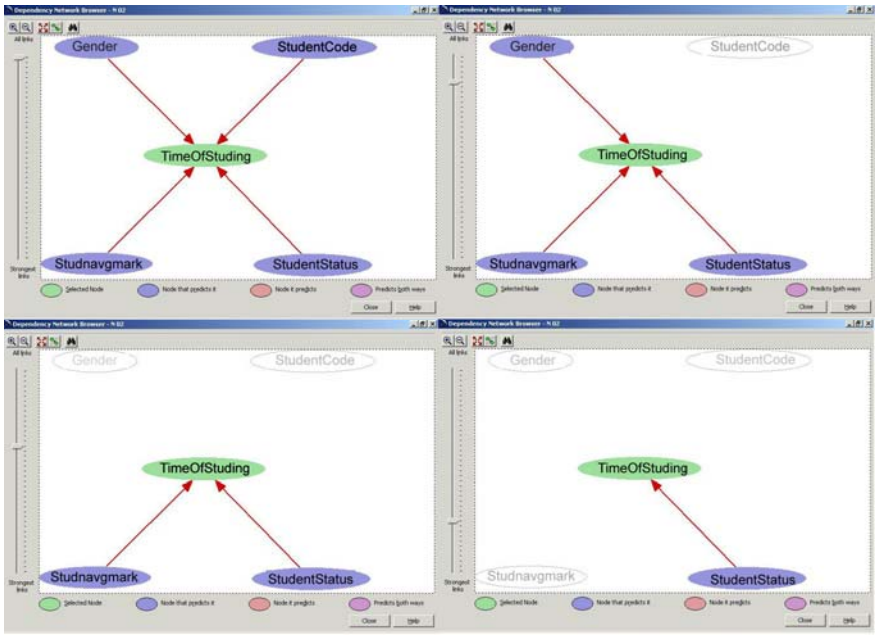


Figure 18: Analysis of dependence between criterions according to evaluation of time of studying based on [18]

This example shows clustering by criterion of average mark for those students whose data are in data warehouse.

Table 2: Average student's mark based on [18]

Student type	Cluster No.	Avg Mark	Number of student	Min. Avg	Max Avg
The worst	1	Without pass exams	1176	------	------
Poor	2	6.54	1760	6.000	7.093
Average	3	7.32	1140	7.095	7.556
Very good	4	7.77	1080	7.558	8.000
The best	5	8.89	1121	8.023	10.000

From the Table 2 we can see that 1176 students belong to the first category, without passed exams, and at the same time without any possibility to have the average mark calculated. Students with average mark 6.54 or within the interval from 6.00 to 7.093 belong to the second category, while the total number is 1760. 1140 students with 7.32 average mark or within the interval of average mark from 7.095 to 7.556 belong to the third category. 1080 students with 7.77 average mark or within the interval of average from 7.558 to 8.000 belong to the fourth category. The last one, the fifth category comprises students with the best average mark within the interval from 8.023 to 10.00, while the total number of them is 1121.

The second example is clustering students by criterion length of studies.

Table 3: The results of clustering by time of studying for bachelor students based on [18]

length of studies	Cluster No.	Count of student	Avg length of studing (Year)	Min time of studing (Year)	Max time of studing (Year)	Percent
Short	1	84	4.67	4.03	5.30	25.15%
average	2	83	5.69	5.31	6.09	24.85%
long	3	84	6.44	6.091	6.79	25.15%
very long	4	83	8.17	6.80	9.55	24.85%

The result of this clustering comprises four categories. The first category consists of the students with the shortest length of studies. The average is 4.67 years, while the total number of those students, from the interval of average length of studies from 4.03 to 5.30 years, is 84 or 25.15%. The second category comprises students with average length of studies of 5.69 years, within the interval from 5.31 to 6.09 years, while the total number of them is 83 or 24.85%, etc. It has to be noted that the observed data refer to bachelor students during the period of not more than 10 years, so the interval of the last category is from 6.80 years to, approximately 10 years.

4. IMPLEMENTATION

In implementation of DW and DM we have used MS SQL Server 2000, DTS services SQL Server's 2000 and OLAP Services 8.0. Excel 2000, ProClarity and Web application. On the Figure 18 is shown schema of data warehouse and data mining solution for student data service.

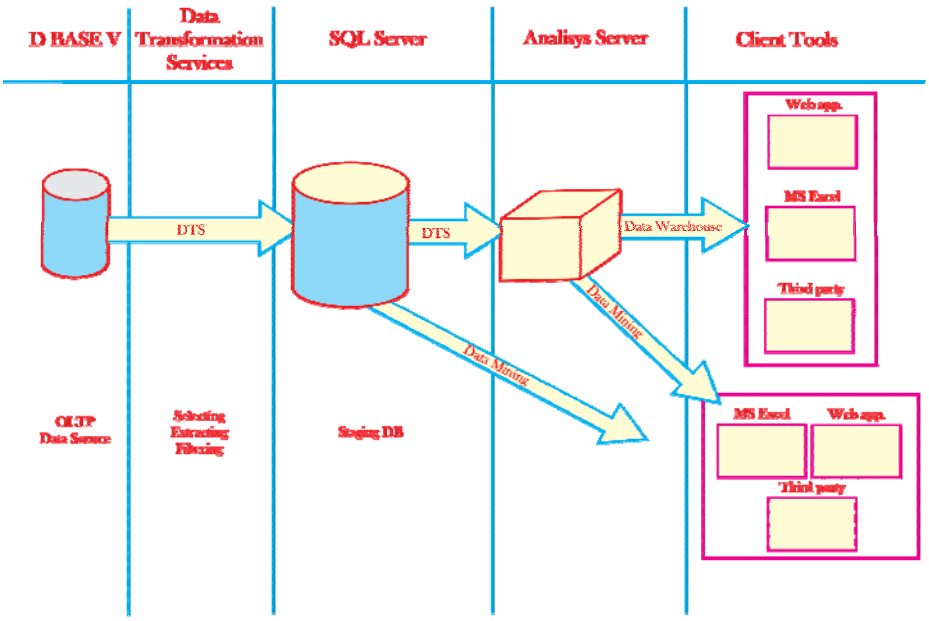


Figure 18: Scheme of data warehouse and data mining solution based on [11]

In conclusion data warehouse is a very flexible solution fitting end user purposes, which by tools in everyday usage (e.g. Microsoft Excel) can explore database more efficiently than any other tool from OLTP environment. Significant advantage of this approach to knowledge and information research in databases is that the user does not have to possess knowledge of relational model and complex query languages.

It is obvious that MDT algorithms give remarkable results for analysis with the aim of good prediction.

MDT is based on possibility of various attributes and it is used when prediction of cases is necessary. MDT algorithm also generates rules. Every tie in the tree can be expressed as a set of rules that describes it, as ties that led to it. Besides, these clusters are based on statistic of various attributes. They provide for the creation of the model that is not used for predictions but a very efficient one in finding records that have common attributes so that they can be put in the same group.

MDT enable the user to analyze a great number of various DM problems, with the aim of timely prediction. Using good elements, which follow the prediction, it is possible to make good business plans and lead business system to the benchmark. It can be said with pleasure that this method of analysis of data and making-decision process becomes more and more popular in solving new problems.

5. COMPARATIVE ANALYSIS

The following table contains comparative features of existing and new solution

Table 4: Comparative features of existing and new solution based on [11]

	Existing system	New system
Memory Space	100 MB	150 MB
Avg. query time	3 sec	< 0.5 sec
User defined queries	NO	YES
Detection of faulty data	NO	YES
Sophisticated analysis	NO	YES
Used for	Data processing and data analysis	Data analysis
Data security	NO	YES
Data granularity	Detailed data	Aggregated data
Dependency analysis	NO	YES
Complex statistical analysis	NO	YES
Current technique	Relational databases	Data Warehouse and Data Mining

Referring to table 4 we could conclude that the features of the new system in the sense of data analysis are much better, the only disadvantage is that it takes up more Memory Space for storing aggregated data and that it cannot be used for data analysis

6. CONCLUSION

This paper shows the phases through which a DW and DM solution is formed. Based on the demonstration we can conclude that DW offers a flexible solution to the user, who can use tools, like Excel, with user-defined queries to explore the database more efficiently in comparison to all other tools from the OLTP environment.

The significant benefit from this solution of information and knowledge retrieval in databases is that the user does not need to possess knowledge concerning the relational model and the complex query languages.

This approach in data analysis becomes more and more popular because it enables OLTP systems to get optimized for their purpose and to transfer data analysis to OLAP systems.

REFERENCES

[1] Barry, D., *Data Warehouse from Architecture to Implementation*, Addison-Wesley, 1997.
[2] Berry, M.J.A., and Linoff, G., "Mastering data mining", *The Art and Science of Customer Relationship Management*, 1999.
[3] Bhavani, T., *Data Mining: Technologies, Techniques, Tools and Trends*, 1999.
[4] Birkes, D., and Dodge, Y., *Alternative Methods of Regression.* John Wiley & Sons, 1993.
[5] Breiman, L., and Meisel, W.S., "General estimates of the intrinsic variability of data in nonlinear regression models", *Journal of the American Statistical Association*, 71 (1976) 301-307.

[6] Breiman, L.J.H., Friedman, R.A.O., and Stone, C.J., *Classification and Regression Trees*, Belmont Wadsworth Int. Group, 1984.
[7] De Rosa, J.C., Viega, A., and Medeiros, M.C., *Tree-Structured Smooth Transition Regression Models Based on CART Algorithm*, Department of Economics, Janeiro, 2003.
[8] Denison, T., Mallick, B.K., and Smith, A.F.M., "A Bayesian CART algorithm", *Biometrika* 85 (1998) 363-377.
[9] Gunderloy, M., and Sneath, T., *SQL SERVER Developer's Guide to OLAP with Analysis Services*, Sybex, 2001.
[10] Jiwei, H., and Micheline, K., *Data Mining: Concepts and Techniques*, Simon Fraser University, 2001.
[11] Krulj, D., "Design and implementation of data warehouse systems", M Sc. Thesis, Faculty of Organizational Sciences, Belgrade, 2003.
[12] Krulj, D., Suknović, M., Čupić, M., Martić, M., and Vujnović, T., "Design and development of OLAP system FOS student service", *INFOFEST*, Budva, 2002.
[13] Krulj, D., Vujnović, T., Suknović, M., Čupić, M., and Martić, M., "Algorithm of Data Mining, good base for decision making", *SYM-OP-IS*, Tara, 2002.
[14] Lewis, P.A.W., and Stevens, J.G., "Nonlinear modeling of time series using multivariate adaptive regression splines (MARS)", *Journal of the American Statistical Association*, 86 (1991) 864-877.
[15] Lory, O., and Crandall, M., *Programmers Guide for Microsoft SQL Server 2000*, Microsoft Press, 2001.
[16] Narula, S.C., and Wellington, J.F., "The minimum sum of absolute errors regression: A state of the art survey", *Internat. Statist. Rev*., 50 (1982) 317-326.
[17] Seidman, C., *Data Mining with Microsoft SQL Server 2000*, Microsoft Press, 2001.
[18] Suknović, M., Čupić, M., Martić, M., and Krulj, D., "Design and development of FOS Data Warehouse", *SYM-OP-IS*, Tara, 2002.
[19] Suknović, M., Krulj, D., Čupić, M., and Martić, M., "Design and development of FOS Data Warehouse", *SYMORG*, Zlatibor, 2002.
[20] Vidette, P., *Building a Data Warehouse for Decision Support*, Prentice Hall, 1996.