Iterative, Interactive and Intuitive Analytical Data Mining

Petr Chmelar and Lukas Stryka

Faculty of Information Technology, Brno University of Technology

Abstract. In this paper, we propose a framework, internally called OLAM SE (Self Explaining On-Line Analytical Mining), for iterative, interactive, and intuitive mining of multilevel association, characterization and classification rules. This framework is proposed as an extension of OLAP or an alternative to Han's OLAM. OLAM processes high structured data sets. The structure is based on given conceptual hierarchy (expressed, for example, in XML). OLAM SE determines minimum support value from user defined cover of data with usage of entropy coding principle. We also determine the maximum threshold to avoid explaining knowledge that is obvious. The presentation of results is realized similarl to the UML notation. In fact, it is a visual graph which nodes are frequent concepts sets represented as packages including sub-concepts – data classes or items. Edges represent patterns between packages and items. These patterns can be interactively explored by the user. Other possibly interesting sets are intuitively offered to the user. This is well suitable for the characterization and not-naïve Bayessian classification.

1 Introduction

There are huge amounts of data stored in databases. Thus, it is very difficult to make decisions based on this data. Decision support problems have been motivating a development of sophisticated tools which provide a new view on data for better data understanding. These tools are used for business analysis, scientific research, medical research and many other areas. These tools can be based on data mining techniques, OLAP (On-Line Analytical Processing), data warehouses, etc.

There are many algorithms and methods for data mining on transactional and relational data [3][5][12]. But following the requirements of science or commercial sphere the expansion of storing structured or semi-structured data has been coming up. Thus it's necessary to developed new methods or techniques for data mining on this kind of data. The data mining on data with high complexity is mostly very time-consuming. So there are some approaches to save the computing time by a reduction of scanned state space.

Our approach uses some principles from theory of information and user interactivity to determine data sets that are interesting for the data mining analysis. The main asset is to provide fast and interactive system for data mining on structured or semi-structured data. The method is proposed to process data with given conceptual hierarchy.

1.1 Organization of the Work

The following chapter is dedicated to the Han's OLAM principles based on OLAP and data mining techniques. Chapter 3 describes our contribution. There are explained principles of major improvements, parameters and algorithms as well as the knowledge presentation and the OLAM SE architecture. The work is concluded in chapter 4.

2 OLAP + Data Mining = OLAM

On-line analytical mining (OLAM) [6], also called OLAP mining, integrates OLAP with data mining. OLAM takes advantage of data warehouses such as high quality of data stored in data warehouses. Thus OLAM works on integrated, consistent, and cleaned data so there are no necessary preprocessing steps. OLAM provides facilities for data mining on different subsets of data and at different levels of abstraction, by drilling, pivoting, filtering, dicing, and slicing on a data cube and on some intermediate data mining results.

2.1 OLAP & Data Warehouses

Data warehouses and data marts are used in a wide range of applications. There are three kinds of data warehouse applications: information processing, analytical processing, and data mining. Information processing supports querying, basic statistical analysis, and reporting using crosstabs, tables, charts or graphs. Analytical processing supports basic OLAP operations, including slice-and-dice, drill-down, roll-up, and pivoting. It works in summarized and detailed forms. Data mining supports knowledge discovery by finding hidden patterns and associations and presenting the results using visualization tools.

According to OLAP we can find similarities in the basic operations of our method. First basic operations are pivot and drill. In our methods are these operation represented by rolling and unrolling packages in result graphical representation. We can identify the slice and dice operations in our method as well. These operations are represented by dragging and dropping of packages to and from litter bin.

2.2 Data Mining

A generally accepted definition of data mining and knowledge discovery is given by Fayyad et al. (1996) as the non-trivial process of identifying valid, novel, potentially useful and ultimately understandable patterns in data. Data mining is not a straightforward analysis nor does it necessarily equate with machine learning.

2.3 Data Characterization & Classification

Data can be associated with classes or concepts. It can be useful to describe individual classes and concepts in summarized, concise, and yet precise terms. This description can be derived via data characterisation. Data characterisation is an aggregation of the general characteristics and features of a class under study.

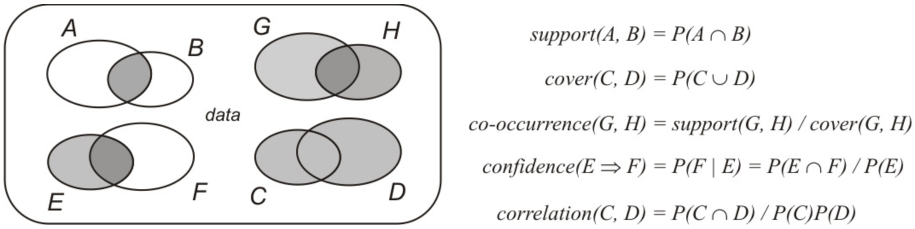
Classification predicts categorical class labels and classifies data based on the model which is constructed from the training set. The classification doesn't create such labels but examines properties (often called observations) of each class in the learning phase. In the classification step the classification algorithm compares observations of an unknown object and tries to decise which class is the most suitable for those data.

2.4 Frequent Patterns & Multilevel Association Analysis

Frequent patterns are patterns that appear in a data frequently. These patterns could be itemsets, subsequences, or substructures. Frequent itemset could be milk and bread in shopping basket analysis, where milk and bread appears frequently together in a transaction data set. To determine which patterns are frequent, the minimum support metrics has to be satisfied. Ilustration of support and other metrics we use is in the Figure 1.

For many mining tasks, it is difficult to find strong patterns in data at low or primitive levels of abstraction. Strong associations discovered at high levels of abstraction may represent common sense knowledge. Sometimes common sense knowledge for one user may be novel for another. Therefore, methods providing capabilities for mining association rules at multiple levels of abstraction with sufficient flexibility for easy traversal among different abstraction spaces has been developed.

These methods use a conceptual hierarchy defining a sequence of mappings from a set of low-level concepts (also called classes) to higher-lever, more general concepts. Conceptual hierarchy is represented by rooted tree, where nodes are general item sets, leafs are data items and the root represents most generalized abstraction of all data items. So these data can be generalized by replacing low-level concepts by their higher-level concepts in a concept hierarchy.



$$support(A, B) = P(A \cap B)$$

$$cover(C, D) = P(C \cup D)$$

$$co\text{-}occurrence(G, H) = support(G, H) / cover(G, H)$$

$$confidence(E \Rightarrow F) = P(F \mid E) = P(E \cap F) / P(E)$$

$$correlation(C, D) = P(C \cap D) / P(C)P(D)$$

Fig. 1. Demonstration of usefulness metrics

2.5 OLAM

On-Line Analitical Mining (also called OLAP mining)[13] integrates OLAP with data mining. OLAM takes advantage of data warehouses such as high quality of data stored in data warehouses. Thus OLAM works on integrated, consistent, and cleaned data so there are no necessary preprocessing steps. OLAM provides facilities for data mining on different subsets of data and at different levels of abstraction, by OLAP operations on a data cube and on some intermediate data mining results.

OLAP is a data summarization/aggregation tool that helps simplify data analysis, while data mining allows the automated discovery of implicit patterns and interesting knowledge hidden in large amounts of data. The main difference between OLAP and data mining is that OLAP is based on interactive user-defined hypothesis testing while data mining is relatively slow generation of such hypotheses.

OLAM represents methods which integrate OLAP principles and data mining methods for multi-dimensional data mining in large databases and data warehouses on different granularities (different subsets or different levels of abstraction by drilling, pivoting, filtering, dicing and slicing on a data cube). An OLAM engine performs analytical mining in data cubes in a similar manner as an OLAP engine performs on-line analytical processing. This engine accepts user's on-line queries and work with the data cube in the analysis. So the engine may perform multiple data mining tasks, such as concept description, association, classification, prediction, clustering, etc. OLAM uses more powerful data cube construction than OLAP because OLAM analysis often involves the analysis of large number of dimensions with finer granularities. Construction of data cube is following: if data cube contains a small number of dimensions, or if it generalized to high level, the cube is constructed as compressed sparse array but is still stored in a relational database to reduce costs of construction and indexing of different data structures.

3 OLAM SE Concepts

We propose OLAM SE system (Self Explaining On-Line Analytical Mining) that is in some aspects similar to the Han's OLAM system. We have taken the idea of OLAP interactivity and applied it to several data mining functions that are adapted to be iterative and interactive to user. Also mining preprocessed data - cleaned and aggregated in data warehouses is common to the OLAM as well as the use of conceptual hierarchy.

The common goal of the OLAM SE is to simplify data mining to the professionals who don't have attitude of mind in data mining but who understand their data and want more significant, interesting and useful information. The proposed simplification can be done by shielding internal concepts (association, classification, characterization) and thresholds (support, confidence) from user.

There are four main improved areas compared to data mining, OLAP and Han's OLAM: (1) Simplified usefulness metrics - *cover* and *obviosity.* (2) Improvement of interactivity and mining performance. (3) Intuitive graphical interface based on UML diagrams. (4) Intuitive suggestion of relevant items for classification and association as the main novelty proposed in this paper.

3.1 Simplified Thresholds

Standard data mining techniques are too complicated for beginners – they have to learn the concept, somehow guess threshold values and wait comparatively long time for irrelevant results – due to the wrong threshold specification and expectation. In case of multilevel conceptual hierarchy the users' confusion might go further – see example 3.1 in [1] where the author supposes *support* "4" for level 1 and "3" for level 2 and 3 which is mystifying not only for a common user, even experts have to guess the proper values.

The very first idea of the simplification was based on the Paretto analysis. It shows that usually 80% of consequences stem from 20% of the cause. For instance the 20% of components causes 80% of malfunctions [14]. So we identified the first problem of inuitive data mining – to find the significant factors as simple as possible. This has led to the *cover* parameter.

The second problem of useful data mining is mining unnecessary, obvious information (eg. Sports Shop sells Sporting Goods). That's the reason we employed the *obviosity* metric.

Technically, those interesting and significant data are derived from frequent itemsets with its' support values. That is quite usual in (multi-level) association rule mining - it is supposed that the most significant data can be described by frequent itemsets [3]. Description of the rest data can be done using OLAP or naïve Bayessian classification.

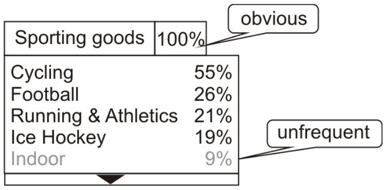


Fig. 2. Illustration of frequent (Cycling, Football, . . .) and obvious concepts

Cover We have introduced the new parameter called *cover*. It means a percentage cover of examined data. It is based on the Paretto analysis – minor part of concepts covers major part of data. In other words, the *cover* parameter determines which classes are significant and essential to *cover* required data.

We have used ideas of information theory. Especially it is the coding where the entropy is used for lossless data compression – our algorithm works similarly (but reversely) to the Huffman coding technique. The Huffman algorithm (building the coding tree) merges the least significant values – data sets with lowest probability of occurrence, which we understand as support. Our Inverse Huffman algorithm is merging the most significant values and the least significant doesn't take into consideration at all. So we introduced and employed an easy lossy compression of analyzed data.

The technique works with data concepts in hierarchies if available. It can be illustrated as merging most frequent classes on each concept level. The *covered* part is a set of most frequent classes that has higher or equal support than the specified or somehow determined (see the Interactivity Improvement chapter) *cover* value (all together). We will depict the algorithm in the following example rather than formalizing the problem.

Example 1 - support from cover.

Suppose that we want analyze shopping data of the Sports Shop oriented on the cycling. The initial setup of the *cover* threshold is 80% of investigated data (by default). The *obviosity* level is set to 85% (see below). The illustration of the algorithm is on the Figure 3.

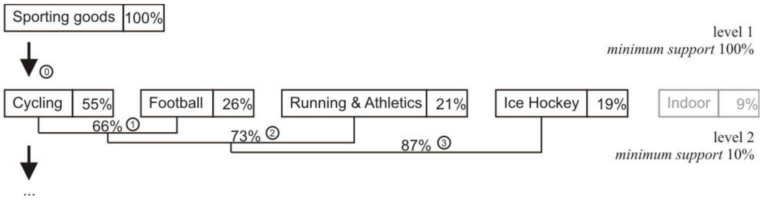


Fig. 3. Example

There is only one class at the top-most level of concept hierarchy. The determined *minimum support* is 100%, but the presence of concept Sporting Goods is obvious in the context of Sports Shop so we won't consider the top-most level at all.

The second concept level contains seven classes but it is necessary to merge only four of them to satisfy the default *cover* value – 80%. It is done by iterative merging of most frequent concepts. For instance the *support* of Cycling is 55% and together with Football equipment the *cover* is 66% and so on. Finally including Ice Hockey goods it is 87%. So there are four scans over the database as shown in 3in small circles.

It is not necessary to investigate the fifths (Indoor) and other classes but there should be embedded a politic to determine the *minimum support* value. In the second level the *minimum support* may be any value between 9% and 19%. The value 19% is a simple solution, but for further mining of more interesting rules it should be less. We prefer the algorithm sets *minimum support* a bit more than the first infrequent class – 10% in this case. So there can be one more database scan.

The algorithm continues processing the next level in the same manner.

Obviosity The *obviosity* parameter determines a maximum support of a useful concept. This parameter was established to eliminate generating of obvious knowledge – the information gain of this knowledge is low. Obvious knowledge is a concept with high frequency of occurrence so this knowledge can be presumed without any data mining process – using a common sense.

Obvious concepts are not processed by the data mining except leaf classes at the lowest level of hierarchy. If there is a level in a concept hierarchy containing only obvious classes (eg. Customers buy Products and Solutions), whole level is removed and is moved to the trash, as described in the next chapter.

3.2 Mining rules and On-Line Presentation

One of the most important points of knowledge discovery in databases is the knowledge presentation. The presentation layer of OLAM SE is much more closed to processing than in other systems. Therefore we present it together in this chapter.

The OLAM SE presentation layer is similar to the UML (Unified Modeling Language) Structure diagrams [10] that are commonly used in application development and business modelling so even economists may be familiar with it. We need description of aggregated data in its concept hierarchies which is somehow different to the Class diagram. It is a graph. Each node represents a Concept as described in the chapter 2.4. In addition it contains the *support* value on right of a name of class (concept) and sub-concepts. Its example is in Figure 2.

Edges represent relations. There are three types of relations in OLAM SE:

Relationship is an undirected line. Technically it is a relation among frequent concepts that merges two concepts. Relations haven't names but are marked with numbers (per cent) that mean the support or other metrics of interrelated concepts (See Figure 1). Also the temporal evaluation of the relations is possible – it describes trends of support, confidence, cover and correlation metrics.

Association is a special kind of relationship, it is a directed line and the *confidence* value(s) are situated near appropriate side of the association.

Aggregation is a special and important relation, it is an undirected link terminated by a diamond. It represents a concept hierarchy or an attribute relation. It is very important relation in OLAM SE because each item of the concept (subconcept in conceptual hierarchy) can be easily extracted by a mouse – using drag and drop.

The expansion of an attribute concept to a new concept on the desktop is similar to the drill-down in OLAP. Hiding an attribute to its super-concept is equivalent to the OLAP roll-up operation. The class can be dropped anywhere on the desktop but relations are built only within the on-line workspace. See Figure 4 for illustration.

Workspaces The application consists of at least two workspaces – on-line and off-line. On the Figure 4, the top most window represents the off-line one. It contains classes or itemsets that are not currently under on-line investigation.

Although it provides more data mining functions, we will think of the window as a storage space for data that are temporarily unused.

The off-line workspace is necessary for two reasons. First it is the ability of slice/dice operation on one or more concept hierarchies (OLAP dimensions) simply by dragging concepts (subconcepts) from on-line to off-line workspace similarly to OLAP. The second reason is the on-line computational efficiency.

On the Figure 4 the lower window is an on-line workspace. It works similarly to Han's OLAM [6] but it's simplified. OLAM SE uses Apriori algorithm [3]. There are investigated frequent itemsets that correspond to the frequent concept sets. First step of Apriori (generation of frequented itemsesets) is done using Inverse Huffman algorithem.

The second cycle of Apriori - discovery of frequented 2-itemsets is important. These results are displayed using relationships. The support is counted and edges are evaluated. It means maximally $\frac{N(N-1)}{2}$ database scans for N itemsets in on-line workspace (each other). Larger itemsets the OLAM SE computes iteratively after that it displays first results.

OLAM SE investigates multilevel data – different *minimum support* values derived from *cover*using Inverse Huffman should be considered between different concept's levels. We propose using the lower *minimum support* value, belonging to the deeper level where more levels are investigated together.

If the threshold is not reached, no association within classes is displayed – this can be forced and a dashed line appears and is evaluated. Note that the support is not counted among sets and its' supersets or subclasses (aggregation). It is similar to the *obviosity* metric. Technical problem is the lay-out of rules therefore an intelligent algorithm should be invented.

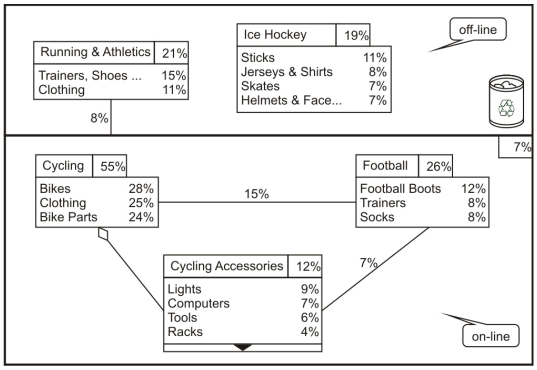


Fig. 4. Work space

In such way we get (connected) Concept diagram as shown in Figure 4. After the OLAM SE display frequented 2-concepts *Apriori* goes further if all relations on the on-line workspace are frequent (together) and it shows the overall support value in right corner. Mined rules are displayed classically in a separate window.

The *minimum confidence* is set to 50% by default to display rules better than a chance. But it can be counted using user defined count of desired rules.

If the overall support value is below the minimum support value, only some association rules are generated. The workspace is investigated by the OLAM SE system to find the weakest relation - a concept that is connected by weakest relation. The OLAM SE tries to suggest removing such concepts – association is marked dashed, to obtain frequent itemset out of all remaining sets on the on-line workspace intuitively (off-line) as described in the following chapter.

3.3 Off-line Suggestion - Classification and Characterization

In contrast to the relatively fast on-line hypothesis testing techniques (OLAP) and ad-hoc query-based data mining (OLAM), the standard data mining techniques provide knowledge that cannot be discovered interactively due to its computational complexity.

We propose running these methods ”off-line” as background processes with much lower priority than on-line operations. We call it suggestion in contrast to more or less hypothesis testing in OLAM. It works in idle (system) time when the user analyzes the on-line patterns.

The example of an off-line operation is at the end of the previous chapter – removing itemsets that prohibit creation of frequent itemsets among whole workspace. It is inverse suggestion of most relevant concept on an off-line sheet. This is very simple.

OLAM SE sorts the offline concepts by they're support (zero DB scans). After that it analyzes one by one offline concept with (idle) online rule. This process creates some frequent 2-itemsets (1 online, 1 offline) similarly to the online analysis. The offline workspace is then sorted by the new *suport*. After finishing these computations, OLAM SE continues computation for each subitem (sub-concept) of every offline class. In that manner also aggregated concepts are sorted by relevance and the available time. The off-line computation aborts any on-line operation.

So whenever the user adds the first frequented off-line (sub) concept, she might be sure that it is the best selection available at the moment. The other possibility is wait a while.

Characterization The previous part of the paper was concerning on frequent patterns mining from transactional data (e.g. N:N relation in ER diagram) that we can imagine as a sparse table. But there is something more about the data mining than association analysis of multilevel data in OLAM SE. It is characterization.

Suppose customer data (e.g. 1:N relation in ER). OLAM SE user creates a class containing all customers living in towns between 10 000 and 100 000 (using data warehouse – OLAP or an SQL query) and drags it to the on-line workspace. After some while she can see what products these customers most likely buy and other characteristics like how old are they. It works without any other action. Just watching.

Classification An easy classification (Bayesian) can be performed in the similar way. The user drags & drops some concepts from off-line to on-line sheet and puts other products to the litter bin so that only some concepts (classes) are present in off-line workspace(s). The OLAM SE without any other user action determines what classes (customers) buy those products. Investigation of places of living with highest consume is thus trivial.

Technically it is an improved naïve Bayessian MAP (maximum a posteriori) classification [7]. Suppose that x is an (on-line) observation and y an unknown (off-line) class:

$$\begin{aligned} *estimation(x) &= \arg\max_y P(x|y)P(y) \\ P(x|y)P(y) &= \frac{P(y \cap x)}{P(y)} P(y) = P(y \cap x) = support(y, x) \\ estimation(x) &= \arg\max_y support(y, x) \end{aligned}$$

In such a way it appears that maximizing the support is the optimal Bayesian classification. And the improvement to the naïve Bayessian classification is that OLAM SE doesn't have to presume that observations are independent [7]. They are interrelated using association rules. So OLAM SE incidentally includes the not-naïve Bayessian classification what is quite rare. In addition it works without any user action.

3.4 Litter Bin as a Threshold Interactivity Improvement

The straightest way how to shift (decrease) the *cover* and (increase) the *obviosity* threshold is to drag & drop the hidden concept to the working sheet from the litter bin, illustrated in Figure 4. After confirmation a dialog appears – it may take a long time, the appropriate parameter automatically shifts to the corresponding threshold. It is performed using InverseHuffman algorithm. If a user doesn't confirm the dialog, the itemset is added but the threshold is not shifted.

Decreasing those thresholds is also intuitive. You can remove the unnecessary concept from the workspace in two ways. The first way is moving the concept to the off-line part. The second is deleting it from the sheet – dropping into the litter bin. The dialog asks whether to hide it or to decrease the affected threshold, which may last a long time.

3.5 OLAM SE Architecture

OLAM SE consists of four layers. The lowest layer is database management layer. It provides access to data sources such as databases or XML documents. Next layer is MDDB (Multidimensional Database) layer. This layer provides us multidimensional view on data. It can use metadata describing conceptual hierarchy of data. Third layer is OLAM SE layer. This is a crucial layer for data mining. It works over data with implemented methods and algorithms. The last layer is GUI (Graphical User Interface) layer which provides interactivity between user and OLAM SE and it enables construction of constraint based knowledge.

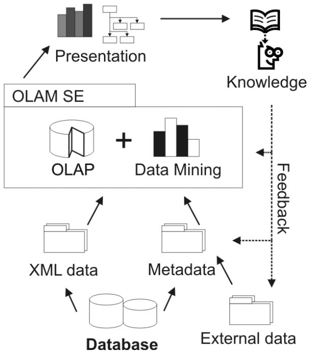


Fig. 5. OLAM SE Architecture

4 Conclusion

The data mining tasks are very useful in selective marketing, decision analysis, business management and many other areas. We have focused on multi-level frequent pattern analysis, which provides us an information about interesting relationships between data sets on the same or different levels of given conceptual hierarchy.

Our OLAM SE system provides user simplification of processing and understanding huge amounts of data. We use interactivity principles of OLAP to user-driven processing of input data. We have established two parameters - *cover* and *obviosity*. The *cover* parameter is based on Paretto analysis and entropy coding to determine interesting patterns. It's leading to the lossy compression of data sets on each level of conceptual hierarchy. The second parameter is the *obviosity*. On the basis of this parameter the frequent pattern with low information gain are moved to the litter bin.

Our method works in two modes. In online mode it is processed interactive and fast mining of knowledge. In the offline mode the data is processed with data mining algorithms with high computation complexity but iteratively – depending on how much time the user has.

The main goal is that the user that know the data doesn't have to know OLAP or data mining techniques – it is either characterization, not-naive Bayessian classification, frequent pattern analysis, association and correlation rules mining nor the appropriate thresholds and parameters.

We are hard working on the implementation of OLAM SE system at the moment, to see the experimental results of the quality and performance of proposed algorithms.

References

1. J. HAN and Y. FU: Discovery of Multiple-Level Association Rules from Large Databases. Proc. of 1995 Int'l Conf. on Very Large Data Bases (VLDB'95), citeseer.ist.psu.edu/han95discovery.html 420–431 Zürich, Switzerland, September (1995)
2. J. HAN and M. KAMBER: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers (1995)
3. J. HAN and M. KAMBER: Data Mining: Concepts and Techniques. Boston, Morgan Kaufmann Publishers (2006)
4. F. BARTES: Quality management, Zdenek Novotny pub., Brnom (2006)
5. WARNER: Data mining techniques. http://www.statsoft.com/textbook/stdatmin.html (2006)
6. J. HAN: Towards On-Line Analytical Mining in Large Databases. ACM Special Interest Group on Management of Data, SIGMOD Record urlhttp://www-faculty.cs.uiuc.edu/ hanj/pdf/sum98.pdf, (1998)
7. P. CHMELAR: Bayesian Concepts for Human Tracking and Behavior Discovery. Student EEICT 2006, Vol. 4, 360-364, Brno, CZ, VUT v Brn (2006)
8. MultiLevel Association Rule Mining An Object Oriented Approach based on Dynamic Hierarchies. http://citeseer.ist.psu.edu/fortin96multilevel.html (2006)
9. W3C: EXtensible Markup Language. http://www.w3.org/XML/ (2006)
10. UML: Unified Modeling Language. http://www.uml.org (2006)
11. H. ZHU: On-Line Analytical Mining of Association Rules. Burnaby University, Burnaby, British Columbia V5A 1S6, Canada citeseer.ist.psu.edu/zhu98line.html (2003)
12. L. STRYKA: Association Rules Mining Modul (in czech). Brno. (2003)
13. J. HAN and et al: DBMiner: A System for Mining Knowledge in Large Relational Databases. Proc. 1996 Int'l Conf. on Data Mining and Knowledge Discovery (KDD'96), Portland, Oregon, 250–255" "http://citeseer.ist.psu.edu/article/han96dbminer.html (1996)
14. J.J.Juran: Juran's quality handbook. New York, NY [u.a.] : McGraw-Hill, 5. ed (1999)