1. Trends in Data Mining and Knowledge Discovery

Krzysztof J. Cios [1,3,4,5] and Lukasz A. Kurgan [2]

[1] University of Colorado at Denver, Department of Computer Science and Engineering, Campus Box 109, Denver, CO 80217-3364, U.S.A.
e mail: Krys.Cios@cudenver.edu
[2] University of Alberta, Department of Electrical and Computer Engineering, ECERF 2nd floor, Edmonton, AB T6G 2V4, Canada
e mail: lkurgan@ece.ualberta.ca
[3] University of Colorado at Boulder, Department of Computer Science, Boulder, CO, U.S.A.
[4] University of Colorado Health Sciences Center, Denver, CO, U.S.A.
[5] 4cData. LLC, Golden, CO, U.S.A.

Data Mining and Knowledge Discovery (DMKD) is one of the fast growing computer science fields. Its popularity is caused by an increased demand for tools that help with the analysis and understanding of huge amounts of data. Such data are generated on a daily basis by institutions like banks, insurance companies, retail stores, and on the Internet. This explosion came into being through the ever increasing use of computers, scanners, digital cameras, bar codes, etc. We are in a situation when rich sources of data, stored in databases, warehouses, and other data repositories, are readily available. This in turn causes big interest of business and industrial communities in the field of DMKD. What is needed is a clear and simple methodology for extracting the knowledge that is hidden in the data. In this chapter, an integrated DMKD process model based on the emerging technologies like XML, PMML, SOAP, UDDI, and OLE BD-DM is introduced. These technologies help designing flexible, semi-automated, and easy to use DMKD model. They enable the building of knowledge repositories. They allow for communication between several data mining tools, databases and knowledge repositories. They also enable integration and automation of DMKD tasks. The chapter describes a six-step DMKD process model, the above mentioned technologies, and their implementation details.

1.1. The Knowledge Discovery and Data Mining Process

Knowledge Discovery (KD) is a nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns from large collections of data [30]. One of the KD steps is Data Mining (DM). DM is the step that is concerned with the actual extraction of knowledge from data, in contrast to the KD process that is concerned with many other things like understanding and preparation of the data, verification and application of the discovered knowledge. In practice, however, people use terms DM, KD, and DMKD as synonymous.

The design of a framework for a knowledge discovery process is an important issue. Several researchers described a series of steps that constitute the KD process. They range from very simple models, incorporating few steps that usually include data collection and understanding, data mining, and implementation, to more sophisticated models like the nine-step model proposed by Fayyad et al. [31]. In this chapter we describe the six-step DMKD process model [18], [19]. The advantage of this model is that it is based on the industry-initiated study that led to the development of an industry- and tool-independent DM process model [26]. It has been successfully applied to several medical problem domains [18], [44], [47], [61].

1.1.1. XML: the key to unlocking Data Mining and Knowledge Discovery

One of technologies that can help in carrying out the DMKD process is XML (eXtensible Markup Language) - a standard proposed by the World Wide Web Consortium [12]. It is a subset of SGML that uses custom-defined tags [42]. XML allows for the description and storage of structured or semi-structured data and their relationships. One of the most important features of XML is that it can be used to exchange data in a platform-independent way. XML is easy to use since many off-the-shelf tools exist for the automatic processing of XML. From the DMKD point of view, XML is a key technology that helps:

- to standardize communication between diverse DM tools and databases. This may result in a new generation of DM tools that can communicate with a number of different database products.
- to build standard data repositories sharing data between different DM tools that work on different software platforms. This may help to consolidate the DMKD market and open it to new users and applications.
- to implement communication protocols between the DM tools. This may result in the development of DM toolboxes [45] that consist of different DM tools, developed by different companies, but able to communicate and provide protocols to extract consolidated, more understandable, more accurate, and more easily applicable knowledge.
- to provide a framework for the integration of and communication between different DMKD steps. The information collected during the domain and data understanding steps (see below) can be stored as XML documents. They can then be used in the data preparation and data mining steps as a source of information that can be accessed automatically, across platforms and across tools. In addition, the extracted knowledge can be stored using XML and PMML (Predictive Model Markup Language) documents. This may enable the automation of the sharing of the discovered knowledge between different domains and tools that utilize the discovered knowledge, as long as they are XML and PMML compliant.

Since DMKD is known to be a very complex process that only includes DM as one of the steps, the importance of XML's utility in automating and consolidating the DMKD process, as well as making it cross-platform and cross-tool, indicates that this technology should be widely used in the DMKD domain. A number of other XML goals defined by the W3C, like the support of a wide variety of applications, ease of writing programs that process XML documents, human-legibility of XML documents, quick design of an XML document, all provide additional evidence for the usefulness of XML technology. XML is one of the most important technologies currently revolutionizing a number of fields, including DMKD.

This chapter describes a six-step DMKD process, provides examples of applications of the model, and discusses its relation to other DMKD models. Later, new technologies, like XML, XML-RPC, PMML, SOAP, UDDI, OLE BD-DM and DM methods and tools are described. The differences between the methods and tools in the context of the DMKD process are also discussed along with the comments on future of DMKD.

1.1.2. Why Data Mining and Knowledge Discovery?

DMKD was brought into attention in 1989 during the IJCAI Workshop on Knowledge Discovery in Databases (KDD) [54]. The workshops were then continued annually until 1994. In 1995, the International Conference on Knowledge Discovery and Data Mining became the most important annual event for DMKD. The framework of DMKD was outlined in two books: „Knowledge Discovery in Databases“ [55] and „Advances in Knowledge Discovery and Data Mining“ [30]. DMKD conferences like ACM SIGKDD, SPIE, PKDD and SIAM, and journals like Data Mining and Knowledge Discovery Journal (1997), Journal of Knowledge and Information Systems (1999), and IEEE Transactions on Knowledge and Data Engineering (1989) have become an integral part of the DMKD field.

In spite of the theoretical advances in DMKD, it is not easy to describe the current status of the field because it changes very quickly. We try here to describe the status of the DMKD field based on the web-based online research service Axiom® [6]. The Axiom service provides access to INSPEC, Compendex®, PageOne™ and the Derwent World Patents Index databases. It can find research papers using a user-specified set of keywords and a time frame. To analyze past developments, current state and future directions of the DMKD field we performed several queries that are summarized in Figures 1 and 2.

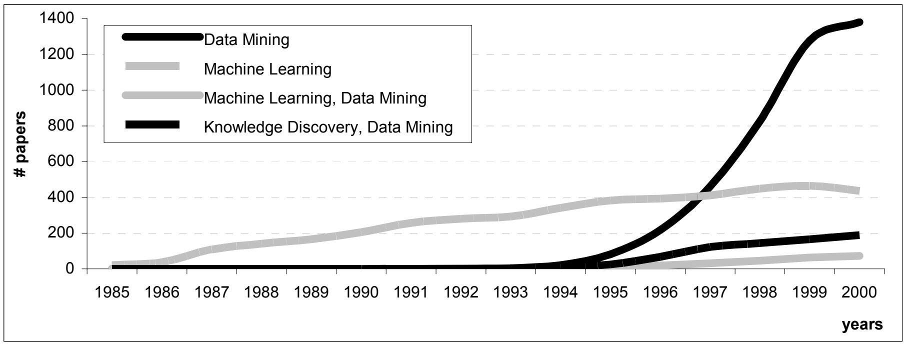

Fig. 1. Evolution of Data Mining and Data Mining and Knowledge Discovery fields

The DM revolution started in the mid 1990's. It was characterized by fast growth, as evidenced by the increase over a 5-year period of the number of DM papers from about 20 to about 1270. One of the reasons for that growth was due to the incorporation of existing tools and algorithms into the DM framework. The majority of the DM tools, e.g. machine learning (ML), were already well established. Figure 1 shows number of ML papers in the context of DM papers. The number of papers covering both ML and DM grows slowly; in 2000 there were 74 such papers, which constituted 6% of the entire DM research. The DMKD field emerged around 1995. In 2000 it constituted 15% of all DM research. This does not necessarily mean that only this percentage of the research is devoted to DMKD since some people still treat DM and DMKD as one and the same.

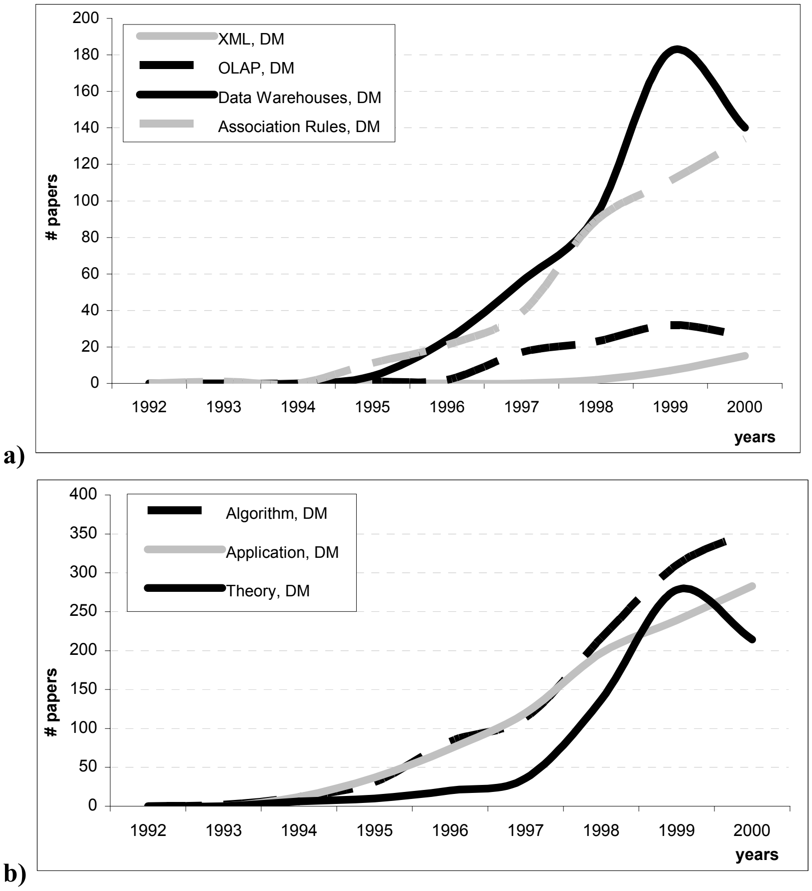

Fig. 2. a) Trends in Data Mining b) Data Mining theory and applications

The trends in DMKD over the last few years include OLAP, data warehousing, association rules, high performance DMKD systems, visualization techniques, and applications of DM. The first three trends are summarized in Figure 2a. The research interest in association rules follows a pattern generally similar to that of the DM field. On the other hand, the research in OLAP (On-Line Analytical Processing) and data warehouses initially was growing, getting maximum attention around 1999. Our observation is that some of the trends that initially had the greatest impact on the DM field began to decline because the majority of the issues concerned with those areas may have been solved, and thus the attention shifted towards new areas and applications. Recently, new trends have emerged that have great potential to benefit the DMKD field, like XML and XML-related technologies, database products that incorporate DM tools, and new developments in the design and implementation of the DMKD process. Among these, XML technology may have the greatest influence on DMKD since it helps to tie DM with other technologies like databases or e-commerce. XML can also help to standardize the I/O procedures of the DM tools, which in turn will help to consolidate the DM market and help to carry out the DMKD process. Figure 2a shows that XML has gained increasing interest, and being a very young concept, the interest in this technology can explode within a very short time.

The other important DMKD issue is the relationship between theoretical DM research and DM applications, see Figure 2b. The number of DM application papers increased rapidly over the last few years. The growth rate of theoretical research was slower initially but accelerated around 1998. After 1999 the theoretical research in DM started to decline. This trend may indicate that much attention has been given to practical applications of DM while setting aside theoretical research. This situation calls for a more balanced approach since applications need to be well grounded in theory if a real progress is to be made. The research in DM algorithms (see Figure 2b) shows stable growth over time, which confirms that there is still more interest in DM applications than in theoretical studies and in the development of new DM algorithms. The situation is mostly caused by a strong demand for applied DM because of increased funding opportunities. Although this is positive, more effort should be extended to developing new DM tools that can handle huge amounts of textual data generated by the Internet, or that could extract knowledge from hypertext and images, as often encountered in medicine.

In a nutshell, DMKD is an exponentially growing field with strong emphasis on applications. It is important to note that the number of papers cited above is not as important as the trends they are pointing to.

1.2. The Six-Step Knowledge Discovery and Data Mining Process

The goal of designing a DMKD process model is to come up with a set of processing steps to be followed by practitioners when they execute their DMKD projects. Such process model should help to plan, work through, and reduce the cost of any given project by detailing procedures to be performed in each of the steps. The DMKD process model should provide a complete description of all the steps from problem specification to deployment of the results.

A useful DMKD process model must be validated in real-life business applications. One such initiative was taken by the CRISP-DM (CRoss-Industry Standard Process for Data Mining) group [72], [26]. Their design was based on the study supported by DaimlerChrysler (automotive, aerospace, telecommunication and consultancy company), OHRA (insurance company), NCR (data warehouse supplier), and Integral Solutions Limited, currently part of SPSS (developer of the DM tool Clementine). The project included two key ingredients of any DMKD process: databases and DM tools. The OHRA and DaimlerChrysler provided large-scale applications that were used to validate the DMKD process model. The goal of the project was to develop a DMKD process that would help to save project costs, shorten project time, and adopt DM as a core part of the business. As a result, the six-step DM process was developed: business understanding, data understanding, data preparation, modeling, evaluation, and deployment. They called the entire process a data mining process which was different from the term (DM) understanding in the U.S.

Another six-step DMKD process model [18] was based on the CRISP-DM model but it included the following differences and extensions:

- the entire process is called the DMKD process that resolved the confusing use of the DM term (it is just a step – not the process)
- the DM step is used instead of the modeling step. The DM step is concerned with actual extraction of knowledge from a large dataset.
- several new feedback mechanisms are introduced. The CRISP-DM model has only three major feedback sources, from data understanding to business understanding, from modeling to data preparation, and from evaluation to business understanding. Our experience shows that the new feedback mechanisms are as important as the old three [18], [44], [47], [61], see Figure 3.
- the model is able to communicate with other domains; the knowledge discovered for a domain may be applied to other domains.

The six-step DMKD process [18] is described as follows:

1. **Understanding the problem domain.**
   In this step one works closely with domain experts to define the problem and determine the project goals, identify key people, and learn about current solutions to the problem. It involves learning domain-specific terminology. A description of the problem including its restrictions is done. The project goals then need to be translated into the DMKD goals, and may include initial selection of potential DM tools.
2. **Understanding the data.**
   This step includes collection of sample data, and deciding which data will be needed including its format and size. If a background knowledge does exist some attributes may be ranked as more important. Next, we need to verify usefulness of the data in respect to the DMKD goals. Data needs to be checked for completeness, redundancy, missing values, plausibility of attribute values, etc.

3. **Preparation of the data.**

   This is the key step upon which the success of the entire knowledge discovery process depends; it usually consumes about half of the entire project effort. In this step, we decide which data will be used as input for data mining tools of step 4. It may involve sampling of data, running correlation and significance tests, data cleaning like checking completeness of data records, removing or correcting for noise, etc. The cleaned data can be further processed by feature selection and extraction algorithms (to reduce dimensionality), by derivation of new attributes (say by discretization), and by summarization of data (data granularization). The result would be new data records, meeting specific input requirements for the planned to be used DM tools.

4. **Data mining.**

   This is another key step in the knowledge discovery process. Although it is the data mining tools that discover new information, their application usually takes less time than data preparation. This step involves usage of the planned data mining tools and selection of the new ones. Data mining tools include many types of algorithms, such as rough and fuzzy sets, Bayesian methods, evolutionary computing, machine learning, neural networks, clustering, preprocessing techniques, etc. Detailed description of these algorithms and their applications can be found in [17]. Description of data summarization and generalization algorithms can be found in [22]. This step involves the use of several DM tools on data prepared in step 3. First, the training and testing procedures are designed and the data model is constructed using one of the chosen DM tools; the generated data model is verified by using testing procedures.

   One of the major difficulties in this step is that many off-the-shelf tools may not be available to the user, or that the commonly used tools may not scale up to huge volume of data. The latter is a very important issue. Scalable DM tools are characterized by linear increase of their runtime with the increase of the number of data points within a fixed amount of available memory. Most of the DM tools are not scalable but there are examples of tools that scale well with the size of the input data; examples include clustering [11], [32], [78], machine learning [34], [63], and association rules [2], [3], [70]. An overview of scalable DM tools is given in [33]. Most recent approach for dealing with scalability of DM tools is connected with the meta-mining framework. The meta-mining generates meta-knowledge from knowledge generated by data mining tools [67]. It is done by dividing data into subsets, generating data models for these subsets, and generation of meta-knowledge from these data models. In this approach small data models are processed as input data instead of huge amounts of the original data, which greatly reduces computational overhead [46], [48].

5. **Evaluation of the discovered knowledge.**

   This step includes understanding the results, checking whether the new information is novel and interesting, interpretation of the results by domain experts, and checking the impact of the discovered knowledge. Only the approved models (results of applying many data mining tools) are retained. The entire DMKD process may be revisited to identify which alternative actions could have been taken to improve the results. A list of errors made in the process is prepared.

6. **Using the discovered knowledge.**
This step is entirely in the hands of the owner of the database. It consists of planning where and how the discovered knowledge will be used. The application area in the current domain should be extended to other domains. A plan to monitor the implementation of the discovered knowledge should be created, and the entire project documented.

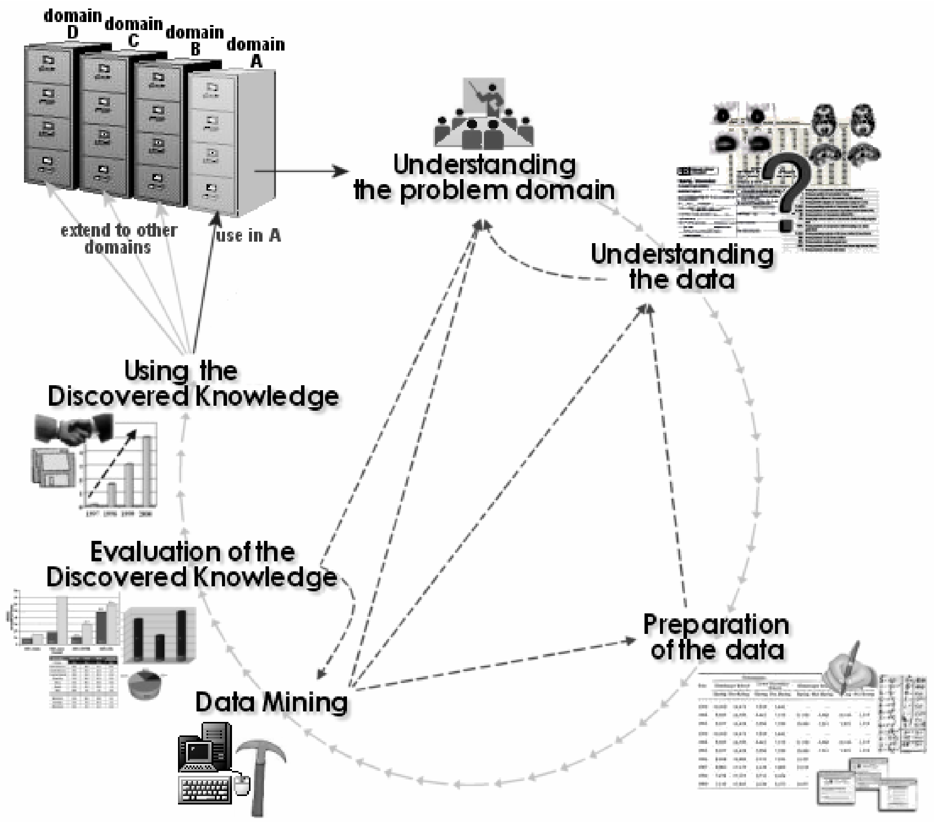

Fig. 3. The six-step DMKD process model

The DMKD process model just described is visualized in Figure 3. The important issues are the iterative and interactive aspects of the process. Since any changes and decisions made in one of the steps can result in changes in later steps, the feedback loops are necessary. The model identifies several such feedback mechanisms:

- from Step 2 to Step 1 because additional domain knowledge may be needed to better understand the data

- from Step 3 to Step 2 because additional or more specific information about the data may be needed before choosing specific data preprocessing algorithms (for instance data transformation or discretization)
- from Step 4 to Step 1 when the selected DM tools do not generate satisfactory results, and thus the project goals must be modified
- from Step 4 to Step 2 in a situation when data was misinterpreted causing the failure of a DM tool (e.g. data was misrecognized as continuous and discretized in Step 3). The most common scenario is when it is unclear which DM tool should be used because of poor understanding of the data.
- from Step 4 to Step 3 to improve data preparation because of the specific requirements of the used DM tool, which may have not been not known during the data preparation step.
- from Step 5 to Step 1 when the discovered knowledge is not valid. There are several possible sources of such a situation: incorrect understanding or interpretation of the domain, incorrect design or understanding of problem restrictions, requirements, or goals. In these cases the entire DMKD process needs to be repeated.
- from Step 5 to Step 4 when the discovered knowledge is not novel/interesting/useful. In this case, we may choose different DM tools and repeat Step 4 to extract new and potentially novel, interesting, and thus useful knowledge.

The feedback paths are shown as dashed lines in the Figure 3. The described feedback paths are by no means exhaustive.

Table 1. Comparison of three DMKD process models

6 step DMKD process [18]	**9 step DMKD process** [31]	**5 step DMKD process** [16]
1. Understanding the domain	1. Understanding application domain, identifying the DMKD goals	1. Business objective determination
2. Understanding the data	2. Creating target data set	2. Data preparation
3. Preparation of the data	3. Data cleaning and preprocessing	
	4. Data reduction and projection	
	5. Matching goal to particular data mining method	
	6. Exploratory analysis, model and hypothesis selection	
4. Data mining	7. Data mining	3. Data mining
5. Evaluation of the discovered knowledge	8. Interpreting mined patterns	4. Analysis of results
6. Using the discovered knowledge	9. Consolidating discovered knowledge	5. Knowledge assimilation

To evaluate the six-step DMKD process model and compare it with the nine-step DMKD process model [31] and the five-step model [16] we show in Table 1 shows the corresponding steps of the three models.

The common steps for the three models are domain understanding, data mining, and evaluation of the discovered knowledge. The Fayyad's model is very detailed and although it may provide the most guidance, it performs steps 5 and 6 too late in the process. We think that these steps should be performed during the step of understanding of the domain and understanding of the data steps, to guide the process of data preparation. In other words, the goal of the data preparation is to prepare the data to be used with the already chosen DM tools, while their model suggests that the DM tool is selected in Step 6, depending on the outcome of data preparation. This may cause problems when choosing a DM tool since the prepared data may not be suitable for the tool. Thus, an unnecessary feedback loop may be needed to change data preparation in Steps 2, 3 and 4. The Cabena model is very similar to the Cios model, except that the data understanding step is missing. The incompleteness of the Cabena model was pointed out by Hirji [39]. He used the Cabena model in a business domain and one of the conclusions was the necessity of adding one more step between data preparation and data mining, which he called data audit. Adding this step makes the Cabena model very similar to the Cios model. The latter has the advantage of being close to the CRISP-DM model that was validated on large real-life business applications. The model has been also used in several projects like a computerized system for diagnoses of SPECT bulleye images [18], creating and mining a database of cardiac SPECT images [61], creating an automated diagnostic system for cardiac SPECT images [44], and mining clinical information concerning cystic fibrosis patients [47].

The important characteristic of the DMKD process is the relative time spent to complete each of the steps. [16] estimates that about 20% of the effort is spent on business objective determination, about 60% on data preparation and about 10% for data mining and analysis of knowledge and knowledge assimilation steps, respectively. On the other hand, [10] show that about 15-25% of the project time is spent on the DM step. Usually it is assumed that about 50% of the project effort is spent on data preparation. There are several reasons why this step requires so much time: data collected by enterprise companies consist of about 1-5% errors, often the data are redundant (especially across databases) and inconsistent, also companies may not collect all the necessary data [57]. These serious data quality problems contribute to the extensive data preprocessing step. In a study at a Canadian fast-food company [39] it was shown that the DM step took about 45% of the total project effort, while data preparation took only about 30%. To accommodate for the above, we propose to use time ranges rather than fixed times for the steps, as shown in Figure 4.

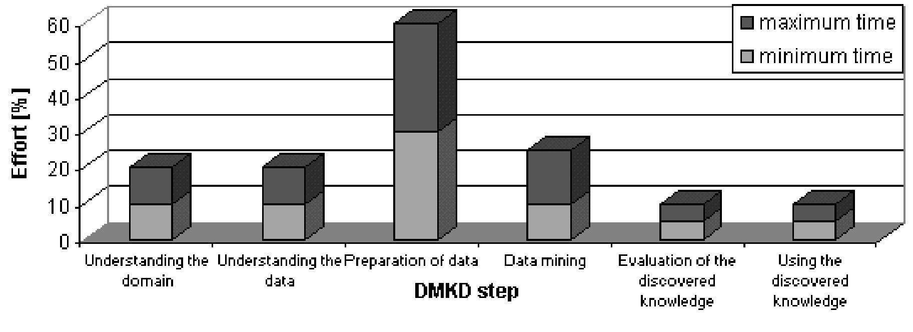

Fig. 4. Relative effort spent on each of the DMKD steps

A very important issue is how to carry out the DMKD process without possessing extensive background knowledge, without manual data manipulation and manual procedures to exchange data between different DM applications. The next two sections describe technologies that may help in automating the DMKD process and thus making its implementation easier.

1.3. New Technologies

Automating, or more realistically semi-automating, of the DMKD process is a very complex task. User input is always necessary to perform the entire DMKD task since only domain experts have the necessary knowledge about the domain and data. In addition, evaluation of results, at each DMKD step, is needed. To semi-automate the DMKD process several technologies are necessary:

- a data repository that stores the data, background knowledge, and models
- protocols for data and information exchange between data repositories and DM tools, and between different DM tools
- standards for describing the data and models.

XML has been studied and applied extensively over last few years. This technology, along with other technologies that are built on top of the XML, like PMML, XML-RPC, SOAP, and UDDI can provide solutions to the problem of semi-automating the DMKD process. In what follows, these technologies are introduced, and their applications within the DMKD process are described. Also, technologies like OLAP and OLE-DB DM and their impact on DMKD process are discussed.

XML

XML is a markup language for documents that contain structured information. Structured information consists of content (numbers, character strings, images, etc.) and information of what role that content plays, i.e. context of the information (e.g., a rule is built out of selectors, and a selector is a pair of attributes (name and value)).

XML defines a standard to add markup, or in other words to identify structures in documents.

XML is primarily used to create, share, and process information. XML enables users to define tags (element names) that are specific to a particular purpose. XML tags are used to describe the meaning or context of the data in a precisely defined manner. It is the information modeling features of XML that made it popular. Thanks to these features, processing of XML documents can be performed automatically.

XML technology is widely used in industry to transfer and share information. One of the most important properties of XML is that the current database management systems (DBMS) support the XML standard. From the DMKD point of view this means that XML can be used as a transport medium between DM tools and XML-based knowledge repositories, which are used to store discovered knowledge and information about the data, and the DBMS that store the data.

There are two major kinds of the DBMS that can handle XML documents: XML-native DBMS, and XML-enabled DBMS:

- The majority of *XML-native DBMS* are based on the standard DB physical storage model, like relational, object-relational, or object-oriented, but they use XML documents as the fundamental storage unit, just as relational DBMS uses tuples as its fundaments storage unit. Their main advantage lies in the possibility of storing an XML document and then retrieving the same document without losing any information, both on structural and data levels (not yet possible using the XML-enabled DBMS). The two well known XML-native DBMS are: Lore [49], and Tamino [62]. XML-native DBMSs can be divided into two groups: created over the relational model (examples include DBDOM, eXist, Xfinity and XML Agent), and created over the object oriented model (examples include eXcelon, X-Hive and ozone). There are also XML-native DBMS that are not built on neither relational nor object oriented model. They are schema independent, information centric, and are characterized by treating context as flexible as the data. Example of such DBMS is the NeoCore's XML database [50].
- The *XML-enabled DBMS* incorporates the XML document into the traditional database technology. Examples of commercial XML-enabled DBMSs (all use the relational model) are: Oracle 8i [7], DB2 [23], Informix [41], Microsoft SQLServer 2000 [68] and Microsoft Access2002 [74]. Since the above systems are used on a large scale in the business world they may become a dominant method for storing the XML documents.

  However, there are several problems associated with using XML-enabled DBMS. First, the existing models of storing XML documents do not fully preserve the context of the XML documents (e.g. the order of tags is lost). Second, some content, like comments or processing instructions of the XML document, is also lost. In contrast, any native XML DBMS preserves that information. The researchers already developed some solutions to the problem by proposing new schemas for storing XML documents within both relational and object-relational DBMS, which either use [9], [64], or do not use the Document Type Definition (DTD) documents. [65]

Another advantage of XML is the ability to query it to retrieve and manipulate data stored in the document. A number of query languages have been developed, including Lorel [1], Quilt [20], UnQL [14], XDuce [40], XML-QL [27], XPath [24], XQL [60], Xquery [21], and YaTL [25]. XPath and Xquery are two query languages that received the W3C recommendation.

XML-RPC

XML-RPC (XML-Remote Procedure Call) is a protocol that allows software running on disparate operating systems, and in different environments to make procedure calls over the Internet [75]. It uses HTTP as the transport and XML for the encoding. XML-RPC is designed to be as simple as possible to allow for the transmittal, processing, and return of complex data structures. XML-RPC implementations are available for virtually all operating systems, programming languages, dynamic and static environments, which include implementations in Perl, Python, Java, Frontier, C/C++, Lisp, PHP, Microsoft .NET, Rebol, Real Basic, Tcl, Delphi, WebObjects and Zope.

SOAP

SOAP (Simple Object Access Protocol) is another XML/HTTP based protocol for accessing services, objects, and servers on the Internet [66]. It is platform-independent. It consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined data types, and a convention for representing remote procedure calls and responses. SOAP is derived from XML-RPC, and it is a superset of XML-RPC, but they are not compatible.

From the DMKD point of view, both XML-RPC and SOAP can be used as protocols to communicate between DM tools in order to create DM toolboxes. Such toolboxes would use multiple DM tools, choosing ones that are suitable to work with the supplied data, and provide the user with combined results without the necessity of running the data separately using all chosen DM tools [45]. Using these protocols, the DM toolbox can access the DM tools over the Internet; as a results distributed and user-customized toolboxes can be easily built.

PMML

PMML (Predictive Model Markup Language) is an XML-based language used to define predictive data models and share them between compliant applications [56]. PMML was designed by the Data Mining Group (DMG) [29]. DMG is an independent, vendor-led group, which develops data mining standards; its members are IBM, Oracle, SPSS Inc., Angoss, MineIt Software Ltd., and others. PMML is supported by products from IBM, Oracle, SPSS, NCR, Magnify, Angoss, and other companies.

PMML defines the vendor-independent method for defining models. It removes the issues of incompatibility between applications and proprietary formats. This, in turn, enables exchanging models between applications. For example, it allows users to generate data models using one vendor application, and then use other vendor application to analyze, still another to evaluate the models, and yet another vendor application to visualize the model. This is yet another very important element that would enable building DM toolboxes. Previous solutions to the problem of sharing data models were incorporated into custom-built systems, and thus exchange of models with an application outside of the system was virtually impossible.

The PMML currently supports the following DM models: decision trees, naive Bayes models, regression models, sequence and association rules, neural networks, and center- and distribution-based clustering algorithms [29]. The PMML describes the models using eight modules: header, data schema, DM schema, predictive model schema, definition for predictive models, definition for ensemble of models, rules for selecting and combining models and ensembles of models, and rules for exception handling [36]. The PMML not only supports several DM models but also the ensemble of models and mechanisms for selecting and combining the models.

UDDI

Universal Description Discovery and Integration (UDDI) is a platform-independent framework for describing, discovering, and integrating services using the Internet and operational registry [71]. The framework uses XML, SOAP, HTTP and Domain Name System (DNS) protocols. Currently over 220 companies use the UDDI. The UDDI can help to solve problems like finding the right service from millions currently available, or interfacing with a service using Web Services Description Language (WSDL), which is the XML based format for describing network services [73]. At the same time, because of the benefits like reaching new customers, expanding offerings and market reach, it is almost certain that service providers will register their services using the UDDI.

From the DMKD point of view, services can be DM tools that are published as online services. DM toolboxes can be implemented as clients that can use those services [45]. The DM toolbox would then check availability of the online DM tools using UDDI, and invoke the ones that can provide meaningful results for the currently processed data. The DM tools (services) then would take the data provided by the DM toolbox, process it, and return results to the toolbox. Using this protocol, a DM toolbox can dynamically access and use several DM tools, which process data and generate results. The toolbox would collect the results, processes them, present them to the user, and finally store them in the knowledge base. This simple mechanism, which is powered by dynamic finding online DM tools, can be used to build flexible and widely applicable DM toolboxes.

The above technologies will certainly help in semi-automating the DMKD process. XML can be used to store data and PMML to store data models. SOAP and XML-RPC can be used for platform-independent communication between different DM applications, and UDDI can be used to find DM services offered by DM companies. A more detailed description of how to incorporate the technologies into the DMKD process is given below. A big advantage of the above technologies is that they are open source and thus can be freely downloaded and used.

OLAP

OLAP (On-Line Analytical Processing) is a relatively old DMKD technology. OLAP's main purpose is to provide users with multidimensional views of aggregate data for quick access to the needed information for further analysis. OLAP gives fast, consistent, interactive access to a variety of views of any information. OLAP and Data Warehouses (DW) are complementary technologies. A DW stores and manages data while OLAP transforms the data into possibly strategic information. OLAP services range from basic navigation and browsing (called "slice and dice") data, to analyses such as time series processing. OLAP gives the user some decision-making power. The most common applications of OLAP are marketing, promotions, customer analysis, sales forecasting, and market and customer segmentation. OLAP has the following characteristics:

- *Multidimensional views*, which help in analytical processing of the data through flexible access to information. Users can analyze data in any dimension and at any level of aggregation.
- *Time intelligence*, which means that OLAP systems can deal with the sequential nature of time. The notion of time should be built as an integral feature to any analytical package.
- *Complex calculations***,** which give the user a tool to, for instance perform share calculations (percentage of the total), allocations (which use hierarchies from a top-down perspective); they use trend algorithms such as moving averages and percentage growth.

One advantage of OLAP system is that it can be evaluated using a standardized set of benchmarks, for example the OLAP Council APB-1 performance benchmark simulates a realistic OLAP business situation [4]. The goal of the APB-1 is to measure an overall OLAP performance rather than the performance of specific tasks. The operations performed during APB-1 test include: bulk loading of data, incremental loading of data, aggregation of data along hierarchies, calculation of new data based on business models, time series analysis, queries with a high degree of complexity, drill-down through hierarchies, ad hoc queries, and multiple on-line sessions. In short, OLAP provides fast data summarization and basic data processing. It can be used as one of the preprocessing tools during the DMKD process, to make it more efficient and easier to perform. Also, OLAP technology can be directly integrated with a majority of other DM algorithms like association rules, classification, prediction and clustering [38].

OLAP is well coupled with DW because the data warehouses are designed differently than traditional relational DBMS. DW is a central data repository that defines integrated data model for data that is normally stored in a number of different locations. It incorporates a subject oriented, read-only historical data. This not only guarantees stability of the data but also gives flexibility to effectively query the data stored in a warehouse.

OLE DB-DM

OLE DB-DM (OLE DB for Data Mining) is an extension of the SQL query language that allows users to train and test DM models [51]. Its primary use is to integrate different DM tools by using a common API. The OLE DB-DM supports all of the most popular DM tools and applies DM analysis directly against a relational database. OLE DB-DM consists of these elements:

- *Data mining model* (DMM) is modeled by a relational table, except that it contains columns used for training and predictions. After the data are inserted into the table, a DM algorithm processes them and the resulting DM model is saved. Thus, the DMM can be browsed, refined, or used by a user.
- *Prediction join operation*, which is an operation that does a join query between a trained DM model and data to generate a prediction result that can be sent to the user's application as either an OLE DB rowset, or an ADO (Active Data Objects) recordset.
- *OLE DB-DM schema rowsets*, which allow user applications to find available DM services and models, and the model contents.

One of the advantages of the OLE DB-DM is its support of standard DM data types by using flags, instead of using only the SLQ data types. The following data types are supported:

- *key* - discrete attribute that is a key
- *continuous* – attribute with continuous values
- *discrete* - attribute with discrete values
- *discretized* - attribute is continuous and should be discretized
- *ordered* - attribute with discrete values that are ordered
- *cyclical* - attribute with discrete values that are ordered and cyclical, e.g. week days
- *sequence time* – attribute containing time measurement units
- *sequence* – attribute containing the sorting key of the related attributes

The OLE DB-DM supports the following DM models:

- classification when the predicted attribute is categorical
- regression when the predicted attribute is continuous
- clustering
- association (data summarization) including association rules
- sequence and deviation analysis
- dependency modeling that is used to identify dependencies among attributes

There are two advantages of the OLE DB-DM:

- it can interface with PMML, since all of the structure and content of a DMM may be expressed as an XML string in PMML format
- it can interface with the OLAP technology.

The above described technologies, (i.e. XML, PMML, XML-RPC, SOAP, UDDI, OLAP and OLE DB-DM) can be used to integrate and semi-automate the DMKD process, both on the level of manipulation and sharing of data and data models. XML based technologies can be used to store data and DM data models, and to provide communication protocols between DM tools. OLAP can be used during the data preprocessing step, and OLE DB-DM can be used to integrate DM tools with relational DBMSs.

1.4. The Future of DMKD: Tools and the DMKD Process

IDC, a well known provider of technology intelligence and industry analysis, estimates that the data mining tools market will reach $1.85 billion in 2006. In 1998, Simoudis of IBM, predicted that "within five years, [data mining] will be as important to running a business as the business systems are today". Business managers are willing to conduct DMKD in their data but they are not sure where to start [8].

The DMKD community developed several successful DM methods over the last few years. A survey of software implementations of DM methods presents a comparison of 43 existing implementations of DM methods [35]. Unfortunately, just having a variety of DM methods does not solve current problems of DMKD, like the necessity of integrating DM methods, integrating them with the DBMS, and providing support for novice users.

To provide a framework for addressing these issues we start by defining what DM methods and DM tools are. A DM method is simply an implementation of a DM algorithm, while a DM tool is a DM method that can communicate and operate in the DMKD environment. Development of DM tools, or upgrading the existing methods to tools, as well as improved integration of the entire DMKD process may solve the above problems. XML and XML-based technology provides tools for transforming DM methods into DM tools, of combining them into DM toolboxes, and most importantly for semi-automating the DMKD process. The DMKD research community recognizes the importance of XML technology for data preparation step, and as a medium to store, retrieve and use the domain knowledge via the use of PMML [15].

XML is a universal format for storing structured data. Since it is supported by current DBMSs it is becoming a standard not only for data transportation but also for data storage. The PMML language can be used to transmit and store metadata. It is one of the technologies that can substantially simplify the design of complete DMKD systems and increase their flexibility at the same time [36]. We thus predict the creation of metadata repositories (knowledge repositories) that would use the PMML format to store their content. SOAP and XML-RPC are two communication protocols that are not only platform-independent but also eliminate the need for direct API calls, make the communication easy, and support compatibility between applications that exchange data. Since these protocols are loosely coupled, one can communicate in the developer- and user-friendly manner; say, between applications written in C++ on the Linux operating system and another application written in

COBOL on the Windows platform. Traditional communication protocols based on COM, DCOM and CORBA models are tightly coupled, which makes development of the integration procedures very difficult, inefficient, and costly [5]. The SOAP communication protocol is seamless in terms of implementation because most of the software development packages already offer libraries that support this technology. As a result it is very easy to communicate between DM tools and the DM toolbox using these protocols. The UDDI is the key technology that enables building flexible DM toolboxes. By using it we can build online toolboxes that can dynamically search, access, and use DM tools that are published as web services. OLE DB-DM is the technology that allows the use of DM algorithms within the existing DBMSs products while avoiding problems of interfacing between DM tools and the DBMSs.

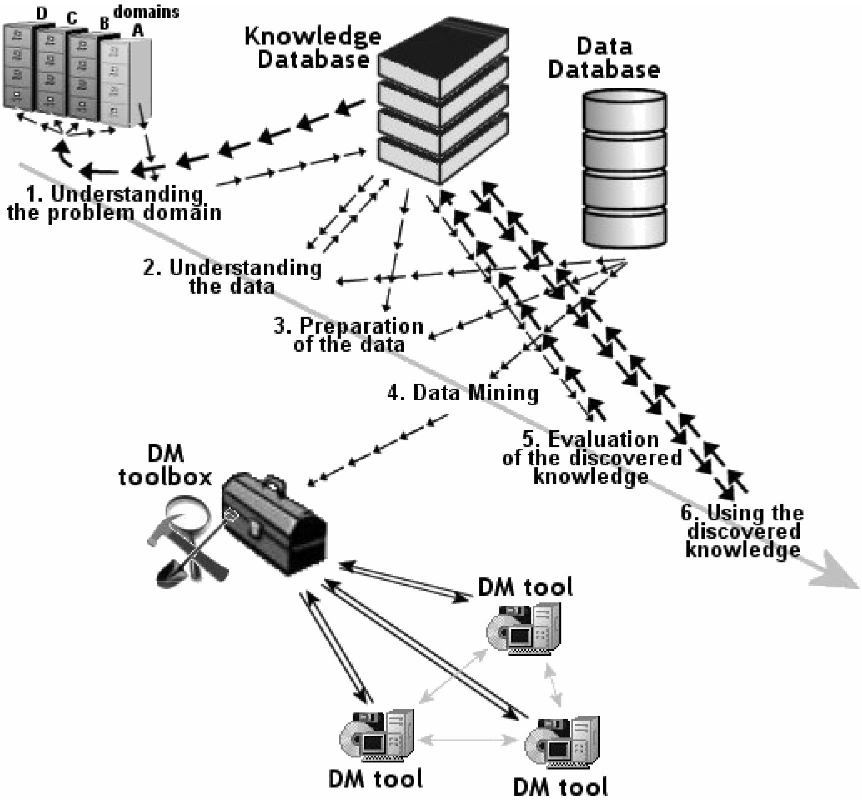
D
C
B
A
domains
Knowledge Database
Data Database
1. Understanding the problem domain
2. Understanding the data
3. Preparation of the data
4. Data Mining
5. Evaluation of the discovered knowledge
6. Using the discovered knowledge
DM toolbox
DM tool
DM tool
DM tool

Fig. 5. The automation of the DMKD process using XML based technologies

The above described technologies can, and we think will, be used to support all stages of the DMKD process. The diagram showing a design of the DMKD model based on these technologies, which supports semi-automation of the DMKD process, is shown in Figure 5.

The data database and knowledge database can be stored using a single DBMS that supports XML, since the PMML used to store the knowledge complies with the XML format. We separate the two to underscore the difference in format and functionality of the information they store. The data database is used to store and query the data. All of the DMKD steps, however, can store information and communicate using the knowledge database. The advantages of implementing the knowledge database are automation of the knowledge storage and retrieval, sharing of the discovered knowledge between different domains, and support for semi-automation of two DMKD steps: understanding of the data, and preparation of the data. The architecture shown in Figure 5 has the advantage of supporting the iterative and interactive aspects of the DMKD process. It simply makes a sense to support the entire DMKD process rather than only a single DM step.

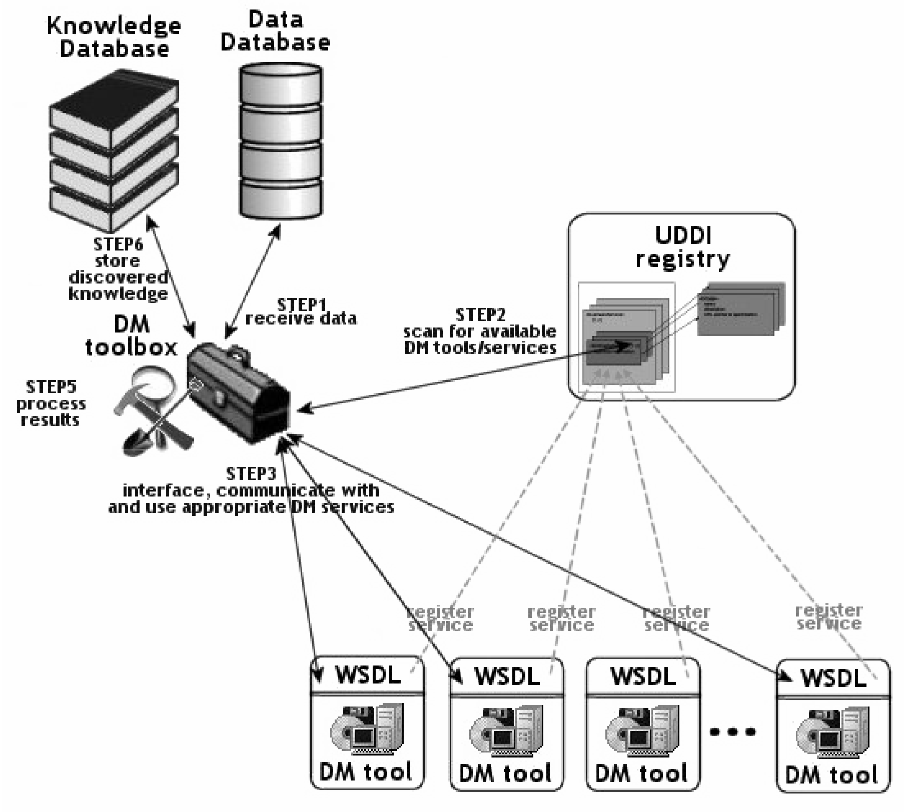

Fig. 6. DM toolbox architecture using Internet via HTTP, XML, SOAP or XML-RPC and UDDI

A DM step may use a DM toolbox that integrates multiple DM tools [45]. The DM toolbox architecture, based on XML, is shown in Figure 6.

The idea of implementing DM toolboxes arises from a simple observation that no single DM tool performs well on all types of data. XML and XML-based technology like SOAP and UDDI make the implementation of such toolboxes easy. First, the DM tools are registered as web services using the UDDI registry. The DM toolbox performs a series of steps to generate knowledge from data. It loads the data from a database and then using UDDI and WSDL descriptions it scans for DM tools that are available and suitable for particular analysis. Next, it communicates with the selected DM tools, provides them with the data, and receives the results of analyses. Finally, it processes the results and stores them in the knowledge database.

The business community already tries to integrate the DMKD process. During the last few –years businesses showed growing interest in DMKD. The biggest DBMS vendors like IBM, Microsoft and Oracle integrated some of the DM tools into their commercial systems. Their goal is to make the use of DM methods easier, in particular for users that use their DBMS products. IBM’s DM tool, called Intelligent Miner, which integrates with DB2, consists of three components: Intelligent Miner for Data, Intelligent Miner for Text, and Intelligent Miner Scoring [58], [59]. The Intelligent Miner for Data uses clustering based on Kohonen neural network, factor analysis, linear and polynomial regression, and decision trees, to find associations and patterns in data [28], [43], [17]. The Intelligent Miner for Text includes a search engine, Web access tools, and text analysis tools. Intelligent Miner Scoring is the DM component designed to work in real-time. The Intelligent Miner incorporates some data preprocessing methods like feature selection, sampling, aggregation, filtering, cleansing, and data transformations like principal component analysis [17]. It also supports the PMML format. Microsoft’s SQLServer2000 incorporates two DM algorithms: decision trees and clustering [69]. The implementation is based on the OLE DB-DM specification. Oracle has a DM tool called Oracle Darwin®, which is a part of the Oracle Data Mining Suite [52]. It supports DM algorithms like neural networks, classification and regression trees, memory-based reasoning (based on *k*-nearest neighbor approach), and clustering (based on *k*-means algorithm) [13], [17]. Their solution integrates with the Oracle 9i DBMS.

The above mentioned products provide tools to automate several steps of the DMKD process like preparation of the data and DM. However, they only partially solve the issue of semi-automation of the entire DMKD process because they do not provide an overall framework for carrying out the DMKD process.

1.5. Conclusions

At present the DMKD industry is fragmented. It consists of research groups and field experts which do not work closely with decision makers. This is caused by the situation where the DMKD community generates new solutions that are not widely accessible to a broader audience; the major obstacle being that they are very complex to use. Because of the complexity and high cost of the DMKD process, the DMKD projects are deployed in situations where there is an urgent need for them, while many other businesses reject it because of the high costs involved. To come up with the solution to this problem may require consolidation of the DMKD community by providing integrated DM tools and services, and making the DMKD process easier to implement by the end-users by semi-automating it.

The technologies described in the chapter (XML, XMP-RPC, SOAP, PMML, UDDI, OLAP and OLE DB-DM) will play a significant role in the design of the next-generation DMKD process framework. These technologies will make it possible to build DM toolboxes which span multiple DM tools, to build knowledge repositories, to communicate and interact between DM tools, DBMSs and knowledge repositories, and most importantly to semi-automate the entire DMKD process. These technologies also can be used to deploy the DMKD process that will include elements that will run on different platforms since they are platform-independent. Another advantage of these technologies is that they will bring the DMKD industry to a new level of usability. New users, who will follow these standards, in spite of their lack of knowledge of DMKD, will be exposed to and attracted to DMKD applications.

In addition to the design and implementation of a new DMKD framework, a more traditional course of action may have to be carried out [37]. It includes design and implementation of a new generation of high performance DM systems that incorporate multiple DM methods [76], and are capable of mining heterogeneous sources of knowledge like multimedia data [77], can visualize the results, and handle huge amounts of complex data. One of the goals in designing such systems should be the design of much better user interfaces. This will result in a wider acceptance of the products, particularly by midsize and small companies where users may have only limited technical skills. Another very important issue is to learn about the user perception of the novelty, understandability, and simplicity of the knowledge generated by the DMKD process. We must take into account the human cognitive processes and learn how people assimilate new knowledge to increase the usefulness of the new generation of DMKD tools [53], if we are to make progress. Such studies would greatly help to increase acceptance of DMKD tools.

In a nutshell, being able to model real problems in an easy to follow way is the major reason for designing the integrated DMKD process. Having such a process will help organizations to respond more quickly to market demands, increase revenues, operational efficiencies, and reduce costs.

Acknowledgements

The authors acknowledge comments of Harry Direen and Michael Trombley that helped to improve the readability of the article.

References

1. Abiteboul, S., Quass, D., McHugh, J., Widom, J., and Wiener, J., The Lorel Query Language for Semistructured Data, *International Journal on Digital Libraries*, 1:1, pp.68-88, 1997
2. Agrawal, R., and Srikant, R., Fast Algorithms for Mining Association Rules, *Proceedings of the 20th International Conference on Very Large Databases*, Santiago, Chile, 1994
3. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., and Verkamo, A.I., Fast Discovery of Association Rules, *Advances in Knowledge Discovery and Data Mining*, Chapter 12, AAAI/MIT Press, 1995
4. APB-1, *OLAP Council APB-1 OLAP Benchmark Release II*, http://www. olapcouncil.org/ research/bmarkco.htm, 1998
5. Apps, E., New Mining Industry Standards: Moving from Monks to the Mainstream, *PC AI*, 14:6, pp.46-50, 2000
6. AXIOM, http://axiom.iop.org/, 2001
7. Banerjee, S., Krishnamurthy, V., Krishnaprasad, M., and Murthy, R., Oracle8i - The XML Enabled Data Management System, *Proceedings of the Sixteenth International Conference on Data Engineering*, pp. 561-568, San Diego, California, 2000
8. Bauer C.J., Data Mining Digs In, Special Advertising Recruitment Supplement to The Washington Post, *Washington Post*, Sunday, March 15, 1998
9. Bourret, R., Bornhvd, C. and Buchmann, A.P., A Generic Load/Extract Utility for Data Transfer Between XML Documents and Relational Databases, *Proceeding of the 2nd International Workshop on Advanced Issues of E-Commerce and Web-based Information Systems*, pp.134-143, San Jose, California, June, 2000
10. Brachman, R., and Anand, T., The Process of Knowledge Discovery in Databases: A human-centered Approach, In Fayyad, U.M., Piatesky-Shapiro, G., Smyth, P., and Uthurusamy, R., (Eds), *Advances in Knowledge Discovery and Data Mining*, AAAi/MIT Press, 1996
11. Bradley, P., Fayyad, U., and Reina, C., Scaling Clustering Algorithms to Large Databases, *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, Menlo Park, California, pp. 9-15, 1998
12. Bray, T., Paoli, J., and Maler E., *Extensible Markup Language (XML) 1.0* (Second Edition), W3C Recommendation, http://www.w3.org/TR/2000/REC-xml-20001006, October 2000
13. Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J., *Classification and Regression Trees*, Wadsworth Int. Group, Belmont, California, USA, 1984
14. Buneman, P., Fernandez, M.F., Suciu, D., UnQL: A Query Language and Algebra for Semistructured Data Based on Structural Recursion, *Very Large Data Bases Journal*, 9(1), pp.76-110, 2000
15. Büchner, A.G., Baumgarten, M., Mulvenna, M.D., Böhm, R., and Anand, S.S., Data Mining and XML: Current and Future Issues, *Proceedings of the First International*

Conference on Web Information Systems Engineering (WISE'00), pp127-131, Hong Kong, 2000

16. Cabena, P., Hadjinian, P., Stadler, R., Verhees, J., Zanasi, A., *Discovering Data Mining: From Concepts to Implementation*. Perentice Hall, 1998
17. Cios, K.J., Pedrycz, W., Swiniarski, R., *Data Mining Methods for Knowledge Discovery*, Kluwer, 1998
18. Cios, K.J., Teresinska, A., Konieczna, S., Potocka, J., Sharma, S., Diagnosing Myocardial Perfusion from PECT Bull's-eye Maps - A Knowledge Discovery Approach, *IEEE Engineering in Medicine and Biology Magazine*, Special issue on Medical Data Mining and Knowledge Discovery, 19:4, pp. 17-25, 2000
19. Cios, K.J. (ed.), *Medical Data Mining and Knowledge Discovery*, Springer, 2001
20. Chamberlin, D., Robie, J., and Florescu, D., Quilt: An XML Query Language for Heterogeneous Data Sources, *Proceedings of Third International Workshop on the Web and Databases (WebDB 2000)*, Dallas, Texas, volume 1997 of *Lecture Notes in Computer Science*, 2000
21. Chamberlin, D., Clark, J., Florescu, D., Robie, J., Siméon, J., Stefanescu, M., *XQuery 1.0: An XML Query Language*, W3C Working Draft, http://www.w3.org/TR/xquery/, 2001
22. Chen, M.S., Han, J., and Yu, P.S., Data Mining: An Overview from Database Perspective, *IEEE Transactions on Knowledge and Data Engineering*, 8:6, pp. 866-883, 1996
23. Cheng, J., and Xu, J., IBM DB2 Extender, *Proceedings of the Sixteenth International Conference on Data Engineering*, pp. 569-573, San Diego, California, 2000
24. Clark, J., and DeRose, S., *XPath: XML Path Language* (Version 1.0), W3C Recommendation, http://www.w3.org/TR/xpath, 1999
25. Cluet, S., and Simeon, J., *YATL: a Functional and Declarative Language for XML*, draft manuscript, 2000
26. CRISP-DM, *CRoss-Industry Standard Process for Data Mining*, www.crisp-dm.org, 2001
27. Deutsch, A., Fernandez, M., Florescu, D., Levy, A., and Suciu, D., *XML-QL: A Query Language for XML*, W3C Note, http://www.w3.org/TR/NOTE-xml-ql/, 1998
28. Dillon, W.R., and Goldstein, M., *Multivariate Analysis: Methods and Applications*, New York: Wiley, 1984
29. DMG, *The Data Mining Group*, http://www.dmg.org/, 2001
30. Fayyad, U.M., Piatesky-Shapiro, G., Smyth, P., and Uthurusamy, R., *Advances in Knowledge Discovery and Data Mining*, AAAi/MIT Press, 1996
31. Fayyad, U.M., Piatetsky-Shapiro, G., and Smyth, P., Knowledge Discovery and Data Mining: Towards a Unifying Framework, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD96)*, Portland, OR. AAAI Press, 1996

32. Ganti, V., Ramakrishnan, R., Gehrke, J., Powell, A.L., French, J.C., Clustering Large Datasets in Arbitrary Metric Spaces. *Proceedings of the 15th International Conference on Data Engineering (ICDE)*, pp.502-511, 1999

33. Ganti, V., Gehrke, J., and Ramakrishnan, R., Mining Very Large Databases, *IEEE Computer*, 32:8, pp.38-45, 1999

34. Gehrke, J., Ramakrishnan, R., and Ganti, V., RainForest - a Framework for Fast Decision Tree Construction of Large Datasets, *Proceedings of the 24th International Conference on Very Large Data Bases*, San Francisco, pp. 416-427, 1998

35. Goebel, M., and Gruenwald, L., A Survey of Data Mining Software Tools, *SIGKDD Explorations*, 1:1, pp. 20-33, 1999

36. Grossman, R.L., Bailey, S., Ramu, A., Malhi, B., Hallstrom, P., Pulleyn, I., and Qin, X., The Management and Mining of Multiple Predictive Models Using the Predictive Modeling Markup Language, *Information and Software Technology*, 41:9, pp 589-595, 1999

37. Han, J., Data Mining: Where Is It Heading? (Panel abstract), *Proceedings of the 1997 International Conference on Data Engineering* (ICDE'97), Birmingham, England, pp. 462, 1997

38. Han, J., OLAP Mining: An Integration of OLAP with Data Mining, In S. Spaccapietra and F. Maryanski (Eds.), *Data Mining and Reverse Engineering: Searching for Semantics*, Chapman & Hall, pp. 3-20, 1998

39. Hirji, K.K., Exploring Data Mining Implementation, *Communications of the ACM*, 44:7, pp. 87-93, July 2001

40. Hosoya, H., and Pierce, B.C., XDuce: A Typed XML Processing Language, *Proceedings of Third International Workshop on the Web and Databases (WebDB 2000)*, Dallas, Texas, vol. 1997 of *Lecture Notes in Computer Science*, pp. 226-244, 2000.

41. Informix Object Translator, http://www.informix.com/idn-secure/webtools/ot/, 2001

42. ISO, ISO 8879:1986. *Information processing - Text and office systems - Standard Generalized Markup Language* (SGML), 1986

43. Kohonen, T., *Self-Organisation and Associative Memory*, Springer-Verlag, (Berlin) Springer Series in Information Sciences 8, 1989

44. Kurgan, L., Cios, K.J., Tadeusiewicz, R., Ogiela, M. and Goodenday, L.S., Knowledge Discovery Approach to Automated Cardiac SPECT Diagnosis, *Artificial Intelligence in Medicine*, 23:2, pp. 149-169, 2001

45. Kurgan, L., Cios, K.J., and Trombley, M., The WWW Based Data Mining Toolbox Architecture, *Proceedings of the 6th International Conference on Neural Networks and Soft Computing*, pp. 855-860, Zakopane, Poland, 2002

46. Kurgan, L., and Cios, K.J., DataSqueezer Algorithm that Generates Small Number of Short Rules, submitted to the *IEE Proceedings: Vision, Image and Signal Processing,* 2002

47. Kurgan, L., Cios, K.J., Sontag, M., and Accurso, F.J., Mining a Cystic Fibrosis Database, In: Zurada, J., and Kantardzic, M. (Eds.), *Novel Applications in Data Mining*, submitted, 2003

48. Kurgan, L., *Meta Mining System for Supervised Learning*, Ph.D. dissertation, University of Colorado at Boulder, Department of Computer Science, 2003

49. McHugh, J., Abiteboul, S., Goldman, R., Quass, D., and Widom, J., Lore: a Database Management System for Semistructured Data, *SIGMOD Record*, 26:3, pp.54-66, 1997

50. Native XML DBMS, http://www.rpbourret.com/xml/XMLDatabaseProds.htm, 2001

51. OLE DB-DM, *OLE DB for Data Mining Specification*, version 1.0, Microsoft Corporation, http://www.microsoft.com/data/oledb/dm.htm, July 2000

52. Oracle Data Mining Suite, Oracle Darwin®, http://technet.oracle.com/products/datamining/htdocs/datasheet.htm, 2001

53. Pazzani, M.J., Knowledge discovery from data?, *IEEE Intelligent Systems*, pp.10-13, March/April 2000

54. Piatesky-Shapiro, G., Knowledge Discovery in Real Databases: A Report on the IJCAI-89 Workshop, *AI Magazine*, 11:5, pp. 68-70, Jan. 1991

55. Piatetsky-Shapiro, G., and Frawley, W., (Eds), *Knowledge Discovery in Databases*, AAAI/MIT Press, 1991

56. PMML*, Second Annual Workshop on the Predictive Model Markup Language*, San Francisco, CA, August, 2001

57. Redman, T.C., The Impact of Poor Data Quality on the Typical Enterprise, *Communications of the ACM,* 41:2, pp.79-81, 1998

58. Rennhackkamp, M., IBM's Intelligent Family, *DBMS Magazine*, http://www.dbmsmag.com/9808d17.html, August 1998

59. Reinschmidt, J., Gottschalk, H., Kim, H., and Zwietering, D., *Intelligent Miner for Data: Enhance Your Business Intelligence*, IBM International Technical Support Organization (IBM Redbooks), IBM Corporation, 1999

60. Robie, J., Lapp J., Schach D., XML Query Language (XQL), *The Query Languages Workshop (QL 1998)*, Boston, Massachussets, 1998

61. Sacha, J.P., Cios, K.J., and Goodenday, L.S., Issues in Automating Cardiac SPECT Diagnosis. *IEEE Engineering in Medicine and Biology Magazine*, special issue on Medical Data Mining and Knowledge Discovery, 19:4, pp. 78-88, 2000

62. Schoening, H., Tamino-a DBMS Designed for XML, *Proceedings of the Seventeenth IEEE International Conference on Data Engineering*, pp.149-154, Los Alamos, CA, USA, 2001

63. Shafer, J., Agrawal, R., and Mehta, M., SPRINT: A Scalable Parallel Classifier for Data Mining, *Proceedings of the 22nd International Conference on Very Large Data Bases*, San Francisco, pp. 544-555, 1996

64. Shanmugasundaram, J., Gang, H., Tufte, K., Zhang, C., DeWitt, D.J., Naughton, J.F., Relational Databases for Querying XML Documents: Limitations and Opportunities. *Proceedings of 25th International Conference on Very Large Data Bases*, pp.302-314, 1999

65. Shimura, T., Yoshikawa, M. and Uemura, S., Storage and Retrieval of XML Documents using Object-Relational Databases, *Proceedings of the 10th International Conference on Database and Expert Systems Applications*, Lecture Notes in

Computer Science, Vol. 1677, Springer-Verlag, pp. 206-217, August-September 1999

66. SOAP 1.1, W3C Note, http://www.w3.org/TR/SOAP/, May 2000
67. Spiliopoulou, M., and Roddick, J. F., Higher Order Mining: Modelling and Mining the Results of Knowledge Discovery, *Data Mining II – Proceedings of the Second International Conference on Data Mining Methods and Databases*, Cambridge, UK, pp. 309-320, 2000
68. SQL Server Magazine, *SQL Server Magazine: The XML files*, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnsqlmag2k/html/TheXMLFiles.asp, 2000
69. Tang, Z., and Kim, P., *Building Data Mining Solutions with SQL Server 2000, DM Review*, White Paper Library, http://www.dmreview.com/whitepaper/wid292.pdf, 2001
70. Toivonen, H., Sampling Large Databases for Association Rules, *Proceedings of the 22nd International Conference on Very Large Data Bases*, San Francisco, pp. 134-145, 1996
71. UDDI, Universal Description, Discovery, and Integration (UDDI) specification, version 2.0, http://www.uddi.org/, 2001
72. Wirth, R., and Hipp, J., CRISP-DM: Towards a Standard Process Model for Data Mining, *Proceedings of the Fourth International Conference on the Practical Applications of Knowledge Discovery and Data Mining*, pp. 29-39, Manchester, UK, 2000
73. WSDL, Web Services Description Language (WSDL) 1.1, W3C Note, http://www.w3.org/ TR/wsdl, March 2001
74. XML and Access2002, *Exploring XML and Access 2002*, http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnacc2k2/html/odc_acxmllnk.asp, 2001
75. XML-RPC, *UserLand Software*, Inc., http://www.xmlrpc.com/, 2001
76. Yaginuma, Y., High-performance Data Mining System, *Fujitsu Scientific and Technical Journal*, special issue on *Information Technologies in the Internet Era*, 36:2, pp.201-210, 2000
77. Zaïane, O.R., Han, J., Li, Z.N., Hou, J., Mining Multimedia Data, *Proceedings of the CASCON'98: Meeting of Minds*, Toronto, Canada, 1998, pp. 83-96, 1998
78. Zhang, T., Ramakrishnan, R., Livny, M., BIRCH: An Efficient Data Clustering Method for Very Large Databases, *Proceedings of the SIGMOD Conference*, pp.103-114, 1996